Representing, Reasoning and Answering Questions about Biological Pathways

Various Applications

by

Saadat Anwar

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved November 2013 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Katsumi Inoue
Yi Chen
Hasan Davulcu
Joohyung Lee

ARIZONA STATE UNIVERSITY

May 2014

ABSTRACT

Biological organisms are made up of cells containing numerous interconnected biochemical processes. Diseases occur when normal functionality of these processes is disrupted, manifesting as disease symptoms. Thus, understanding these biochemical processes and their interrelationships is a primary task in biomedical research and a prerequisite for activities including diagnosing diseases and drug development. Scientists studying these interconnected processes have identified various pathways involved in drug metabolism, diseases, and signal transduction, etc.

High-throughput technologies, new algorithms and speed improvements over the last decade have resulted in deeper knowledge about biological systems, leading to more refined pathways. Such pathways tend to be large and complex, making it difficult for an individual to remember all aspects. Thus, computer models are needed to represent and analyze them. The refinement activity itself requires reasoning with a pathway model by posing queries against it and comparing the results against the real biological system.

Many existing models focus on structural and/or factoid questions, relying on surface-level information. These are generally not the kind of questions that a biologist may ask someone to test their understanding of biological processes. Examples of questions requiring understanding of biological processes are available in introductory college level biology text books. Such questions serve as a model for the question answering system developed in this thesis.

Thus, the main goal of this thesis is to develop a system that allows the encoding of knowledge about biological pathways to answer questions demonstrating understanding of the pathways. To that end, a language is developed to specify a pathway and pose questions against it. Some existing tools are modified and used to accomplish this goal. The utility of the framework developed in this thesis is illustrated with applications in the biological domain.

Finally, the question answering system is used in real world applications by extracting pathway knowledge from text and answering questions related to drug development.

This dissertation is dedicated to my parents, my family, and especially my grand mother.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis adviser Dr. Chitta Baral for his support, guidance and patience throughout my degree. Dr. Baral's ability to find creative research dimensions and application areas is an inspiration for me. I would also like to thank my committee members for their time in supervising this dissertation. In particular, I want to thank Dr. Katsumi Inoue, Dr. Luis Tari, and Dr. Jörg Hakenberg for their collaboration in the research.

I would also like to thank my colleagues from Dr. Baral's lab, who were always ready to lend me an ear to bounce ideas, review presentations, and give valuable feedback. In this regard, I specifically thank Vo Ha Nguyen for his efforts.

None of this would have been possible without the support of my boss at work, Dr. Phil Christensen, who persuaded me to finish my degree and wholeheartedly supported my efforts. I thank my colleagues at my work place for taking over tasks and responsibilities in order to lighten my load.

Lastly, my words are not enough for my family for their support, understanding and patience in this long journey.

TABLE OF CONTENTS

CHAPTER

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Biological organisms are composed of cells that contain numerous interconnected and interacting biochemical processes occurring simultaneously. Disruptions in the normal functionality of these processes causes diseases, which appear as symptoms (of these diseases). As a result understanding these processes is a fundamental activity in the biological domain and is prerequisite for activities such as disease diagnosis and drug discovery. One aspect of understanding the biological systems is the identification of pathways responsible for drug metabolism, diseases, and signal transduction, etc. The availability of high throughput approaches like micro-arrays, improvements in algorithms, and hardware that have come online during the last decade has resulted in significant refinement in these pathways. The pathways have become much larger in size and complexity to the degree that it is not reasonable for one person to fully retain all aspects of the pathway. As a result, computer based models of pathways are needed that allow the biologists to ask questions against them and compare them with real-world knowledge. The model should be such that it has an understanding of the pathway. Such a system would be considered intelligent and would assist the biologists in expanding the breadth of their search for new drugs and diagnoses. Source knowledge for these pathways comes from volumes of research papers published every year. Though there are a number of curated pathway resources available, they significantly lag behind the current state of the research in biology. As a result, we need a way to extract this pathway information from published text.

1.1 Choosing the right questions

A large body of research exists on computer modeling of biological processes and it continues to be an active area of research. However, many such models focus on surface properties, like structure; or factoid questions. Though important, we feel these systems do not test the understanding of the underlying system being modeled. We want to go beyond this surface level information and answer questions requiring deeper reasoning. We want our system to answer questions that a biology teacher expects his / her students to answer after reading the required text. So, we turned to college level biological text books for the questions that we feel are more indicative of such understanding. Following questions from Reece *et al.* (2010) illustrate the kind of questions we are interested in answering:

- "What would happen to the rate of glycolysis if DHAP were removed from the process of glycolysis as quickly as it was produced?"
- "Membranes must be fluid to function properly. How would decreased fluidity of the membrane affect the efficient of the electron transport chain?"

These questions and others like it were the subject of a recent deep knowledge representation challenge[1]. In this thesis, we focus on questions that require reasoning over simulations.

1.2 Choosing the right tools

Data about biological systems can be qualitative or quantitative in nature. The fully quantitative data about reaction dynamics is based on ordinary differential equations (ODEs) of reaction kinetics, which are often lacking (Chaouiya, 2007). Qualitative data is more prevalent. It is less precise, but tends to capture general relationships

[1] https://sites.google.com/site/2nddeepkrchallenge/

between various components of a biological pathway. Adding quantitative information to a qualitative model provides the next step in refinement of the biological pathways (Heiner *et al.*, 2004), providing better coverage of biological systems and processes. We want to use this qualitative+quantitative data for our modeling.

To simulate and reason with the pathways, we need tools that can model a biological pathway that contains qualitative+quantitative information, simulate the pathway and reason with the results.

1.3 Need for a pathway a specification and a query language

Pathway information comes in various formats, such as cartoon drawings, formal graphical representations like Kohn's Maps (Kohn *et al.*, 2006), curated databases of pathways (Kanehisa and Goto, 2000; Karp *et al.*, 2002; Croft *et al.*, 2011) and free text. The depth of this knowledge as well as its taxonomy varies with the source. Thus, a common specification language is needed. Such a language must be easy to understand and must have a well defined semantics.

Queries are normally specified in natural language, which is vague. So, a more precise query language is needed. One could ask queries in one of the existing formal languages (Gelfond and Lifschitz, 1998), but that will be burdensome for a user to become fluent. As a result, we need a language that has a simple English-like syntax, but a well defined semantics, so that it does not have the vagaries of the natural language.

1.4 Text extraction

Knowledge about biological pathways is spread over collections of published papers as nuggets of information, such as relationships between proteins; between proteins and drugs; genetic variation; and association of population groups with genetic variation; to name a few. Published research may also contain contradictory information, e.g. an earlier conjecture that was proven to be untrue in later research, or knowledge with limited amount of certainty. To extract these nuggets and to assemble them into a coherent pathway requires background knowledge, similar to other technical fields. Portions of this knowledge are published in books and online repositories. Thus, we need a method of text extraction that allows one to extract nuggets of information, consult available databases and produce knowledge about pathway that is self-consistent.

1.5 Overview

In this thesis, we propose to build a system, called BioPathQA, to answer deeper reasoning questions using existing tools with modifications. To that end, we develop a language to specify pathways and queries. Our system is designed to answer reasoning questions requiring simulation. We demonstrate the applicability of our system with applications to drug development on knowledge obtained from text extraction.

To implement an answering system that can answer simulation based reasoning questions, we first looked for available tools that could help in this task and we found Petri Nets as providing the right level of formalism for our application. Petri Nets (Peterson, 1977) are a popular representation formalism used to model biological systems and to simulate them. They have been used to model and analyze the dynamic behavior as well as structural properties of biological systems. However, such analysis is usually limited to invariant determination, liveness, boundedness and reachability. To our knowledge they have not been used to answer questions similar to the aforementioned.

In order to represent deeper reasoning questions, we have to make extensions to the Petri Net model as the basic model lacks sufficient richness. For example, we may change the firing semantics to limit the state space or maximize parallel activity. Although numerous Petri Net modeling, simulation and analysis systems exist (Jensen *et al.*, 2007; Rohr *et al.*, 2010; Kounev *et al.*, 2006; Berthomieu* *et al.*, 2004; Nagasaki *et al.*, 2010; Kummer *et al.*, 2000), we found certain limitations in the default implementation of these systems that prevented us from using them as is. For example, the Colored Petri Net implementation CPNtools [2] does not allow inhibitor arcs (we use to model protein inhibition); Cell Illustrator (Nagasaki *et al.*, 2010) is closed source and does not support colored tokens (we use to model locations); Snoopy (Rohr *et al.*, 2010) supports a large number of extensions, but it is unclear how one exports the simulation results for further reasoning; and most did not allow exploring all possible state evolutions of a pathway, or different firing semantics.

To make these extensions in an easy manner we use Answer Set Programming (ASP) (Lifschitz, 2008) as the language to represent and simulate Petri Nets. It allows a simple encoding of the Petri Net and can be easily extended to incorporate extensions [3]. In addition, ASP allows powerful reasoning capability and the possibility of implementing additional constructs not supported by Petri Nets directly, such as the ability to filter trajectories.

Petri Net to ASP translation has been studied before (Behrens and Dix, 2007; Heljanko and Niemelä, 2000). However, these implementations have been limited to specific classes of Petri Nets and have different focus. For example, the Simple Logic Petri Nets (Behrens and Dix, 2007) do not allow numerical accumulation of the same tokens from multiple transitions to a single place and the Binary Petri Nets (Heljanko and Niemelä, 2000) do not allow more than one tokens at any place.

[2] http://cpntools.org

[3] Some commercial tools, like Cell Illustrator do allow exporting their model into a regular programming language, but we believe that declarative language describes the problem more succinctly.

1.6 Specific contributions

The research contribution of this thesis can be divided into four major parts. The first part gives a general encoding of Petri Nets in ASP, which allows easy extension by making local changes. The second part shows how the ASP encoding of Petri Nets can be used to answer simulation based reasoning questions. The third part describes the high-level language for pathway and query specification; and the system that we have developed to answer deep reasoning questions. The fourth part shows how knowledge is extracted from text of research papers, cleaned and assembled into a pathway to answer simulation based reasoning questions using our system.

1.6.1 General ASP encoding of Petri Net for simulation

Although previous work on encoding Petri Nets in ASP exists, it is limited to specific classes of Petri Nets. We present an encoding of a basic Petri Net in ASP to show it is done in an intuitive yet concise way. The default execution semantics of a Petri Net is the *set*-semantics, which allows a subset of transitions to fire simultaneously when ready. This can result in far too many combinations of transition firing arrangements. A simpler approach is to use the so called *interleaved* execution semantics, in which at most one transition fires when ready. This too can generate many firing arrangements. Biological systems are highly parallel in nature, as a result it is beneficial to model maximum parallel activity. So, we introduce a new firing semantics, called the *maximal firing set* semantics by extending the *set semantics*. In this semantics, a maximal subset of non-conflicting enabled transitions fire simultaneously when ready.

Then, we extend the basic ASP encoding to include Petri Net extensions like reset-arcs (to model immediate consumption of any amount of substrate), inhibit-arcs (to model gene/protein inhibition), read-arcs (to model additional pre-conditions of a reaction, such as different start vs. maintenance quantity of a reactant), colored-tokens (to model quantities of different types of substances at the same location), priority-transitions (to select between alternate metabolic paths), and timed-transitions (to model slow reactions) that allow modeling of various concepts in biological systems. We show how ASP allows us to make these extensions with small amount of local changes.

This component is one of the major focuses of our research. It is described in Chapter 2 and is the basis for implementation of our system to model pathways and answer questions about them.

1.6.2 Answering simulation based reasoning questions

We use the encoding developed in Chapter 2 to questions in (Reece *et al.*, 2010, Chapter 9) that were a part of the Second Deep Knowledge Representation Challenge [4]. These questions are focused on the mechanism of cellular respiration and test the understanding of the student studying the material; and appear in two main forms: (i) inquiry about change of rate of a process due to a change in the modeled system, and (ii) explanation of a change due to a change in the modeled system.

We built Petri Net models for the situations specified in the questions, encoded them in ASP and simulated them over a period of time. For change of rate questions, we computed the rate for both nominal and modified cases and observed that they matched the responses provided with the challenge questions. For the explanation of change questions, we collected the summary of firing transitions as well as substance quantities produced at various times. This information formed the basis of our answer. We compared our results with the answers provided with the challenge questions.

[4] https://sites.google.com/site/2nddeepkrchallenge/

A novel aspect of our approach is that we apply the initial conditions and interventions mentioned in the questions as modifications to the pathway representation. These interventions can be considered as a generalized form of actions.

For certain questions, additional domain knowledge outside the source material was required. We filled this gap in knowledge as necessary. We also kept the models to a subset of the pathway for performance as well as to reduce clutter in the output that can bury the results with unnecessary details.

This component of our research is described in Chapter 3.

1.6.3 BioPathQA: a system and a language to represent pathways and query them

We combined the techniques learned from Chapter 2, action languages, and biological modeling languages to build a question answering system that takes a pathway and a query as input. Both the pathway specification language and the query language have strict formal semantics, which allow them to be free of the vagaries of natural language, the language of the research papers as well as the query statements.

Since the biological pathways are constructed of biochemical reactions, they are effected by environmental changes. Mutations within the cell can also result in conditional change in behavior of certain processes. As a result, we needed actions with conditional effects. Our Petri Net model wasn't rich enough to model conditional actions, so we extended the Petri Nets with conditional arcs. We call this extension, the Guarded-Arc Petri Net, where a guard is a condition on an arc, which must be true for that arc to be traversed. With this extension, a Petri Net transition can have different outcomes for different markings. Our model is similar to the model in (Jensen *et al.*, 2007) in many aspects, with differences in certain key semantics related to biological modeling.

This component of our research is described in Chapter 4.

1.6.4 Text Extraction to Answer Questions about Real World Applications

To apply our system to real world applications, we have to extract pathway knowledge from published papers, which are published in natural language text. For this, we use a system called the Parse Tree Query Language (PTQL) (Tari *et al.*, 2012) to nuggets of information from the abstracts published on PubMed [5]. Sentences are parsed using the Link Grammar (Sleator and Temperley, 1993) or Stanford Parser (de Marneffe and Manning, 2008); with various object-classes identified within the sentence. Unlike Information Retrieval (IR) approaches that tend to treat documents as unstructured bags-of-words, PTQL treats words (or word-groups) as sentence elements with syntactic as well as dependency relationships between them. PTQL queries combine lexical, syntactic and semantic features of sentence elements. Thus with PTQL, one can ask for the first-noun of a noun-phrase that is the direct-object of a verb-phrase for some specific verb string. To accomplish its task, PTQL performs a number of pre-processing steps on its input useful for text extraction and leverages on various existing databases. These include sentence splitting, tokenization, part-of-speech (POS) tagging, named entity recognition, entity-mention normalization, cross-linking with concepts from external databases, such as Gene Ontology (Camon *et al.*, 2004) and UniProt (Consortium, 2012). We extract gene-gene, gene-drug, and gene-disease relationships using PTQL, assemble them into a pathway specification and reason with the extracted knowledge to determine possible drug interactions.

Facts and relationships extracted using PTQL are further subject to filtering to remove inconsistent information. A pathway specification is then constructed from the extracted facts, which can be queried using the query specification language. We

[5] http://www.ncbi.nlm.nih.gov/pubmed

illustrate the use of our deep reasoning system by an example from the drug-drug interaction domain.

This component is described in Chapter 5.

1.7 Summary

The main contributions of this thesis can be summarized as follows:

1. Generalized Petri Net encoding in ASP, including a new maximal firing set semantics (Chapter 2)
   - An easy to extend encoding is developed, that allows adding extensions using local changes
   - A new Petri Net firing semantics, the so called maximal firing set semantics is defined, which ensures maximum possible parallel activity at any given point
2. Answering simulation based deep reasoning questions using our ASP encoding (Chapter 3)
   - It is shown, how deep reasoning questions requiring simulation based reasoning can be answered.
3. Developed a system called BioPathQA and a language to specify biological pathways and answer deep reasoning questions about it (Chapter 4)
   - A pathway specification language is developed, combining concepts from Petri Nets, Action Languages, and Biological Pathways
   - A query specification language is developed, which is english like, with well defined semantics, avoiding the vagaries of Natural Language

- A description of our implementation using ASP and Python is given; and an execution trace is shown

4. Performed text extraction to extract pathway knowledge (Chapter 5)

    - It is shown pathway knowledge is extracted and used to answer questions in the drug-drug interaction domain

Chapter 2

PETRI NET ENCODING IN ASP FOR BIOLOGICAL DOMAIN

2.1 Introduction

Petri Net (Peterson, 1977) is a graphical modeling language with formal semantics used for description of distributed systems. It is named after Carl Adam Petri, who formally defined Petri Nets in his PhD thesis in the 1960's (Brauer and Reisig, 2006). Petri nets have been widely used to model a wide range of systems, from distributed systems to biological pathways. The main advantages of Petri Net representation include its simplicity and the ability to model concurrent and asynchronous systems and processes.

A variety of Petri Net extensions have been proposed in the literature, e.g. inhibitor arcs, reset transitions, timed transitions, stochastic transitions, prioritized transitions, colored petri nets, logic petri nets, hierarchical petri nets, hybrid petri nets and functional petri nets to a name a few (Behrens and Dix, 2007; Music, 2012; Hardy and Robillard, 2004).

Our interest in Petri Nets is for representing biological pathways and simulating them in order to answer simulation based reasoning questions. We show how Petri nets can be represented in ASP. We also demonstrate how various extensions of basic Petri nets can be easily expressed and implemented by making small changes to the initial encoding. During this process we will relate the extensions to their use in the biological domain. Later chapters will show how this representation and simulation is used to answer biologically relevant questions.

The rest of this chapter is organized as follows: We present some background material on Answer Set Programming (ASP) and Petri Nets. Following that, we present the Answer Set encoding of a basic Petri Net. After that we will introduce various Petri Nets extensions and the relevant ASP code changes to implement such extensions.

2.2 Background

2.2.1 Answer Set Programming

Answer Set Programming (ASP) is a declarative logic programming language that is based on the Stable Model Semantics (Gelfond and Lifschitz, 1988). It has been applied to a problems ranging from spacecrafts, work flows, natural language processing and biological systems modeling.

Although ASP language is quite general, we limit ourselves to language and extensions relevant to our work.

Definition 1 (Term) *A* term *is a term in the propositional logic sense.*

Definition 2 (Literal) *A* literal *is an atom in the propositional logic sense. A literal prefixed with* **not** *is referred to as a negation-as-failure literal or a* naf-literal*, with* **not** *representing negation-as-failure.*

We will refer to propositional atoms as *basic atoms* to differentiate them from other atoms, such as the aggregate atoms defined below.

Definition 3 (Aggregate Atom) *A* sum aggregate atom *is of the form:*

$$L\ [B_0 = w_0, \dots, B_m = w_m]\ U \tag{2.1}$$

where, B_i are basic atoms, w_i are positive integer weight terms, L, U are integer terms specifying the lower and upper limits of aggregate weights. The lower and upper limits are assumed to be $-\infty$ and ∞, if not specified.

A *count aggregate atom* is a special case of the *sum aggregate atom* in which all weights are 1, i.e. $L\ [B_0 = 1, \ldots, B_m = 1]\ U$ and it is represented by:

$$L\ \{B_0, \ldots, B_m\}\ U \tag{2.2}$$

A *choice atom* is a special case of the *count aggregate atom* (2.2) in which $n = m$.

Definition 4 (ASP Program) *An* ASP program Π *is a finite set of rules of the following form:*

$$A_0 \leftarrow A_1, \ldots, A_m, \textbf{not}\ B_1, \ldots, \textbf{not}\ B_n, C_1, \ldots, C_k. \tag{2.3}$$

where each A_0 is either a basic atom or a choice atom, A_i and B_i are basic atoms, C_i are aggregate atoms and **not** *is negation-as-failure.*

In rule (2.3), $\{A_0\}$ is called the *head* of the rule, and $\{A_1, \ldots,\ A_m,\ \textbf{not}\ B_1, \ldots,\ \textbf{not}\ B_n,\ C_1, \ldots,\ C_k\}$ is called its *tail.* A rule in which A_0 is a choice atom is called a *choice rule.* A rule without a head is called a *constraint.* A rule with a basic atom as its head and empty tail is called a *fact* in which case the "$\leftarrow$" is dropped.

Let R be an ASP rule of the form (2.3) and let $pos(R) = \{A_1, \ldots, A_m\}$ represent the positive atoms, $neg(R) = \{B_1, \ldots, B_n\}$ the negation-as-failure atoms, and $agg(R) = \{C_1, \ldots, C_k\}$ represent the aggregate atoms in the body of a rule R. Let $lit(A)$ represent the set of basic literals in atom A, i.e. $lit(A) = \{A\}$ if A is a basic atom; $lit(A) = \{B_0, \ldots, B_n\}$ if A is an aggregate atom. Let C be an aggregate atom of the form (2.1) and let $pos(C) = \{B_0, \ldots, B_m\}$ be the sets of basic positive literals such that $lit(C) = pos(C)$.

Let $lit(R) = lit(head(R)) \cup pos(R) \cup neg(R) \cup \bigcup_{C \in agg(R)} lit(C)$ for a rule $R \in \Pi$ and $lit(\Pi) = \bigcup_{R \in \Pi} lit(R)$ be the set of basic literals in ASP program Π.

Definition 5 (Aggregate Atom Satisfaction) *A ground aggregate atom C of the form* (2.1) *is satisfied by a set of basic ground atoms S, if $L \leq \sum_{0 \leq i \leq m, B_i \in S} w_i \leq U$ and we write $S \models C$.*

Given a set of basic ground literals S and a basic ground atom A, we say $S \models A$ if $A \in S$, $S \models$ **not** A if $A \notin S$. For a rule R of the form (2.3) $S \models body(R)$ if $\forall A \in \{A_1, \ldots, A_m\}, S \models A$, $\forall B \in \{B_1, \ldots, B_n\}, S \models$ **not** B, and $\forall C \in \{C_1, \ldots, C_k\}, S \models C$; $S \models head(R)$ if $S \models A_0$.

Definition 6 (Rule Satisfaction) *A ground rule $R \in \Pi$ is satisfied by a set of basic ground atoms S, iff, $S \models body(R)$ implies $S \models head(R)$. A constraint rule $R \in \Pi$ is satisfied by set S if $S \not\models body(R)$.*

We define reduct of an ASP program by treating aggregate atoms in a similar way as negation-as-failure literals, since our code does not contain recursion through aggregation (which can yield non-intuitive answer-sets (Son and Pontelli, 2007)).

Definition 7 (Reduct) *Let S be a set of ground basic atoms, the reduct of ground ASP program Π w.r.t. S, written Π^S is the set of rules: $\{p \leftarrow A_1, \ldots, A_m. \mid A_0 \leftarrow A_1, \ldots,\ A_m,$ **not** $B_1, \ldots,$ **not** $B_n,\ C_1, \ldots,\ C_k. \in \Pi, p \in lit(A_0) \cap S, \{A_1, \ldots, A_m\} \subseteq S, \{B_1, \ldots, B_n\} \cap S = \emptyset,\ \nexists C \in \{C_1, \ldots, C_k\}, S \not\models C\}$.*

Intuitively, this definition of reduct removes all rules which contain a naf-literal or an aggregate atom in their bodies that does not hold in S, and it removes aggregate atoms as well as naf-literals from the body of the remaining rules.

Heads of choice-rules are split into multiple rules containing at most one atom in their heads. The resulting reduct is a program that does not contain any aggregate atoms or negative literals. The rules of such a program are monotonic, such that if it satisfied by a set S of atoms, it is also satisfied by any superset of S.

A **deductive closure** of such a (positive) monotonic program is defined as the unique smallest set of atoms S such that whenever all body atoms of a rule hold in S, the head also holds in S. The deductive closure can be iteratively computed by starting with an empty set and adding heads of rules for which the bodies are satisfied, until a fix point is reached, where no additional rules can be satisfied. (adopted from Baral (2003))

Definition 8 (Answer Set) *A set of basic ground atoms S is an answer set of a ground ASP program Π, iff S is equal to the deductive closure of Π^S and S satisfies each rule of Π. (adopted from Baral (2003))*

Clingo Specific Syntactic Elements

The ASP code in this thesis is in the syntax of ASP solver called clingo (Gebser *et al.*, 2011). The "←" in ASP rules is replaced by the symbol "`:-`". Though the semantics of ASP are defined on ground programs, Clingo allows variables and other constructs for compact representation. We intuitively describe specific syntactic elements and their meanings below:

Comments: Text following "%" to the end of the line is treated as a comment.

Interval: Atoms defined over an contiguous range of integer values can be compactly written as intervals, e.g. $p(1\ ..\ 5)$ represents atoms $p(1), p(2), p(3), p(4), p(5)$.

Pooling: Symbol ";" allows for pooling alternative terms. For example, an atom $p(\dots, X, \dots)$ and $p(\dots, Y, \dots)$ can be pooled together into a single atom as $p(\dots, X; Y, \dots)$.

Aggregate assignment atom: An aggregate assignment atom $Q = \#sum[A_0 = w_0, \dots, A_m = w_m, \textbf{not } A_{m+1} = w_{m+1}, \dots, \textbf{not } A_n = w_n]$ assigns the sum

$\sum_{A_i \in S, 0 \leq i \leq m} w_i + \sum_{A_j \notin S, m+1 \leq n} w_j$ to Q w.r.t. a consistent set of basic ground atoms S.

Condition: Conditions allow instantiating variables to collections of terms within aggregates, e.g. $\{p(X) : q(X)\}$ instantiates $p(X)$ for only those X that $q(X)$ satisfies. For example, if we have $p(1..5)$ but only $q(3;5)$, then $\{p(X) : q(X)\}$ is expanded to $\{p(3), p(5)\}$.

Grounding

Grounding makes a program variable free by replacing variables with the possible values they can take. Clingo uses the grounder Gringo (Gebser *et al.*, 2007) for "smart" grounding, which results in substantial reduction in the size of the program. Details of this grounding are implementation specific. We present the intuitive process of grounding below.

1. A set of ground terms is constructed, where a ground term is a term that contains no variables.

2. The variables are split into two categories: local and global. Local variables are the ones that appear only within an aggregate atom (minus the limits) and nowhere else in a rule. Such variables are considered local writ. the aggregate atom. All other variables are considered global.

3. First the global variables are eliminated in the rules as follows:

   - Each rule r containing an aggregate assignment atom of the form (2.1) is replaced with set of rules r' in which the aggregate assignment atom is is replaced with an aggregate atom with lower and upper bounds of Q for all possible substitutions of Q. This is generalized to multiple aggregate assignment atoms by repeating this step for each such atom, where output of previous iteration forms the input of the next iteration.

- Each rule r', is replaced with the set of all rules r'' obtained by all possible substitutions of ground terms for global variables in r.

4. Then the local variables are eliminated in the rules by expanding conditions, such that $p(\dots, X, \dots) : d(X)$ are replaced by $p(\dots, d_1, \dots), \dots, p(\dots, d_k, \dots)$ for the extent $\{d_1, \dots, d_k\}$ of $d(X)$. This is generalized to multiple conditions in the obvious way.

Following the convention of the Clingo system, Variables in rules presented in this thesis start with capital letters while lower-case text and numbers are constants. Italicized text represents a constant term from a definition in context.

A recent work Harrison *et al.* (2013) gives the semantics of Gringo with ASP Core 2 syntax (Calimeri *et al.*, 2013) using Infintary Propositional Formulas, which translate Gringo to propositional formulas with infinitely long conjunctions and disjunctions. Their approach removes the safety requirement, but the subset of Gringo presented does appear to cover assignments. Although their approach provides a way to improve our ASP encoding by removing the requirement of specifying the maximum number of tokens or running simulations until a condition holds, our simpler (limited) semantics is sufficient for the limited syntax and semantics we use.

2.2.2 Multiset

A **multiset** A over a domain set D is a pair $\langle D, m \rangle$, where $m : D \rightarrow \mathbb{N}$ is a function giving the multiplicity of $d \in D$ in A. Given two multsets $A = \langle D, m_A \rangle, B = \langle D, m_B \rangle$, $A \odot B$ if $\forall d \in D : m_A(d) \odot m_B(d)$, where $\odot \in \{<, >, \leq, \geq, =\}$, and $A \neq B$ if $\exists d \in D : m_A(d) \neq m_B(d)$. Multiset sum/difference is defined in the usual way. We use the short-hands $d \in A$ to represent $m_A(d) > 0$, $A = \emptyset$ to represent $\forall d \in D, m(d) = 0$, $A \otimes n$ to represent $\forall d \in D, m(d) \otimes n$, where $n \in \mathbb{N}$, $\otimes \in \{<, >, \leq, \geq, =, \neq\}$. We use the notation $d/n \in A$ to represent that d appears n-times in A; we drop A when clear from context. The reader is referred to Syropoulos (2001) for details.

2.2.3 Petri Net

A Petri Net is a graph of a finite set of nodes and directed arcs, where nodes are split between places and transitions, and each arc either connects a place to a transition or a transition to a place. Each place has a number of tokens (called the its marking) [1]. Collective marking of all places in a Petri Net is called its *marking* (or *state*). Arc labels represent arc weights. When missing, arc-weight is assumed as one, and place marking is assumed as zero.

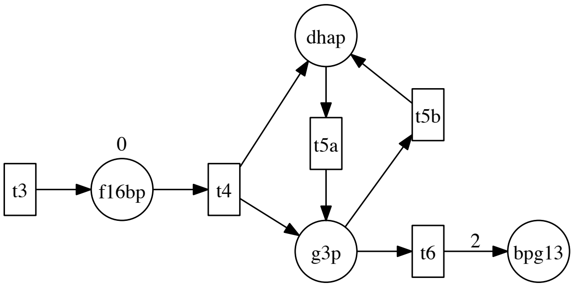


Figure 2.1: Petri Net graph (of sub-section of glycolysis pathway) showing places as circles, transitions as boxes and arcs as directed arrows. Places have token count (or marking) written above them, assumed 0 when missing. Arcs labels represent arc-weights, assumed 1 when missing.

The set of place nodes on incoming and outgoing arcs of a transition are called its pre-set (input place set or input-set) and post-set (output place set or output-set), respectively. A transition t is enabled when each of its pre-set place p has at least the number of tokens equal to the arc-weight from p to t. An enabled transition may fire,

[1] Standard convention is to use dots in place nodes to represent the marking of the place. We use numbers for compact representation.

consuming tokens equal to arc-weight from place p to transition t from each pre-set place p, producing tokens equal to arc-weight from transition t to place p to each post-set place p.

Multiple transitions may fire as long as they consume no more than the available tokens, with the assumption that tokens cannot be shared. Fig. 2.1 shows a representation of a portion of the glycolysis pathway as given in Reece *et al.* (2010). In this figure, places represent reactants and products, transitions represent reactions, and arc weights represent reactant quantity consumed or the product quantity produced by the reaction. When unspecified, arc-weight is assumed to be 1 and place-marking is assumed to be 0.

Definition 9 (Petri Net) *A Petri Net is a tuple $PN = (P, T, E, W)$, where, $P = \{p_1, \ldots, p_n\}$ is a finite set of places; $T = \{t_1, \ldots, t_m\}$ is a finite set of transitions, $P \cap T = \emptyset$; $E^+ \subseteq T \times P$ is a set of arcs from transitions to places; $E^- \subseteq P \times T$ is a set of arcs from places to transitions; $E = E^+ \cup E^-$; and $W : E \to \mathbb{N} \setminus \{0\}$ is the arc-weight function*

Definition 10 (Marking) *A marking $M = (M(p_1), \ldots, M(p_n))$ is the token assignment of each place node $p_i \in P$ of PN, where $M(p_i) \in \mathbb{N}$. Initial token assignment $M_0 : P \to \mathbb{N}$ is called the initial marking. Marking at step k is written as M_k.*

Definition 11 (Pre-set & post-set of a transition) *Pre-set / input-set of a transition $t \in T$ of PN is $\bullet t = \{p \in P : (p, t) \in E^-\}$, while the post-set / output-set is $t\bullet = \{p \in P : (t, p) \in E^+\}$*

Definition 12 (Enabled Transition) *A transition $t \in T$ of PN is enabled with respect to marking M, $enabled_M(t)$, if $\forall p \in \bullet t, W(p, t) \leq M(p)$. An enabled transition may fire.*

Definition 13 (Transition Execution) *A transition execution is the simulation of change of marking from M_k to M_{k+1} due to firing of a transition $t \in T$ of PN. M_{k+1} is computed as follows:*

$$\forall p_i \in \bullet t, M_{k+1}(p_i) = M_k(p_i) - W(p_i, t)$$

$$\forall p_j \in t\bullet, M_{k+1}(p_j) = M_k(p_j) + W(t, p_j)$$

Petri Nets allow simultaneous firing of a set of enabled transitions w.r.t. a marking as long as they do not conflict.

Definition 14 (Conflicting Transitions) *Given PN with marking M. A set of enabled transitions $T_e = \{t \in T : enabled_M(t)\}$ of PN conflict if their simultaneous firing will consume more tokens than are available at an input place:*

$$\exists p \in P : M(p) < \sum_{t \in T_e \wedge p \in \bullet t} W(p, t)$$

Definition 15 (Firing Set) *A firing set is a set $T_k = \{t_1, \ldots, t_m\} \subseteq T$ of simultaneously firing transitions that are enabled and do not conflict w.r.t. to the current marking M_k of PN.*

Definition 16 (Firing Set Execution) *Execution of a firing set T_k of PN on a marking M_k computes the new marking M_{k+1} as follows:*

$$\forall p \in P, M_{k+1}(p) = M_k(p) - \sum_{t \in T_k \wedge p \in \bullet t} W(p, t) + \sum_{t \in T_k \wedge p \in t\bullet} W(t, p)$$

where $\sum_{t \in T_k \wedge p \in \bullet t} W(p, t)$ is the total consumption from place p and $\sum_{t \in T_k \wedge p \in t\bullet} W(t, p)$ is the total production at place p.

Definition 17 (Execution Sequence) *An execution sequence $X = M_0, T_0, M_1, T_1, \ldots, M_k, T_k, M_{k+1}$ of PN is the simulation of a firing set sequence $\sigma = T_1, T_2, \ldots, T_k$ w.r.t.*

an initial marking M_0, producing the final marking M_{k+1}. M_{k+1} is the transitive closure of firing set executions, where subsequent marking become the initial marking for the next firing set.

For an execution sequence $X = M_0, T_0, M_1, T_1, \ldots, M_k, T_k, M_{k+1}$, the firing of T_0 with respect to marking M_0 produces the marking M_1 which becomes the initial marking for T_1, which produces M_2 and so on.

2.3 Translating Basic Petri Net Into ASP

In this section we present ASP encoding of simple Petri Nets. We describe, how a given Petri Net PN, and an initial marking M_0 are encoded into ASP for a simulation length k. Following sections will show how Petri Net extensions can be easily added to it. We represent a **Petri Net** with the following facts:

f1: Facts `place(`p_i`).` where $p_i \in P$ is a place.

f2: Facts `trans(`t_j`).` where $t_j \in T$ is a transition.

f3: Facts `ptarc(`$p_i, t_j, W(p_i, t_j)$`).` where $(p_i, t_j) \in E^-$ with weight $W(p_i, t_j)$.

f4: Facts `tparc(`$t_i, p_j, W(t_i, p_j)$`).` where $(t_i, p_j) \in E^+$ with weight $W(t_i, p_j)$.

Petri Net **execution simulation** proceeds in discrete time-steps, these time steps are encoded by the following facts:

f5: Facts `time(`ts_i`)` where $0 \leq ts_i \leq k$.

The **initial marking** (or initial state) of the Petri Net is represented by the following facts:

i1: Facts `holds(`$p_i, M_0(p_i), 0$`)` for every place $p_i \in P$ with initial marking $M_0(p_i)$.

ASP requires all variables in rule bodies be domain restricted. So, we add the following facts to capture the possible token quantities produced during the simulation [2]:

f6: Facts `num(`n`).`, where $0 \leq n \leq ntok$

A transition t_i is enabled if each of its input places $p_j \in \bullet t_i$ has at least arc-weight $W(p_j, t_i)$ tokens. Conversely, t_i is not enabled if $\exists p_j \in \bullet t_i : M(p_j) < W(p_j, t_i)$, and is only enabled when no such place p_j exists. These are captured in $e1$ and $e2$ respectively:

e1: `notenabled(T,TS):-ptarc(P,T,N),holds(P,Q,TS),Q<N, place(P),`

```
  trans(T), time(TS),num(N),num(Q).
```

e2: `enabled(T,TS) :- trans(T), time(TS), not notenabled(T, TS).`

Rule $e1$ encodes `notenabled(T,TS)` which captures the existence of an input place P of transition T that violates the minimum token requirement N at time-step TS. Where, the predicate `holds(P,Q,TS)` encodes the marking Q of place P at TS.

Rule $e2$ encodes `enabled(T,TS)` which captures that transition T is enabled at TS since there is no input place P of transition T that violates the minimum input token requirement at TS. In biological context, $e2$ captures the conditions when a reaction (represented by T) is ready to proceed. A subset of enabled transitions may fire simultaneously at a given time-step. This is encoded as:

a1: `{fires(T,TS)} :- enabled(T,TS), trans(T), time(TS).`

Rule $a1$ encodes `fires(T,TS)`, which captures the firing of transition T at TS. The rule is encoded with a count atom as its head, which makes it a choice rule. This rule

[2] Note that $ntok$ can be arbitrarily chosen to be larger than the maximum expected token quantity produced during the simulation.

either picks the enabled transition T for firing at TS or not, effectively enumerating a subset of enabled transitions to fire. Whether this set can fire or not in an answer set is subject to conflict checking, which is done by rules $a2, a3, a4$ shown later.

In biological context, the selected transition-set models simultaneously occurring reactions and the conflict models limited reactant supply that cannot be shared. Such a conflict can lead to multiple choices in parallel reaction evolutions and different outcomes.

The next set of rules captures the consumption and production of tokens due to the firing of individual transitions in a firing-set as well as their aggregate effect, which computes the marking for the next time step:

```
r1: add(P,Q,T,TS) :- fires(T,TS), tparc(T,P,Q), time(TS).

r2: del(P,Q,T,TS) :- fires(T,TS), ptarc(P,T,Q), time(TS).

r3: tot_incr(P,QQ,TS) :- QQ=#sum[add(P,Q,T,TS)=Q:num(Q):trans(T)], time(TS),
      num(QQ), place(P).

r4: tot_decr(P,QQ,TS) :- QQ=#sum[del(P,Q,T,TS)=Q:num(Q):trans(T)], time(TS),
      num(QQ), place(P).

r5: holds(P,Q,TS+1) :-holds(P,Q1,TS),tot_incr(P,Q2,TS),time(TS+1),
      tot_decr(P,Q3,TS),Q=Q1+Q2-Q3,place(P),num(Q;Q1;Q2;Q3),time(TS).
```

Rule $r1$ encodes `add(P,Q,T,TS)` and captures the addition of Q tokens to place P due to firing of transition T at time-step TS. Rule $r2$ encodes `del(P,Q,T,TS)` and captures the deletion of Q tokens from place P due to firing of transition T at TS.

Rules $r3$ and $r4$ aggregate all `add`'s and `del`'s for place P due to $r1$ and $r2$ at time-step TS, respectively, by using the `QQ=#sum[]` construct to sum the Q values into QQ. Rule $r5$ which encodes `holds(P,Q,TS+1)` uses these aggregate adds and removes and updates P's marking for the next time-step $TS + 1$. In biological context, these rules capture the effect of a reaction on reactant and product quantities available in the next simulation step. To prevent overconsumption at a place following rules are added:

a2: `consumesmore(P,TS) :- holds(P,Q,TS), tot_decr(P,Q1,TS), Q1 > Q.`

a3: `consumesmore :- consumesmore(P,TS).`

a4: `:- consumesmore.`

Rule $a2$ encodes `consumesmore(P,TS)` which captures overconsumption of tokens at input place P at time TS due to the firing set selected by $a1$. Overconsumption (and hence conflict) occurs when tokens $Q1$ consumed by the firing set are greater than the tokens Q available at P. Rule $a3$ generalizes this notion of overconsumption and constraint $a4$ eliminates answers where overconsumption is possible.

Definition 18 *Given a Petri Net PN, its initial marking M_0 and its encoding $\Pi(PN$, M_0, k, $ntok)$ for k-steps and maximum ntok tokens at any place. We say that there is a 1-1 correspondence between the answer sets of $\Pi(PN, M_0, k, ntok)$ and the execution sequences of PN iff for each answer set A of $\Pi(PN, M_0, k, ntok)$, there is a corresponding execution sequence $X = M_0, T_0, M_1, \ldots, M_k, T_k, M_{k+1}$ of PN and for each execution sequence X of PN there is an answer-set A of $\Pi(PN, M_0, k, ntok)$ such that*

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\}$$

$$\{holds(p, q, ts) : p \in P, q = M_{ts}(p), 0 \leq ts \leq k+1\}$$
$$= \{holds(p, q, ts) : holds(p, q, ts) \in A\}$$

Proposition 1 *There is a 1-1 correspondence between the answer sets of $\Pi^0(PN$, M_0, k, $ntok)$ and the execution sequences of PN.*

2.3.1 An example execution

Next we look at an example execution of the Petri Net shown in Figure 2.1. The Petri Net and its initial marking are encoded as follows[3]:

```
num(0..60).time(0..5).place(f16bp;dhap;g3p;bpg13).

trans(t3;t4;t5a;t5b;t6).tparc(t3,f16bp,1).ptarc(f16bp,t4,1).

tparc(t4,dhap,1).tparc(t4,g3p,1).ptarc(dhap,t5a,1).

tparc(t5a,g3p,1).ptarc(g3p,t5b,1).tparc(t5b,dhap,1).

ptarc(g3p,t6,1).tparc(t6,bpg13,2).holds(f16bp;dhap;g3p;bgp13,0,0).
```

we get thousands of answer-sets, for example[4]:

```
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)

holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,1,1) holds(g3p,0,1)

holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,1,2) holds(g3p,1,2)

holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,1,3) holds(g3p,2,3)

holds(bpg13,2,4) holds(dhap,3,4) holds(f16bp,1,4) holds(g3p,2,4)

holds(bpg13,4,5) holds(dhap,4,5) holds(f16bp,1,5) holds(g3p,2,5)

fires(t3,0) fires(t3;t4,1) fires(t3;t4;t5a;t5b,2)

fires(t3;t4;t5a;t5b;t6,3) fires(t3;t4;t5a;t5b;t6,4)

fires(t3;t4;t5a;t5b;t6,5)
```

2.4 Changing Firing Semantics

The ASP code above implements the *set firing* semantics. It can produce a large number of answer-sets, since any subset of a firing set will also be fired as a firing set.

[3] {`holds(p1,0,0)`,...,`holds(pN,0,0)`}, {`num(0)`,...,`num(60)`}, {`time(0)`,...,`time(5)`} have been written as `holds(p1;...;pN,0,0)`, `num(0..60)`, `time(0..5)`, respectively, to save space.

[4] {`fires(t1,ts1)`,...,`fires(tN,ts1)`} have been written as `fires(t1;...;tN;ts1)` to save space.

For our biological system modeling, it is often beneficial to simulate only the maximum activity at any given time-step. We accomplish this by defining the *maximal firing set* semantics, which requires that a maximal subset of non-conflicting transitions fires at a single time step[5]. Our semantics is different from the firing multiplier approach used by Krepska *et al.* (2008), in which a transition can fire as many times as allowed by the tokens available in its source places. Their approach requires an exponential time firing algorithm in the number of transitions. Our maximal firing set semantics is implemented by adding the following rules to the encoding in Section 2.3:

a5: `could_not_have(T,TS) :- enabled(T,TS), not fires(T,TS), ptarc(S,T,Q), holds(S,QQ,TS), tot_decr(S,QQQ,TS), Q > QQ - QQQ.`

a6: `:- not could_not_have(T,TS), enabled(T,TS), not fires(T,TS), trans(T), time(TS).`

Rule $a5$ encodes `could_not_have(T,TS)` which means that an enabled transition T that did not fire at time TS, could not have fired because its firing would have resulted in overconsumption. Rule $a6$ eliminates any answer-sets in which an enabled transition did not fire, that could not have caused overconsumption. Intuitively, these two rules guarantee that the only reason for an enabled transition to not fire is conflict avoidance (due to overconsumption). With this firing semantics, the number of answer-sets produced for Petri Net in Figure 2.1 reduces to 2.

Proposition 2 *There is 1-1 correspondence between the answer sets of* $\Pi^1(PN, M_0, k, ntok)$ *and the execution sequences of* PN.

Other firing semantics can be encoded with similar ease. For example, if *interleaved* firing semantics is desired, replace rules $a5, a6$ with the following:

[5] Such a semantics reduces the reachable markings. See Burkhard (1980) for the analysis of its computational power.

a5': `more_than_one_fires :- fires(T1,TS), fires(T2, TS), T1 != T2, time(TS).`

a6': `:- more_than_one_fires.`

We now look at Petri Net extensions and show how they can be easily encoded in ASP.

2.5 Extension - Reset Arcs

Definition 19 (Reset Arc) *A Reset Arc in a Petri Net PN^R is an arc from place p to transition t that consumes all tokens from its input place p upon firing of t. A Reset Petri Net is a tuple $PN^R = (P, T, E, W, R)$ where, P, T, E, W are the same as for PN; and $R : T \rightarrow 2^P$ defines reset arcs*

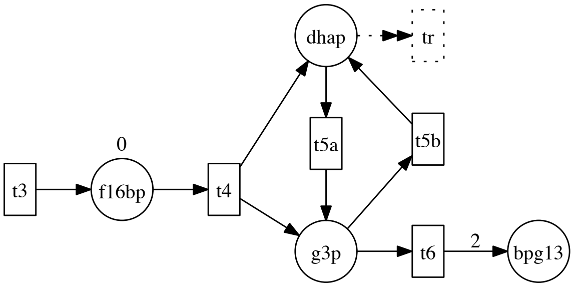


Figure 2.2: Petri Net of Fig 2.1 extended with a reset arc from *dhap* to *tr* shown with double arrowhead.

Figure 2.2 shows an extended version of the Petri Net in Figure 2.1 with a reset arc from *dhap* to *tr* (shown with double arrowhead). In biological context it models the removal of all quantity of compound *dhap*. Petri Net execution semantics with reset arcs is modified for conflict detection and execution as follows:

Definition 20 (Reset Transition) *A transition $t \in T$ of PN^R is called a reset-transition if it has a reset arc incident on it, i.e. $R(t) \neq \emptyset$.*

Definition 21 (Firing Set) *A firing set is a set $T_k = \{t_1, \ldots, t_m\} \subseteq T$ of simultaneously firing transitions that are enabled and do not conflict w.r.t. to the current marking M_k of PN^R. T_k is not a firing set if there is an enabled reset-transition that is not in T_k, i.e. $\exists t : enabled_{M_k}(t), R(t) \neq \emptyset, t \not\in T_k$.* [6].

Definition 22 (Transition Execution in PN^R) *A transition execution is the simulation of change of marking from M_k to M_{k+1} due to firing of a transition $t \in T$ of PN^R. M_{k+1} is computed as follows:*

$$\forall p_i \in \bullet t, M_{k+1}(p_i) = M_k(p_i) - W(p_i, t)$$

$$\forall p_j \in t\bullet, M_{k+1}(p_j) = M_k(p_j) + W(t, p_j)$$

$$\forall p_r \in R(t), M_{k+1}(p_r) = M_k(p_r) - M_k(p_r)$$

Definition 23 (Conflicting Transitions in PN^R) *A set of enabled transitions conflict in PN^R w.r.t. M_k if firing them simultaneously will consume more tokens than are available at any one of their common input-places. $T_e = \{t \in T : enabled_{M_k}(t)\}$ conflict if:*

$$\exists p \in P : M_k(p) < (\sum_{t \in T_e \wedge (p,t) \in E^-} W(p,t) + \sum_{t \in T_e \wedge p \in R(t)} M_k(p))$$

Definition 24 (Firing Set Execution in PN^R) *Execution of a transition set T_i in PN^R has the following effect:*

$$\forall p \in P \setminus R(T_i), M_{k+1}(p) = M_k(p) - \sum_{t \in T_i \wedge p \in \bullet t} W(p,t) + \sum_{t \in T_i \wedge p \in t\bullet} W(t,p)$$

[6] The reset arc is involved here because we use a modified execution semantics of reset arcs compared to the standard definition (Araki and Kasami, 1976). Even though both capture similar operation, our definition allows us to model elimination of all quantity of a substance as soon as it is produced, even in a maximal firing set semantics. Our semantics considers reset arc's token consumption in contention with other arcs, while the standard definition does not.

$$\forall p \in R(T_i), M_{k+1}(p) = \sum_{t \in T_i \wedge p \in t\bullet} W(t,p)$$

where $R(T_i) = \bigcup_{t \in T_i} R(t)$ *and represents the places emptied by* T_i *due to reset arcs* [7]*.*

Since a reset arc from p to t, $p \in R(t)$ consumes current marking dependent tokens, we extend `ptarc` to include time-step and replace $f3, f4, e1, r1, r2, a5$ with $f7, f8, e3, r6, r7, a7$, respectively in the Section 2.4 encoding and add rule $f9$ for each reset arc:

f7: Rules `ptarc(`$p_i, t_j, W(p_i, t_j), ts_k$`):-time(`$ts_k$`).` for each non-reset arc $(p_i, t_j) \in E^-$

f8: Rules `tparc(`$t_i, p_j, W(t_i, p_j), ts_k$`):-time(`$ts_k$`).` for each non-reset arc $(t_i, p_j) \in E^+$

e3: `notenabled(T,TS) :- ptarc(P,T,N,TS), holds(P,Q,TS), Q < N,`

`place(P), trans(T), time(TS), num(N), num(Q).`

r6: `add(P,Q,T,TS) :- fires(T,TS), tparc(T,P,Q,TS), time(TS).`

r7: `del(P,Q,T,TS) :- fires(T,TS), ptarc(P,T,Q,TS), time(TS).`

f9: Rules `ptarc(`p_i, t_j, X, ts_k`) :- holds(`p_i, X, ts_k`), num(`X`),` $X > 0$. for each reset arc between p_i and t_j using $X = M_k(p_i)$ as arc-weight at time step ts_k.

f10: Rules `:- enabled(`t_j, ts_k`),not fires(`t_j, ts_k`), time(`ts_k`).` for each transition t_j with an incoming reset arc, i.e. $R(t_j) \neq \emptyset$.

a7: `could_not_have(T,TS) :- enabled(T,TS), not fires(T,TS),`

`ptarc(S,T,Q,TS), holds(S,QQ,TS), tot_decr(S,QQQ,TS), Q>QQ-QQQ.`

Rule $f9$ encodes place-transition arc with marking dependent weight to capture the notion of a reset arc, while rule $f10$ ensures that the reset-transition (i.e. the transition on which the reset arc terminates) always fires when enabled.

[7] Our definition of conflicting transitions allows at most one transition with a reset arc from a place to fire, any more create a conflict. Thus, the new marking computation is equivalent to $\forall p \in P, M_{k+1}(p) = M_k(p) - (\sum_{t \in T_k \wedge p \in \bullet t} W(p,t) + \sum_{t \in T_k \wedge p \in R(t)} M_k(p)) + \sum_{t \in T_k \wedge p \in t\bullet} W(t,p)$

Proposition 3 *There is 1-1 correspondence between the answer sets of* $\Pi^2(PN^R, M_0, k, ntok)$ *and the execution sequences of* PN^R*.*

The execution semantics of our definition are slightly different from the standard definition in Araki and Kasami (1976), even though both capture similar operations. Our implementation considers token consumption by reset arc in contention with other token consuming arcs from the same place, while the standard definition considers token consumption as a side effect, not in contention with other arcs.

We chose our definition to allow modeling of biological process that removes all available quantity of a substance in a maximal firing set. Consider Figure 2.2, if *dhap* has 1 or more tokens, our semantics would only permit either $t5a$ or tr to fire in a single time-step, while the standard semantics can allow both $t5a$ and tr to fire simultaneously, such that the reset arc removes left over tokens after $(dhap, t5a)$ consumes one token.

We could have, instead, extended our encoding to include self-modifying nets (Valk, 1978), but our modified-definition provides a simpler solution. Standard semantics, however, can be easily encoded by replacing $r5$ by $r5a', r5b'$; replacing $f9, f10$ with $f9'$; and adding $a8$ as follows:

f9': `rptarc(`p_i`,`t_j`).` - for each reset arc between $p_i \in R(t_j)$ and t_j.

a8: `reset(P,TS) :- rptarc(P,T), place(P), trans(T), fires(T,TS), time(TS).`

r5a':
```
holds(P,Q,TS+1) :- holds(P,Q1,TS), tot_incr(P,Q2,TS),
    tot_decr(P,Q3,TS), Q=Q1+Q2-Q3, place(P), num(Q;Q1;Q2;Q3), time(TS),
    time(TS+1), not reset(P,TS).
```

r5b':
```
holds(P,Q,TS+1) :- tot_incr(P,Q,TS), place(P), num(Q), time(TS),
    time(TS+1), reset(P,TS).
```

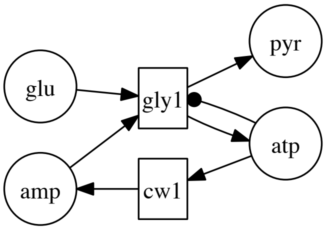


Figure 2.3: Petri Net showing feedback inhibition arc from *atp* to *gly*1 with a bullet arrowhead. Inhibitor arc weight is assumed 1 when not specified.

where, the fact $f9'$ encodes the reset arc; rule $a8$ encodes if place P will be reset at time TS due to firing of transition T that has a reset arc on it from P to T; rule $r5a'$ computes marking at $TS+1$ when place P is not being reset; and rule $r5b'$ computes marking at $TS+1$ when P is being reset.

2.6 Extension - Inhibitor Arcs

Definition 25 (Inhibitor Arc) *An inhibitor arc (Peterson, 1977) is a place-transition arc that inhibits its transition from firing as long as the place has any tokens in it. An inhibitor arc does not consume any tokens from its input place. A Petri Net with reset and inhibitor arcs is a tuple* $PN^{RI} = (P, T, E, W, R, I)$*, where,* P, T, E, W, R *are the same as for* PN^R*; and* $I : T \to 2^P$ *defines inhibitor arcs.*

Figure 2.3 shows a Petri Net with inhibition arc from *atp* to *gly*1 with a bulleted arrowhead. It models biological feedback regulation in simplistic terms, where excess *atp* downstream causes the upstream *atp* production by glycolysis *gly* to be inhibited until the excess quantity is consumed (Reece *et al.*, 2010). Petri Net execution semantics with inhibit arcs is modified for determining enabled transitions as follows:

Definition 26 (Enabled Transition in PN^{RI}) *A transition t is enabled with respect to marking M, $enabled_M(t)$, if all its input places p have at least the number* *of tokens as the arc-weight* $W(p,t)$ *and all* $p \in I(t)$ *have zero tokens, i.e.* $(\forall p \in \bullet t, W(p,t) \leq M(p)) \wedge (\forall p \in I(t), M(p) = 0)$

We add inhibitor arcs to our encoding in Section 2.5 as follows:

f11: Rules `iptarc(`$p_i, t_j, 1, ts_k$`):-time(`ts_k`).` for each inhibitor arc between $p_i \in I(t_j)$ and t_j.

e4: `notenabled(T,TS) :- iptarc(P,T,N,TS), holds(P,Q,TS), place(P),`
`trans(T), time(TS), num(N), num(Q), Q >= N.`

The new rule $e4$ encodes another reason for a transition to be disabled (or not enabled). An inhibitor arc from p to t with arc weight N will cause its target transition t to not enable when the number of tokens at its source place p is greater than or equal to N, where N is always 1 per rule $f11$.

Proposition 4 *There is 1-1 correspondence between the answer sets of* $\Pi^3(PN^{RI}, M_0, k, ntok)$ *and the execution sequences of* PN.

2.7 Extension - Read Arcs

Definition 27 (Read Arc) *A read arc (a test arc or a query arc) (Christensen and Hansen, 1993) is an arc from place to transition, which enables its transition only when its source place has at least the number of tokens as its arc weight. It does not consume any tokens from its input place. A Petri Net with reset, inhibitor and read arcs is a tuple* $PN^{RIQ} = (P, T, W, R, I, Q, QW)$*, where,* P, T, E, W, R, I *are the same as for* PN^{RI}*;* $Q \subseteq P \times T$ *defines read arcs; and* $QW : Q \rightarrow \mathbb{N} \setminus \{0\}$ *defines read arc weight.*

Figure 2.4 shows a Petri Net with read arc from *h_is* to *syn* shown with arrowhead on both ends. It models the ATP synthase *syn* activation requiring a higher

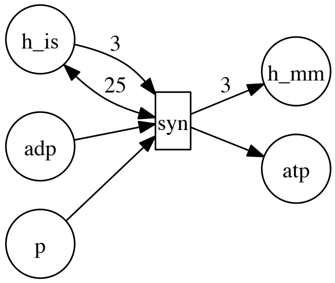


Figure 2.4: Petri Net with read arc from *h_is* to *syn* shown with arrowhead on both ends. The transition *syn* will not fire unless there are at least 25 tokens in *h_is*, but when it executes, it only consumes 3 tokens.

concentration of $H+$ ions *h_is* in the intermembrane space [8]. The reaction itself consumes a lower quantity of $H+$ ions represented by the regular place-transition arc (Reece *et al.*, 2010; Berg *et al.*, 2002). Petri Net execution semantics with read arcs is modified for determining enabled transitions as follows:

Definition 28 (Enabled Transition in PN^{RIQ}) *A transition t is enabled with respect to marking M, $enabled_M(t)$, if all its input places p have at least the number of tokens as the arc-weight $W(p,t)$, all $p_i \in I(t)$ have zero tokens and all $p_q : (p_q, t) \in Q$ have at least the number of tokens as the arc-weight $W(p,t)$, i.e.* $(\forall p \in \bullet t, W(p,t) \leq M(p)) \wedge (\forall p \in I(t), M(p) = 0) \wedge (\forall (p,t) \in Q, M(p) \geq QW(p,t))$

We add read arcs to our encoding of Section 2.6 as follows:

f12: Rules `tptarc(`$p_i, t_j, QW(p_i, t_j), ts_k$`):-time(`$ts_k$`).` for each read arc $(p_i, t_j) \in Q$.

e5:
```
notenabled(T,TS):-tptarc(P,T,N,TS),holds(P,Q,TS),
   place(P),trans(T), time(TS), num(N), num(Q), Q < N.
```

The new rule $f12$ captures the read arc and its arc-weight; and the new rule $e5$ encodes another reason for a transition to not be enabled. A read arc from p to t

[8] This is an oversimplified model of *syn* (ATP synthase) activation, since the actual model requires an $H+$ concentration differential across membrane.

with arc weight N will cause its target transition t to not enable when the number of tokens at its source place p is less than the arc weight N.

Proposition 5 *There is a 1-1 correspondence between the answer sets of* $\Pi^4(PN^{RIQ}, M_0, k, ntok)$ *and the execution sequences of* PN^{RIQ}.

2.8 Extension - Colored Tokens

Higher level Petri Nets extend the notion of tokens to typed (or colored) tokens. This allows a more compact representation of complicated networks (Peterson *et al.*, 1980).

Definition 29 (Petri Net with Colored Tokens) *A Petri Net with Colored Tokens (with reset, inhibit and read arcs) is a tuple* $PN^C = (P, T, E, C, W, R, I, Q, QW)$, *where* P, T, E, R, I, Q *are the same as for basic Petri Nets,* $C = \{c_1, \dots, c_l\}$ *is a finite set of colors (or types), and arc weights* $W : E \rightarrow \langle C, m \rangle$, $QW : Q \rightarrow \langle C, m \rangle$ *are specified as multi-sets of colored tokens over color set* C. *The state (or marking) of place nodes* $M(p_i) = \langle C, m \rangle, p_i \in P$ *is specified as a multiset of colored tokens over set* C.

We will now update some definitions related to Petri Nets to include colored tokens.

Definition 30 (Marking) *A marking* $M = (M(p_1), \dots, M(p_n))$ *assigns a colored multi-set of tokens over the domain of colors* C *to each place* $\{p_1, \dots, p_n\} \in P$ *of* PN^C. *The initial marking is the initial token assignment of place nodes and is represented by* M_0. *The marking at time-step* k *is written as* M_k.

Definition 31 (Pre-set and post-set of a transition) *The pre-set (or input-set) of a transition* t *is* $\bullet t = \{p \in P | (p, t) \in E^-\}$, *while the post-set (or output-set) is* $t\bullet = \{p \in P | (t, p) \in E^+\}$.

Definition 32 (Enabled Transition) *A transition t is enabled with respect to marking M, $enabled_M(t)$, if each of its input places p has at least the number of colored-tokens as the arc-weight $W(p,t)$*[9]*, each of its inhibiting places $p_i \in I(t)$ have zero tokens and each of its read places $p_q : (p_q, t) \in Q$ have at least the number of colored-tokens as the read-arc-weight $QW(p_q, t)$, i.e. $(\forall p \in \bullet t, W(p,t) \leq M(p)) \wedge (\forall p \in I(t), M(p) = \emptyset) \wedge (\forall (p,t) \in Q, M(p) \geq QW(p,t))$ for a given t.*[10]

Definition 33 (Transition Execution) *Execution of a transition t of PN^C on a marking M_k computes a new marking M_{k+1} as:*

$$\forall p \in \bullet t M_{k+1}(p) = M_k(p) - W(p,t)$$

$$\forall p \in t \bullet M_{k+1}(p) = M_k(p) + W(t,p)$$

$$\forall p \in R(t) M_{k+1}(p) = M_k(p) - M_k(p)$$

Any number of enabled transitions may fire simultaneously as long as they don't conflict. A transition when fired consumed tokens from its pre-set places equivalent to the (place,transition) arc-weight.

Definition 34 (Conflicting Transitions) *A set of transitions $T_c \subseteq \{t : enabled_{M_k}(t)\}$ is in conflict in PN^C with respect to M_k if firing them will consume more tokens than are available at one of their common input places, i.e., $\exists p \in P : M_k(p) < (\sum_{t \in T_c \wedge p \in \bullet t} W(p,t) + \sum_{t \in T_c \wedge p \in R(t)} M_k(p))$*

Definition 35 (Firing Set) *A firing set is a set $T_k = \{t_{k_1}, \ldots, t_{k_n}\} \subseteq T$ of simultaneously firing transitions of PN^C that are enabled and do not conflict w.r.t. to*

[9] In the following text, for simplicity, we will use $W(p,t)$ to mean $W(\langle p,t \rangle)$. We use similar simpler notation for QW.

[10] This is equivalent to $\forall c \in C, (\forall p \in \bullet t, m_{W(p,t)}(c) \leq m_{M(p)}(c)) \wedge (\forall p \in I(t), m_{M(p)}(c) = 0) \wedge (\forall (p,t) \in Q, m_{M(p)}(c) \geq m_{QW(p,t)}(c))$.

the current marking M_k of PN. A set T_k is not a firing set if there is an enabled reset-transition that is not in T_k, i.e. $\exists t \in enabled_{M_k}, R(t) \neq \emptyset, t \notin T_k$. [11]

Definition 36 (Firing Set Execution) *Execution of a firing set T_k of PN^C on a marking M_k computes a new marking M_{k+1} as:*

$$\forall p \in P \setminus R(T_k), M_{k+1}(p) = M_k(p) - \sum_{t \in T_k \wedge p \in \bullet t} W(p,t) + \sum_{t \in T_k \wedge p \in t \bullet} W(t,p)$$

$$\forall p \in R(T_k), M_{k+1}(p) = \sum_{t \in T_k \wedge p \in t \bullet} W(t,p)$$

where $R(T_k) = \bigcup_{t \in T_k} R(t)$ [12]

Definition 37 (Execution Sequence) *An execution sequence $X = M_0, T_0, M_1, T_1, \ldots, M_k, T_k, M_{k+1}$ of PN is the simulation of a firing set sequence $\sigma = T_1, T_2, \ldots, T_k$ w.r.t. an initial marking M_0, producing the final marking M_{k+1}. M_{k+1} is the transitive closure of firing set executions, where subsequent marking become the initial marking for the next firing set.*

For an execution sequence $X = M_0, T_0, M_1, T_1, \ldots, M_k, T_k, M_{k+1}$, the firing of T_0 with respect to marking M_0 produces the marking M_1 which becomes the initial marking for T_1, which produces M_2 and so on.

If the Figure 2.5 Petri Net has the marking: $M_0(mm) = [nadh/2, h/6]$, $M_0(q) = [e/2]$, $M_0(cytc) = [e/2]$, $M_0(is) = [o2/1]$, then transitions $t1, t3, t4$ are enabled. However, either $\{t1, t3\}$ or $\{t4\}$ can fire simultaneously in a single firing at time 0 due to limited h tokens in mm. $t4$ is said to be in conflict with $t1, t3$.

[11] See footnote 6

[12] See footnote 7

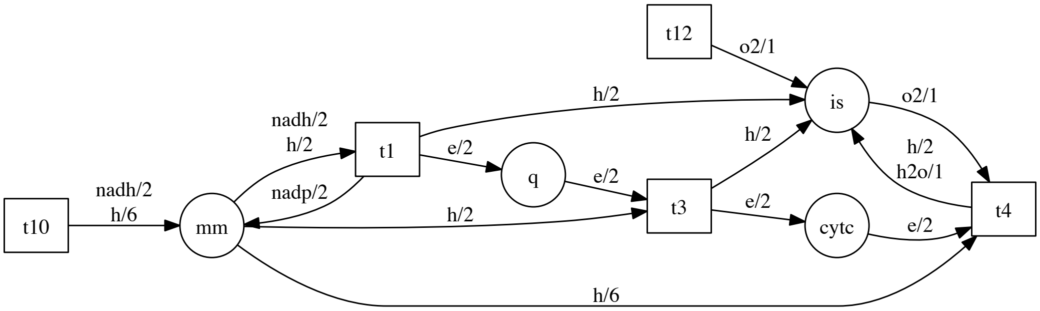


Figure 2.5: Petri Net with tokens of colors $\{e, h, h2o, nadh, nadp, o2\}$. Circles represent places, and rectangles represent transitions. Arc weights such as "$nadh/2, h/2$", "$h/2, h2o/1$" specify the number of tokens consumed and produced during the execution of their respective transitions, where "$nadh/2, h/2$" means 2 tokens of color $nadh$ and 2 tokens of h. Similar notation is used to specify marking on places, when not present, the place is assumed to be empty of tokens.

2.9 Translating Petri Nets with Colored Tokens to ASP

In order to represent the Petri Net PN^C with colored tokens, initial marking M_0, and simulation length k, we modify our encoding in Section 2.7 to add a new color parameter to all rules and facts containing token counts in them. We keep rules $f1, f2, f5, f6, f10$ remain as they were for basic Petri Nets. We add a new rule $f13$ for possible set of token colors and replace rules $f7, f8, f9, f11, f12, i1$ with $f14, f15, f16, f17, f18, i2$ to add the color parameter as follows:

f13: Facts `col(`c_k`)` where $c_k \in C$ is a color.

f14: Rules `ptarc(`p_i, t_j, n_c, c, ts_k`) :- time(`ts_k`).` for each $(p_i, t_j) \in E^-$, $c \in C$, $n_c = m_{W(p_i,t_j)}(c) : n_c > 0$.[13]

f15: Rules `tparc(`t_i, p_j, n_c, c, ts_k`) :- time(`ts_k`).` for each $(t_i, p_j) \in E^+$, $c \in C$, $n_c = m_{W(t_i,p_j)}(c) : n_c > 0$.

[13] The time parameter ts_k allows us to capture reset arcs, which consume tokens equal to the current (time-step based) marking of their source nodes.

f16: Rules ptarc(p_i, t_j, n_c, c, ts_k) :- holds(p_i, n_c, c, ts_k), num(n_c), n_c>0, time(ts_k). for each $(p_i, t_j) : p_i \in R(t_j)$, $c \in C$, $n_c = m_{M_k(p_i)}(c)$.

f17: Rules iptarc($p_i, t_j, 1, c, ts_k$) :- time(ts_k). for each $(p_i, t_j) : p_i \in I(t_j)$, $c \in C$.

f18: Rules tptarc(p_i, t_j, n_c, c, ts_k) :- time(ts_k). for each $(p_i, t_j) \in Q$, $c \in C$, $n_c = m_{QW(p_i,t_j)}(c) : n_c > 0$.

i2: Facts holds($p_i, n_c, c, 0$). for each place $p_i \in P, c \in C, n_c = m_{M0(p_i)}(c)$.

Next, we encode Petri Net's execution behavior, which proceeds in discrete time steps. Rules $e3, e4, e5, e2$ are replaced by $e6, e7, e8, e9$. For a transition t_i to be enabled, it must satisfy the following conditions: (i) $\nexists p_j \in \bullet t_i : M(p_j) < W(p_j, t_i)$, (ii) $\nexists p_j \in I(t_i) : M(p_j) > 0$, and (iii) $\nexists (p_j, t_i) \in Q : M(p_j) < QW(p_j, t_i)$. These three conditions are encoded as $e6, e7, e8$, respectively and we encode the absence of any of these conditions for a transition as $e9$:

e6: `notenabled(T,TS) :- ptarc(P,T, N,C,TS), holds(P,Q,C,TS), place(P), trans(T), time(TS), num(N), num(Q), col(C), Q<N.`

e7: `notenabled(T,TS) :- iptarc(P,T,N,C,TS), holds(P,Q,C,TS), place(P), trans(T), time(TS), num(N), num(Q), col(C), Q>=N.`

e8: `notenabled(T,TS) :- tptarc(P,T,N,C,TS), holds(P,Q,C,TS), place(P), trans(T), time(TS), num(N), num(Q), col(C), Q<N.`

e9: `enabled(T,TS) :- trans(T), time(TS), not notenabled(T,TS).`

Rule $e6$ captures the existence of an input place P with insufficient number of tokens for transition T to fire. Rule $e7$ captures existence of a non-empty source place P of an inhibitor arc to T preventing T from firing. Rule $e8$ captures existence of a source place P with less than arc-weight tokens required by the read arc to transition T for T to be enabled. The, `holds(P,Q,C,TS)` predicate captures the marking of place P at time TS as Q tokens of color C. Rule $e9$ captures enabling of transition T when no reason for it to be not enabled is determined by $e6, e7, e8$. In a biological context, this enabling is equivalent to a reaction's pre-conditions being satisfied. A reaction can proceed when its input substances are available in the required quantities, it is not inhibited, and any required activation quantity of activating substances is available.

Any subset of enabled transitions can fire simultaneously at a given time-step. We select a subset of fireable transitions using the choice rule $a1$ The choice rule $a1$ either picks an enabled transition T for firing at time TS or not. The combined effect over all transitions is to pick a subset of enabled transitions to fire. Rule $f10$ ensures that enabled reset-transitions will be a part of this firing set. Whether these transitions are in conflict are checked by later rules $a9, a3, a4$. In a biological context, the multiple firing models parallel processes occurring simultaneously. The marking is updated according to the firing set using rules $r8, r9, r10, r11, r12$ which replaced $r6, r7, r3, r4, r5$ as follows:

```
r8: add(P,Q,T,C,TS) :- fires(T,TS), tparc(T,P,Q,C,TS), time(TS).

r9: del(P,Q,T,C,TS) :- fires(T,TS), ptarc(P,T,Q,C,TS), time(TS).

r10: tot_incr(P,QQ,C,TS) :- col(C), QQ = #sum[add(P,Q,T,C,TS) = Q : num(Q)
      : trans(T)], time(TS), num(QQ), place(P).

r11: tot_decr(P,QQ,C,TS) :- col(C), QQ = #sum[del(P,Q,T,C,TS) = Q : num(Q)
      : trans(T)], time(TS), num(QQ), place(P).
```

r12:
```
holds(P,Q,C,TS+1):-place(P),num(Q;Q1;Q2;Q3),time(TS),time(TS+1),col(C),
    holds(P,Q1,C,TS), tot_incr(P,Q2,C,TS), tot_decr(P,Q3,C,TS),
    Q=Q1+Q2-Q3.
```

Rules $r8$ and $r9$ capture that Q tokens of color C will be added or removed to/from place P due to firing of transition T at the respective time-step TS. Rules $r10$ and $r11$ aggregate these tokens for each C for each place P (using aggregate assignment `QQ = #sum[...]`) at the respective time-step TS. Rule $r12$ uses the aggregates to compute the next marking of P for color C at the time-step $(TS + 1)$ by subtracting removed tokens and adding added tokens to the current marking. In a biological context, this captures the effect of a process / reaction, which consumes its inputs and produces outputs for the downstream processes. We capture token overconsumption using the rules $a9, a3, a4$ of which $a9$ is a colored replacement for $a2$ and is encoded as follows:

a9:
```
consumesmore(P,TS) :- holds(P,Q,C,TS), tot_decr(P,Q1,C,TS), Q1 > Q.
```

Rule $a9$ determines whether firing set selected by $a1$ will cause overconsumption of tokens at P at time TS by comparing available tokens to aggregate tokens removed as determined by $r11$. Rule $a3$ generalizes the notion of overconsumption, while rule $a4$ eliminates answer with such overconsumption.

In a biological context, conflict (through overconsumption) models the limitation of input substances, which dictate which downstream processes can occur simultaneously.

We remove rules $a5, a6$ from previous encoding to get the *set firing* semantics. Now, we extend the definition (18) of 1-1 correspondence between the execution sequence of Petri Net and the answer-sets of its ASP encoding to Petri Nets with colored tokens as follows.

Definition 38 *Given a Petri Net PN with colored tokens, its initial marking M_0 and its encoding $\Pi(PN, M_0, k, ntok)$ for k-steps and maximum ntok tokens at any place. We say that there is a 1-1 correspondence between the answer sets of $\Pi(PN, M_0, k, ntok)$ and the execution sequences of PN iff for each answer set A of $\Pi(PN, M_0, k, ntok)$, there is a corresponding execution sequence $X = M_0, T_0, M_1, \ldots, M_k, T_k, M_{k+1}$ of PN and for each execution sequence X of PN there is an answer-set A of $\Pi(PN, M_0, k, ntok)$ such that*

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\}$$

$$\{holds(p, q, c, ts) : p \in P, c/q = M_{ts}(p), 0 \leq ts \leq k + 1\}$$
$$= \{holds(p, q, c, ts) : holds(p, q, c, ts) \in A\}$$

Proposition 6 *There is 1-1 correspondence between the answer sets of $\Pi^5(PN^C, M_0, k, ntok)$ and the execution sequences of PN.*

To add *maximal firing semantics*, we add $a6$ as it is and replace $a5$ with $a10$ as follows:

a10:
```
could_not_have(T,TS):-enabled(T,TS),not fires(T,TS), ptarc(S,T,Q,C,TS),
  holds(S,QQ,C,TS), tot_decr(S,QQQ,C,TS), Q > QQ - QQQ.
```

Rule $a10$ captures the fact that transition T, though enabled, could not have fired at TS, as its firing would have caused overconsumption. Rule $a6$ eliminates any answers where an enabled transition could have fired without causing overconsumption but did not. This modification reduces the number of answers produced for the Petri Net in Figure 2.5 to 4. We can encode other firing semantics with similar ease[14]. We now look at how additional extensions can be easily encoded by making small code changes.

[14] For example, if *interleaved* semantics is desired, rules $a10, a6$ can changed to capture and eliminate answer-sets in which more than one transition fires in a firing set as:

a10':
```
more_than_one_fires :- fires(T1,TS),fires(T2,TS),T1!=T2,time(TS).
```

2.10 Extension - Priority Transitions

Priority transitions enable ordering of Petri Net transitions, favoring high priority transitions over lower priority ones (Best and Koutny, 1992). In a biological context, this is used to model primary (or dominant) vs. secondary pathways / processes in a biological system. This prioritization may be due to an intervention (such as prioritizing elimination of a metabolite over recycling it).

Definition 39 (Priority Colored Petri Net) *A* Priority Colored Petri Net *with reset, inhibit, and read arcs is a tuple $PN^{pri} = (P, T, E, C, W, R, I, Q, QW, Z)$, where: $P, T, E, C, W, R, I, Q, QW$ are the same as for PN^C, and $Z : T \rightarrow \mathbb{N}$ is a priority function that assigns priorities to transitions. Lower number signifies higher priority.*

Definition 40 (Enabled Transition) *A transition t_i is enabled in PN^{pri} w.r.t. a marking M ($prenabled_M(t)$) if it would be enabled in PN^C w.r.t. M and there isn't another transition t_j that would be enabled in PN^C (with respect to M) s.t. $Z(t_j) < Z(t_i)$.*

Definition 41 (Firing Set) *A firing set is a set $T_k = \{t_{k_1}, \dots, t_{k_n}\} \subseteq T$ of simultaneously firing transitions of PN^{pri} that are priority enabled and do not conflict w.r.t. to the current marking M_k of PN. A set T_k is not a firing set if there is an priority enabled reset-transition that is not in T_k, i.e. $\exists t : prenabled_{M_k}(t), R(t) \neq \emptyset, t \notin T_k$.* [15]

We add the following facts and rules to encode transition priority and enabled priority transitions:

f19: Facts `transpr(`t_i`,`pr_i`)` where $pr_i = Z(t_i)$ is $t_i's$ priority.

a6': `:-more_than_one_fires.`

[15] See footnote 6

a11: `notprenabled(T,TS) :- enabled(T,TS), transpr(T,P), enabled(TT,TS), transpr(TT,PP), PP < P.`

a12: `prenabled(T,TS) :- enabled(T,TS), not notprenabled(T,TS).`

Rule $a11$ captures that an enabled transition T is not priority-enabled, if there is another enabled transition with higher priority at TS. Rule $a12$ captures that transition T is priority-enabled at TS since there is no enabled transition with higher priority. We replace rules $a1, f10, a10, a6$ with $a13, f20, a14, a15$ respectively to propagate priority as follows:

a13: `{fires(T,TS)} :- prenabled(T,TS), trans(T), time(TS).`

f20: Rules `:- prenabled(`t_j, ts_k`),not fires(`t_j, ts_k`), time(`ts_k`).` for each transition t_j with an incoming reset arc.

a14: `could_not_have(T,TS) :- prenabled(T,TS), not fires(T,TS), ptarc(S,T,Q,C,TS), holds(S,QQ,C,TS), tot_decr(S,QQQ,C,TS), Q > QQ - QQQ.`

a15: `:- not could_not_have(T,TS), time(TS), prenabled(T,TS), not fires(T,TS), trans(T).`

Rules $a13, f10, a14, a15$ perform the same function as $a1, f20, a10, a6$, except that they consider only priority-enabled transitions as compared all enabled transitions.

Proposition 7 *There is 1-1 correspondence between the answer sets of* $\Pi^6(PN^{pri}, M_0, k, ntok)$ *and the execution sequences of* PN^{pri}.

2.11 Extension - Timed Transitions

Biological processes vary in time required for them to complete. Timed transitions (Ramchandani, 1974) model this variation of duration. The timed transitions can be reentrant or non-reentrant[16]. We extend our encoding to allow reentrant timed transitions.

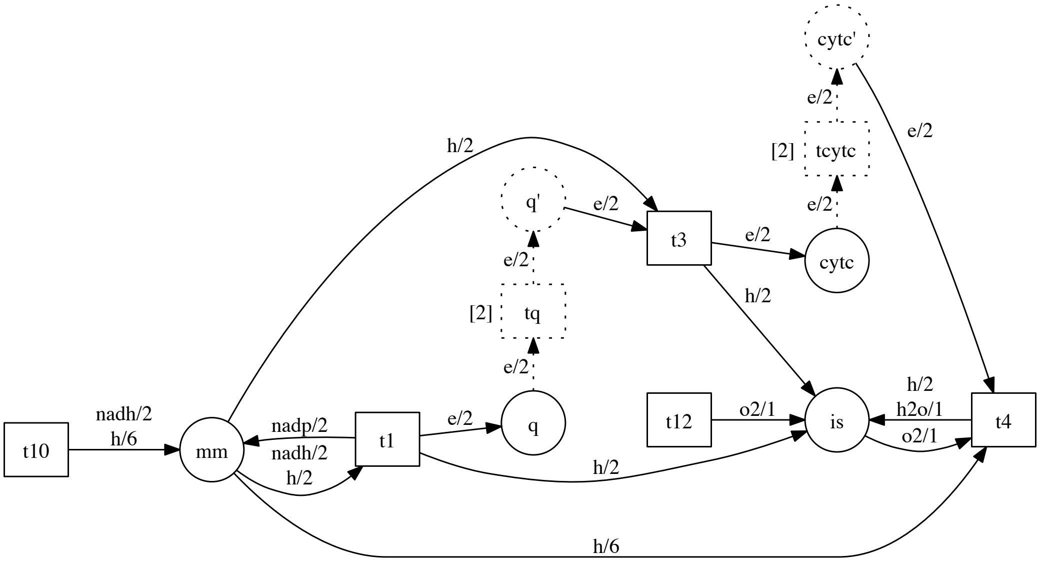


Figure 2.6: An extended version of the Petri Net model from Fig. 2.5. The new transitions $tq, tcytc$ have a duration of 2 each (shown in square brackets ("[ ]") next to the transition). When missing, transition duration is assumed to be 1.

Definition 42 (Priority Colored Petri Net with Timed Transitions) *A* Priority Colored Petri Net with Timed Transitions*, reset, inhibit, and query arcs is a tuple* $PN^D = (P, T, E, C, W, R, I, Q, QW, Z, D)$*, where* $P, T, E, C, W, R, I, Q, QW, Z$ *are the same as for* PN^{pri}*, and* $D : T \to \mathbb{N} \setminus \{0\}$ *is a duration function that assigns positive integer durations to transitions.*

Figure 2.6 shows an extended version of Petri Net model of the Electron Transport Chain (Reece *et al.*, 2010) shown in Figure 2.5. The new transitions tq and $tcytc$

[16] A **reentrant** transition is like a vehicle assembly line, which accepts new parts while working on multiple vehicles at various stages of completion; whereas a **non-reentrant** transition only accepts new input when the current processing is finished.

(shown in dotted outline) are timed transitions modeling the speed of the small carrier molecules, Coenzyme Q (q) and Cytochrome C ($cytc$) as an effect of membrane fluidity. Higher numbers for transition duration represent slower movement of the carrier molecules due to lower fluidity.

Definition 43 (Transition Execution) *A transition t in PN^D consumes tokens from its input places and reset places immediately, while it produces tokens in its output places at the end of transition duration $D(t)$, as follows:*

$$\forall p \in \bullet t, M_{k+1}(p) = M_k(p) - W(p,t)$$

$$\forall p \in t\bullet, M_{k+D(t)}(p) = M_{k+D(t)-1}(p) + W(p,t)$$

$$\forall p \in R(t), M_{k+1}(p) = M_k(p) - M_k(p)$$

Execution in PN^D changes, since the token update from M_k to M_{k+1} can involve transitions that started at some time l before time k, but finish at $k+1$.

Definition 44 (Firing Set Execution) *New marking due to firing set execution is computed as follows:*

$$\forall p \in P \setminus R(T_k), M_{k+1}(p) = M_k(p) - \sum_{t \in T_k, p \in \bullet t} W(p,t) + \sum_{t \in T_l, p \in t\bullet : 0 \leq l \leq k, l+D(t)=k+1} W(t,p)$$

$$\forall p \in R(T_k), M_{k+1}(p) = \sum_{t \in T_l, p \in t\bullet : l \leq k, l+D(t)=k+1} W(t,p)$$

where $R(T_i) = \cup_{t \in T_i} R(t)$.

A timed transition t produces its output $D(t)$ time units after being fired. We replace $f15$ with $f21$ adding transition duration and replace rule $r8$ with $r13$ that produces tokens at the end of transition duration:

f21: Rules `tparc(`$t_i, p_j, n_c, c, ts_k, D(t_i)$`):-time(`$ts_k$`).` for each $(t_i, p_j) \in E^+$, $c \in C$, $n_c = m_{W(t_i,p_j)}(c) : n_c > 0$.

r13: `add(P,Q,T,C,TS):-fires(T,TS0),time(TS0;TS), tparc(T,P,Q,C,TS0,D),`

```
     TS=TS0+D-1.
```

Proposition 8 *There is 1-1 correspondence between the answer sets of* $\Pi^7(PN^D, M_0, k, ntok)$ *and the execution sequences of* PN^D.

Above implementation of timed-transition is reentrant, however, we can easily make these timed transitions non-reentrant by adding rule $e10$ that disallows a transition from being enabled if it is already in progress:

e10: `notenabled(T,TS):-fires(T,TS0), num(N), TS>TS0, tparc(T,P,N,C,TS0,D),`

```
     col(C), time(TS0), time(TS), TS<(TS0+D).
```

2.12 Other Extensions

Other Petri Net extensions can be implemented with similar ease. For example, *Guard Conditions* on transitions can be trivially implemented as a `notenabled/2` rules. *Self Modifying Petri Nets* (Valk, 1978), which allow marking-dependent arc-weights can be implemented in a similar manner as the *Reset Arc* extension in section 2.5. *Object Petri Nets* (Valk, 2004), in which each token is a Petri Net itself can be implemented (using token reference semantics) by adding an additional "network-id" parameter to our encoding, where "id=0" is reserved for system net and higher numbers are used for token nets. Transition coordination between system & token nets is enforced through constraints on transition labels, where transition labels are added as additional facts about transitions.

2.13 Related Work

Petri Nets have been previously encoded in ASP, but the previous implementations have been limited to restricted classes of Petri Nets. For example, 1-safe Petri Net to

ASP translation has been presented in Heljanko and Niemelä (2000), which is limited to binary Petri Nets. Translation of Logic Petri Nets to ASP has been presented in Behrens and Dix (2007), but their model cannot handle numerical aggregation of tokens from multiple input transitions to the same place. Our work focused on problems in the biological domain and is more generalized. We can represent reset arcs, inhibition arcs, priority arcs as well as durative transitions.

2.14 Conclusion

We have presented an encoding of basic Petri Nets in ASP and showed how it can be easily extended to include extension to model various biological constructs. Portions of this work were published in Anwar *et al.* (2013b) and Anwar *et al.* (2013a). In the next chapter we will use Petri Nets and their ASP encoding to model biological pathways to answer questions about them.

Chapter 3

ANSWERING QUESTIONS USING PETRI NETS AND ASP

3.1 Introduction

In this chapter we use various Petri Net extensions presented in Chapter 2 and their ASP encoding to answer question from Reece *et al.* (2010) that were a part of the Second Deep Knowledge Representation Challenge[1].

Definition 45 (Rate) *Rate of product P is defined as the quantity of P produced per unit-time. Rate of an action A is defined as the number of time A occurs per unit-time.*

3.2 Comparing Altered Trajectories due to Reset Intervention

Question 1 *At one point in the process of glycolysis, both dihydroxyacetone phosphate (DHAP) and glyceraldehyde 3-phosphate (G3P) are produced. Isomerase catalyzes the reversible conversion between these two isomers. The conversion of DHAP to G3P never reaches equilibrium and G3P is used in the next step of glycolysis. What would happen to the rate of glycolysis if DHAP were removed from the process of glycolysis as quickly as it was produced?*

Provided Answer:

> "Glycolysis is likely to stop, or at least slow it down. The conversion of the two isomers is reversible, and the removal of DHAP will cause the reaction to shift in that direction so more G3P is converted to DHAP. If

[1] https://sites.google.com/site/2nddeepkrchallenge/

less (or no) G3P were available, the conversion of G3P into DHAP would slow down (or be unable to occur)."

Solution 1 *The process of glycolysis is shown in Fig 9.9 of Campbell's book. Glycolysis splits Glucose into Pyruvate. In the process it produces ATP and NADH. Any one of these can be used to gauge the glycolysis rate, since they will be produced in proportion to the input Glucose. The amount of pyruvate produced is the best choice since it is the direct end product of glycolysis. The ratio of the quantity of pyruvate produced over a representative span of time gives us the glycolysis rate. We assume a steady supply of Glucose is available and also assume that sufficient quantity of various enzymes used in glycolysis is available, since the question does not place any restriction on these substances.*

We narrow our focus to a subsection from Fructose 1,6-bisphosphate (F16BP) to 1,3-Bisphosphoglycerate (BPG13) as shown in Figure 3.1 since that is the part the question is concerned with. We can ignore the linear chain up-stream of F16BP as well as the linear chain down-stream of BPG13 since the amount of F16BP available will be equal to Glucose and the amount of BPG13 will be equal to the amount of Pyruvate given our steady supply assumption.

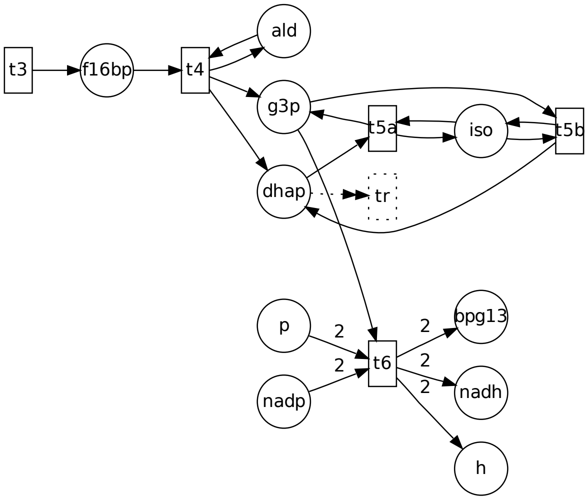


Figure 3.1: Petri Net graph relevant to question 1. "f16bp" is the compound Fructose 1,6-biphosphate, "bpg13" is 1,3-Bisphosphoglycerate. Transition *tr* shown in dotted lines is added to model the elimination of *dhap* as soon as it is produced.

We fulfill the steady supply requirement of Glucose by a source transition-node t3*. We fulfill sufficient enzyme supply by a fixed quantity for each enzyme such that this quantity is in excess of what can be consumed during our simulation interval. Where the simulation interval is the number of time-steps over which we will measure the rate of glycolysis.*

We model the elimination of DHAP as soon as it is produced with a reset arc, shown with a dotted style in Figures 3.1. Such an arc removes all tokens from its source place when it fires. Since we have added it as an unconditional arc, it is always enabled for firing. We encode both situations in ASP with the maximal firing set policy. Both situations (without and with reset arc) are encoded in ASP and run for 10 steps. At the end of those 10 steps the amount of BPG13 is compared to determine the difference in the rate of glycolysis.

In normal situation (without $(dhap, tr)$ *reset arc), unique quantities of "bpg13" from all (2) answer-sets after 10 steps were as follows:*

```
holds(bpg13,14,10)

holds(bpg13,16,10)
```

with reset arc tr, unique quantities of "bpg13" from all (512) answer-sets after 10 steps were as follows:

```
holds(bpg13,0,10)

holds(bpg13,10,10)

holds(bpg13,12,10)

holds(bpg13,14,10)

holds(bpg13,16,10)

holds(bpg13,2,10)

holds(bpg13,4,10)

holds(bpg13,6,10)

holds(bpg13,8,10)
```

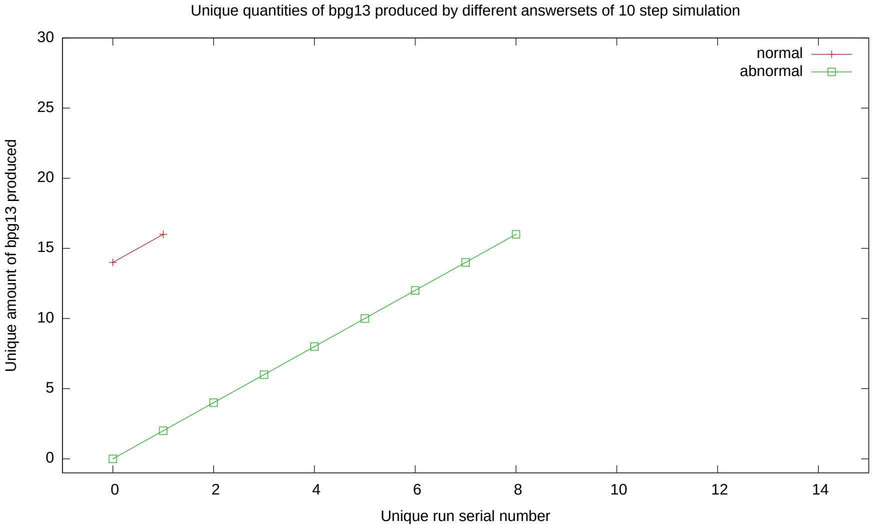


Figure 3.2: Amount of "bpg13" produced in unique answer-sets produced by a 10 step simulation. The graph shows two situations, without the $(dhap, tr)$ reset arc (normal situation) and with the reset arc (abnormal situation). The purpose of this graph is to depict the variation in the amounts of glycolysis produced in various answer sets.

Note that the rate of glycolysis is generally lower when DHAP is immediately consumed. It is as low as zero essentially stopping glycolysis. The range of values are *due to the choice between G3P being converted to DHAP or BPG13. If more G3P is converted to DHAP, then less BPG13 is produced and vice versa. Also, note that if G3P is not converted to BPG13, no NADH or ATP is produced either due to the liner chain from G3P to Pyruvate. The unique quantities of BPG13 are shown in a graphical format in Figure 3.2, while a trend of average quantity of BPG13 produced is shown in Figure 3.3.*

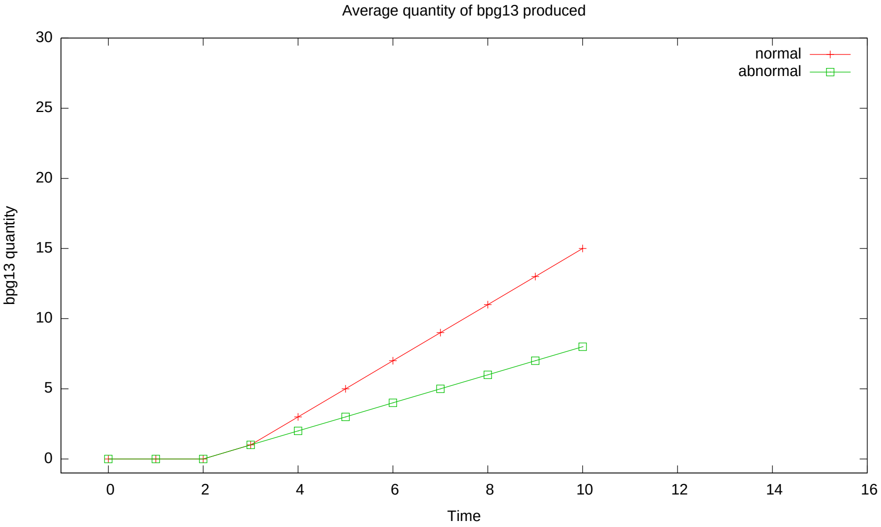


Figure 3.3: Average amount of "bpg13" produced during the 10-step simulation at various time steps. The average is over all answer-sets. The graph shows two situations, without the $(dhap, tr)$ reset arc (normal situation) and with the reset arc (abnormal situation). The divergence in "bpg13" production is clearly shown.

We created a minimal model of the Petri Net in Figure 3.1 by removing enzymes and reactants that were not relevant to the question and did not contribute to the estimation of glycolysis. This is shown in Figure 3.4.

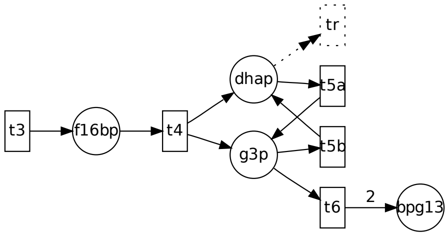


Figure 3.4: Minimal version of the Petri Net graph in Figure 3.1. All reactants that do not contribute to the estimation of the rate of glycolysis have been removed.

Simulating it for 10 steps with the same initial marking as the Petri Net in Figure 3.1 produced the same results as for Figure 3.1.

3.3 Determining Conditions Leading to an Observation

Question 2 *When and how does the body switch to B oxidation versus glycolysis as the major way of burning fuel?*

Provided Answer:

> "The relative volumes of the raw materials for B oxidation and glycolysis indicate which of these two processes will occur. Glycolysis uses the raw material glucose, and B oxidation uses Acyl CoA from fatty acids. When the blood sugar level decreases below its homeostatic level, then B oxidation will occur with available fatty acids. If no fatty acids are immediately available, glucagon and other hormones regulate release of stored sugar and fat, or even catabolism of proteins and nucleic acids, to be used as energy sources."

Solution 2 *The answer provided requires background knowledge about the mechanism that regulates which source of energy will be used. This information is not presented* *in Chapter 9 of Campbell's book, which is the source material of this exercise. However, we can model it based on background information combined with Figure 9.19 of Campbell's book. Our model is presented in Figure 3.5*[2].

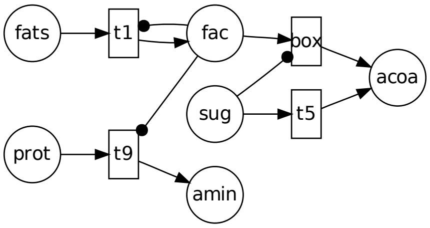


Figure 3.5: Petri Net graph relevant to question 2. "fats" are fats, "prot" are proteins, "fac" are fatty acids, "sug" are sugars, "amin" are amino acids and "acoa" is ACoA. Transition "box" is the beta oxidation, "t5" is glycolysis, "t1" is fat digestion into fatty acids, and "t9" is protein deamination.

We can test the model by simulating it for a time period and testing whether beta oxidation (box) is started when sugar (sug) is finished. We do not need a steady supply of sugar in this case, just enough to be consumed in a few time steps to capture the switch over. Fats and proteins may or may not modeled as a steady supply, since their bioavailability is dependent upon a number of external factors. We assume a steady supply of both and model it with large enough initial quantity that will last beyond the simulation period.

We translate the petri net model into ASP and run it for 10 iterations. Following are the results:

```
holds(acoa,0,0) holds(amin,0,0) holds(fac,0,0) holds(fats,5,0)
holds(prot,3,0) holds(sug,4,0)


fires(t1,0) fires(t5,0) fires(t9,0)
```

[2] We can extend this model by adding expressions to inhibition arcs that compare available substances.

```
holds(acoa,1,1) holds(amin,1,1) holds(fac,1,1) holds(fats,4,1)
holds(prot,3,1) holds(sug,3,1)

fires(t5,1)

holds(acoa,2,2) holds(amin,1,2) holds(fac,1,2) holds(fats,4,2)
holds(prot,3,2) holds(sug,2,2)

fires(t5,2)

holds(acoa,3,3) holds(amin,1,3) holds(fac,1,3) holds(fats,4,3)
holds(prot,3,3) holds(sug,1,3)

fires(t5,3)

holds(acoa,4,4) holds(amin,1,4) holds(fac,1,4) holds(fats,4,4)
holds(prot,3,4) holds(sug,0,4)

fires(box,4)

holds(acoa,5,5) holds(amin,1,5) holds(fac,0,5) holds(fats,4,5)
holds(prot,3,5) holds(sug,0,5)

fires(t1,5) fires(t9,5)

holds(acoa,5,6) holds(amin,2,6) holds(fac,1,6) holds(fats,3,6)
holds(prot,3,6) holds(sug,0,6)

fires(box,6)c
```

```
holds(acoa,6,7) holds(amin,2,7) holds(fac,0,7) holds(fats,3,7)
holds(prot,3,7) holds(sug,0,7)

fires(t1,7) fires(t9,7)

holds(acoa,6,8) holds(amin,3,8) holds(fac,1,8) holds(fats,2,8)
holds(prot,3,8) holds(sug,0,8)

fires(box,8)

holds(acoa,7,9) holds(amin,3,9) holds(fac,0,9) holds(fats,2,9)
holds(prot,3,9) holds(sug,0,9)

fires(t1,9) fires(t9,9)

holds(acoa,7,10) holds(amin,4,10) holds(fac,1,10) holds(fats,1,10)
holds(prot,3,10) holds(sug,0,10)

fires(box,10)
```

We can see that by time-step 4, the sugar supply is depleted and beta oxidation starts occurring.

3.4 Comparing Altered Trajectories due to Accumulation Intervention

Question 3 *ATP is accumulating in the cell. What affect would this have on the rate of glycolysis? Explain.*

Provided Answer:

> "ATP and AMP regulate the activity of phosphofructokinase. When there is an abundance of AMP in the cell, this indicates that the rate of ATP

> consumption is high. The cell is in need for more ATP. If ATP is accumulating in the cell, this indicates that the cell's demand for ATP had decreased. The cell can decrease its production of ATP. Therefore, the rate of glycolysis will decrease."

Solution 3 *Control of cellular respiration is summarized in Fig 9.20 of Campbell's book. We can gauge the rate of glycolysis by the amount of Pyruvate produced, which is the end product of glycolysis. We assume a steady supply of glucose is available. Its availability is not impacted by any of the feedback mechanism depicted in Fig 9.20 of Campbell's book or restricted by the question. We can ignore the respiration steps after glycolysis, since they are directly dependent upon the end product of glycolysis, i.e. Pyruvate. These steps only reinforce the negative effect of ATP. The Citrate feed-back shown in Campbell's Fig 9.20 is also not relevant to the question, so we can assume a constant level of it and leave it out of the picture. Another simplification that we do is to treat the inhibition of Phosphofructokinase (PFK) by ATP as the inhibition of glycolysis itself. This is justified, since PFK is on a linear path from Glucose to Fructose 1,6-bisphosphate (F16BP), and all downstream product quantities are directly dependent upon the amount of F16BP (as shown in Campbell's Fig 9.9), given steady supply of substances involved in glycolysis. Our assumption also applies to ATP consumed in Fig 9.9. Our simplified picture is shown in Figure 3.6 as a Petri Net.*

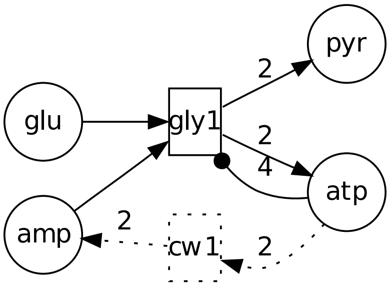


Figure 3.6: Petri Net graph relevant to question 3. “glu” is Glucose, “pyr” is Pyruvate. Transitions “gly1” represents glycolysis and “cw1” is cellular work that consumes ATP and produces AMP. Transition “gly1” is inhibited only when the number of *atp* tokens is greater than 4.

We model cellular work that recycles ATP to AMP (see p/181 of Campbell's book) by the $cw1$ *transition, shown in dotted style. In normal circumstances, this arc does not let ATP to collect. If we reduce the arc-weights incident on* $cw1$ *to 1, we get the situation where less work is being done and some ATP will collect, as a result glycolysis will pause and resume. If we remove* $cw1$ *(representing no cellular work), ATP will start accumulating and glycolysis will stop. We use an arbitrary arc-weight of* 4 *on the inhibition arc* $(atp, gly1)$ *to model an elevated level of ATP beyond normal that would cause inhibition* [3]. *We encode all three situations in ASP with maximal firing set policy. We run them for 10 steps and compare the quantity of pyruvate produced to determine the difference in the rate of glycolysis.*

In normal situation when cellular work is being performed ($cw1$ *arc is present), unique quantities of “pyr” after 10 step are as follows:*

```
holds(pyr,20,10)
```

[3] An alternate modeling would be compare the number of tokens on the *amp* node and the *atp* node and set a level-threshold that inhibits *gly*1. Such technique is common in colored-peri nets.

when the cellular work is reduced, i.e. ($(atp, cw1)$, $(cw1, amp)$ arc weights changed to 1), unique quantities of "pyr" after 10 steps are as follows:

```
holds(pyr,14,10)
```

with no cellular work ($cw1$ arc removed), unique quantities of "pyr" after 10 steps are as follows:

```
holds(pyr,6,10)
```

The results show the rate of glycolysis reducing as the cellular work decreases to the point where it stops once ATP reaches the inhibition threshold. Higher numbers of ATP produced in later steps of cellular respiration will reinforce this inhibition even more quickly. Trend of answers from various runs is shown in Figure 3.7.

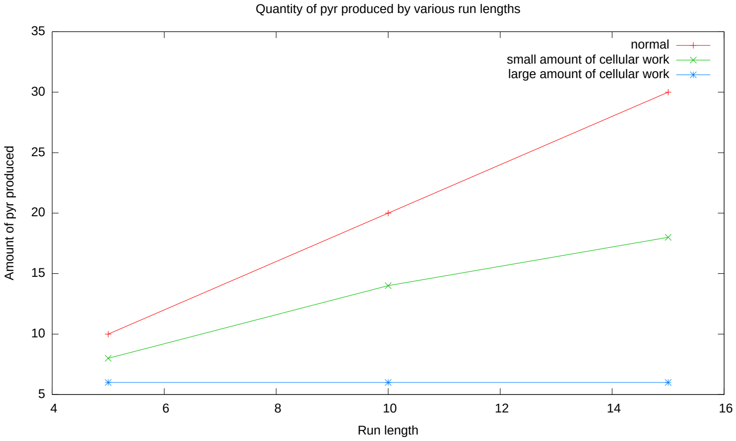


Figure 3.7: Amount of pyruvate produced from various lengths of runs.

3.5 Comparing Altered Trajectories due to Initial Value Intervention

Question 4 *A muscle cell had used up its supply of oxygen and ATP. Explain what affect would this have on the rate of cellular respiration and glycolysis?*

Provided Answer:

> "Oxygen is needed for cellular respiration to occur. Therefore, cellular respiration would stop. The cell would generate ATP by glycolysis only. Decrease in the concentration of ATP in the cell would stimulate an increased rate of glycolysis in order to produce more ATP."

Solution 4 *Figure 9.18 of Campbell's book gives the general idea of what happens when oxygen is not present. Figure 9.20 of Campbell's book shows the control of glycolysis by ATP. To formulate the answer, we need pieces from both.*

ATP inhibits Phosphofructokinase (Fig 9.20 of Campbell), which is an enzyme used in glycolysis. No ATP means that enzyme is no longer inhibited and glycolysis can proceed at full throttle. Pyruvate either goes through aerobic respiration when oxygen is present or it goes through fermentation when oxygen is absent (Fig 9.18 of Campbell). We can monitor the rate of glycolysis and cellular respiration by observing these operations occurring (by looking at corresponding transition firing) over a simulation time period. Our simplified Petri Net model is shown in Figure 3.8.

We ignore the details of processes following glycolysis, except that these steps produce additional ATP. We do not need an exact number of ATP produced as long as we keep it higher than the ATP produced by glycolysis. Higher numbers will just have a higher negative feed-back (or inhibition) effect on glycolysis. We ignore citrate's inhibition of glycolysis since that is not relevant to the question and since it gets recycled by the citric acid cycle (see Fig 9.12 of Campbell). We also ignore AMP, since it is not relevant to the question, by assuming sufficient supply to maintain glycolysis. We also assume continuous cellular work consuming ATP, without that ATP will accumulate almost immediately and stop glycolysis.

We assume a steady supply of glucose is available to carry out glycolysis and fulfill this requirement by having a quantity in excess of the consumption during our simulation interval. All other substances participating in glycolysis are assumed to be available in a steady supply so that glycolysis can continue.

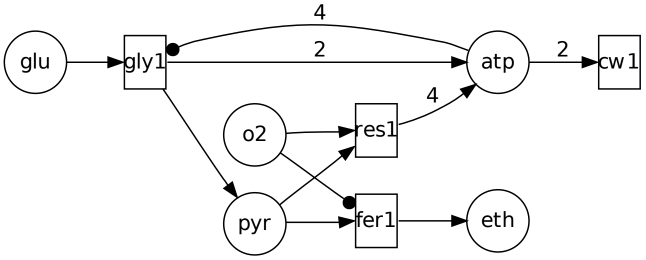


Figure 3.8: Petri Net graph relevant to question 4. “glu” is Glucose, “pyr” is Pyruvate, “atp” is ATP, “eth” is ethenol or other products of fermentation, and “o2” is Oxygen. Transitions “gly1” represents glycolysis, “res1” is respiration in presence of oxygen, “fer1” is fermentation when no oxygen is present, and “cw1” is cellular work that consumes ATP. Transition “gly1” is inhibited only when the number of *atp* tokens is greater than 4.

We then consider two scenarios, one where oxygen is present and where oxygen is absent and determine the change in rate of glycolysis and respiration by counting the firings of their respective transitions. We encode both situations in ASP with maximal firing set policy. Both situations are executed for 10 steps. At the end of those steps the firing count of “gly1” and “res1” is computed and compared to determine the difference in the rates of glycolysis and respiration respectively.

In the normal situation (when oxygen is present), we get the following answer sets:

```
fires(gly1,0)

fires(cw1,1) fires(gly1,1) fires(res1,1)

fires(cw1,2) fires(res1,2)

fires(cw1,3)
fires(cw1,4)
fires(cw1,5) fires(gly1,5)
fires(cw1,6) fires(gly1,6) fires(res1,6)
fires(cw1,7) fires(res1,7)
fires(cw1,8)
fires(cw1,9)
fires(cw1,10)
```

while in the abnormal situation (when oxygen is absent), we get the following firings:

```
fires(gly1,0)
fires(cw1,1) fires(fer1,1) fires(gly1,1)
fires(cw1,2) fires(fer1,2) fires(gly1,2)
fires(cw1,3) fires(fer1,3) fires(gly1,3)
fires(cw1,4) fires(fer1,4) fires(gly1,4)
fires(cw1,5) fires(fer1,5) fires(gly1,5)
fires(cw1,6) fires(fer1,6) fires(gly1,6)
fires(cw1,7) fires(fer1,7) fires(gly1,7)
fires(cw1,8) fires(fer1,8) fires(gly1,8)
fires(cw1,9) fires(fer1,9) fires(gly1,9)
fires(cw1,10) fires(fer1,10) fires(gly1,10)
```

Note that the number of firings of glycolysis for normal situation is lower when oxygen is present and higher when oxygen is absent. While, the number of firings is zero when no oxygen is present. Thus, respiration stops when no oxygen is present and the need of ATP by cellular work is fulfilled by a higher amount of glycolysis. Trend from various runs is shown in Figure 3.9.

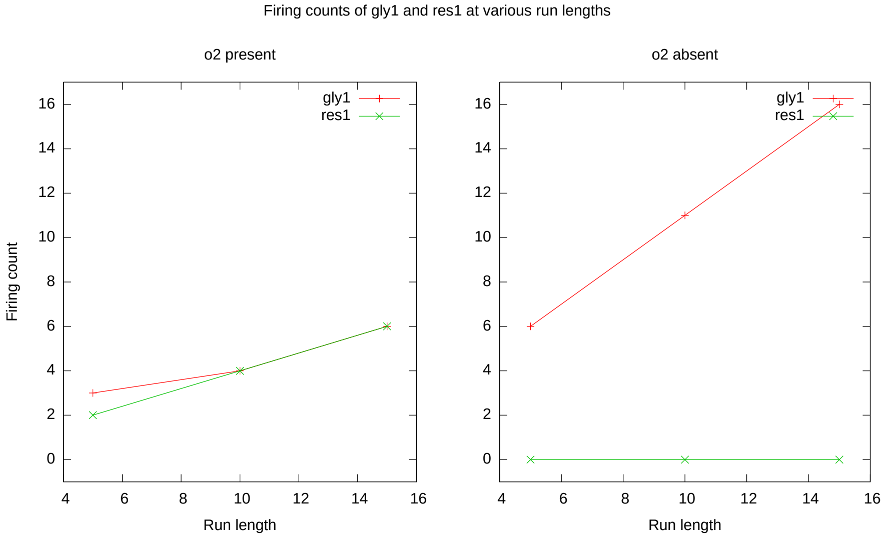


Figure 3.9: Firing counts of glycolysis (gly1) and respiration (res1) for different simulation lengths for the petri net in Figure 3.8

3.6 Comparing Altered Trajectories due to Inhibition Intervention

Question 5 *The final protein complex in the electron transport chain of the mitochondria is non-functional. Explain the effect of this on pH of the intermembrane space of the mitochondria.*

Provided Answer:

> "The H+ ion gradient would gradually decrease and the pH would gradually increase. The other proteins in the chain are still able to produce the H+ ion gradient. However, a non-functional, final protein in the electron transport chain would mean that oxygen is not shuttling electrons away from the electron transport chain. This would cause a backup in the chain, and the other proteins in the electron transport chain would no longer be able to accept electrons and pump H+ ions into the intermembrane space. A concentration decrease in the H+ ions means an increase in the pH."

Solution 5 *The electron transport chain is shown in Fig 9.15 (1) of Campbell's book. In order to explain the effect on pH, we will show the change in the execution of the electron transport chain with both a functioning and non-functioning final protein. Since pH depends upon the concentration of H+ ions, we will quantify the difference its quantity in the intermembrane space in both scenarios as well. We assume that a steady input of NADH, FADH2, H+ and O2 are available in the mitochondrial matrix. We also assume an electron carrying capacity of 2 for both ubiquinone (Q)*[4] *and cytochrome c (Cyt c). This carrying capacity is background information not provided in Campbell's Chapter 9. As with previous questions, we fulfill the steady supply requirement of substances by having input quantities in excess of what would be consumed during our simulation interval.*

[4] http://www.benbest.com/nutrceut/CoEnzymeQ.html

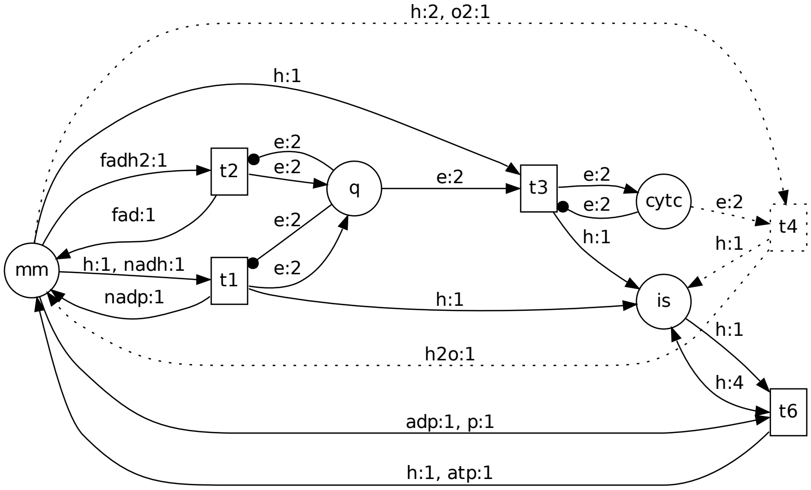


Figure 3.10: Petri Net graph relevant to question 5. "is" is the intermembrane space, "mm" is mitochondrial matrix, "q" is ubiquinone and "cytc" is cytochrome c. The inhibition arcs $(q, t1)$, $(q, t2)$ and $(cytc, t3)$ capture the electron carrying capacities of q and $cytc$. Over capacity will cause backup in electron transport chain. Tokens are colored, e.g. "h:1, nadh:1" specify one token of h and $nadh$ each. Token types are "h" for H+, "nadh" for NADH, "nadp" for NADP, "fadh2" for FADH2, "fad" for FAD, "e" for electrons, "o2" for oxygen and "h2o" for water. We remove $t4$ to model non-functioning protein complex IV.

We model this problem as a colored petri net shown in Figure 3.10. The normal situation is made up of the entire graph. The abnormal situation (with non-functional final protein complex) is modeled by removing transition $t4$ *from the graph*[5]*. We encode both situations in ASP with maximal firing set policy. Both are run for 10 steps. The amount of h (H+) is compared in the is (intermembrane space) to determine change in pH and the firing sequence is compared to explain the effect.*

In normal situation (entire graph), we get the following:

```
fires(t1,0) fires(t2,0)

fires(t3,1)
```

[5] Alternatively, we can model a non-functioning transition by attaching an inhibition arc to it with one token at its source place

```
fires(t1,2) fires(t2,2) fires(t3,2) fires(t4,2)

fires(t3,3) fires(t4,3) fires(t6,3)

fires(t1,4) fires(t2,4) fires(t3,4) fires(t4,4) fires(t6,4)

fires(t3,5) fires(t4,5) fires(t6,5)

fires(t1,6) fires(t2,6) fires(t3,6) fires(t4,6) fires(t6,6)

fires(t3,7) fires(t4,7) fires(t6,7)

fires(t1,8) fires(t2,8) fires(t3,8) fires(t4,8) fires(t6,8)

fires(t3,9) fires(t4,9) fires(t6,9)

fires(t1,10) fires(t2,10) fires(t3,10) fires(t4,10) fires(t6,10)


holds(is,15,h,10)
```

with $t4$ *removed, we get the following:*

```
fires(t1,0) fires(t2,0)

fires(t3,1)

fires(t1,2) fires(t2,2) fires(t3,2)

fires(t6,3)

fires(t6,4)


holds(is,2,h,10)
```

Note that the amount of H+ (h) produced in the intermembrane space (is) is much smaller when the final protein complex is non-functional ($t4$ *removed). Lower H+ translates to higher pH. Thus, the pH of intermembrane space will increase as a result of nonfunctional final protein. Also, note that the firing of* $t3$*,* $t1$ *and* $t2$ *responsible for shuttling electrons also stop very quickly when* $t4$ *no longer removes the electrons (e) from Cyt c (cytc) to produce* H_2O*. This is because cytc and q are at their capacity on electrons that they can carry and stop the electron transport chain by inhibiting transitions* $t3$*,* $t2$ *and* $t1$*. Trend for various runs is shown in Figure 3.11.*

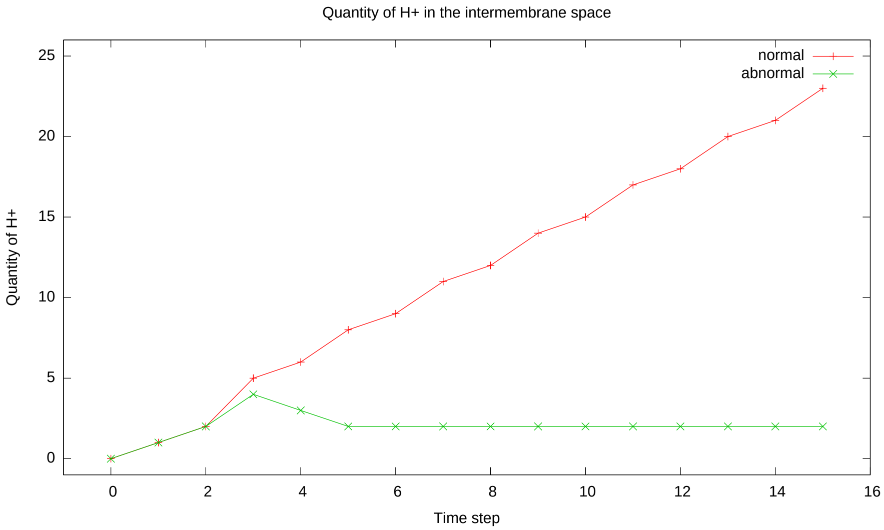


Figure 3.11: Simulation of Petri Net in Figure 3.10. In a complete model of the biological system, there will be a mechanism that keeps the quantity of H+ in check in the intermembrane space and will plateau at some point.

3.7 Comparing Altered Trajectories due to Gradient Equilization Intervention

Question 6 *Exposure to a toxin caused the membranes to become permeable to ions. In a mitochondrion, how would this affect the pH in the intermembrane space and also ATP production?*

Provided Answer:

> “The pH of the intermembrane space would decrease as H+ ions diffuse through the membrane, and the H+ ion gradient is lost. The H+ gradient is essential in ATP production b/c facilitated diffusion of H+ through ATP synthase drives ATP synthesis. Decreasing the pH would lead to a decrease in the rate of diffusion through ATP synthase and therefore a decrease in the production of ATP.”

Solution 6 *Oxidative phosphorylation is shown in Fig 9.15 of Campbell’s book. In order to explain the effect on pH in the intermembrane space and the ATP production* *we will show the change in the amount of H+ ions in the intermembrane space as well as the amount of ATP produced when the inner mitochondrial membrane is impermeable and permeable. Note that the concentration of H+ determines the pH. we have chosen to simplify the diagram by not having FADH2 in the picture. Its removal does not change the response, since it provides an alternate input mechanism to electron transport chain. We will assume that a steady input of NADH, H+, O2, ADP and P is available in the mitochondrial matrix. We also assume an electron carrying capacity of 2 for both ubiquinone (Q) and cytochrome c (Cyt c). We fulfill the steady supply requirement of substances by having input quantities in excess of what would be consumed during our simulation interval.*

We model this problem as a colored petri net shown in Figure 3.12. Transition $t6, t7$ *shown in dotted style are added to model the abnormal situation*[6]*. They capture the diffusion of H+ ions back from Intermembrane Space to the Mitochondrial matrix. One or both may be enabled to capture degrees of permeability. we have added a condition on the firing of* $t5$ *(ATP Synthase activation) to enforce gradient to pump ATP Synthase.*

[6] If reverse permeability is also desired additional arcs may be added from mm to is

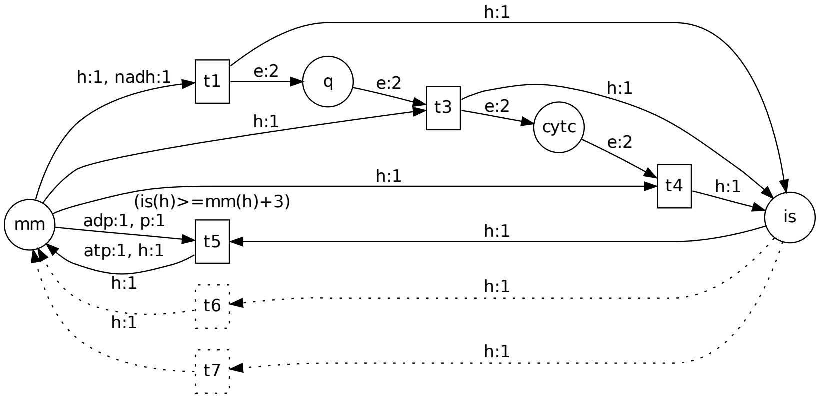


Figure 3.12: Petri Net graph relevant to question 6. “is” is the intermembrane space, “mm” is mitochondrial matrix, “q” is ubiquinone and “cytc” is cytochrome c. Tokens are colored, e.g. “h:1, nadh:1” specify one token of h and $nadh$ each. Token types are “h” for H+, “nadh” for NADH, “e” for electrons, “o2” for oxygen, “h2o” for water, “atp” for ATP and “adp” for ADP. We add $t6, t7$ to model cross domain diffusion from intermembrane space to mitochondrial matrix. One or both of $t6, t7$ may be enabled at a time to control the degree of permeability. The text above “t5” is an additional condition which must be satisfied for “t5” to be enabled.

We encode both situations in ASP with maximal firing set policy. Both are run for 10 steps each and the amount of h and atp is compared to determine the effect of pH and ATP production. We capture the gradient requirement as the following ASP code[7]*:*

```
notenabled(T,TS) :-
  T==t5, C==h, trans(T), col(C), holds(is,Qis,C,TS),
  holds(mm,Qmm,C,TS), Qmm+3 > Qis,
  num(Qis;Qmm), time(TS).
```

In the normal situation, we get the following h token distribution after 10 steps:

```
holds(is,11,h,10) holds(mm,1,h,10) holds(mm,6,atp,10)
```

[7] We can alternatively model this by having a threshold arc from “is” to “t5” if only a minimum trigger quantity is required in the intermembrane space.

we change the permeability to 1 (t6 enabled), we get the following token distribution instead:

```
holds(is,10,h,10) holds(mm,2,h,10) holds(mm,5,atp,10)
```

we change the permeability to 2 (t6, t7 enabled), the distribution changes as follows:

```
holds(is,8,h,10) holds(mm,4,h,10) holds(mm,2,atp,10)
```

Note that as the permeability increases, the amount of H+ (h) in intermembrane space (is) decreases and so does the amount of ATP (h) in mitochondrial matrix. Thus, an increase in permeability will increase the pH. If the permeability increases even beyond 2, no ATP will be produced from ADP due to insufficient H+ gradient. Trend from various runs is shown in Figure 3.13.

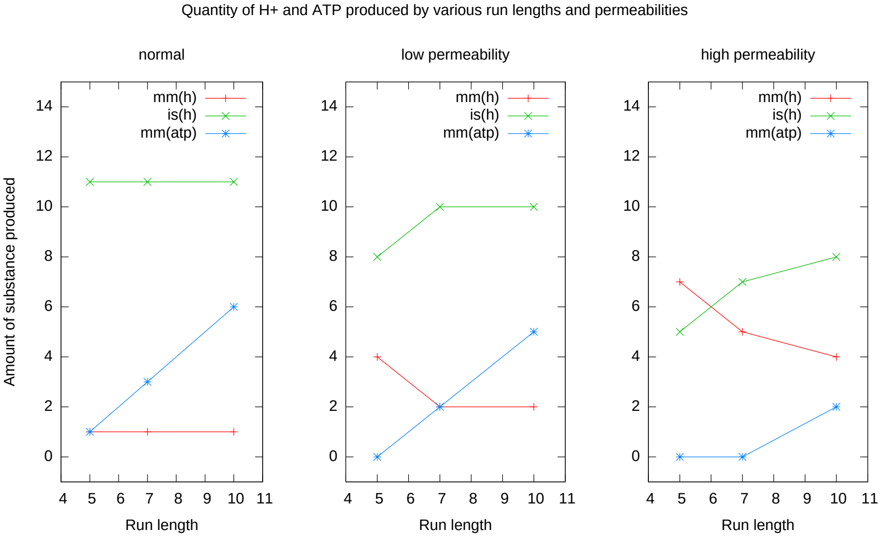


Figure 3.13: Quantities of H+ and ATP at various run lengths and permeabilities for the Petri Net model in Figure 3.12.

3.8 Comparing Altered Trajectories due to Delay Intervention

Question 7 *Membranes must be fluid to function properly. How would decreased fluidity of the membrane affect the efficiency of the electron transport chain?*

Provided Answer:

> “Some of the components of the electron transport chain are mobile electron carriers, which means they must be able to move within the membrane. If fluidity decreases, these movable components would be encumbered and move more slowly. This would cause decreased efficiency of the electron transport chain.”

Solution 7 *The answer deals requires background knowledge about fluidity and how it relates to mobile carriers not presented in the source chapter. From background knowledge we find that the higher the fluidity, higher the mobility. The electron transport chain is presented in Fig 9.15 of Campbell’s book. From background knowledge, we know that the efficiency of the electron transport chain is measured by the amount of ATP produced per NADH/FADH2. The ATP production happens due to the gradient of H+ ions across the mitochondrial membrane. The higher the number of H+ ions in the intermembrane space, the higher would be the gradient and the resulting efficiency. So we measure the efficiency of the chain by the amount of H+ transported to intermembrane space, assuming all other (fixed) molecules behave normally. This is a valid assumption since H+ transported from mitochondrial matrix is directly proportional to the amount of electrons shuttled through the non-mobile complexes and there is a linear chain from the electron carrier to oxygen.*

We model this chain using a Petri Net with durative transitions shown in Figure 3.14. Higher the duration of transitions, lower the fluidity of the membrane. We assume that a steady supply of NADH and H+ is available in the mitochondrial membrane. We fulfill this requirement by having quantities in excess of what will be consumed during the simulation. We ignore FADH2 from the diagram, since it is just an alternate path to the electron chain. Using it by itself will produce a lower *number of H+ transporter to intermembrane space, but it will not change the result. We compare the amount of H+ transported into the intermembrane space to gauge the efficiency of the electron transport chain. More efficient the chain is, more H+ will it transport.*

We model three scenarios: normal fluidity, low fluidity with transitions $t3$ *and* $t4$ *having an execution time of 2 and an lower fluidity with transitions* $t3, t4$ *having execution time of 4.*

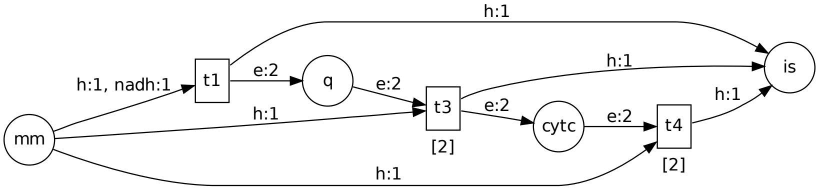


Figure 3.14: Petri Net graph relevant to question 7. "is" is the intermembrane space, "mm" is mitochondrial matrix, "q" is ubiquinone and "cytc" is cytochrome c. Tokens are colored, e.g. "h:1, nadh:1" specify one token of *h* and *nadh* each. Token types are "h" for H+, "nadh" for NADH, "e" for electrons. Numbers in square brackets below the transition represent transition durations with default of one time unit, if the number is missing.

We encode these cases in ASP with maximal firing set semantics and simulate them for 10 time steps. For the normal fluidity we get:

```
holds(is,27,h,10)
```

for low fluidity we get:

```
holds(is,24,h,10)
```

for lower fluidity we get:

```
holds(is,18,h,10)
```

Note that as the fluidity decreases, so does the amount of H+ transported to intermembrane space, pointing to lower efficiency of electron transport chain. Trend of various runs is shown in Figure 3.15.

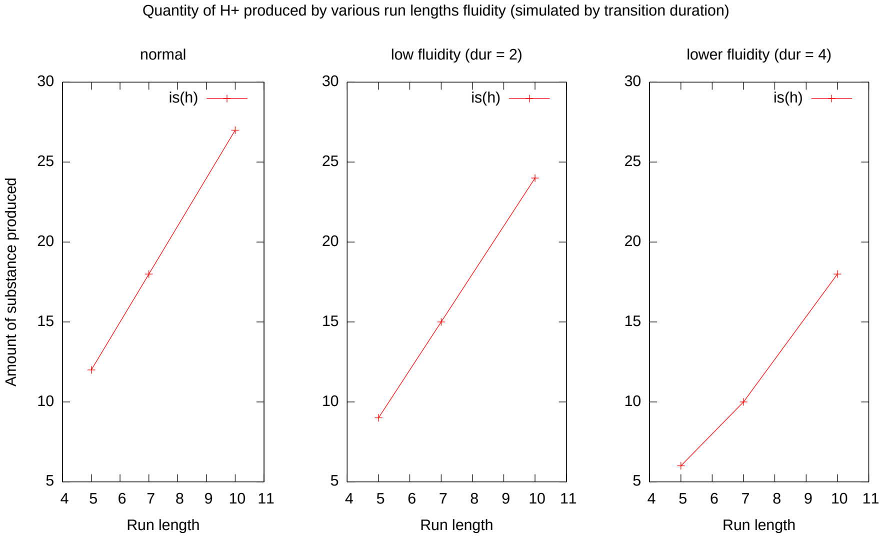


Figure 3.15: Quantities of H+ produced in the intermembrane space at various run lengths and fluidities for the Petri Net model in Figure 3.14.

3.9 Comparing Altered Trajectories due to Priority and Read Interventions

Question 8 *Phosphofructokinase (PFK) is allosterically regulated by ATP. Considering the result of glycolysis, is the allosteric regulation of PFK likely to increase or decrease the rate of activity for this enzyme?*

Provided Answer:

> "Considering that one of the end products of glycolysis is ATP, PFK is inhibited when ATP is abundant and bound to the enzyme. The inhibition decreases ATP production along this pathway."

Solution 8 *Regulation of Phosphofructokinase (PFK) is presented in Figure 9.20 of Campbell's book. We ignore substances upstream of Fructose 6-phosphate (F6P) by assuming they are available in abundance. We also ignore AMP by assuming normal supply of it. We also ignore any output of glycolysis other than ATP production since the downstream processes ultimately produce additional ATP. Citric acid is also* *ignored since it is not relevant to the question at hand. We monitor the rate of activity of PFK by the number of times it gets used for glycolysis.*

We model this problem as a Petri Net shown in Figure 3.16. Allosteric regulation of PFK is modeled by a compound "pfkatp" which represents PFK's binding with ATP to form a compound. Details of allosteric regulation are not provided in the same chapter, they are background knowledge from external sources. Higher than normal quantity of ATP is modeled by a threshold arc (shown with arrow-heads at both ends) with an arbitrary threshold value of 4. This number can be increased as necessary. The output of glycolysis and down stream processes "t3" has been set to 2 to run the simulation in a reasonable amount of time. It can be made larger as necessary. The allosteric regulation transition "t4" has also been given a higher priority than glycolysis transition "t3". This way, ATP in excess will cause PFK to be converted to PFK+ATP compound, reducing action of PFK.

We assume that F6P is available in sufficient quantity and so is PFK. This requirement is fulfilled by having more quantity than can be consumed in the simulation duration. We model both the normal situation including transition t4 *shown in dotted style and the abnormal situation where* t4 *is removed.*

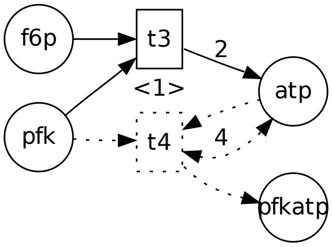


Figure 3.16: Petri Net graph relevant to question 8. "pfk" is phosphofructokinase, "f6p" is fructose 6-phosphate, "atp" is ATP and "pfkatp" is the pfk bound with atp for allosteric regulation. Transition "t3" represents enzymic action of pfk, "t4" represents the binding of pfk with atp. The double arrowed arc represents a threshold arc, which enables "t4" when there are at least 4 tokens available at "atp". Numbers above transitions in angular brackets represent arc priorities.

We encode both situations in ASP with maximal firing set policy and run them for 10 time steps. At the end of the run we compare the firing count of transition t3 *for both cases. For the normal case (with* t4*), we get the following results:*

```
holds(atp,0,c,0) holds(f6p,20,c,0) holds(pfk,20,c,0) holds(pfkatp,0,c,0)

fires(t3,0)

holds(atp,2,c,1) holds(f6p,19,c,1) holds(pfk,19,c,1) holds(pfkatp,0,c,1)

fires(t3,1)

holds(atp,4,c,2) holds(f6p,18,c,2) holds(pfk,18,c,2) holds(pfkatp,0,c,2)

fires(t4,2)

holds(atp,3,c,3) holds(f6p,18,c,3) holds(pfk,17,c,3) holds(pfkatp,1,c,3)

fires(t3,3)

holds(atp,5,c,4) holds(f6p,17,c,4) holds(pfk,16,c,4) holds(pfkatp,1,c,4)

fires(t4,4)

holds(atp,4,c,5) holds(f6p,17,c,5) holds(pfk,15,c,5) holds(pfkatp,2,c,5)

fires(t4,5)

holds(atp,3,c,6) holds(f6p,17,c,6) holds(pfk,14,c,6) holds(pfkatp,3,c,6)

fires(t3,6)

holds(atp,5,c,7) holds(f6p,16,c,7) holds(pfk,13,c,7) holds(pfkatp,3,c,7)
```

```
fires(t4,7)
holds(atp,4,c,8) holds(f6p,16,c,8) holds(pfk,12,c,8) holds(pfkatp,4,c,8)
fires(t4,8)
holds(atp,3,c,9) holds(f6p,16,c,9) holds(pfk,11,c,9) holds(pfkatp,5,c,9)
fires(t3,9)
holds(atp,5,c,10) holds(f6p,15,c,10) holds(pfk,10,c,10) holds(pfkatp,5,c,10)
fires(t4,10)
```

Note that t3 fires only when the ATP falls below our set threshold, above it PFK is converted to PFK+ATP compound via t4. For the abnormal case (without t4) we get the following results:

```
holds(atp,0,c,0) holds(f6p,20,c,0) holds(pfk,20,c,0) holds(pfkatp,0,c,0)
fires(t3,0)
holds(atp,2,c,1) holds(f6p,19,c,1) holds(pfk,19,c,1) holds(pfkatp,0,c,1)
fires(t3,1)
holds(atp,4,c,2) holds(f6p,18,c,2) holds(pfk,18,c,2) holds(pfkatp,0,c,2)
fires(t3,2)
holds(atp,6,c,3) holds(f6p,17,c,3) holds(pfk,17,c,3) holds(pfkatp,0,c,3)
fires(t3,3)
holds(atp,8,c,4) holds(f6p,16,c,4) holds(pfk,16,c,4) holds(pfkatp,0,c,4)
fires(t3,4)
holds(atp,10,c,5) holds(f6p,15,c,5) holds(pfk,15,c,5) holds(pfkatp,0,c,5)
fires(t3,5)
holds(atp,12,c,6) holds(f6p,14,c,6) holds(pfk,14,c,6) holds(pfkatp,0,c,6)
fires(t3,6)
holds(atp,14,c,7) holds(f6p,13,c,7) holds(pfk,13,c,7) holds(pfkatp,0,c,7)
fires(t3,7)
holds(atp,16,c,8) holds(f6p,12,c,8) holds(pfk,12,c,8) holds(pfkatp,0,c,8)
fires(t3,8)
holds(atp,18,c,9) holds(f6p,11,c,9) holds(pfk,11,c,9) holds(pfkatp,0,c,9)
fires(t3,9)
holds(atp,20,c,10) holds(f6p,10,c,10) holds(pfk,10,c,10) holds(pfkatp,0,c,10)
```

```
fires(t3,10)
```

Note that when ATP is not abundant, transition t3 fires continuously, which represents the enzymic activity that converts F6P to downstream substances. Trend of various runs is shown in Figure 3.17.

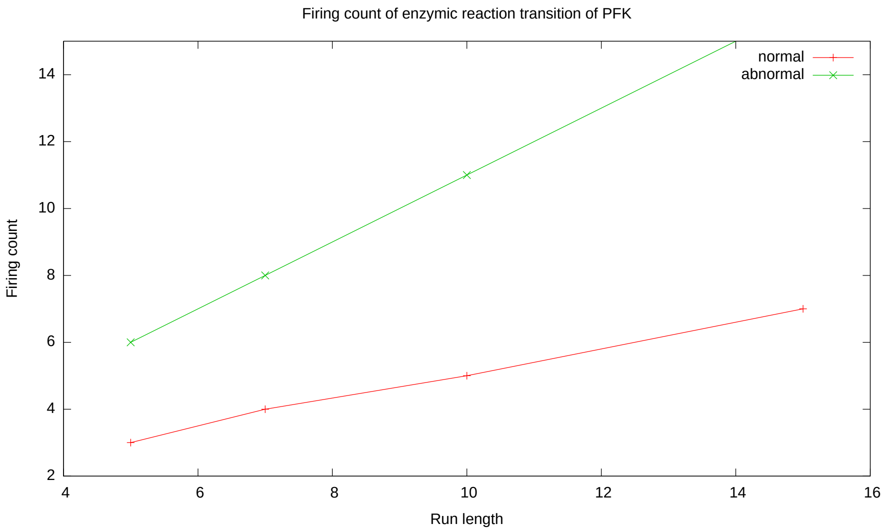


Figure 3.17: Petri Net model in Figure 3.16.

3.10 Comparing Altered Trajectories due to Automatic Conversion Intervention

Question 9 *How does the oxidation of NADH affect the rate of glycolysis?*

Provided Answer:

> "NADH must be oxidized back to NAD+ in order to be used in glycolysis. Without this molecule, glycolysis cannot occur."

Solution 9 *Cellular respiration is summarized in Fig 9.6 of Campbell's book. NAD+ is reduced to NADH during glycolysis (see Campbell's Fig 9.9) during the process of converting Glyceraldehyde 3-phosphate (G3P) to 1,3-Bisphosphoglycerate (BPG13). NADH is oxidized back to NAD+ during oxidative phosphorylation by the electron* *transport chain (see Campbell's Fig 9.15). We can gauge the rate of glycolysis by the amount of Pyruvate produced, which is the end product of glycolysis. We simplify our model by abstracting glycolysis as a black-box that takes Glucose and NAD+ as input and produces NADH and Pyruvate as output, since there is a linear chain from Glucose to Pyruvate that depends upon the availability of NAD+. We also abstract oxidative phosphorylation as a black-box which takes NADH as input and produces NAD+ as output. None of the other inner workings of oxidative phosphorylation play a role in answering the question assuming they are functioning normally. We also ignore the pyruvate oxidation and citric acid cycle stages of cellular respiration since their end products only provide additional raw material for oxidative phosphorylation and do not add value to answering the question.*

We assume a steady supply of Glucose and all other substances used in glycolysis but a limited supply of NAD+, since it can be recycled from NADH and we want to model its impact. We fulfill the steady supply requirement of Glucose with sufficient initial quantity in excess of what will be consumed during our simulation interval. We also ensure that we have sufficient initial quantity of NAD+ to maintain glycolysis as long as it can be recycled.

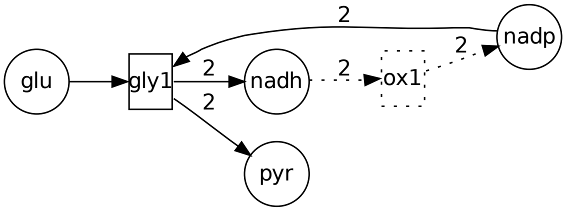


Figure 3.18: Petri Net graph relevant to question 9. "glu" represents glucose, "gly1" represents glycolysis, "pyr" represents pyruvate, "ox1" represents oxidative phosphorylation, "nadh" represents NADH and "nadp" represents NAD+. "ox1" is removed to model stoppage of oxidation of NADH to NAD+.

Figure 3.18 is a Petri Net representation of our simplified model. Normal situation is modeled by the entire graph, where NADH is recycled back to NAD+, while the abnormal situation is modeled by the graph with the transition $ox1$ (shown in dotted style) removed. We encode both situations in ASP with the maximal firing set policy. Both situations are run for 5 steps and the amount of pyruvate is compared to determine the difference in the rate of glycolysis.

In normal situation (with $ox1$ transition), unique quantities of pyruvate (pyr) are as follows:

```
fires(gly1,0)
fires(gly1,1)
fires(gly1,2)
fires(gly1,3)
fires(gly1,4)

holds(pyr,10,5)
```

while in abnormal situation (without $ox1$ transition), unique quantities of pyruvate are as follows:

```
fires(gly1,0)
fires(gly1,1)
fires(gly1,2)

holds(pyr,6,5)
```

Note that the rate of glycolysis is lower when NADH is not recycled back to NAD+, as the glycolysis stops after the initial quantity of 6 NAD+ is consumed. Also, the $gly1$ transition does not fire after time-step 2, indicating glycolysis has stopped. Trend of various runs is shown in Figure 3.19.

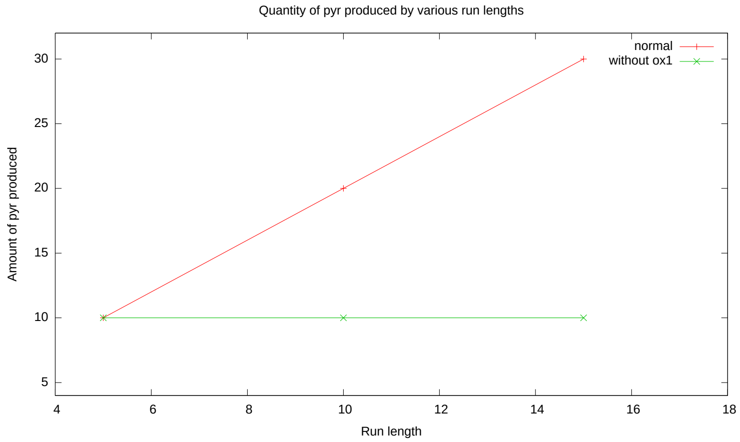


Figure 3.19: Amount of "pyr" produced by runs of various lengths of Petri Net in Figure 3.18. It shows results for both normal situation where "nadh" is recycled to "nadp" as well as the abnormal situation where this recycling is stopped.

3.11 Comparing Altered Trajectories due to Initial Value Intervention

Question 10 *During intense exercise, can a muscle cell use fat as a concentrated source of chemical energy? Explain.*

Provided Answer:

> "When oxygen is present, the fatty acid chains containing most of the energy of a fat are oxidized and fed into the citric acid cycle and the electron transport chain. During intense exercise, however, oxygen is scarce in muscle cells, so ATP must be generated by glycolysis alone. A very small part of the fat molecule, the glycerol backbone, can be oxidized via glycolysis, but the amount of energy released by this portion is insignificant compared to that released by the fatty acid chains. (This is why moderate exercise, staying below 70% maximum heart rate, is better for burning fat because enough oxygen remains available to the muscles.)"

Solution 10 *The process of fat consumption in glycolysis and citric acid cycle is summarized in Fig 9.19 of Campbell's book. Fats are digested into glycerol and fatty acids. Glycerol gets fed into glycolysis after being converted into Gyceraldehyde 3-phosphate (G3P), while fatty acids get fed into citric acid cycle after being broken down through beta oxidation and converted into Acetyl CoA. Campbell's Fig 9.18 identify a junction in catabolism where aerobic respiration or fermentation take place depending upon whether oxygen is present or not. Energy produced at various steps is in terms of ATP produced. In order to explain whether fat can be used as a concentrated source of chemical energy or not, we have to show the different ways of ATP production and when they kick in.*

We combine the various pieces of information collected from Fig 9.19, second paragraph on second column of p/180, Fig 9.15, Fig 9.16 and Fig 9.18 of Campbell's book into Figure 3.20. We model two situations when oxygen is not available in the muscle cells (at the start of a intense exercise) and when oxygen is available in the muscle cells (after the exercise intensity is plateaued). We then compare and contrast them on the amount of ATP produced and the reasons for the firing sequences.

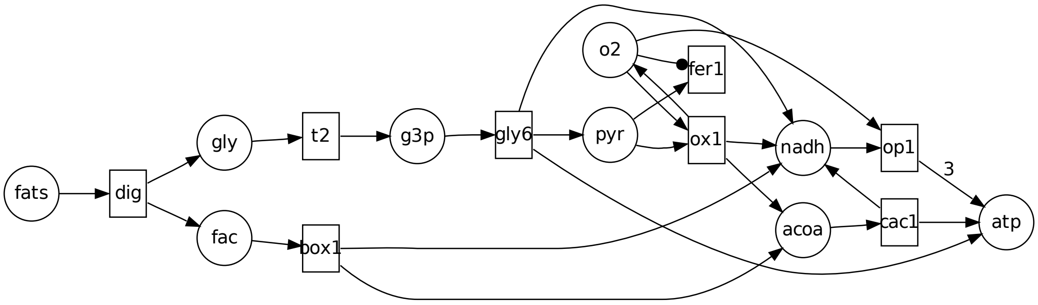


Figure 3.20: Petri Net graph relevant to question 10. "fats" are fats, "dig" is digestion of fats, "gly" is glycerol, "fac" is fatty acid, "g3p" is Glyceraldehyde 3-phosphate, "pyr" is pyruvate, "o2" is oxygen, "nadh" is NADH, "acoa" is Acyl CoA, "atp" is ATP, "op1" is oxidative phosphorylation, "cac1" is citric acid cycle, "fer1" is fermentation, "ox1" is oxidation of pyruvate to Acyl CoA and "box1" is beta oxidation.

Figure 3.20 is a petri net representation of our simplified model. Our edge labels have lower numbers on them than the yield in Fig 9.16 of Campbell's book but they still capture the difference in volume that would be produced due to oxidative phosphorylation vs. glycolysis. Using exact amounts will only increase the difference of ATP production due to the two mechanisms. We encode both situations (when oxygen is present and when it is not) in ASP with maximal firing set policy. We run them for 10 steps. The firing sequence and the resulting yield of ATP explain what the possible use of fat as a source of chemical energy.

At he start of intense exercise, when oxygen is in short supply:

```
holds(acoa,0,0) holds(atp,0,0) holds(fac,0,0) holds(fats,5,0)

holds(g3p,0,0) holds(gly,0,0) holds(nadh,0,0) holds(o2,0,0) holds(pyr,0,0)

fires(dig,0)

holds(acoa,0,1) holds(atp,0,1) holds(fac,1,1) holds(fats,4,1)

holds(g3p,0,1) holds(gly,1,1) holds(nadh,0,1) holds(o2,0,1) holds(pyr,0,1)

fires(box1,1) fires(dig,1) fires(t2,1)

holds(acoa,1,2) holds(atp,0,2) holds(fac,1,2) holds(fats,3,2)
holds(g3p,1,2) holds(gly,1,2) holds(nadh,1,2) holds(o2,0,2) holds(pyr,0,2)
fires(box1,2) fires(cac1,2) fires(dig,2) fires(gly6,2) fires(t2,2)
holds(acoa,1,3) holds(atp,2,3) holds(fac,1,3) holds(fats,2,3)
holds(g3p,1,3) holds(gly,1,3) holds(nadh,3,3) holds(o2,0,3) holds(pyr,1,3)
fires(box1,3) fires(cac1,3) fires(dig,3) fires(fer1,3) fires(gly6,3) fires(t2,3)
holds(acoa,1,4) holds(atp,4,4) holds(fac,1,4) holds(fats,1,4)
holds(g3p,1,4) holds(gly,1,4) holds(nadh,5,4) holds(o2,0,4) holds(pyr,1,4)
fires(box1,4) fires(cac1,4) fires(dig,4) fires(fer1,4) fires(gly6,4) fires(t2,4)
holds(acoa,1,5) holds(atp,6,5) holds(fac,1,5) holds(fats,0,5)
holds(g3p,1,5) holds(gly,1,5) holds(nadh,7,5) holds(o2,0,5) holds(pyr,1,5)
fires(box1,5) fires(cac1,5) fires(fer1,5) fires(gly6,5) fires(t2,5)
holds(acoa,1,6) holds(atp,8,6) holds(fac,0,6) holds(fats,0,6)
holds(g3p,1,6) holds(gly,0,6) holds(nadh,9,6) holds(o2,0,6) holds(pyr,1,6)
fires(cac1,6) fires(fer1,6) fires(gly6,6)
holds(acoa,0,7) holds(atp,10,7) holds(fac,0,7) holds(fats,0,7)
holds(g3p,0,7) holds(gly,0,7) holds(nadh,10,7) holds(o2,0,7) holds(pyr,1,7)
fires(fer1,7)
holds(acoa,0,8) holds(atp,10,8) holds(fac,0,8) holds(fats,0,8)
holds(g3p,0,8) holds(gly,0,8) holds(nadh,10,8) holds(o2,0,8) holds(pyr,0,8)
holds(acoa,0,9) holds(atp,10,9) holds(fac,0,9) holds(fats,0,9)
holds(g3p,0,9) holds(gly,0,9) holds(nadh,10,9) holds(o2,0,9) holds(pyr,0,9)
holds(acoa,0,10) holds(atp,10,10) holds(fac,0,10) holds(fats,0,10)
holds(g3p,0,10) holds(gly,0,10) holds(nadh,10,10) holds(o2,0,10) holds(pyr,0,10)
```

when the exercise intensity has plateaued and oxygen is no longer in short supply:

```
holds(acoa,0,0) holds(atp,0,0) holds(fac,0,0) holds(fats,5,0)
holds(g3p,0,0) holds(gly,0,0) holds(nadh,0,0) holds(o2,10,0) holds(pyr,0,0)
fires(dig,0)
holds(acoa,0,1) holds(atp,0,1) holds(fac,1,1) holds(fats,4,1)
holds(g3p,0,1) holds(gly,1,1) holds(nadh,0,1) holds(o2,10,1) holds(pyr,0,1)
fires(box1,1) fires(dig,1) fires(t2,1)
holds(acoa,1,2) holds(atp,0,2) holds(fac,1,2) holds(fats,3,2)
holds(g3p,1,2) holds(gly,1,2) holds(nadh,1,2) holds(o2,10,2) holds(pyr,0,2)
fires(box1,2) fires(cac1,2) fires(dig,2) fires(gly6,2) fires(op1,2) fires(t2,2)
holds(acoa,1,3) holds(atp,5,3) holds(fac,1,3) holds(fats,2,3)
holds(g3p,1,3) holds(gly,1,3) holds(nadh,2,3) holds(o2,9,3) holds(pyr,1,3)
fires(box1,3) fires(cac1,3) fires(dig,3) fires(gly6,3) fires(op1,3)
        fires(ox1,3) fires(t2,3)
holds(acoa,2,4) holds(atp,10,4) holds(fac,1,4) holds(fats,1,4)
holds(g3p,1,4) holds(gly,1,4) holds(nadh,4,4) holds(o2,8,4) holds(pyr,1,4)
fires(box1,4) fires(cac1,4) fires(dig,4) fires(gly6,4) fires(op1,4)
        fires(ox1,4) fires(t2,4)
holds(acoa,3,5) holds(atp,15,5) holds(fac,1,5) holds(fats,0,5)
holds(g3p,1,5) holds(gly,1,5) holds(nadh,6,5) holds(o2,7,5) holds(pyr,1,5)
fires(box1,5) fires(cac1,5) fires(gly6,5) fires(op1,5) fires(ox1,5) fires(t2,5)
holds(acoa,4,6) holds(atp,20,6) holds(fac,0,6) holds(fats,0,6)
holds(g3p,1,6) holds(gly,0,6) holds(nadh,8,6) holds(o2,6,6) holds(pyr,1,6)
fires(cac1,6) fires(gly6,6) fires(op1,6) fires(ox1,6)
holds(acoa,4,7) holds(atp,25,7) holds(fac,0,7) holds(fats,0,7)
holds(g3p,0,7) holds(gly,0,7) holds(nadh,9,7) holds(o2,5,7) holds(pyr,1,7)
fires(cac1,7) fires(op1,7) fires(ox1,7)
holds(acoa,4,8) holds(atp,29,8) holds(fac,0,8) holds(fats,0,8)
holds(g3p,0,8) holds(gly,0,8) holds(nadh,10,8) holds(o2,4,8) holds(pyr,0,8)
fires(cac1,8) fires(op1,8)
holds(acoa,3,9) holds(atp,33,9) holds(fac,0,9) holds(fats,0,9)
holds(g3p,0,9) holds(gly,0,9) holds(nadh,10,9) holds(o2,3,9) holds(pyr,0,9)
fires(cac1,9) fires(op1,9)
holds(acoa,2,10) holds(atp,37,10) holds(fac,0,10) holds(fats,0,10)
holds(g3p,0,10) holds(gly,0,10) holds(nadh,10,10) holds(o2,2,10) holds(pyr,0,10)
fires(cac1,10) fires(op1,10)
```

We see that more ATP is produced when oxygen is available. Most ATP (energy) is produced by the oxidative phosphorylation which requires oxygen. When oxygen is *not available, small amount of energy is produced due to glycolysis of glycerol (gly). With oxygen a lot more energy is produced, most of it due to fatty acids (fac). Trend of various runs is shown in Figure 3.21.*

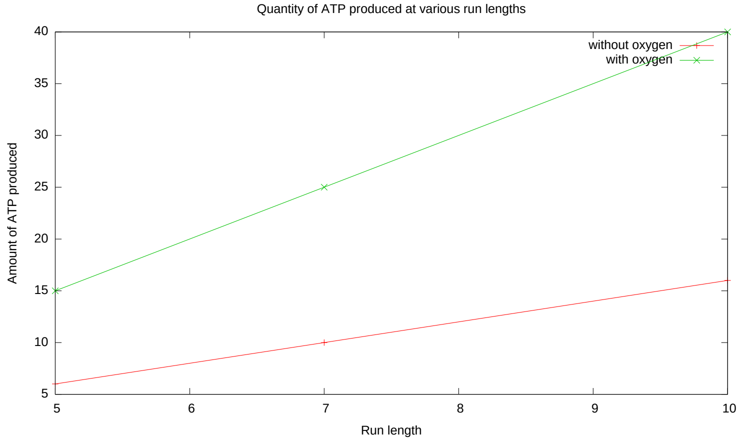


Figure 3.21: Amount of “atp” produced by runs of various lengths of Petri Net in Figure 3.20. Two situations are shown: when oxygen is in short supply and when it is abundant.

3.12 Conclusion

In this chapter we presented how to model biological systems as Petri Nets, translated them into ASP, reasoned with them and answered questions about them. We used diagrams from Campbell's book, background knowledge and assumptions to facilitate our modeling work. However, source knowledge for real world applications comes from published papers, magazines and books. This means that we have to do text extraction. In one of the following chapters we look at some of the real applications that we have worked on in the past in collaboration with other researchers to develop models using text extraction. But first, we look at how we use the concept of answering questions using Petri Nets to build a question answering system. We will extend the Petri Nets even more for this.

Chapter 4

BIOPATHQA - A SYSTEM FOR MODELING, SIMULATING, AND QUERYING BIOLOGICAL PATHWAYS

4.1 Introduction

In this chapter we combine the methods from Chapter 3, notions from action languages, and ASP to build a system *BioPathQA* and a language to specify pathways and query them. We show how various biological pathways are encoded in BioPathQA and how it computes answers of queries against them.

4.2 Description of BioPathQA

Our system has the following components: (i) a pathway specification language (ii) a query language to specify the deep reasoning question, (iii) an ASP program that encodes the pathway model and its extensions for simulation.

Knowledge about biological pathways comes in many different forms, such as cartoon diagrams, maps with well defined syntax and semantics (e.g. Kohn's maps (Kohn *et al.*, 2006)), and biological publications. Similar to other technical domains, some amount of domain knowledge is also required. Users want to collect information from disparate sources and encode it in a pathway specification. We have developed a language to allow users to describe their pathway. This description includes describing the substances and actions that make up the pathway, the initial state of the substances, and how the state of the pathway changes due to the actions. An evolution of a pathway's state from the initial state, through a set of actions is called a trajectory. Being a specification language targeted at biological systems, multiple actions autonomously execute in parallel as soon as their preconditions are satisfied. The amount of parallelism is dictated by any resource conflicts between the actions. When that occurs, only one sub-set of the possible actions can execute, leading to multiple outcomes from that point on.

Questions are usually provided in natural language, which is vague. To avoid the vagaries of natural language, we developed a language with syntax close to natural language but with a well defined formal semantics. The query language allows a user to make changes to the pathway through interventions, and restrict its trajectories through observations and query on aggregate values in a trajectory, across a set of trajectories and even over two sets of trajectories. This allows the user to compare a base case of a pathway specification with an alternate case modified due to interventions and observations. This new feature is a major contribution of our research.

Inspiration for our high level language comes from action languages and query languages such as Gelfond and Lifschitz (1993). While action languages generally describe transition systems (Gelfond and Lifschitz, 1998), our language describes trajectories. In addition, our language is geared towards modeling natural systems, in which actions occur autonomously (Reiter, 1996) when their pre-conditions are satisfied; and we do not allow the quantities to become negative (as the quantities represent amounts of physical entities).

Next we describe the syntax of our pathway specification language and the query language. Following that we will describe the syntax of our language and how we encode it in ASP.

4.3 Syntax of Pathway Specification Language (BioPathQA-PL)

The alphabet of pathway specification language $\mathcal{P}$ consists of disjoint nonempty domain-dependent sets A, F, L representing actions, fluents, and locations, respectively; a fixed set S of firing styles; a fixed set K of keywords providing syntactic sugar (shown in bold face in pathway specification language below); a fixed set of punctuations $\{`,'\}$; and a fixed set of special constants $\{`1', `*', `max'\}$; and integers.

Each fluent $f \in F$ has a domain $dom(f)$ which is either integer or binary and specifies the values f can take. A fluent is either simple, such as f or locational, such as $f[l]$, where $l \in L$. A *state* s is an interpretation of F that maps fluents to their values. We write $s(f) = v$ to represent "f has the value v in state s". States are indexed, such that consecutive states s_i and s_{i+1} represent an evolution over one time step from i to $i+1$ due to firing of an action set T_i in s_i.

We illustrate the role of various symbols in the alphabet with examples from the biological domain. Consider the following example of a hypothetical pathway specification:

$$\textbf{domain of } sug \textbf{ is } integer, fac \textbf{ is } integer, acoa \textbf{ is } integer, h2o \textbf{ is } integer \tag{4.1}$$

$$gly \textbf{ may execute causing } sug \textbf{ change value by } -1, acoa \textbf{ change value by } +1 \tag{4.2}$$

$$box \textbf{ may execute causing } fac \textbf{ change value by } -1, acoa \textbf{ change value by } +1 \tag{4.3}$$

$$\qquad \textbf{if } h2o \textbf{ has value } 1 \textbf{ or higher} \tag{4.4}$$

$$\textbf{inhibit } box \textbf{ if } sug \textbf{ has value } 1 \textbf{ or higher} \tag{4.5}$$

$$\textbf{initially } sug \textbf{ has value } 3, fac \textbf{ has value } 4, acoa \textbf{ has value } 0 \tag{4.6}$$

It describes two processes glycolysis and beta-oxidation represented by actions '*gly*' and '*box*' in lines (4.2) and (4.3)-(4.4) respectively. Substances used by the pathway, i.e. sugar, fatty-acids, acetyl-CoA, and water are represented by numeric fluents '*sug*','*fac*','*acoa*', and '*h2o*' respectively in line (4.1). When glycolysis occurs, it consumes 1 unit of sugar and produces 1 unit of acetyl-CoA (line (4.2). When beta-oxidation occurs, it consumes 1 unit of fatty-acids and produces 1 unit of acetyl-CoA (line (4.3)). The inputs of glycolysis implicitly impose a requirement that glycolysis can only occur when at least 1 unit of sugar is available. Similarly, the input of beta-oxidation implicitly a requirement that beta-oxidation can only occur when at least 1 unit of fatty-acids is available. Beta oxidation has an additional condition imposed on it in line (4.4) that there must be at least 1 unit of water available. We call this a *guard* condition on beta-oxidation. Line (4.5) explictly inhibits beta-oxidation when there is any sugar available; and line (4.6) sets up the initial conditions of the pathway, i.e. Initially 3 units of each sugar, 4 units of fatty-acids are available and no acetyl-CoA is available. The words '{*domain*, *is*, *may*, *execute*, *causing*, *change*, *value*, *by*, *has*, *or*, *higher*, *inhibit*, *if*, *initially*}' are keywords.

When locations are involved, locational fluents take place of simple fluents and our representation changes to include locations. For example:

$$\begin{array}{l} gly \textbf{ may execute causing} \\ \quad sug \textbf{ atloc } mm \textbf{ change value by } -1, \\ \quad acoa \textbf{ atloc } mm \textbf{ change value by } +1 \end{array}$$

represents glycolysis taking 1 unit of sugar from mitochondrial matrix (represented by '*mm*') and produces acetyl-CoA in the mitochondrial matrix. Here '*atloc*' is an additional keyword.

A pathway is composed of a collection of different types of statements and clauses. We first introduce their syntax, following that we will give intuitive definitions, and following that we will show how they are combined together to construct a pathway specification.

Definition 46 (Fluent domain declaration statement) *A fluent domain declaration statement declares the values a fluent can take. It has the form:*

$$\textbf{domain of } f \textbf{ is } `integer'|`binary' \tag{4.7}$$

$$\textbf{domain of } f \textbf{ atloc } l \textbf{ is } `integer'|`binary' \tag{4.8}$$

for simple fluent "f", and locational fluent "f[l]". Multiple domain statements are compactly written as:

$$\textbf{domain of } f_1 \textbf{ is } `\mathit{integer}'|`\mathit{binary}', \dots, f_n \textbf{ is } `\mathit{integer}'|`\mathit{binary}' \tag{4.9}$$

$$\textbf{domain of } f_1 \textbf{ atloc } l_1 \textbf{ is } `\mathit{integer}'|`\mathit{binary}', \dots, f_n \textbf{ atloc } l_n \textbf{ is } `\mathit{integer}'|`\mathit{binary}' \tag{4.10}$$

Binary domain is usually used for representing substances in a signaling pathway, while a metabolic pathways take positive numeric values. Since the domain is for a physical entity, we do not allow negative values for fluents.

Definition 47 (Guard condition) *A* guard condition *takes one of the following forms:*

$$f \textbf{ has value } w \textbf{ or higher} \tag{4.11}$$

$$f \textbf{ atloc } l \textbf{ has value } w \textbf{ or higher} \tag{4.12}$$

$$f \textbf{ has value lower than } w \tag{4.13}$$

$$f \textbf{ atloc } l \textbf{ has value lower than } w \tag{4.14}$$

$$f \textbf{ has value equal to } w \tag{4.15}$$

$$f \textbf{ atloc } l \textbf{ has value equal to } w \tag{4.16}$$

$$f_1 \textbf{ has value higher than } f_2 \tag{4.17}$$

$$f_1 \textbf{ atloc } l_1 \textbf{ has value higher than } f_2 \textbf{ atloc } l_2 \tag{4.18}$$

where, each f in (4.11), (4.13), (4.15), (4.17) *is a simple fluent, each $f[l]$ in* (4.12), (4.14), (4.16), (4.18) *is a locational fluent with location l, and each $w \in \mathbb{N}^+ \cup \{0\}$.*

Definition 48 (Effect clause) *An* effect *clause can take one of the following forms:*

$$f \textbf{ change value by } e \tag{4.19}$$

$$f \textbf{ atloc } l \textbf{ change value by } e \tag{4.20}$$

where a is an action, f in (4.19) *is a simple fluent, $f[l]$ in* (4.20) *is a locational fluent with location l, $e \in \mathbb{N}^+ \cup \{*\}$ for integer fluents or $e \in \{1, -1, *\}$ for binary fluents.*

Definition 49 (May-execute statement) *A* may-execute *statement captures the conditions for firing an action a and its impact. It is of the form:*

$$\begin{aligned} a \textbf{ may execute causing } & \mathit{effect}_1, \ldots, \mathit{effect}_m \\ & \textbf{if } \mathit{guard\_cond}_1, \ldots, \mathit{guard\_cond}_n \end{aligned} \tag{4.21}$$

where effect_i is an effect *clause; and $\mathit{guard\_cond}_j$ is a guard condition clause, $m > 0$, and $n \geq 0$. If $n = 0$, the effect statement is unconditional (guarded by $\top$) and the* **if** *is dropped. A single* may-execute *statement must not have $\mathit{effect}_i, \mathit{effect}_j$ with $e_i < 0, e_j < 0$ for the same fluent; or $e_i > 0, e_j > 0$ for the same fluent.*

Definition 50 (Must-execute statement) *An* must-execute *statement captures the impact of firing of an action a that must fire when enabled (as long as it is not inhibited). It is an expression of the form:*

$$\begin{aligned} a \textbf{ normally must execute causing } & \mathit{effect}_1, \ldots, \mathit{effect}_m \\ & \textbf{if } \mathit{guard\_cond}_1, \ldots, \mathit{guard\_cond}_n \end{aligned} \tag{4.22}$$

where effect_i, and $\mathit{guard\_cond}_j$ are as in (4.21) *above.*

Definition 51 (Inhibit statement) *An* inhibit *statement captures the conditions that inhibit an action from occurring. It is an expression of the form:*

$$\textbf{inhibit } a \textbf{ if } \mathit{guard\_cond}_1, \ldots, \mathit{guard\_cond}_n \tag{4.23}$$

*where a is an action, $\mathit{guard\_cond}_i$ is a guard condition clause, and $n \geq 0$. if $n = 0$, the inhibition of action 'a' is unconditional '***if***' is dropped.*

Definition 52 (Initial condition statement) *An* initial condition *statement captures the initial state of pathway. It is of the form:*

$$\textbf{initially } f \textbf{ has value } w \tag{4.24}$$

$$\textbf{initially } f \textbf{ atloc } l \textbf{ has value } w \tag{4.25}$$

where each f in (4.24) *is a simple fluent,* $f[l]$ *in* (4.25) *is a locational fluent with location l, and each w is a non-negative integer. Multiple* initial condition *statements are written compactly as:*

$$\textbf{initially } f_1 \textbf{ has value } w_1, \ldots, f_n \textbf{ has value } w_n$$

$$\textbf{initially } f_1 \textbf{ atloc } l_1 \textbf{ has value } w_1, \ldots, f_n \textbf{ atloc } l_n \textbf{ has value } w_n$$

Definition 53 (Duration Statement) *A* duration *statement represents the duration of an action that takes longer than a single time unit to execute. It is of the form:*

$$a \textbf{ executes in } d \textbf{ time units} \tag{4.26}$$

where d is a positive integer representing the action duration.

Definition 54 (Stimulate Statement) *A* stimulate *statement changes the rate of an action. It is an expression of the form:*

$$\textbf{normally stimulate } a \textbf{ by factor } n \textbf{ if } guard\_cond_1, \ldots, guard\_cond_n \tag{4.27}$$

where $guard\_cond_i$ is a condition, $n > 0$. *When $n = 0$, the stimulation is unconditional and* **if** *is dropped. A stimulation causes the effect in* may-cause *and* must-fire *multiplied by n.*

Actions execute automatically when fireable, subject to the available fluent quantities.

Definition 55 (Firing Style Statement) *A* firing style *statement specifies how many actions execute simultaneously (or action parallelism). It is of the form:*

$$\textbf{firing style } S \tag{4.28}$$

where, S is either "1", "$*$", or "max" for serial execution, interleaved execution, and maximum parallelism.

We will now give the intuitive meaning of these statements and put them into context w.r.t. the biological domain. Though our description below uses simple fluents only, it applies to locational fluents in a obvious manner. The reason for having locational fluents at all is that they allow a more natural pathway specification when substance locations are involved instead of devising one's own encoding scheme. For example, in a mitochondria, hydrogen ions (H+) appear in multiple locations (intermembrane space and mitochondrial matrix), with each location carrying its distinct quantity separate from other locations.

Intuitively, a *may-execute statement* (4.21) represents an action a that may fire if all conditions '$guard\_cond_1, \dots, guard\_cond_n$' hold in the current state. When it executes, it impacts the state as specified in *effect*s. In biological context, action a represents a process, such as a reaction, *effect*s represent the inputs / ingredients of the reaction, and *guard_cond* represent additional preconditions necessary for the reaction to proceed. Condition (4.11) holds in a state s if $s(f) \geq w$. It could represent an initiation concentration w of a substance f which is higher than the quantity consumed by the reaction a. Condition (4.13) holds in a state s if $s(f) < w$. Condition (4.15) holds in a state s if $s(f) = w$. Condition (4.17) holds in a state s if $s(f_1) > s(f_2)$ capturing a situation where a substance gradient is required for a biological process to occur. An example of one such process is the synthesis of ATP by ATP Synthase, which requires a H+ (Hydrogen ion) gradient across the inner mitochondrial matrix (Reece *et al.*, 2010, Figure 9.15).

Intuitively, the *effect clause* (4.19) of an action describes the impact of an action on a fluent. When an action a fires in a state s, the value of f changes according to the effect clause for f. The value of f increases by e if $e > 0$, decreases by e if $e < 0$, or decreases by $s(f)$ if $e =$ '$*'$ (where '$*'$ can be interpreted as $-s(f)$). For a reaction a, a fluent with $e < 0$ represents an ingredient consumed in quantity $|e|$

by the reaction; a fluent with $e > 0$ represents a product of the reaction in quantity e; a fluent with $e =$ '$*'$ represents consuming all quantity of the substance due to the reaction. Since the fluents represent physical substances, their quantities cannot become negative. As a result, any action that may cause a fluent quantity to go below zero is disallowed.

Intuitively, a *must-execute statement* (4.22) is similar to a *may-exec*, except that when enabled, it preferentially fires over other actions as long as there isn't an *inhibit* proposition that will cause the action to become inactive. It captures the effect of an action that must happen whenever enabled.

Intuitively, an *inhibit statement* (4.23) specifies the conditions that inhibits an action. In a biological context, it defines inhibition of reactions, e.g., through biological feedback control. Though we could have added these conditions to *may-exec*, it is more intuitive to keep them separate as inhibition conditions are usually discovered separately in a biological domain. Including them as part of *may-fire* would constitute a surgery of existing knowledge bases.

Intuitively, an *initial condition statement* (4.24) specifies the initial values of fluents. The collection of such propositions defines the initial state s_0 of the pathway. In a biological context, this defines the initial distribution of substances in the biological system.

Intuitively, an action *duration statement* (4.26) represents action durations, especially when an action takes longer to execute. When an action a with duration d fires in state s_k, it immediately decreases the values of fluents with $e < 0$ and $e = *$ upon execution, however, it does not increase the value of fluents with $e > 0$ until time the end of its execution in state s_{k+d}. In a biological context the action duration captures a reaction's duration. A reaction consumes its ingredients immediately on firing, processes them for duration d and generates its products at the end of this duration.

Intuitively, a *stimulate statement* (4.27) represents a change in the rate of an action a. The stimulation causes the action to change its rate of consumption of its ingredients and production of its products by a factor n. In biological context, this stimulation can be a result of an enzyme or a stimulant's availability, causing a reaction that normally proceeds slowly to occur faster.

Intuitively, a *firing style statement* (4.28) specifies the parallelism of actions. When it is "1", at most one action may fire, when it is "*max*", the maximum allowable actions must fire, and when it is "$*$", any subset of fireable actions may fire simultaneously. In a biological domain the firing style allows one to model serial operations, parallel operations and maximally parallel operations. The maximum parallelism is also useful in quickly discovering changes that occur in a biological system.

Definition 56 (Pathway Specification) *A* pathway specification *is composed of one or more* may-execute*,* must-execute*,* effect*,* inhibit*,* stimulate*,* initially*,* priority*,* duration *statements, and one* firing style *statement. When a* duration *statement is not specified for an action, it is assumed to be* 1*. Any fluents for which an initial quantity is not specified are assumed to have a value of zero.*

A pathway specification is consistent *if (i) there is at most one firing style, priority, duration statement for each action a; (ii) the $guard\_cond_1, \dots, guard\_cond_n$ from a* may-execute *or* must-execute *are disjoint from any other* may-execute *or* must-execute [1]*; (iii) locational and non-locational fluents may not be intermixed; (iv) domain of fluents, effects, conditions and numeric values are consistent, i.e., effects*

[1] Note that '$f1$ has value 5 or higher ' overlaps with '$f1$ has value 7 or higher' and the two conditions are not considered disjoint.

and conditions on binary fluents must be binary; and (v) the pathway specification does not cause it to violate fluent domains by producing non-binary values for binary fluents.

Each pathway specification **D** represents a collection of trajectories of the form: $\sigma = s_0, T_0, s_1, \dots, s_{k-1}, T_{k-1}, s_k$. Each trajectory encodes an evolution of the pathway starting from an initial state s_0, where s_i's are states, and T_i's are sets of actions that fired in state s_i.

Intuitively, a trajectory starts from the *initial state* s_0. Each s_i, s_{i+1} pair represents the state evolution in one time step due to the action set T_i. An action set T_i is only executable in state s_i, if the sum of changes to fluents due to $e_i < 0$ and $e_i = *$ will not result in any of the fluents going negative. Changes to fluents due to $e_i > 0$ for the action set T_i occur over subsequent time-steps depending upon the durations of actions involved. Thus, the state $s_i(f_i)$ is the sum of $e_i > 0$ for actions of duration d that occurred d time steps before (current time step) i, i.e. $a \in T_{i-d}$, where the default duration d of an action is 1 if none specified.

Next we describe the semantics of the pathway specification language, which describes how these trajectories are generated.

4.4 Semantics of Pathway Specification Language (BioPathQA-PL)

The semantics of the pathway specification language are defined in terms of the trajectories of the domain description **D**. Since our pathway specification language is inspired by Petri Nets, we use Petri Nets execution semantics to define its trajectories. However, some constructs in our pathway language specification are not directly representable in standard Petri Nets, as a result, we will have to extend them.

Let an *arc-guard* be a conjunction of *guard conditions* of the form (4.11)-(4.18), such that it is wholly constructed of either locational or non-locational fluents, but not both.

We introduce a new type of Guarded-arc Petri Net in which each arc has an *arc-guard* expression associated with it. Arcs with the same arc-guard are traversed when a transition connected to them fires and the arc-guard is found to hold in the current state. The arc-guards of arcs connected to the same transition form an exclusive set, such that only arcs corresponding to one guard expression may fire (for one transition). This setup can lead to different outcomes of an action[2].

The transitions in this new type of Petri Net can have the following inscriptions on them:

1. Propositional formula, specifying the executability conditions of the transition.

2. Arc-durations, represented as "$dur(n)$", where $n \in \mathbb{N}^+$

3. A *must-execute* inscription, "$must\text{-}execute(guard)$", requires that when the *guard* holds in a state where this transition is enabled, it must fire, unless explicitly inhibited. The *guard* has the same form as an *arc-guard*

4. A *stimulation* inscription, "$stimulate(n, guard)$", applies a multiplication factor $n \in \mathbb{N}^+$ to the input and output quantities consumed and produced by the transition, when *guard* hold in the current state, where *guard* has the same form as an *arc-guard*.

Certain aspects of our nets are similar to CPNs (Jensen *et al.*, 2007). However, the CPNs do not allow our semantics of the reset arcs, or must-fire guards.

[2] Arcs for different guard expressions emanating / terminating at a place can further be combined into a single conditional arc with conditional arc-weights. If none of the condition applies then the arc is assumed to be missing.

4.4.1 Guarded-Arc Petri Net

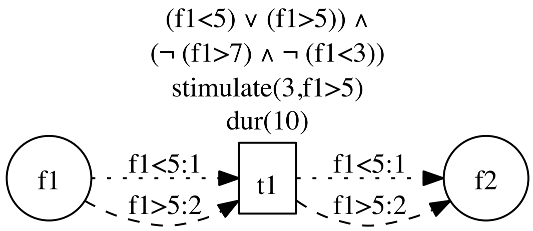


Figure 4.1: Example of a guarded-arc Petri Net.

Figure 4.1 shows an example of a guarded-arc Petri Net. There are two arc-guard expressions $f1 < 5$ and $f1 > 5$. When $f1 < 5$, $t1$ consumes one token from place $f1$ and produces one token in place $f2$. When $f1 > 5$, $t1$'s consumption and production of the same tokens doubles. The transition $t1$ implicitly gets the guards for each arc represented the *or*-ed conditions $(f1 < 5) \vee (f1 > 5))$. The arc also has two conditions inhibiting it, they are represented by the *and*-ed conditions $\neg(f1 > 7) \wedge \neg(f1 < 3)$, where '¬' represents logical not. Transition $t1$ is stimulated by factor 3 when $f1 > 5$ and it has a duration of 10 time units. A transition cannot fire even though one of its arc-guards is enabled, unless the token requirements on the arc itself are also fulfilled, e.g. if $f1$ has value 0 in the current state, even though $f1 < 5$ guard is satisfied, the transition cannot execute, because the input arc $(f1, t1)$ for this guard needs 1 token.

Definition 57 (Guard) *A* condition *is of the form:* $(f < v), (f \leq v), (f > v), (f \geq v), (f = v)$*, where f is a fluent and v either a fluent or a numeric constant. Then, a guard is a propositional formula of conditions, with each condition treated as a proposition, subject to the restriction that all fluents in all conditions in a guard are either locational or simple, but not both.*

Definition 58 (Interpretation of a Guard) *An interpretation of a guard G is a possible assignment of a value to each fleuent $f \in G$ from the domain of f.*

Definition 59 (Guard Satisfaction) *A guard G with simple fluents is satisfied w.r.t. a state s, written $s \models G$ iff G has an interpretation in which each of its fluents f has the value $s(f)$ and G is true. A guard G with locational fluents is satisfied w.r.r. a state s, written $s \models G$ iff G has an interpretation in which each of its fluents $f[l]$ has the value $m_{s(l)}(f)$ and G is true, where $m_X(f)$ is the multiplicity of f in X.*

Definition 60 (Guarded-Arc Petri Net) *A Guarded-Arc Petri Net is a tuple $PN^G = (P, T, G, E, R, W, D, B, TG, MF, L)$, where:*

$$\begin{aligned}
P &\text{ is a finite set of places}\\
T &\text{ is a finite set of transitions}\\
G &\text{ is a set of guards as defined in definition (57)}\\
TG : T \to G &\text{ are the transition guards}\\
E \subseteq (T \times P \times G) \cup (P \times T \times G) &\text{ are the guarded arcs}\\
R \subseteq P \times T \times G &\text{ are the guarded reset arcs}\\
W : E \to \mathbb{N}^+ &\text{ are arc weights}\\
D : T \to \mathbb{N}^+ &\text{ are the transition durations}\\
B : T \to G \times \mathbb{N}^+ &\text{ transition stimulation or boost}\\
MF : T \to 2^G &\text{ must-fire guards for a transition}\\
L : P \to \mathbb{N}^+ &\text{ specifies maximum number to tokens for each place}
\end{aligned}$$

subject to constraints:

1. $P \cap T = \emptyset$

2. $R \cap E = \emptyset$

3. *Let $t_1 \in T$ and $t_2 \in T$ be any two distinct transitions, then $g1 \in MF(t_1)$ and $g2 \in MF(t_2)$ must not have an interpretation that make both $g1$ and $g2$ true.*

4. *Let $t \in T$ be a transition, and $gg_t = \{g : (t, p, g) \in E\} \cup \{g : (p, t, g) \in E\} \cup \{g : (p, t, g) \in R\}$ be the set of arc-guards for normal and reset arcs connected to it, then $g_1 \in gg_t, g_2 \in gg_t$ must not have an interpretation that makes both g_1 and g_2 true.*

5. *Let $t \in T$ be a transition, and $gg_t = \{g : (g, n) \in B(t)\}$ be its stimulation arc-guards, then $g_1 \in gg_t, g_2 \in gg_t$ must not have an interpretation that makes both g_1 and g_2 true.*

6. *Let $t \in T$ be a transition, and $(g, n) \in B(t) : n > 1$, then there must not exist a place $p \in P : (p, t, g) \in E, L(p) = 1$. Intuitively, stimulation of binary inputs is not supported.*

We will make a simplifying assumption that all places are readable by using their place names. Execution of the PN^G occurs in discrete time steps.

Definition 61 (Marking (or State)) *Marking (or State) of a Guarded-Arc Petri Net PN^G is the token assignment of each place $p_i \in P$. Initial marking $M_0 : P \to \mathbb{N}^0$, while the token assignment at step k is written as M_k.*

Next we define the execution semantics of PN^G. First we introduce some terminology that will be used below. Let

1. $s_0 = M_0$ represent the the initial marking (or state), $s_k = M_k$ represent the marking (or state) at time step k,

2. $s_k(p)$ represent the marking of place p at time step k, such that $s_k = [s_k(p_0), \ldots, s_k(p_n)]$, where $P = \{p_0, \ldots, p_n\}$

3. T_k be the firing-set that fired in step k,
4. $b_k(t)$ be the stimulation value applied to a transition t w.r.t. step k
5. en_k be the set of enabled transitions in state s_k,
6. mf_k be the set of must-execute transitions in state s_k,
7. $consume_k(p, \{t_1, \ldots, t_n\})$ be the sum of tokens that will be consumed from place p if transitions $t_1, \ldots, t_n$ fired in state s_k,
8. $overc_k(\{t_1, \ldots, t_n\})$ be the set of places that will have over-consumption of tokens if transitions $t_1, \ldots, t_n$ were to fire simultaneously in state s_k,
9. $sel_k(fs)$ be the set of possible firing-set choices in state s_k using *fs* firing style
10. $produce_k(p)$ be the total production of tokens in place p (in state s_k) due to actions terminating in state s_k,
11. s_{k+1} be the next state computed from current state s_k due to firing of transition-set T_k
12. $min(a, b)$ gives the minimum of numbers a, b, such that $min(a, b) = a$ if $a < b$ or b otherwise .

Then, the execution semantics of the guarded-arc Petri Net starting from state s_0 using firing-style *fs* is given as follows:

$$b_k(t) = \begin{cases} n & \text{if } (g, n) \in B(t), s_k \models g \\ 1 & \text{otherwise} \end{cases}$$

$$\begin{aligned} en_k = \{t : t \in T, s_k \models TG(t), \forall (p, t, g) \in E, \\ (s_k \models g, s_k(p) \geq W(p, t, g) * b_k(t))\} \end{aligned}$$

$$
\begin{aligned}
mf_k &= \{t : t \in en_k, \exists g \in MF(t), s_k \models g\} \\
consume_k(p, \{t_1, \dots, t_n\}) &= \sum_{i=1,\dots,n} W(p, t_i, g) * b_k(t) : (p, t_i, g) \in E, s_k \models g \\
&+ \sum_{i=1,\dots,n} s_k(p) : (p, t_i, g) \in R, s_k \models g \\
overc_k(\{t_1, \dots, t_n\}) &= \{p : p \in P, s_k(p) < consume_k(p, \{t_1, \dots, t_n\})\} \\
sel_k(1) &= \begin{cases} mf_k & \text{if } |mf_k| = 1 \\ \{\{t\} : t \in en_k\} & \text{if } |mf_k| < 1 \end{cases} \\
sel_k(*) &= \{ss : ss \in 2^{en_k}, mf_k \subseteq ss, overc_k(ss) = \emptyset\} \\
sel_k(max) &= \{ss : ss \in 2^{en_k}, mf_k \subseteq ss, overc_k(p, ss) = \emptyset, \\
&\quad (\nexists ss' \in 2^{en_k} : ss \subset ss', mf_k \subseteq ss', overc_k(ss') = \emptyset)\} \\
T_k &= T'_k : T'_k \in sel_k(fs), (\nexists t \in en_k \setminus T'_k, t \text{ is a reset transition })\\
produce_k(p) &= \sum_{j=0,\dots,k} W(t_i, p, g) * b_j(t_i) : (t_i, p, g) \in E, t_i \in T_j, D(t_i) + j = k + 1 \\
s_{k+1}(p) &= min(s_k(p) - consume_k(p, T_k) + produce_k(p), L(p)) \qquad (4.29)
\end{aligned}
$$

Definition 62 (Trajectory) *$\sigma = s_0, T_0, s_1, \dots, s_k, T_k, s_{k+1}$ is a trajectory of PN^G iff given $s_0 = M_0$, each T_i is a possible firing-set in s_i whose firing produces s_{i+1} per PN^G's execution semantics in* (4.29)*.*

4.4.2 Construction of Guarded-Arc Petri Net from a Pathway Specification

Now we describe how to construct such a Petri Net from a given pathway specification **D** with locational fluents. We proceed as follows:

1. The set of transitions $T = \{a : a \text{ is an action in } \mathbf{D}\}$.

2. The set of places $P = \{f : f \text{ is a fluent in } \mathbf{D}\}$.

3. The limit relation for places $L(f) = \begin{cases} 1 & \text{if '}\textbf{domain of } f \textbf{ is } \textit{binary}' \in \mathbf{D} \\ \infty & \text{otherwise} \end{cases}$.

4. An arc-guard expression $guard\_cond_1, \ldots, guard\_cond_n$ is translated into the conjunction $(guard_1, \ldots, guard_n)$, where $guard_i$ is obtained from $guard\_cond_i$ as follows:

   (a) A *guard condition* (4.11) is translated to $f \geq w$

   (b) A *guard condition* (4.13) is translated to $f < w$

   (c) A *guard condition* (4.15) is translated to $f = w$

   (d) A *guard condition* (4.17) is translated to $f_1 > f_2$

5. A *may-execute statement* (4.21) is translated into guarded arcs as follows:

   (a) Let guard G be the translation of arc-guard conditions $guard\_cond_1, \ldots,$ $guard\_cond_n$ specified in the *may-execute* proposition.

   (b) The *effect clause* (4.19) are translated into arcs as follows:

      i. An effect clause with $e < 0$ is translated into an input arc (f, a, G), with arc-weight $W(f, a, G) = |e|$.

      ii. An effect clause with $e =$ '$*$' is translated into a reset set (f, a, G) with arc-weight $W(f, a, G) = *$.

      iii. An effect clause with $e > 0$ is translated into an output arc (a, f, G), with arc-weight $W(a, f, G) = e$.

6. A *must-execute statement* (4.22) is translated into guarded arcs in the same way as *may execute*. In addition, it adds an arc-inscription *must-exec*(G), where G is the translation of the arc-guard.

7. An *inhibit statement* (4.23) is translated into $IG = (guard_1, \ldots, guard_n)$, where $(guard_1, \ldots, guard_n)$ is the translation of $(guard\_cond\_1, \ldots, guard\_cond_n)$

8. An *initial condition statement* (4.25) sets the initial marking of a specific place p to w, i.e. $M_0(p) = w$.

9. An *duration statement* (4.26) adds a $dur(d)$ inscription to transition a.

10. A *stimulate statement* (4.27) adds a $stimulate(n, G)$ inscription to transition a, where G is the translation of the stimulate guard, a conjunction of $guard\_cond_1$, $\ldots, guard\_cond_n$.

11. A guard $(G_1 \vee \cdots \vee G_n) \wedge (\neg IG_1 \wedge \cdots \wedge \neg IG_m)$ is added to each transition a, where $G_i, 1 \leq i \leq n$ is a guard for a *may-execute* or a *must-execute* statement and $IG_i, 1 \leq i \leq m$ is a guard for an *inhibit* statement.

12. A *firing style statement* (4.28) does not visibly appear on a Petri Net diagram, but it specifies the transition firing regime the Petri Net follows.

Example: Consider the following pathway specification:

$$
\begin{array}{lll}
\textbf{domain of} & f_1 \textbf{ is } \textit{integer}, & f_2 \textbf{ is } \textit{integer} \\
t_1 \textbf{ may execute} & & \\
\quad\textbf{causing} & f_1 \textbf{ change value by } -1, & f_2 \textbf{ change value by } +1 \\
\quad\textbf{if} & f_1 \textbf{ has value lower than } 5 & \\
t_1 \textbf{ may execute} & & \\
\quad\textbf{causing} & f_1 \textbf{ change value by } -2, & f_2 \textbf{ change value by } +2 \\
\quad\textbf{if} & f_1 \textbf{ has value higher than } 5 & \\
\textbf{duration of} & t1 \textbf{ is } 10 & \\
\textbf{inhibit } t1 \textbf{ if} & f_1 \textbf{ has value higher than } 7 & \\
\textbf{inhibit } t1 \textbf{ if} & f_1 \textbf{ has value lower than } 3 & \\
\textbf{normally stimulate } t1 & \textbf{ by factor } 3 & \\
\quad\textbf{if} & f_2 \textbf{ has value higher than } 5 & \\
\textbf{initially} & f_1 \textbf{ has value } 0, & f_2 \textbf{ has value } 0, \\
\textbf{firing style} & \textit{max} &
\end{array}
\tag{4.30}
$$

This pathway specification is encoded as the Petri Net in figure 4.1.

4.4.3 Guarded-Arc Petri Net with Colored Tokens

Next we extend the Guarded-arc Petri Nets to add Colored tokens. We will use this extension to model pathways with locational fluents.

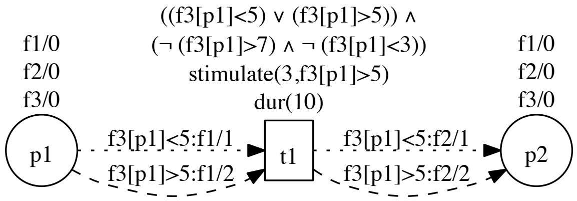


Figure 4.2: Example of a guarded-arc Petri Net with colored tokens.

Figure 4.2 shows an example of a guarded-arc Petri Net with colored tokens. There are two arc-guard expressions $f3[p1] < 5$ and $f3[p1] > 5$. When $f3[p1] < 5$, $t1$ consumes one token of color $f1$ from place $p1$ and produces one token of color $f2$ in place $p2$. When $f3[p1] > 5$, $t1$'s consumption and production of the same colored tokens doubles. The transition $t1$ implicitly gets the guards for each arc represented the *or*-ed conditions $((f3[p1] < 5) \vee (f3[p1] > 5))$. The arc also has two conditions inhibiting it, they are represented by the *and*-ed conditions $\neg(f3[p1] > 7) \wedge \neg(f3[p1] < 3)$, where '¬' represents logical not. Transition $t1$ is stimulated by factor 3 when $f3[p1] > 5$ and it has a duration of 10 time units.

Definition 63 (Guarded-Arc Petri Net with Colored Tokens) *A Guarded-Arc Petri Net with Colored Tokens is a tuple* $PN^{GC} = (P, T, C, G, E, R, W, D, B, TG, MF, L)$*, such that:*

P : *finite set of places*

T : *finite set of transitions*

C : *finite set of colors*

G : *set of guards as defined in definition* (57) *with locational fluents*

$TG : T \rightarrow G$ *are the transition guards*

$E \subseteq (T \times P \times G) \cup (P \times T \times G)$ *are the guarded arcs*

$R \subseteq P \times T \times G$ *are the guarded reset arcs*

$W : E \rightarrow \langle C, m \rangle$ *are arc weights; each arc weight is a multiset over* C

$D : T \rightarrow \mathbb{N}^+$ *are the transition durations*

$B : T \rightarrow G \times \mathbb{N}^+$ *transition stimulation or boost*

$MF : T \rightarrow 2^G$ *must-fire guards for a transition*

$L : P \times C \rightarrow \mathbb{N}^+$ *specifies maximum number of tokens for each color in each place* *subject to constraints:*

1. $P \cap T = \emptyset$

2. $R \cap E = \emptyset$

3. *Let* $t_1 \in T$ *and* $t_2 \in T$ *be any two distinct transitions, then* $g1 \in MF(t_1)$ *and* $g2 \in MF(t_2)$ *must not have an interpretation that make both* $g1$ *and* $g2$ *true.*

4. *Let* $t \in T$ *be a transition, and* $gg_t = \{g : (t, p, g) \in E\} \cup \{g : (p, t, g) \in E\} \cup \{g : (p, t, g) \in R\}$ *be the set of arc-guards for normal and reset arcs connected to it, then* $g_1 \in gg_t, g_2 \in gg_t$ *must not have an interpretation that makes both* g_1 *and* g_2 *true.*

5. *Let* $t \in T$ *be a transition, and* $gg_t = \{g : (g, n) \in B(t)\}$ *be its stimulation guards, then* $g_1 \in gg_t, g_2 \in gg_t$ *must not have an interpretation that makes both* g_1 *and* g_2 *true.*

6. *Let* $t \in T$ *be a transition, and* $(g, n) \in B(t) : n > 1$*, then there must not exist a place* $p \in P$ *and a color* $c \in C$ *such that* $(p, t, g) \in E, L(p, c) = 1$*. Intuitively, stimulation of binary inputs is not supported.*

Definition 64 (Marking (or State)) *Marking (or State) of a Guarded-Arc Petri Net with Colored Tokens* PN^{GC} *is the colored token assignment of each place* $p_i \in P$*. Initial marking is written as* $M_0 : P \to \langle C, m \rangle$*, while the token assignment at step* k *is written as* M_k*.*

We make a simplifying assumption that all places are readable by using their place name. Next we define the execution semantics of the guarded-arc Petri Net. First we introduce some terminology that will be used below. Let

1. $s_0 = M_0$ represent the the initial marking (or state), $s_k = M_k$ represent the marking (or state) at time step k,

2. $s_k(p)$ represent the marking of place p at time step k, such that $s_k = [s_k(p_0), \ldots, s_k(p_n)]$, where $P = \{p_0, \ldots, p_n\}$.

3. T_k be the firing-set that fired in state s_k,

4. $b_k(t)$ be the stimulation value applied to transition t w.r.t. state s_k,

5. en_k be the set of enabled transitions in state s_k,

6. mf_k be the set of must-fire transitions in state s_k,

7. $consume_k(p, \{t_1, \ldots, t_n\})$ be the sum of colored tokens that will be consumed from place p if transitions $t_1, \ldots, t_n$ fired in state s_k,

8. $overc_k(\{t_1, \ldots, t_n\})$ be the set of places that will have over-consumption of tokens if transitions $t_1, \ldots, t_n$ were to fire simultaneously in state s_k,

9. $sel_k(fs)$ be the set of possible firing-sets in state s_k using *fs* firing style,

10. $produce_k(p)$ be the total production of tokens in place p in state s_k due to actions terminating in state s_k,

11. s_{k+1} be the next state computed from current state s_k and T_k,

12. $m_X(c)$ represents the multiplicity of $c \in C$ in multiset $X = \langle C, m \rangle$,

13. c/n represents repetition of an element c of a multi-set n-times,

14. multiplication of multiset $X = \langle C, m \rangle$ with a number n be defined in terms of multiplication of element multiplicities by n, i.e. $\forall c \in C, m_X(c) * n$, and

15. $min(a, b)$ gives the minimum of numbers a, b, such that $min(a, b) = a$ if $a < b$ or b otherwise .

Then, the execution semantics of the guarded-arc Petri Net starting from state s_0 using firing-style fs is given as follows:

$$b_k(t) = \begin{cases} n & \text{if } (g, n) \in B(t), s_k \models g \\ 1 & \text{otherwise} \end{cases}$$

$$\begin{aligned} en_k &= \{t \in T, s_k \models TG(t), \forall (p, t, g) \in E, \\ &\qquad (s_k \models g, s_k(p) \geq W(p, t, g) * b_k(t))\} \end{aligned}$$

$$mf_k = \{t \in en_k, \exists g \in MF(t), s_k \models g\}$$

$$\begin{aligned} consume_k(p, \{t_1, \ldots, t_n\}) &= \sum_{i=1,\ldots,n} W(p, t_i, g) * b_k(t) : (p, t_i, g) \in E, s_k \models g \\ &+ \sum_{i=1,\ldots,n} s_k(p) : (p, t_i, g) \in R, s_k \models g \end{aligned}$$

$$overc_k(\{t_1, \ldots, t_n\}) = \{p \in P : \exists c \in C, m_{s_k(p)}(c) < m_{consume_k(p,\{t_1,\ldots,t_n\})}(c)\}$$

$$sel_k(1) = \begin{cases} mf_k & \text{if } |mf_k| = 1 \\ \{\{t\} : t \in en_k\} & \text{if } |mf_k| < 1 \end{cases}$$

$$sel_k(*) = \left\{ ss \in 2^{en_k} : mf_k \subseteq ss, overc_k(ss) = \emptyset \right.$$

$$sel_k(max) = \begin{cases} ss \in 2^{en_k} : mf_k \subseteq ss, overc_k(p, ss) = \emptyset, \\ (\nexists ss' \in 2^{en_k} : ss \subset ss', mf_k \subseteq ss', overc_k(ss') = \emptyset) \end{cases}$$

$$T_k = T'_k : T'_k \in sel_k(fs), (\nexists t \in en_k \setminus T'_k, t \text{ is a reset transition })$$

$$produce_k(p) = \sum_{j=0,\ldots,k} W(t_i, p, g) * b_j(t_i) : (t_i, p, g) \in E, t_i \in T_j), D(t_i) + j = k + 1$$

$$\begin{aligned} s_{k+1} &= [c/n : c \in C, \\ &\qquad n = min(m_{s_k(p)}(c) \end{aligned}$$

$$
\begin{aligned}
&- m_{consume_k(p,T_k)}(c) \\
&+ m_{produce_k(p)}(c), L(p,c))] \qquad (4.31)
\end{aligned}
$$

Definition 65 (Trajectory) *$\sigma = s_0, T_0, s_1, \ldots, s_k, T_k, s_{k+1}$ is a trajectory of PN^{GC} iff given $s_0 = M_0$, each T_i is a possible firing-set in s_i whose firing produces s_{i+1} per PN^{GC}'s execution semantics in* (4.31).

4.4.4 Construction of Guarded-Arc Petri Net with Colored Tokens from a Pathway Specification with Locational Fluents

Now we describe how to construct such a Petri Net from a given pathway specification **D** with locational fluents. We proceed as follows:

1. The set of transitions $T = \{a : a \text{ is an action in } \mathbf{D}\}$.
2. The set of colors $C = \{f : f[l] \text{ is a fluent in } \mathbf{D}\}$.
3. The set of places $P = \{l : f[l] \text{ is a fluent in } \mathbf{D}\}$.
4. The limit relation for each colored token in a place
$$L(f,c) = \begin{cases} 1 & \text{if `}\textbf{domain of } f \textbf{ atloc } l \textbf{ is } \textit{binary}\text{'} \in \mathbf{D} \\ \infty & \text{otherwise} \end{cases}.$$
5. A guard expression $guard\_cond_1, \ldots, guard\_cond_n$ is translated into the conjunction $(guard_1, \ldots,\ guard_n)$, where $guard_i$ is obtained from $guard\_cond_i$ as follows:
   (a) A *guard condition* (4.14) is translated to $f[l] < w$
   (b) A *guard condition* (4.16) is translated to $f[l] = w$

   (c) A *guard condition* (4.12) is translated to $f[l] \geq w$

   (d) A *guard condition* (4.18) is translated to $f_1[l_1] > f_2[l_2]$

6. A *may-execute statement* (4.21) is translated into guarded arcs as follows:

   (a) Let guard G be the translation of guard conditions $guard\_cond_1, \ldots,$ $guard\_cond_n$ specified in the *may-execute* proposition.

   (b) The *effect clauses* of the form (4.20) are grouped into input, reset and output effect sets for an action as follows:

      i. The clauses with $e < 0$ for the same place l are grouped together into an input set of a requiring input from place l.

      ii. The clauses with $e =$ '$*'$ for the same place l are grouped together into a reset set of a requiring input from place l.

      iii. The clauses with $e > 0$ for the same place l are grouped together into an output set of a to place l.

   (c) A group of input *effect* clauses $effect_1, \ldots, effect_m, m > 0$ of the form (4.20) is translated into an input arc (l, a, G), with arc-weight $W(l, a, G) = w^+$, where w^+ is the multi-set union of f_i/e_i in $effect_i, 1 \leq i \leq m$.

   (d) A group of output *effect* clauses $effect_1, \ldots, effect_m, m > 0$ of the form (4.20) is translated into an output arc (a, l, G), with arc-weight $W(a, l, G) = w^-$, where w^- is the multi-set union of $f_i/|e_i|$ in $effect_i, 1 \leq i \leq m$.

   (e) A group of reset *effect* clauses $effect_1, \ldots, effect_m, m > 0$ of the form (4.20) is translated into a reset arc (l, a, G) with arc-weight $W(l, a, G) = *$.

7. A *must-execute statement* (4.22) is translated into guarded arcs in the same way as *may execute*. In addition, it adds an arc-inscription *must-exec*(G), where G is the guard, which is the translation of $guard\_cond_1, \ldots, guard\_cond_n$.

8. An *inhibit statement* (4.23) is translated into $IG = (guard_1, \dots, guard_n)$, where $(guard_1, \dots, guard_n)$ is the translation of $(guard\_cond\_1, \dots, guard\_cond_n)$

9. An *initial condition statement* (4.25) sets the initial marking of a specific place l for a specific color f to w, i.e. $m_{(M_0(l))}(f) = w$.

10. An *duration statement* (4.26) adds a $dur(d)$ inscription to transition a.

11. A *stimulate statement* (4.27) adds a $stimulate(n, G)$ inscription to transition a, where guard G is the translation of its guard expression $guard\_cond_1, \dots,$ $guard\_cond_n$.

12. A guard $(G_1 \vee \dots \vee G_n) \wedge (\neg IG_1 \wedge \dots \wedge \neg IG_m)$ is added to each transition a, where $G_i, 1 \leq i \leq n$ is the guard for a *may-execute* or a *must-execute* proposition and $IG_i, 1 \leq i \leq m$ is a guard for an *inhibit* proposition.

13. A *firing style statement* (4.28) does not visibly show on a Petri Net, but it specifies the transition firing regime the Petri Net follows.

Example

Consider the following pathway specification:

$$
\begin{array}{l}
\textbf{domain of} \\
\quad f_1 \textbf{ atloc } l_1 \textbf{ is } integer, f_2 \textbf{ atloc } l_1 \textbf{ is } integer, \\
\quad f_3 \textbf{ atloc } l_1 \textbf{ is } integer, f_1 \textbf{ atloc } l_2 \textbf{ is } integer, \\
\quad f_2 \textbf{ atloc } l_2 \textbf{ is } integer, f_3 \textbf{ atloc } l_2 \textbf{ is } integer \\
t_1 \textbf{ may execute causing} \\
\qquad f_1 \textbf{ atloc } l_1 \textbf{ change value by } -1, \\
\qquad f_2 \textbf{ atloc } l_2 \textbf{ change value by } +1
\end{array}
$$

$$\begin{array}{l}
\quad \textbf{if } f_3 \textbf{ atloc } l_1 \textbf{ has value lower than } 5 \\
t_1 \textbf{ may execute causing} \\
\qquad f_1 \textbf{ atloc } l_1 \textbf{ change value by } -2, \\
\qquad f_2 \textbf{ atloc } l_2 \textbf{ change value by } +2 \\
\quad \textbf{if } f_3 \textbf{ atloc } l_1 \textbf{ has value higher than } 5 \\
\textbf{duration of } t1 \textbf{ is } 10 \\
\textbf{inhibit } t1 \textbf{ if } f_3 \textbf{ atloc } l_1 \textbf{ has value higher than } 7 \\
\textbf{inhibit } t1 \textbf{ if } f_3 \textbf{ atloc } l_1 \textbf{ has value lower than } 3 \\
\textbf{normally stimulate } t1 \textbf{ by factor } 3 \\
\quad \textbf{if } f_2 \textbf{ atloc } l_2 \textbf{ has value higher than } 5 \\
\textbf{initially} \\
\quad f_1 \textbf{ atloc } l_1 \textbf{ has value } 0, f_1 \textbf{ atloc } l_2 \textbf{ has value } 0, \\
\quad f_2 \textbf{ atloc } l_1 \textbf{ has value } 0, f_2 \textbf{ atloc } l_2 \textbf{ has value } 0, \\
\quad f_3 \textbf{ atloc } l_1 \textbf{ has value } 0, f_3 \textbf{ atloc } l_2 \textbf{ has value } 0 \\
\textbf{firing style } max
\end{array} \tag{4.32}$$

4.5 Syntax of Query Language (BioPathQA-QL)

The alphabet of query language $\mathcal{Q}$ consists of the same sets A, F, L from $\mathcal{P}$ representing actions, fluents, and locations, respectively; a fixed set of reserved keywords K shown in bold in syntax below; a fixed set $\{`:', `;', `,', `''\}$ of punctuations; a fixed set of $\{` <', ` >', ` ='\}$ of directions; and constants. Our query language asks questions about biological entities and processes in a biological pathway described through the pathway specification language. This is our domain description. A query statement is composed of a query description (the quantity, observation, or condition being sought by the question), interventions (changes to the pathway), observations (about states and actions of the pathway), and initial setup conditions.

The query statement is evaluated against the trajectories of the pathway, generated by simulating the pathway. These trajectories are modified by the initial setup and interventions. The resulting trajectories are then filtered to retain only those which satisfy the observations specified in the query statement.

A query statement can take various forms: The simplest queries do not modify the pathway and check if a specific observation is true on a trajectory or not. An observation can be a point observation or an interval observation depending upon whether they can be evaluated w.r.t. a point or an interval on a trajectory. More complex queries modify the pathway in various ways and ask for comparison of an observation before and after such modification.

Following query statements about the rate of production of '*bpg*13' illustrate the kind of queries that can be asked from our system about the specified glycolysis pathway as given in (Reece *et al.*, 2010, Figure 9.9).

Determine if '*n*' is a possible rate of production of substance '*bpg*13':

$$\begin{aligned}&\textbf{rate of production of } 'bpg13' \textbf{ is } n\\&\quad\textbf{when observed between time step } 0 \textbf{ and } \textbf{ time step } k;\end{aligned} \tag{4.33}$$

Determine if '*n*' is a possible rate of production of substance '*bpg*13' in a pathway when it is being supplied with a limited supply of an upstream substance '*f*16*bp*':

$$\begin{aligned}&\textbf{rate of production of } 'bpg13' \textbf{ is } n\\&\quad\textbf{when observed between time step } 0 \textbf{ and } \textbf{ time step } k;\\&\quad\textbf{using initial setup}: \textbf{ set value of } `f16bp' \textbf{ to } 5;\end{aligned} \tag{4.34}$$

Determine if ‘n’ is a possible rate of production of substance ‘$bpg13$’ in a pathway when it is being supplied with a steady state supply of an upstream substance ‘$f16bp$’ at the rate of 5 units per time-step:

$$\begin{aligned}&\textbf{rate of production of } 'bpg13' \textbf{ is } n\\ &\qquad \textbf{when observed between time step } 0 \textbf{ and } \ \textbf{time step } k;\\ &\qquad \textbf{using initial setup}:\ \textbf{continuously supply } `f16bp' \textbf{ in quantity } 5;\end{aligned}\tag{4.35}$$

Determine if ‘n’ is a possible rate of production of substance ‘$bpg13$’ in a pathway when it is being supplied with a steady state supply of an upstream substance ‘$f16bp$’ at the rate of 5 units per time-step and the pathway is modified to remove all quantity of the substance ’$dhap$’ as soon as it is produced:

$$\begin{aligned}&\textbf{rate of production of } 'bpg13' \textbf{ is } n\\ &\qquad \textbf{when observed between time step } 0 \textbf{ and } \ \textbf{time step } k;\\ &\qquad \textbf{due to interventions}:\ \textbf{remove } `dhap' \textbf{ as soon as produced};\\ &\qquad \textbf{using initial setup}:\ \textbf{continuously supply } `f16bp' \textbf{ in quantity } 5;\end{aligned}\tag{4.36}$$

Determine if ‘n’ is a possible rate of production of substance ‘$bpg13$’ in a pathway when it is being supplied with a steady state supply of an upstream substance ‘$f16bp$’ at the rate of 5 units per time-step and the pathway is modified to remove all quantity of the substance ’$dhap$’ as soon as it is produced and a non-functional pathway process / reaction named ‘$t5b$’:

$$\begin{aligned}&\textbf{rate of production of } 'bpg13' \textbf{ is } n\\ &\qquad \textbf{when observed between time step } 0 \textbf{ and } \ \textbf{time step } k;\\ &\qquad \textbf{due to interventions}:\ \textbf{remove } `dhap' \textbf{ as soon as produced};\\ &\qquad \textbf{due to observations}:\ `t5b' \textbf{ does not occur};\\ &\qquad \textbf{using initial setup}:\ \textbf{continuously supply } `f16bp' \textbf{ in quantity } 5;\end{aligned}\tag{4.37}$$

Determine if ‘n’ is the average rate of production of substance ‘$bpg13$’ in a pathway when it is being supplied with a steady state supply of an upstream substance ‘$f16bp$’ at the rate of 5 units per time-step and the pathway is modified to remove all quantity of the substance ’$dhap$’ as soon as it is produced and a non-functional pathway process / reaction named ‘$t5b$’:

$$\begin{aligned}&\textbf{average rate of production of } {'bpg13'} \textbf{ is } n\\&\qquad\textbf{when observed between time step } 0 \textbf{ and } \textbf{ time step } k;\\&\qquad\textbf{due to interventions}: \textbf{ remove } `dhap' \textbf{ as soon as produced};\\&\qquad\textbf{due to observations}: `t5b' \textbf{ does not occur};\\&\qquad\textbf{using initial setup}: \textbf{ continuously supply } `f16bp' \textbf{ in quantity } 5;\end{aligned}\tag{4.38}$$

Determine if ‘d’ is the direction of change in the average rate of production of substance ‘$bpg13$’ with a steady state supply of an upstream pathway input when compared with a pathway with the same steady state supply of an upstream pathway input, but in which the substance ‘$dhap$’ is removed from the pathway as soon as it is produced and pathway process / reaction called ‘$t5b$’ is non-functional:

$$\begin{aligned}&\textbf{direction of change in average rate of production of } {'bpg13'} \textbf{ is } d\\&\quad\textbf{when observed between time step } 0 \textbf{ and } \textbf{ time step } k;\\&\quad\textbf{comparing nominal pathway with modified pathway obtained}\\&\qquad\textbf{due to interventions}: \textbf{ remove } `dhap' \textbf{ as soon as produced };\\&\qquad\textbf{due to observations}: `t5b' \textbf{ does not occur};\\&\quad\textbf{using initial setup}: \textbf{ continuously supply } `f16bp' \textbf{ in quantity } 5;\end{aligned}\tag{4.39}$$

Queries can also be about actions, as illustrated in the following examples. Determine if action ‘$t5b$’ ever occurs when there is a continuous supply of ‘$f16bp$’ is available and ‘$t5a$’ is disabled:

$$
\begin{aligned}
&`t5b' \textbf{ occurs };\\
&\qquad \textbf{due to interventions}: \ \textbf{disable } `t5a';\\
&\qquad \textbf{using initial setup}: \ \textbf{continuously produce } `f16bp' \textbf{ in quantity } 5;
\end{aligned}
\tag{4.40}
$$

Determine if glycolysis (‘gly') gets replaced with beta-oxidation (‘box') when sugar (‘sug') is exhausted but fatty acids (‘fac') are available, when starting with a fixed initial supply of sugar and fatty acids in quantity 5:

$$
\begin{aligned}
&`gly' \textbf{ switches to } `box' \textbf{ when}\\
&\quad \textbf{value of } 'sug' \textbf{ is } 0,\\
&\quad \textbf{value of } 'fac' \textbf{ is higher than } 0\\
&\quad \textbf{in all trajectories};\\
&\textbf{due to observations}:\\
&\quad `gly' \textbf{ switches to } `box'\\
&\textbf{using initial setup}:\\
&\quad \textbf{set value of } `sug' \textbf{ to } 5,\\
&\quad \textbf{set value of } `fac' \textbf{ to } 5;
\end{aligned}
\tag{4.41}
$$

Next we define various syntactic elements of a query statement, give their intuitive meaning, and how these components fit together to form a query statement. We will define the formal semantics in a later section. Note that some of the single-trajectory queries can be represented as LTL formulas. However, we have chosen to keep the current representation as it is more intuitive for our biological domain.

In the following description, f_i's are fluents, l_i's are locations, n's are numbers, q's are positive integer numbers, d is one of the directions from $\{<, >, =\}$.

Definition 66 (Point) *A point is a time-step on the trajectory. It has the form:*

$$\textbf{time step } ts \tag{4.42}$$

Definition 67 (Interval) *An interval is a sub-sequence of time-steps on a trajectory. It has the form:*

$$\langle point \rangle \textbf{ and } \langle point \rangle \tag{4.43}$$

Definition 68 (Aggregate Operator (aggop)) *An aggregate operator computes an aggregate quantity over a sequence of values. It can be one of the following:*

$$\textbf{minimum} \tag{4.44}$$

$$\textbf{maximum} \tag{4.45}$$

$$\textbf{average} \tag{4.46}$$

Definition 69 (Quantitative Interval Formula) *A quantitative interval formula is a formula that is evaluated w.r.t. an interval on a trajectory for some quantity n.*

$$\textbf{rate of production of } f \textbf{ is } n \tag{4.47}$$

$$\textbf{rate of production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.48}$$

$$\textbf{rate of firing of } a \textbf{ is } n \tag{4.49}$$

$$\textbf{total production of } f \textbf{ is } n \tag{4.50}$$

$$\textbf{total production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.51}$$

Intuitively, the rate of production of a fluent f in interval $s_i, \dots, s_j$ on a trajectory $s_0, T_0, \dots, T_{k-1}, s_k$ is $n = (s_j(f) - s_i(f))/(j-i)$; rate of firing of an action a in interval $s_i, \dots, s_j$ is $n = |\{T_l : a \in T_l, i \leq l \leq j-1\}|/(j-i)$; and total production of a fluent f in interval $s_i, \dots, s_j$ is $n = s_j(f) - s_i(f)$. If the given n equals the computed n, then the formula holds. The same intuition extends to locational fluents, except that fluent f is replaced by $f[l]$, e.g. rate of production of fluent f at location l in interval $s_i, \dots, s_j$ on a trajectory is $n = (s_j(f[l]) - s_i(f[l]))/(j-i)$. In biological context, the actions represent reactions and fluents substances used in these reactions. The rate and total production formulas are used in aggregate observations to determine if reactions are slowing down or speeding up during various portions of a simulation.

Definition 70 (Quantitative Point Formula) *A quantitative point formula is a formula that is evaluated w.r.t. a point on a trajectory for some quantity n.*

$$\textbf{value of } f \textbf{ is higher than } n \tag{4.52}$$

$$\textbf{value of } f \textbf{ atloc } l \textbf{ is higher than } n \tag{4.53}$$

$$\textbf{value of } f \textbf{ is lower than } n \tag{4.54}$$

$$\textbf{value of } f \textbf{ atloc } l \textbf{ is lower than } n \tag{4.55}$$

$$\textbf{value of } f \textbf{ is } n \tag{4.56}$$

$$\textbf{value of } f \textbf{ atloc } l \textbf{ is } n \tag{4.57}$$

Definition 71 (Qualitative Interval Formula) *A qualitative interval formula is a formula that is evaluated w.r.t. an interval on a trajectory.*

$$f \textbf{ is accumulating} \tag{4.58}$$

$$f \textbf{ is accumulating } \textbf{ atloc } l \tag{4.59}$$

$$f \textbf{ is decreasing} \tag{4.60}$$

$$f \textbf{ is decreasing } \textbf{ atloc } l \tag{4.61}$$

Intuitively, a fluent f is accumulating in interval $s_i, \ldots, s_j$ on a trajectory if f's value monotonically increases during the interval. A fluent f is decreasing in interval $s_i, \ldots, s_j$ on a trajectory if f's value monotonically decreases during the interval. The same intuition extends to locational fluents by replacing f with $f[l]$.

Definition 72 (Qualitative Point Formula) *A qualitative point formula is a formula that is evaluated w.r.t. a point on a trajectory.*

$$a \textbf{ occurs} \tag{4.62}$$

$$a \textbf{ does not occur} \tag{4.63}$$

$$a1 \textbf{ switches to } a2 \tag{4.64}$$

$$\tag{4.65}$$

Intuitively, an action occurs at a point i on the trajectory if $a \in T_i$; an action does not occur at point i if $a \notin T_i$; an action $a1$ switches to $a2$ at point i if $a1 \in T_{i-1}$, $a2 \notin T_{i-1}$, $a1 \notin T_i$, $a2 \in T_i$.

Definition 73 (Quantitative All Interval Formula) *A quantitative all interval formula is a formula that is evaluated w.r.t. an interval on a set of trajectories* $\sigma_1, \ldots, \sigma_m$ *and corresponding quantities* $r_1, \ldots, r_m$.

$$\textbf{rates of production of } f \textbf{ are } [r_1, \ldots, r_m] \tag{4.66}$$

$$\textbf{rates of production of } f \textbf{ altoc } l \textbf{ are } [r_1, \ldots, r_m] \tag{4.67}$$

$$\textbf{rates of firing of } f \textbf{ are } [r_1, \ldots, r_m] \tag{4.68}$$

$$\textbf{totals of production of } f \textbf{ are } [r_1, \ldots, r_m] \tag{4.69}$$

$$\textbf{totals of production of } f \textbf{ altoc } l \textbf{ are } [r_1, \ldots, r_m] \tag{4.70}$$

Intuitively, a quantitative all interval formula holds on some interval $[i, j]$ over a set of trajectories $\sigma_1, \ldots, \sigma_m$ for values $[r_1, \ldots, r_m]$ if for each r_x the corresponding quantitative interval formula holds in interval $[i, j]$ in trajectory σ_x . For example, **rates of production of** f **are** $[r_1, \ldots, r_m]$ in interval $[i, j]$ over a set of trajectories $\sigma_1, \ldots, \sigma_m$ if for each $x \in \{1 \ldots m\}$, **rate of production of** f **is** r_x in interval $[i, j]$ in trajectory σ_x.

Definition 74 (Quantitative All Point Formula) *A quantitative all point formula is a formula that is evaluated w.r.t. a point on a set of trajectories* $\sigma_1, \ldots, \sigma_m$ *and corresponding quantities* $r_1, \ldots, r_m$.

$$\textbf{values of } f \textbf{ are } [r_1, \ldots, r_m] \tag{4.71}$$

$$\textbf{values of } f \textbf{ atloc } l \textbf{ are } [r_1, \ldots, r_m] \tag{4.72}$$

Intuitively, a quantitative all point formula holds at some point i over a set of trajectories $\sigma_1, \ldots, \sigma_m$ for values $[r_1, \ldots, r_m]$ if for each r_x the corresponding quantitative point formula holds at point i in trajectory σ_x . For example, **values of** f **are** $[r_1, \ldots, r_m]$

at point i over a set of trajectories $\sigma_1, \ldots, \sigma_m$ if for each $x \in \{1 \ldots m\}$, **value of** f **is** r_x at point i in trajectory σ_x.

Definition 75 (Quantitative Aggregate Interval Formula) *A quantitative aggregate interval formula is a formula that is evaluated w.r.t. an interval on a set of trajectories* $\sigma_1, \ldots, \sigma_m$ *and an aggregate value* r*, where* r *is the aggregate of* $[r_1, \ldots, r_m]$ *using aggop.*

$$\langle aggop \rangle \textbf{ rate of production of } f \textbf{ is } n \tag{4.73}$$

$$\langle aggop \rangle \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.74}$$

$$\langle aggop \rangle \textbf{ rate of firing of } a \textbf{ is } n \tag{4.75}$$

$$\langle aggop \rangle \textbf{ total production of } f \textbf{ is } n \tag{4.76}$$

$$\langle aggop \rangle \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.77}$$

Intuitively, a quantitative aggregate interval formula holds on some interval $[i, j]$ over a set of trajectories $\sigma_1, \ldots, \sigma_m$ for a value r if there exist $[r_1, \ldots, r_m]$ whose aggregate value per *aggop* is r, such that for each r_x the quantitative interval formula (corresponding to the quantitative aggregate interval formual) holds in interval $[i, j]$ in trajectory σ_x. For example, *average* **rate of production of** f **is** r in interval $[i, j]$ over a set of trajectories $\sigma_1, \ldots, \sigma_m$ if $r = (r_1 + \cdots + r_m)/m$ and for each $x \in \{1 \ldots m\}$, **rate of production of** f **is** r_x in interval $[i, j]$ in trajectory σ_x.

Definition 76 (Quantitative Aggregate Point Formula) *A quantitative aggregate point formula is a formula that is evaluated w.r.t. a point on a set of trajectories* $\sigma_1, \ldots, \sigma_m$ *and an aggregate value* r*, where* r *is the aggregate of* $[r_1, \ldots, r_m]$ *using aggop.*

$$\langle aggop \rangle \textbf{ value of } f \textbf{ is } r \tag{4.78}$$

$$\langle aggop \rangle \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r \tag{4.79}$$

Intuitively, a quantitative aggregate point formula holds at some point i over a set of trajectories $\sigma_1, \ldots, \sigma_m$ for a value r if there exist $[r_1, \ldots, r_m]$ whose aggregate value per *aggop* is r, such that for each r_x the quantitative point formula (corresponding to the quantitative aggregate point formual) holds at point i in trajectory σ_x. For example, *average* **value of** f **is** r at point i over a set of trajectories $\sigma_1, \ldots, \sigma_m$ if $r = (r_1 + \cdots + r_m)/m$ and for each $x \in \{1 \ldots m\}$, **value of** f **is** r_x at point i in trajectory σ_x.

Definition 77 (Quantitative Comparative Aggregate Interval Formula) *A quantitative comparative aggregate interval formula is a formula that is evaluated w.r.t. an interval over two sets of trajectories and a direction 'd' such that 'd' relates the two sets of trajectories.*

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ rate of production of } f \textbf{ is } d \tag{4.80}$$

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } d \tag{4.81}$$

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ rate of firing of } a \textbf{ is } d \tag{4.82}$$

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ total production of } f \textbf{ is } d \tag{4.83}$$

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } d \tag{4.84}$$

Intuitively, a comparative quantitative aggregate interval formula compares two quantitative interval formulas over using the direction d over a given interval.

Definition 78 (Quantitative Comparative Aggregate Point Formula) *A quantitative comparative aggregate point formula is a formula that is evaluated w.r.t. a point over two sets of trajectories and a direction 'd' such that 'd' relates the two sets of trajectories.*

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ value of } f \textbf{ is } d \tag{4.85}$$

$$\textbf{direction of change in } \langle aggop \rangle \textbf{ value of } f \textbf{ atloc } l \textbf{ is } d \tag{4.86}$$

Intuitively, a comparative quantitative aggregate point formula compares two quantitative point formulas over using the direction d at a given point.

Definition 79 (Simple Interval Formula) *A simple interval formula takes the following forms:*

$$\langle \textit{quantitative interval formula} \rangle \tag{4.87}$$

$$\langle \textit{qualitative interval formula} \rangle \tag{4.88}$$

Definition 80 (Simple Point Formula) *A simple interval formula takes the following forms:*

$$\langle \textit{quantitative point formula} \rangle \tag{4.89}$$

$$\langle \textit{qualitative point formula} \rangle \tag{4.90}$$

Definition 81 (Internal Observation Description) *An internal observation description takes the following form:*

$$\langle \textit{simple point formula} \rangle \tag{4.91}$$

$$\langle \textit{simple point formula} \rangle \ \mathbf{at} \ \langle \textit{point} \rangle \tag{4.92}$$

$$\langle \textit{simple interval formula} \rangle \tag{4.93}$$

$$\langle \textit{simple interval formula} \rangle \ \mathbf{when\ observed\ between} \ \langle \textit{interval} \rangle \tag{4.94}$$

Definition 82 (Simple Point Formula Cascade) *A simple point formula cascade takes the following form:*

$$\begin{aligned}
&\langle \textit{simple point formula} \rangle_0 \\
&\qquad \mathbf{after} \ \langle \textit{simple point formula} \rangle_{1,1}, \ldots, \langle \textit{simple point formula} \rangle_{1,n_1} \\
&\qquad \vdots \\
&\qquad \mathbf{after} \ \langle \textit{simple point formula} \rangle_{u,1}, \ldots, \langle \textit{simple point formula} \rangle_{u,n_u}
\end{aligned} \tag{4.95}$$

$$\begin{aligned}
&\langle \textit{simple point formula} \rangle_0 \\
&\qquad \mathbf{when} \ \langle \textit{simple point formula} \rangle_{1,1}, \ldots, \langle \textit{simple point formula} \rangle_{1,n_1}
\end{aligned} \tag{4.96}$$

$$\begin{aligned}
&\langle \textit{simple point formula} \rangle_0 \\
&\qquad \mathbf{when} \ \langle \textit{cond} \rangle
\end{aligned} \tag{4.97}$$

where $u \geq 1$ and 'cond' is a conjunction of simple point formulas that is true in the same point as the simple point formula.

Intuitively, the simple point formula cascade (4.95) holds if a given sequence of point formulas hold in order in a trajectory. Intuitively, simple point formula cascade (4.96) holds if a given point formula occurs at the same point as a set of simple point formulas in a trajectory. Note that these formulas and many other of our single trajectory formulas can be replaced by an LTL (Manna and Pnueli, 1992) formula, but we have kept this syntax as it is more relevant to the question answering needs in the biological domain.

Definition 83 (Query Description) *A query description specifies a non-comparative observation that can be made either on a trajectory or a set of trajectories.*

$$\langle \textit{quantitative aggregate interval formula} \rangle \textbf{ when observed between } \langle \textit{interval} \rangle \tag{4.98}$$

$$\langle \textit{quantitative aggregate point formula} \rangle \textbf{ when observed at } \langle \textit{point} \rangle \tag{4.99}$$

$$\langle \textit{quantitative all interval formula} \rangle \textbf{ when observed between } \langle \textit{interval} \rangle \tag{4.100}$$

$$\langle \textit{quantitative all point formula} \rangle \textbf{ when observed at } \langle \textit{point} \rangle \tag{4.101}$$

$$\langle \textit{internal observation description} \rangle \tag{4.102}$$

$$\langle \textit{internal observation description} \rangle \textbf{ in all trajectories} \tag{4.103}$$

$$\langle \textit{simple point formula cascade} \rangle \tag{4.104}$$

$$\langle \textit{simple point formula cascade} \rangle \textbf{ in all trajectories} \tag{4.105}$$

The single trajectory observations are can be represented using LTL formulas, but we have chosen to keep them in this form for ease of use by users from the biological domain.

Definition 84 (Comparative Query Description) *A comparative query description specifies a comparative observation that can be made w.r.t. two sets of trajectories.*

$$\begin{gathered}\langle \textit{quantitative comparative aggregate interval formula} \rangle \\ \textbf{when observed between } \langle \textit{interval} \rangle\end{gathered} \tag{4.106}$$

$$\langle \textit{quantitative comparative aggregate point formula} \rangle$$

$$\textbf{when observed at } \langle \textit{point} \rangle \tag{4.107}$$

Definition 85 (Intervention) *Interventions define modifications to domain descriptions.*

$$\textbf{remove } f_1 \textbf{ as soon as produced} \tag{4.108}$$

$$\textbf{remove } f_1 \textbf{ atloc } l_1 \textbf{ as soon as produced} \tag{4.109}$$

$$\textbf{disable } a_2 \tag{4.110}$$

$$\textbf{continuously transform } f_1 \textbf{ in quantity } q_1 \textbf{ to } f_2 \tag{4.111}$$

$$\textbf{continuously transform } f_1 \textbf{ atloc } l_1 \textbf{ in quantity } q_1 \textbf{ to } f_2 \textbf{ atloc } l_2 \tag{4.112}$$

$$\textbf{make } f_3 \textbf{ inhibit } a_3 \tag{4.113}$$

$$\textbf{make } f_3 \textbf{ atloc } l_3 \textbf{ inhibit } a_3 \tag{4.114}$$

$$\textbf{continuously supply } f_4 \textbf{ in quantity } q_4 \tag{4.115}$$

$$\textbf{contiunously supply } f_4 \textbf{ atloc } l_4 \textbf{ in quantity } q_4 \tag{4.116}$$

$$\textbf{continuously transfer } f_1 \textbf{ in quantity } q_1 \textbf{ across } l_1, l_2 \textbf{ to lower gradient} \tag{4.117}$$

$$\textbf{add delay of } q_1 \textbf{ time units in availability of } f_1 \tag{4.118}$$

$$\textbf{add delay of } q_1 \textbf{ time units in availability of } f_1 \textbf{ atloc } l_1 \tag{4.119}$$

$$\textbf{set value of } f_4 \textbf{ to } q_4 \tag{4.120}$$

$$\textbf{set value of } f_4 \textbf{ atloc } l_4 \textbf{ to } q_4 \tag{4.121}$$

Intuitively, intervention (4.108) modifies the pathway such that all quantity of f_1 is removed as soon as it is produced; intervention (4.109) modifies the pathway such that all quantity of $f_1[l_1]$ is removed as soon as it is produced; intervention (4.110) disables the action a_2; intervention (4.111) modifies the pathway such that f_1 gets converted to f_2 at the rate of q_1 units per time-unit; intervention (4.112) modifies the pathway such that $f_1[l_1]$ gets converted to $f_2[l_2]$ at the rate of q_1 units per time-unit; intervention (4.113) modifies the pathway such that action a_3 is now inhibited each time there is 1 or more units of f_3 and sets value of f_3 to 1 to initially inhibit a_3; intervention (4.114) modifies the pathway such that action a_3 is now inhibited each time there is 1 or more units of $f_3[l_3]$ and sets value of $f_3[l_3]$ to 1 to initially inhibit a_3; intervention (4.115) modifies the pathway to continuously supply f_4 at the rate of q_4 units per time-unit; intervention (4.116) modifies the pathway to continuously supply $f_4[l_4]$ at the rate of q_4 units per time-unit; intervention (4.117) modifies the pathway to transfer $f_1[l_1]$ to $f_1[l_2]$ in quantity q_1 or back depending upon whether $f_1[l_1]$ is higher than $f_1[l_2]$ or lower; intervention (4.118) modifies the pathway to add delay of q_1 time units between when f_1 is produced to when it is made available to next action; intervention (4.119) modifies the pathway to add delay of q_1 time units between when $f_1[l_1]$ is produced to when it is made available to next action; intervention (4.120) modifies the pathway to set the initial value of f_4 to q_4; and intervention (4.121) modifies the pathway to set the initial value of $f_4[l_4]$ to q_4.

Definition 86 (Initial Condition) *An initial condition is one of the intervention* (4.115)*,* (4.116)*,* (4.120)*,* (4.121) *as given in definition 85.*

Intuitively, initial conditions are interventions that setup fixed or continuous supply of substances participating in a pathway.

Definition 87 (Query Statement) *A query statement can be of the following forms:*

$$\begin{array}{l} \langle \textit{query description} \rangle; \\ \quad \textbf{due to interventions} : \langle \textit{intervention} \rangle_1, \dots, \langle \textit{intervention} \rangle_{N1}; \\ \quad \textbf{due to observations} : \langle \textit{internal observation} \rangle_1, \dots, \langle \textit{internal observation} \rangle_{N2}; \\ \quad \textbf{using initial setup} : \langle \textit{initial condition} \rangle_1, \dots, \langle \textit{initial condition} \rangle_{N3}; \end{array} \qquad (4.122)$$

$$\begin{array}{l} \langle \textit{comparative query description} \rangle; \\ \quad \textbf{comparing nominal pathway with modified pathway obtained} \\ \qquad \textbf{due to interventions} : \langle \textit{intervention} \rangle_1, \dots, \langle \textit{intervention} \rangle_{N1}; \\ \qquad \textbf{due to observations} : \langle \textit{internal observation} \rangle_1, \dots, \langle \textit{internal observation} \rangle_{N2}; \end{array}$$

using initial setup : $\langle$*initial condition*$\rangle_1, \dots, \langle$*initial condition*$\rangle_{N3}$; (4.123)

where interventions, observations, and initial setup are optional.

Intuitively, a *query statement* asks whether a *query description* holds in a pathway, perhaps after modifying it with initial setup, interventions and observations. Intuitively, a *comparative query statement* asks whether a *comparative query description* holds with a nominal pathway is compared against a modified pathway, where both pathways have the same initial setup, but only the modified pathway has been modified with interventions and observations.

4.6 Semantics of the Query Language (BioPathQA-QL)

In this section we give the semantics of our pathway specification language and the query language. The semantics of the query language is in terms of the trajectories of a domain description **D** that satisfy a query **Q**. We will present the semantics using LTL-style formulas. First, we informally define the semantics of the query language as follows.

Let **Q** be a *query statement* of the form (4.122) with a query description U, interventions $V_1, \dots, V_{|V|}$, internal observations $O_1, \dots, O_{|O|}$, and initial setup conditions $I_1, \dots, I_{|I|}$. We construct a modified domain description $\mathbf{D_1}$ by applying $I_1, \dots, I_{|I|}$ and$V_1, \dots, V_{|V|}$ to **D**. We filter the trajectories of $\mathbf{D_1}$ to retain only those trajectories that satisfy the observations $O_1, \dots, O_{|O|}$. Then we determine if U holds on any of the retained trajectories. If it does, then we say that **D** satisfies **Q**.

Let **Q** be a *comparative query statement* of the form (4.123) with *quantitative comparative aggregate query description* U, interventions $V_1, \dots, V_{|V|}$, internal observations $O_1, \dots, O_{|O|}$, and initial conditions $I_1, \dots, I_{|I|}$. Then we evaluate **Q** by deriving two sub-query statements. $\mathbf{Q_0}$ is constructed by removing the interventions

$V_1, \ldots, V_{|V|}$ and observations $O_1, \ldots, O_{|O|}$ from $\mathbf{Q}$ and replacing the *quantitative comparative aggregate query description* U with the corresponding *quantitative aggregate query description* U', $\mathbf{Q_1}$ is constructed by replacing the quantitative comparative aggregate query description U with the corresponding quantitative aggregate query description U'. Then $\mathbf{D}$ satisfies $\mathbf{Q}$ iff we can find $d \in \{<, >, =\}$ s.t. $n \; d \; n'$, where $\mathbf{D}$ satisfies $\mathbf{Q_0}$ for some value n and $\mathbf{D}$ satisfies $\mathbf{Q_1}$ for some value n'.

4.6.1 An Illustrative Example

In this section, we illustrate with an example how we intuitively evaluate a comparative query statement. In the later sections, we will give the formal semantics of query satisfaction.

Consider the following simple pathway specification:

$$\begin{array}{lll}
\textbf{domain of} & f_1 \textbf{ is } \mathit{integer}, & f_2 \textbf{ is } \mathit{integer} \\
t_1 \textbf{ may fire causing} & f_1 \textbf{ change value by } -1, & f_2 \textbf{ change value by } +1 \\
\textbf{initially} & f_1 \textbf{ has value } 0, & f_2 \textbf{ has value } 0, \\
\textbf{firing style} & \mathit{max} &
\end{array} \tag{4.124}$$

Let the following specify a query statement $\mathbf{Q}$:

$$\begin{array}{l}
\textbf{direction of change in } \mathit{average} \textbf{ rate of production of } f_2 \textbf{ is } d \\
\quad \textbf{when observed between time step } 0 \textbf{ and time step } k; \\
\quad \textbf{comparing nominal pathway with modified pathway obtained} \\
\qquad \textbf{due to interventions}: \ \textbf{remove } f_2 \textbf{ as soon as produced}; \\
\quad \textbf{using initial setup}: \ \textbf{continuously supply } f_1 \textbf{ in quantity } 1;
\end{array}$$

that we want to evaluate against $\mathbf{D}$ using a simulation length k with maximum *ntok* tokens at any place to determine 'd' that satisfies it.

We construct the baseline query $\mathbf{Q_0}$ by removing interventions and observations, and replacing the comparative aggregate quantitative query description with the corresponding aggregate quantitative query description as follows:

average **rate of production of** f_2 **is** n

when observed between time step 0 **and time step** k;

using initial setup : **continuously supply** f_1 **in quantity** 1;

We construct the alternate query $\mathbf{Q_1}$ by replacing the comparative aggregate quantitative query description with the corresponding aggregate quantitative query description as follows:

average **rate of production of** f_2 **is** n'

when observed between time step 0 **and time step** k;

due to interventions : **remove** f_2 **as soon as produced**;

using initial setup : **continuously supply** f_1 **in quantity** 1;

We build a modified domain description $\mathbf{D_0}$ as $\mathbf{D} \diamond$ (**continuously supply** f_1 **in quantity** 1) based on initial conditions in $\mathbf{Q_0}$. We get:

domain of	f_1 **is** *integer*,	f_2 **is** *integer*
t_1 **may fire causing**	f_1 **change value by** -1,	f_2 **change value by** $+1$
t_{f_1} **may fire causing**	f_1 **change value by** $+1$	
initially	f_1 **has value** 0,	f_2 **has value** 0,
firing style	*max*	

We evaluate $\mathbf{Q_0}$ against $\mathbf{D_0}$. It results in m_0 trajectories with rate of productions $n_j = (s_k(f_2) - s_0(f_2))/k$ on trajectory $\tau_j = s_0, \dots, s_k, 1 \leq j \leq m_0$ using time interval $[0, k]$. The rate of productions are averaged to produce $n = (n_1 + \cdots + n_{m_0})/m_0$.

Next, we construct the alternate domain description $\mathbf{D_1}$ as $\mathbf{D_0}\diamond$(**remove** f_2 **as soon as produced**) based on initial conditions and interventions in $\mathbf{Q_1}$. We get:

domain of	f_1 **is** *integer*,	f_2 **is** *integer*
t_1 **may fire causing**	f_1 **change value by** -1,	f_2 **change value by** $+1$
t_{f_1} **may fire causing**	f_1 **change value by** $+1$	
t_{f_2} **may fire causing**	f_2 **change value by** $*$	
initially	f_1 **has value** 0,	f_2 **has value** 0
firing style	*max*	

We evaluate $\mathbf{Q_1}$ against $\mathbf{D_1}$. Since there are no observations, no filtering is required. This results in m_1 trajectories, each with rate of production $n'_j = (s_k(f_2) - s_0(f_2))/k$ on trajectory $\tau'_j = s_0, \ldots, s_k, 1 \leq j \leq m_1$ using time interval $[0, k]$. The rate of productions are averaged to produce $n' = (n'_1 + \cdots + n'_{m_1})/m_1$.

Due to the simple nature of our domain description, it has only one trajectory for each of the two domains. As a result, for any $k > 1$, $n' < n$. Thus, $\mathbf{D}$ satisfies $\mathbf{Q}$ iff $d = `` <''$.

We will now define the semantics of how a domain description $\mathbf{D}$ is modified according to the interventions and initial conditions, the semantics of conditions imposed by the internal observations. We will then formally define how $\mathbf{Q}$ is entailed in $\mathbf{D}$.

4.6.2 Domain Transformation due to Interventions and Initial Conditions

An intervention I modifies a given domain description $\mathbf{D}$, potentially resulting in a different set of trajectories than $\mathbf{D}$. We define a binary operator $\diamond$ that transforms $\mathbf{D}$ by applying an intervention I as a set of edits to $\mathbf{D}$ using the pathway specification language. The trajectories of the modified domain description $\mathbf{D'} = \mathbf{D} \diamond I$ are given by the semantics of the pathway specification language. Below, we give the intuitive impact and edits required by each of the interventions.

Domain modification by intervention (4.108) $\mathbf{D}' = \mathbf{D} \diamond$ (**remove** f_1 **as soon as produced**) modifies the pathway by removing all existing quantity of f_1 at each time step:

$$\mathbf{D}' = \mathbf{D} + \left\{ \; tr \textbf{ may execute causing } f_1 \textbf{ change value by } * \; \right\}$$

Domain modification by intervention (4.109) $\mathbf{D}' = \mathbf{D} \diamond$(**remove** f_1 **atloc** l_1) modifies the pathway by removing all existing quantity of f_1 at each time step:

$$\mathbf{D}' = \mathbf{D} + \left\{ \; tr \textbf{ may execute causing } f_1 \textbf{ atloc } l_1 \textbf{ change value by } * \; \right\}$$

Domain modification by intervention (4.110) $\mathbf{D}' = \mathbf{D} \diamond$ (**disable** a_2) modifies the pathway such that its trajectories have $a_2 \notin T_i$, where $i \geq 0$.

$$\mathbf{D}' = \mathbf{D} + \left\{ \; \textbf{inhibit } a_2 \; \right\}$$

Domain modification by intervention (4.111) $\mathbf{D}' = \mathbf{D} \diamond$(**continuously transform** f_1 **in quantity** q_1 **to** f_2) where $s_{i+1}(f_1)$ decreases, and $s_{i+1}(f_2)$ increases by q_1 at each time step $i \geq 0$, when $s_i(f_1) \geq q_1$.

$$\mathbf{D}' = \mathbf{D} + \left\{ \begin{array}{ll} a_{f_{1,2}} \textbf{ may execute causing} & f_1 \textbf{ change value by } -q_1, \\ & f_2 \textbf{ change value by } +q_1 \end{array} \right\}$$

Domain modification by intervention (4.112) $\mathbf{D}' = \mathbf{D} \diamond$(**continuously transform** f_1 **atloc** l_1 **in quantity** q_1 **to** f_2 **atloc** l_2) where $s_{i+1}(f_1[l_1])$ decreases, and $s_{i+1}(f_2[l_2])$ increases by q_1 at each time step $i \geq 0$, when $s_i(f_1[l_1]) \geq q_1$.

$$\mathbf{D}' = \mathbf{D} + \left\{ \begin{array}{ll} a_{f_{1,2}} \textbf{ may execute causing} & f_1 \textbf{ atloc } l_1 \textbf{ change value by } -q_1, \\ & f_2 \textbf{ atloc } l_2 \textbf{ change value by } +q_1 \end{array} \right\}$$

Domain modification by intervention (4.113) $\mathbf{D'} = \mathbf{D}\diamond(\textbf{make } f_3 \textbf{ inhibit } a_3)$ modifies the pathway such that it has a_3 inhibited due to f_3.

$$\begin{aligned}\mathbf{D'} = \mathbf{D} - &\left\{ \textbf{ initially } f_3 \textbf{ has value } q \in \mathbf{D} \right\} \\ &+ \left\{ \begin{array}{ll} \textbf{inhibit } a_3 \textbf{ if} & f_3 \textbf{ has value } 1 \textbf{ or higher} \\ \textbf{initially} & f_3 \textbf{ has value } 1 \end{array} \right\}\end{aligned}$$

Domain modification by intervention (4.114) $\mathbf{D'} = \mathbf{D}\diamond(\textbf{make } f_3 \textbf{ atloc } l_3 \textbf{ inhibit } a_3)$ modifies the pathway such that it has a_3 inhibited due to $f_3[l_3]$.

$$\begin{aligned}\mathbf{D'} = \mathbf{D} - &\left\{ \textbf{ initially } f_3 \textbf{ atloc } l_3 \textbf{ has value } q \in \mathbf{D} \right\} \\ &+ \left\{ \begin{array}{ll} \textbf{inhibit } a_3 \textbf{ if} & f_3 \textbf{ atloc } l_3 \textbf{ has value } 1 \textbf{ or higher}, \\ \textbf{initially} & f_3 \textbf{ atloc } l_3 \textbf{ has value } 1 \end{array} \right\}\end{aligned}$$

Domain modification by intervention (4.115) $\mathbf{D'} = \mathbf{D}\diamond(\textbf{ continuously supply } f_4 \textbf{ in quantity } q_4)$ modifies the pathway such that a quantity q_4 of substance f_4 is supplied at each time step.

$$\mathbf{D'} = \mathbf{D} + \left\{ t_{f_4} \textbf{ may execute causing } \quad f_4 \textbf{ change value by } +q_4 \right\}$$

Domain modification by intervention (4.116) $\mathbf{D'} = \mathbf{D}\diamond(\textbf{ continuously supply } f_4 \textbf{ atloc } l_4 \textbf{ in quantity } q_4)$ modifies the pathway such that a quantity q_4 of substance f_4 at location l_4 is supplied at each time step.

$$\mathbf{D'} = \mathbf{D} + \left\{ t_{f_4} \textbf{ may execute causing } \quad f_4 \textbf{ atloc } l_4 \textbf{ change value by } +q_4 \right\}$$

Domain modification by intervention (4.117) $\mathbf{D'} = \mathbf{D}\diamond(\textbf{continuously transfer } f_1 \textbf{ in quantity } q_1 \textbf{ across } l_1, l_2 \textbf{ to lower gradient})$ modifies the pathway such that substance represented by f_1 is transferred from location l_1 to l_2 or l_2 to l_1 depending upon whether it is at a higher quantity at l_1 or l_2.

$$\mathbf{D}' = \mathbf{D} + \left\{ \begin{array}{ll} t_{f_1}\ \textbf{may execute causing} & f_1\ \textbf{atloc}\ l_1\ \textbf{change value by}\ -q_1, \\ & f_1\ \textbf{atloc}\ l_2\ \textbf{change value by}\ +q_1 \\ \quad\textbf{if} & f_1\ \textbf{atloc}\ l_1\ \textbf{has higher value than}\ f_1\ \textbf{atloc}\ l_2, \\ t'_{f_1}\ \textbf{may execute causing} & f_1\ \textbf{atloc}\ l_2\ \textbf{change value by}\ -q_1, \\ & f_1\ \textbf{atloc}\ l_1\ \textbf{change value by}\ +q_1 \\ \quad\textbf{if} & f_1\ \textbf{atloc}\ l_2\ \textbf{has higher value than}\ f_1\ \textbf{atloc}\ l_1, \end{array} \right\}$$

Domain modification by intervention (4.118) $\mathbf{D}' = \mathbf{D}\diamond($**add delay of** q_1 **time units in availability of** $f_1)$ modifies the pathway such that f_1's arrival is delayed by q_1 time units. We create additional cases for all actions that produce f_1, such that it goes through an additional delay action.

$$\mathbf{D}' = \mathbf{D} - \left\{ \begin{array}{ll} a\ \textbf{may execute causing} & f_1\ \textbf{change value by}\ +w_1, \\ & \mathit{effect}_1, \ldots, \mathit{effect}_n \\ & \textbf{if}\ \mathit{cond}_1, \ldots, \mathit{cond}_m \in \mathbf{D} \end{array} \right\}$$
$$+ \left\{ \begin{array}{ll} a\ \textbf{may execute causing} & f'_1\ \textbf{change value by}\ +w_1, \\ & \mathit{effect}_1, \ldots, \mathit{effect}_n, \\ & \textbf{if}\ \mathit{cond}_1, \ldots, \mathit{cond}_m \\ a_{f_1}\ \textbf{may execute causing} & f'_1\ \textbf{change value by}\ -w_1, \\ & f_1\ \textbf{change value by}\ +w_1 \\ a_{f_1}\ \textbf{executes} & \textbf{in}\ q_1\ \textbf{time units} \end{array} \right\}$$

Domain modification by intervention (4.119) $\mathbf{D}' = \mathbf{D}\diamond($**add delay of**$q_1$ **time units in availability of** f_1 **atloc** $l_1)$ modifies trajectories such that $f_1[l_1]$'s arrival is delayed by q_1 time units. We create additional cases for all actions that produce f_1 **atloc** l_1, such that it goes through an additional delay action.

$$\mathbf{D}' = \mathbf{D} - \left\{ \begin{array}{ll} a\ \textbf{may execute causing} & f_1\ \textbf{atloc}\ l_1\ \textbf{change value by}\ +w_1, \\ & \mathit{effect}_1, \ldots, \mathit{effect}_n \\ & \textbf{if}\ \mathit{cond}_1, \ldots, \mathit{cond}_m \in \mathbf{D} \end{array} \right\}$$

$$+\left\{\begin{array}{ll} a \textbf{ may execute causing} & f_1 \textbf{ atloc } l'_1 \textbf{ change value by } +w_1, \\ & \mathit{effect}_1, \dots, \mathit{effect}_n \\ & \textbf{ if } \mathit{cond}_1, \dots, \mathit{cond}_n \\ a_{f_1} \textbf{ may execute causing} & f_1 \textbf{ atloc } l'_1 \textbf{ change value by } -w_1, \\ & f_1 \textbf{ atloc } l_1 \textbf{ change value by } +w_1 \\ a_{f_1} \textbf{ executes} & \textbf{ in } q_1 \textbf{ time units} \end{array}\right\}$$

Domain modification by intervention (4.120) $\mathbf{D}' = \mathbf{D} \diamond (\textbf{ set value of } f_4 \textbf{ to } q_4)$ modifies the pathway such that its trajectories have $s_0(f_4) = q_4$.

$$\begin{aligned} \mathbf{D}' = \mathbf{D} - &\Big\{ \textbf{ initially } f_4 \textbf{ has value } n \in \mathbf{D} \Big\} \\ + &\Big\{ \textbf{ initially } f_4 \textbf{ has value } q_4 \Big\} \end{aligned}$$

Domain modification by intervention (4.121) $\mathbf{D}' = \mathbf{D} \diamond (\textbf{ set value of } f_4 \textbf{ atloc } l_4 \textbf{ to } q_4)$ modifies the pathway such that its trajectories have $s_0(f_{4_{l_4}}) = q_4$.

$$\begin{aligned} \mathbf{D}' = \mathbf{D} - &\Big\{ \textbf{ initially } f_4 \textbf{ atloc } l_4 \textbf{ has value } n \in \mathbf{D} \Big\} \\ + &\Big\{ \textbf{ initially } f_4 \textbf{ atloc } l_4 \textbf{ has value } q_4 \Big\} \end{aligned}$$

4.6.3 Formula Semantics

We will now define the semantics of some common formulas that we will use in the following sections. First we introduce the LTL-style formulas that we will be using to define the syntax.

A formula $\langle s_i, \sigma \rangle \models F$ represents that F holds at point i.

A formula $\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\} \models F$ represents that F holds at point i on a set of trajectories $\sigma_1, \dots, \sigma_m$.

A formula $\big\{\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1 \rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}} \rangle\}\big\} \models F$ represents that F holds at point i on two sets of trajectories $\sigma_1, \dots, \sigma_m$ and $\{\bar{\sigma}_1, \dots, \bar{\sigma}_{\bar{m}}\}$.

A formula $(\langle s_i, \sigma \rangle, j) \models F$ represents that F holds in the interval $[i, j]$ on trajectory σ.

A formula $(\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\}, j) \models F$ represents that F holds in the interval $[i, j]$ on a set of trajectories $\sigma_1, \dots, \sigma_m$.

A formula $(\big\{\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1 \rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}} \rangle\}\big\}, j) \models F$ represents that F holds in the interval $[i, j]$ over two sets of trajectories $\{\sigma_1, \dots, \sigma_m\}$ and $\{\bar{\sigma}_1, \dots, \bar{\sigma}_{\bar{m}}\}$.

Given a domain description **D** with simple fluents represented by a Guarded-Arc Petri Net as defined in definition 60. Let $\sigma = s_0, T_0, s_1, \dots, T_{k-1}, s_k$ be its trajectory as defined in (62), and s_i be a state on that trajectory. Let actions T_i firing in state s_i be observable in s_i such that $T_i \subseteq s_i$.

First we define how interval formulas are satisfied on a trajectory σ, starting state s_i and an ending point j:

$$(\langle s_i, \sigma \rangle, j) \models \textbf{rate of production of } f \textbf{ is } n$$

$$\text{if } n = (s_j(f) - s_i(f))/(j - i) \tag{4.125}$$

$$(\langle s_i, \sigma \rangle, j) \models \textbf{rate of firing of } a \textbf{ is } n$$

$$\text{if } n = \sum_{i \leq k \leq j, \langle s_k, \sigma \rangle \models a \textbf{ occurs}} 1/(j - i) \tag{4.126}$$

$$(\langle s_i, \sigma \rangle, j) \models \textbf{total production of } f \textbf{ is } n$$

$$\text{if } n = (s_j(f) - s_i(f)) \tag{4.127}$$

$$(\langle s_i, \sigma \rangle, j) \models f \textbf{ is accumulating}$$

$$\text{if } (\nexists k, i \leq k \leq j : s_{k+1}(f) < s_k(f)) \text{ and } s_j(f) > s_i(f) \tag{4.128}$$

$$(\langle s_i, \sigma \rangle, j) \models f \textbf{ is decreasing}$$

$$\text{if } (\nexists k, i \leq k \leq j : s_{k+1}(f) > s_k(f)) \text{ and } s_j(f) < s_i(f) \tag{4.129}$$

Next we define how a point formula is satisfied on a trajectory σ, in a state s_i:

$$\langle s_i, \sigma \rangle \models \textbf{value of } f \textbf{ is higher than } n$$

$$\text{if } s_i(f) > n \tag{4.130}$$

$$\langle s_i, \sigma \rangle \models \textbf{value of } f \textbf{ is lower than } n$$

$$\text{if } s_i(f) < n \tag{4.131}$$

$$\langle s_i, \sigma \rangle \models \textbf{value of } f \textbf{ is } n$$

$$\text{if } s_i(f) = n \tag{4.132}$$

$$\langle s_i, \sigma \rangle \models a \textbf{ occurs}$$

$$\text{if } a \in s_i \tag{4.133}$$

$$\langle s_i, \sigma \rangle \models a \textbf{ does not occur}$$

$$\text{if } a \notin s_i \tag{4.134}$$

$$\langle s_i, \sigma \rangle \models a_1 \textbf{ switches to } a_2$$

$$\text{if } a_1 \in s_{i-1} \text{ and } a_2 \notin s_{i-1} \text{ and } a_1 \notin s_i \text{ and } a_2 \in s_i \tag{4.135}$$

Next we define how a quantitative all interval formula is satisfied on a set of trajectories $\sigma_1, \ldots, \sigma_m$ with starting states $s_i^1, \ldots, s_i^m$ and end point j:

$$\begin{aligned}(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textbf{rates of production of } f \textbf{ are } [r_1, \ldots, r_m] \\ & \text{if } (\langle s_i^1, \sigma_1 \rangle, j) \models\ \textbf{rate of production of } f \textbf{ is } r_1 \\ & \vdots \\ & \text{and } (\langle s_i^m, \sigma_m \rangle, j) \models\ \textbf{rate of production of } f \textbf{ is } r_m \end{aligned} \tag{4.136}$$

$$\begin{aligned}(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textbf{rates of firing of } a \textbf{ are } [r_1, \ldots, r_m] \\ & \text{if } (\langle s_i^1, \sigma_1 \rangle, j) \models\ \textbf{rate of firing of } a \textbf{ is } r_1 \\ & \vdots \\ & \text{and } (\langle s_i^m, \sigma_m \rangle, j) \models\ \textbf{rate of firing of } a \textbf{ is } r_m \end{aligned} \tag{4.137}$$

$$\begin{aligned}(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textbf{totals of production of } f \textbf{ are } [r_1, \ldots, r_m] \\ & \text{if } (\langle s_i^1, \sigma_1 \rangle, j) \models\ \textbf{total production of } f \textbf{ is } r_1 \\ & \vdots \\ & \text{and } (\langle s_i^m, \sigma_m \rangle, j) \models\ \textbf{total production of } f \textbf{ is } r_m \end{aligned} \tag{4.138}$$

Next we define how a quantitative all point formula is satisfied on a set of trajectories $\sigma_1, \ldots, \sigma_m$ in states $s_i^1, \ldots, s_i^m$:

$$\begin{aligned} \{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models & \textbf{ values of } f \textbf{ are } [r_1, \ldots, r_m] \\ & \text{if } \langle s_i^1, \sigma_1\rangle \models \textbf{ value of } f \textbf{ is } r_1 \\ & \vdots \\ & \text{and } \langle s_i^m, \sigma_m\rangle \models \textbf{ value of } f \textbf{ is } r_m \end{aligned} \tag{4.139}$$

Next we define how a quantitative aggregate interval formula is satisfied on a set of trajectories $\sigma_1, \ldots, \sigma_m$ with starting states $s_i^1, \ldots, s_i^m$ and end point j:

$$\begin{aligned} (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models & \textit{ average } \textbf{rate of production of } f \textbf{ is } r \\ & \text{if } \exists [r_1, \ldots, r_m]: \\ & \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } r_1 \\ & \quad \vdots \\ & \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } r_m \\ & \text{and } r = (r_1 + \cdots + r_m)/m \end{aligned} \tag{4.140}$$

$$\begin{aligned} (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models & \textit{ minimum } \textbf{rate of production of } f \textbf{ is } r \\ & \text{if } \exists [r_1, \ldots, r_m]: \\ & \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } r_1 \\ & \quad \vdots \\ & \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } r_m \\ & \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x \end{aligned} \tag{4.141}$$

$$\begin{aligned} (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models & \textit{ maximum } \textbf{rate of production of } f \textbf{ is } r \\ & \text{if } \exists [r_1, \ldots, r_m]: \\ & \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } r_1 \\ & \quad \vdots \\ & \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } r_m \end{aligned}$$

$$\text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x \tag{4.142}$$

$$
\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{average} \textbf{ rate of firing of } f \textbf{ is } r \\
& \text{iff } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_m \\
& \text{and } r = (r_1 + \cdots + r_m)/m
\end{aligned} \tag{4.143}
$$

$$
\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{minimum} \textbf{ rate of firing of } f \textbf{ is } r \\
& \text{iff } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x
\end{aligned} \tag{4.144}
$$

$$
\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{maximum} \textbf{ rate of firing of } f \textbf{ is } r \\
& \text{iff } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x
\end{aligned} \tag{4.145}
$$

$$
\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{average} \textbf{ total production of } f \textbf{ is } r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ total production of } f \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ total production of } f \textbf{ is } r_m \\
& \text{and } r = (r_1 + \cdots + r_m)/m
\end{aligned} \tag{4.146}
$$

$$
\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{minimum}\ \textbf{total production of}\ f\ \textbf{is}\ r \\
& \text{if } \exists[r_1, \ldots, r_m]: \\
& \quad (\langle s_i^1, \sigma_1\rangle, j) \models\ \textbf{total production of}\ f\ \textbf{is}\ r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m\rangle, j) \models\ \textbf{total production of}\ f\ \textbf{is}\ r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x
\end{aligned}
\tag{4.147}
$$

$$
\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{maximum}\ \textbf{total production of}\ f\ \textbf{is}\ r \\
& \text{if } \exists[r_1, \ldots, r_m]: \\
& \quad (\langle s_i^1, \sigma_1\rangle, j) \models\ \textbf{total production of}\ f\ \textbf{is}\ r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m\rangle, j) \models\ \textbf{total production of}\ f\ \textbf{is}\ r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x
\end{aligned}
\tag{4.148}
$$

Next we define how a quantitative aggregate point formula is satisfied on a set of trajectories $\sigma_1, \ldots, \sigma_m$ in states $s_i^1, \ldots, s_i^m$:

$$
\begin{aligned}
\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models\ & \textit{average}\ \textbf{value of}\ f\ \textbf{is}\ r \\
& \text{if } \exists[r_1, \ldots, r_m]: \\
& \quad \langle s_i^1, \sigma_1\rangle \models\ \textbf{value of}\ f\ \textbf{is}\ r_1 \\
& \quad \vdots \\
& \quad \langle s_i^m, \sigma_m\rangle \models\ \textbf{value of}\ f\ \textbf{is}\ r_m \\
& \text{and } r = (r_1 + \cdots + r_m)/m
\end{aligned}
\tag{4.149}
$$

$$
\begin{aligned}
\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models\ & \textit{minimum}\ \textbf{value of}\ f\ \textbf{is}\ r \\
& \text{if } \exists[r_1, \ldots, r_m]: \\
& \quad \langle s_i^1, \sigma_1\rangle \models\ \textbf{value of}\ f\ \textbf{is}\ r_1 \\
& \quad \vdots \\
& \quad \langle s_i^m, \sigma_m\rangle \models\ \textbf{value of}\ f\ \textbf{is}\ r_m
\end{aligned}
$$

$$\text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x \tag{4.150}$$

$$\begin{aligned}
\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models\ & \textit{maximum}\ \textbf{value of}\ f\ \textbf{is}\ r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
& \quad \langle s_i^1, \sigma_1\rangle \models\ \textbf{value of}\ f\ \textbf{is}\ r_1 \\
& \quad \vdots \\
& \quad \langle s_i^m, \sigma_m\rangle \models\ \textbf{value of}\ f\ \textbf{is}\ r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x
\end{aligned} \tag{4.151}$$

Next we define how a comparative quantitative aggregate interval formula is satisfied on two sets of trajectories $\sigma_1, \ldots, \sigma_m$, and $\bar{\sigma}_1, \ldots, \bar{\sigma}_{\bar{m}}$ a starting point i and an ending point j:

$$\begin{aligned}
&\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big) \\
&\quad \models\ \textbf{direction of change in}\ \textit{average}\ \textbf{rate of production of}\ f\ \textbf{is}\ d \\
&\quad \text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ \textit{average}\ \textbf{rate of production of}\ f\ \text{is}\ n_1 \\
&\quad \text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models\ \textit{average}\ \textbf{rate of production of}\ f\ \text{is}\ n_2 \\
&\quad \text{and } n_2\ d\ n_1
\end{aligned} \tag{4.152}$$

$$\begin{aligned}
&\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big) \\
&\quad \models\ \textbf{direction of change in}\ \textit{minimum}\ \textbf{rate of production of}\ f\ \textbf{is}\ d \\
&\quad \text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ \textit{minimum}\ \textbf{rate of production of}\ f\ \text{is}\ n_1 \\
&\quad \text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models\ \textit{minimum}\ \textbf{rate of production of}\ f\ \text{is}\ n_2 \\
&\quad \text{and } n_2\ d\ n_1
\end{aligned} \tag{4.153}$$

$$\begin{aligned}
&\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big) \\
&\quad \models\ \textbf{direction of change in}\ \textit{maximum}\ \textbf{rate of production of}\ f\ \textbf{is}\ d \\
&\quad \text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ \textit{maximum}\ \textbf{rate of production of}\ f\ \text{is}\ n_1 \\
&\quad \text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models\ \textit{maximum}\ \textbf{rate of production of}\ f\ \text{is}\ n_2 \\
&\quad \text{and } n_2\ d\ n_1
\end{aligned} \tag{4.154}$$

$$\Big(\Big\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\},j\Big)$$

$$\begin{aligned}
&\models\ \textbf{direction of change in } \textit{average} \textbf{ rate of firing of } f \textbf{ is } d\\
&\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models\ \textit{average}\ \textbf{rate of firing of } f \text{ is } n_1\\
&\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models\ \textit{average}\ \textbf{rate of firing of } f \text{ is } n_2\\
&\text{and } n_2\ d\ n_1
\end{aligned} \tag{4.155}$$

$$\Big(\Big\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\},j\Big)$$

$$\begin{aligned}
&\models\ \textbf{direction of change in } \textit{minimum} \textbf{ rate of firing of } f \textbf{ is } d\\
&\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models\ \textit{minimum}\ \textbf{rate of firing of } f \text{ is } n_1\\
&\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models\ \textit{minimum}\ \textbf{rate of firing of } f \text{ is } n_2\\
&\text{and } n_2\ d\ n_1
\end{aligned} \tag{4.156}$$

$$\Big(\Big\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\},j\Big)$$

$$\begin{aligned}
&\models\ \textbf{direction of change in } \textit{maximum} \textbf{ rate of firing of } f \textbf{ is } d\\
&\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models\ \textit{maximum}\ \textbf{rate of firing of } f \text{ is } n_1\\
&\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models\ \textit{maximum}\ \textbf{rate of firing of } f \text{ is } n_2\\
&\text{and } n_2\ d\ n_1
\end{aligned} \tag{4.157}$$

$$\Big(\Big\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\},j\Big)$$

$$\begin{aligned}
&\models\ \textbf{direction of change in } \textit{average} \textbf{ total production of } f \textbf{ is } d\\
&\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models\ \textit{average}\ \textbf{total production of } f \text{ is } n_1\\
&\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models\ \textit{average}\ \textbf{total production of } f \text{ is } n_2\\
&\text{and } n_2\ d\ n_1
\end{aligned} \tag{4.158}$$

$$\Big(\Big\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\},j\Big)$$

$$\begin{aligned}
&\models\ \textbf{direction of change in } \textit{minimum} \textbf{ total production of } f \textbf{ is } d\\
&\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models\ \textit{minimum}\ \textbf{total production of } f \text{ is } n_1\\
&\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models\ \textit{minimum}\ \textbf{total production of } f \text{ is } n_2\\
&\text{and } n_2\ d\ n_1
\end{aligned} \tag{4.159}$$

$$\Big(\Big\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\},j\Big)$$

$$\begin{aligned}&\models \ \textbf{direction of change in } \mathit{maximum} \textbf{ total production of } f \textbf{ is } d\\&\text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\}, j) \models \mathit{maximum} \textbf{ total production of } f \text{ is } n_1\\&\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models \mathit{maximum} \textbf{ total production of } f \text{ is } n_2\\&\text{and } n_2\ d\ n_1 \qquad (4.160)\end{aligned}$$

Next we define how a comparative quantitative aggregate point formula is satisfied on two sets of trajectories $\sigma_1, \dots, \sigma_m$, and $\bar{\sigma}_1, \dots, \bar{\sigma}_{\bar{m}}$ at point i:

$$\begin{aligned}&\Big\{\{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}\\&\quad\models \ \textbf{direction of change in } \mathit{average} \textbf{ value of } f \textbf{ is } d\\&\quad\text{if } \exists\, n_1 : \{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\} \models \mathit{average} \textbf{ value of } f \text{ is } n_1\\&\quad\text{and } \exists\, n_2 : \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\} \models \mathit{average} \textbf{ value of } f \text{ is } n_2\\&\quad\text{and } n_2\ d\ n_1 \qquad (4.161)\\&\Big\{\{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}\\&\quad\models \ \textbf{direction of change in } \mathit{minimum} \textbf{ value of } f \textbf{ is } d\\&\quad\text{if } \exists\, n_1 : \{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\} \models \mathit{minimum} \textbf{ value of } f \text{ is } n_1\\&\quad\text{and } \exists\, n_2 : \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\} \models \mathit{minimum} \textbf{ value of } f \text{ is } n_2\\&\quad\text{and } n_2\ d\ n_1 \qquad (4.162)\\&\Big\{\{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}\\&\quad\models \ \textbf{direction of change in } \mathit{maximum} \textbf{ value of } f \textbf{ is } d\\&\quad\text{if } \exists\, n_1 : \{\langle s_i^1, \sigma_1\rangle, \dots, \langle s_i^m, \sigma_m\rangle\} \models \mathit{maximum} \textbf{ value of } f \text{ is } n_1\\&\quad\text{and } \exists\, n_2 : \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \dots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\} \models \mathit{maximum} \textbf{ value of } f \text{ is } n_2\\&\quad\text{and } n_2\ d\ n_1 \qquad (4.163)\end{aligned}$$

Given a domain description **D** with simple fluents represented by a Guarded-Arc Petri Net with Colored tokens as defined in definition 63. Let $\sigma = s_0, T_0, s_1, \dots, T_{k-1}, s_k$

be its trajectory as defined in definition 65, and s_i be a state on that trajectory. Let actions T_i firing in state s_i be observable in s_i such that $T_i \subseteq s_i$. We define observation semantics using LTL below. We will use $s_i(f[l])$ to represent $m_{s_i(l)}(f)$ (multiplicity / value of f in location l) in state s_i.

First we define how interval formulas are satisfied on a trajectory σ, starting state s_i and an ending point j:

$$(\langle s_i, \sigma \rangle, j) \models \textbf{rate of production of } f \textbf{ atloc } l \textbf{ is } n$$

$$\text{if } n = (s_j(f[l]) - s_i(f[l]))/(j - i) \tag{4.164}$$

$$(\langle s_i, \sigma \rangle, j) \models \textbf{rate of firing of } a \textbf{ is } n$$

$$\text{if } n = \sum_{i \leq k \leq j, \langle s_k, \sigma \rangle \models a \textbf{ occurs}} 1/(j - i) \tag{4.165}$$

$$(\langle s_i, \sigma \rangle, j) \models \textbf{total production of } f \textbf{ atloc } l \textbf{ is } n$$

$$\text{if } n = (s_j(f[l]) - s_i(f[l])) \tag{4.166}$$

$$(\langle s_i, \sigma \rangle, j) \models f \textbf{ is accumulating atloc } l$$

$$\text{if } (\nexists k, i \leq k \leq j : s_{k+1}(f[l]) < s_k(f[l])) \text{ and } s_j(f[l]) > s_i(f[l]) \tag{4.167}$$

$$(\langle s_i, \sigma \rangle, j) \models f \textbf{ is decreasing atloc } l$$

$$\text{if } (\nexists k, i \leq k \leq j : s_{k+1}(f[l]) > s_k(f[l])) \text{ and } s_j(f[l]) < s_i(f[l]) \tag{4.168}$$

Next we define how a point formula is satisfied on a trajectory σ, in a state s_i:

$$\langle s_i, \sigma \rangle \models \textbf{value of } f \textbf{ atloc } l \textbf{ is higher than } n$$

$$\text{if } s_i(f[l]) > n \tag{4.169}$$

$$\langle s_i, \sigma \rangle \models \textbf{value of } f \textbf{ atloc } l \textbf{ is lower than } n$$

$$\text{if } s_i(f[l]) < n \tag{4.170}$$

$$\langle s_i, \sigma \rangle \models \textbf{value of } f \textbf{ atloc } l \textbf{ is } n$$

$$\text{if } s_i(f[l]) = n \tag{4.171}$$

$$\langle s_i, \sigma \rangle \models a \textbf{ occurs}$$

$$\text{if } a \in s_i \tag{4.172}$$

$$\langle s_i, \sigma \rangle \models a \textbf{ does not occur}$$

$$\text{if } a \notin s_i \tag{4.173}$$

$$\langle s_i, \sigma \rangle \models a_1 \textbf{ switches to } a_2$$

$$\text{if } a_1 \in s_{i-1} \text{ and } a_2 \notin s_{i-1} \text{ and } a_1 \notin s_i \text{ and } a_2 \in s_i \tag{4.174}$$

Next we define how a quantitative all interval formula is satisfied on a set of trajectories $\sigma_1, \dots, \sigma_m$ with starting states $s_i^1, \dots, s_i^m$ and end point j:

$$(\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\}, j) \models \textbf{rates of production of } f \textbf{ atloc } l \textbf{ are } [r_1, \dots, r_m]$$

$$\text{if } (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$\text{and } (\langle s_i^m, \sigma_m \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_m \tag{4.175}$$

$$(\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\}, j) \models \textbf{rates of firing of } a \textbf{ are } [r_1, \dots, r_m]$$

$$\text{if } (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ rate of firing of } a \textbf{ is } r_1$$

$$\vdots$$

$$\text{and } (\langle s_i^m, \sigma_m \rangle, j) \models \textbf{ rate of firing of } a \textbf{ is } r_m \tag{4.176}$$

$$(\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\}, j) \models \textbf{totals of production of } f \textbf{ atloc } l \textbf{ are } [r_1, \dots, r_m]$$

$$\text{if } (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$\text{and } (\langle s_i^m, \sigma_m \rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_m \tag{4.177}$$

Next we define how a quantitative all point formula is satisfied w.r.t. a set of trajectories $\sigma_1, \dots, \sigma_m$ in states $s_i^1, \dots, s_i^m$:

$$\{\langle s_i^1, \sigma_1 \rangle, \dots, \langle s_i^m, \sigma_m \rangle\} \models \textbf{values of } f \textbf{ atloc } l \textbf{ are } [r_1, \dots, r_m]$$

$$\text{if } \langle s_i^1, \sigma_1 \rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$\text{and } \langle s_i^m, \sigma_m \rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_m \tag{4.178}$$

Next we define how a quantitative aggregate interval formula is satisfied w.r.t. a set of trajectories $\sigma_1, \ldots, \sigma_m$ with starting states $s_i^1, \ldots, s_i^m$ and end point j:

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textit{average} \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_m \\
& \text{and } r = (r_1 + \cdots + r_m)/m
\end{aligned} \tag{4.179}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textit{minimum} \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x
\end{aligned} \tag{4.180}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textit{maximum} \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_1 \\
& \quad \vdots \\
& \quad (\langle s_i^m, \sigma_m \rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } r_m \\
& \text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x
\end{aligned} \tag{4.181}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1 \rangle, \ldots, \langle s_i^m, \sigma_m \rangle\}, j) \models\ & \textit{average} \textbf{ rate of firing of } f \textbf{ is } r \\
& \text{iff } \exists [r_1, \ldots, r_m] : \\
& \quad (\langle s_i^1, \sigma_1 \rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_1
\end{aligned}$$

$$\begin{aligned}
&\qquad\vdots \\
&\qquad(\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_m \\
&\text{and } r = (r_1 + \cdots + r_m)/m
\end{aligned} \tag{4.182}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{minimum} \textbf{ rate of firing of } f \textbf{ is } r \\
& \text{iff } \exists [r_1, \ldots, r_m] : \\
&\qquad(\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_1 \\
&\qquad\vdots \\
&\qquad(\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_m \\
& \text{and } \exists k, 1 \le k \le m : r = r_k \text{ and } \forall x, 1 \le x \le m, r_k \le r_x
\end{aligned} \tag{4.183}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{maximum} \textbf{ rate of firing of } f \textbf{ is } r \\
& \text{iff } \exists [r_1, \ldots, r_m] : \\
&\qquad(\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_1 \\
&\qquad\vdots \\
&\qquad(\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ rate of firing of } f \textbf{ is } r_m \\
& \text{and } \exists k, 1 \le k \le m : r = r_k \text{ and } \forall x, 1 \le x \le m, r_k \ge r_x
\end{aligned} \tag{4.184}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{average} \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
&\qquad(\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_1 \\
&\qquad\vdots \\
&\qquad(\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_m \\
& \text{and } r = (r_1 + \cdots + r_m)/m
\end{aligned} \tag{4.185}$$

$$\begin{aligned}
(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models\ & \textit{minimum} \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r \\
& \text{if } \exists [r_1, \ldots, r_m] : \\
&\qquad(\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_1 \\
&\qquad\vdots \\
&\qquad(\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_m
\end{aligned}$$

$$\text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x \qquad (4.186)$$

$$(\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models \textit{maximum } \textbf{total production of } f \textbf{ atloc } l \textbf{ is } r$$

$$\text{if } \exists [r_1, \ldots, r_m] :$$

$$(\langle s_i^1, \sigma_1\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$(\langle s_i^m, \sigma_m\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } r_m$$

$$\text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x \qquad (4.187)$$

Next we define how a quantitative aggregate point formula is satisfied w.r.t. a set of trajectories $\sigma_1, \ldots, \sigma_m$ in states $s_i^1, \ldots, s_i^m$:

$$\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models \textit{average } \textbf{value of } f \textbf{ atloc } l \textbf{ is } r$$

$$\text{if } \exists [r_1, \ldots, r_m] :$$

$$\langle s_i^1, \sigma_1\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$\langle s_i^m, \sigma_m\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_m$$

$$\text{and } r = (r_1 + \cdots + r_m)/m \qquad (4.188)$$

$$\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models \textit{minimum } \textbf{value of } f \textbf{ atloc } l \textbf{ is } r$$

$$\text{if } \exists [r_1, \ldots, r_m] :$$

$$\langle s_i^1, \sigma_1\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$\langle s_i^m, \sigma_m\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_m$$

$$\text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \leq r_x \qquad (4.189)$$

$$\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models \textit{maximum } \textbf{value of } f \textbf{ atloc } l \textbf{ is } r$$

$$\text{if } \exists [r_1, \ldots, r_m] :$$

$$\langle s_i^1, \sigma_1\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_1$$

$$\vdots$$

$$\langle s_i^m, \sigma_m \rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } r_m$$

$$\text{and } \exists k, 1 \leq k \leq m : r = r_k \text{ and } \forall x, 1 \leq x \leq m, r_k \geq r_x \qquad (4.190)$$

Next we define how a comparative quantitative aggregate interval formula is satisfied w.r.t. two sets of trajectories $\sigma_1, \ldots, \sigma_m$, and $\bar{\sigma}_1, \ldots, \bar{\sigma}_{\bar{m}}$ a starting point i and an ending point j:

$$\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big)$$

$$\models \textbf{ direction of change in } \textit{average} \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } d$$

$$\text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models \textit{ average } \textbf{rate of production of } f \textbf{ atloc } l \text{ is } n_1$$

$$\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models \textit{ average } \textbf{rate of production of } f \textbf{ atloc } l \text{ is } n_2$$

$$\text{and } n_2\ d\ n_1 \qquad (4.191)$$

$$\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big)$$

$$\models \textbf{ direction of change in } \textit{minimum} \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } d$$

$$\text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models \textit{ minimum } \textbf{rate of production of } f \textbf{ atloc } l \text{ is } n_1$$

$$\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models \textit{ minimum } \textbf{rate of production of } f \textbf{ atloc } l \text{ is } n_2$$

$$\text{and } n_2\ d\ n_1 \qquad (4.192)$$

$$\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big)$$

$$\models \textbf{ direction of change in } \textit{maximum} \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } d$$

$$\text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models \textit{ maximum } \textbf{rate of production of } f \textbf{ atloc } l \text{ is } n_1$$

$$\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models \textit{ maximum } \textbf{rate of production of } f \textbf{ atloc } l \text{ is } n_2$$

$$\text{and } n_2\ d\ n_1 \qquad (4.193)$$

$$\Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big)$$

$$\models \textbf{ direction of change in } \textit{average} \textbf{ rate of firing of } f \textbf{ is } d$$

$$\text{if } \exists\, n_1 : (\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, j) \models \textit{ average } \textbf{rate of firing of } f \text{ is } n_1$$

$$\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}, j) \models \textit{ average } \textbf{rate of firing of } f \text{ is } n_2$$

$$\text{and } n_2\ d\ n_1 \qquad (4.194)$$

$$\begin{aligned}
&\left(\left\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\right\},j\right)\\
&\quad\models\ \textbf{direction of change in } \textit{minimum} \textbf{ rate of firing of } f \textbf{ is } d\\
&\quad\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models \textit{minimum} \textbf{ rate of firing of } f \text{ is } n_1\\
&\quad\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models \textit{minimum} \textbf{ rate of firing of } f \text{ is } n_2\\
&\quad\text{and } n_2\ d\ n_1
\end{aligned}\tag{4.195}$$

$$\begin{aligned}
&\left(\left\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\right\},j\right)\\
&\quad\models\ \textbf{direction of change in } \textit{maximum} \textbf{ rate of firing of } f \textbf{ is } d\\
&\quad\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models \textit{maximum} \textbf{ rate of firing of } f \text{ is } n_1\\
&\quad\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models \textit{maximum} \textbf{ rate of firing of } f \text{ is } n_2\\
&\quad\text{and } n_2\ d\ n_1
\end{aligned}\tag{4.196}$$

$$\begin{aligned}
&\left(\left\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\right\},j\right)\\
&\quad\models\ \textbf{direction of change in } \textit{average} \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } d\\
&\quad\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models \textit{average} \textbf{ total production of } f \textbf{ atloc } l \text{ is } n_1\\
&\quad\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models \textit{average} \textbf{ total production of } f \textbf{ atloc } l \text{ is } n_2\\
&\quad\text{and } n_2\ d\ n_1
\end{aligned}\tag{4.197}$$

$$\begin{aligned}
&\left(\left\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\right\},j\right)\\
&\quad\models\ \textbf{direction of change in } \textit{minimum} \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } d\\
&\quad\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models \textit{minimum} \textbf{ total production of } f \textbf{ atloc } l \text{ is } n_1\\
&\quad\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models \textit{minimum} \textbf{ total production of } f \textbf{ atloc } l \text{ is } n_2\\
&\quad\text{and } n_2\ d\ n_1
\end{aligned}\tag{4.198}$$

$$\begin{aligned}
&\left(\left\{\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\right\},j\right)\\
&\quad\models\ \textbf{direction of change in } \textit{maximum} \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } d\\
&\quad\text{if } \exists\, n_1 : (\{\langle s_i^1,\sigma_1\rangle,\ldots,\langle s_i^m,\sigma_m\rangle\},j) \models \textit{maximum} \textbf{ total production of } f \textbf{ atloc } l \text{ is } n_1\\
&\quad\text{and } \exists\, n_2 : (\{\langle \bar{s}_i^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_i^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\},j) \models \textit{maximum} \textbf{ total production of } f \textbf{ atloc } l \text{ is } n_2\\
&\quad\text{and } n_2\ d\ n_1
\end{aligned}\tag{4.199}$$

Next we define how a comparative quantitative aggregate point formula is satisfied w.r.t. two sets of trajectories $\sigma_1, \ldots, \sigma_m$, and $\bar{\sigma}_1, \ldots, \bar{\sigma}_{\bar{m}}$ at point i:

$$\begin{aligned}
&\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\quad \models \ \textbf{direction of change in } \textit{average} \textbf{ value of } f \textbf{ atloc } l \textbf{ is } d \\
&\quad \text{if } \exists\, n_1 : \{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models \ \textit{average} \textbf{ value of } f \textbf{ atloc } l \text{ is } n_1 \\
&\quad \text{and } \exists\, n_2 : \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\} \models \ \textit{average} \textbf{ value of } f \textbf{ atloc } l \text{ is } n_2 \\
&\quad \text{and } n_2 \ d \ n_1 \qquad (4.200)
\end{aligned}$$

$$\begin{aligned}
&\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\quad \models \ \textbf{direction of change in } \textit{minimum} \textbf{ value of } f \textbf{ atloc } l \textbf{ is } d \\
&\quad \text{if } \exists\, n_1 : \{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models \ \textit{minimum} \textbf{ value of } f \textbf{ atloc } l \text{ is } n_1 \\
&\quad \text{and } \exists\, n_2 : \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\} \models \ \textit{minimum} \textbf{ value of } f \textbf{ atloc } l \text{ is } n_2 \\
&\quad \text{and } n_2 \ d \ n_1 \qquad (4.201)
\end{aligned}$$

$$\begin{aligned}
&\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\quad \models \ \textbf{direction of change in } \textit{maximum} \textbf{ value of } f \textbf{ atloc } l \textbf{ is } d \\
&\quad \text{if } \exists\, n_1 : \{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\} \models \ \textit{maximum} \textbf{ value of } f \textbf{ atloc } l \text{ is } n_1 \\
&\quad \text{and } \exists\, n_2 : \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\} \models \ \textit{maximum} \textbf{ value of } f \textbf{ atloc } l \text{ is } n_2 \\
&\quad \text{and } n_2 \ d \ n_1 \qquad (4.202)
\end{aligned}$$

4.6.4 Trajectory Filtering due to Internal Observations

The trajectories produced by the Guarded-Arc Petri Net execution are filtered to retain only the trajectories that satisfy all internal observations in a query statement. Let $\sigma = s_0, \ldots, s_k$ be a trajectory as given in definition 62. Then σ satisfies an observation:

$$\begin{aligned}
&\textbf{rate of production of } f \textbf{ is } n \\
&\qquad \text{if } (\langle s_0, \sigma\rangle, k) \models \ \textbf{rate of production of } f \textbf{ is } n \qquad (4.203)
\end{aligned}$$

rate of production of f **is** n

$$\text{if } (\langle s_0, \sigma\rangle, k) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.204}$$

total production of f **is** n

$$\text{if } (\langle s_0, \sigma\rangle, k) \models \textbf{ total production of } f \textbf{ is } n \tag{4.205}$$

f **is accumulating**

$$\text{if } (\langle s_0, \sigma\rangle, k) \models f \textbf{ is accumulating} \tag{4.206}$$

f **is decreasing**

$$\text{if } (\langle s_0, \sigma\rangle, k) \models f \textbf{ is decreasing} \tag{4.207}$$

rate of production of f **is** n

when observed between time step i **and time step** j

$$\text{if } (\langle s_i, \sigma\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } n \tag{4.208}$$

rate of production of f **is** n

when observed between time step i **and time step** j

$$\text{if } (\langle s_i, \sigma\rangle, j) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.209}$$

total production of f **is** n

when observed between time step i **and time step** j

$$\text{if } (\langle s_i, \sigma\rangle, j) \models \textbf{ total production of } f \textbf{ is } n \tag{4.210}$$

f **is accumulating**

when observed between time step i **and time step** j

$$\text{if } (\langle s_i, \sigma\rangle, j) \models f \textbf{ is accumulating} \tag{4.211}$$

f **is decreasing**

when observed between time step i **and time step** j

$$\text{if } (\langle s_i, \sigma\rangle, j) \models f \textbf{ is decreasing} \tag{4.212}$$

f **is higher than** n

$$\text{if } \exists i, 0 \leq i \leq k : \langle s_i, \sigma\rangle \models \textbf{ value of } f \textbf{ is higher than } n \tag{4.213}$$

f **is lower than** n

$$\text{if } \exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models \textbf{ value of } f \textbf{ is lower than } n \tag{4.214}$$

f **is** n

$$\text{if } \exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models \textbf{ value of } f \textbf{ is } n \tag{4.215}$$

a **occurs**

$$\text{if } \exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models a \textbf{ occurs} \tag{4.216}$$

a **does not occur**

$$\text{if } \exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models a \textbf{ does not occur} \tag{4.217}$$

a_1 **switches to** a_2

$$\text{if } \exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models a_1 \textbf{ switches to } a_2$$

f **is higher than** n

at time step i

$$\text{if } \langle s_i, \sigma \rangle \models \textbf{ value of } f \textbf{ is higher than } n \tag{4.218}$$

f **is lower than** n

at time step i

$$\text{if } \langle s_i, \sigma \rangle \models \textbf{ value of } f \textbf{ is lower than } n \tag{4.219}$$

f **is** n

at time step i

$$\text{if } \langle s_i, \sigma \rangle \models \textbf{ value of } f \textbf{ is } n \tag{4.220}$$

a **occurs**

at time step i

$$\text{if } \langle s_i, \sigma \rangle \models a \textbf{ occurs} \tag{4.221}$$

a **does not occur**

at time step i

$$\text{if } \langle s_i, \sigma \rangle \models a \textbf{ does not occur} \tag{4.222}$$

a_1 **switches to** a_2

at time step i

if $\langle s_i, \sigma \rangle \models a_1$ **switches to** a_2 (4.223)

Let $\sigma = s_0, \dots, s_k$ be a trajectory of the form (65). Then σ satisfies an observation:

rate of production of f **atloc** l **is** n

if $(\langle s_0, \sigma \rangle, k) \models$ **rate of production of** f **atloc** l **is** n (4.224)

rate of production of f **atloc** l **is** n

if $(\langle s_0, \sigma \rangle, k) \models$ **rate of firing of** a **is** n (4.225)

total production of f **atloc** l **is** n

if $(\langle s_0, \sigma \rangle, k) \models$ **total production of** f **atloc** l **is** n (4.226)

f **is accumulating atloc** l

if $(\langle s_0, \sigma \rangle, k) \models f$ **is accumulating atloc** l (4.227)

f **is decreasing atloc** l

if $(\langle s_0, \sigma \rangle, k) \models f$ **is decreasing atloc** l (4.228)

rate of production of f **atloc** l **is** n

when observed between time step i **and time step** j

if $(\langle s_i, \sigma \rangle, j) \models$ **rate of production of** f **atloc** l **is** n (4.229)

rate of production of f **atloc** l **is** n

when observed between time step i **and time step** j

if $(\langle s_i, \sigma \rangle, j) \models$ **rate of firing of** a **is** n (4.230)

total production of f **atloc** l **is** n

when observed between time step i **and time step** j

if $(\langle s_i, \sigma \rangle, j) \models$ **total production of** f **atloc** l **is** n (4.231)

f **is accumulating atloc** l

when observed between time step i **and time step** j

if $(\langle s_i, \sigma \rangle, j) \models f$ **is accumulating atloc** l (4.232)

f **is decreasing atloc** l

when observed between time step i **and time step** j

if $(\langle s_i, \sigma \rangle, j) \models f$ **is decreasing atloc** l (4.233)

f **atloc** l **is higher than** n

if $\exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models$ **value of** f **atloc** l **is higher than** n (4.234)

f **atloc** l **is lower than** n

if $\exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models$ **value of** f **atloc** l **is lower than** n (4.235)

f **atloc** l **is** n

if $\exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models$ **value of** f **atloc** l **is** n (4.236)

a **occurs**

if $\exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models a$ **occurs** (4.237)

a **does not occur**

if $\exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models a$ **does not occur** (4.238)

a_1 **switches to** a_2

if $\exists i, 0 \leq i \leq k : \langle s_i, \sigma \rangle \models a_1$ **switches to** a_2 (4.239)

f **atloc** l **is higher than** n

at time step i

if $\langle s_i, \sigma \rangle \models$ **value of** f **atloc** l **is higher than** n (4.240)

f **atloc** l **is lower than** n

at time step i

if $\langle s_i, \sigma \rangle \models$ **value of** f **atloc** l **is lower than** n (4.241)

f **atloc** l **is** n

at time step i

if $\langle s_i, \sigma \rangle \models$ **value of** f **atloc** l **is** n (4.242)

a **occurs**

at time step i

if $\langle s_i, \sigma \rangle \models a$ **occurs** (4.243)

a **does not occur**

at time step i

if $\langle s_i, \sigma \rangle \models a$ **does not occur** (4.244)

a_1 **switches to** a_2

at time step i

if $\langle s_i, \sigma \rangle \models a_1$ **switches to** a_2 (4.245)

A trajectory σ is kept for further processing w.r.t. a set of internal observations $\langle$internal observation$\rangle_1, \ldots, \langle$internal observation$\rangle_n$ if $\sigma \models \langle$internal observation$\rangle_i$, $1 \leq i \leq n$.

4.6.5 Query Description Satisfaction

Now, we define query statement semantics using LTL syntax. Let $\mathbf{D}$ be a domain description with simple fluents and $\sigma = s_0, \ldots, s_k$ be its trajectory of length k as defined in (62). Let $\sigma_1, \ldots, \sigma_m$ represent the set of trajectories of $\mathbf{D}$ filtered by observations as necessary, with each trajectory has the form $\sigma_i = s^i_0, \ldots, s^i_k$ $1 \leq i \leq m$. Let $\bar{\mathbf{D}}$ be a modified domain description and $\bar{\sigma}_1, \ldots, \bar{\sigma}_{\bar{m}}$ be its trajectories of the form $\bar{\sigma}_i = \bar{s}^i_0, \ldots, \bar{s}^i_k$. Then we define query satisfaction using the formula satisfaction in section 4.6.3 as follows.

Two sets of trajectories $\sigma_1, \ldots, \sigma_m$ and $\bar{\sigma}_1, \ldots, \bar{\sigma}_{\bar{m}}$ satisfy a comparative query description based on formula satisfaction of section 4.6.3 as follows:

$$\begin{aligned}
&\Big\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\qquad \models \textbf{ direction of change in } \textit{aggop} \textbf{ rate of production of } f \textbf{ is } d \\
&\qquad\qquad \textbf{when observed between time step } i \textbf{ and time step } j \\
&\qquad \text{if } \Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big) \\
&\qquad\quad \models \textbf{ direction of change in } \textit{aggop} \textbf{ rate of production of } f \textbf{ is } d
\end{aligned} \tag{4.246}$$

$$\begin{aligned}
&\Big\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\qquad \models \textbf{ direction of change in } \textit{aggop} \textbf{ rate of firing of } f \textbf{ is } d \\
&\qquad\qquad \textbf{when observed between time step } i \textbf{ and time step } j \\
&\qquad \text{if } \Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big) \\
&\qquad\quad \models \textbf{ direction of change in } \textit{aggop} \textbf{ rate of firing of } f \textbf{ is } d
\end{aligned} \tag{4.247}$$

$$\begin{aligned}
&\Big\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\qquad \models \textbf{ direction of change in } \textit{aggop} \textbf{ total production of } f \textbf{ is } d \\
&\qquad\qquad \textbf{when observed between time step } i \textbf{ and time step } j \\
&\qquad \text{if } \Big(\Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\}, j\Big) \\
&\qquad\quad \models \textbf{ direction of change in } \textit{aggop} \textbf{ total production of } f \textbf{ is } d
\end{aligned} \tag{4.248}$$

$$\begin{aligned}
&\Big\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\qquad \models \textbf{ direction of change in } \textit{aggop} \textbf{ value of } f \textbf{ is } d \\
&\qquad\qquad \textbf{when observed at time step } i \\
&\qquad \text{if } \Big\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\Big\} \\
&\qquad\quad \models \textbf{ direction of change in } \textit{aggop} \textbf{ value of } f \textbf{ is } d
\end{aligned} \tag{4.249}$$

A set of trajectories $\sigma_1, \ldots, \sigma_m$, each of length k satisfies a quantitative or a qualitative interval query description based on formula satisfaction of section 4.6.3 as follows:

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of production of } f \textbf{ is } n$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ rate of production of } f \textbf{ is } n \tag{4.250}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of firing of } a \textbf{ is } n$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.251}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ total production of } f \textbf{ is } n$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ total production of } f \textbf{ is } n \tag{4.252}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of production of } f \textbf{ is } n$$

$$\textbf{when observed between time step } i \textbf{ and time step } j$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models \textbf{ rate of production of } f \textbf{ is } n \tag{4.253}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of firing of } a \textbf{ is } n$$

$$\textbf{when observed between time step } i \textbf{ and time step } j$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.254}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ total production of } f \textbf{ is } n$$

$$\textbf{when observed between time step } i \textbf{ and time step } j$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models \textbf{ total production of } f \textbf{ is } n \tag{4.255}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is accumulating}$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models f \textbf{ is accumulating} \tag{4.256}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is decreasing}$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models f \textbf{ is decreasing} \tag{4.257}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f$ **is accumulating**

when observed between time step i **and time step** j

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models f \textbf{ is accumulating} \tag{4.258}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f$ **is decreasing**

when observed between time step i **and time step** j

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models f \textbf{ is decreasing} \tag{4.259}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **rate of production of** f **is** n

in all trajectories

$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models \textbf{ rate of production of } f \textbf{ is } n \tag{4.260}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **rate of firing of** a **is** n

in all trajectories

$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.261}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **total production of** f **is** n

in all trajectories

$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models \textbf{ total production of } f \textbf{ is } n \tag{4.262}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **rate of production of** f **is** n

when observed between time step i **and time step** j

in all trajectories

$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models \textbf{ rate of production of } f \textbf{ is } n \tag{4.263}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **rate of firing of** a **is** n

when observed between time step i **and time step** j

in all trajectories

$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.264}$$

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **total production of** f **is** n

when observed between time step i **and time step** j

$$\textbf{in all trajectories}$$

$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models \textbf{ total production of } f \textbf{ is } n \tag{4.265}$$

$$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f \textbf{ is accumulating}$$
$$\textbf{in all trajectories}$$
$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models f \textbf{ is accumulating} \tag{4.266}$$

$$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f \textbf{ is decreasing}$$
$$\textbf{in all trajectories}$$
$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models f \textbf{ is decreasing} \tag{4.267}$$

$$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f \textbf{ is accumulating}$$
$$\textbf{when observed between time step } i \textbf{ and time step } j$$
$$\textbf{in all trajectories}$$
$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models f \textbf{ is accumulating} \tag{4.268}$$

$$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f \textbf{ is decreasing}$$
$$\textbf{when observed between time step } i \textbf{ and time step } j$$
$$\textbf{in all trajectories}$$
$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models f \textbf{ is decreasing} \tag{4.269}$$

A set of trajectories $\sigma_1, \ldots, \sigma_m$, each of length k satisfies a quantitative or a qualitative point query description based on formula satisfaction of section 4.6.3 as follows:

$$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models \textbf{ value of } f \textbf{ is higher than } n$$
$$\text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x \rangle \models \textbf{ value of } f \textbf{ is higher than } n \tag{4.270}$$

$$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models \textbf{ value of } f \textbf{ is lower than } n$$
$$\text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x \rangle \models \textbf{ value of } f \textbf{ is lower than } n \tag{4.271}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is } n$$

$$\text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is } n \tag{4.272}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is higher than } n$$

$$\textbf{at time step } i$$

$$\text{if } \exists x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is higher than } n \tag{4.273}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is lower than } n$$

$$\textbf{at time step } i$$

$$\text{if } \exists x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is lower than } n \tag{4.274}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is } n$$

$$\textbf{at time step } i$$

$$\text{if } \exists x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is } n \tag{4.275}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is higher than } n$$

$$\textbf{in all trajectories}$$

$$\text{if } \forall x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is higher than } n \tag{4.276}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is lower than } n$$

$$\textbf{in all trajectories}$$

$$\text{if } \forall x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is lower than } n \tag{4.277}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ is } n$$

$$\textbf{in all trajectories}$$

$$\text{if } \forall x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ is } n \tag{4.278}$$

$$\begin{aligned}\{\langle s_0^1,\sigma_1\rangle,\ldots,\langle s_0^m,\sigma_m\rangle\} \models\ & \textbf{value of } f \textbf{ is higher than } n\\ & \quad\textbf{at time step } i\\ & \quad\textbf{in all trajectories}\\ & \text{if } \forall x, 1 \leq x \leq m : \langle s_i^x,\sigma_x\rangle \models\ \textbf{value of } f \textbf{ is higher than } n \end{aligned} \tag{4.279}$$

$$\begin{aligned}\{\langle s_0^1,\sigma_1\rangle,\ldots,\langle s_0^m,\sigma_m\rangle\} \models\ & \textbf{value of } f \textbf{ is lower than } n\\ & \quad\textbf{at time step } i\\ & \quad\textbf{in all trajectories}\\ & \text{if } \forall x, 1 \leq x \leq m : \langle s_i^x,\sigma_x\rangle \models\ \textbf{value of } f \textbf{ is lower than } n \end{aligned} \tag{4.280}$$

$$\begin{aligned}\{\langle s_0^1,\sigma_1\rangle,\ldots,\langle s_0^m,\sigma_m\rangle\} \models\ & \textbf{value of } f \textbf{ is } n\\ & \quad\textbf{at time step } i\\ & \quad\textbf{in all trajectories}\\ & \text{if } \forall x, 1 \leq x \leq m : \langle s_i^x,\sigma_x\rangle \models\ \textbf{value of } f \textbf{ is } n \end{aligned} \tag{4.281}$$

Now, we turn our attention to domain descriptions with locational fluents. Let $\mathbf{D}$ be a domain description and $\sigma = s_0,\ldots,s_k$ be its trajectory of length k as defined in (65). Let $\sigma_1,\ldots,\sigma_m$ represent the set of trajectories of $\mathbf{D}$ filtered by observations as necessary, with each trajectory has the form $\sigma_i = s_0^i,\ldots,s_k^i$ $1 \leq i \leq m$. Let $\bar{\mathbf{D}}$ be a modified domain description and $\bar{\sigma}_1,\ldots,\bar{\sigma}_{\bar{m}}$ be its trajectories of the form $\bar{\sigma}_i = \bar{s}_0^i,\ldots,\bar{s}_k^i$. Then we define query satisfaction using the formula satisfaction in section 4.6.3 as follows.

Two sets of trajectories $\sigma_1,\ldots,\sigma_m$ and $\bar{\sigma}_1,\ldots,\bar{\sigma}_{\bar{m}}$ satisfy a comparative query description based on formula satisfaction of section 4.6.3 as follows:

$$\begin{aligned}&\Big\{\{\langle s_0^1,\sigma_1\rangle,\ldots,\langle s_0^m,\sigma_m\rangle\},\{\langle \bar{s}_0^1,\bar{\sigma}_1\rangle,\ldots,\langle \bar{s}_0^{\bar{m}},\bar{\sigma}_{\bar{m}}\rangle\}\Big\}\\ &\quad\models\ \textbf{direction of change in } \langle\text{aggop}\rangle \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } d\\ &\quad\quad\quad\textbf{when observed between time step } i \textbf{ and time step } j\end{aligned}$$

$$\text{if } \left(\left\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}, j\right)$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } d \tag{4.282}$$

$$\left\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ rate of firing of } f \textbf{ is } d$$

$$\textbf{when observed between time step } i \textbf{ and time step } j$$

$$\text{if } \left(\left\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}, j\right)$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ rate of firing of } f \textbf{ is } d \tag{4.283}$$

$$\left\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } d$$

$$\textbf{when observed between time step } i \textbf{ and time step } j$$

$$\text{if } \left(\left\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}, j\right)$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } d \tag{4.284}$$

$$\left\{\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\}, \{\langle \bar{s}_0^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_0^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ value of } f \textbf{ atloc } l \textbf{ is } d$$

$$\textbf{when observed at time step } i$$

$$\text{if } \left\{\{\langle s_i^1, \sigma_1\rangle, \ldots, \langle s_i^m, \sigma_m\rangle\}, \{\langle \bar{s}_i^1, \bar{\sigma}_1\rangle, \ldots, \langle \bar{s}_i^{\bar{m}}, \bar{\sigma}_{\bar{m}}\rangle\}\right\}$$

$$\models \textbf{ direction of change in } \langle\text{aggop}\rangle \textbf{ value of } f \textbf{ atloc } l \textbf{ is } d \tag{4.285}$$

A set of trajectories $\sigma_1, \ldots, \sigma_m$, each of length k satisfies a quantitative or a qualitative interval query description based on formula satisfaction of section 4.6.3 as follows:

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.286}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of firing of } a \textbf{ is } n$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.287}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } n$$

$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.288}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n$$
$$\qquad \textbf{when observed between time step } i \textbf{ and time step } j$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.289}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of firing of } a \textbf{ is } n$$
$$\qquad \textbf{when observed between time step } i \textbf{ and time step } j$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models \textbf{ rate of firing of } a \textbf{ is } n \tag{4.290}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } n$$
$$\qquad \textbf{when observed between time step } i \textbf{ and time step } j$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models \textbf{ total production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.291}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is accumulating } \textbf{ atloc } l$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models f \textbf{ is accumulating } \textbf{ atloc } l \tag{4.292}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is decreasing } \textbf{ atloc } l$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models f \textbf{ is decreasing } \textbf{ atloc } l \tag{4.293}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is accumulating } \textbf{ atloc } l$$
$$\qquad \textbf{when observed between time step } i \textbf{ and time step } j$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models f \textbf{ is accumulating } \textbf{ atloc } l \tag{4.294}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is decreasing } \textbf{ atloc } l$$
$$\qquad \textbf{when observed between time step } i \textbf{ and time step } j$$
$$\text{if } \exists x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models f \textbf{ is decreasing } \textbf{ atloc } l \tag{4.295}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n$$
$$\qquad \textbf{in all trajectories}$$
$$\text{if } \forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x\rangle, k) \models \textbf{ rate of production of } f \textbf{ atloc } l \textbf{ is } n \tag{4.296}$$

$$\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ rate of firing of } a \textbf{ is } n$$

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models$ **rate of firing of** a **is** n (4.297)

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **total production of** f **atloc** l **is** n

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models$ **total production of** f **atloc** l **is** n (4.298)

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **rate of production of** f **atloc** l **is** n

when observed between time step i **and time step** j

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models$ **rate of production of** f **atloc** l **is** n (4.299)

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **rate of firing of** a **is** n

when observed between time step i **and time step** j

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models$ **rate of firing of** a **is** n (4.300)

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models$ **total production of** f **atloc** l **is** n

when observed between time step i **and time step** j

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x \rangle, j) \models$ **total production of** f **atloc** l **is** n till j (4.301)

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f$ **is accumulating atloc** l

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models f$ **is accumulating atloc** l (4.302)

$\{\langle s_0^1, \sigma_1 \rangle, \ldots, \langle s_0^m, \sigma_m \rangle\} \models f$ **is decreasing atloc** l

in all trajectories

if $\forall x, 1 \leq x \leq m : (\langle s_0^x, \sigma_x \rangle, k) \models f$ **is decreasing atloc** l (4.303)

$$\begin{aligned}
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is accumulating } \textbf{ atloc } l\\
&\qquad \textbf{when observed between time step } i \textbf{ and time step } j\\
&\qquad \textbf{in all trajectories}\\
&\quad \text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models f \textbf{ is accumulating } \textbf{ atloc } l && (4.304)\\
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models f \textbf{ is decreasing } \textbf{ atloc } l\\
&\qquad \textbf{when observed between time step } i \textbf{ and time step } j\\
&\qquad \textbf{in all trajectories}\\
&\quad \text{if } \forall x, 1 \leq x \leq m : (\langle s_i^x, \sigma_x\rangle, j) \models f \textbf{ is decreasing } \textbf{ atloc } l && (4.305)
\end{aligned}$$

A set of trajectories $\sigma_1, \ldots, \sigma_m$, each of length k satisfies a quantitative or a qualitative point query description based on formula satisfaction of section 4.6.3 as follows:

$$\begin{aligned}
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n\\
&\quad \text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n && (4.306)\\
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n\\
&\quad \text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n && (4.307)\\
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\\
&\quad \text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n && (4.308)
\end{aligned}$$

$$\begin{aligned}
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n\\
&\qquad \textbf{at time step } i\\
&\quad \text{if } \exists x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n && (4.309)\\
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n\\
&\qquad \textbf{at time step } i\\
&\quad \text{if } \exists x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n && (4.310)
\end{aligned}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\\&\qquad\textbf{at time step } i\\&\quad \text{if } \exists x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\end{aligned}\tag{4.311}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n\\&\qquad\textbf{in all trajectories}\\&\quad \text{if } \forall x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n\end{aligned}\tag{4.312}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n\\&\qquad\textbf{in all trajectories}\\&\quad \text{if } \forall x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n\end{aligned}\tag{4.313}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\\&\qquad\textbf{in all trajectories}\\&\quad \text{if } \forall x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\end{aligned}\tag{4.314}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n\\&\qquad\textbf{at time step } i\\&\qquad\textbf{in all trajectories}\\&\quad \text{if } \forall x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is higher than } n\end{aligned}\tag{4.315}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n\\&\qquad\textbf{at time step } i\\&\qquad\textbf{in all trajectories}\\&\quad \text{if } \forall x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is lower than } n\end{aligned}\tag{4.316}$$

$$\begin{aligned}&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\\&\qquad\textbf{at time step } i\\&\qquad\textbf{in all trajectories}\\&\quad \text{if } \forall x, 1 \leq x \leq m : \langle s_i^x, \sigma_x\rangle \models \textbf{ value of } f \textbf{ atloc } l \textbf{ is } n\end{aligned}\tag{4.317}$$

Next, we generically define the satisfaction of a *simple point formula cascade* query w.r.t. a set of trajectories $\sigma_1, \ldots, \sigma_m$. The trajectories will either be as defined in definitions (62) or (65) for *simple point formula cascade* query statement made up of simple fluents or locational fluents, respectively. A set of trajectories $\sigma_1, \ldots, \sigma_m$, each of length k satisfies a *simple point formula cascade* based on formula satisfaction of section 4.6.3 as follows:

$$\begin{aligned}
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \langle\text{simple point formula}\rangle_0\\
&\qquad \textbf{after } \langle\text{simple point formula}\rangle_{1,1}, \ldots, \langle\text{simple point formula}\rangle_{1,n_1}\\
&\qquad \vdots\\
&\qquad \textbf{after } \langle\text{simple point formula}\rangle_{u,1}, \ldots, \langle\text{simple point formula}\rangle_{u,n_u}\\
&\quad \text{if } \exists x \exists i_0 \exists i_1 \ldots \exists i_u, 1 \leq x \leq m, i_1 < i_0 \leq k, \ldots, i_u < i_{u-1} \leq k, 0 \leq i_u \leq k :\\
&\quad \langle s_{i_0}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_0 \text{ and}\\
&\quad \langle s_{i_1}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_{1,1} \text{ and } \ldots \text{ and } \langle s_{i_1}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_{1,n_1} \text{ and}\\
&\qquad \vdots\\
&\quad \langle s_{i_u}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_{u,1} \text{ and } \ldots \text{ and } \langle s_{i_u}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_{u,n_u}
\end{aligned} \tag{4.318}$$

$$\begin{aligned}
&\{\langle s_0^1, \sigma_1\rangle, \ldots, \langle s_0^m, \sigma_m\rangle\} \models \langle\text{simple point formula}\rangle_0\\
&\qquad \textbf{after } \langle\text{simple point formula}\rangle_{1,1}, \ldots, \langle\text{simple point formula}\rangle_{1,n_1}\\
&\qquad \vdots\\
&\qquad \textbf{after } \langle\text{simple point formula}\rangle_{u,1}, \ldots, \langle\text{simple point formula}\rangle_{u,n_u}\\
&\qquad \textbf{in all trajectories}\\
&\quad \text{if } \forall x, 1 \leq x \leq m, \exists i_0 \exists i_1 \ldots \exists i_u, i_1 < i_0 \leq k, \ldots, i_u < i_{u-1} \leq k, 0 \leq i_u \leq k :\\
&\quad \langle s_{i_0}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_0 \text{ and}\\
&\quad \langle s_{i_1}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_{1,1} \text{ and } \ldots \text{ and } \langle s_{i_1}^x, \sigma_x\rangle \models \langle\text{simple point formula}\rangle_{1,n_1} \text{ and}
\end{aligned}$$

$$\vdots$$

$$\langle s^x_{i_u}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_{u,1} \text{ and } \ldots \text{ and } \langle s^x_{i_u}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_{u,n_u} \tag{4.319}$$

$$\begin{aligned}
&\{\langle s^1_0, \sigma_1 \rangle, \ldots, \langle s^m_0, \sigma_m \rangle\} \models \langle \text{simple point formula} \rangle_0 \\
&\qquad \textbf{when } \langle \text{simple point formula} \rangle_{1,1}, \ldots, \langle \text{simple point formula} \rangle_{1,n_1} \\
&\quad \text{if } \exists x \exists i, 1 \leq x \leq m, 0 \leq i \leq k : \\
&\qquad \langle s^x_{i_0}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_0 \text{ and} \\
&\qquad \langle s^x_{i_1}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_{1,1} \text{ and } \ldots \text{ and } \langle s^x_{i_1}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_{1,n_1}
\end{aligned} \tag{4.320}$$

$$\begin{aligned}
&\{\langle s^1_0, \sigma_1 \rangle, \ldots, \langle s^m_0, \sigma_m \rangle\} \models \langle \text{simple point formula} \rangle_0 \\
&\qquad \textbf{when } \langle \text{simple point formula} \rangle_{1,1}, \ldots, \langle \text{simple point formula} \rangle_{1,n_1} \\
&\qquad \textbf{in all trajectories} \\
&\quad \text{if } \forall x, 1 \leq x \leq m, \exists i, 0 \leq i \leq k : \\
&\qquad \langle s^x_{i_0}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_0 \text{ and} \\
&\qquad \langle s^x_{i_1}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_{1,1} \text{ and } \ldots \text{ and } \langle s^x_{i_1}, \sigma_x \rangle \models \langle \text{simple point formula} \rangle_{1,n_1}
\end{aligned} \tag{4.321}$$

4.6.6 Query Statement Satisfaction

Let **D** as defined in section 4.3 be a domain description and **Q** be a query statement (4.122) as defined in section 4.5 with query description U, interventions $V_1, \ldots, V_{|V|}$, internal observations $O_1, \ldots, O_{|O|}$, and initial conditions $I_1, \ldots, I_{|I|}$. Let $\mathbf{D_1} \equiv \mathbf{D} \diamond I_1 \diamond \cdots \diamond I_{|I|} \diamond V_1 \diamond \cdots \diamond V_{|V|}$ be the modified domain description constructed by applying the initial conditions and interventions from $\mathbf{Q}$ as defined in section 4.6.2. Let $\sigma_1, \dots, \sigma_m$ be the trajectories of $\mathbf{D_1}$ that satisfy $O_1, \dots, O_{|O|}$ as given in section 4.6.4. Then, $\mathbf{D}$ satisfies $\mathbf{Q}$ if $\{\sigma_1, \dots, \sigma_m\} \models U$ as defined in section 4.6.5.

Let $\mathbf{D}$ as defined in section 4.3 be a domain description and $\mathbf{Q}$ be a query statement (4.123) as defined in section 4.5 with query description U, interventions $V_1, \dots, V_{|V|}$, internal observations $O_1, \dots, O_{|O|}$, and initial conditions $I_1, \dots, I_{|I|}$. Let $\mathbf{D_)} \equiv \mathbf{D} \diamond I_1 \diamond \cdots \diamond I_{|I|}$ be the nominal domain description constructed by applying the initial conditions from $\mathbf{Q}$ as defined in section 4.6.2. Let $\sigma_1, \dots, \sigma_m$ be the trajectories of $\mathbf{D_0}$. Let $\mathbf{D_1} \equiv \mathbf{D} \diamond I_1 \diamond \cdots \diamond I_{|I|} \diamond V_1 \diamond \cdots \diamond V_{|V|}$ be the modified domain description constructed by applying the initial conditions and interventions from $\mathbf{Q}$ as defined in section 4.6.2. Let $\bar{\sigma}_1, \dots, \bar{\sigma}_{\bar{m}}$ be the trajectories of $\mathbf{D_1}$ that satisfy $O_1, \dots, O_{|O|}$ as given in section 4.6.4. Then, $\mathbf{D}$ satisfies $\mathbf{Q}$ if $\Big\{\{\sigma_1, \dots, \sigma_m\}, \{\bar{\sigma}_1, \dots, \bar{\sigma}_{\bar{m}}\}\Big\} \models U$ as defined in section 4.6.5.

4.6.7 Example Encodings

In this section we give some examples of how we will encode queries and pathways related to these queries. We will also show how the pathway is modified to answer questions [3].

Question 11 *At one point in the process of glycolysis, both DHAP and G3P are produced. Isomerase catalyzes the reversible conversion between these two isomers. The conversion of DHAP to G3P never reaches equilibrium and G3P is used in the next step of glycolysis. What would happen to the rate of glycolysis if DHAP were removed from the process of glycolysis as quickly as it was produced?*

[3] Some of the same pathways appear in previous chapters, they have been updated here with additional background knowledge.

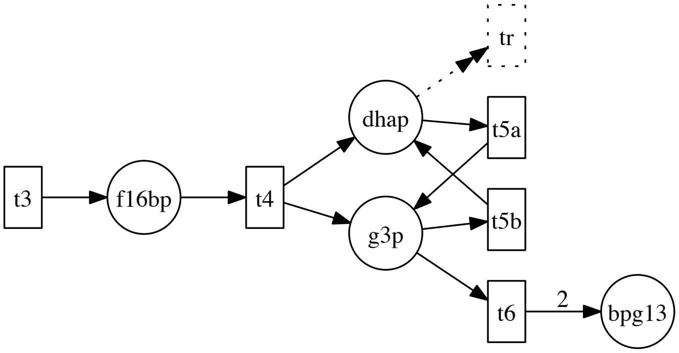


Figure 4.3: Petri Net for question 11

The question is asking for the direction of change in the *rate of glycolysis* when the nominal glycolysis pathway is compared against a modified pathway in which *dhap* is removed as soon as it is produced. Since this rate can vary with the trajectory followed by the world evolution, we consider the average change in rate. From the domain knowledge (Reece *et al.*, 2010, Figure 9.9) we know that the rate of *glycolysis* can be measured by the rate of *pyruvate* (the end product of glycolysis) and that the rate of *pyruvate* is equal to the rate of $bpg13$ (due to linear chain from $bpg13$ to *pyruvate*). Thus, we can monitor the rate of $bpg13$ instead to determine the rate of glycolysis. To ensure that our pathway is not starved due to source ingredients, we add a continuous supply of $f16bp$ in quantity 1 to the pathway.

Then, the following pathway specification encodes the domain description $\mathbf{D}$ for question 11 and produces the PN in Fig. 4.3 minus the $tr, t3$ transitions:

$$\begin{array}{lll}
\textbf{domain of} & f16bp \textbf{ is } integer, & dhap \textbf{ is } integer, \\
 & g3p \textbf{ is } integer, & bpg13 \textbf{ is } integer \\
t4 \textbf{ may execute causing} & f16bp \textbf{ change value by } -1, & dhap \textbf{ change value by } +1, \\
 & g3p \textbf{ change value by } +1 & \\
t5a \textbf{ may execute causing} & dhap \textbf{ change value by } -1, & g3p \textbf{ change value by } +1 \\
t5b \textbf{ may execute causing} & g3p \textbf{ change value by } -1, & dhap \textbf{ change value by } +1 \\
t6 \textbf{ may execute causing} & g3p \textbf{ change value by } -1, & bpg13 \textbf{ change value by } +2 \\
\textbf{initially} & f16bp \textbf{ has value } 0, & dhap \textbf{ has value } 0, \\
 & g3p \textbf{ has value } 0, & bpg13 \textbf{ has value } 0 \\
\textbf{firing style} & max &
\end{array} \tag{4.322}$$

And the following query $\mathbf{Q}$ for a simulation of length k encodes the question:

$$\begin{array}{l}
\textbf{direction of change in } average \textbf{ rate of production of } bpg13 \textbf{ is } d \\
\quad \textbf{when observed between time step } 0 \textbf{ and time step } k; \\
\quad \textbf{comparing nominal pathway with modified pathway obtained} \\
\qquad \textbf{due to interventions} : \textbf{remove } dhap \textbf{ as soon as produced}; \\
\quad \textbf{using initial setup} : \textbf{ continuously supply } f16bp \textbf{ in quantity } 1;
\end{array} \tag{4.323}$$

Since this is a comparative quantitative query statement, it is decomposed into two sub-queries, $\mathbf{Q_0}$ capturing the nominal case of average rate of production w.r.t. given initial conditions:

$$\begin{array}{l}
average \textbf{ rate of production of } bpg13 \textbf{ is } n_{avg} \\
\qquad \textbf{when observed between time step } 0 \textbf{ and time step } k; \\
\quad \textbf{using initial setup} : \textbf{ continuously supply } f16bp \textbf{ in quantity } 1;
\end{array}$$

and $\mathbf{Q_1}$ capturing the modified case in which the pathway is subject to interventions and observations w.r.t. initial conditions:

$$\begin{array}{l} \textit{average}\ \textbf{rate of production of}\ bpg13\ \textbf{is}\ n'_{avg} \\ \qquad \textbf{when observed between time step}\ 0\ \textbf{and time step}\ k; \\ \quad \textbf{due to interventions}: \textbf{remove}\ dhap\ \textbf{as soon as produced}; \\ \quad \textbf{using initial setup}:\ \textbf{continuously supply}\ f16bp\ \textbf{in quantity}\ 1; \end{array}$$

Then the task is to determine d, such that $\mathbf{D} \models \mathbf{Q_0}$ for some value of n_{avg}, $\mathbf{D} \models \mathbf{Q_1}$ for some value of n'_{avg}, and $n'_{avg}\ d\ n_{avg}$. To answer the sub-queries $\mathbf{Q_0}$ and $\mathbf{Q_1}$ we build modified domain descriptions $\mathbf{D_0}$ and $\mathbf{D_1}$, where, $\mathbf{D_0} \equiv \mathbf{D} \diamond$ (**continuously supply** $f16bp$ **in quantity** 1) is the nominal domain description $\mathbf{D}$ modified according to $\mathbf{Q_0}$ to include the initial conditions; and $\mathbf{D_1} \equiv \mathbf{D_0} \diamond$ (**remove** *dhap* **as soon as produced**) is the modified domain description $\mathbf{D}$ modified according to $\mathbf{Q_1}$ to include the initial conditions as well as the interventions. The $\diamond$ operator modifies the domain description to its left by adding, removing or modifying pathway specification language statements to add the *interventions* and *initial conditions* description to its right. Thus,

$$\mathbf{D_0} = \mathbf{D} + \left\{\ tf_{f16bp}\ \textbf{may execute causing}\quad f16bp\ \textbf{change value by}\ +1 \right.$$

$$\mathbf{D_1} = \mathbf{D_0} + \left\{\ tr\ \textbf{may execute causing}\quad dhap\ \textbf{change value by}\ * \right.$$

Performing a simulation of $k = 5$ steps with $ntok = 20$ max tokens, we find that the average rate of $bpg13$ production decreases from $n_{avg} = 0.83$ to $n'_{avg} = 0.5$. Thus, $\mathbf{D} \models \mathbf{Q}$ iff $d = '<'$. Alternatively, we say that the rate of glycolysis decreases when DHAP is removed as quickly as it is produced.

Question 12 *When and how does the body switch to B-oxidation versus glycolysis as the major way of burning fuel?*

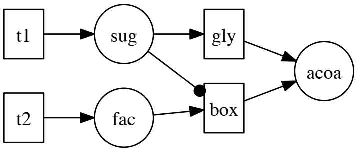


Figure 4.4: Petri Net for question 12

The following pathway specification encodes the domain description **D** for question 12 and produces the PN in Fig. 4.4:

domain of	*gly* **is** *integer*,	*sug* **is** *integer*,
	fac **is** *integer*,	*acoa* **is** *integer*
gly **may execute causing**	*sug* **change value by** -1,	*acoa* **change value by** $+1$
box **may execute causing**	*fac* **change value by** -1,	*acoa* **change value by** $+1$
inhibit *box* **if**	*sug* **has value** 1 **or higher**	
initially	*sug* **has value** 3,	*fac* **has value** 3
	acoa **has value** 0	
$t1$ **may execute causing**	*sug* **change value by** $+1$	
$t2$ **may execute causing**	*fac* **change value by** $+1$	
firing style	$*$	

where, *fac* represents fatty acids, *sug* represents sugar, *acoa* represents acetyl coenzyme-A, *gly* represents the process of glycolysis, and *box* represents the process of beta oxidation.

The question is asking for the general conditions when glycolysis switches to beta-oxidation, which is some property "p" that holds after which the switch occurs. The query **Q** is encoded as:

gly **switches to** *box* **when** p;
due to observations :
gly **switches to** *box*
in all trajectories

where condition 'p' is a conjunction of *simple point formulas*. Then the task is to determine a minimal such conjunction of formulas that is satisfied in the state where 'gly' **switches to** 'box' holds over all trajectories.[4]

Since there is no change in initial conditions of the pathway and there are no interventions, the modified domain description $\mathbf{D_1} \equiv \mathbf{D}$.

Intuitively, p is the property that holds over fluents of the transitional state s_j in which the switch takes palce, such that $gly \in T_{j-1}, box \notin T_{j-1}, gly \notin T_j, box \in T_j$ and the minimal set of firings leading up to it. The only trajectories to consider are the ones in which the observation is true. Thus the condition p is determined as the intersection of sets of fluent based conditions that were true at the time of the switch, such as:

$$\begin{aligned}
&\{sug \textbf{ has value } s_j(sug), sug \textbf{ has value higher than } 0, \dots\\
&sug \textbf{ has value higher than } s_j(sug) - 1, sug \textbf{ has value lower than } s_j(sug) + 1,\\
&fac \textbf{ has value } s_j(fac), fac \textbf{ has value higher than } 0, \dots\\
&fac \textbf{ has value higher than } s_j(fac) - 1, fac \textbf{ has value lower than } s_j(fac) + 1,\\
&acoa \textbf{ has value } s_j(acoa), acoa \textbf{ has value higher than } 0, \dots\\
&acoa \textbf{ has value higher than } s_j(acoa) - 1, acoa \textbf{ has value lower than } s_j(acoa) + 1\}
\end{aligned}$$

Simulating it for $k = 5$ steps with $ntok = 20$ max tokens, we find the condition $p = acoa$ has value greater than 0, sug has value 0, sug has value lower than 1, fac has value higher than 0$\}$. Thus, the state when this switch occurs must sugar (sug) depleted and available supply of fatty acids (fac).

Question 13 *The final protein complex in the electron transport chain of the mitochondria is non-functional. Explain the effect of this on pH of the intermembrane space of the mitochondria.*

[4] Note that this could be an LTL formula that must hold in all trajectories, but we did not add it here to keep the language simple.

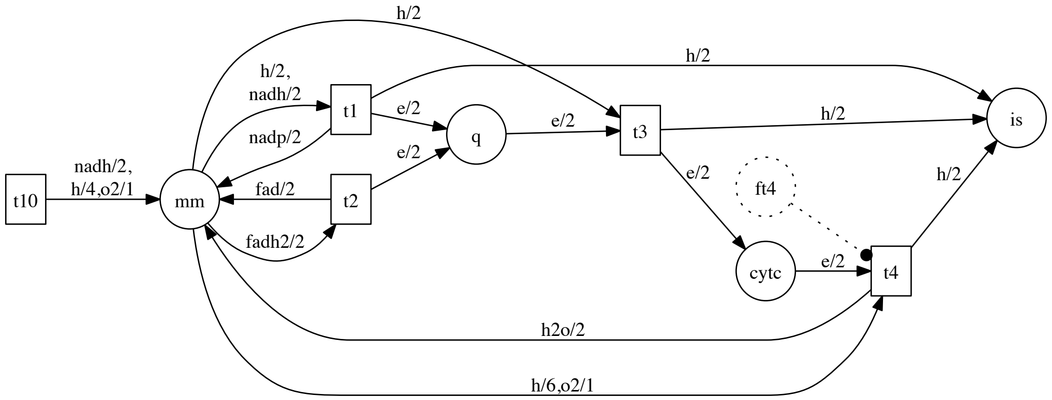


Figure 4.5: Petri Net for question 13

The question is asking for the direction of change in the pH of the intermembrane space when the nominal case is compared against a modified pathway in which the complex 4 ($t4$) is defective. Since pH is defined as $-log_{10}(H+)$, we monitor the total production of $H+$ ions to determine the change in pH value. However, since different world evolutions can follow different trajectories, we consider the average production of H+. Furthermore, we model the defective $t4$ as being unable to carry out its reaction, by disabling/inhibiting it.

Then the following pathway specification encodes the domain description **D** for question 13 and produces the PN in Fig. 4.5 without the $ft4$ place node. In the pathway description below, we have included only one *domain* declaration for all fluents as an example and left the rest out to save space. Assume integer *domain* declaration for all fluents.

domain of

nadh **atloc** *mm* **is** *integer*

$t1$ **may execute causing**

nadh **atloc** *mm* **change value by** $-2, h$ **atloc** *mm* **change value by** -2,

e **atloc** *q* **change value by** $+2, h$ **atloc** *is* **change value by** $+2$,

$nadp$ **atloc** mm **change value by** $+2$

$t2$ **may execute causing**

$fadh2$ **atloc** mm **change value by** $-2, e$ **atloc** q **change value by** $+2,$

fad **atloc** mm **change value by** $+2$

$t3$ **may execute causing**

e **atloc** q **change value by** $-2, h$ **atloc** mm **change value by** $-2,$

e **atloc** $cytc$ **change value by** $+2, h$ **atloc** is **change value by** $+2$

$t4$ **may execute causing**

$o2$ **atloc** mm **change value by** $-1, e$ **atloc** $cytc$ **change value by** $-2,$

h **atloc** mm **change value by** $-6, h2o$ **atloc** mm **change value by** $+2,$

h **atloc** is **change value by** $+2$

$t10$ **may execute causing**

$nadh$ **atloc** mm **change value by** $+2,$

h **atloc** **change value by** $+4, o2$ **atloc** mm **change value by** $+1$

initially

$fadh2$ **atloc** mm **has value** $0, e$ **atloc** mm **has value** $0,$

fad **atloc** mm **has value** $0, o2$ **atloc** mm **has value** $0,$

$h2o$ **atloc** mm **has value** $0, atp$ **atloc** mm **has value** 0

initially

$fadh2$ **atloc** is **has value** $0, e$ **atloc** is **has value** $0,$

fad **atloc** is **has value** $0, o2$ **atloc** is **has value** $0,$

$h2o$ **atloc** is **has value** $0, atp$ **atloc** is **has value** 0

initially

$fadh2$ **atloc** q **has value** $0, e$ **atloc** q **has value** $0,$

fad **atloc** q **has value** $0, o2$ **atloc** q **has value** $0,$

$h2o$ **atloc** q **has value** $0, atp$ **atloc** q **has value** 0

initially

$fadh2$ **atloc** $cytc$ **has value** $0, e$ **atloc** $cytc$ **has value** 0,

fad **atloc** $cytc$ **has value** $0, o2$ **atloc** $cytc$ **has value** 0,

$h2o$ **atloc** $cytc$ **has value** $0, atp$ **atloc** $cytc$ **has value** 0

firing style max

where mm represents the mitochondrial matrix, is represents the intermembrane space, $t1 - t4$ represent the reaction of the four complexes making up the electron transport chain, h is the $H+$ ion, $nadh$ is $NADH$, $fadh2$ is $FADH_2$, fad is FAD, e is electrons, $o2$ is oxygen O_2, atp is ATP, $h2o$ is water H_2O, and $t10$ is a source transition that supply a continuous supply of source ingredients for the chain to function, such as $nadh$, h, $o2$.

As a result, the query **Q** asked by the question is encoded as follows:

direction of change in $average$ **total production of** h **atloc** is **is** d
when observed between time step 0 **and time step** k;
comparing nominal pathway with modified pathway obtained
due to intervention $t4$ **disabled**;

Since this is a comparative quantitative query statement, it is decomposed into two queries, $\mathbf{Q_0}$ capturing the nominal case of average production w.r.t. given initial conditions:

$average$ **total production of** h **atloc** is **is** n_{avg}
when observed between time step 0 **and time step** k;

and $\mathbf{Q_1}$ the modified case w.r.t. initial conditions, modified to include interventions and subject to observations:

$average$ **total production of** h **atloc** is **is** n'_{avg}
when observed between time step 0 **and time step** k;
due to intervention $t4$ **disabled**;

The the task is to determine d, such that $\mathbf{D} \models \mathbf{Q_0}$ for some value of n_{avg}, $\mathbf{D} \models \mathbf{Q_1}$ for some value of n'_{avg}, and $n'_{avg}\ d\ n_{avg}$. To answer the sub-queries $\mathbf{Q_0}$ and $\mathbf{Q_1}$ we build nominal description $\mathbf{D_0}$ and modified pathway $\mathbf{D_1}$, where $\mathbf{D_0} \equiv \mathbf{D}$ since there are no initial conditions; and $\mathbf{D_1} \equiv \mathbf{D_0} \diamond (t4\ \textbf{disabled})$ is the domain description $\mathbf{D}$ modified according to $\mathbf{Q_1}$ to modified to include the initial conditions as well as interventions. The $\diamond$ operator modifies the domain description to its left by adding, removing or modifying pathway specification language statements to add the *intervention* and *initial conditions* to its right. Thus,

$$\mathbf{D_1} = \mathbf{D_0} + \Big\{\ \textbf{inhibit}\ t4$$

Performing a simulation of $k = 5$ steps with $ntok = 20$ max tokens, we find that the average total production of H+ in the intermembrane space (h at location is) reduces from 16 to 14. Lower quantity of H+ translates to a higer numeric value of $-log_{10}$, as a result the pH increases.

Question 14 *Membranes must be fluid to function properly. How would decreased fluidity of the membrane affect the efficiency of the electron transport chain?*

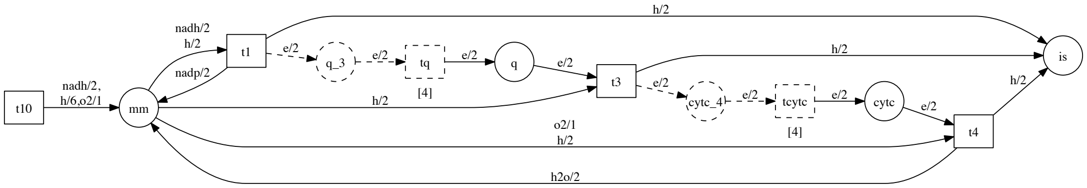


Figure 4.6: Petri Net with colored tokens alternate for question 14

From background knowledge, we know that, "Establishing the H+ gradient is a major function of the electron transport chain" (Reece *et al.*, 2010, Chapter 9), we measure the efficiency in terms of H+ ions moved to the intermembrane space (is) over time. Thus, we interpret the question is asking for the direction of change in the production of H+ moved to the intermembrane space when the nominal case is compared against a modified pathway with decreased fluidity of membrane. Additional background knowledge from Reece *et al.* (2010) tells us that the decreased fluidity reduces the speed of mobile carriers such as q and $cytc$. Fluidity can span a range of values, but we will consider one such value v per query.

The following pathway specification encodes the domain description **D** for question 14 and produces the PN in Fig. 4.6 minus the places $q\_3, cytc\_4$ and transitions $tq, tcytc$. In the pathway description below, we have included only one *domain* declaration for all fluents as an example and left the rest out to save space. Assume integer *domain* declaration for all fluents.

domain of

nadh **atloc** *mm* **is** *integer*

$t1$ **may execute causing**

nadh **atloc** *mm* **change value by** -2, *h* **atloc** *mm* **change value by** -2,

e **atloc** *q* **change value by** $+2$, *h* **atloc** *is* **change value by** $+2$,

nadp **atloc** *mm* **change value by** $+2$

$t3$ **may execute causing**

e **atloc** *q* **change value by** -2, *h* **atloc** *mm* **change value by** -2,

e **atloc** *cytc* **change value by** $+2$, *h* **atloc** *is* **change value by** $+2$

$t4$ **may execute causing**

$o2$ **atloc** *mm* **change value by** -1, *e* **atloc** *cytc* **change value by** -2,

h **atloc** *mm* **change value by** -2, $h2o$ **atloc** *mm* **change value by** $+2$,

h **atloc** *is* **change value by** $+2$

$t10$ **may execute causing**

nadh **atloc** *mm* **change value by** $+2$, *h* **atloc** *mm* **change value by** $+4$,

$o2$ **atloc** *mm* **change value by** $+1$

initially

nadh **atloc** *mm* **has value** 0, *h* **atloc** *mm* **has value** 0,

nadp **atloc** *mm* **has value** 0, *o*2 **atloc** *mm* **has value** 0

*h*2*o* **atloc** *mm* **has value** 0

initially

nadh **atloc** *is* **has value** 0, *h* **atloc** *is* **has value** 0,

nadp **atloc** *is* **has value** 0, *o*2 **atloc** *is* **has value** 0

*h*2*o* **atloc** *is* **has value** 0

initially

nadh **atloc** *q* **has value** 0, *h* **atloc** *q* **has value** 0,

nadp **atloc** *q* **has value** 0, *o*2 **atloc** *q* **has value** 0

*h*2*o* **atloc** *q* **has value** 0

initially

nadh **atloc** *cytc* **has value** 0, *h* **atloc** *cytc* **has value** 0,

nadp **atloc** *cytc* **has value** 0, *o*2 **atloc** *cytc* **has value** 0

*h*2*o* **atloc** *cytc* **has value** 0

firing style *max*

The query **Q** asked by the question is encoded as follows:

direction of change in *average* **total production of** *h* **atloc** *is* **is** *d*

when observed between time step 0 **and time step** *k*;

comparing nominal pathway with modified pathway obtained

due to interventions :

add delay of *v* **time units in availability of** *e* **atloc** *q*,

add delay of *v* **time units in availability of** *e* **atloc** *cytc*;

Since this is a comparative quantitative query statement, we decompose it into two queries, $\mathbf{Q_0}$ capturing the nominal case of average rate of production w.r.t. given initial conditions:

average **total production of** h **atloc** *is* **is** d

when observed between time step 0 **and time step** k;

and $\mathbf{Q_1}$ the modified case w.r.t. initial conditions, modified to include interventions and subject to observations:

average **total production of** h **atloc** *is* **is** d

when observed between time step 0 **and time step** k;

due to interventions :

add delay of v **time units in availability of** e **atloc** q,

add delay of v **time units in availability of** e **atloc** $cytc$;

Then the task is to determine d, such that $\mathbf{D} \models \mathbf{Q_0}$ for some value of n_{avg}, $\mathbf{D} \models \mathbf{Q_1}$ for some value of n'_{avg}, and $n'_{avg}\ d\ n_{avg}$. To answer the sub-queries $\mathbf{Q_0}$ and $\mathbf{Q_1}$ we build modified domain descriptions $\mathbf{D_0}$ and $\mathbf{D_1}$, where, $\mathbf{D_0} \equiv \mathbf{D}$ is the nominal domain description $\mathbf{D}$ modified according to $\mathbf{Q_0}$ to include the initial conditions; and $\mathbf{D_1} \equiv \mathbf{D_0} \diamond$ (**add delay of** v **time units in availability of** e **atloc** q) $\diamond$ (**add delay of** v **time units in availability of** e **atloc** $cytc$) is the modified domain description $\mathbf{D}$ modified according to $\mathbf{Q_1}$ to include the initial conditions as well as the interventions. The $\diamond$ operator modifies the domain description to its left by adding, removing or modifying pathway specification language statements to add the *interventions* and *initial conditions* description to its right. Thus,

$$\begin{array}{rl}
\mathbf{D_1} = \mathbf{D} - & \left\{\begin{array}{ll}
t1\ \textbf{may execute causing} & nadh\ \textbf{atloc}\ mm\ \textbf{change value by}\ -2, \\
 & h\ \textbf{atloc}\ mm\ \textbf{change value by}\ -2, \\
 & e\ \textbf{atloc}\ q\ \textbf{change value by}\ +2, \\
 & h\ \textbf{atloc}\ is\ \textbf{change value by}\ +2, \\
 & nadp\ \textbf{atloc}\ mm\ \textbf{change value by}\ +2 \\
t3\ \textbf{may execute causing} & e\ \textbf{atloc}\ q\ \textbf{change value by}\ -2, \\
 & h\ \textbf{atloc}\ mm\ \textbf{change value by}\ -2, \\
 & e\ \textbf{atloc}\ cytc\ \textbf{change value by}\ +2, \\
 & h\ \textbf{atloc}\ is\ \textbf{change value by}\ +2
\end{array}\right\} \\
+ & \left\{\begin{array}{ll}
t1\ \textbf{may execute causing} & nadh\ \textbf{atloc}\ mm\ \textbf{change value by}\ -2, \\
 & h\ \textbf{atloc}\ mm\ \textbf{change value by}\ -2, \\
 & e\ \textbf{atloc}\ q\_3\ \textbf{change value by}\ +2, \\
 & h\ \textbf{atloc}\ is\ \textbf{change value by}\ +2, \\
 & nadp\ \textbf{atloc}\ mm\ \textbf{change value by}\ +2 \\
t3\ \textbf{may execute causing} & e\ \textbf{atloc}\ q\ \textbf{change value by}\ -2, \\
 & h\ \textbf{atloc}\ mm\ \textbf{change value by}\ -2, \\
 & e\ \textbf{atloc}\ cytc\_4\ \textbf{change value by}\ +2, \\
 & h\ \textbf{atloc}\ is\ \textbf{change value by}\ +2 \\
tq\ \textbf{may fire causing} & e\ \textbf{atloc}\ q\_3\ \textbf{change value by}\ -2, \\
 & e\ \textbf{atloc}\ q\ \textbf{change value by}\ +2 \\
tcytc\ \textbf{may fire causing} & e\ \textbf{atloc}\ cytc\_4\ \textbf{change value by}\ -2, \\
 & e\ \textbf{atloc}\ cytc\ \textbf{change value by}\ +2 \\
tq\ \textbf{executes} & \ \textbf{in}\ 4\ \textbf{time units} \\
tcytc\ \textbf{executes} & \ \textbf{in}\ 4\ \textbf{time units}
\end{array}\right\}
\end{array}$$

Performing a simulation of k = 5 steps with ntok = 20 max tokens with a fluidity based delay of 2, we find that the average total production of H+ in the intermembrane space (h at location is) reduces from 16 to 10. Lower quantity of H+ going into the intermembrane space means lower efficiency, where we define the efficiency as the total amount of $H+$ ions transferred to the intermembrane space over the simulation run.

4.6.8 Example Encoding with Conditional Actions

Next, we illustrate how conditional actions would be encoded in our high-level language with an example. Consider the pathway from question 11. Say, the reaction step $t4$ has developed a fault, in which it has two modes of operation, in the first mode, when $f16bp$ has less than 3 units available, the reaction proceeds normally, but when $f16bp$ is available in 3 units or higher, the reaction continues to produce $g3p$ but not $dhap$ directly. $dhap$ can still be produced by subsequent step from $g3p$. The modified pathway is given in our pathway specification language below:

$t3$ **may execute causing**	$f16bp$ **change value by** $+1$	
$t4$ **may execute causing**	$f16bp$ **change value by** -1,	$dhap$ **change value by** $+1$,
	$g3p$ **change value by** $+1$	
if	$g3p$ **has value lower than** 3	
$t4$ **may execute causing**	$f16bp$ **change value by** -1,	$dhap$ **change value by** $+1$
if	$g3p$ **has value** 3 **or higher**	
$t5a$ **may execute causing**	$dhap$ **change value by** -1,	$g3p$ **change value by** $+1$
$t5b$ **may execute causing**	$g3p$ **change value by** -1,	$dhap$ **change value by** $+1$
$t6$ **may execute causing**	$g3p$ **change value by** -1,	$bpg13$ **change value by** $+2$
initially	$f16bp$ **has value** 0,	$dhap$ **has value** 0,
	$g3p$ **has value** 0,	$bpg13$ **has value** 0
firing style	max	

We ask the same question **Q**:

> **direction of change in** *average* **rate of production of** $bpg13$ **is** d
>
> **when observed between time step** 0 **and time step** k;
>
> **comparing nominal pathway with modified pathway obtained**

due to interventions : **remove** *dhap* **as soon as produced**;

using initial setup : **continuously supply** $f16bp$ **in quantity** 1;

Since this is a comparative quantitative query statement, we decompose it into two queries, $\mathbf{Q_0}$ capturing the nominal case of average rate of production w.r.t. given initial conditions:

average **rate of production of** $bpg13$ **is** n
when observed between time step 0 **and time step** k;
using initial setup : **continuously supply** $f16bp$ **in quantity** 1;

and $\mathbf{Q_1}$ the modified case w.r.t. initial conditions, modified to include interventions and subject to observations:

average **rate of production of** $bpg13$ **is** n
when observed between time step 0 **and time step** k;
due to interventions : **remove** *dhap* **as soon as produced**;
using initial setup : **continuously supply** $f16bp$ **in quantity** 1;

Then the task is to determine d, such that $\mathbf{D} \models \mathbf{Q_0}$ for some value of n_{avg}, $\mathbf{D} \models \mathbf{Q_1}$ for some value of n'_{avg}, and $n'_{avg}\ d\ n_{avg}$. To answer the sub-queries $\mathbf{Q_0}$ and $\mathbf{Q_1}$ we build modified domain descriptions $\mathbf{D_0}$ and $\mathbf{D_1}$, where, $\mathbf{D_0} \equiv \mathbf{D} \diamond$ (**continuously supply** $f16bp$ **in quantity** 1) is the nominal domain description $\mathbf{D}$ modified according to $\mathbf{Q_0}$ to include the initial conditions; and $\mathbf{D_1} \equiv \mathbf{D_0} \diamond$ (**remove** *dhap* **as soon as produced**) is the modified domain description $\mathbf{D}$ modified according to $\mathbf{Q_1}$ to include the initial conditions as well as the interventions. The $\diamond$ operator modifies the domain description to its left by adding, removing or modifying pathway specification language statements to add the *interventions* and *initial conditions* description to its right. Thus,

$$\mathbf{D_0} = \mathbf{D} + \left\{ \; tf_{f16bp} \textbf{ may execute causing } \quad f16bp \textbf{ change value by } +1 \right.$$

$$\mathbf{D_1} = \mathbf{D_0} + \left\{ \; tr \textbf{ may execute causing } \quad dhap \textbf{ change value by } * \right.$$

4.6.9 ASP Program

Next we briefly outline how the pathway specification and the query statement components are encoded in ASP (using Clingo syntax). In the following section, we will illustrate the process using an example.

As evident from the previous sections, we need to simulate non-comparative queries only. Any comparative queries are translated into non-comparative subqueries, each of which is simulated and their results compared to evaluate the comparative query.

The ASP program is a concatenation of the translation of a pathway specification (domain description which includes the firing style) and internal observations. Any initial setup conditions and interventions are pre-applied to the pathway specification using intervention semantics in section 4.6.2 before it is translated to ASP using the translation in chapter 2 as our basis. The encoded pathway specification has the semantics defined in section 4.4. Internal observations in the '*due to observations:*' portion of query statement are translated into ASP constraints using the internal observation semantics defined in section 4.6.4 and added to the encoding of the pathway specification.

The program if simulated for a specified simulation length k produces all trajectories of the pathway for the specified firing style, filtered by the internal observations. The query description specified in the query statement is then evaluated w.r.t. these trajectories. Although this part can be done in ASP, we have currently implemented it outside ASP in our implementation for ease of using floating point math.

Next, we describe an implementation of our high level language and illustrate the construction of an ASP program, its simulation, and query statement evaluation.

4.6.10 Implementation

We have developed an implementation [5] of a subset of our high level (Pathway and Query specification) language in Python. We use the Clingo ASP implementation for our simulation. In this section we describe various components of this implementation. An architectural overview of our implementation is shown in figure 4.7.

The *Pathway Specification Language (BioPathQA-PL) Parser* component is responsible for parsing the Pathway Specification Language (BioPathQA-PL). It use PLY (Python Lex-Yacc)[6] to parse a given pathway specification using grammar based on section 4.3. On a successful parse, a Guarded-Arc Petri Net pathway model based on section 4.4 is constructed for the pathway specification.

The *Query Language Parser* component is responsible for parsing the Query Specification Language (BioPathQA-QL). It uses PLY to parse a given query statement using grammar based on section 4.5. On a successful parse, an internal representation of the query statement is constructed. Elements of this internal representation include objects representing the query description, the list of interventions, the list of internal observations, and the list of initial setup conditions. Each intervention and initial setup condition object has logic in it to modify a given pathway per the intervention semantics described in section 4.6.2. The Query Statement Model component is also responsible for generating basic queries for aggregate queries and implementing interventions in the Petri Net Pathway Model.

The *Dictionary* of fluents, locations, and actions is consulted by the ASP code generator to standardize symbol names in the ASP code produced for the pathway specification and the internal observations.

[5] Implementation available at: https://sites.google.com/site/deepqa2014/

[6] http://www.dabeaz.com/ply

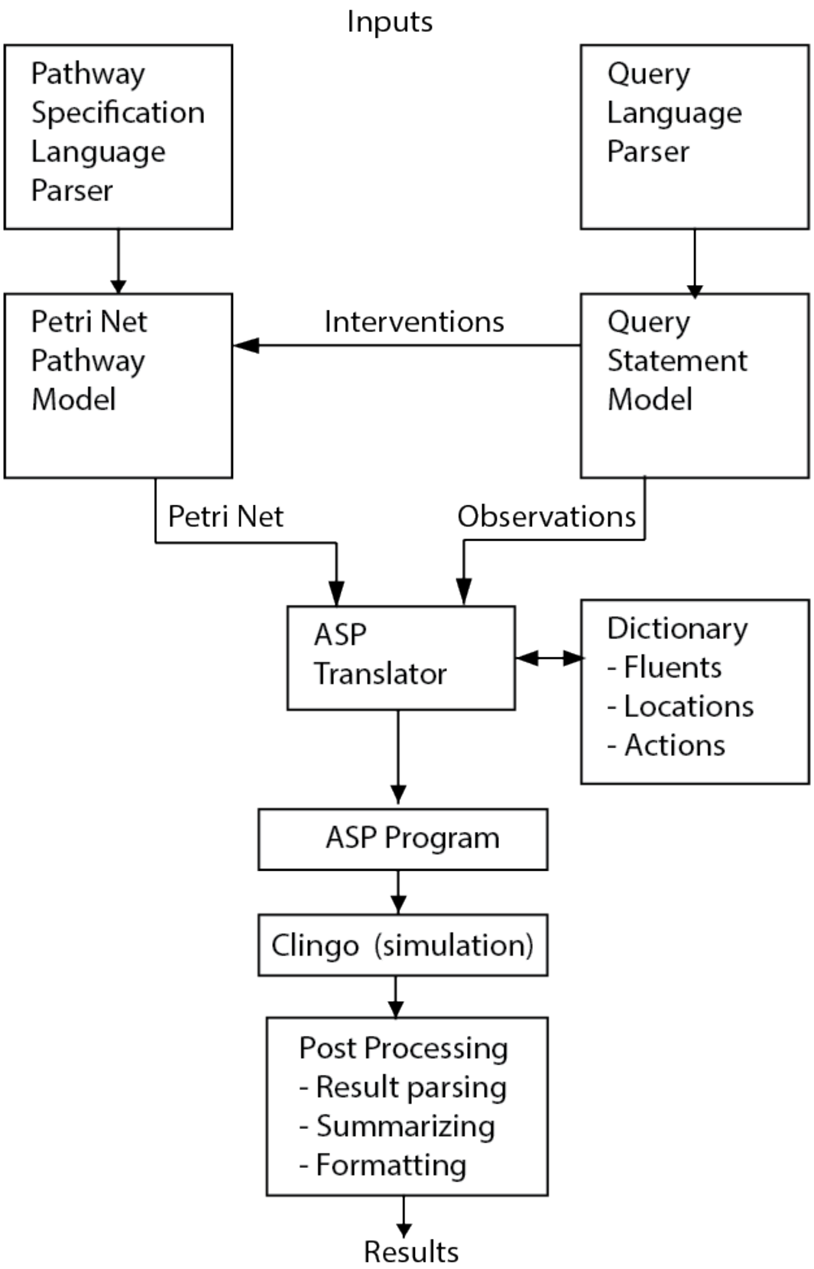


Figure 4.7: BioPathQA Implementation System Architecture

The *ASP Translator* component is responsible for translating the Guarded-Arc Petri Net model into ASP facts and rules; and the driver needed to simulate the model using the firing semantics specified in the pathway model. The code generated is based on the ASP translation of Petri Nets and its various extensions given in chapter 2. To reduce the ASP code and its complexity, the translator limits the output model to the extensions used in the Petri Net model to be translated. Thus, the colored tokens extension code is not produced unless colored tokens have been used. Similarly, guarded-arcs code is not produced if no arc-guards are used in the model.

The ASP Translator component is also responsible for translating internal observations from the Query Statement into ASP constraints to filter Petri Net trajectories based on the observation semantics in section 4.6.4. Following examples illustrate our encoding scheme. The observation 'a_1 **switches to** a_2' is encoded as a constraint using the following rules:

```
obs_1_occurred(TS+1) :- time(TS;TS+1), trans(a1;a2),
  fires(a1,TS), not fires(a2,TS),
  not fires(a1,TS+1), fires(a2,TS+1).
obs_1_occurred :- obs_1_occurred(TS), time(TS).
obs_1_had_occurred(TSS) :- obs_1_occurred(TS), TS<=TSS, time(TSS;TS).
:- not obs_1_occurred.
```

The observation 'a_1 **occurs at time step** 5' is encoded as a constraint using the following rules:

```
obs_2_occurred(TS) :- fires(a1,TS), trans(a1), time(TS), TS=5.
obs_2_occurred :- obs_2_occurred(TS), time(TS).
obs_2_had_occurred(TSS) :- obs_2_occurred(TS), TS<=TSS, time(TSS;TS).
:- not obs_2_occurred.
```

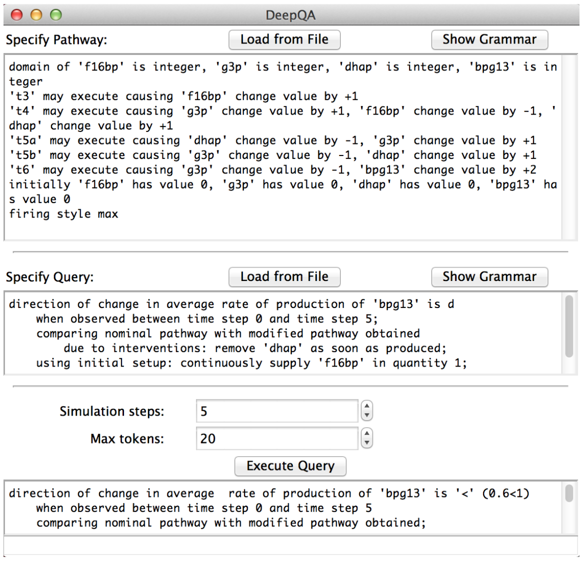
The observation 's_1 **is decreasing atloc** l_1 **when observed between time step** 0 **and time step** 5' is encoded as a constraint using the following rules:

```
%% increase in quantity quantity violates accumulation
obs_3_violated(TS) :- place(l1), col(s1),
  holds(l1,Q1,s1,TS), holds(l1,Q2,s1,TS+1),
  num(Q1;Q2), Q2 > Q1, time(TS;TS+1), TS=0, TS+1=5.
obs_3_violated :- obs_3_violated(TS), time(TS).

%% there must be a decrease in quantity at least once
obs_3_occurred(TS+1) :- not obs_3_violated,
  holds(l1,Q1,s1,TS), holds(l1,Q2,s1,TS+1), time(TS;TS+1), num(Q1;Q2),
  Q2<Q1, TS=0, TS+1=5.
obs_3_occurred :- obs_3_occurred(TS), time(TS).

obs_3_had_occurred(TSS) :- obs_3_occurred(TS), TS<=TSS, time(TSS;TS).
:- obs_3_occurred.
```

In addition, the translator is also responsible for any rules needed to ease post-processing of the query description. For example, for qualitative queries, a generic predicate `tgt_obs_occurred(TS)` is generated that is *true* when the given qualitative description holds in an answer-set at time step TS. The output of the translator is an ASP program, which when simulated using Clingo produces the (possibly) filtered trajectories of the pathway.

The *Post Processor* component is responsible for parsing the ASP answer sets, identifying the correct atoms from it, extracting quantities from atom-bodies as necessary, organizing them into a matrix form, and aggregating them as needed.

The *User Interface* component is responsible for coordinating the processing of query statement. It presents the user with a graphical user interface shown in figure 4.8. The user types a Pathway Specification (in BioPathQA-PL syntax), a Query Specification (in BioPathQA-QL syntax), and simulation parameters. On pressing "Execute Query", the user interface component processes the query as prints results



```
domain of 'f16bp' is integer, 'g3p' is integer, 'dhap' is integer, 'bpg13' is in
teger
't3' may execute causing 'f16bp' change value by +1
't4' may execute causing 'g3p' change value by +1, 'f16bp' change value by -1, '
dhap' change value by +1
't5a' may execute causing 'dhap' change value by -1, 'g3p' change value by +1
't5b' may execute causing 'g3p' change value by -1, 'dhap' change value by +1
't6' may execute causing 'g3p' change value by -1, 'bpg13' change value by +2
initially 'f16bp' has value 0, 'g3p' has value 0, 'dhap' has value 0, 'bpg13' ha
s value 0
firing style max
```

```
direction of change in average rate of production of 'bpg13' is d
    when observed between time step 0 and time step 5;
    comparing nominal pathway with modified pathway obtained
        due to interventions: remove 'dhap' as soon as produced;
    using initial setup: continuously supply 'f16bp' in quantity 1;
```

Simulation steps: 5

Max tokens: 20

```
direction of change in average  rate of production of 'bpg13' is '<' (0.6<1)
    when observed between time step 0 and time step 5
    comparing nominal pathway with modified pathway obtained;
```

Figure 4.8: BioPathQA Graphical User Interface

in the bottom box. Query evaluation differs by the type of query. We describe the query evaluation methodology used below.

For non-comparative quantitative queries:

1. Pathway specification is parsed into a Guarded-Arc Petri Net model.

2. Query statement is parsed into an internal form.

3. Initial conditions from the query are applied to the pathway model.

4. Interventions are applied to the pathway model.

5. Modified pathway model is translated to ASP.

6. Internal observations are added to the ASP code as ASP constraints.

7. Answer sets of the ASP code are computed using Clingo.

8. Relevant atoms are extracted: `fires/2` predicate for firing rate, `holds/3` (or `holds/4` – colored tokens) predicate for fluent quantity or rate formulas.

9. Fluent value or firing-count values are extracted and organized as matrices with rows representing answer-sets and columns representing time-steps.

10. Within answer-set interval or point value sub-select is done and the values converted to rates or totals as needed.

11. If aggregation, such as average, minimum, or maximum is desired, it is performed over rows of values from the last step.

12. If a value was specified in the query, it is compared against the computed aggregate for boolean result.

13. If a value was not specified, the computed value is returned as the value satisfying the query statement.

14. For queries over all trajectories, the same value must hold over all trajectories, otherwise, only one match is required to satisfy the query.

For non-comparative qualitative queries:

1. Follow steps (1)-(6) of non-comparative quantitative queries.

2. Add rules for the query description for post-processing.

3. Answer sets of the ASP code are computed using Clingo.

4. Relevant atoms are extracted: `tgt_obs_occurred/1` identifying the time step when the observation within the query description is satisfied.

5. Truth value of the query observation is determined, including determining the truth value over all trajectories.

For comparative quantitative queries:

1. Query statement is decomposed into two non-comparative quantitative sub-query statements as illustrated in section 4.6.1:

   (a) A nominal sub-query which has the same initial conditions as the comparative query, but none of its interventions or observations

   (b) A modified sub-query which has the same initial conditions, interventions, and observations as the comparative query

   both sub-query statements have the same query description, which is the non-aggregate form of the comparative query description. Thus, a comparative average rate query is translated to non-comparative average rate sub-queries.

2. Each sub-query statements is evaluated using steps (1)-(11) from the non-comparative quantitative query processing.

3. A direction of change is computed by comparing the computed aggregate value for the modified query statement to the nominal query statement.

4. If the comparative quantitative query has a direction specified, it is the compared against the computed value for a boolean result.

5. If the comparative quantitative query did not have a direction specified, the computed value is returned as the value satisfying the query statement.

For explanation queries with query description with formula of the form (4.97), it is expected that the number of answer-sets will be quite large. So, we avoid generating all answer-sets before processing them, instead we process them in-line as they are generated. It is a bit slower, but leads to a smaller memory foot print.

1. Follow steps (1)-(6) of non-comparative quantitative queries.

2. Add rules for the query description for post-processing.

3. Compute answer sets of the ASP code using Clingo.

4. Extract relevant atoms:

   (a) extract `tgt_obs_occurred/1` identifying the time step when the query description is satisfied

   (b) extract `holds/3` (or `holds/4` – for colored tokens) at the same time-step as `tgt_obs_occurred/1` to construct fluent-based conditions

5. Construct fluent-based conditions as explanation of the query observation.

6. If the query is over all trajectories, fluent-based conditions for each trajectory are intersected across trajectories to determine the minimum set of conditions explaining the query observation.

Next we illustrate query processing through an execution trace of question (11). The following shows the encoding of the base case domain, which includes the pathway specification from (4.322) with initial setup conditions from the query statement (4.323) applied:

```
%% time
#const nts=1.
time(0..nts).

%% number of tokens
#const ntok=1.
num(0..ntok).
```

```
%% places
place(bpg13).
place(dhap).
place(f16bp).
place(g3p).

%% transitions
trans(src_f16bp_1).
trans(t3).
trans(t4).
trans(t5a).
trans(t5b).
trans(t6).

%% arcs
tparc(src_f16bp_1,f16bp,1,TS) :- time(TS).

tparc(t3,f16bp,1,TS) :- time(TS).

ptarc(f16bp,t4,1,TS) :- time(TS).
tparc(t4,g3p,1,TS) :- time(TS).
tparc(t4,dhap,1,TS) :- time(TS).

ptarc(dhap,t5a,1,TS) :- time(TS).
tparc(t5a,g3p,1,TS) :- time(TS).

ptarc(g3p,t5b,1,TS) :- time(TS).
tparc(t5b,dhap,1,TS) :- time(TS).

ptarc(g3p,t6,1,TS) :- time(TS).
tparc(t6,bpg13,2,TS) :- time(TS).
```

```
%% initial state
holds(bpg13,0,0).
holds(dhap,0,0).
holds(f16bp,0,0).
holds(g3p,0,0).

%% transition inhibition guards

% firing style - 'max'
could_not_have(T,TS) :- enabled(T,TS), not fires(T,TS), ptarc(P,T,Q,TS),
    holds(P,QQ,TS), tot_decr(P,QQQ,TS), Q > QQ - QQQ.
:- not could_not_have(T,TS), time(TS), enabled(T,TS), not fires(T,TS), trans(T).


%% DRIVER

%% min(A,B,C) - C is the minimum of A and B
min(A,B,A) :- A<=B, num(A;B).
min(A,B,B) :- B<=A, num(A;B).
#hide min/3.

%% Init unspecified marking as zeroes
holdspos(P):- holds(P,N,0), place(P), num(N), N > 0.
holds(P,0,0) :- place(P), not holdspos(P).

%% Determine when a transition cannot be enabled due to normal arcs and reset arcs
notenabled(T,TS) :- ptarc(P,T,N,TS), holds(P,Q,TS), Q < N, place(P), trans(T),
    time(TS), num(N), num(Q).
```

```
%% Determine when a transition can be enabled
enabled(T,TS) :- trans(T), time(TS), not notenabled(T, TS).

%%  An enabled transition may fire subject to certain conditions
{ fires(T,TS) } :- enabled(T,TS), trans(T), time(TS).

%% Impact of individual firings
add(P,Q,T,TS) :- fires(T,TS), tparc(T,P,Q,TS), time(TS).
del(P,Q,T,TS) :- fires(T,TS), ptarc(P,T,Q,TS), time(TS).

%% Combined impact of firings
tot_incr(P,QQ,TS) :- QQ = #sum[add(P,Q,T,TS) = Q : num(Q) : trans(T)],
     time(TS), num(QQ), place(P).
tot_decr(P,QQ,TS) :- QQ = #sum[del(P,Q,T,TS) = Q : num(Q) : trans(T)],
    time(TS), num(QQ), place(P).

%% Next state computation
holds(P,Q,TS+1) :- holds(P,Q1,TS), tot_incr(P,Q2,TS), tot_decr(P,Q3,TS),
    Q=Q1+Q2-Q3, place(P), num(Q;Q1;Q2;Q3), time(TS), time(TS+1).

%% Cannot fire transitions that can together consume more than available number
%% of tokens
consumesmore(P,TS) :- holds(P,Q,TS), tot_decr(P,Q1,TS), Q1 > Q.
consumesmore :- consumesmore(P,TS).
:- consumesmore.
```

The following shows encoding of the alternate case domain, which consists of the pathway specification from (4.322) with initial setup conditions and interventions applied; and any internal observations from the query statement (4.323) added:

```
%% time
#const nts=1.
time(0..nts).
```

```
%% number of tokens
#const ntok=1.
num(0..ntok).

%% places
place(bpg13).
place(dhap).
place(f16bp).
place(g3p).

%% transitions
trans(reset_dhap_1).
trans(src_f16bp_1).
trans(t3).
trans(t4).
trans(t5a).
trans(t5b).
trans(t6).

%% arcs
ptarc(dhap,reset_dhap_1,Q,TS) :- holds(dhap,Q,TS), Q>0, time(TS).
:- enabled(reset_dhap_1,TS), not fires(reset_dhap_1,TS), time(TS).

tparc(src_f16bp_1,f16bp,1,TS) :- time(TS).

tparc(t3,f16bp,1,TS) :- time(TS).

ptarc(f16bp,t4,1,TS) :- time(TS).
tparc(t4,g3p,1,TS) :- time(TS).
tparc(t4,dhap,1,TS) :- time(TS).
```

```
ptarc(dhap,t5a,1,TS) :- time(TS).
tparc(t5a,g3p,1,TS) :- time(TS).

ptarc(g3p,t5b,1,TS) :- time(TS).
tparc(t5b,dhap,1,TS) :- time(TS).

ptarc(g3p,t6,1,TS) :- time(TS).
tparc(t6,bpg13,2,TS) :- time(TS).

%% initial state
holds(bpg13,0,0).
holds(dhap,0,0).
holds(f16bp,0,0).
holds(g3p,0,0).

%% transition inhibition guards

% firing style - 'max'
could_not_have(T,TS) :- enabled(T,TS), not fires(T,TS), ptarc(P,T,Q,TS),
    holds(P,QQ,TS), tot_decr(P,QQQ,TS), Q > QQ - QQQ.
:- not could_not_have(T,TS), time(TS), enabled(T,TS), not fires(T,TS), trans(T).


%% DRIVER

%% min(A,B,C) - C is the minimum of A and B
min(A,B,A) :- A<=B, num(A;B).
min(A,B,B) :- B<=A, num(A;B).
#hide min/3.
```

```
%% Init unspecified marking as zeroes
holdspos(P):- holds(P,N,0), place(P), num(N), N > 0.
holds(P,0,0) :- place(P), not holdspos(P).

%% Determine when a transition cannot be enabled due to normal arcs and reset arcs
notenabled(T,TS) :- ptarc(P,T,N,TS), holds(P,Q,TS), Q < N, place(P), trans(T),
    time(TS), num(N), num(Q).

%% Determine when a transition can be enabled
enabled(T,TS) :- trans(T), time(TS), not notenabled(T, TS).

%%  An enabled transition may fire subject to certain conditions
{ fires(T,TS) } :- enabled(T,TS), trans(T), time(TS).

%% Impact of individual firings
add(P,Q,T,TS) :- fires(T,TS), tparc(T,P,Q,TS), time(TS).
del(P,Q,T,TS) :- fires(T,TS), ptarc(P,T,Q,TS), time(TS).

%% Combined impact of firings
tot_incr(P,QQ,TS) :- QQ = #sum[add(P,Q,T,TS) = Q : num(Q) : trans(T)],
     time(TS), num(QQ), place(P).
tot_decr(P,QQ,TS) :- QQ = #sum[del(P,Q,T,TS) = Q : num(Q) : trans(T)],
    time(TS), num(QQ), place(P).

%% Next state computation
holds(P,Q,TS+1) :- holds(P,Q1,TS), tot_incr(P,Q2,TS), tot_decr(P,Q3,TS),
    Q=Q1+Q2-Q3, place(P), num(Q;Q1;Q2;Q3), time(TS), time(TS+1).

%% Cannot fire transitions that can together consume more than available number
%% of tokens
consumesmore(P,TS) :- holds(P,Q,TS), tot_decr(P,Q1,TS), Q1 > Q.
```

```
consumesmore :- consumesmore(P,TS).
:- consumesmore.
```

Both programs are simulated for 5 time-steps and 20 max tokens using the following Clingo command:

```
clingo 0 -cntok=20 -cnts=5 program.lp
```

Answer sets of the base case are as follows:

```
Answer: 1
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5a,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,2,3)
fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5a,3) fires(t5b,3) fires(t6,3)
holds(bpg13,2,4) holds(dhap,3,4) holds(f16bp,5,4) holds(g3p,2,4)
fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5a,4) fires(t5b,4) fires(t6,4)
holds(bpg13,4,5) holds(dhap,4,5) holds(f16bp,6,5) holds(g3p,2,5)
fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5a,5) fires(t5b,5) fires(t6,5)
Answer: 2
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5a,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,2,3)
fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5a,3) fires(t5b,3) fires(t6,3)
holds(bpg13,4,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,2,4)
fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5a,4) fires(t5b,4) fires(t6,4)
```

```
holds(bpg13,6,5) holds(dhap,3,5) holds(f16bp,6,5) holds(g3p,2,5)
fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5a,5) fires(t5b,5) fires(t6,5)
```

Answer sets of the alternate case are as follows:

```
Answer: 1
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)
holds(bpg13,2,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,2,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 2
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)
holds(bpg13,2,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,2,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
Answer: 3
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
```

```
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)

holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)

fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)

holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)

fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)

holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)

fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)

holds(bpg13,2,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)

fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)

holds(bpg13,4,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)

fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)

Answer: 4

holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)

fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)

holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)

fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)

holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)

fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)

holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)

fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)

holds(bpg13,2,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)

fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)

holds(bpg13,4,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)

fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)

Answer: 5

holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)

fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)

holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)

fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)

holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)

fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
```

```
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)
holds(bpg13,0,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,0,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 6
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)
holds(bpg13,0,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,0,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
Answer: 7
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)
holds(bpg13,0,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)
holds(bpg13,2,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)
```

```
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 8
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t5b,3)
holds(bpg13,0,4) holds(dhap,2,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)
holds(bpg13,2,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
Answer: 9
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,2,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,2,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 10
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
```

```
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,2,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,2,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
Answer: 11
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,2,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)
holds(bpg13,4,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 12
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t5b,2)
holds(bpg13,0,3) holds(dhap,2,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
```

```
holds(bpg13,2,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)
holds(bpg13,4,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
Answer: 13
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,4,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,4,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
Answer: 14
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,4,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t5b,4)
holds(bpg13,4,5) holds(dhap,2,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 15
```

```
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,4,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)
holds(bpg13,6,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t5b,5)
Answer: 16
holds(bpg13,0,0) holds(dhap,0,0) holds(f16bp,0,0) holds(g3p,0,0)
fires(reset_dhap_1,0) fires(src_f16bp_1,0) fires(t3,0)
holds(bpg13,0,1) holds(dhap,0,1) holds(f16bp,2,1) holds(g3p,0,1)
fires(reset_dhap_1,1) fires(src_f16bp_1,1) fires(t3,1) fires(t4,1)
holds(bpg13,0,2) holds(dhap,1,2) holds(f16bp,3,2) holds(g3p,1,2)
fires(reset_dhap_1,2) fires(src_f16bp_1,2) fires(t3,2) fires(t4,2) fires(t6,2)
holds(bpg13,2,3) holds(dhap,1,3) holds(f16bp,4,3) holds(g3p,1,3)
fires(reset_dhap_1,3) fires(src_f16bp_1,3) fires(t3,3) fires(t4,3) fires(t6,3)
holds(bpg13,4,4) holds(dhap,1,4) holds(f16bp,5,4) holds(g3p,1,4)
fires(reset_dhap_1,4) fires(src_f16bp_1,4) fires(t3,4) fires(t4,4) fires(t6,4)
holds(bpg13,6,5) holds(dhap,1,5) holds(f16bp,6,5) holds(g3p,1,5)
fires(reset_dhap_1,5) fires(src_f16bp_1,5) fires(t3,5) fires(t4,5) fires(t6,5)
```

Atoms selected for $bpg13$ quantity extraction for the nominal case:

```
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,2,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,4,4),holds(bpg13,6,5)]
```

Atoms selected for *bpg*13 quantity extraction for the modified case:

```
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,2,4),holds(bpg13,2,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,2,4),holds(bpg13,2,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,2,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,2,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,0,4),holds(bpg13,0,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,0,4),holds(bpg13,0,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,0,4),holds(bpg13,2,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,0,4),holds(bpg13,2,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,2,4),holds(bpg13,2,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,2,4),holds(bpg13,2,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,2,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,0,3),holds(bpg13,2,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,4,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,4,4),holds(bpg13,4,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,4,4),holds(bpg13,6,5)],
[holds(bpg13,0,0),holds(bpg13,0,1),holds(bpg13,0,2),
holds(bpg13,2,3),holds(bpg13,4,4),holds(bpg13,6,5)]
```

Progression from raw matrix of *bpg*13 quantities in various answer-sets (rows) at various simulation steps (columns) to rates and finally to the average aggregate for the nominal case:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 2 & 4 & 6 \end{bmatrix} = rate => \begin{bmatrix} 0.8 \\ 1.2 \end{bmatrix} = average => 1.0$$

Progression from raw matrix of *bpg*13 quantities in various answer-sets (rows) at various simulation steps (columns) to rates and finally to the average aggregate for the modified case:

$$\begin{bmatrix} 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 & 4 \\ 0 & 0 & 0 & 2 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 2 & 4 & 4 \\ 0 & 0 & 0 & 2 & 4 & 4 \\ 0 & 0 & 0 & 2 & 4 & 6 \\ 0 & 0 & 0 & 2 & 4 & 6 \end{bmatrix} = rate => \begin{bmatrix} 0.4 \\ 0.4 \\ 0.8 \\ 0.8 \\ 0. \\ 0. \\ 0.4 \\ 0.4 \\ 0.4 \\ 0.4 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 1.2 \\ 1.2 \end{bmatrix} = average => 0.6$$

We find that $d ='<'$ comparing the modified case rate of 0.6 to the nominal case rate of 1.0. Since the direction d was an unknown in the query statement, our system generates produces the full query specification with d replaced by $' <'$ as follows:

```
direction of change in average  rate of production of 'bpg13' is '<' (0.6<1)
    when observed between time step 0 and time step 5
    comparing nominal pathway with modified pathway obtained;
    due to interventions:
        remove 'dhap' as soon as produced;
    using initial setup:
        continuously supply 'f16bp' in quantity 1;
```

4.6.11 Evaluation Methodolgy

A direct comparison against other tools is not possible, since most other programs explore one state evolution, while we explore all possible state evolutions. In addition ASP has to ground the program completely, irrespective of whether we are computing one answer or all. So, to evaluate our system, we compare our results for the questions from the 2nd Deep KR Challenge against the answers they have provided. Our results in essence match the responses given for the questions.

4.7 Related Work

In this section, we relate our high level language with other high level action languages.

4.7.1 Comparison with π-Calculus

π-calculus is a formalism that is used to model biological systems and pathways by modeling biological systems as mobile communication systems. We use the biological model described by Regev (2001) for comparison against our system. In their model they represent molecules and their domains as computational processes, interacting elements of molecules as communication channels (two molecules interact if they fit together as in a lock-and-key mechanism), and reactions as communication through channel transmission.

- π-calculus models have the ability of changing their structure during simulation. Our system on the other hand only allows modification of the pathway at the start of simulation.

- Regular π-calculus models appear qualitative in nature. However, stochastic extensions allow representation of quantitative data (Priami *et al.*, 2001). In contrast, the focus of our system is on the quantitative+qualitative representation using numeric fluents.

- It is unclear how one can easily implement maximal-parallelism of our system in π-calculus, where a maximum number of simultaneous actions occur such that they do not cause a conflict. Where, a set of actions is said to be in conflict if their simultaneous execution will cause a fluent to become negative.

4.7.2 Comparison with Action Language $\mathcal{A}$

Action language $\mathcal{A}$ (Gelfond and Lifschitz, 1993) is a formalism that has been used to model biological systems and pathways. First we give a brief overview of $\mathcal{A}$ in an intuitive manner. Assume two sets of disjoint symbols containing fluent names and action names, then a *fluent expression* is either a fluent name F or $\neg F$. A domain description is composed of propositions of the following form:

value proposition: F **after** $A_1; \dots ; A_m$, where $(m \geq 0)$, F is a fluent and $A_1, \dots , A_m$ are fluents.

effect propostion: A **causes** F **if** $P_1, \ldots, P_n$, where $(n \geq 0)$, A is an action, $F, P_1, \ldots, P_n$ are fluent expressions. $P_1, \ldots, P_n$ are called preconditions of A and the effect proposition describes the effect on F.

We relate it to our work:

- Fluents are boolean. We support numeric valued fluents, with binary fluents.
- Fluents are non-inertial, but inertia can be added. Our fluents are always intertial.
- Action description specifies the effect of an action. Our domain description specifies 'natural'-actions, which execute automatically when their pre-conditions are satisfied, subject to certain conditions. As a result our domain description represents trajectories.
- No built in support for aggregates exists. We support a selected set of aggregates, on a single trajectory and over multiple trajectories.
- Value propositions in $\mathcal{A}$ are representable as observations in our query language.

4.7.3 Comparison with Action Language $\mathcal{B}$

Action language $\mathcal{B}$ extends $\mathcal{A}$ by adding static causal laws, which allows one to specify indirect effects or ramifications of an action (Gelfond and Lifschitz, 2012). We relate it to our work below:

- Inertia is built into the semantics of $\mathcal{B}$ (Gelfond and Lifschitz, 2012). Our language also has intertia built in.
- $\mathcal{B}$ supports static causal laws that allow defining a fluent in terms of other fluents. We do not support static causal laws.

4.7.4 Comparison with Action Language $\mathcal{C}$

Action language $\mathcal{C}$ is based on the theory of causal explanation, i.e. a formula is true if there is a cause for it to be true (Giunchiglia and Lifschitz, 1998). It has been previously used to represent and reason about biological pathways (Dworschak *et al.*, 2008). We relate it to our work below:

- $\mathcal{C}$ supports boolean fluents only. We support numeric valued fluents, and binary fluents.
- $\mathcal{C}$ allows both inertial and non-inertial fluents. While our fluents are always inertial.
- $\mathcal{C}$ support static causal laws (or ramifications), that allow defining a fluent in terms of other fluents. We do not support them.
- $\mathcal{C}$ describes causal relationships between fluents and actions. Our language on the other hand describes trajectories.

4.7.5 Comparison with Action Language $\mathcal{C}+$

First, we give a brief overview of $\mathcal{C}+$ (Giunchiglia *et al.*, 2004). Intuitively, atoms $\mathcal{C}+$ are of the form $c = v$, where c is either a fluent or an action constant, v belongs to the domain of c, and fluents and actions form a disjoint set. A formula is a propositional combination of atoms. A fluent formula is a formula in which all constants are fluent constants; and an action formula is a formula with one action constant but no fluent constants. An action description in $\mathcal{C}+$ is composed of causal laws of the following forms:

static law: **caused** F **if** G, where F and G are fluent formulas *action dynamic law*: **caused** F **if** G, where F is an action formula and G is a formula

fluent dynamic law: **caused** F **if** G **after** H, where F and G are fluent formulas and H is a formula

Concise forms of these laws exist, e.g. '**intertial** $f \equiv$ **caused** $f = v$ **if** $f = v$ **after** $f = v, \forall v \in$ domain of f' that allow a more intuitive program declaration.

We now relate it to our work.

- Some of the main differences include:
  - Fluents are multi-valued, fluent values can be integer, boolean or other types. We support integer and binary valued fluents only.
  - Actions are multi-valued. We do not support multi-valued actions.
  - Both inertial and non-inertial fluents are supported. In comparison we allow inertial fluents only.
  - Static causal laws are supported that allow changing the value of a fluent based on other fluents (ramifications). We do not allow static causal laws.
  - Effect of parallel actions on numeric fluents is not additive. However, the additive fluents extension (Lee and Lifschitz, 2003) adds the capability of additive fluents through new rules. The extended language, however, imposes certain restrictions on additive fluents and also restricts the domain of additive actions to boolean actions only. Our fluents are always additive.
  - Supports defaults. We do not have the same notion as defaults, but allow initial values for fluents in our domain description.
  - Action's occurrence and its effect are defined in separate statements. In our case, the action's occurrence and effect are generally combined in one statement.

- Although parallel actions are supported, it is unclear how one can concisely describe the condition implicit in our system that simultaneously occurring actions may not conflict. Two actions conflict if their simultaneous execution will cause a fluent to become negative.

- Exogenous actions seem the closest match to our *may execute* actions. However, our actions are 'natural', in that they execute automatically when their pre-conditions are satisfied, they are not explicitly inhibited, and they do not conflict. Actions conflict when their simultaneous execution will cause one of the fluents to become negative. The exogenous-style character of our actions holds when the *firing style* is '$*$'. When the firing style changes, the execution semantics change as well. Consider the following two *may execute* statements in our language:

  a_1 **may execute causing** f_1 **change value by** -5 **if** f_2 **has value** 3 **or higher**

  a_2 **may execute causing** f_1 **change value by** -3 **if** f_2 **has value** 2 **or higher**

  and two states: (i) $f_1 = 10, f_2 = 5$, (ii) $f_1 = 6, f_2 = 5$. In state (i) both a_1, a_2 can occur simultaneously (at one point) resulting in firing-choices $\{\{a_1, a_2\}, \{a_1\}, \{a_2\}, \{\}, \}$; whereas, in state (ii) only one of a_1 or a_2 can occur at one point resulting in the firing-choices: $\{\{a_1\}, \{a_2\}, \{\}, \}$ because of a conflict due to the limited amount of f_1. These firing choices apply for *firing style* '*', which allows any combination of fireable actions to occur. If the *firing style* is set to 'max', the maximum set of non-conflicting actions may fire, and the firing choices for state (i) change to $\{\{a_1, a_2\}\}$ and the firing choices for state (ii) change to $\{\{a_1\}, \{a_2\}\}$. If the *firing style* is set to '1', at most one action may fire at one point, and the firing choices for both state (i) and state (ii) reduce to $\{\{a_1\}, \{a_2\}, \{\}\}$. So, the case with '*' firing style can be represented in $\mathcal{C}+$ with exogenous actions a_1, a_2; the case with '1' firing style can be represented in $\mathcal{C}+$ with exogenous actions a_1, a_2 and a constraint requiring that both a_1, a_2 do not occur simultaneously; while the case with 'max' firing style can be represented by exogenous actions a_1, a_2 with additional action dynamic laws. They will still be subject to the conflict checking. Action dynamic laws can be used to force actions similar to our *must execute* actions.

- Specification of initial values of fluents seem possible through the query language. The *default* statement comes close, but it does not have the same notion as setting a fluent to a value once. We support specifying initial values both in the domain description as well as the query.

- There does not appear built-in support for aggregation of fluent values within the same answer set, such as sum, count, rate, minimum, maximum etc. Although some of it could be implemented using the additive fluents extension. We support a set of aggregates, such as total, and rate. Additional aggregates can be easily added.

- We support queries over aggregates (such as minimum, maximum, average) of single-trajectory aggregates (such as total, and rate etc.) over a set of trajectories. We also support comparative queries over two sets of trajectories. Our queries allow modification of the domain description as part of query evaluation.

4.7.6 Comparison with $\mathcal{BC}$

Action language $\mathcal{BC}$ combines features of $\mathcal{B}$ and $\mathcal{C}+$ (Lee *et al.*, 2013). First we give a brief overview of $\mathcal{BC}$.

Intuitively, $\mathcal{BC}$ has actions and valued fluents. A valued fluent, called an atom, is of the form '$f = v$', where f is a fluent constant and $v \in domain(f)$. A fluent can be regular or statically determined. An action description in $\mathcal{BC}$ is composed of static and dynamic laws of the following form:

static law: A_0 **if** $A_1, \ldots, A_m$ **ifcons** $A_{m+1}, \ldots, A_n$, where $(n \geq m \geq 0)$, and each A_i is an atom.

dynamic law: A_0 **after** $A_1, \ldots, A_m$ **ifcons** $A_{m+1}, \ldots, A_n$, where $(n \geq m \geq 0)$, A_0 is a regular fluent atom, each of $A_1, \ldots, A_m$ is an atom or an action constant, and $A_{m+1}, \ldots, A_n$ are atoms.

Concise forms of these laws exist that allow a more intuitive program declaration.

Now we relate it to our work.

- Some of the main differences include:
  - Fluents are multi-valued, fluent values can which can be integer, boolean, or other types. We only support integer and binary fluents.
  - Static causal laws are allowed. We do not support static causal laws.
  - Similar to $\mathcal{C}+$ numeric fluent accumulation is not supported. It is supported in our system.
  - It is unclear how aggregate queries within a trajectory can be concisely represented. Aggregate queries such as rate are supported in our system.
  - It does not seem that queries over multiple trajectories or sets of trajectories are supported. Such queries are supported in our system.

4.7.7 Comparison with ASPMT

ASPMT combines ASP with Satisfiability Modulo Theories. We relate the work in Lee and Meng (2013) where $\mathcal{C}+$ is extended using ASPMT with our work.

- It adds support for real valued fluents to $\mathcal{C}+$ including additive fluents. Thus, it allows reasoning with continuous and discrete processes simultaneously. Our language does not support real numbers directly.

Several systems also exist to model and reason with biological pathway. For example:

4.7.8 Comparison with BioSigNet-RR

BioSigNet-RR (Baral *et al.*, 2004) is a system for representing and reasoning with signaling networks. We relate it to our work in an intuitive manner.

- Fluents are boolean, so qualitative queries are possible. We support both integer and binary fluents, so quantiative queries are also possible.
- Indirect effects (or ramifications) are supported. We do not support these.
- Action effects are captured separately in ' **causes** statement' from action triggering statements ' **triggers** ' and ' **n_triggers** '. We capture both components in a ' **may execute causing** ' or ' **normally must execute causing** ' statement.
- Their action triggering semantics have some similarity to our actions. Just like their actions get triggered when the action's pre-conditions are satisfied, our actions are also triggered when their pre-conditions are satisfied. However, the triggering semantics are different, e.g. their **triggers** statement causes an action to occur even if it is disabled, we do not have an equivalent for it; and their **n_triggers** is similar in semantics to *normally must execute causing* statement.
- It is not clear how loops in biological systems can be modeled in their system. Loops are possible in our by virtue of the Petri Net semantics.
- Their queries can have time-points and their precedence relations as part of the query. Though our queries allow the specification of some time points for

interval queries, time-points are not supported in a similar way. However, we do support certain types of observation relative queries.

- The intervention in their planning queries has similarities to interventions in our system. However, it appears that our intervention descriptions are higher level.

4.8 Conclusion

In this chapter we presented the BioPathQA system and the languages to represent and query biological pathways. We also presented a new type of Petri Net, the so called Guarded-Arc Petri Net that is used as a model behind our pathway specification language, which shares certain aspects with CPNs (Jensen *et al.*, 2007), but our semantics for reset arcs is different, and we allow must-fire actions that prioritize actions for firing over other actions. We also showed how the system can be applied to questions from college level text books that require deeper reasoning and cannot be answered by using surface level knowledge. Although our system is developed with respect to the biological domain, it can be applied to non-biological domain as well.

Some of the features of our language include: natural-actions that automatically fire when their prerequisite conditions are met (subject to certain restrictions); an automatic default constraint that ensures fluents do not go negative, since they model natural systems substances; a more natural representation of locations; and control of the level of parallelism to a certain degree. Our query language also allows interventions similar to Pearl's surgeries (Pearl, 1995), which are more general than actions.

Next we want to apply BioPathQA to a real world application by doing text extraction. Knowledge for real world is extracted from research papers. In the next chapter we show how such text extraction is done for pathway construction and drug development. We will then show how we can apply BioPathQA to the extracted knowledge to answer questions about the extracted knowledge.

Chapter 5

TEXT EXTRACTION FOR REAL WORLD APPLICATIONS

In the previous chapter we looked at the BioPathQA system and how it answers simulation based reasoning questions about biological pathways, including questions that require comparison of alternate scenarios through simulation. These so called 'what-if' questions arise in biological activities such as drug development, drug interaction, and personalized medicine. We will now put our system and language in context of such activities.

Cutting-edge knowledge about pathways for activities such as drug development, drug interaction, and personalized medicine comes in the form of natural language research papers, thousands of which are published each year. To use this knowledge with our system, however, we need to perform extraction. In this chapter we describe techniques we use for such knowledge extraction for discovering drug interactions. We illustrate with an example extraction how we organize the extracted knowledge into a pathway specification and give examples of relevant what-if questions that a researcher performing may ask in the drug development domain.

5.1 Introduction

Thousands of research papers are published each year about biological systems and pathways covering a broad spectrum of activities, including interactions between dugs and diseases, the associated pathways, and genetic variation. In order to use this knowledge in a computer model, one has to perform text extraction to extract relationships between the biochemical processes, their involvement in diseases, and their interaction with drugs. For personalized medicine, one also needs to extract how these interrelationships change in the presence of genetic variation. In short, we are looking to extract relationships between various components of the biochemical processes and their internal and external stimuli. These component relationships can be combined with background domain knowledge to produce pathway specifications.

Many approaches exist for extracting relationships from a document. Most rely on some form of co-occurrence, relative distance, or order of words in a single document. Some use shallow parsing as well. Although these techniques tend to have a higher recall, they focus on extracting explicit relationships, which are relationships that are fully captured in a sentence or a document. Such techniques generally do not extract relationships that are spread across multiple documents. Many do not handle negative statements either. Additional issues arise due to the level of detail in older vs. newer texts and seemingly contradictory information due to various levels of confidence in the techniques used.

We primarily use a system called Parse Tree Query Language (PTQL) (Tari *et al.*, 2012) to extract such relationships. PTQL allows combining the syntactic structure (parse tree), semantic dependencies, and word order in a relationship query. The sentences are pre-processed by using named-entity recognition, entity normalization, part-of-speech tagging, and semantic linking ; and stored in a database, called the parse tree database. This preprocessing allows querying on classes of entity types, such as drugs, and diseases; and also to allow cross-linking relationships across documents when they refer to the same entity with a different name. Queries that use such semantic association between words/phrases are likely to produce higher precision results. Our source knowledge for such extraction primarily comes from thousands of biological abstracts published each year in PubMed [1]. PTQL has been used in the

[1] http://www.ncbi.nlm.nih.gov/pubmed

past to assemble bimolecular networks (Tari *et al.*, 2009) including pharmacokinetic pathways (Tari *et al.*, 2010a).

Next we briefly describe how we extract relationships about drug interactions. Following that we briefly describe how we extract association of drugs and diseases with genetic variation. We conclude this chapter with an illustrative example of how the extracted drug interaction relationships are used with our system to answer questions about drug interactions and how genetic variation could be utilized in our system.

5.2 Extracting Relationships about Drug Interactions

In this section we summarize the techniques developed for automated extraction of relationships about drug interactions from our work on extracting drug-drug interactions presented in Tari *et al.* (2010b).

Studying drug-drug interactions is a major activity in drug development. Drug interactions occur due to the interactions between the biological processes / pathways that are responsible metabolizing and transporting drugs. Metabolic processes remove a drug from the system within a certain time period. For a drug to remain effective, it must be maintained within its therapeutic window for the period of treatment, requiring taking the drug periodically. Outside the therapeutic window, a drug can become toxic if a quantity greater than the therapeutic window is retained; or it can become ineffective if a quantity less than the therapeutic window is retained.

Since liver enzymes metabolize most drugs, it is the location where most metabolic-interaction takes place. Induction or inhibition of these enzymes can affect the bioavailability of a drug through transcriptional regulation, either directly or indirectly. For example, if drug A inhibits enzyme E, which metabolizes drug B, then the bioavailability of drug B will be higher than normal, rendering it toxic. On the other hand, if drug A induces enzyme E, which metabolizes drug B, then drug B's bioavailability will be lesser than normal, rendering it ineffective.

Inhibition of enzymes is a common form of drug-drug interactions (Boobis, 2009). In direct inhibition, a drug A inhibit enzyme E, which is responsible for metabolism of drug B. Drug A, leads to a decrease in the level of enzyme E, which in turn can increase bioavailability of drug B potentially leading to toxicity. Alternatively, insufficient metabolism of drug B can lead to smaller amount of drug B's metabolites being produced, leading to therapeutic failure. An example of one such direct inhibition is the interaction between CYP2D6 inhibitor quinidine and CYP2D6 substrates (i.e. substances metabolized by CYP2D6), such as Codeine. The inhibition of CYP2D6 increases the bioavailability of drugs metabolized by CYP2D6 leading to adverse side effects.

Another form of drug interactions is through induction of enzymes (Boobis, 2009). In direct induction, a drug A induces enzyme E, which is responsible for metabolism of drug B. An example of such direct induction is between phenobarbital, a CYP2C9 inducer and warfarin (a CYP2C9 substrate). Phenobarbital leads to increased metabolism of warfarin, decreasing warfarins bioavailability. Direct interaction due to induction though possible is not as common as indirect interaction through transcription factors, which regulate the drug metabolizing enzymes. In such an interaction, drug A activates a transcription factor TF, which regulates and induces enzyme E, where enzyme E metabolizes drug B. Transcription factors are referred to as regulators of xenobiotic-metabolizing enzymes. Examples of such regulators include aryl hydrocarbon receptor AhR, pregnane X receptor PXR and constitutive androstane receptor CAR.

Drug interactions can also occur due to the induction or inhibition of transporters. Transporters are mainly responsible for cellular uptake or efflux (elimination) of drugs.

They play an important part in drug disposition, by transporting drugs into the liver cells, for example. Transporter-based drug interactions, however, are not as well studies as metabolism-based interactions (Boobis, 2009).

5.2.1 Methods

Extraction of drug-drug interactions from the text can either be explicit or implicit. Explicit extraction refers to extraction of drug-drug interaction mentioned within a single sentence, while implicit extraction requires extraction of biological properties of drug transport (or distribution), metabolism and elimination (or excretion) that can lead to drug-drug interaction. This type of indirect extraction combines background information about biological processes, identification of protein families and the interactions that are involved in drug metabolism. Both types of interactions are summarized in Fig 5.1.

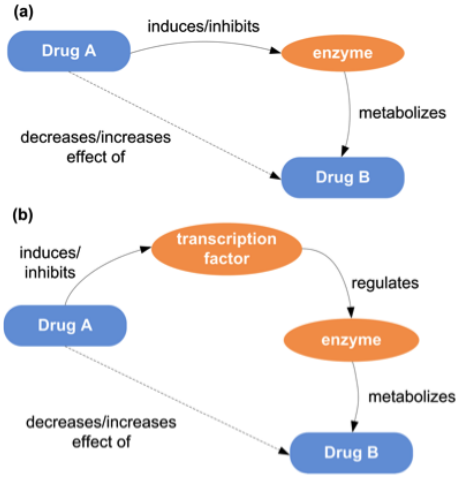


Figure 5.1: Figure from Tari *et al.* (2010b) outlining the effects of drug A on drug B through (a) direct induction/inhibition of enzymes; (b) indirect induction/inhibition of transportation factors that regulate the drug-metabolizing enzymes.

Explicit Drug Interaction Extraction

Explicit extraction mainly extracts drug-drug interactions directly mentioned in PubMed (or Medline) abstracts. For example, the following sentences each have a metabolic interaction mentioned within the same sentence:

S1: *Ciprofloxacin* strongly inhibits *clozapine* metabolism. (PMID: 19067475)

S2: Enantioselective induction of *cyclophosphamide* metabolism by *phenytoin*.

which can be extracted by using the following PTQL query using the underlined keywords from above sentences:

```
//S{//?[Tag='Drug'](d1) =>
   //?[Value IN {'induce','induces','inhibit','inhibits'}](v) =>
   //?[Tag='Drug'](d2) => //?[Value='metabolism'](w)} :::
   [d1 v d2 w] 5 : d1.value, v.value, d2.value.
```

This PTQL query specifies that a drug (denoted by d1) must be followed by one of the keywords from $\{`induce', `inhibit', `inhibits'\}$ (denoted by v), which in turn must be followed by another drug (denoted by d2) followed the keyword $`metabolism'$ (denoted by w); all found within a proximity of 5 words of each other. The query produces tripes of $\langle d1, v, d2\rangle$ values as output. Thus the results will produce triples $\langle d1, induces, d2\rangle$ and $\langle d1, inhibits, d2\rangle$ which mean that the drug d1 increases the effect of d2 (i.e. $\langle d1, increases, d2\rangle$) and decreases the effect of d2 (i.e. $\langle d1, decreases, d2\rangle$) respectively. For example, the sentence S1 above matches this PTQL query and the query will produce the triplet $\langle \text{ciprofloxacin, increases, clozapine}\rangle$.

Implicit Drug Interaction Extraction

Implicit extraction mainly extracts drug-drug interactions not yet published, but which can be inferred from published articles and properties of drug metabolism.

$\langle actor1, relation, actor2\rangle$	Description
$\langle d, metabolizes, p\rangle$	Drug *d* is metabolized by protein *p*
$\langle d, inhibits, p\rangle$	Drug *d* inhibits the activity of protein *p*
$\langle d, induces, p\rangle$	Drug *d* induces the activity of protein *p*
$\langle p1, regulates, p2\rangle$	Protein *p1* regulates the activity of protein *p2*

Table 5.1: Table showing triplets representing various properties relevant to the extraction of implicit drug interactions and their description from Tari *et al.* (2010b).

PMID	Extracted Interaction	Evidence
8689812	$\langle cyp3a4, metabolizes, lovastatin\rangle$	Lovastatin is metabolized by CYP3A4
8477556	$\langle fluoxetine, inhibits, cyp2d6\rangle$	Inhibition by fluoxetine of cytochrome P450 2D6 activity
10678302	$\langle phenytoin, induces, cyp2c\rangle$	Phenytoin induces CYP2C and CYP3A4 isoforms, but not CYP2E1
11502872	$\langle pxr, regulates, cyp2b6\rangle$	The CYP2B6 gene is directly regulated by PXR

Table 5.2: Table showing triplets representing various properties relevant to the extraction of implicit drug interactions and their evidence from Tari *et al.* (2010b).

The metabolic properties themselves have their origin in various publications. The metabolic interactions extracted from published articles and the background knowledge of properties of drug metabolism are reasoned with in an automated fashion to infer drug interactions. Table 5.1 summarizes the kinds of interactions extracted from the text for implicit interactions. Multiple PTQL queries are executed for these interactions. As an example, the following PTQL query is used to extract $\langle protein, metabolizes, drug\rangle$ triplets shown in Table 5.2:

```
//S{/?[Tag=‘Drug’](d1) =>
   //VP{//?[Value IN {‘metabolized’,‘metabolised’}](rel) =>
   //?[Tag=‘GENE’](g1)}} ::: g1.value, rel.value, d1.value
```

which specifies that the extracted triplets must have a drug (denoted by d1) followed by a verb phrase (denoted by VP) with the verb in $\{‘metabolized', ‘metabolised'\}$, followed by a gene (denoted by g1).

Data Cleaning

The protein-protein and protein-drug relationships extracted from the parse tree database need an extra step of refinement to ensure that they correspond to the known properties of drug metabolism. For instance, for a protein to metabolize a drug, the protein must be an enzyme. Similarly, for a protein to regulate an enzyme, the protein must be a transcription factor. Thus, the $\langle protein, metabolizes, drug\rangle$ facts get refined to $\langle enzyme, metabolizes, drug\rangle$ and $\langle protein, regulates, protein\rangle$ gets refined to $\langle transcriptionfactor, regulates, enzyme\rangle$ respectively. Classification of proteins is done using UniProt, the Gene Ontology (GO) and Entrez Gene summary by applying rules such as:

- A protein p is an enzyme if it belongs to a known enzyme family, such as CYP, UGT or SULT gene families; or is annotated under UniProt with the hydrolase, ligase, lyase or transferase keywords; or is listed under the *"metabolic process"* GO-term; or its Entrez Gene summary mentions key phrases like *"drug metabolism"* or roots for *"enzyme"* or *"catalyzes"*.

- A protein p is considered as a transcription factor if it is annotated with keywords transcription, transcription-regulator or activator under UniProt; or it is listed under the *"transcription factor activity"* category in GO; or its Entrez Gene summary contains the phrase *"transcription factor"*.

Additional rules are applied to remove conflicting information, such as, favoring negative extractions (such as 'P does not metabolize D') over positive extractions (such as 'P metabolizes D'). For details, see Tari *et al.* (2010b). A list of PTQL queries used for data extraction is given in the appendix.

Relations	Precision (# TP)	Recall (# FN)	F-measure
$< protein, metabolizes, drug >$	93.1% (54)	26.7% (148)	41.5%
$< drug, induces, protein >$	61.8% (42)	30.7% (95)	41.0%
$< drug, inhibits, protein >$	58.6% (99)	48.5% (105)	53.1%
$< protein, regulates, protein >$	68.7% (46)	100.0% (0)	81.4%
negation	84.4% (38)	–	–

Table 5.3: Performance of interactions extracted from Medline abstracts from Tari *et al.* (2010b). TP represents true-positives, while FN represents false-negatives

5.2.2 Results

Table 5.3 summarizes the kinds of relations extracted from the text. They were compared against a manually compiled gold standard for each type of interaction using co-occurrence queries to determine performance of our techniques.

5.3 Extracting Knowledge About Genetic Variants

The relationships extracted in the previous section apply to the population in general. The kind of drug-drug interactions that could be inferred from them will also be the general case. Current prescription practices of doctors are also rooted in the average case of drug response in patients, even though a person's drug response may actually be lower or higher than the average case. This can can lead to a prescription drug becoming ineffective or causing an adverse response. A large part of this drug response can be attributed to single nucleotide polymorphisms (SNPs). For example, the enzyme CYP2D6 has 70 known allelic variations, 50 of which are non-functional (H *et al.*, 2002). Patients with *poor metabolizer* variations may retain higher concentration of drug for typical dosage, while patients with *rapid metabolizers* may not even reach therapeutic level of drug or toxic level of drug metabolites (SF, 2009). Thus, it is important to consider the individual's genetic composition for dosage determination, especially for narrow therapeutic index drugs.

Scientists studying these variations have grouped metabolizers into categories of poor (PM), intermediate (IM), rapid (RM) and ultra-rapid metabolizers (UM) and found that for some drugs, only 20% of usual dosage was required for PM and up to 140% for UM (J *et al.*, 2001). Information about SNPs, their frequency in various population groups, their effect on genes (enzymic activity) and related data is stored in research papers, clinical studies and surveys. However, it is spread-out among them. Various databases collect this information in different forms. PharmGKB collects information such information and how it related to drug response (Thorn *et al.*, 2010). However, it is a small percentage of the total number of articles on PharmGKB, due to time consuming nature of manual curation. Thus, automated extraction is needed to keep up with the cutting edge research.

In this section, we summarize the automated extraction techniques developed from our work on associating genetic variants with drugs, diseases and adverse drug reactions presented in Hakenberg *et al.* (2012a). Our approach leverages on many existing tools.

5.3.1 Methods

Next, we describe the methods used in our extraction, including: named entity recognition, entity normalization, and relation extraction.

Named Entity Recognition

The first step in linking enzymes to their genetic variation is to identify various entities such as genes (which include proteins and enzymes), drugs, diseases, adverse drug reactions (ADRs), mutations (SNPs), RefSNPs (rs-numbers), names of alleles, populations and frequencies.

For identifying genes, we use BANNER (Leaman *et al.*, 2008) trained on BioCreative II GM training set (Krallinger *et al.*, 2008). For genotypes (genetic variations including SNPs) a combination of MutationFinder (Caporaso *et al.*, 2007) and custom components is used.

For identifying diseases, a combination of BANNER and dictionary matching is used. BANNER is trained on AZDC corpus [2] (Leaman *et al.*, 2009) and sentences from BioCreative II GM data, while, the dictionary is constructed from data from UMLS in disease related categories. The dictionary is filtered to remove unspecific or spurious disease names. For identifying ADRs a dictionary based on SIDER Side Effect Resource (Kuhn *et al.*, 2010) is used.

Drug entities are recognized using a dictionary of name brand drug names and their generics based on DrugBank (Wishart *et al.*, 2006), their synonyms and identifiers from PharmGKB, and compound and substance identifiers from PubChem. Since drug names typically do not appear in different variations, only an exact case-insensitive match is performed.

Population entities are recognized using a dictionary of terms referring to countries, regions, region's inhabitants and their ethnicities from Wikipedia, filtering occurrences observed in irrelevant phrases.

Frequency entities are recognized as numbers, percentages, and ranges co-occurring in a sentence containing any of the words "allele", "variant", "mutation", or "population". False positives are removed by removing occurrences referring to p-values, odd ratios, confidence intervals and common trigger words.

[2] http://diego.asu.edu/downloads/AZDC

Entity Normalization

Entities recognized by the entity recognition process may appear with different names in different places. Especially, genes, diseases and drugs can appear with many different names in the text. For example, “CYP2D6” can appear as “Cytochrome p450 2D6” or “P450 IID6” among others, all referring to the same enzyme (EntrezGene ID 1565). Entity normalization maps these different names to the same entity (identified by a unique identifier). This is accomplished by using dictionaries and format guidelines appropriate for the entity type, as summarized below.

Each gene is assigned a unique identifier using GNAT (Hakenberg *et al.*, 2008) by first removing modifiers (such as “isozyme”, etc.) around it and matching the resulting name against a dictionary of gene names constructed from EntrezGene. Ambiguity is resolved by considering gene’s species and matching the gene mention with gene’s annotation from a set of resources, like EntrezGene, UniProt.

Drugs and diseases/ADRs are resolved to their official IDs/terms using dictionaries constructed from DrugBank and UMLS. In case of no match, a unique term is generated based on entity name.

Genetic variations are converted to the HGVS (den Dunnen and Antonarakis, 200) recommended format (with some exceptions). Populations mentions are mapped to controlled vocabulary taking their orthographic and lexical variations into account.

Relation Extraction

Twelve type of binary relations are extracted between the detected entities as given in Table 5.4. Different methods are applied to detect different relations depending upon relation type, sentence structure and whether another method is able to extract a relation beforehand. Gene-drug, gene-disease, drug-disease are extracted using sentence based co-occurrence (possibly refined by using relation-specific keywords) due to its good precision yield for such relations. For other relations where co-occurrence does not yield sufficient precision, additional extraction methods are implemented, as summarized below.

High-confidence co-occurrence that includes keywords: Such co-occurences have the relation keywords in them. It is applied to gene-drug, gene-disease, drug-ADR, drug-disease and mutation-disease associations using keywords from PolySearch (Wishart *et al.*, 2008) augmented with our own keywords; and produces high confidence results.

Co-occurrence without keywords: Such co-occurrences do not require relationship keywords to be present. This method is used for allele-population and variant-population relationships; and has the possibility of misidentifying negative relationships. This approach is used on a sentence if the high confidence co-occurrence failed on it, and as a result produces lower confidence results.

1:n co-occurrence: Identifies relationships where a single instance entity in a sentence is associated with an entity occurring multiple times. This method is useful in identifying gene mutations, where a gene is mentioned in a sentence along with a number of its mutations.

Enumerations with matching counts: Identifies relationships in a sentence with 1-1 pairing. This method is useful in capturing alleles and their associated frequencies, e.g. "The frequencies of CYP1B1*1, *2, *3, and *4 alleles were 0.087, 0.293, 0.444, and 0.175, respectively."

Least common ancestor (LCA) sub-tree: Identifies relationships based on distance in the sentence dependency parse tree generated by Stanford parser (Klein and Manning, 2003). It is useful in relating verb to its subject and noun to its modifiers by picking the closest pair in the lowest common ancestor (dependency) sub-tree of entities. We use it to associate allele frequencies with alleles.

	Total	Unique	PharmGKB
Gene-drug	191,054	31,593	6820/3014
Gene-disease	709,987	102,881	8147/4478
Drug-disease	117,834	26,268	4343/939
Drug-adverse effect	73,696	16,569	
Gene-variant	101,477	21,704	645/516
Gene-allele	65,569	6802	146/99
Gene-RefSNP	12,881	5748	1820/1125
Allele-population	7181	1891	
Variant-population	12,897	6765	
Allele frequency	6,893	279	
Variant frequency	6,646	1654	
Population frequency	8404	144	
Drugs-populations	12,849	4388	
Drugs-alleles	6,778	1858	
Drug-RefSNP	1,161	721	
Drug-mutation	10,809	5491	
Sum	1,315,811	233,964	

Table 5.4: Unique binary relations identified between detected entities from 179,935 abstracts (Hakenberg *et al.*, 2012a).

m:n co-occurrence: This method builds associations between all pairs of entities.

Low confidence co-occurrence: This acts as the catch-all case if none of the above methods were able to identify a relationship. Although it has the lowest confidence score.

These methods were applied in order to each sentence, stoping at the first method that extracted the desired relationship. Order of these methods was determined empirically based of their precision. The order of the method used determines our confidence in the result. If none of the higher confidence methods are successful, a co-occurrence based method is used for extraction with low confidence.

5.3.2 Results

Table 5.4 summarizes the kinds of relationships extracted from PubMed abstracts. Performance of the techniques was evaluated by processing a smaller set of 3500 PubMed abstracts (found via PharmGKB relations) and checking the predictions manually both for correctness and coverage against DrugBank and PharmGKB. The techniques produced 91% of data in DrugBank and 94% in PharmGKB.

5.4 Applying BioPathQA to Drug-Drug Interaction Discovery

Now we use our BioPathQA system from chapter 4 to answer questions about drug-drug interaction using knowledge extracted from research publications using the approach in sections 5.2,5.3. We supplement the extracted knowledge with domain knowledge as needed.

Let the extracted facts be as follows:

- The drug $gefitinib$ is metabolized by CYP3A4.
- The drug *phenytoin* induces CYP3A4.

Following additional facts have been provided about a new drug currently in development:

- A new drug being developed *test_drug* is a CYP3A4 inhibitor

We show the pathway specification based on the above facts and background knowledge, then elaborate on each component:

$$\begin{aligned}&\textbf{domain of } gefitinib \textbf{ is } integer, cyp3a4 \textbf{ is } integer,\\ &\qquad phenytoin \textbf{ is } integer, test\_drug \textbf{ is } integer\end{aligned} \tag{5.1}$$

$t1$ **may execute causing**

$$gefitinib\ \textbf{change value by}\ -1,$$

$$cyp3a4\ \textbf{change value by}\ -1,$$

$$cyp3a4\ \textbf{change value by}\ +1 \tag{5.2}$$

$$\textbf{normally stimulate}\ t1\ \textbf{by factor}\ 2$$

$$\textbf{if}\ phenytoin\ \textbf{has value}\ 1\ \textbf{or higher} \tag{5.3}$$

$$\textbf{inhibit}\ t1\ \textbf{if}\ test\_drug\ \textbf{has value}\ 1\ \textbf{or higher} \tag{5.4}$$

$$\textbf{initially}\ gefitinib\ \textbf{has value}\ 20, cyp3a4\ \textbf{has value}\ 60,$$

$$phenytoin\ \textbf{has value}\ 0, test\_drug\ \textbf{has value}\ 0 \tag{5.5}$$

$$\textbf{firing style}\ max \tag{5.6}$$

Line 5.1 declares the domain of the fluents as integer numbers. Line 5.2 represents the activity of enzyme *cyp*3*a*4 as the action $t1$. Due to the enzymic action $t1$, one unit of $gefitinib$ is metabolized, and thus converted to various metabolites (not shown here). The enzymic action uses one unit of *cyp*3*a*4 as catalyst, which is used in the reaction and released afterwards. Line 5.3 represents the knowledge that *phenytoin* induces the activity of *cyp*3*a*4. From background knowledge we find out that the stimulation in the activity can be as high as 2-times (Luo *et al.*, 2002). Line 5.4 represents the knowledge that there is a new drug *test_drug* being tested that is known to inhibit the activity of *cyp*3*a*4. Line 5.5 specifies the initial distribution of the drugs and enzymes in the pathway. Assuming the patient has been given some fixed dose, say 20 units, of the medicine $gefitinib$. It also specifies there is a large 60 units quantity of *cyp*3*a*4 available to ensure reactions do not slow down due to unavailability of enzyme availability. Additionaly, the drug *phenytoin* is absent from the system and a new drug *test_drug* to be tested is not in the system either. This gives us our pathway specification. Now we consider two application scenarios for drug development.

5.4.1 Drug Administration

A patient is taking 20 units of $gefitinib$, and is being prescribed additional drugs to be co-administered. The drug administrator wants to know if there will be an interaction with $gefitinib$ if 5 units of *phenytoin* are co-administered. If there is an interaction, what will be the bioavailability of $gefitinib$ so that its dosage could be appropriately adjusted.

The first question is asking whether giving the patient 5-units of *phenytoin* in addition to the existing $gefitinib$ dose will cause a drug-interaction. It is encoded as the following query statement **Q** for a k-step simulation:

$$\begin{aligned} &\textbf{direction of change in } \textit{average} \textbf{ value of } gefitinib \textbf{ is } d \\ &\quad\textbf{when observed at time step } k; \\ &\quad\textbf{comparing nominal pathway with modified pathway obtained} \\ &\qquad\textbf{due to interventions}: \textbf{ set value of } \textit{phenytoin} \textbf{ to } 5; \end{aligned}$$

If the direction of change is "=" then there was no drug-interaction. Otherwise, an interaction was noticed. For a simulation of length $k = 5$, we find 15 units of $gefitinib$ remained at the end of simulation in the nominal case when no *phenytoin* is administered. The amount drops to 10 units of $gefitinib$ when *phenytoin* is co-administered. The change in direction is "<". Thus there is an interaction.

The second question is asking about the bioavailability of the drug $gefitinib$ after some after giving *phenytoin* in 5 units. If this bioavailability falls below the efficacy level of the drug, then the drug would not treat the disease effectively. It is encoded as the following query statement **Q** for a k-step simulation:

average **value of** *gefitinib* **is** n
when observed at time step k;
due to interventions :
set value of *phenytoin* **to** 5;

For a simulation of length $k = 5$, we find 10 units of *gefitinib* remain. A drug administrator (such as a pharmacist) can adjust the drug accordingly.

5.4.2 Drug Development

A drug manufacturer is developing a new drug *test_drug* that is known to inhibit CYP3A4 that will be co-administered with drugs *gefitinib* and *phenytoin*. He wants to determine the bioavailability of *gefitinib* over time to determine the risk of toxicity.

The question is asking about the bioavailability of the drug *gefitinib* after 10 time units after giving *phenytoin* in 5 units and the new drug *test_drug* in 5 units. If this bioavailability remains high, there is chance for toxicity due to the drug at the subsequent dosage intervals. It is encoded as the following query statement **Q** for a k-step simulation:

average **value of** *gefitinib* **is** n
when observed at time step k;
due to interventions :
set value of *phenytoin* **to** 5,
set value of *test_drug* **to** 5;

For a simulation of length $k = 5$, we find all 20 units of *gefitinib* remain. This could lead to toxicity by building high concentration of *gefitinib* in the body.

5.4.3 Drug Administration in Presence of Genetic Variation

A drug administrator wants to establish the dosage of *morphine* for a person based on its genetic profile using its bioavailability.

Consider the following facts extracted about a simplified morphine pathway:

- *codeine* is metabolized by $CYP2D6$ producing *morphine*
- $CYP2D6$ has three allelic variations
  - "*1" – (EM) effective metabolizer (normal case)
  - "*2" – (UM) ultra rapid metabolizer
  - "*9" – (PM) poor metabolizer

For simplicity, assume UM allele doubles the metabolic rate, while PM allele halves the metabolic rate of CYP2D6. Then, the resulting pathway is given by:

```
**domain of** $cyp2d6\_allele$ **is** $integer, cyp2d6$ **is** $integer$
**domain of** $codeine$ **is** $integer, morphine$ **is** $integer$
$cyp2d6\_action$ **may execute causing**
    $codeine$ **change value by** $-2, morphine$ **change value by** $+2$
    **if** $cyp2d6\_allele$ **has value equal to** $1$
$cyp2d6\_action$ **may execute causing**
    $codeine$ **change value by** $-4, morphine$ **change value by** $+4$
    **if** $cyp2d6\_allele$ **has value equal to** $2$
$cyp2d6\_action$ **may execute causing**
    $codeine$ **change value by** $-1, morphine$ **change value by** $+1$
    **if** $cyp2d6\_allele$ **has value equal to** $9$
```

initially

$codeine$ **has value** 0, $morphine$ **has value** 0,

$cyp2d6$ **has value** 20, $cyp2d6\_allele$ **has value** 1

firing style max

Then, the bioavailability of *morphine* can be determined by the following query:

$average$ **value of** $morphine$ **is** n

when observed at time step k;

due to interventions :

set value of $codeine$ **to** 20,

set value of $cyp2d6$ **to** 20,

set value of $cyp2d6\_allele$ **to** 9;

Simulation for 5 time steps reveal that the average bioavailability of *morphine* after 5 time-steps is 5 for PM (down from 10 for EM).

Although this is a toy example, it is easy to see the potential of capturing known genetic variations in the pathway and setting the complete genetic profile of a person in the intervention part of the query.

5.5 Conclusion

In this chapter we presented how we extract biological pathway knowledge from text, including knowledge about drug-gene interactions and their relationship to genetic variation. We showed how the information extracted is used to build pathway specification and illustrated how biologically relevant questions can be answered about drug-drug interaction using the BioPathQA system developed in chapter 4. Next we look at the future directions in which the research work done in this thesis can be extended.

Chapter 6

CONCLUSION AND FUTURE WORK

The field of knowledge representation and reasoning (KR) is currently one of the most active research areas. It represents the next step in the evolution of systems that know how to organize knowledge, and have the ability to intelligently respond to questions about this knowledge. Such questions could be about static knowledge or the dynamic processes.

Biological systems are uniquely positioned as role models for this next evolutionary step due to their precise vocabulary and mechanical nature. As a result, a number of recent research challenges in the KR field are focused on it. The biological field itself needs systems that can intelligently answer questions about such biological processes and systems in an automated fashion, given the large number of research papers published each year. Curating these publications is time consuming and expensive, as a result the state of over all knowledge about biological systems lags behind the cutting edge research.

An important class of questions asked about biological systems are the so called "what-if" questions that compare alternate scenarios of a biological pathway. To answer such questions, one has to perform simulation on a nominal pathway against a pathway modified due to the interventions specified for the alternate scenario. Often, this means creating two pathways (for nominal and alternate cases) and simulate them separately. This opens up the possibility that the two pathways can become out of synchronization. A better approach is to allow the user to specify the needed interventions in the query statement itself. In addition, to understand the full spread of possible outcomes, given the parallel nature of biological pathways, one must consider all possible pathway evolutions, otherwise, some outcomes may remain hidden.

If a system is to be used by biologists, it must have a simple interface, lowering the barrier of entry. Since biological pathway knowledge can arrive from different sources, including books, published articles, and lab experiments, a common input format is desired. Such a format allows specification of pathways due to automatic extraction, as well as any changes / additions due to locally available information.

A comprehensive end-to-end system that accomplish all the goals would take a natural language query along with any additional specific knowledge about the pathway as input, extract the relevant portion of the relevant pathway from published material (and background knowledge), simulate it based on the query, and generate the results in a visual format. Each of these tasks comes with its own challenges, some of which have been addressed in this thesis.

In this thesis, we have developed a system and a high level language to specify a biological pathway and answer simulation based reasoning questions about it. The high level language uses controlled-English vocabulary to make it more natural for a researcher to use directly. The high level language has two components: a pathway specification language, and a query specification language. The pathway specification language allows the user to specify a pathway in a source independent form, thus locally obtained knowledge (e.g. from lab) can be combined with automatically extracted knowledge. We believe that our pathway specification language is easy for a person to understand and encode, lowering the bar to using our system. Our pathway specification language allows conditional actions, enabling the encoding of alternate action outcomes due to genetic variation. An important aspect of our pathway specification language is that it specifies trajectories, which includes specifying the initial configuration of substances, as well as state evolution style, such as maximal firing of actions, or serialized actions etc.

Our query specification language provides a bridge between natural language questions and their formal representation. It is English-like but with precise semantics. A main feature of our query language is its support for comparative queries over alternate scenarios, which is not currently supported by any of the query languages (associated with action languages) we have reviewed. Our specification of alternate scenarios uses interventions (a general form of actions), that allow the user to modify the pathway as part of the query processing. We believe our query language is easier for a biologist to understand without requiring formal training.

To model the pathways, we use Petri Nets, which have been used in the past to model and simulate biological pathways. Petri Nets have a simple visual representation, which closely matches biological pathways; and they inherently support parallelism. We extended the Petri Nets to add features that we needed to suit our domain, e.g., reset arcs that remove all quantity of a substance as soon as it is produced, and conditional arcs that specify the conditional outcome of an action.

For simulation, we use ASP, which allowed us straight forward way to implement Petri Nets. It also gave us the ability to add extensions to the Petri Net by making local edits, implement different firing semantics, filter trajectories based on observations, and reason with the results. One of the major advantage of using Petri Net based simulation is the ability to generate all possible state evolutions, enabling us to process queries that determine the conditions when a certain observation becomes true.

Our post-processing step is done in Python, which allows strong text processing capabilities using regular expressions, as well as libraries to easy process large matrices of numbers for summarization of results.

Now we present additional challenges that need to be addressed.

6.1 Pathway Extraction

In Chapter 5 we described how we extract facts for drug-drug interaction and gene variation. This work needs to be extended to include some of the newer databases that have come online recently. This may provide us with enzyme reaction rates, and substance quantities used in various reactions. The relation extraction for pathways must also be cognizant of any genetic variation mentioned in the text.

Since the knowledge about the pathway appears in relationships at varying degree of detail, a process needs to be devised to assemble the pathway from the same level to granularity together, while also maintaining pathways at different levels of granularities.

Since pathway extraction is a time consuming task, it would be best to create a catalog of the pathways. The cataloged pathways could be manually edited by the user as necessary. Storing pathways in this way means that would have to be updated periodically, requiring merging of new knowledge into existing pathways. Manual edits would have to be identified, such that the updated pathway does not overwrite them without the user's knowledge.

6.2 Pathway Selection

Questions presented in biological texts do not explicitly mention the relevant pathway to use for answering the question. One way to address this issue is to maintain a catalog of pre-defined pathways with keywords associated with them. Such keywords can include names of the substances, names of the processes, and other relevant metadata about the pathway. The catalog can be searched to find the closest match to the query being asked.

An additional aspect in proper pathway selection is to use the proper abstraction level. If our catalog contains a pathway at different abstraction levels, the coarsest pathway that contains the processes and substances in the query should be selected. Any higher fidelity will increase the processing time and generate too much irrelevant data. Alternatively, the catalog could contain the pathway in a hierarchical form, allowing extraction of all elements of a pathway at the same depth. A common way to hierarchically organize the pathway related to our system is to have hierarchical actions, which is the approach taken by hierarchical Petri nets.

Lastly, the question may only ask about a small subsection of a much larger pathway. For better performance, it is beneficial to extract the smallest biological pathway network model that can answer the question.

6.3 Pathway Modeling

In Chapter 3, we presented our modeling of biological questions using Petri Nets and their extensions encoded in ASP. We came across concepts like allosteric regulation, inhibition of inhibition, and inhibition of activation that we currently do not model. In allosteric regulation, an enzyme is not fully enabled or disabled, the enzyme's shape changes, making it more or less active. The level of its activity depends upon concentrations of activators and inhibitors. In inhibition of inhibition, the inhibition of a reaction is itself inhibited by another inhibition; while in inhibition of activation (or stimulation), a substance inhibits the stimulation produced by a substance. Both of these appear to be actions on actions, something that Petri Nets do not allow. An alternate coding for these would have to be devised.

As more detailed information about pathways becomes available, the reactions and processes that we have in current pathways may get replaced with more detailed sub-pathways themselves. However, such refinement may not come at the same time for separate legs of the pathway. Just replacing the coarse transition with a refined transition may not be sufficient due to relative timing constraints. Hence, a hierarchical Petri Net model may need to be implemented (see Fehling (1993); Huber *et al.* (1991)).

6.4 Pathway Simulation

In Chapter 2 we presented our approach to encode Petri Nets and their extensions. We used a discrete solver called *clingo* for our ASP encoding. As the number of simulation length increases in size or larger quantities are involved, the solver slows down significantly. This is due to an increased grounding cost of the program. Incremental solving (using *iclingo*) does not help, since the program size still increases, and the increments merely delays the slow down but does not stop it.

Systems such as constraint logic solvers (such as Ostrowski and Schaub (2012)) could be used for discrete cases. Alternatively, a system developed on the ASPMT (Lee and Meng, 2013) approach could be used, since it can represent longer runs, larger quantities, and real number quantities.

6.5 Extend High Level Language

In Chapter 4 we described the BioPathQA system, the pathway specification and the query specification high level languages. As we enhance the modeling of the biological pathways, we will need to improve or extend the system as well as the high level language. We give a few examples of such extensions.

Our pathway specification language currently does not support continuous quantities (real numbers). Extending to real numbers will improve the coverage of the pathways that can be modeled. In addition, derived quantities (fluents) can be added, e.g. pH could be defined as a formula that is read-only in nature.

Certain observations and queries can be easily specified using a language such as LTL, especially for questions requiring conditions leading to an action or a state. As a result, it may be useful to add LTL formulas to the query language. We did not take this approach because it would have presented an additional non-English style syntax for the biologists.

Our substance production / consumption rates and amounts are currently tied to the fluents. In certain situations it is desirable to analyze the quantity of a substance produced / consumed by a specific action, e.g. one is interested in finding the amount of H+ ions produced by a multi-protein complex IV only.

Interventions (that are a part of the query statement) presented in this thesis are applied at the start of the simulation. Eliminating this restriction would allow applying surgeries to the pathway mid execution. Thus, instead of specifying the steady state conditions in the query statement, one could apply the intervention when a certain steady state is reached.

6.6 Result Formatting and Visualization

In Chapter 4 we described our system that answers questions specified in our high level language. At the end of its process, it outputs the final result. This output method can be enhanced by allowing to look at the progression of results in addition to the final result. This provides the biologist with the whole spread of possible outcomes. An example of such a spread is shown in Fig. 3.2 for question 1. A graphical representation of the simulation progression is also beneficial in enhancing the confidence of the biologist. Indeed many existing tools do this. A similar effect can be achieved by parsing and showing the relevant portion of the answer set.

6.7 Summary

In Chapter 1 we introduced the thesis topic and summarized specific research contributions

In Chapter 2 we introduced the foundational material of this thesis including Petri Nets and ASP. We showed how ASP could be used to encode basic Petri Nets. We also showed how ASP's elaboration tolerance and declarative syntax allows us to encode various Petri Net extensions with small localized changes. We also introduced a new firing semantics, the so called *maximal firing set semantics* to simulate a Petri Net with maximum parallel activity.

In Chapter 3 we showed how the Petri Net extensions and the ASP encoding can be used to answer simulation based deep reasoning questions. This and the work in Chapter 2 was published in Anwar *et al.* (2013a,b).

In Chapter 4 we developed a system called BioPathQA to allow users to specify a pathway and answer queries against it. We also developed a pathway specification language and a query language for this system in order to avoid the vagaries of natural language. We introduced a new type of Guarded-arc Petri Nets to model conditional actions as a model for pathway simulation. We also described our implementation developed around a subset of the pathway specification language.

In Chapter 5 we briefly described how text extraction is done to extract real world knowledge about pathways and drug interactions. We then used the extracted knowledge to answer question using BioPathQA. The text extraction work was published in Tari *et al.* (2010b,a); Hakenberg *et al.* (2012b).

REFERENCES

Anwar, S., C. Baral and K. Inoue, “Encoding higher level extensions of petri nets in answer set programming”, arXiv preprint arXiv:1306.3548 (2013a).

Anwar, S., C. Baral and K. Inoue, “Encoding petri nets in answer set programming for simulation based reasoning”, arXiv preprint arXiv:1306.3542 (2013b).

Araki, T. and T. Kasami, “Some decision problems related to the reachability problem for Petri nets”, Theoretical Computer Science **3**, 1, 85–104 (1976).

Baral, C., *Knowledge Representation, Reasoning and Declarative Problem Solving* (Cambridge University Press, 2003).

Baral, C., K. Chancellor, N. Tran, N. Tran, A. Joy and M. Berens, “A knowledge based approach for representing and reasoning about signaling networks”, Bioinformatics **20**, suppl 1, i15–i22 (2004).

Behrens, T. M. and J. Dix, “Model checking with logic based petri nets”, Pre-Proceedings of CLIMA **7**, 6–21 (2007).

Berg, J. M., J. L. Tymoczko and L. Stryer, *Biochemistry. 5th edition.* (New York: WH Freeman, 2002).

Berthomieu*, B., P.-O. Ribet and F. Vernadat, “The tool tina–construction of abstract state spaces for petri nets and time petri nets”, International Journal of Production Research **42**, 14, 2741–2756 (2004).

Best, E. and M. Koutny, “Petri net semantics of priority systems”, Theoretical Computer Science **96**, 1, 175–215 (1992).

Boobis, A., “Drug interactions”, Drug Metabolism Review **41**, 3, 486–527 (2009).

Brauer, W. and W. Reisig, “Carl Adam Petri and “Petri Nets””, Informatik-Spektrum **29**, 5, 369–374 (2006).

Burkhard, H.-D., “Ordered firing in petri nets”, Journal of Information Processing and Cybernetics **17**, 2-3, 71–86 (1980).

Calimeri, F., W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, F. Ricca and T. Schaub, “Asp-core-2 input language format”, (2013).

Camon, E., D. Barrell and V. Lee, “The gene ontology annotation (goa) database - an integrated resource of go annotations to the uniprot knowledgebase”, Silico Biology **4**, 1, 5–6 (2004).

Caporaso, J. G., W. A. B. Jr, D. A. Randolph, K. B. Cohen and L. Hunter, “Mutationfinder: A high-performance system for extracting point mutation mentions from text”, Bioinformatics **23**, 14, 1862–1865 (2007).

Chaouiya, C., "Petri net modelling of biological networks", Briefings in bioinformatics **8**, 4, 210–219 (2007).

Christensen, S. and N. D. Hansen, *Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs* (Springer, 1993).

Consortium, T. U., "Reorganizing the protein space at the universal protein resource (uniprot)", Nucleic Acids research **40** (2012).

Croft, D., G. O'Kelly, G. Wu, R. Haw, M. Gillespie, L. Matthews, M. Caudy, P. Garapati, G. Gopinath, B. Jassal *et al.*, "Reactome: a database of reactions, pathways and biological processes", Nucleic acids research **39**, suppl 1, D691–D697 (2011).

de Marneffe, M.-C. and C. D. Manning, "The stanford typed dependencies representation", Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation (2008).

den Dunnen, J. and S. Antonarakis, "Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion", Human Mutation **15**, 1, 7–12 (200).

Dworschak, S., T. Grote, A. König, T. Schaub and P. Veber, "Tools for representing and reasoning about biological models in action language c", in "Proceedings of the Twelfth International Workshop on Nonmonotonic Reasoning (NMR 2008). The University of New South Wales, Technical Report Series", pp. 94–102 (2008).

Fehling, R., "A concept of hierarchical petri nets with building blocks", in "Advances in Petri Nets 1993", pp. 148–168 (Springer, 1993).

Gebser, M., R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub and M. Schneider, "Potassco: The Potsdam answer set solving collection", aicom **24**, 2, 105–124 (2011).

Gebser, M., T. Schaub and S. Thiele, "Gringo: A new grounder for answer set programming", in "Logic Programming and Nonmonotonic Reasoning", pp. 266–271 (Springer, 2007).

Gelfond, M. and V. Lifschitz, "The stable model semantics for logic programming", Logic Programming: Proceedings of the Fifth International Conference and Symposium pp. 1070–1080 (1988).

Gelfond, M. and V. Lifschitz, "Representing action and change by logic programs", The Journal of Logic Programming **17**, 2, 301–321 (1993).

Gelfond, M. and V. Lifschitz, "Action languages", Electronic Transactions on AI **3**, 16 (1998).

Gelfond, M. and V. Lifschitz, "The common core of action languages B and C", in "Proceedings of the 14th International Workshop on Non-Monotonic Reasoning (NMR 2012)", (2012).

Giunchiglia, E., J. Lee, V. Lifschitz, N. McCain and H. Turner, "Nonmonotonic causal theories", Artificial Intelligence **153**, 1, 49–104 (2004).

Giunchiglia, E. and V. Lifschitz, "An action language based on causal explanation: Preliminary report", in "AAAI/IAAI", pp. 623–630 (Citeseer, 1998).

H, W., R. T, H. R, B. K, B. M and W. J, "Increased frequency of cytochrome p450 2d6 poor metabolizers among patients with metoprolol-associcated adverse effects", Pharmacogenomics in Clinical Therapeutics **72**, 429–437 (2002).

Hakenberg, J., C. Plake, R. Leaman, M. Schroeder and G. Gonzalez, "Inter-species normalization of gene mentions", Bioinformatics **24**, 16, 126–132 (2008).

Hakenberg, J., D. Voronov, V. H. Nguyên, S. Liang, S. Anwar, B. Lumpkin, R. Leaman, L. Tari and C. Baral, "A snpshot of pubmed to associate genetic variants with drugs, diseases, and adverse reactions", Journal of biomedical informatics **45**, 5, 842–850 (2012a).

Hakenberg, J., D. Voronov, V. H. Nguyen, S. Liang, B. Lumpkin, S. Anwar, R. Leaman, L. Tari and C. Baral, "Taking a snpshot of pubmed - a repository of genetic variants and their drug response phenotypes", Journal of Biomedical Informatics (2012b).

Hardy, S. and P. N. Robillard, "Modeling and simulation of molecular biology systems using petri nets: Modeling goals of various approaches", Journal of Bioinformatics and Computational Biology **2**, 4, 619–637 (2004).

Harrison, A., V. Lifschitz and F. Yang, "On the semantics of gringo", arXiv preprint arXiv:1312.6149 (2013).

Heiner, M., I. Koch and J. Will, "Model validation of biological pathways using petri nets—demonstrated for apoptosis", Biosystems **75**, 1, 15–28 (2004).

Heljanko, K. and I. Niemelä, "Petri net analysis and nonmonotomic reasoning", Leksa Notes in Computer Science pp. 7–19 (2000).

Huber, P., K. Jensen and R. M. Shapiro, "Hierarchies in coloured petri nets", in "Advances in Petri Nets 1990", pp. 313–341 (Springer, 1991).

J, K., B. K, D. ML, G. LF, K. S, R. I, S. F, S. E and B. J, "Cyp2d6 and cyp2c19 genotype-based dose recommendations for antidepressants: a first step towards subpopulation-specific dosages", Acta Psychiatr Scand **104**, 3, 173–192 (2001).

Jensen, K., L. M. Kristensen and L. Wells, "Coloured petri nets and cpn tools for modelling and validation of concurrent systems", International Journal on Software Tools for Technology Transfer **9**, 3-4, 213–254 (2007).

Kanehisa, M. and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes", Nucleic acids research **28**, 1, 27–30 (2000).

Karp, P. D., M. Riley, S. M. Paley and A. Pellegrini-Toole, "The metacyc database", Nucleic acids research **30**, 1, 59–61 (2002).

Klein, D. and C. Manning, "Accurate unlexicalized parsing", Proceedings of the 41st meeting of the association for computation linguistics pp. 423–430 (2003).

Kohn, K. W., M. I. Aladjem, J. N. Weinstein and Y. Pommier, "Molecular interaction maps of bioregulatory networks: a general rubric for systems biology", Molecular biology of the cell **17**, 1, 1–13 (2006).

Kounev, S., C. Dutz and A. Buchmann, "Qpme-queueing petri net modeling environment", in "Quantitative Evaluation of Systems, 2006. QEST 2006. Third International Conference on", pp. 115–116 (IEEE, 2006).

Krallinger, M., A. Morgan, L. Smith, F. Leitner, L. Tanabe, J. Wilbur, L. Hirschman and A. Valencia, "Evaluation of text-mining systems for biology: overview of the second biocreative community challenge", Gnome Biology **9**, 2 (2008).

Krepska, E., N. Bonzanni, A. Feenstra, W. Fokkink, T. Kielmann, H. Bal and J. Heringa, "Design issues for qualitative modelling of biological cells with petri nets", Formal Methods in Systems Biology pp. 48–62 (2008).

Kuhn, M., M. Campillos, I. Letunic, L. J. Jensen and P. Bork, "A side effect resource to capture phenotypic effects of drugs", Molecular Systems Biology **6**, 343 (2010).

Kummer, O., F. Wienberg, M. Duvigneau, M. Köhler, D. Moldt and H. Rölke, "Renew–the reference net workshop", in "Tool Demonstrations, 21st International Conference on Application and Theory of Petri Nets, Computer Science Department, Aarhus University, Aarhus, Denmark", pp. 87–89 (2000).

Leaman, R., G. Gonzalez *et al.*, "Banner: An executable survey of advances in biomedical named entity recognition", in "Pacific Symposium on Biocomputing", vol. 13, pp. 652–663 (2008).

Leaman, R., C. Miller and G. Gonzalez, "Enabling recognition of diseases in biomedical text with machine learning: Corpus and benchmark", International Symposium on Languages in Biology and Medicine pp. 82–89 (2009).

Lee, J. and V. Lifschitz, "Describing additive fluents in action language $\mathcal{C}$+", in "Proc. of IJCAI 2003", (2003).

Lee, J., V. Lifschitz and F. Yang, "Action language bc: Preliminary report", in "Proceedings of the Twenty-Third international joint conference on Artificial Intelligence", pp. 983–989 (AAAI Press, 2013).

Lee, J. and Y. Meng, "Answer set programming modulo theories and reasoning about continuous changes", in "Proceedings of the Twenty-Third international joint conference on Artificial Intelligence", pp. 990–996 (AAAI Press, 2013).

Lifschitz, V., "What is answer set programming?", Proceedings of the Twenty-Third AAAI Conferece on Artificial Intelligence pp. 1594–1597 (2008).

Lifschitz, V. and H. Turner, “Splitting a logic program.”, in “ICLP”, vol. 94, pp. 23–37 (1994).

Luo, G., M. Cunningham, S. Kim, T. Burn, J. Lin, M. Sinz, G. Hamilton, C. Rizzo, S. Jolley, D. Gilbert *et al.*, “Cyp3a4 induction by drugs: correlation between a pregnane x receptor reporter gene assay and cyp3a4 expression in human hepatocytes”, Drug Metabolism and Disposition **30**, 7, 795–804 (2002).

Manna, Z. and A. Pnueli, *The temporal logic of reactive and concurrent systems: specifications*, vol. 1 (springer, 1992).

Music, G., “Schedule optimization based on coloured petri nets and local search”, in “Mathematical Modelling”, vol. 7, pp. 352–357 (2012).

Nagasaki, M., A. Saito, E. Jeong, C. Li, K. Kojima, E. Ikeda and S. Miyano, “Cell illustrator 4.0: A computational platform for systems biology”, In silico biology **10**, 1, 5–26 (2010).

Ostrowski, M. and T. Schaub, “Asp modulo csp: The clingcon system”, arXiv preprint arXiv:1210.2287 (2012).

Pearl, J., “Action as local surgery”, in “Working Notes of AAAI Spring Symposium Series, Extending Theories of Actions”, pp. 157–162 (1995).

Peterson, J. *et al.*, “A note on colored Petri nets”, Information Processing Letters **11**, 1, 40–43 (1980).

Peterson, J. L., “Petri nets”, Computing Surveys **9**, 3, 223–252 (1977).

Priami, C., A. Regev, E. Shapiro and W. Silverman, “Application of a stochastic name-passing calculus to representation and simulation of molecular processes”, Information processing letters **80**, 1, 25–31 (2001).

Ramchandani, C., “Analysis of asynchronous concurrent systems by Petri nets.”, Tech. rep., DTIC Document (1974).

Reece, J., M. Cain, L. Urry, P. Minorsky and S. Wasserman, *Campbell Biology* (Pearson Benjamin Cummings, 2010), URL `http://books.google.com/books?id=39vMSgAACAAJ`.

Regev, A., “Representation and simulation of molecular pathways in the stochastic pi-calculus”, in “Proceedings of the 2nd workshop on Computation of Biochemical Pathways and Genetic Networks”, vol. 88 (2001).

Reiter, R., “Natural actions, concurrency and continuous time in the situation calculus”, KR **96**, 2–13 (1996).

Rohr, C., W. Marwan and M. Heiner, “Snoopy—a unifying petri net framework to investigate biomolecular networks”, Bioinformatics **26**, 7, 974–975 (2010).

SF, Z., “Polymorphism of human cytochrome p450 2d6 and its clinical significance: Part i”, Clinical Pharmacoknetics **48**, 11, 689–723 (2009).

Sleator, D. and D. Temperley, “Parsing english with a link grammar”, Third International Workshop on Parsing Technologies (1993).

Son, T. C. and E. Pontelli, “A constructive semantic characterization of aggregates in answer set programming”, Theory and Practice of Logic Programming **7**, 3, 355 (2007).

Syropoulos, A., “Mathematics of multisets”, Multiset Processing pp. 347–358 (2001).

Tari, L., C. Baral, S. Anwar, S. Liang and J. Hakenberg, “Synthesis of pharmacokinetic pathways through knowledge acquisition and automated reasoning”, Pacific Symposium on Biocomputing **15**, 465–476 (2010a).

Tari, L., J. Cai, C. Baral, S. Liang and S. Anwar, “Discovering drug interactions: a text-mining and reasoning approach based on properties of drug metabolism.”, European Conference on Computational Biology (2010b).

Tari, L., J. Hakenberg, G. Gonzalez and C. Baral, “Querying a parse tree database of medline text to synthesize user-specific biomolecular networks”, Proc Pac Symp Biocomput **14**, 87–98 (2009).

Tari, L., P. H. Tu, J. Hakenberg, Y. Chen, T. C. Son, G. Gonzalez and C. Baral, “Incremental information extraction using relational databases”, IEEE Transactions on Knowledge and Data Engineering **24**, 1, 86–99 (2012).

Thorn, C. F., T. E. Klein and R. B. Altman, “Pharmacogenomics and bioinformatics: Pharmgkb”, Pharmacogenomics **11**, 4, 501–505 (2010).

Valk, R., “Self-modifying nets, a natural extension of Petri nets”, in “Automata, Languages and Programming”, vol. 62 of *Lecture Notes in Computer Science*, pp. 464–476 (Springer, 1978).

Valk, R., “Object Petri nets: Using the nets-within-nets paradigm”, in “Advances in Petri Nets: Lectures on Concurrency and Petri Nets”, edited by J. Desel, W. Reisig and G. Rozenberg, vol. 3098 of *Lecture Notes in Computer Science*, pp. 819–848 (Springer-Verlag, 2004).

Wishart, D., C. Knox, A. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam and M. Hassanali, “Drugbank: a knowledgebase for drugs, drug actions and drug targets.”, Nucleic Acids research **36**, 901–906 (2008).

Wishart, D. S., C. Knox and A. C. Guo, “Drugbank: a comprehensive resource for in silico drug discovery and exploration”, Nucleic Acids research **34**, 1, 668–672 (2006).

APPENDIX A

PROOFS OF VARIOUS PROPOSITIONS

Assumption: The definitions in this section assume the programs Π do not have recursion through aggregate atoms. Our ASP translation ensures this due to the construction of programs Π.

First we extend some definitions and properties related to ASP, such that they apply to rules with aggregate atoms. We will refer to the non-aggregate atoms as basic atoms. Recall the definitions of an ASP program given in section 2.2.1.

Proposition 9 (Forced Atom Proposition) *Let S be an answer set of a ground ASP program Π as defined in definition 4. For any ground instance of a rule R in Π of the form A_0:-$A_1, \ldots, A_m,$ **not** $B_1, \ldots,$ **not** $B_n, C_1, \ldots, C_k$. if $\forall A_i, 1 \leq i \leq m, S \models A_i$, and $\forall B_j, 1 \leq j \leq n, S \not\models B_j$, $\forall C_l, 1 \leq l \leq k, S \models C_l$ then $S \models A_0$.*

Proof: Let S be an answer set of a ground ASP program Π, $R \in \Pi$ be a ground rule such that $\forall B_j, 1 \leq j \leq n, S \not\models B_j$; and $\forall C_l, 1 \leq l \leq k, S \models C_l$. Then, the reduct $R^S \equiv \{p_1$:-$A_1, \ldots, A_m.; \ldots; p_h$:-$A_1, \ldots, A_m. \mid \{p_1, \ldots, p_h\} = S \cap lit(A_0)\}$. Since S is an answer set of Π, it is a model of Π^S. As a result, whenever, $\forall A_i, 1 \leq i \leq m, S \models A_i$, $\{p_1, \ldots, p_h\} \subseteq S$ and $S \models A_0$.

Proposition 10 (Supporting Rule Proposition) *If S is an answer set of a ground ASP program Π as defined in definition 4 then S is supported by Π. That is, if $S \models A_0$, then there exists a ground instance of a rule R in Π of the type A_0:-$A_1, \ldots,$ A_m, **not** $B_1, \ldots,$ **not** B_n, $C_1, \ldots, C_k$. such that $\forall A_i, 1 \leq i \leq m, S \models A_i$, $\forall B_j, 1 \leq j \leq n, S \not\models B_j$, and $\forall C_l, 1 \leq l \leq k, S \models C_l$.*

Proof: For S to be an answer set of Π, it must be the deductive closure of reduct Π^S. The deductive closure S of Π^S is iteratively built by starting from an empty set S, and adding head atoms of rules $R_h^S \equiv p_h$:-$A_1, \ldots, A_m.$, $R_h^S \in \Pi^S$, whenever, $S \models A_i, 1 \leq i \leq m$, where, R_h^S is a rule in the reduct of ground rule $R \in \Pi$ with $p_h \in lit(A_0) \cap S$. Thus, there is a rule $R \equiv A_0$:-$A_1, \ldots,$ A_m, **not** $B_1, \ldots,$ **not** B_n, $C_1, \ldots, C_k.$, $R \in \Pi$, such that $\forall C_l, 1 \leq l \leq k$ and $\forall B_j, 1 \leq j \leq n, S \not\models B_j$. Nothing else belongs in S.

Next, we extend the splitting set theorem Lifschitz and Turner (1994) to include aggregate atoms.

Definition 88 (Splitting Set) *A* Splitting Set *for a program Π is any set U of literals such that, for every rule $R \in \Pi$, if $U \models head(R)$ then $lit(R) \subset U$. The set U splits Π into upper and lower parts. The set of rules $R \in \Pi$ s.t. $lit(R) \subset U$ is called the bottom of Π w.r.t. U, denoted by $bot_U(\Pi)$. The rest of the rules, i.e. $\Pi \setminus bot_U(\Pi)$ is called the top of Π w.r.t. U, denoted by $top_U(\Pi)$.*

Proposition 11 *Let U be a splitting set of Π with answer set S and let $X = S \cap U$ and $Y = S \setminus U$. Then, the reduct of Π w.r.t. S, i.e. Π^S is equal to $bot_U(\Pi)^X \cup (\Pi \setminus bot_U(\Pi))^{X \cup Y}$.*

Proof: We can rewrite Π as $bot_U(\Pi) \cup (\Pi \setminus bot_U(\Pi))$ using the definition of splitting set. Then the reduct of Π w.r.t. S can be written in terms of X and Y, since $S = X \cup Y$. $\Pi^S = \Pi^{X \cup Y} = (bot_U(\Pi) \cup (\Pi \setminus bot_U(\Pi)))^{X \cup Y} = bot_U(\Pi)^{X \cup Y} \cup (\Pi \setminus bot_U(\Pi))^{X \cup Y}$. Since $lit(bot_U(\Pi)) \subseteq U$ and $Y \cap U = \emptyset$, the reduct of $bot_U(\Pi)^{X \cup Y} = bot_U(\Pi)^X$. Thus, $\Pi^{X \cup Y} = bot_U(\Pi)^X \cup (\Pi \setminus bot_U(\Pi))^{X \cup Y}$.

Proposition 12 *Let S be an answer set of a program Π, then $S \subseteq lit(head(\Pi))$.*

Proof: If S is an answer set of a program Π then S is produced by the deductive closure of Π^S (the reduct of Π w.r.t S). By definition of the deductive closure, nothing can be in S unless it is the head of some rule supported by S.

Splitting allow computing the answer set of a program Π in layers. Answer sets of the bottom layer are first used to partially evaluate the top layer, and then answer sets of the top layer are computed. Next, we define how a program is partially evaluated.

Intuitively, the partial evaluation of an aggregate atom c given splitting set U w.r.t. a set of literals X removes all literals that are part of the splitting set U from c and updates c's lower and upper bounds based on the literals in X, which usually come from bot_U of a program. The set X represents our knowledge about the positive literals, while the set $U \setminus X$ represents our knowledge about naf-literals at this stage. We can remove all literals in U from c, since the literals in U will not appear in the head of any rule in top_U.

Definition 89 (Partial Evaluation of Aggregate Atom) *The partial evaluation of an aggregate atom $c = l\ [B_0 = w_0, \ldots, B_m = w_m]\ u$, given splitting set U w.r.t. a set of literals X, written $eval_U(c, X)$ is a new aggregate atom c' constructed from c as follows:*

1. $pos(c') = pos(c) \setminus U$
2. $d = \sum_{B_i \in pos(c) \cap U \cap X} w_i$
3. $l' = l - d$, $u' = u - d$ *are the lower and upper limits of* c'

Next, we define how a program is partially evaluated given a splitting set U w.r.t. a set of literals X that form the answer-set of the lower layer. Intuitively, a partial evaluation deletes all rules from the partial evaluation for which the body of the rule is determined to be not supported by U w.r.t. X. This includes rules which have an aggregate atom c in their body s.t. $lit(c) \subseteq U$, but $X \not\models c$ [1]. In the remaining rules, the positive and negative literals that overlap with U are deleted, and so are the aggregate atoms that have $lit(c) \subseteq U$ (since such a c can be fully evaluated w.r.t. X). Each remaining aggregate atom is updated by removing atoms that belong to U [2], and updating its limits based on the answer-set X [3]. The head atom is not modified, since $eval_U(...)$ is performed on $\Pi \setminus bot_U(\Pi)$, which already removes all rules with heads atoms that intersect U.

Definition 90 (Partial Evaluation) *The* partial evaluation *of Π, given splitting set U w.r.t. a set of literals X is the program $eval_U(\Pi, X)$ composed of rules R' for each $R \in \Pi$ that satisfies all the following conditions:*

1. $pos(R) \cap U \subseteq X$,

[1] Note that we can fully evaluate an aggregate atom c w.r.t. answer-set X if $lit(c) \subseteq U$.

[2] Since the atoms in U will not appear in the head of any atoms in top_U and hence will not form a basis in future evaluations of c.

[3] The limit update requires knowledge of the current answer-set to update limit values.

2. $((neg(R) \cap U) \cap X) = \emptyset$, and

3. if there is a $c \in agg(R)$ s.t. $lit(c) \subseteq U$, then $X \models c$

A new rule R' is constructed from a rule R as follows:

1. $head(R') = head(R)$,

2. $pos(R') = pos(R) \setminus U$,

3. $neg(R') = neg(R) \setminus U$,

4. $agg(R') = \{eval_U(c, X) : c \in agg(R), lit(c) \not\subseteq U\}$

Proposition 13 *Let U be a splitting set for Π, X be an answer set of $bot_U(\Pi)$, and Y be an answer set of $eval_U(\Pi \setminus bot_U(\Pi), X)$. Then, $X \subseteq lit(\Pi) \cap U$ and $Y \subseteq lit(\Pi) \setminus U$.*

Proof: By proposition 12, $X \subseteq lit(head(bot_U(\Pi)))$, and $Y \subseteq lit(head(eval_U(\Pi \setminus bot_U(\Pi), X)))$. In addition, $lit(head(bot_U(\Pi))) \subseteq lit(bot_U(\Pi))$ and $lit(bot_U(\Pi)) \subseteq lit(\Pi) \cap U$ by definition of $bot_U(\Pi)$. Then $X \subseteq lit(\Pi) \cap U$, and $Y \subseteq lit(\Pi) \setminus U$.

Definition 91 (Solution) *Baral (2003) Let U be a splitting set for a program Π. A solution to Π w.r.t. U is a pair $\langle X, Y \rangle$ of literals such that:*

- *X is an answer set for $bot_U(\Pi)$*
- *Y is an answer set for $eval_U(top_U(\Pi), X)$; and*
- *$X \cup Y$ is consistent.*

Proposition 14 (Splitting Theorem) *Baral (2003) Let U be a splitting set for a program Π. A set S of literals is a consistent answer set for Π iff $S = X \cup Y$ for some solution $\langle X, Y \rangle$ of Π w.r.t. U.*

Lemma 1 *Let U be a splitting set of Π, C be an aggregate atom in Π, and X and Y be subsets of $lit(\Pi)$ s.t. $X \subseteq U$, and $Y \cap U = \emptyset$. Then, $X \cup Y \models C$ iff $Y \models eval_U(C, X)$.*

Proof:

1. Let $C' = eval_U(C, X)$, then by definition of partial evaluation of aggregate atom, $pos(C') = pos(C) \setminus U$, with lower limit $l' = l - d$, and upper limit $u' = u - d$, computed from l, u, the lower and upper limits of C, where

$$d = \sum_{B_i \in pos(C) \cap U \cap X} w_i$$

2. $Y \models C'$ iff

$$l' \leq \left(\sum_{B'_i \in pos(C') \cap Y} w'_i \right) \leq u'$$

– by definition of aggregate atom satisfaction.

3. then $Y \models C$ iff

$$l \leq \left(\sum_{B_i \in pos(C) \cap U \cap X} w_i + \sum_{B'_i \in (pos(C) \setminus U) \cap Y} w'_i \right) \leq u$$

4. however, $(pos(C) \cap U) \cap X$ and $(pos(C) \setminus U) \cap Y$ combined represent $pos(C) \cap (X \cup Y)$ – since

$$\begin{aligned} pos(C) \cap (X \cup Y) &= ((pos(C) \cap U) \cup (pos(C) \setminus U)) \cap (X \cup Y) \\ &= [((pos(C) \cap U) \cup (pos(C) \setminus U)) \cap X] \\ &\quad \cup [((pos(C) \cap U) \cup (pos(C) \setminus U)) \cap Y] \\ &= [(pos(C) \cap U) \cap X) \cup ((pos(C) \setminus U) \cap X)] \\ &\quad \cup [(pos(C) \cap U) \cap Y) \cup ((pos(C) \setminus U) \cap Y)] \\ &= [((pos(C) \cap U) \cap X) \cup \emptyset] \cup [\emptyset \cup ((pos(C) \setminus U) \cap Y)] \\ &= ((pos(C) \cap U) \cap X) \cup ((pos(C) \setminus U) \cap Y) \end{aligned}$$

where $X \subseteq U$ and $Y \cap U = \emptyset$

5. thus, $Y \models C$ iff

$$l \leq \left(\sum_{B_i \in pos(C) \cap (X \cup Y)} w_i \right) \leq u$$

6. which is the same as $X \cup Y \models C$

Lemma 2 *Let U be a splitting set for Π, and X, Y be subsets of $lit(\Pi)$ s.t. $X \subseteq U$ and $Y \cap U = \emptyset$. Then the body of a rule $R' \in eval_U(\Pi, X)$ is satisfied by Y iff the body of the rule $R \in \Pi$ it was constructed from is satisfied by $X \cup Y$.*

Proof: Y satisfies $body(R')$

iff $pos(R') \subseteq Y$, $neg(R') \cap Y = \emptyset$, $Y \models C'$ for each $C' \in agg(R')$ – by definition of rule satisfaction

iff $(pos(R) \cap U) \subseteq X$, $(pos(R) \setminus U) \subseteq Y$, $(neg(R) \cap U) \cap X) = \emptyset$, $(neg(R) \setminus U) \cap Y) = \emptyset$, X satisfies C for all $C \in agg(C)$ in which $lit(C) \subseteq U$, and Y satisfies $eval_U(C, X)$ for all $C \in agg(C)$ in which $lit(C) \not\subseteq U$ – using definition of partial evaluation

iff $pos(R) \subseteq X \cup Y$, $neg(R) \cap (X \cup Y) = \emptyset$, $X \cup Y \models C$ – using

- $(A \cap U) \cup (A \setminus U) = A$
- $A \cap (X \cup Y) = ((A \cap U) \cup (A \setminus U)) \cap (X \cup Y) = ((A \cap U) \cap (X \cup Y)) \cup ((A \setminus U) \cap (X \cup Y)) = (A \cap U) \cap X) \cup ((A \setminus U) \cap Y)$ – given $X \subseteq U$ and $Y \cap U = \emptyset$.
- and lemma 1

Proof of Splitting Theorem: Let U be a splitting set of Π, then a consistent set of literals S is an answer set of Π iff it can be written as $S = X \cup Y$, where X is an answer set of $bot_U(\Pi)$; and Y is an answer set of $eval_U(\Pi \setminus bot_U(\Pi), Y)$.

($\Leftarrow$) Let X is an answer set of $bot_U(\Pi)$; and Y is an answer set of $eval_U(\Pi \setminus bot_U(\Pi), X)$; we show that $X \cup Y$ is an answer set of Π.

By definition of $bot_U(\Pi)$, $lit(bot_U(\Pi)) \subseteq U$. In addition, by proposition 13, $Y \cap U = \emptyset$. Then, $\Pi^{X \cup Y} = (bot_U(\Pi) \cup (\Pi \setminus bot_U(\Pi)))^{X \cup Y} = bot_U(\Pi)^{X \cup Y} \cup (\Pi \setminus bot_U(\Pi))^{X \cup Y} = bot_U(\Pi)^X \cup (\Pi \setminus bot_U(\Pi))^{X \cup Y}$.

Let r be a rule in $\Pi^{X \cup Y}$, s.t. $X \cup Y \models body(r)$ then we show that $X \cup Y \models head(r)$. The rule r either belongs to $bot_U(\Pi)^X$ or $(\Pi \setminus bot_U(\Pi))^{X \cup Y}$.

Case 1: say $r \in bot_U(\Pi)^X$ be a rule whose body is satisfied by $X \cup Y$

1. then there is a rule $R \in bot_U(\Pi)$ s.t. $r \in R^X$
2. then $X \models body(R)$ – since $X \cup Y \models body(r)$; $lit(bot_U(\Pi)) \subseteq U$ and $Y \cap U = \emptyset$
3. we already have $X \models head(R)$ – since X is an answer set of $bot_U(\Pi)$; given
4. then $X \cup Y \models head(R)$ – because $lit(R) \subseteq U$ and $Y \cap U = \emptyset$
5. consequently, $X \cup Y \models head(r)$

Case 2: say $r \in (\Pi \setminus bot_U(\Pi))^{X \cup Y}$ be a rule whose body is satisfied by $X \cup Y$

1. then there is a rule $R \in (\Pi \setminus bot_U(\Pi))$ s.t. $r \in R^{X \cup Y}$
2. then $lit(head(R)) \cap U = \emptyset$ – otherwise, R would have belonged to $bot_U(\Pi)$, by definition of splitting set
3. then $head(r) \in Y$ – since $X \subseteq U$
4. in addition, $pos(R) \subseteq X \cup Y$, $neg(R) \cap (X \cup Y) = \emptyset$, $X \cup Y \models C$ for each $C \in agg(R)$ – using definition of reduct
5. then $pos(R) \cap U \subseteq X$ or $pos(R) \setminus U \subseteq Y$; $(neg(R) \cap U) \cap X = \emptyset$ and $(neg(R) \setminus U) \cap Y = \emptyset$; and for each $C \in agg(R)$, either $X \models C$ if $lit(C) \subseteq U$, or $Y \models eval_U(C, X)$ if $lit(C) \not\subseteq U$ – by rearranging, lemma 1, $X \subseteq U$, $Y \cap U = \emptyset$, and definition of partial evaluation of an aggregate atom
6. note that $pos(R) \cap U \subseteq X$, $(neg(R) \cap U) \cap X = \emptyset$, and for each $C \in agg(R)$, s.t. $lit(C) \subseteq U$, $X \models C$, represent conditions satisfied by each rule that become part of a partial evaluation – using definition of partial evaluation
7. and $pos(R) \setminus U$, $neg(R) \setminus U$, and for each $C \in agg(R)$, $eval_U(C, X)$ are the modifications made to the rule during partial evaluation given splitting set U w.r.t. X – using definition of partial evaluation

8. and $pos(R) \setminus U \subseteq Y$, $(neg(R) \setminus U) \cap Y = \emptyset$, and for each $C \in agg(R)$, $Y \models eval_U(C, X)$ if $lit(C) \nsubseteq U$ represent conditions satisfied by rules that become part of the reduct w.r.t Y – using definition of partial evaluation and reduct

9. then r is a rule in reduct $eval_U(\Pi \setminus bot_U(\Pi), X)^Y$ – using (8), (2) above

10. in addition, given that Y satisfies $eval_U(\Pi \setminus bot_U(\Pi), X)$, and $head(r) \cap U = \emptyset$, we have $X \cup Y \models head(r)$

Next we show that $X \cup Y$ satisfies all rules of Π. Say, R is a rule in Π not satisfied by $X \cup Y$. Then, either it belongs to $bot_U(\Pi)$ or $(\Pi \setminus bot_U(\Pi))$. If it belongs to $bot_U(\Pi)$, it must not be satisfied by X, since $lit(bot_U(\Pi)) \subseteq U$ and $Y \cap U = \emptyset$. However, the contrary is given to be true. On the other hand if it belongs to $(\Pi \setminus bot_U(\Pi))$, then $X \cup Y$ satisfies $body(R)$ but not $head(R)$. That would mean that its $head(R)$ is not satisfied by Y, since $head(R) \cap U = \emptyset$ by definition of splitting set. However, from lemma 2 we know that if $body(R)$ is satisfied by $X \cup Y$, $body(R')$ is satisfied by Y for $R' \in eval_U(\Pi \setminus bot_U(\Pi), X)$. We also know that Y satisfies all rules in $eval_U(\Pi \setminus bot_U(\Pi), X)$. So, R' must be satisfied by Y contradicting our assumption. Thus, all rules of Π are satisfied by $X \cup Y$ and $X \cup Y$ is an answer set of Π.

($\Rightarrow$) Let S be a consistent answer set of Π, we show that $S = X \cup Y$ for sets X and Y s.t. X is an answer set of $bot_U(\Pi)$ and Y is an answer set of $eval_U(\Pi \setminus bot_U(\Pi), X)$. We take $X = S \cap U$, $Y = S \setminus U$, then $S = X \cup Y$.

Case 1: We show that X is answer set of $bot_U(\Pi)$

1. Π can be split into $bot_U(\Pi) \cup (\Pi \setminus bot_U(\Pi))$ – by definition of splitting

2. then $X \cup Y$ satisfies $bot_U(\Pi)$ – $X \cup Y$ is an answer set of Π; given

3. however $lit(bot_U(\Pi)) \subseteq U$, $Y \cap U = \emptyset$ – by definition of splitting

4. then X satisfies $bot_U(\Pi)$ – since elements of Y do not appear in the rules of $bot_U(\Pi)$

5. then X is an answer set of $bot_U(\Pi)$

Case 2: We show that Y is answer set of $eval_U(\Pi \setminus bot_U(\Pi), X)$

1. let r be a rule in $eval_U(\Pi \setminus bot_U(\Pi), X)^Y$, s.t. its body is satisfied by Y

2. then $r \in R^Y$ for an $R \in eval_U(\Pi \setminus bot_U(\Pi), X)$ s.t. (i) $pos(R) \subseteq Y$ (ii) $neg(R) \cap Y = \emptyset$ (iii) $Y \models C$ for all $C \in agg(R)$ (iv) $head(R) \cap Y \neq \emptyset$ – using definition of reduct

3. each R is constructed from $R' \in \Pi$ that satisfies all the following conditions (i) $pos(R') \subseteq U \cap X$ (ii) $(neg(R') \cap U) \cap X = \emptyset$ (iii) if there is a $C' \in agg(R')$ s.t. $lit(C') \subseteq U$, then $X \models C'$; and each $C \in agg(R)$ is a partial evaluation of $C' \in agg(R')$ s.t. $C = eval_U(C', X)$ – using definition of partial evaluation

4. then the $body(R')$ satisfies all the following conditions:
   (a) $pos(R') \subseteq X \cup Y$ – since $X \subseteq U$, $X \cap Y = \emptyset$
   (b) $neg(R') \cap (X \cup Y) = \emptyset$ – since $X \subseteq U$, $X \cap Y = \emptyset$
   (c) $X \cup Y \models C'$ for each $C' \in agg(R')$ – since (i) each $C' \in agg(R')$ with $lit(C') \subseteq U$ satisfied by X is also satisfied by $X \cup Y$ as $lit(Y) \cap lit(C') = \emptyset$; and (ii) each $C' \in agg(R')$ with $lit(C') \nsubseteq U$ is satisfied by $X \cup Y$ – using partial evaluation, reduct construction, and $X \cap Y = \emptyset$

5. then $X \cup Y$ satisfies $body(R')$ – from previous line

6. in addition, $lit(head(R')) \cap U = \emptyset$, otherwise, R' would have belonged to $bot_U(\Pi)$ by definition of splitting set

7. then R' is a rule in $\Pi \setminus bot_U(\Pi)$ – from the last two lines

8. we know that $X \cup Y$ satisfies every rule in $(\Pi \setminus bot_U(\Pi))$ – given; and that elements of U do not appear in the head of rules in $(\Pi \setminus bot_U(\Pi))$ – from definition of splitting; then Y must satisfy the head of these rules

9. then Y satisfies $head(R')$ – from (8)

10. Next we show that Y satisfies all rules of $eval_U(\Pi \setminus bot_U(\Pi), X)$. Let R' be a rule in $eval_U(\Pi \setminus bot_U(\Pi), X)$ such that $body(R')$ is satisfied by Y but not $head(R')$. Since $head(R') \cap Y = \emptyset$, $head(R')$ is not satisfied by $X \cup Y$ either. Then, there is an $R \in (\Pi \setminus bot_U(\Pi))$ such that $X \cup Y$ satisfies $body(R)$ but not $head(R)$, which contradicts given. Thus, Y satisfies all rules of $eval_U(\Pi \setminus bot_U(\Pi), X)$.

11. Then Y is an answer set of $eval_U(\Pi \setminus bot_U(\Pi), X)$

Definition 92 (Splitting Sequence) *Baral (2003) A* splitting sequence *for a program* Π *is a monotone, continuous sequence* $\langle U_\alpha \rangle_{\alpha<\mu}$ *of splitting sets of* Π *such that* $\bigcup_{\alpha<\mu} U_\mu = lit(\Pi)$.

Definition 93 (Solution) *Baral (2003) Let* $U = \langle U_\alpha \rangle_{\alpha<\mu}$ *be a splitting sequence for a program* Π. *A* solution *to* Π *w.r.t* U *is a sequence* $\langle X_\alpha \rangle_{\alpha<\mu}$ *of sets of literals such that:*

1. X_0 *is an answer set for* $bot_{U_0}(\Pi)$

2. for any ordinal $\alpha + 1 < \mu$, $X_{\alpha+1}$ *is an answer set of the program:*

$$eval_{U_\alpha}(bot_{U_{\alpha+1}}(\Pi) \setminus bot_{U_\alpha}(\Pi), \bigcup_{\nu \leq \alpha} X_\nu)$$

3. for any limit ordinal $\alpha < \mu, X_\alpha = \emptyset$, *and*

4. $\bigcup_{\alpha\leq\mu}(X_\alpha)$ *is consistent*

Proposition 15 (Splitting Sequence Theorem) *Baral (2003) Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a program Π. A set S of literals is a consistent answer set for Π iff $S = \bigcup_{\alpha<\mu} X_\alpha$ for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to Π w.r.t U.*

Proof: Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence of Π, then a consistent set of literals $S = \bigcup_{\alpha<\mu} X_\alpha$ is an answer set of Π^S iff X_0 is an answer set of $bot_{U_0}(\Pi)$ and for any ordinal $\alpha + 1 < \mu$, $X_{\alpha+1}$ is an answer set of $eval_{U_\alpha}(bot_{U_{\alpha+1}}(\Pi) \setminus bot_{U_\alpha}(\Pi), \bigcup_{\nu\leq\alpha} X_\nu)$.

Note that every literal in $bot_{U_0}(\Pi)$ belongs to $lit(\Pi) \cap U_0$, and every literal occurring in $eval_{U_\alpha}(bot_{U_{\alpha+1}}(\Pi) \setminus bot_{U_\alpha}(\Pi), \bigcup_{\nu\leq\alpha} X_\nu)$, $(\alpha+1 < \mu)$ belongs to $lit(\Pi) \cap (U_{\alpha+1} \setminus U_\alpha)$. In addition, X_0, and all $X_{\alpha+1}$ are pairwise disjoint.

We prove the theorem by induction over the splitting sequence.
Base case: $\alpha = 1$. The splitting sequence is $U_0 \subseteq U_1$.

Then the sub-program $\Pi_1 = bot_{U_1}(\Pi)$ contains all literals in U_1; and U_0 splits Π_1 into $bot_{U_0}(\Pi_1)$ and $bot_{U_1}(\Pi_1) \setminus bot_{U_0}(\Pi_1)$. Then, $S_1 = X_0 \cup X_1$ is a consistent answer set of Π_1 iff $X_0 = S_1 \cap U_0$ is an answer set of $bot_{U_0}(\Pi_1)$ and $X_1 = S_1 \setminus U_0$ is an answer set of $eval_{U_0}(\Pi_1 \setminus bot_{U_0}(\Pi_1), X_1)$ – by the splitting theorem

Since $bot_{U_0}(\Pi_1) = bot_{U_0}(\Pi)$ and $bot_{U_1}(\Pi_1) \setminus bot_{U_0}(\Pi_1) = bot_{U_1}(\Pi) \setminus bot_{U_0}(\Pi)$; $S_1 = X_0 \cup X_1$ is an answer set for $\Pi \setminus bot_{U_1}(\Pi)$.
Induction: Assume theorem holds for $\alpha = k$, show theorem holds for $\alpha = k + 1$.

The inductive assumption holds for the splitting sequence $U_0 \subseteq \cdots \subseteq U_k$. Then the sub-program $\Pi_k = bot_{U_k}(\Pi)$ contains all literals in U_k and $S_k = X_0 \cup \cdots \cup X_k$ is a consistent answer set of Π^{S_k} iff X_0 is an answer set for $bot_{U_0}(\Pi_k)$ and for any $\alpha \leq k$, $X_{\alpha+1}$ is answer set of $eval_{U_\alpha}(bot_{U_{\alpha+1}}(\Pi) \setminus bot_{U_\alpha}(\Pi), X_0 \cup \cdots \cup X_\alpha)$

We show that the theorem holds for $\alpha = k+1$. The splitting sequence is $U_0 \subseteq U_{k+1}$.

Then the sub-program $\Pi_{k+1} = bot_{U_{k+1}}(\Pi)$ contains all literals U_{k+1}. We have U_k split Π_{k+1} into $bot_{U_k}(\Pi_{k+1})$ and $bot_{U_{k+1}}(\Pi_{k+1}) \setminus bot_{U_k}(\Pi_{k+1})$. Then, $S_{k+1} = X_{0:k} \cup X_{k+1}$ is a consistent answer set of Π_{k+1} iff $X_{0:k} = S_{k+1} \cap U_k$ is an answer set of $bot_{U_k}(\Pi_{k+1})$ and $X_{k+1} = S_{k+1} \setminus U_k$ is an answer set of $eval_{U_k}(\Pi_{k+1} \setminus bot_{U_k}(\Pi_{k+1}, X_{k+1})$ – by the splitting theorem

Since $bot_{U_k}(\Pi_{k+1}) = bot_{U_k}(\Pi)$ and $bot_{U_{k+1}}(\Pi_{k+1}) \setminus bot_{U_k}(\Pi_{k+1}) = bot_{U_{k+1}}(\Pi) \setminus bot_{U_k}(\Pi)$; $S_{k+1} = X_{0:k} \cup X_1$ is an answer set for $\Pi \setminus bot_{U_{k+1}}(\Pi)$.

From the inductive assumption we know that $X_0 \cup \cdots \cup X_k$ is a consistent answer set of $bot_{U_k}(\Pi)$, X_0 is the answer set of $bot_{U_0}(\Pi)$, and for each $0 \leq \alpha \leq k$, $X_{\alpha+1}$ is answer set of $eval_{U_\alpha}(bot_{U_{\alpha+1}}(\Pi) \setminus bot_{U_\alpha}(\Pi), X_0 \cup \cdots \cup X_\alpha)$. Thus, $X_{0:k} = X_0 \cup \cdots \cup X_k$.

Combining above with the inductive assumption, we get $S_{k+1} = X_0 \cup \cdots \cup X_{k+1}$ is a consistent answer set of $\Pi^{S_{k+1}}$ iff X_0 is an answer set for $bot_{U_0}(\Pi_{k+1})$ and for any $\alpha \leq k + 1$, $X_{\alpha+1}$ is answer set of $eval_{U_\alpha}(bot_{U_{\alpha+1}}(\Pi) \setminus bot_{U_\alpha}(\Pi), X_0 \cup \cdots \cup X_\alpha)$.

In addition, for some $\alpha < \mu$, where μ is the length of the splitting sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ of Π, $bot_{U_\alpha}(\Pi)$ will be the entire Π, i.e. $lit(\Pi) = U_\alpha$. Then the set S of literals is a consistent answer set of Π iff $S = \bigcup_{\alpha<\mu}(X_\alpha)$ for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to Π w.r.t U.

A.1 Proof of Proposition 1

Let $PN = (P, T, E, W)$ be a Petri Net, M_0 be its initial marking and let $\Pi^0(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum *ntok* tokens on any place node, as defined in section 2.3. Then $X = M_0, T_0, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of a PN (w.r.t. M_0) iff there is an answer set A of $\Pi^0(PN, M_0, k, ntok)$ such that:

$$\{fires(t,ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t,ts) : fires(t,ts) \in A\} \tag{A.1}$$

$$\begin{aligned}\{holds(p,q,ts) : p \in P, q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ = \{holds(p,q,ts) : holds(p,q,ts) \in A\}\end{aligned} \tag{A.2}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.1) and (A.2) and show that A is an answer set of Π^0

(II) Given an answer set A of Π^0, we create an execution sequence X such that (A.1) and (A.2) are satisfied

First we show (I): Given PN and its execution sequence X, we create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$
2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$
3. $A_3 = \{place(p) : p \in P\}$
4. $A_4 = \{trans(t) : t \in T\}$
5. $A_5 = \{ptarc(p,t,n) : (p,t) \in E^-, n = W(p,t)\}$, where $E^- \subseteq E$
6. $A_6 = \{tparc(t,p,n) : (t,p) \in E^+, n = W(t,p)\}$, where $E^+ \subseteq E$
7. $A_7 = \{holds(p,q,0) : p \in P, q = M_0(p)\}$
8. $A_8 = \{notenabled(t,ts) : t \in T, 0 \leq ts \leq k, \exists p \in \bullet t, M_{ts}(p) < W(p,t)\}$
   per definition 12 (enabled transition)
9. $A_9 = \{enabled(t,ts) : t \in T, 0 \leq ts \leq k, \forall p \in \bullet t, W(p,t) \leq M_{ts}(p)\}$
   per definition 12 (enabled transition)
10. $A_{10} = \{fires(t,ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
    from definition 15 (firing set) , only an enabled transition may fire
11. $A_{11} = \{add(p,q,t,ts) : t \in T_{ts}, p \in t\bullet, q = W(t,p), 0 \leq ts \leq k\}$
    per definition 13 (transition execution)
12. $A_{12} = \{del(p,q,t,ts) : t \in T_{ts}, p \in \bullet t, q = W(p,t), 0 \leq ts \leq k\}$
    per definition 13 (transition execution)
13. $A_{13} = \{tot\_incr(p,q,ts) : p \in P, q = \sum_{t \in T_{ts}, p \in t\bullet} W(t,p), 0 \leq ts \leq k\}$
    per definition 16 (firing set execution)

14. $A_{14} = \{tot\_decr(p,q,ts) : p \in P, q = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t), 0 \leq ts \leq k\}$
per definition 16 (firing set execution)

15. $A_{15} = \{consumesmore(p,ts) : p \in P, q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t), q1 > q, 0 \leq ts \leq k\}$
per definition 14 (conflicting transitions) for enabled transition set T_{ts}

16. $A_{16} = \{consumesmore : \exists p \in P : q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t), q1 > q, 0 \leq ts \leq k\}$
per definition 14 (conflicting transitions)

17. $A_{17} = \{holds(p,q,ts+1) : p \in P, q = M_{ts+1}(p), 0 \leq ts < k\}$,
where $M_{ts+1}(p) = M_{ts}(p) - \sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in t \bullet} W(t,p)$
according to definition 15 (firing set execution)

We show that A satisfies (A.1) and (A.2), and A is an answer set of Π^0.

A satisfies (A.1) and (A.2) by its construction (given above). We show A is an answer set of Π^0 by splitting. We split $lit(\Pi^0)$ (literals of Π^0) into a sequence of $6(k+1)+2$ sets:

- $U_0 = head(f1) \cup head(f2) \cup head(f3) \cup head(f4) \cup head(f5) \cup head(f6) \cup head(i1) = \{place(p) : p \in P\} \cup \{trans(t) : t \in T\} \cup \{ptarc(p,t,n) : (p,t) \in E^-, n = W(p,t)\} \cup \{tparc(t,p,n) : (t,p) \in E^+, n = W(t,p)\} \cup \{time(0), \dots, time(k)\} \cup \{num(0), \dots, num(ntok)\} \cup \{holds(p,q,0) : p \in P, q = M_0(p)\}$
- $U_{6k+1} = U_{6k+0} \cup head(e1)^{ts=k} = U_{6k+0} \cup \{notenabled(t,k) : t \in T\}$
- $U_{6k+2} = U_{6k+1} \cup head(e2)^{ts=k} = U_{6k+1} \cup \{enabled(t,k) : t \in T\}$
- $U_{6k+3} = U_{6k+2} \cup head(a1)^{ts=k} = U_{6k+2} \cup \{fires(t,k) : t \in T\}$
- $U_{6k+4} = U_{6k+3} \cup head(r1)^{ts=k} \cup head(r2)^{ts=k} = U_{6k+3} \cup \{add(p,q,t,k) : p \in P, t \in T, q = W(t,p)\} \cup \{del(p,q,t,k) : p \in P, t \in T, q = W(p,t)\}$
- $U_{6k+5} = U_{6k+4} \cup head(r3)^{ts=k} \cup head(r4)^{ts=k} = U_{6k+4} \cup \{tot\_incr(p,q,k) : p \in P, 0 \leq q \leq ntok\} \cup \{tot\_decr(p,q,k) : p \in P, 0 \leq q \leq ntok\}$
- $U_{6k+6} = U_{6k+5} \cup head(r5)^{ts=k} \cup head(a2)^{ts=k} = U_{6k+5} \cup \{holds(p,q,k+1) : p \in P, 0 \leq q \leq ntok\} \cup \{consumesmore(p,k) : p \in P\}$
- $U_{6k+7} = U_{6k+6} \cup head(a3) = U_{7k+6} \cup \{consumesmore\}$

where $head(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\dots,ts) : a(\dots,ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 6k+7$ form a splitting sequence, since each U_i is a splitting set of Π^0, and $\langle U_\alpha \rangle_{\alpha < \mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \dots \subseteq U_{6k+7}$ and $\bigcup_{\alpha < \mu} U_\alpha = lit(\Pi^0)$.

We compute the answer set of Π^0 using the splitting sets as follows:

1. $bot_{U_0}(\Pi^0) = f1 \cup f2 \cup f3 \cup f4 \cup f5 \cup i1 \cup f6$ and $X_0 = A_1 \cup \dots \cup A_7$ $(= U_0)$ is its answer set – using forced atom proposition

2. $\mathit{eval}_{U_0}(\mathit{bot}_{U_1}(\Pi^0) \setminus \mathit{bot}_{U_0}(\Pi^0), X_0) = \{\mathit{notenabled}(t,0)\text{:-}.|\{\mathit{trans}(t), \mathit{ptarc}(p,t,n), \mathit{holds}(p,q,0)\} \subseteq X_0, \text{ where } q < n\}$. Its answer set $X_1 = A_8^{ts=0}$ – using forced atom proposition and construction of A_8.
   (a) where, $q = M_0(p)$, $n = W(p,t)$, and for an arc $(p,t) \in E^-$ – by construction of $i1$ and $f3$ in Π^0, and
   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)
   (c) thus, $\mathit{notenabled}(t,0) \in X_1$ represents $\exists p \in \bullet t : M_0(p) < W(p,t)$.

3. $\mathit{eval}_{U_1}(\mathit{bot}_{U_2}(\Pi^0) \setminus \mathit{bot}_{U_1}(\Pi^0), X_0 \cup X_1) = \{\mathit{enabled}(t,0)\text{:-}.|\mathit{trans}(t) \in X_0 \cup X_1, \mathit{notenabled}(t,0) \notin X_0 \cup X_1\}$. Its answer set is $X_2 = A_9^{ts=0}$ – using forced atom proposition and construction of A_9.
   (a) since an $\mathit{enabled}(t,0) \in X_2$ if $\nexists$ $\mathit{notenabled}(t,0) \in X_0 \cup X_1$, which is equivalent to $\nexists p \in \bullet t : M_0(p) < W(p,t) \equiv \forall p \in \bullet t : M_0(p) \geq W(p,t)$.

4. $\mathit{eval}_{U_2}(\mathit{bot}_{U_3}(\Pi^0) \setminus \mathit{bot}_{U_2}(\Pi^0), X_0 \cup X_1 \cup X_2) = \{\{\mathit{fires}(t,0)\}\text{:-}.|\mathit{enabled}(t,0)$ holds in $X_0 \cup X_1 \cup X_2\}$. It has multiple answer sets $X_{3.1}, \dots, X_{3.n}$, corresponding to elements of power set of $\mathit{fires}(t,0)$ atoms in $\mathit{eval}_{U_2}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^0 determined using splitting is equal to A, we only consider the set that matches the $\mathit{fires}(t,0)$ elements in A and call it X_3, ignoring the rest. Thus, $X_3 = A_{10}^{ts=0}$, representing T_0.

5. $\mathit{eval}_{U_3}(\mathit{bot}_{U_4}(\Pi^0) \setminus \mathit{bot}_{U_3}(\Pi^0), X_0 \cup \dots \cup X_3) = \{\mathit{add}(p,n,t,0)\text{:-}.|\{\mathit{fires}(t,0), \mathit{tparc}(t,p,n)\} \subseteq X_0 \cup \dots \cup X_3\} \cup \{\mathit{del}(p,n,t,0)\text{:-}.|\{\mathit{fires}(t,0), \mathit{ptarc}(p,t,n)\} \subseteq X_0 \cup \dots \cup X_3\}$. It's answer set is $X_4 = A_{11}^{ts=0} \cup A_{12}^{ts=0}$ – using forced atom proposition and definitions of A_{11} and A_{12}.
   (a) where each add atom is equivalent to $n = W(t,p) : p \in t\bullet$,
   (b) and each del atom is equivalent to $n = W(p,t) : p \in \bullet t$,
   (c) representing the effect of transitions in T_0 – by construction

6. $\mathit{eval}_{U_4}(\mathit{bot}_{U_5}(\Pi^0) \setminus \mathit{bot}_{U_4}(\Pi^0), X_0 \cup \dots \cup X_4) = \{\mathit{tot\_incr}(p,qq,0)\text{:-}.| qq = \sum_{\mathit{add}(p,q,t,0) \in X_0 \cup \dots \cup X_4} q\} \cup \{\mathit{tot\_decr}(p,qq,0)\text{:-}.|qq = \sum_{\mathit{del}(p,q,t,0) \in X_0 \cup \dots \cup X_4} q\}$. It's answer set is $X_5 = A_{13}^{ts=0} \cup A_{14}^{ts=0}$ – using forced atom proposition and definitions of A_{13} and A_{14}, ad definition 3 (semantics of aggregate assignment atom).
   (a) where each $\mathit{tot\_incr}(p,qq,0)$, $qq = \sum_{\mathit{add}(p,q,t,0) \in X_0 \cup \dots X_4} q$ $\equiv qq = \sum_{t \in X_3, p \in t\bullet} W(p,t)$,
   (b) and each $\mathit{tot\_decr}(p,qq,0)$, $qq = \sum_{\mathit{del}(p,q,t,0) \in X_0 \cup \dots X_4} q$ $\equiv qq = \sum_{t \in X_3, p \in \bullet t} W(t,p)$,
   (c) represent the net effect of transitions in T_0 – by construction

7. $\mathit{eval}_{U_5}(\mathit{bot}_{U_6}(\Pi^0) \setminus \mathit{bot}_{U_5}(\Pi^0), X_0 \cup \cdots \cup X_5) = \{\mathit{consumesmore}(p,0)\text{:-.}| \{\mathit{holds}(p,q,0),\ \mathit{tot\_decr}(p,q1,0)\} \subseteq X_0 \cup \cdots \cup X_5,\ q1 > q\} \cup \{\mathit{holds}(p,q,1)\text{:-.}| \{\mathit{holds}(p,q1,0),\ \mathit{tot\_incr}(p,q2,0),\ \mathit{tot\_decr}(p,q3,0)\} \subseteq X_0 \cup \cdots \cup X_5,\ q = q1 + q2 - q3\}$. It's answer set is $X_6 = A_{15}^{ts=0} \cup A_{17}^{ts=0}$ – using forced atom proposition.

   (a) where, $\mathit{consumesmore}(p,0)$ represents $\exists p : q = M_0(p)$, $q1 = \sum_{t \in T_0, p \in \bullet t} W(p,t), q1 > q$ – indicating place p will be overconsumed if T_0 is fired as defined in definition 14 (conflicting transitions)

   (b) and $\mathit{holds}(p,q,1)$ represents $q = M_1(p)$ – by construction

$\vdots$

8. $\mathit{eval}_{U_{6k+0}}(\mathit{bot}_{U_{6k+1}}(\Pi^0) \setminus \mathit{bot}_{U_{6k+0}}(\Pi^0), X_0 \cup \cdots \cup X_{6k+0}) = \{\mathit{notenabled}(t,k)\text{:-.}|\{\mathit{trans}(t), \mathit{ptarc}(p,t,n), \mathit{holds}(p,q,k)\} \subseteq X_0 \cup \cdots \cup X_{6k+0}$, where $q < n\}$. Its answer set $X_{6k+1} = A_8^{ts=k}$ – using forced atom proposition and construction of A_8.

   (a) where, $q = M_k(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of holds and ptarc predicates in Π^0, and

   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

   (c) thus, $\mathit{notenabled}(t,k) \in X_{6k+1}$ represents $\exists p \in \bullet t : M_k(p) < W(p,t)$.

9. $\mathit{eval}_{U_{6k+1}}(\mathit{bot}_{U_{6k+2}}(\Pi^0) \setminus \mathit{bot}_{U_{6k+1}}(\Pi^0), X_0 \cup \cdots \cup X_{6k+1}) = \{\mathit{enabled}(t,k)\text{:-.}| \mathit{trans}(t) \in X_0 \cup \cdots \cup X_{6k+1}, \mathit{notenabled}(t,k) \notin X_0 \cup \cdots \cup X_{6k+1}\}$. Its answer set is $X_{6k+2} = A_9^{ts=k}$ – using forced atom proposition and construction of A_9.

   (a) since an $\mathit{enabled}(t,k) \in X_{6k+2}$ if $\nexists\ \mathit{notenabled}(t,k) \in X_0 \cup \cdots \cup X_{6k+1}$, which is equivalent to $\nexists p \in \bullet t : M_k(p) < W(p,t) \equiv \forall p \in \bullet t : M_k(p) \geq W(p,t)$.

10. $\mathit{eval}_{U_{6k+2}}(\mathit{bot}_{U_{6k+3}}(\Pi^0) \setminus \mathit{bot}_{U_{6k+2}}(\Pi^0), X_0 \cup \cdots \cup X_{6k+2}) = \{\{\mathit{fires}(t,k)\}\text{:-.}|\ \mathit{enabled}(t,k)$ holds in $X_0 \cup \cdots \cup X_{6k+2}\}$. It has multiple answer sets $X_{6k+3.1}, \ldots, X_{6k+3.n}$, corresponding to elements of power set of $\mathit{fires}(t,k)$ atoms in $\mathit{eval}_{U_{6k+2}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^0 determined using splitting is equal to A, we only consider the set that matches the $\mathit{fires}(t,k)$ elements in A and call it X_{6k+3}, ignoring the reset. Thus, $X_{6k+3} = A_{10}^{ts=k}$, representing T_k.

11. $\mathit{eval}_{U_{6k+3}}(\mathit{bot}_{U_{6k+4}}(\Pi^0) \setminus \mathit{bot}_{U_{6k+3}}(\Pi^0), X_0 \cup \cdots \cup X_{6k+3}) = \{\mathit{add}(p,n,t,k)\text{:-.}| \{\mathit{fires}(t,k), \mathit{tparc}(t,p,n)\} \subseteq X_0 \cup \cdots \cup X_{6k+3}\} \cup \{\mathit{del}(p,n,t,k)\text{:-.}|\ \{\mathit{fires}(t,k), \mathit{ptarc}(p,t,n)\} \subseteq X_0 \cup \cdots \cup X_{6k+3}\}$. It's answer set is $X_{6k+4} = A_{11}^{ts=k} \cup A_{12}^{ts=k}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

   (a) where, each add atom is equivalent to $n = W(t,p) : p \in t\bullet$,

   (b) and each del atom is equivalent to $n = W(p,t) : p \in \bullet t$

   (c) representing the effect of transitions in T_k

12. $eval_{U_{6k+4}}(bot_{U_{6k+5}}(\Pi^0) \setminus bot_{U_{6k+4}}(\Pi^0), X_0 \cup \dots \cup X_{6k+4}) = \{tot\_incr(p, qq, k)\text{:-.}|$ $qq = \sum_{add(p,q,t,k) \in X_0 \cup \dots \cup X_{6k+4}} q\} \cup$ $\{tot\_decr(p, qq, k)\text{:-.}|$ $qq = \sum_{del(p,q,t,k) \in X_0 \cup \dots \cup X_{6k+4}} q\}$. It's answer set is $X_5 = A_{13}^{ts=k} \cup A_{14}^{ts=k}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

    (a) where, each $tot\_incr(p, qq, k)$, $qq = \sum_{add(p,q,t,k) \in X_0 \cup \dots X_{7k+4}} q$ $\equiv qq = \sum_{t \in X_{6k+3}, p \in t\bullet} W(p, t)$,

    (b) and each $tot\_decr(p, qq, k)$, $qq = \sum_{del(p,q,t,k) \in X_0 \cup \dots X_{7k+4}} q$ $\equiv qq = \sum_{t \in X_{6k+3}, p \in \bullet t} W(t, p)$,

    (c) represent the net effect of transitions in T_k

13. $eval_{U_{6k+5}}(bot_{U_{6k+6}}(\Pi^0) \setminus bot_{U_{6k+5}}(\Pi^0), X_0 \cup \dots \cup X_{6k+5}) = \{consumesmore(p, k)\text{:-.}|$ $\{holds(p, q, k), tot\_decr(p, q1, k)\} \subseteq X_0 \cup \dots \cup X_{6k+5}, q1 > q\} \cup \{holds(p, q, k+1)\text{:-.}, |\{holds(p, q1, k), tot\_incr(p, q2, k),$ $tot\_decr(p, q3, k)\} \subseteq X_0 \cup \dots \cup X_{6k+5}, q = q1 + q2 - q3\}$. It's answer set is $X_{6k+6} = A_{15}^{ts=k} \cup A_{17}^{ts=k}$ – using forced atom proposition.

    (a) where, $holds(p, q, k+1)$ represents the marking of place p in the next time step due to firing T_k,

    (b) and, $consumesmore(p, k)$ represents $\exists p : q = M_k(p), q1 = \sum_{t \in T_k, p \in \bullet t} W(p, t)$, $q1 > q$ indicating place p that will be overconsumed if T_k is fired as defined in definition 14 (conflicting transitions)

14. $eval_{U_{6k+6}}(bot_{U_{6k+7}}(\Pi^0) \setminus bot_{U_{6k+6}}(\Pi^0), X_0 \cup \dots \cup X_{6k+6}) = \{consumesmore\text{:-.}|$ $\{consumesmore(p, 0), \dots, consumesmore(p, k)\} \cap (X_0 \cup \dots \cup X_{6k+7}) \neq \emptyset\}$. It's answer set is $X_{6k+7} = A_{16}^{ts=k}$ – using forced atom proposition

    (a) X_{6k+7} will be empty since none of $consumesmore(p, 0), \dots,$ $consumesmore(p, k)$ hold in $X_0 \cup \dots \cup X_{6k+6}$ due to the construction of A and encoding of $a2$, and it is not eliminated by the constraint $a4$.

The set $X = X_0 \cup \dots \cup X_{6k+7}$ is the answer set of Π^0 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 6k+7$ matches a distinct partition of A, and $X = A$, thus A is an answer set of Π^0.

Next we show (II): Given Π^0 be the encoding of a Petri Net $PN(P, T, E, W)$ with initial marking M_0, and A be an answer set of Π^0 that satisfies (A.1) and (A.2), then we can construct $X = M_0, T_0, \dots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \dots, M_i(p_n))$, where $\{holds(p_0, M_i(p_0), i), \dots holds(p_n, M_i(p_n), i)\}$ $\subseteq A$, for $0 \leq i \leq k+1$

2. $T_i = \{t : fires(t, i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, show (1) T_0 is a valid firing set for M_0, and (2) T_0's firing in M_0 producing marking M_1.

1. We show T_0 is a valid firing set w.r.t. M_0. Let $\{fires(t_0, 0), \dots, fires(t_x, 0)\}$ be the set of all $fires(\dots, 0)$ atoms in A,
   (a) Then for each $fires(t_i, 0) \in A$
      i. $enabled(t_i, 0) \in A$ – from rule $a1$ and supported rule proposition
      ii. Then $notenabled(t_i, 0) \notin A$ – from rule $e2$ and supported rule proposition
      iii. Then $body(e1)$ must not hold in A – from rule $e1$ and forced atom proposition
      iv. Then $q \not< n_i \equiv q \geq n_i$ in $e1$ for all $\{holds(p, q, 0), ptarc(p, t_i, n_i)\} \subseteq A$ – from $e1$, forced atom proposition, and the following:
         A. $holds(p, q, 0) \in A$ represents $q = M_0(p)$ – rule $i1$ construction
         B. $ptarc(p, t_i, n_i) \in A$ represents $n_i = W(p, t_i)$ – rule $f3$ construction
      v. Then $\forall p \in \bullet t_i, M_0(p) > W(p, t_i)$ – from definition 11 of preset $\bullet t_i$ in PN
      vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 12 of enabled transition
   (b) And $consumesmore \notin A$, since A is an answer set of Π^0 – from rule $a4$ and supported rule proposition
      i. Then $\nexists consumesmore(p, 0) \in A$ – from rule $a3$ and supported rule proposition
      ii. Then $\nexists \{holds(p, q, 0), tot\_decr(p, q1, 0)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition
      iii. Then $\nexists p : \sum_{t_i \in \{t_0, \dots, t_x\}, p \in \bullet t_i} W(p, t_i) > M_0(p)$ – from the following
         A. $holds(p, q, 0)$ represents $q = M_0(p)$ – from rule $i1$ construction, given
         B. $tot\_decr(p, q1, 0) \in A$ if $\{del(p, q1_0, t_0, 0), \dots del(p, q1_x, t_x, 0)\} \subseteq A$, where $q1 = q1_0 + \cdots + q1_x$ – from $r4$ and forced atom proposition
         C. $del(p, q1_i, t_i, 0) \in A$ if $\{fires(t_i, 0), ptarc(p, t_i, q1_i)\} \subseteq A$ – from $r2$ and supported rule proposition
         D. $del(p, q1_i, t_i, 0) \in A$ represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$ – from $r2$, supported rule proposition, and definition 13 of transition execution in PN
      iv. Then the set of transitions in T_0 do not conflict – by the definition 14 of conflicting transitions
   (c) Then $\{t_0, \dots, t_x\} = T_0$ – using 1(a),1(b) above, and definition of firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $\mathit{holds}(p, q, 1) \in A$

   (a) Then $\{\mathit{holds}(p, q1, 0), \mathit{tot\_incr}(p, q2, 0), \mathit{tot\_decr}(p, q3, 0)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

   (b) Then $\mathit{holds}(p, q1, 0) \in A$ represents $q1 = M_0(p)$ – given, rule $i1$ construction; and $\{\mathit{add}(p, q_0, t_0, 0), \dots,\ \mathit{add}(p, q_j, t_j, 0)\} \subseteq A : q_0 + \cdots + q_j = q2$; and $\{\mathit{del}(p, q_0, t_0, 0), \dots,\ \mathit{del}(p, q_l, t_l, 0)\} \subseteq A : q_0 + \cdots + q_l = q3$ – rules $r3, r4$ and supported rule proposition, respectively

   (c) Then $\{\mathit{fires}(t_0, 0), \dots, \mathit{fires}(t_j, 0)\} \subseteq A$ and $\{\mathit{fires}(t_0, 0), \dots, \mathit{fires}(t_l, 0)\} \subseteq A$ – rules $r1, r2$ and supported rule proposition; and the following

      i. $\mathit{tparc}(t_y, p, q_y) \in A, 0 \leq y \leq j$ represents $q_y = W(t_y, p)$ – given

      ii. $\mathit{ptarc}(p, t_z, q_z) \in A, 0 \leq z \leq l$ represents $q_z = W(p, t_z)$ – given

   (d) Then $\{\mathit{fires}(t_0, 0), \dots, \mathit{fires}(t_j, 0)\} \cup \{\mathit{fires}(t_0, 0), \dots, \mathit{fires}(t_l, 0)\} \subseteq A$ $= \{\mathit{fires}(t_0, 0), \dots, \mathit{fires}(t_x, 0)\} \subseteq A$ – set union of subsets

   (e) Then for each $\mathit{fires}(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

   (f) Then $q = M_0(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x)$ – from (2b) above and the following

      i. Each $\mathit{add}(p, q_j, t_j, 0) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r1$ encoding, and definition 13 of transition execution in PN

      ii. Each $\mathit{del}(p, t_y, q_y, 0) \in A$ represents $q_y = W(p, t_y)$ for $p \in \bullet t_y$ – from rule $r2$ encoding, and definition 13 of transition execution in PN

      iii. Each $\mathit{tot\_incr}(p, q2, 0) \in A$ represents $q2 = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

      iv. Each $\mathit{tot\_decr}(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x)$ – aggregate assignment atom semantics in rule $r4$

   (g) Then, $M_1(p) = q$ – since $\mathit{holds}(p, q, 1) \in A$ encodes $q = M_1(p)$ – by construction

Inductive Step: Assume M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set for M_k. Let $\{\mathit{fires}(t_0, k), \dots, \mathit{fires}(t_x, k)\}$ be the set of all $\mathit{fires}(\dots, k)$ atoms in A,

   (a) Then for each $\mathit{fires}(t_i, k) \in A$

      i. $\mathit{enabled}(t_i, k) \in A$ – from rule $a1$ and supported rule proposition

      ii. Then $\mathit{notenabled}(t_i, k) \notin A$ – from rule $e2$ and supported rule proposition

      iii. Then body of $e1$ must hold in A – from rule $e1$ and forced proposition

      iv. Then $q \not< n_i \equiv q \geq n_i$ in $e1$ for all $\{\mathit{holds}(p, q, k), \mathit{ptarc}(p, t_i, n_i)\} \subseteq A$ – from $e1$ using forced atom proposition, and the following

A. $holds(p, q, k) \in A$ represents $q = M_k(p)$ – construction, inductive assumption

B. $ptarc(p, t, n_i) \in A$ represents $n_i = W(p, t)$ – rule $f4$ construction

v. Then $\forall p \in \bullet t_i, M_k(p) > W(p, t_i)$ – from definition 11 of preset $\bullet t_i$ in PN

vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 12 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^0 – from rule $a4$ and supported rule proposition

i. Then $\nexists consumesmore(p, k) \in A$ – from rule $a3$ and forced atom proposition

ii. Then $\nexists \{holds(p, q, k), tot\_decr(p, q1, k)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition

iii. Then $\nexists p : \sum_{t_i \in \{t_0, \dots, t_x\}, p \in \bullet t_i} W(p, t_i) > M_k(p)$ – from the following

A. $holds(p, q, k) \in A$ represents $q = M_k(p)$ – by construction, and the inductive assumption

B. $tot\_decr(p, q1, k) \in A$ if $\{del(p, q1_0, t_0, k), \dots, del(p, q1_x, t_x, k)\} \subseteq A$, where $q1 = q1_0 + \dots + q1_x$ – from $r4$ and forced atom proposition

C. $del(p, q1_i, t_i, k) \in A$ if $\{fires(t_i, k), ptarc(p, t_i, q1_i)\} \subseteq A$ – from $r2$ and supported rule proposition

D. $del(p, q1_i, t_i, k)$ represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$ – from construction rule $r2$, supported rule proposition, and definition 13 of transition execution in PN

iv. Then the set of transitions T_k does conflict – by the definition 14 of conflicting transitions

(c) Then $\{t_0, \dots, t_x\} = T_k$ – using 1(a),1(b) above

2. We show M_{k+1} is produced by firing T_k in M_k. Let $holds(p, q, k+1) \in A$

(a) Then $\{holds(p, q1, k), tot\_incr(p, q2, k), tot\_decr(p, q3, k)\} \in A : q = q2 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) Then $holds(p, q1, k) \in A$ represents $q1 = M_k(p)$ – inductive assumption and construction; and $\{add(p, q2_0, t_0, k), \dots, add(p, q2_j, t_j, k)\} \subseteq A : q2_0 + \dots + q2_j = q2$ and $\{del(p, q3_0, t_0, k), \dots, del(p, q3_l, t_l, k)\} \subseteq A : q3_0 + \dots + q3_l = q3$ – from rules $r3, r4$ using supported rule proposition, respectively

(c) Then $\{fires(t_0, k), \dots, fires(t_j, k)\} \subseteq A$ and $\{fires(t_0, k), \dots, fires(t_l, k)\} \subseteq A$ – from rules $r1, r2$ using supported rule proposition, respectively

(d) Then $\{fires(t_0, k), \dots, fires(t_j, k)\} \cup \{fires(t_0, k), \dots, fires(t_l, k)\} = \{fires(t_0, k), \dots, fires(t_x, k)\} \subseteq A$ – subset union property

(e) Then for each $fires(t_x, k) \in A$ we have $t_x \in T_k$ - already shown in item 1 above

(f) Then $q = M_k(p) + \sum_{t_x \in T_k \wedge p \in t_x \bullet} W(t_x, p) - \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x$ – from (2b) above and the following

i. Each $add(p, q_j, t_j, 0) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j\bullet$ – encoding of $r1$ and definition 13 of transition execution in PN
ii. Each $del(p, t_y, q_y, 0) \in A$ represents $q_y = W(p, t_y)$ for $p \in \bullet t_y$ – encoding of $r2$ and definition 13 of transition execution in PN
iii. Each $tot\_incr(p, q2, 0) \in A$ represents $q2 = \sum_{t_x \in T_k \wedge p \in t_x\bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$
iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x)$ – aggregate assignment atom semantics in rule $r4$

(g) Then $M_{k+1}(p) = q$ – since $holds(p, q, k+1) \in A$ encodes $q = M_{k+1}(p)$ by construction

As a result, for any $n > k$, T_n is a valid firing set w.r.t. M_n and its firing produces marking M_{n+1}.

Conclusion: Since both I and II hold, $X = M_0, T_0, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of $PN(P, T, E, W)$ (w.r.t. M_0) iff there is an answer set A of $\Pi^0(PN, M_0, k, ntok)$ such that (A.1) and (A.2) hold.

A.2 Proof of Proposition 2

Let $PN = (P, T, E, W)$ be a Petri Net, M_0 be its initial marking and let $\Pi^1(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.4. Then $X = M_0, T_0, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t. M_0) iff there is an answer set A of $\Pi^1(PN, M_0, k, ntok)$ such that:

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\} \tag{A.3}$$

$$\begin{aligned} \{holds(p, q, ts) : p \in P, q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ = \{holds(p, q, ts) : holds(p, q, ts) \in A\} \end{aligned} \tag{A.4}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.3) and (A.4) and show that A is an answer set of Π^1

(II) Given an answer set A of Π^1, we create an execution sequence X such that (A.3) and (A.4) are satisfied.

First we show (I): Given a PN and an execution sequence X of PN, we create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$
2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$
3. $A_3 = \{place(p) : p \in P\}$
4. $A_4 = \{trans(t) : t \in T\}$

5. $A_5 = \{ptarc(p,t,n) : (p,t) \in E^-, n = W(p,t)\}$, where $E^- \subseteq E$

6. $A_6 = \{tparc(t,p,n) : (t,p) \in E^+, n = W(t,p)\}$, where $E^+ \subseteq E$

7. $A_7 = \{holds(p,q,0) : p \in P, q = M_0(p)\}$

8. $A_8 = \{notenabled(t,ts) : t \in T, 0 \leq ts \leq k, \exists p \in \bullet t, M_{ts}(p) < W(p,t)\}$
   per definition 12 (enabled transition)

9. $A_9 = \{enabled(t,ts) : t \in T, 0 \leq ts \leq k, \forall p \in \bullet t, W(p,t) \leq M_{ts}(p)\}$
   per definition 12 (enabled transition)

10. $A_{10} = \{fires(t,ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
    per definition 15 (firing set), only an enabled transition may fire

11. $A_{11} = \{add(p,q,t,ts) : t \in T_{ts}, p \in t\bullet, q = W(t,p), 0 \leq ts \leq k\}$
    per definition 13 (transition execution)

12. $A_{12} = \{del(p,q,t,ts) : t \in T_{ts}, p \in \bullet t, q = W(p,t), 0 \leq ts \leq k\}$
    per definition 13 (transition execution)

13. $A_{13} = \{tot\_incr(p,q,ts) : p \in P, q = \sum_{t \in T_{ts}, p \in t\bullet} W(t,p), 0 \leq ts \leq k\}$
    per definition 15 (firing set execution)

14. $A_{14} = \{tot\_decr(p,q,ts) : p \in P, q = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t), 0 \leq ts \leq k\}$
    per definition 15 (firing set execution)

15. $A_{15} = \{consumesmore(p,ts) : p \in P, q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t), q1 > q, 0 \leq ts \leq k\}$
    per definition 14 (conflicting transitions) for enabled transition set T_{ts}

16. $A_{16} = \{consumesmore : \exists p \in P : q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t), q1 > q, 0 \leq ts \leq k\}$
    per definition 14 (conflicting transitions)

17. $A_{17} = \{could\_not\_have(t,ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), t \notin T_{ts}, (\exists p \in \bullet t : W(p,t) > M_{ts}(p) - \sum_{t' \in T_{ts}, p \in \bullet t'} W(p,t')), 0 \leq ts \leq k\}$
    per the maximal firing set semantics

18. $A_{18} = \{holds(p,q,ts+1) : p \in P, q = M_{ts+1}(p), 0 \leq ts < k\}$,
    where $M_{ts+1}(p) = M_{ts}(p) - \sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in t\bullet} W(t,p)$
    according to definition 16 (firing set execution)

We show that A satisfies (A.3) and (A.4), and A is an answer set of Π^1.

A satisfies (A.3) and (A.4) by its construction above. We show A is an answer set of Π^1 by splitting. We split $lit(\Pi^1)$ into a sequence of $6k+8$ sets:

- $U_0 = head(f1) \cup head(f2) \cup head(f3) \cup head(f4) \cup head(f5) \cup head(f6) \cup head(i1) = \{place(p) : p \in P\} \cup \{trans(t) : t \in T\} \cup \{ptarc(p,t,n) : (p,t) \in E^-, n = W(p,t)\} \cup \{tparc(t,p,n) : (t,p) \in E^+, n = W(t,p)\} \cup \{time(0), \ldots, time(k)\} \cup \{num(0), \ldots, num(ntok)\} \cup \{holds(p,q,0) : p \in P, q = M_0(p)\}$

- $U_{6k+1} = U_{6k+0} \cup head(e1)^{ts=k} = U_{6k+0} \cup \{notenabled(t,k) : t \in T\}$
- $U_{6k+2} = U_{6k+1} \cup head(e2)^{ts=k} = U_{6k+1} \cup \{enabled(t,k) : t \in T\}$
- $U_{6k+3} = U_{6k+2} \cup head(a1)^{ts=k} = U_{6k+2} \cup \{fires(t,k) : t \in T\}$
- $U_{6k+4} = U_{6k+3} \cup head(r1)^{ts=k} \cup head(r2)^{ts=k} = U_{6k+3} \cup \{add(p,q,t,k) : p \in P, t \in T, q = W(t,p)\} \cup \{del(p,q,t,k) : p \in P, t \in T, q = W(p,t)\}$
- $U_{6k+5} = U_{6k+4} \cup head(r3)^{ts=k} \cup head(r4)^{ts=k} = U_{6k+4} \cup \{tot\_incr(p,q,k) : p \in P, 0 \leq q \leq ntok\} \cup \{tot\_decr(p,q,k) : p \in P, 0 \leq q \leq ntok\}$
- $U_{6k+6} = U_{6k+5} \cup head(r5)^{ts=k} \cup head(a2)^{ts=k} \cup head(a5)^{ts=k} = U_{6k+5}$ $\cup \{consumesmore(p,k) : p \in P\} \cup \{holds(p,q,k+1) : p \in P, 0 \leq q \leq ntok\} \cup$ $\{could\_not\_have(t,k) : t \in T\}$
- $U_{6k+7} = U_{6k+6} \cup head(a3) = U_{6k+6} \cup \{consumesmore\}$

where $head(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\dots,ts) : a(\dots,ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 6k+7$ form a splitting sequence, since each U_i is a splitting set of Π^1, and $\langle U_\alpha \rangle_{\alpha<\mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{6k+7}$ and $\bigcup_{\alpha<\mu} U_\alpha = lit(\Pi^1)$.

We compute the answer set of Π^1 using the splitting sets as follows:

1. $bot_{U_0}(\Pi^1) = f1 \cup f2 \cup f3 \cup f4 \cup f5 \cup i1 \cup f6$ and $X_0 = A_1 \cup \dots \cup A_7$ $(= U_0)$ is its answer set – using forced atom proposition
2. $eval_{U_0}(bot_{U_1}(\Pi^1) \setminus bot_{U_0}(\Pi^1), X_0) = \{notenabled(t,0)\text{:-}.|\{trans(t), ptarc(p,t,n), holds(p,q,0)\} \subseteq X_0,$ where $q < n\}$. Its answer set $X_1 = A_8^{ts=0}$ – using forced atom proposition and construction of A_8.
   (a) where, $q = M_0(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of $i1$ and $f3$ in Π^1, and
   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)
   (c) thus, $notenabled(t,0) \in X_1$ represents $\exists p \in \bullet t : M_0(p) < W(p,t)$.
3. $eval_{U_1}(bot_{U_2}(\Pi^1) \setminus bot_{U_1}(\Pi^1), X_0 \cup X_1) = \{enabled(t,0)\text{:-}.|trans(t) \in X_0 \cup X_1, notenabled(t,0) \notin X_0 \cup X_1\}$. Its answer set is $X_2 = A_9^{ts=0}$ – using forced atom proposition and construction of A_9.
   (a) since an $enabled(t,0) \in X_2$ if $\nexists$ $notenabled(t,0) \in X_0 \cup X_1$, which is equivalent to $\nexists p \in \bullet t : M_0(p) < W(p,t) \equiv \forall p \in \bullet t : M_0(p) \geq W(p,t)$.
4. $eval_{U_2}(bot_{U_3}(\Pi^1) \setminus bot_{U_2}(\Pi^1), X_0 \cup X_1 \cup X_2) = \{\{fires(t,0)\}\text{:-}.|enabled(t,0)$ holds in $X_0 \cup X_1 \cup X_2\}$. It has multiple answer sets $X_{3.1}, \dots, X_{3.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $eval_{U_2}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^1 determined using splitting is equal to A, we only consider the set that matches the $fires(t,0)$ elements in A and call it X_3, ignoring the rest. Thus, $X_3 = A_{10}^{ts=0}$, representing T_0.

5. $eval_{U_3}(bot_{U_4}(\Pi^1) \setminus bot_{U_3}(\Pi^1), X_0 \cup \cdots \cup X_3) = \{add(p,n,t,0)\text{:-.}|\{fires(t,0), tparc(t,p,n)\} \subseteq X_0 \cup \cdots \cup X_3\} \cup \{del(p,n,t,0)\text{:-.}|\{fires(t,0), ptarc(p,t,n)\} \subseteq X_0 \cup \cdots \cup X_3\}$. It's answer set is $X_4 = A_{11}^{ts=0} \cup A_{12}^{ts=0}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

    (a) where, each add atom encodes $n = W(t,p) : p \in t\bullet$,

    (b) and each del atom encodes $n = W(p,t) : p \in \bullet t$

    (c) representing the effect of transitions in T_0 – by construction

6. $eval_{U_4}(bot_{U_5}(\Pi^1) \setminus bot_{U_4}(\Pi^1), X_0 \cup \cdots \cup X_4) = \{tot\_incr(p,qq,0)\text{:-.}| qq = \sum_{add(p,q,t,0) \in X_0 \cup \cdots \cup X_4} q\} \cup \{tot\_decr(p,qq,0)\text{:-.}|qq = \sum_{del(p,q,t,0) \in X_0 \cup \cdots \cup X_4} q\}$. It's answer set is $X_5 = A_{13}^{ts=0} \cup A_{14}^{ts=0}$ – using forced atom proposition and definitions of A_{13}, A_{14}, and semantics of aggregate assignment atom

    (a) where, each $tot\_incr(p,qq,0)$, $qq = \sum_{add(p,q,t,0) \in X_0 \cup \ldots X_4} q$
    $\equiv qq = \sum_{t \in X_3, p \in t\bullet} W(p,t)$,

    (b) and, each $tot\_decr(p,qq,0)$, $qq = \sum_{del(p,q,t,0) \in X_0 \cup \ldots X_4} q$
    $\equiv qq = \sum_{t \in X_3, p \in \bullet t} W(t,p)$,

    (c) represent the net effect of actions in T_0 – by construction

7. $eval_{U_5}(bot_{U_6}(\Pi^1) \setminus bot_{U_5}(\Pi^1), X_0 \cup \cdots \cup X_5) = \{consumesmore(p,0)\text{:-.}| \{holds(p,q,0), tot\_decr(p,q1,0)\} \subseteq X_0 \cup \cdots \cup X_5 : q1 > q\} \cup \{holds(p,q,1)\text{:-.},| \{holds(p,q1,0), tot\_incr(p,q2,0), tot\_decr(p,q3,0)\} \subseteq X_0 \cup \cdots \cup X_5, q = q1 + q2 - q3\} \cup \{could\_not\_have(t,0)\text{:-.}|\{enabled(t,0), ptarc(s,t,q), holds(s,qq,0), tot\_decr(s,qqq,0)\} \subseteq X_0 \cup \cdots \cup X_5, fires(t,0) \notin (X_0 \cup \cdots \cup X_5), q > qq - qqq\}$. It's answer set is $X_6 = A_{15}^{ts=0} \cup A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of A_{15}, A_{17}, A_{18}.

    (a) where, $consumesmore(p,0)$ represents $\exists p : q = M_0(p), q1 = \sum_{t \in T_0, p \in \bullet t} W(p,t)$, $q1 > q$ indicating place p will be overconsumed if T_0 is fired, as defined in definition 14 (conflicting transitions)

    (b) and, $holds(p,q,1)$ encodes $q = M_1(p)$ – by construction

    (c) and $could\_not\_have(t,0)$ represents an enabled transition t in T_0 that could not fire due to insufficient tokens

    (d) X_6 does not contain $could\_not\_have(t,0)$, when $enabled(t,0) \in X_0 \cup \cdots \cup X_5$ and $fires(t,0) \notin X_0 \cup \cdots \cup X_5$ due to construction of A, encoding of $a5$ and its body atoms. As a result it is not eliminated by the constraint $a6$

$$\vdots$$

8. $eval_{U_{6k+0}}(bot_{U_{6k+1}}(\Pi^1) \setminus bot_{U_{6k+0}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+0}) = \{notenabled(t,k)\text{:-.}|\{trans(t), ptarc(p,t,n), holds(p,q,k)\} \subseteq X_0 \cup \cdots \cup X_{6k+0}$, where $q < n\}$. Its answer set $X_{6k+1} = A_8^{ts=k}$ – using forced atom proposition and construction of A_8.

(a) where, $q = M_k(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of *holds* and *ptarc* predicates in Π^1, and

(b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

(c) thus, $notenabled(t,k) \in X_{6k+1}$ represents $\exists p \in \bullet t : M_k(p) < W(p,t)$.

9. $eval_{U_{6k+1}}(bot_{U_{6k+2}}(\Pi^1) \setminus bot_{U_{6k+1}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+1}) = \{enabled(t,k)\text{:-}.| trans(t) \in X_0 \cup \cdots \cup X_{6k+1}, notenabled(t,k) \notin X_0 \cup \cdots \cup X_{6k+1}\}$. Its answer set is $X_{6k+2} = A_9^{ts=k}$ – using forced atom proposition and construction of A_9.

   (a) since an $enabled(t,k) \in X_{6k+2}$ if $\nexists$ $notenabled(t,k) \in X_0 \cup \cdots \cup X_{6k+1}$, which is equivalent to $\nexists p \in \bullet t : M_k(p) < W(p,t) \equiv \forall p \in \bullet t : M_k(p) \geq W(p,t)$.

10. $eval_{U_{6k+2}}(bot_{U_{6k+3}}(\Pi^1) \setminus bot_{U_{6k+2}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+2}) = \{\{fires(t,k)\}\text{:-}.|enabled(t,k)$ holds in $X_0 \cup \cdots \cup X_{6k+2}\}$. It has multiple answer sets $X_{6k+3.1}, \ldots, X_{6k+3.n}$, corresponding to elements of power set of $fires(t,k)$ atoms in $eval_{U_{6k+2}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^1 determined using splitting is equal to A, we only consider the set that matches the $fires(t,k)$ elements in A and call it X_{6k+3}, ignoring the reset. Thus, $X_{6k+3} = A_{10}^{ts=k}$, representing T_k.

11. $eval_{U_{6k+3}}(bot_{U_{6k+4}}(\Pi^1) \setminus bot_{U_{6k+3}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+3}) = \{add(p,n,t,k)\text{:-}.|\{fires(t,k), tparc(t,p,n)\} \subseteq X_0 \cup \cdots \cup X_{6k+3}\} \cup \{del(p,n,t,k)\text{:-}.|\{fires(t,k), ptarc(p,t,n)\} \subseteq X_0 \cup \cdots \cup X_{6k+3}\}$. It's answer set is $X_{6k+4} = A_{11}^{ts=k} \cup A_{12}^{ts=k}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

   (a) where, each *add* atom is equivalent to $n = W(t,p) : p \in t\bullet$,

   (b) and, each *del* atom is equivalent to $n = W(p,t) : p \in \bullet t$,

   (c) representing the effect of transitions in T_k

12. $eval_{U_{6k+4}}(bot_{U_{6k+5}}(\Pi^1) \setminus bot_{U_{6k+4}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+4}) = \{tot\_incr(p,qq,k)\text{:-}.| qq = \sum_{add(p,q,t,k) \in X_0 \cup \cdots \cup X_{6k+4}} q\} \cup \{tot\_decr(p,qq,k)\text{:-}.| qq = \sum_{del(p,q,t,k) \in X_0 \cup \cdots \cup X_{6k+4}} q\}$. It's answer set is $X_{6k+5} = A_{13}^{ts=k} \cup A_{14}^{ts=k}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

   (a) where, each $tot\_incr(p,qq,k)$, $qq = \sum_{add(p,q,t,k) \in X_0 \cup \ldots X_{6k+4}} q \equiv$ $qq = \sum_{t \in X_{6k+3}, p \in t\bullet} W(p,t)$,

   (b) and, each $tot\_decr(p,qq,k)$, $qq = \sum_{del(p,q,t,k) \in X_0 \cup \ldots X_{6k+4}} q \equiv$ $qq = \sum_{t \in X_{6k+3}, p \in \bullet t} W(t,p)$,

   (c) represent the net effect of transitions in T_k

13. $eval_{U_{6k+5}}(bot_{U_{6k+6}}(\Pi^1) \setminus bot_{U_{6k+5}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+5}) = \{consumesmore(p,k)\text{:-}.|\{holds(p,q,k), tot\_decr(p,q1,k)\} \subseteq X_0 \cup \cdots \cup X_{6k+5} : q1 > q\} \cup \{holds(p,q,k+1)\text{:-}., |\{holds(p,q1,k), tot\_incr(p,q2,k), tot\_decr(p,q3,k)\} \subseteq X_0 \cup \cdots \cup X_{6k+5} : q = q1 + q2 - q3\} \cup \{could\_not\_have(t,k)\text{:-}$

$\{enabled(t,k), ptarc(s,t,q), holds(s,qq,k), tot\_decr(s,qqq,k)\} \subseteq X_0 \cup \cdots \cup X_{6k+5}$, $fires(t,k) \notin (X_0 \cup \cdots \cup X_{6k+5}), q > qq - qqq\}$. It's answer set is $X_{6k+6} = A_{15}^{ts=k} \cup A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition.

(a) where, $consumesmore(p,k)$ represents $\exists p : q = M_k(p)$, $q1 = \sum_{t \in T_k, p \in \bullet t} W(p,t), q1 > q$

(b) $holds(p,q,k{+}1)$ represents $q = M_{k+1}(p)$ indicating place p that will be over consumed if T_k is fired, as defined in definition 14 (conflicting transitions),

(c) $holds(p,q,k+1)$ represents $q = M_{k+1}(p)$ – by construction

(d) and $could\_not\_have(t,k)$ represents an enabled transition t in T_k that could not fire due to insufficient tokens

(e) X_{6k+6} does not contain $could\_not\_have(t,k)$, when $enabled(t,k) \in X_0 \cup \cdots \cup X_{6k+5}$ and $fires(t,k) \notin X_0 \cup \cdots \cup X_{6k+5}$ due to construction of A, encoding of $a5$ and its body atoms. As a result it is note eliminated by the constraint $a6$

14. $eval_{U_{6k+6}}(bot_{U_{6k+7}}(\Pi^1) \setminus bot_{U_{6k+6}}(\Pi^1), X_0 \cup \cdots \cup X_{6k+6}) = \{consumesmore{:}\text{-}.| \{consumesmore(p,0), \ldots, consumesmore(p,k)\} \cap (X_0 \cup \cdots \cup X_{6k+6}) \neq \emptyset\}$. It's answer set is $X_{6k+7} = A_{16}$ – using forced atom proposition

    (a) X_{6k+7} will be empty since none of $consumesmore(p,0), \ldots,$ $consumesmore(p,k)$ hold in $X_0 \cup \cdots \cup X_{6k+6}$ due to the construction of A, encoding of $a2$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \cdots \cup X_{6k+7}$ is the answer set of Π^0 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 6k+7$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^1.

Next we show (II): Given Π^1 be the encoding of a Petri Net $PN(P,T,E,W)$ with initial marking M_0, and A be an answer set of Π^1 that satisfies (A.3) and (A.4), then we can construct $X = M_0, T_0, \ldots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \ldots, M_i(p_n))$, where $\{holds(p_0, M_i(p_0), i), \ldots holds(p_n, M_i(p_n), i)\} \subseteq A$, for $0 \leq i \leq k+1$

2. $T_i = \{t : fires(t,i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) T_0's firing w.r.t. marking M_0 produces M_1.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0,0),\dots,fires(t_x,0)\}$ be the set of all $fires(\dots,0)$ atoms in A,
   (a) Then for each $fires(t_i,0) \in A$
      i. $enabled(t_i,0) \in A$ – from rule $a1$ and supported rule proposition
      ii. Then $notenabled(t_i,0) \notin A$ – from rule $e2$ and supported rule proposition
      iii. Then $body(e1)$ must not hold in A – from rule $e1$ and forced atom proposition
      iv. Then $q \not< n_i \equiv q \geq n_i$ in $e1$ for all $\{holds(p,q,0), ptarc(p,t_i,n_i)\} \subseteq A$ – from $e1$, forced atom proposition, and the following
         A. $holds(p,q,0) \in A$ represents $q = M_0(p)$ – rule $i1$ construction
         B. $ptarc(p,t_i,n_i) \in A$ represents $n_i = W(p,t_i)$ – rule $f3$ construction
      v. Then $\forall p \in \bullet t_i, M_0(p) > W(p,t_i)$ – from definition 11 of preset $\bullet t_i$ in PN
      vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 12 of enabled transition
   (b) And $consumesmore \notin A$, since A is an answer set of Π^1 – from rule $a4$ and supported rule proposition
      i. Then $\nexists consumesmore(p,0) \in A$ – from rule $a3$ and supported rule proposition
      ii. Then $\nexists\{holds(p,q,0), tot\_decr(p,q1,0)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition
      iii. Then $\nexists p : \sum_{t_i \in \{t_0,\dots,t_x\}, p \in \bullet t_i} W(p,t_i) > M_0(p)$ – from the following
         A. $holds(p,q,0)$ represents $q = M_0(p)$ – from rule $i1$ construction, given
         B. $tot\_decr(p,q1,0) \in A$ if $\{del(p,q1_0,t_0,0),\dots,del(p,q1_x,t_x,0)\} \subseteq A$, where $q1 = q1_0 + \dots + q1_x$ – from $r4$ and forced atom proposition
         C. $del(p,q1_i,t_i,0) \in A$ if $\{fires(t_i,0), ptarc(p,t_i,q1_i)\} \subseteq A$ – from $r2$ and supported rule proposition
         D. $del(p,q1_i,t_i,0)$ represents removal of $q1_i = W(p,t_i)$ tokens from $p \in \bullet t_i$ – from rule $r2$, supported rule proposition, and definition 13 of transition execution in PN
      iv. Then the set of transitions in T_0 do not conflict – by the definition 14 of conflicting transitions
   (c) And for each $enabled(t_j,0) \in A$ and $fires(t_j,0) \notin A$, $could\_not\_have(t_j,0) \in A$, since A is an answer set of Π^1 - from rule $a6$ and supported rule proposition
      i. Then $\{enabled(t_j,0), holds(s,qq,0), ptarc(s,t_j,q,0),$
      $tot\_decr(s,qqq,0)\} \subseteq A$, such that $q > qq - qqq$ and $fires(t_j,0) \notin A$ - from rule $a5$ and supported rule proposition
      ii. Then for an $s \in \bullet t_j$, $W(s,t_j) > M_0(s) - \sum_{t_i \in T_0, s \in \bullet t_i} W(s,t_i)$ - from the following:

A. $ptarc(s, t_i, q)$ represents $q = W(s, t_i)$ – from rule $f7$ construction

B. $holds(s, qq, 0)$ represents $qq = M_0(s)$ – from $i1$ construction

C. $tot\_decr(s, qqq, 0) \in A$ if $\{del(s, qqq_0, t_0, 0), \ldots, del(s, qqq_x, t_x, 0)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

D. $del(s, qqq_i, t_i, 0) \in A$ if $\{fires(t_i, 0), ptarc(s, t_i, qqq_i)\} \subseteq A$ – from rule $r7$ and supported rule proposition

E. $del(s, qqq_i, t_i, 0)$ represents $qqq_i = W(s, t_i) : t_i \in T_0, (s, t_i) \in E^-$ – from rule $f7$ construction

F. $tot\_decr(q, qqq, 0)$ represents $\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i)$ – from (C,D,E) above

iii. Then firing $T_0 \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j$. Thus, T_0 is a maximal set of transitions that can simultaneously fire.

(d) Then $\{t_0, \ldots, t_x\} = T_0$ – using 1(a),1(b) above; and using 1(c) it is a maximal firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $holds(p, q, 1) \in A$

(a) Then $\{holds(p, q1, 0), tot\_incr(p, q2, 0), tot\_decr(p, q3, 0)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) Then, $holds(p, q1, 0) \in A$ represents $q1 = M_0(p)$ – given, rule $i1$ construction; and $\{add(p, q2_0, t_0, 0), \ldots, add(p, q2_j, t_j, 0)\} \subseteq A : q2_0 + \cdots + q2_j = q2$ and $\{del(p, q3_0, t_0, 0), \ldots, del(p, q3_l, t_l, 0)\} \subseteq A : q3_0 + \cdots + q3_l = q3$ – rules $r3, r4$ using supported rule proposition

(c) Then $\{fires(t_0, 0), \ldots, fires(t_j, 0)\} \subseteq A$ and $\{fires(t_0, 0), \ldots, fires(t_l, 0)\} \subseteq A$ – rules $r1, r2$ and supported rule proposition, respectively

(d) Then $\{fires(t_0, 0), \ldots, fires(t_j, 0)\} \cup \{fires(t_0, 0), \ldots, fires(t_l, 0)\} \subseteq A = \{fires(t_0, 0), \ldots, fires(t_x, 0)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

(f) Then $q = M_0(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x)$ – from (2b) above and the following

i. Each $add(p, q_j, t_j, 0) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r1$ encoding, and definition 13 of transition execution in PN

ii. Each $del(p, t_y, q_y, 0) \in A$ represents $q_y = W(p, t_y)$ for $p \in \bullet t_y$ – from rule $r2$ encoding, and definition 13 of transition execution in PN

iii. Each $tot\_incr(p, q2, 0) \in A$ represents $q2 = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x)$ – aggregate assignment atom semantics in rule $r4$

(g) Then, $M_1(p) = q$ – since $holds(p, q, 1) \in A$ encodes $q = M_1(p)$ from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) T_k's firing in M_k produces marking M_{k+1}.

1. We show T_k is a valid firing set. Let $\{fires(t_0, k), \dots, fires(t_x, k)\}$ be the set of all $fires(\dots, k)$ atoms in A,

    (a) Then for each $fires(t_i, k) \in A$

    i. $\mathit{enabled}(t_i, k) \in A$ – from rule $a1$ and supported rule proposition

    ii. Then $\mathit{notenabled}(t_i, k) \notin A$ – from rule $e2$ and supported rule proposition

    iii. Then $\mathit{body}(e1)$ must hold in A – from rule $e1$ and forced atom proposition

    iv. Then $q \not< n_i \equiv q \geq n_i$ in $e1$ for all $\{holds(p, q, k), \mathit{ptarc}(p, t_i, n)\} \subseteq A$ – from $e1$, forced atom proposition, and the following

    A. $holds(p, q, k) \in A$ represents $q = M_k(p)$ – construction, inductive assumption

    B. $\mathit{ptarc}(p, t, n_i) \in A$ represents $n_i = W(p, t_i)$ – rule $f3$ construction

    v. Then $\forall p \in \bullet t_i$, $M_k(p) \geq W(p, t_i)$ – from definition 11 of preset $\bullet t_i$ in PN

    vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 12 of enabled transition

    (b) And $consumesmore \notin A$, since A is an answer set of Π^1 – from rule $a4$ and supported rule proposition

    i. Then $\nexists consumesmore(p, k) \in A$ – from rule $a3$ and forced atom proposition

    ii. Then $\nexists\{holds(p, q, k), tot\_decr(p, q1, k)\} \subseteq A : q1 > q$ in $\mathit{body}(a2)$ – from $a2$ and forced atom proposition

    iii. Then $\nexists p : \sum_{t_i \in \{t_0, \dots, t_x\}, p \in \bullet t_i} W(p, t_i) > M_k(p)$ – from the following

    A. $holds(p, q, k) \in A$ represents $q = M_k(p)$ – by construction of Π^1, and the inductive assumption about $M_k(p)$

    B. $tot\_decr(p, q1, k) \in A$ if $\{\mathit{del}(p, q1_0, t_0, k), \dots, \mathit{del}(p, q1_x, t_x, k)\} \subseteq A$, where $q1 = q1_0 + \cdots + q1_x$ – from $r4$ and forced atom proposition

    C. $\mathit{del}(p, q1_i, t_i, k) \in A$ if $\{fires(t_i, k), \mathit{ptarc}(p, t_i, q1_i)\} \subseteq A$ – from $r2$ and supported rule proposition

    D. $\mathit{del}(p, q1_i, t_i, k)$ represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$ – from construction rule $r2$, supported rule proposition, and definition 13 of transition execution in PN

    iv. Then the set of transitions T_k do not conf – by the definition 14 of conflicting transitions

    (c) And for each $\mathit{enabled}(t_j, k) \in A$ and $fires(t_j, k) \notin A$, $\mathit{could\_not\_have}(t_j, k) \in A$, since A is an answer set of Π^1 - from rule $a6$ and supported rule proposition

i. Then $\{\mathit{enabled}(t_j, k), \mathit{holds}(s, qq, k), \mathit{ptarc}(s, t_j, q, k), \mathit{tot\_decr}(s, qqq, k)\} \subseteq A$, such that $q > qq - qqq$ and $\mathit{fires}(t_j, 0) \notin A$ - from rule $a5$ and supported rule proposition

ii. Then for an $s \in \bullet t_j$, $W(s, t_j) > M_k(s) - \sum_{t_i \in T_k, s \in \bullet t_i} W(s, t_i)$ - from the following:

A. $\mathit{ptarc}(s, t_i, q)$ represents $q = W(s, t_i)$ – from rule $f7$ construction

B. $\mathit{holds}(s, qq, k)$ represents $qq = M_k(s)$ – from $i1$ construction

C. $\mathit{tot\_decr}(s, qqq, k) \in A$ if $\{\mathit{del}(s, qqq_0, t_0, k), \ldots, \mathit{del}(s, qqq_x, t_x, k)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

D. $\mathit{del}(s, qqq_i, t_i, k) \in A$ if $\{\mathit{fires}(t_i, k), \mathit{ptarc}(s, t_i, qqq_i)\} \subseteq A$ – from rule $r7$ and supported rule proposition

E. $\mathit{del}(s, qqq_i, t_i, k)$ represents $qqq_i = W(s, t_i) : t_i \in T_k, (s, t_i) \in E^-$ – from rule $f7$ construction

F. $\mathit{tot\_decr}(q, qqq, k)$ represents $\sum_{t_i \in T_k, s \in \bullet t_i} W(s, t_i)$ – from (C,D,E) above

iii. Then firing $T_k \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j$. Thus, T_k is a maximal set of transitions that can simultaneously fire.

(d) Then $\{t_0, \ldots, t_x\} = T_k$ – using 1(a),1(b) above; and using 1(c) it is a maximal firing set

2. We show M_{k+1} is produced by firing T_k in M_k. Let $\mathit{holds}(p, q, k+1) \in A$

(a) Then $\{\mathit{holds}(p, q1, k), \mathit{tot\_incr}(p, q2, k), \mathit{tot\_decr}(p, q3, k)\} \in A : q = q2 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) $\mathit{holds}(p, q1, k) \in A$ represents $q1 = M_k(p)$ – construction, inductive assumption; and $\{\mathit{add}(p, q2_0, t_0, k), \ldots, \mathit{add}(p, q2_j, t_j, k)\} \subseteq A : q2_0 + \cdots + q2_j = q2$ and $\{\mathit{del}(p, q3_0, t_0, k), \ldots, \mathit{del}(p, q3_l, t_l, k)\} \subseteq A : q3_0 + \cdots + q3_l = q3$ – from rules $r3, r4$ using supported rule proposition, respectively

(c) Then $\{\mathit{fires}(t_0, k), \ldots, \mathit{fires}(t_j, k)\} \subseteq A$ and $\{\mathit{fires}(t_0, k), \ldots, \mathit{fires}(t_l, k)\} \subseteq A$ – from rules $r1, r2$ using supported rule proposition, respectively

(d) Then $\{\mathit{fires}(t_0, k), \ldots, \mathit{fires}(t_j, k)\} \cup \{\mathit{fires}(t_0, k), \ldots, \mathit{fires}(t_l, k)\} = \{\mathit{fires}(t_0, k), \ldots, \mathit{fires}(t_x, k)\} \subseteq A$ – subset union property

(e) Then for each $\mathit{fires}(t_x, k) \in A$ we have $t_x \in T_k$ - already shown in item (1a) above

(f) Then $q = M_k(p) + \sum_{t_x \in T_k \wedge p \in t_x \bullet} W(t_x, p) - \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x)$ – from (2b) above and the following

i. Each $\mathit{add}(p, q_j, t_j, k) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r1$ encoding and definition 13 of transition execution in PN

ii. Each $\mathit{del}(p, t_y, q_y, k) \in A$ represents $q_y = W(p, t_y)$ for $p \in \bullet t_y$ – rule $r2$ encoding and definition 13 of transition execution in PN

iii. Each $\mathit{tot\_incr}(p, q2, k) \in A$ represents $q2 = \sum_{t_x \in T_k \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

iv. Each $tot\_decr(p, q3, k) \in A$ represents $q3 = \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x)$ – aggregate assignment atom semantics in rule $r4$

(g) Then $M_{k+1}(p) = q$ – since $holds(p, q, k+1) \in A$ encodes $q = M_{k+1}(p)$ – from construction

As a result, for any $n > k$, T_n will be a valid firing set w.r.t. M_n and its firing produces marking M_{n+1}.

Conclusion: Since both I and II hold, $X = M_0, T_0, M_1, \ldots, M_k, T_k, M_{k+1}$ is an execution sequence of $PN(P, T, E, W)$ (w.r.t. M_0) iff there is an answer set A of $\Pi^1(PN, M_0, k, ntok)$ such that (A.3) and (A.4) hold.

A.3 Proof of Proposition 3

Let $PN = (P, T, E, W, R)$ be a Petri Net, M_0 be its initial marking and let $\Pi^2(PN, M_0, k, ntok)$ by the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.5. Then $X = M_0, T_0, M_1, \ldots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t M_0) iff there is an answer set A of $\Pi^2(PN, M_0, k, ntok)$ such that:

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\} \tag{A.5}$$

$$\begin{aligned} &\{holds(p, q, ts) : p \in P, q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ &\qquad = \{holds(p, q, ts) : holds(p, q, ts) \in A\} \end{aligned} \tag{A.6}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.5) and (A.6) and show that A is an answer set of Π^2

(II) Given an answer set A of Π^2, we create an execution sequence X such that (A.5) and (A.6) are satisfied.

First we show (I): Given PN and an execution sequence X of PN, we create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$
2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$
3. $A_3 = \{place(p) : p \in P\}$
4. $A_4 = \{trans(t) : t \in T\}$
5. $A_5 = \{ptarc(p, t, n, ts) : (p, t) \in E^-, n = W(p, t), 0 \leq ts \leq k\}$, where $E^- \subseteq E$
6. $A_6 = \{tparc(t, p, n, ts) : (t, p) \in E^+, n = W(t, p), 0 \leq ts \leq k\}$, where $E^+ \subseteq E$
7. $A_7 = \{holds(p, q, 0) : p \in P, q = M_0(p)\}$

8. $A_8 = \{\mathit{notenabled}(t, ts) : t \in T, 0 \leq ts \leq k, \exists p \in \bullet t, M_{ts}(p) < W(p,t)\}$
   per definition 12 (enabled transition)

9. $A_9 = \{\mathit{enabled}(t, ts) : t \in T, 0 \leq ts \leq k, \forall p \in \bullet t, W(p,t) \leq M_{ts}(p)\}$
   per definition 12 (enabled transition)

10. $A_{10} = \{\mathit{fires}(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
    per definition 21 (firing set), only an enabled transition may fire

11. $A_{11} = \{\mathit{add}(p, q, t, ts) : t \in T_{ts}, p \in t\bullet, q = W(t,p), 0 \leq ts \leq k\}$
    per definition 22 (transition execution)

12. $A_{12} = \{\mathit{del}(p, q, t, ts) : t \in T_{ts}, p \in \bullet t, q = W(p,t), 0 \leq ts \leq k\} \cup \{\mathit{del}(p, q, t, ts) : t \in T_{ts}, p \in R(t), q = M_{ts}(p), 0 \leq ts \leq k\}$
    per definition 22 (transition execution)

13. $A_{13} = \{\mathit{tot\_incr}(p, q, ts) : p \in P, q = \sum_{t \in T_{ts}, p \in t\bullet} W(t,p), 0 \leq ts \leq k\}$
    per definition 24 (execution)

14. $A_{14} = \{\mathit{tot\_decr}(p, q, ts) : p \in P, q = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), 0 \leq ts \leq k\}$
    per definition 24 (execution)

15. $A_{15} = \{\mathit{consumesmore}(p, ts) : p \in P, q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), q1 > q, 0 \leq ts \leq k\}$
    per definition 23 (conflicting transitions)

16. $A_{16} = \{\mathit{consumesmore} : \exists p \in P : q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), q1 > q, 0 \leq ts \leq k\}$
    per definition 23 (conflicting transitions)

17. $A_{17} = \{\mathit{could\_not\_have}(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), t \notin T_{ts}, (\exists p \in \bullet t \cup R(t) : q > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p,t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), q = W(p,t)$ if $p \in \bullet t$ or $M_{ts}(p)$ otherwise $), 0 \leq ts \leq k\}$
    per the maximal firing set semantics

18. $A_{18} = \{\mathit{holds}(p, q, ts+1) : p \in P, q = M_{ts+1}(p), 0 \leq ts < k\}$, where $M_{ts+1}(p) = M_{ts}(p) - (\sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p)) + \sum_{t \in T_{ts}, p \in t\bullet} W(t,p)$
    according to definition 24 (firing set execution)

19. $A_{19} = \{\mathit{ptarc}(p, t, n, ts) : p \in R(t), n = M_{ts}(p), n > 0, 0 \leq ts \leq k\}$

We show that A satisfies (A.5) **and** (A.6)**, and A is an answer set of Π^1.**
A satisfies (A.5) and (A.6) by its construction above. We show A is an answer set of Π^1 by splitting. We split $\mathit{lit}(\Pi^1)$ into a sequence of $7(k+1)+2$ sets:

- $U_0 = \mathit{head}(f1) \cup \mathit{head}(f2) \cup \mathit{head}(f5) \cup \mathit{head}(f6) \cup \mathit{head}(i1) = \{\mathit{place}(p) : p \in P\} \cup \{\mathit{trans}(t) : t \in T\} \cup \{\mathit{time}(0), \ldots, \mathit{time}(k)\} \cup \{\mathit{num}(0), \ldots, \mathit{num}(\mathit{ntok})\} \cup \{\mathit{holds}(p, q, 0) : p \in P, q = M_0(p)\}$

- $U_{7k+1} = U_{7k+0} \cup head(f7)^{ts=k} \cup head(f8)^{ts=k} \cup head(f9)^{ts=k} = U_{7k+0} \cup \{ptarc(p,t,n,k) : (p,t) \in E^-, n = W(p,t)\} \cup \{tparc(t,p,n,k) : (t,p) \in E^+, n = W(t,p)\} \cup \{ptarc(p,t,n,k) : p \in R(t), n = M_k(p), n > 0\}$
- $U_{7k+2} = U_{7k+1} \cup head(e3)^{ts=k} = U_{7k+1} \cup \{notenabled(t,k) : t \in T\}$
- $U_{7k+3} = U_{7k+2} \cup head(e2)^{ts=k} = U_{7k+2} \cup \{enabled(t,k) : t \in T\}$
- $U_{7k+4} = U_{7k+3} \cup head(a1)^{ts=k} = U_{7k+3} \cup \{fires(t,k) : t \in T\}$
- $U_{7k+5} = U_{7k+4} \cup head(r6)^{ts=k} \cup head(r7)^{ts=k} = U_{7k+4} \cup \{add(p,q,t,k) : p \in P, t \in T, q = W(t,p)\} \cup \{del(p,q,t,k) : p \in P, t \in T, q = W(p,t)\} \cup \{del(p,q,t,k) : p \in P, t \in T, q = M_k(p)\}$
- $U_{7k+6} = U_{7k+5} \cup head(r3)^{ts=k} \cup head(r4)^{ts=k} = U_{7k+5} \cup \{tot\_incr(p,q,k) : p \in P, 0 \leq q \leq ntok\} \cup \{tot\_decr(p,q,k) : p \in P, 0 \leq q \leq ntok\}$
- $U_{7k+7} = U_{7k+6} \cup head(r5)^{ts=k} \cup head(a2)^{ts=k} \cup head(a7)^{ts=k} = U_{7k+6} \cup \{consumesmore(p,k) : p \in P\} \cup \{holds(p,q,k+1) : p \in P, 0 \leq q \leq ntok\} \cup \{could\_not\_have(t,k) : t \in T\}$
- $U_{7k+8} = U_{7k+7} \cup head(a3) = U_{7k+7} \cup \{consumesmore\}$

where $head(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\ldots, ts) : a(\ldots, ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 7k+8$ form a splitting sequence, since each U_i is a splitting set of Π^1, and $\langle U_\alpha \rangle_{\alpha < \mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{7k+8}$ and $\bigcup_{\alpha<\mu} U_\alpha = lit(\Pi^1)$.

We compute the answer set of Π^2 using the splitting sets as follows:

1. $bot_{U_0}(\Pi^2) = f1 \cup f2 \cup f5 \cup i1 \cup f6$ and $X_0 = A_1 \cup \cdots \cup A_4 \cup A_7$ $(= U_0)$ is its answer set – using forced atom proposition
2. $eval_{U_0}(bot_{U_1}(\Pi^2) \setminus bot_{U_0}(\Pi^2), X_0) = \{ptarc(p,t,q,0)\text{:-.} | q = W(p,t)\} \cup \{tparc(t,p,q,0)\text{:-.} | \; q = W(t,p)\} \cup \{ptarc(p,t,q,0)\text{:-.} | \; q = M_0(p)\}$. Its answer set $X_1 = A_5^{ts=0} \cup A_6^{ts=0} \cup A_{19}^{ts=0}$ – using forced atom proposition and construction of A_5, A_6, A_{19}.
3. $eval_{U_1}(bot_{U_2}(\Pi^2) \setminus bot_{U_1}(\Pi^2), X_0 \cup X_1) = \{notenabled(t,0)\text{:-.} | \{trans(t), ptarc(p,t,n,0), holds(p,q,0)\} \subseteq X_0 \cup X_1, \text{ where } q < n\}$. Its answer set $X_2 = A_8^{ts=0}$ – using forced atom proposition and construction of A_8.
   (a) where, $q = M_0(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of $i1$ and $f7$ in Π^2, and
   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)
   (c) thus, $notenabled(t,0) \in X_1$ represents $\exists p \in \bullet t : M_0(p) < W(p,t)$.
4. $eval_{U_2}(bot_{U_3}(\Pi^2) \setminus bot_{U_2}(\Pi^2), X_0 \cup \cdots \cup X_2) = \{enabled(t,0)\text{:-.} | trans(t) \in X_0 \cup \cdots \cup X_2, notenabled(t,0) \notin X_0 \cup \cdots \cup X_2\}$. Its answer set is $X_3 = A_9^{ts=0}$ – using forced atom proposition and construction of A_9.

   (a) since an $enabled(t,0) \in X_3$ if $\nexists$ $notenabled(t,0) \in X_0 \cup \cdots \cup X_2$; which is equivalent to $\nexists p \in \bullet t : M_0(p) < W(p,t) \equiv \forall p \in \bullet t : M_0(p) \geq W(p,t)$.

5. $eval_{U_3}(bot_{U_4}(\Pi^2) \setminus bot_{U_3}(\Pi^2), X_0 \cup \cdots \cup X_3) = \{\{fires(t,0)\}$:-.$|enabled(t,0)$ holds in $X_0 \cup \cdots \cup X_3\}$. It has multiple answer sets $X_{4.1}, \ldots, X_{4.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $eval_{U_3}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^2 determined using splitting is equal to A, we only consider the set that matches the $fires(t,0)$ elements in A and call it X_4, ignoring the rest. Thus, $X_4 = A_{10}^{ts=0}$, representing T_0.

   (a) in addition, for every t such that $enabled(t,0) \in X_0 \cup \cdots \cup X_3, R(t) \neq \emptyset$; $fires(t,0) \in X_4$ – per definition 21 (firing set); requiring that a reset transition is fired when enabled

   (b) thus, the firing set T_0 will not be eliminated by the constraint $f10$

6. $eval_{U_4}(bot_{U_5}(\Pi^2) \setminus bot_{U_4}(\Pi^2), X_0 \cup \cdots \cup X_4) = \{add(p,n,t,0)$:-.$|\{fires(t,0), tparc(t,p,n,0)\} \subseteq X_0 \cup \cdots \cup X_4\} \cup \{del(p,n,t,0)$:-.$|\{fires(t,0), ptarc(p,t,n,0)\} \subseteq X_0 \cup \cdots \cup X_4\}$. It's answer set is $X_5 = A_{11}^{ts=0} \cup A_{12}^{ts=0}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

   (a) where, each add atom is equivalent to $n = W(t,p) : p \in t\bullet$,

   (b) and each del atom is equivalent to $n = W(p,t) : p \in \bullet t$; or $n = M_k(p) : p \in R(t)$,

   (c) representing the effect of transitions in T_0 – by construction

7. $eval_{U_5}(bot_{U_6}(\Pi^2) \setminus bot_{U_5}(\Pi^2), X_0 \cup \cdots \cup X_5) = \{tot\_incr(p,qq,0)$:-.$| qq = \sum_{add(p,q,t,0) \in X_0 \cup \cdots \cup X_5} q\} \cup \{tot\_decr(p,qq,0)$:-.$|qq = \sum_{del(p,q,t,0) \in X_0 \cup \cdots \cup X_5} q\}$. It's answer set is $X_6 = A_{13}^{ts=0} \cup A_{14}^{ts=0}$ – using forced atom proposition, definitions of A_{13}, A_{14}, and definition 5 (semantics of aggregate assignment atom).

   (a) where, each for $tot\_incr(p,qq,0)$, $qq = \sum_{add(p,q,t,0) \in X_0 \cup \ldots X_5} q$ $\equiv qq = \sum_{t \in X_4, p \in t\bullet} W(p,t)$,

   (b) and each $tot\_decr(p,qq,0)$, $qq = \sum_{del(p,q,t,0) \in X_0 \cup \ldots X_5} q$ $\equiv qq = \sum_{t \in X_4, p \in \bullet t} W(t,p) + \sum_{t \in X_4, p \in R(t)} M_k(p)$,

   (c) represent the net effect of transitions in T_0 – by construction

8. $eval_{U_6}(bot_{U_7}(\Pi^2) \setminus bot_{U_6}(\Pi^2), X_0 \cup \cdots \cup X_6) = \{consumesmore(p,0)$:-.$| \{holds(p,q,0), tot\_decr(p,q1,0)\} \subseteq X_0 \cup \cdots \cup X_6, q1 > q\} \cup \{holds(p,q,1)$:-.,$| \{holds(p,q1,0), tot\_incr(p,q2,0), tot\_decr(p,q3,0)\} \subseteq X_0 \cup \cdots \cup X_6, q = q1 + q2 - q3\} \cup \{could\_not\_have(t,0)$:-.$|\{enabled(t,0), ptarc(s,t,q), holds(s,qq,0), tot\_decr(s,qqq,0)\} \subseteq X_0 \cup \cdots \cup X_6, fires(t,0) \notin (X_0 \cup \cdots \cup X_6), q > qq - qqq\}$. It's answer set is $X_7 = A_{15}^{ts=0} \cup A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of A_{15}, A_{17}, A_{18}.

(a) where, $consumesmore(p, 0)$ represents $\exists p : q = M_0(p), q1 = \sum_{t \in T_0, p \in \bullet t} W(p, t) + \sum_{t \in T_0, p \in R(t)} M_k(p)$, $q1 > q$, indicating place p will be overconsumed if T_0 is fired – as defined in definition 23 (conflicting transitions)

(b) and, $holds(p, q, 1)$ represents $q = M_1(p)$ – by construction of Π^2

(c) and $could\_not\_have(t, 0)$ represents an enabled transition t in T_0 that could not fire due to insufficient tokens

(d) X_7 does not contain $could\_not\_have(t, 0)$, when $enabled(t, 0) \in X_0 \cup \dots \cup X_6$ and $fires(t, 0) \notin X_0 \cup \dots \cup X_6$ due to construction of A, encoding of $a7$ and its body atoms. As a result it is not eliminated by the constraint $a6$

$$\vdots$$

9. $eval_{U_{7k+0}}(bot_{U_{7k+1}}(\Pi^2) \setminus bot_{U_{7k+0}}(\Pi^2), X_0 \cup \dots \cup X_{7k+0}) = \{ptarc(p, t, q, k){:}\text{-}.|q = W(p, t)\} \cup \{tparc(t, p, q, k){:}\text{-}.|q = W(t, p)\} \cup \{ptarc(p, t, q, k){:}\text{-}.|q = M_0(p)\}$. Its answer set $X_{7k+1} = A_5^{ts=k} \cup A_6^{ts=k} \cup A_{19}^{ts=k}$ – using forced atom proposition and construction of A_5, A_6, A_{19}.

10. $eval_{U_{7k+1}}(bot_{U_{7k+2}}(\Pi^2) \setminus bot_{U_{7k+1}}(\Pi^2), X_0 \cup X_{7k+1}) = \{notenabled(t, k){:}\text{-}.|\{trans(t), ptarc(p, t, n, k), holds(p, q, k)\} \subseteq X_0 \cup X_{7k+1}$, where $q < n\}$. Its answer set $X_{7k+2} = A_8^{ts=k}$ – using forced atom proposition and construction of A_8.

    (a) where, $q = M_0(p)$, and $n = W(p, t)$ for an arc $(p, t) \in E^-$ – by construction of $holds$ and $ptarc$ predicates in Π^2, and

    (b) in an arc $(p, t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

    (c) thus, $notenabled(t, k) \in X_{7k+1}$ represents $\exists p \in \bullet t : M_0(p) < W(p, t)$.

11. $eval_{U_{7k+2}}(bot_{U_{7k+3}}(\Pi^2) \setminus bot_{U_{7k+2}}(\Pi^2), X_0 \cup \dots \cup X_{7k+2}) = \{enabled(t, k){:}\text{-}.|trans(t) \in X_0 \cup \dots \cup X_{7k+2}, notenabled(t, k) \notin X_0 \cup \dots \cup X_{7k+2}\}$. Its answer set is $X_{7k+3} = A_9^{ts=k}$ – using forced atom proposition and construction of A_9.

    (a) since an $enabled(t, k) \in X_{7k+3}$ if $\nexists$ $notenabled(t, k) \in X_0 \cup \dots \cup X_{7k+2}$; which is equivalent to $\nexists p \in \bullet t : M_0(p) < W(p, t) \equiv \forall p \in \bullet t : M_0(p) \geq W(p, t)$.

12. $eval_{U_{7k+3}}(bot_{U_{7k+4}}(\Pi^2) \setminus bot_{U_{7k+3}}(\Pi^2), X_0 \cup \dots \cup X_{7k+3}) = \{\{fires(t, k)\}{:}\text{-}.|$ $enabled(t, k)$
    holds in $X_0 \cup \dots \cup X_{7k+3}\}$. It has multiple answer sets $X_{7k+4.1}, \dots, X_{1k+4.n}$, corresponding to elements of power set of $fires(t, k)$ atoms in $eval_{U_{7k+3}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^2 determined using splitting is equal to A, we only consider the set that matches the $fires(t, k)$ elements in A and call it X_{7k+4}, ignoring the rest. Thus, $X_{7k+4} = A_{10}^{ts=k}$, representing T_k.

    (a) in addition, for every t such that $enabled(t, k) \in X_0 \cup \dots \cup X_{7k+3}, R(t) \neq \emptyset$; $fires(t, k) \in X_{7k+4}$ – per definition 21 (firing set); requiring that a reset transition is fired when enabled

(b) thus, the firing set T_k will not be eliminated by the constraint $f10$

13. $eval_{U_{7k+4}}(bot_{U_{7k+5}}(\Pi^2) \setminus bot_{U_{7k+4}}(\Pi^2), X_0 \cup \dots \cup X_{7k+4}) =$ $\{add(p,n,t,k)\text{:-.} | \{fires(t,k), tparc(t,p,n,k)\} \subseteq X_0 \cup \dots \cup X_{7k+4}\} \cup$ $\{del(p,n,t,k)\text{:-.} | \{fires(t,k), ptarc(p,t,n,k)\} \subseteq X_0 \cup \dots \cup X_{7k+4}\}$. It's answer set is $X_{7k+5} = A_{11}^{ts=k} \cup A_{12}^{ts=k}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

    (a) where, each add atom is equivalent to $n = W(t,p) : p \in \bullet t$,

    (b) and each del atom is equivalent to $n = W(p,t) : p \in t\bullet$; or $n = M_k(p) : p \in R(t)$,

    (c) representing the effect of transitions in T_k

14. $eval_{U_{7k+5}}(bot_{U_{7k+6}}(\Pi^2) \setminus bot_{U_{7k+5}}(\Pi^2), X_0 \cup \dots \cup X_{7k+5}) =$ $\{tot\_incr(p,qq,k)\text{:-.} | qq = \sum_{add(p,q,t,k) \in X_0 \cup \dots \cup X_{7k+5}} q\} \cup$ $\{tot\_decr(p,qq,k)\text{:-.} | qq = \sum_{del(p,q,t,k) \in X_0 \cup \dots \cup X_{7k+5}} q\}$. It's answer set is $X_{7k+6} = A_{13}^{ts=k} \cup A_{14}^{ts=k}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

    (a) where, each $tot\_incr(p,qq,k)$, $qq = \sum_{add(p,q,t,k) \in X_0 \cup \dots X_{7k+5}} q$ $\equiv qq = \sum_{t \in X_{7k+4}, p \in t\bullet} W(p,t)$,

    (b) and each $tot\_decr(p,qq,k)$, $qq = \sum_{del(p,q,t,k) \in X_0 \cup \dots X_{7k+5}} q$ $\equiv qq = \sum_{t \in X_{7k+4}, p \in \bullet t} W(t,p) + \sum_{t \in X_{7k+4}, p \in R(t)} M_k(p)$,

    (c) represent the net effect of transitions in T_k

15. $eval_{U_{7k+6}}(bot_{U_{7k+7}}(\Pi^2) \setminus bot_{U_{7k+6}}(\Pi^2), X_0 \cup \dots \cup X_{7k+6}) = \{consumesmore(p,k)\text{:-.} |$ $\{holds(p,q,k), tot\_decr(p,q1,k)\} \subseteq X_0 \cup \dots \cup X_{7k+6}, q1 > q\} \cup \{holds(p,q,k+1)\text{:-.,} |$ $\{holds(p,q1,k), tot\_incr(p,q2,k), tot\_decr(p,q3,k)\} \subseteq X_0 \cup \dots \cup X_{7k+6}, q = q1 + q2 - q3\} \cup \{could\_not\_have(t,k)\text{:-.} | \{enabled(t,k), ptarc(s,t,q), holds(s,qq,k),$ $tot\_decr(s,qqq,k)\} \subseteq X_0 \cup \dots \cup X_{7k+6}, fires(t,k) \notin (X_0 \cup \dots \cup X_{7k+6}), q > qq - qqq\}$. It's answer set is $X_{7k+7} = A_{15}^{ts=k} \cup A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition and definitions of A_{15}, A_{17}, A_{18}.

    (a) where, $consumesmore(p,k)$ represents $\exists p : q = M_0(p), q1 = \sum_{t \in T_0, p \in \bullet t} W(p,t) + \sum_{t \in T_0, p \in R(t)} M_k(p), q1 > q$, indicating place p that will be over consumed if T_k is fired, as defined in definition 23 (conflicting transitions)

    (b) $holds(p,q,k+1)$ represents $q = M_{k+1}(p)$ – by construction of Π^2,

    (c) and $could\_not\_have(t,k)$ represents enabled transition t in T_k that could not be fired due to insufficient tokens

    (d) X_{7k+7} does not contain $could\_not\_have(t,k)$, when $enabled(t,k) \in X_0 \cup \dots \cup X_{7k+6}$ and $fires(t,k) \notin X_0 \cup \dots \cup X_{7k+6}$ due to the construction of A, encoding of $a7$ and its body atoms. As a result it is not eliminated by the constraint $a6$

16. $eval_{U_{7k+7}}(bot_{U_{7k+8}}(\Pi^2) \setminus bot_{U_{7k+7}}(\Pi^2), X_0 \cup \dots \cup X_{7k+7}) = \{consumesmore\text{:-.} | \{consumesmore(p,0), \dots, consumesmore(p,k)\} \subseteq X_0 \cup \dots \cup X_{7k+7}\}$. It's answer set is $X_{7k+8} = A_{16}$ – using forced atom proposition

    (a) X_{7k+8} will be empty since none of $consumesmore(p,0), \dots,$ $consumesmore(p,k)$ hold in $X_0 \cup \dots \cup X_{7k+7}$ due to the construction of A, encoding of $a2$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \dots \cup X_{7k+8}$ is the answer set of Π^2 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 7k+8$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^2.

Next we show (II): Given Π^2 be the encoding of a Petri Net $PN(P,T,E,W,R)$ with initial marking M_0, and A be an answer set of Π^2 that satisfies (A.5) and (A.6), then we can construct $X = M_0, T_0, \dots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \dots, M_i(p_n))$, where $\{holds(p_0, M_i(p_0), i), \dots holds(p_n, M_i(p_n), i)\} \subseteq A$, for $0 \leq i \leq k+1$
2. $T_i = \{t : fires(t,i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) T_0's firing in M_0 produces marking M_1.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0,0), \dots, fires(t_x,0)\}$ be the set of all $fires(\dots,0)$ atoms in A,

    (a) Then for each $fires(t_i,0) \in A$

        i. $enabled(t_i,0) \in A$ – from rule $a1$ and supported rule proposition

        ii. Then $notenabled(t_i,0) \notin A$ – from rule $e2$ and supported rule proposition

        iii. Then $body(e3)$ must not hold in A – from rule $e3$ and forced atom proposition

        iv. Then $q \not< n_i \equiv q \geq n_i$ in $e3$ for all $\{holds(p,q,0), ptarc(p,t_i,n_i,0)\} \subseteq A$ – from $e3$, forced atom proposition, and the following

            A. $holds(p,q,0) \in A$ represents $q = M_0(p)$ – rule $i1$ construction

            B. $ptarc(p,t_i,n_i,0) \in A$ represents $n_i = W(p,t_i)$ – rule $f7$ construction; or it represents $n_i = M_0(p)$ – rule $f9$ construction; the construction of $f9$ ensures that $notenabled(t,0)$ is never true due to the reset arc

v. Then $\forall p \in \bullet t_i, M_0(p) \geq W(p, t_i)$ – from definition 11 of preset $\bullet t_i$ in PN

vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 12 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^2 – from rule $a4$ and supported rule proposition

i. Then $\nexists consumesmore(p, 0) \in A$ – from rule $a3$ and supported rule proposition

ii. Then $\nexists \{holds(p, q, 0), tot\_decr(p, q1, 0)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition

iii. Then $\nexists p : (\sum_{t_i \in \{t_0,\dots,t_x\}, p \in \bullet t_i} W(p, t_i) + \sum_{t_i \in \{t_0,\dots,t_x\}, p \in R(t_i)} M_0(p)) > M_0(p)$ – from the following

A. $holds(p, q, 0)$ represents $q = M_0(p)$ – from rule $i1$ construction, given

B. $tot\_decr(p, q1, 0) \in A$ if $\{del(p, q1_0, t_0, 0), \dots, del(p, q1_x, t_x, 0)\} \subseteq A$, where $q1 = q1_0 + \dots + q1_x$ – from $r4$ and forced atom proposition

C. $del(p, q1_i, t_i, 0) \in A$ if $\{fires(t_i, 0), ptarc(p, t_i, q1_i, 0)\} \subseteq A$ – from $r7$ and supported rule proposition

D. $del(p, q1_i, t_i, 0)$ either represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_i = M_0(p)$ tokens from $p \in R(t_i)$– from rule $r7$, supported rule proposition, and definition 22 of transition execution in PN

iv. Then the set of transitions in T_0 do not conflict – by the definition 23 of conflicting transitions

(c) And for each $enabled(t_j, 0) \in A$ and $fires(t_j, 0) \notin A$, $could\_not\_have(t_j, 0) \in A$, since A is an answer set of Π^2 - from rule $a6$ and supported rule proposition

i. Then $\{enabled(t_j, 0), holds(s, qq, 0), ptarc(s, t_j, q, 0), tot\_decr(s, qqq, 0)\} \subseteq A$, such that $q > qq - qqq$ and $fires(t_j, 0) \notin A$ - from rule $a7$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q > M_0(s) - (\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_0, s \in R(t_i)} M_0(s))$, where $q = W(s, t_j)$ if $s \in \bullet t_j$, or $M_0(s)$ otherwise - from the following:

A. $ptarc(s, t_i, q, 0)$ represents $q = W(s, t_i)$ if $(s, t_i) \in E^-$ or $q = M_0(s)$ if $s \in R(t_i)$ – from rule $f7, f9$ construction

B. $holds(s, qq, 0)$ represents $qq = M_0(s)$ – from $i1$ construction

C. $tot\_decr(s, qqq, 0) \in A$ if $\{del(s, qqq_0, t_0, 0), \dots, del(s, qqq_x, t_x, 0)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

D. $del(s, qqq_i, t_i, 0) \in A$ if $\{fires(t_i, 0), ptarc(s, t_i, qqq_i, 0)\} \subseteq A$ – from rule $r7$ and supported rule proposition

E. $del(s, qqq_i, t_i, 0)$ represents $qqq_i = W(s, t_i) : t_i \in T_0, (s, t_i) \in E^-$, or $qqq_i = M_0(t_i) : t_i \in T_0, s \in R(t_i)$ – from rule $f7, f9$ construction

F. $tot\_decr(q, qqq, 0)$ represents $\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_0, s \in R(t_i)} M_0(s)$ – from (C,D,E) above

iii. Then firing $T_0 \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_0 is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $enabled(t_r, 0) \in A$, $fires(t_r, 0) \in A$, since A is an answer set of Π^2 - from rule $f10$ and supported rule proposition

i. Then, the firing set T_0 satisfies the reset-transition requirement of definition 21 (firing set)

(e) Then $\{t_0, \dots, t_x\} = T_0$ – using 1(a),1(b),1(d) above; and using 1(c) it is a maximal firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $holds(p, q, 1) \in A$

(a) Then $\{holds(p, q1, 0), tot\_incr(p, q2, 0), tot\_decr(p, q3, 0)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) $holds(p, q1, 0) \in A$ represents $q1 = M_0(p)$ – given, rule $i1$ construction; and $\{add(p, q2_0, t_0, 0), \dots, add(p, q2_j, t_j, 0)\} \subseteq A : q2_0 + \cdots + q2_j = q2$ and $\{del(p, q3_0, t_0, 0), \dots, del(p, q3_l, t_l, 0)\} \subseteq A : q3_0 + \cdots + q3_l = q3$ – rules $r3, r4$ using supported rule proposition

(c) Then $\{fires(t_0, 0), \dots, fires(t_j, 0)\} \subseteq A$ and $\{fires(t_0, 0), \dots, fires(t_l, 0)\} \subseteq A$ – rules $r6, r7$ and supported rule proposition, respectively

(d) Then $\{fires(t_0, 0), \dots, fires(t_j, 0)\} \cup \{fires(t_0, 0), \dots, fires(t_l, 0)\} \subseteq A = \{fires(t_0, 0), \dots, fires(t_x, 0)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

(f) Then $q = M_0(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - (\sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} M_0(p))$ – from (2a),(2b) (2b) above and the following

i. Each $add(p, q_j, t_j, 0) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r6$ encoding, and definition 11 of postset in PN

ii. Each $del(p, t_y, q_y, 0) \in A$ represents either $q_y = W(p, t_y)$ for $p \in \bullet t_y$, or $q_y = M_0(p)$ for $p \in R(t_y)$ – from rule $r7, f7$ encoding and definition 11 of preset in PN; or from rule $r7, f9$ encoding and definition of reset arc in PN

iii. Each $tot\_incr(p, q2, 0) \in A$ represents $q2 = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} M_0(p)$ – aggregate assignment atom semantics in rule $r4$

(g) Then $M_1(p) = q$ – since $holds(p, q, 1) \in A$ encodes $q = M_1(p)$ – from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set in M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set in M_k. Let $\{fires(t_0, k), \ldots, fires(t_x, k)\}$ be the set of all $fires(\ldots, k)$ atoms in A,
   (a) Then for each $fires(t_i, k) \in A$
      i. $enabled(t_i, k) \in A$ – from rule $a1$ and supported rule proposition
      ii. Then $notenabled(t_i, k) \notin A$ – from rule $e2$ and supported rule proposition
      iii. Then $body(e3)$ must not hold in A – from rule $e3$ and forced atom proposition
      iv. Then $q \not< n_i \equiv q \geq n_i$ in $e3$ for all $\{holds(p, q, k), ptarc(p, t_i, n_i, k)\} \subseteq A$ – from $e3$, forced atom proposition, and the following
         A. $holds(p, q, k) \in A$ represents $q = M_k(p)$ – construction, inductive assumption
         B. $ptarc(p, t_i, n_i, k) \in A$ represents $n_i = W(p, t_i)$ – rule $f7$ construction; or it represents $n_i = M_k(p)$ – rule $f9$ construction; the construction of $f9$ ensures that $notenabled(t, 0)$ is never true due to the reset arc
      v. Then $\forall p \in \bullet t_i, M_k(p) \geq W(p, t_i)$ – from definition 11 of preset $\bullet t_i$ in PN
      vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 12 of enabled transition
   (b) And $consumesmore \notin A$, since A is an answer set of Π^2 – from rule $a4$ and supported rule proposition
      i. Then $\nexists consumesmore(p, k) \in A$ – from rule $a3$ and supported rule proposition
      ii. Then $\nexists \{holds(p, q, k), tot\_decr(p, q1, k)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition
      iii. Then $\nexists p : (\sum_{t_i \in \{t_0, \ldots, t_x\}, p \in \bullet t_i} W(p, t_i) + \sum_{t_i \in \{t_0, \ldots, t_x\}, p \in R(t_i)} M_k(p)) > M_k(p)$ – from the following
         A. $holds(p, q, k)$ represents $q = M_k(p)$ – inductive assumption, given
         B. $tot\_decr(p, q1, k) \in A$ if $\{del(p, q1_0, t_0, k), \ldots, del(p, q1_x, t_x, k)\} \subseteq A$, where $q1 = q1_0 + \cdots + q1_x$ – from $r4$ and forced atom proposition
         C. $del(p, q1_i, t_i, k) \in A$ if $\{fires(t_i, k), ptarc(p, t_i, q1_i, k)\} \subseteq A$ – from $r7$ and supported rule proposition
         D. $del(p, q1_i, t_i, k)$ either represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_i = M_k(p)$ tokens from $p \in R(t_i)$– from rule $r7$, supported rule proposition, and definition 22 of transition execution in PN
      iv. Then the set of transitions in T_k do not conflict – by the definition 23 of conflicting transitions

(c) And for each $\mathit{enabled}(t_j,k) \in A$ and $\mathit{fires}(t_j,k) \notin A$, $\mathit{could\_not\_have}(t_j,k) \in A$, since A is an answer set of Π^2 - from rule $a6$ and supported rule proposition

  i. Then $\{\mathit{enabled}(t_j,k), \mathit{holds}(s,qq,k), \mathit{ptarc}(s,t_j,q,k), \mathit{tot\_decr}(s,qqq,k)\} \subseteq A$, such that $q > qq - qqq$ and $\mathit{fires}(t_j,k) \notin A$ - from rule $a7$ and supported rule proposition

  ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q > M_k(s) - (\sum_{t_i \in T_k, s \in \bullet t_i} W(s,t_i) + \sum_{t_i \in T_k, s \in R(t_i)} M_k(s))$, where $q = W(s,t_j)$ if $s \in \bullet t_j$, or $M_k(s)$ otherwise - from the following:

    A. $\mathit{ptarc}(s,t_i,q,k)$ represents $q = W(s,t_i)$ if $(s,t_i) \in E^-$ or $q = M_k(s)$ if $s \in R(t_i)$ – from rule $f7, f9$ construction

    B. $\mathit{holds}(s,qq,k)$ represents $qq = M_k(s)$ – construction

    C. $\mathit{tot\_decr}(s,qqq,k) \in A$ if $\{\mathit{del}(s,qqq_0,t_0,k), \dots, \mathit{del}(s,qqq_x,t_x,k)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

    D. $\mathit{del}(s,qqq_i,t_i,k) \in A$ if $\{\mathit{fires}(t_i,k), \mathit{ptarc}(s,t_i,qqq_i,k)\} \subseteq A$ – from rule $r7$ and supported rule proposition

    E. $\mathit{del}(s,qqq_i,t_i,k)$ represents $qqq_i = W(s,t_i) : t_i \in T_k, (s,t_i) \in E^-$, or $qqq_i = M_k(t_i) : t_i \in T_k, s \in R(t_i)$ – from rule $f7, f9$ construction

    F. $\mathit{tot\_decr}(q,qqq,k)$ represents $\sum_{t_i \in T_k, s \in \bullet t_i} W(s,t_i) + \sum_{t_i \in T_k, s \in R(t_i)} M_k(s)$ – from (C,D,E) above

  iii. Then firing $T_k \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_k is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $\mathit{enabled}(t_r,k) \in A$, $\mathit{fires}(t_r,k) \in A$, since A is an answer set of Π^2 - from rule $f10$ and supported rule proposition

  i. Then the firing set T_k satisfies the reset transition requirement of definition 21 (firing set)

(e) Then $\{t_0, \dots, t_x\} = T_k$ – using 1(a),1(b), 1(d) above; and using 1(c) it is a maximal firing set

2. We show that M_{k+1} is produced by firing T_k in M_k. Let $\mathit{holds}(p,q,k+1) \in A$

(a) Then $\{\mathit{holds}(p,q1,k), \mathit{tot\_incr}(p,q2,k), \mathit{tot\_decr}(p,q3,k)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) $\mathit{holds}(p,q1,k) \in A$ represents $q1 = M_k(p)$ – construction, inductive assumption; and $\{\mathit{add}(p,q2_0,t_0,k), \dots, \mathit{add}(p,q2_j,t_j,k)\} \subseteq A : q2_0 + \dots + q2_j = q2$ and $\{\mathit{del}(p,q3_0,t_0,k), \dots, \mathit{del}(p,q3_l,t_l,k)\} \subseteq A : q3_0 + \dots + q3_l = q3$ – rules $r3, r4$ using supported rule proposition

(c) Then $\{\mathit{fires}(t_0,k), \dots, \mathit{fires}(t_j,k)\} \subseteq A$ and $\{\mathit{fires}(t_0,k), \dots, \mathit{fires}(t_l,k)\} \subseteq A$ – rules $r6, r7$ and supported rule proposition, respectively

(d) Then $\{\mathit{fires}(t_0,k), \dots, \mathit{fires}(t_j,k)\} \cup \{\mathit{fires}(t_0,k), \dots, \mathit{fires}(t_l,k)\} \subseteq A = \{\mathit{fires}(t_0,k), \dots, \mathit{fires}(t_x,k)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x, k) \in A$ we have $t_x \in T_k$ – already shown in item 1 above

(f) Then $q = M_k(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - (\sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_k \wedge p \in R(t_x)} M_k(p))$ – from (2b) above and the following

   i. Each $add(p, q_j, t_j, k) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r6$ encoding, and definition 22 of transition execution in PN

   ii. Each $del(p, t_y, q_y, k) \in A$ represents either $q_y = W(p, t_y)$ for $p \in \bullet t_y$, or $q_y = M_k(p)$ for $p \in R(t_y)$ – from rule $r7, f7$ encoding and definition 22 of transition execution in PN; or from rule $r7, f9$ encoding and definition of reset arc in PN

   iii. Each $tot\_incr(p, q2, k) \in A$ represents $q2 = \sum_{t_x \in T_k \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

   iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_k \wedge p \in R(t_x)} M_0(p)$ – aggregate assignment atom semantics in rule $r4$

(g) Then, $M_{k+1}(p) = q$ – since $holds(p, q, k+1) \in A$ encodes $q = M_{k+1}(p)$ – from construction

As a result, for any $n > k$, T_n is a valid firing set w.r.t. M_n and its firing produces marking M_{n+1}.
Conclusion: Since both (I) and (II) hold, $X = M_0, T_0, M_1, \ldots, M_k, T_{k+1}$ is an execution sequence of $PN(P, T, E, W, R)$ (w.r.t M_0) iff there is an answer set A of $\Pi^2(PN, M_0, k, ntok)$ such that (A.5) and (A.6) hold.

A.4 Poof of Proposition 4

Let $PN = (P, T, E, W, R, I)$ be a Petri Net, M_0 be its initial marking and let $\Pi^3(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.6. Then $X = M_0, T_0, M_1, \ldots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t. M_0) iff there is an answer set A of $\Pi^3(PN, M_0, k, ntok)$ such that:

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\} \tag{A.7}$$

$$\begin{aligned} &\{holds(p, q, ts) : p \in P, q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ &\qquad = \{holds(p, q, ts) : holds(p, q, ts) \in A\} \end{aligned} \tag{A.8}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.7) and (A.4) and show that A is an answer set of Π^3

(II) Given an answer set A of Π^3, we create an execution sequence X such that (A.3) and (A.8) are satisfied.

First we show (I): Given PN and an execution sequence X of PN, we create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$

2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$

3. $A_3 = \{place(p) : p \in P\}$

4. $A_4 = \{trans(t) : t \in T\}$

5. $A_5 = \{ptarc(p, t, n, ts) : (p, t) \in E^-, n = W(p, t), 0 \leq ts \leq k\}$, where $E^- \subseteq E$

6. $A_6 = \{tparc(t, p, n, ts) : (t, p) \in E^+, n = W(t, p), 0 \leq ts \leq k\}$, where $E^+ \subseteq E$

7. $A_7 = \{holds(p, q, 0) : p \in P, q = M_0(p)\}$

8. $A_8 = \{notenabled(t, ts) : t \in T, 0 \leq ts \leq k, (\exists p \in \bullet t, M_{ts}(p) < W(p, t)) \vee (\exists p \in I(t), M_{ts}(p) \neq 0)\}$
   per definition 26 (enabled transition)

9. $A_9 = \{enabled(t, ts) : t \in T, 0 \leq ts \leq k, (\forall p \in \bullet t, W(p, t) \leq M_{ts}(p)) \wedge (\forall p \in I(t), M_{ts}(p) = 0)\}$
   per definition 26 (enabled transition)

10. $A_{10} = \{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
    per definition 21 (firing set), only an enabled transition may fire

11. $A_{11} = \{add(p, q, t, ts) : t \in T_{ts}, p \in t\bullet, q = W(t, p), 0 \leq ts \leq k\}$
    per definition 22 (transition execution)

12. $A_{12} = \{del(p, q, t, ts) : t \in T_{ts}, p \in \bullet t, q = W(p, t), 0 \leq ts \leq k\} \cup \{del(p, q, t, ts) : t \in T_{ts}, p \in R(t), q = M_{ts}(p), 0 \leq ts \leq k\}$
    per definition 24 (transition execution)

13. $A_{13} = \{tot\_incr(p, q, ts) : p \in P, q = \sum_{t \in T_{ts}, p \in t\bullet} W(t, p), 0 \leq ts \leq k\}$
    per definition 24 (firing set execution)

14. $A_{14} = \{tot\_decr(p, q, ts) : p \in P, q = \sum_{t \in T_{ts}, p \in \bullet t} W(p, t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), 0 \leq ts \leq k\}$
    per definition 24 (firing set execution)

15. $A_{15} = \{consumesmore(p, ts) : p \in P, q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p, t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), q1 > q, 0 \leq ts \leq k\}$
    per definition 23 (conflicting transitions)

16. $A_{16} = \{consumesmore : \exists p \in P : q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p, t) + \sum_{t \in T_{ts}, p \in R(t)} (M_{ts}(p)), q1 > q, 0 \leq ts \leq k\}$
    per definition 23 (conflicting transitions)

17. $A_{17} = \{could\_not\_have(t, ts) : t \in T, (\forall p \in \bullet t, W(p, t) \leq M_{ts}(p)), t \notin T_{ts}, (\exists p \in \bullet t : W(p, t) > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p, t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), 0 \leq ts \leq k\}$
    per the maximal firing set semantics

18. $A_{18} = \{holds(p, q, ts+1) : p \in P, q = M_{ts+1}(p), 0 \leq ts < k\}$,
where $M_{ts+1}(p) = M_{ts}(p) - (\sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p)) +$
$\sum_{t \in T_{ts}, p \in t\bullet} W(t,p)$
according to definition 21 (firing set execution)

19. $A_{19} = \{ptarc(p, t, n, ts) : p \in R(t), n = M_{ts}(p), n > 0, 0 \leq ts \leq k\}$

20. $A_{20} = \{iptarc(p, t, 1, ts) : p \in P, 0 \leq ts < k\}$

We show that A satisfies (A.7) and (A.8), and A is an answer set of Π^3.

A satisfies (A.7) and (A.8) by its construction above. We show A is an answer set of Π^3 by splitting. We split $lit(\Pi^3)$ into a sequence of $7k+9$ sets:

- $U_0 = head(f1) \cup head(f2) \cup head(f5) \cup head(f6) \cup head(i1) = \{place(p) : p \in P\} \cup \{trans(t) : t \in T\} \cup \{time(0), \dots, time(k)\} \cup \{num(0), \dots, num(ntok)\} \cup \{holds(p, q, 0) : p \in P, q = M_0(p)\}$

- $U_{7k+1} = U_{7k+0} \cup head(f7)^{ts=k} \cup head(f8)^{ts=k} \cup head(f9)^{ts=k} \cup head(f11)^{ts=k} = U_{7k+0} \cup \{ptarc(p, t, n, k) : (p, t) \in E^-, n = W(p, t)\} \cup \{tparc(t, p, n, k) : (t, p) \in E^+, n = W(t, p)\} \cup \{ptarc(p, t, n, k) : p \in R(t), n = M_k(p), n > 0\} \cup \{iptarc(p, t, 1, k) : p \in I(t)\}$

- $U_{7k+2} = U_{7k+1} \cup head(e3)^{ts=k} \cup head(e4)^{ts=k} = U_{7k+1} \cup \{notenabled(t, k) : t \in T\}$

- $U_{7k+3} = U_{7k+2} \cup head(e2)^{ts=k} = U_{7k+2} \cup \{enabled(t, k) : t \in T\}$

- $U_{7k+4} = U_{7k+3} \cup head(a1)^{ts=k} = U_{7k+3} \cup \{fires(t, k) : t \in T\}$

- $U_{7k+5} = U_{7k+4} \cup head(r6)^{ts=k} \cup head(r7)^{ts=k} = U_{7k+4} \cup \{add(p, q, t, k) : p \in P, t \in T, q = W(t, p)\} \cup \{del(p, q, t, k) : p \in P, t \in T, q = W(p, t)\} \cup \{del(p, q, t, k) : p \in P, t \in T, q = M_k(p)\}$

- $U_{7k+6} = U_{7k+5} \cup head(r3)^{ts=k} \cup head(r4)^{ts=k} = U_{7k+5} \cup \{tot\_incr(p, q, k) : p \in P, 0 \leq q \leq ntok\} \cup \{tot\_decr(p, q, k) : p \in P, 0 \leq q \leq ntok\}$

- $U_{7k+7} = U_{7k+6} \cup head(r5)^{ts=k} \cup head(a2)^{ts=k} \cup head(a7)^{ts=k} = U_{7k+6} \cup \{consumesmore(p, k) : p \in P\} \cup \{holds(p, q, k+1) : p \in P, 0 \leq q \leq ntok\} \cup \{could\_not\_have(t, k) : t \in T\}$

- $U_{7k+8} = U_{7k+7} \cup head(a3)^{ts=k} = U_{7k+7} \cup \{consumesmore\}$

where $head(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\dots, ts) : a(\dots, ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 7k+8$ form a splitting sequence, since each U_i is a splitting set of Π^1, and $\langle U_\alpha \rangle_{\alpha < \mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{7k+8}$ and $\bigcup_{\alpha < \mu} U_\alpha = lit(\Pi^1)$.

We compute the answer set of Π^3 using the splitting sets as follows:

1. $bot_{U_0}(\Pi^3) = f1 \cup f2 \cup f5 \cup i1 \cup f6$ and $X_0 = A_1 \cup \dots \cup A_4 \cup A_7$ $(= U_0)$ is its answer set – using forced atom proposition

2. $eval_{U_0}(bot_{U_1}(\Pi^3) \setminus bot_{U_0}(\Pi^3), X_0) = \{ptarc(p,t,q,0)\text{:-.}|q = W(p,t)\} \cup \{tparc(t,p,q,0)\text{:-.}|q = W(t,p)\} \cup \{ptarc(p,t,q,0)\text{:-.}|q = M_0(p)\} \cup \{iptarc(p,t,1,0)\text{:-.}\}$. Its answer set $X_1 = A_5^{ts=0} \cup A_6^{ts=0} \cup A_{19}^{ts=0} \cup A_{20}^{ts=0}$ – using forced atom proposition and construction of A_5, A_6, A_{19}, A_{20}.

3. $eval_{U_1}(bot_{U_2}(\Pi^3) \setminus bot_{U_1}(\Pi^3), X_0 \cup X_1) = \{notenabled(t,0)\text{:-.}|(\{trans(t), ptarc(p,t,n,0), holds(p,q,0)\} \subseteq X_0 \cup X_1,$ where $q < n)$ or $\{notenabled(t,0)\text{:-.}| (\{trans(t), iptarc(p,t,n2,0), holds(p,q,0)\} \subseteq X_0 \cup X_1,$ where $q \geq n2\}\}$. Its answer set $X_2 = A_8^{ts=0}$ – using forced atom proposition and construction of A_8.

   (a) where, $q = M_0(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction $i1$ and $f7$ in Π^3, and

   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

   (c) $n2 = 1$ – by construction of $iptarc$ predicates in Π^3, meaning $q \geq n2 \equiv q \geq 1 \equiv q > 0$,

   (d) thus, $notenabled(t,0) \in X_1$ represents $(\exists p \in \bullet t : M_0(p) < W(p,t)) \vee (\exists p \in I(t) : M_0(p) > 0)$.

4. $eval_{U_2}(bot_{U_3}(\Pi^3) \setminus bot_{U_2}(\Pi^3), X_0 \cup \dots \cup X_2) = \{enabled(t,0)\text{:-.}|trans(t) \in X_0 \cup \dots \cup X_2, notenabled(t,0) \notin X_0 \cup \dots \cup X_2\}$. Its answer set is $X_3 = A_9^{ts=0}$ – using forced atom proposition and construction of A_9.

   (a) since an $enabled(t,0) \in X_3$ if $\nexists$ $notenabled(t,0) \in X_0 \cup \dots \cup X_2$; which is equivalent to $(\nexists p \in \bullet t : M_0(p) < W(p,t)) \wedge (\nexists p \in I(t) : M_0(p) > 0) \equiv (\forall p \in \bullet t : M_0(p) \geq W(p,t)) \wedge (\forall p \in I(t) : M_0(p) = 0)$.

5. $eval_{U_3}(bot_{U_4}(\Pi^3) \setminus bot_{U_3}(\Pi^3), X_0 \cup \dots \cup X_3) = \{\{fires(t,0)\}\text{:-.}|enabled(t,0)$ holds in $X_0 \cup \dots \cup X_3\}$. It has multiple answer sets $X_{4.1}, \dots, X_{4.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $eval_{U_3}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^3 determined using splitting is equal to A, we only consider the set that matches the $fires(t,0)$ elements in A and call it X_4, ignoring the rest. Thus, $X_4 = A_{10}^{ts=0}$, representing T_0.

   (a) in addition, for every t such that $enabled(t,0) \in X_0 \cup \dots \cup X_3, R(t) \neq \emptyset$; $fires(t,0) \in X_4$ – per definition 21 (firing set); requiring that a reset transition is fired when enabled

   (b) thus, the firing set T_0 will not be eliminated by the constraint $f10$

6. $eval_{U_4}(bot_{U_5}(\Pi^3) \setminus bot_{U_4}(\Pi^3), X_0 \cup \dots \cup X_4) = \{add(p,n,t,0)\text{:-.}|\{fires(t,0), tparc(t,p,n,0)\} \subseteq X_0 \cup \dots \cup X_4\} \cup \{del(p,n,t,0)\text{:-.}|\{fires(t,0), ptarc(p,t,n,0)\} \subseteq X_0 \cup \dots \cup X_4\}$. It's answer set is $X_5 = A_{11}^{ts=0} \cup A_{12}^{ts=0}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

   (a) where, each add atom is equivalent to $n = W(t,p) : p \in t\bullet$,

   (b) and each del atom is equivalent to $n = W(p,t) : p \in \bullet t$; or $n = M_k(p) : p \in R(t)$,

   (c) representing the effect of transitions in T_0.

7. $eval_{U_5}(bot_{U_6}(\Pi^3) \setminus bot_{U_5}(\Pi^3), X_0 \cup \cdots \cup X_5) = \{tot\_incr(p, qq, 0){:}\text{-}.| qq = \sum_{add(p,q,t,0) \in X_0 \cup \cdots \cup X_5} q\} \cup \{tot\_decr(p, qq, 0){:}\text{-}.|qq = \sum_{del(p,q,t,0) \in X_0 \cup \cdots \cup X_5} q\}$. It's answer set is $X_6 = A_{13}^{ts=0} \cup A_{14}^{ts=0}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

   (a) where, each $tot\_incr(p, qq, 0)$, $qq = \sum_{add(p,q,t,0) \in X_0 \cup \ldots X_5} q$ $\equiv qq = \sum_{t \in X_4, p \in t\bullet} W(p, t)$,

   (b) and each $tot\_decr(p, qq, 0)$, $qq = \sum_{del(p,q,t,0) \in X_0 \cup \ldots X_5} q$ $\equiv qq = \sum_{t \in X_4, p \in \bullet t} W(t, p) + \sum_{t \in X_4, p \in R(t)} M_k(p)$,

   (c) represent the net effect of transitions in T_0.

8. $eval_{U_6}(bot_{U_7}(\Pi^3) \setminus bot_{U_6}(\Pi^3), X_0 \cup \cdots \cup X_6) = \{consumesmore(p, 0){:}\text{-}.| \{holds(p, q, 0), tot\_decr(p, q1, 0)\} \subseteq X_0 \cup \cdots \cup X_6, q1 > q\} \cup \{holds(p, q, 1){:}\text{-}.,| \{holds(p, q1, 0), tot\_incr(p, q2, 0), tot\_decr(p, q3, 0)\} \subseteq X_0 \cup \cdots \cup X_6, q = q1 + q2 - q3\} \cup \{could\_not\_have(t, 0){:}\text{-}.|\{enabled(t, 0), ptarc(s, t, q), holds(s, qq, 0), tot\_decr(s, qqq, 0)\} \subseteq X_0 \cup \cdots \cup X_6, fires(t, 0) \notin (X_0 \cup \cdots \cup X_6), q > qq - qqq\}$. It's answer set is $X_7 = A_{15}^{ts=0} \cup A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of $A_{15}, A_{17}, A_{18}, A_9$.

   (a) where, $consumesmore(p, 0)$ represents $\exists p : q = M_0(p), q1 = \sum_{t \in T_0, p \in \bullet t} W(p, t) + \sum_{t \in T_0, p \in R(t)} M_k(p), q1 > q$, indicating place p will be over consumed if T_0 is fired, as defined in definition 23 (conflicting transitions)

   (b) $holds(p, q, 1)$ represents $q = M_1(p)$ – by construction of Π^3

   (c) and $could\_not\_have(t, 0)$ represents enabled transition $t \in T_0$ that could not fire due to insufficient tokens

   (d) X_7 does not contain $could\_not\_have(t, 0)$, when $enabled(t, 0) \in X_0 \cup \cdots \cup X_5$ and $fires(t, 0) \notin X_0 \cup \cdots \cup X_6$ due to construction of A, encoding of $a7$ and its body atoms. As a result it is not eliminated by the constraint $a6$

$$\vdots$$

9. $eval_{U_{7k+0}}(bot_{U_{7k+1}}(\Pi^3) \setminus bot_{U_{7k+0}}(\Pi^3), X_0 \cup \cdots \cup X_{7k+0}) = \{ptarc(p, t, q, k){:}\text{-}.|q = W(p, t)\} \cup \{tparc(t, p, q, k){:}\text{-}.|q = W(t, p)\} \cup \{ptarc(p, t, q, k){:}\text{-}.|q = M_k(p)\} \cup \{iptarc(p, t, 1, k){:}\text{-}.\}$. Its answer set $X_{7k+1} = A_5^{ts=k} \cup A_6^{ts=k} \cup A_{19}^{ts=k} \cup A_{20}^{ts=k}$ – using forced atom proposition and construction of A_5, A_6, A_{19}, A_{20}.

10. $eval_{U_{7k+1}}(bot_{U_{7k+2}}(\Pi^3) \setminus bot_{U_{7k+1}}(\Pi^3), X_0 \cup \cdots \cup X_{7k+1}) = \{notenabled(t, k){:}\text{-}.| (\{trans(t), ptarc(p, t, n, k), holds(p, q, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q < n)\}$ or $\{notenabled(t, k){:}\text{-}.|(\{trans(t), iptarc(p, t, n2, k), holds(p, q, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q \geq n2\}\}$. Its answer set $X_{7k+2} = A_8^{ts=k}$ – using forced atom proposition and construction of A_8.

(a) where, $q = M_k(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of *holds* and *ptarc* predicates in Π^3, and

(b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

(c) $n2 = 1$ – by construction of *iptarc* predicates in Π^3, meaning $q \geq n2 \equiv q \geq 1 \equiv q > 0$,

(d) thus, $notenabled(t,k) \in X_{7k+1}$ represents $(\exists p \in \bullet t : M_k(p) < W(p,t)) \vee (\exists p \in I(t) : M_k(p) > k)$.

11. $eval_{U_{7k+2}}(bot_{U_{7k+3}}(\Pi^3) \setminus bot_{U_{7k+2}}(\Pi^3), X_0 \cup \cdots \cup X_{7k+2}) = \{enabled(t,k)\text{:-.} | trans(t) \in X_0 \cup \cdots \cup X_{7k+2} \wedge notenabled(t,k) \notin X_0 \cup \cdots \cup X_{7k+2}\}$. Its answer set is $X_{7k+3} = A_9^{ts=k}$ – using forced atom proposition and construction of A_9.

    (a) since an $enabled(t,k) \in X_{7k+3}$ if $\nexists$ $notenabled(t,k) \in X_0 \cup \cdots \cup X_{7k+2}$; which is equivalent to $(\nexists p \in \bullet t : M_k(p) < W(p,t)) \wedge (\nexists p \in I(t) : M_k(p) > k) \equiv (\forall p \in \bullet t : M_k(p) \geq W(p,t)) \wedge (\forall p \in I(t) : M_k(p) = k)$.

12. $eval_{U_{7k+3}}(bot_{U_{7k+4}}(\Pi^3) \setminus bot_{U_{7k+3}}(\Pi^3), X_0 \cup \cdots \cup X_{7k+3}) = \{\{fires(t,k)\}\text{:-.} | enabled(t,k)$
 holds in $X_0 \cup \cdots \cup X_{7k+3}\}$. It has multiple answer sets $X_{7k+4.1}, \ldots, X_{1k+4.n}$, corresponding to elements of power set of $fires(t,k)$ atoms in $eval_{U_{7k+3}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^3 determined using splitting is equal to A, we only consider the set that matches the $fires(t,k)$ elements in A and call it X_{7k+4}, ignoring the rest. Thus, $X_{7k+4} = A_{10}^{ts=k}$, representing T_k.

    (a) in addition, for every t such that $enabled(t,k) \in X_0 \cup \cdots \cup X_{7k+3}, R(t) \neq \emptyset$; $fires(t,k) \in X_{7k+4}$ – per definition 21 (firing set); requiring that a reset transition is fired when enabled

    (b) thus, the firing set T_k will not be eliminated by the constraint $f10$

13. $eval_{U_{7k+4}}(bot_{U_{7k+5}}(\Pi^3) \setminus bot_{U_{7k+4}}(\Pi^3), X_0 \cup \cdots \cup X_{7k+4}) = \{add(p,n,t,k)\text{:-.} | \{fires(t,k), tparc(t,p,n,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+4}\} \cup \{del(p,n,t,k)\text{:-.} | \{fires(t,k), ptarc(p,t,n,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+4}\}$. It's answer set is $X_{7k+5} = A_{11}^{ts=k} \cup A_{12}^{ts=k}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

    (a) where, each *add* atom is equivalent to $n = W(t,p) : p \in t\bullet$,

    (b) and each *del* atom is equivalent to $n = W(p,t) : p \in \bullet t$; or $n = M_k(p) : p \in R(t)$,

    (c) representing the effect of transitions in T_k

14. $eval_{U_{7k+5}}(bot_{U_{7k+6}}(\Pi^3) \setminus bot_{U_{7k+5}}(\Pi^3), X_0 \cup \cdots \cup X_{7k+5}) = \{tot\_incr(p,qq,k)\text{:-.} | qq = \sum_{add(p,q,t,k) \in X_0 \cup \cdots \cup X_{7k+5}} q\} \cup \{tot\_decr(p,qq,k)\text{:-.} | qq = \sum_{del(p,q,t,k) \in X_0 \cup \cdots \cup X_{7k+5}} q\}$. It's answer set is $X_{7k+6} = A_{13}^{ts=k} \cup A_{14}^{ts=k}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

    (a) where, each $tot\_incr(p,qq,k)$, $qq = \sum_{add(p,q,t,k) \in X_0 \cup \ldots X_{7k+5}} q$
    $\equiv qq = \sum_{t \in X_{7k+4}, p \in t\bullet} W(p,t)$,

(b) and each $tot\_decr(p, qq, k)$, $qq = \sum_{del(p,q,t,k) \in X_0 \cup \dots X_{7k+5}} q$
$\equiv qq = \sum_{t \in X_{7k+4}, p \in \bullet t} W(t, p) + \sum_{t \in X_{7k+4}, p \in R(t)} M_k(p)$,

(c) representing the net effect of transitions in T_k

15. $eval_{U_{7k+6}}(bot_{U_{7k+7}}(\Pi^3) \setminus bot_{U_{7k+6}}(\Pi^3), X_0 \cup \dots \cup X_{7k+6}) = \{consumesmore(p, k)\text{:-.} | \{holds(p, q, k), tot\_decr(p, q1, k)\} \subseteq X_0 \cup \dots \cup X_{7k+6}, q1 > q\} \cup \{holds(p, q, 1)\text{:-.}, | \{holds(p, q1, k), tot\_incr(p, q2, k), tot\_decr(p, q3, k)\} \subseteq X_0 \cup \dots \cup X_{7k+6}, q = q1 + q2 - q3\} \cup \{could\_not\_have(t, k)\text{:-.} | \{enabled(t, k), ptarc(s, t, q), holds(s, qq, k), tot\_decr(s, qqq, k)\} \subseteq X_0 \cup \dots \cup X_{7k+6}, fires(t, k) \notin (X_0 \cup \dots \cup X_{7k+6}), q > qq - qqq\}$. It's answer set is $X_{7k+7} = A_{15}^{ts=k} \cup A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition and definitions of $A_{15}, A_{17}, A_{18}, A_9$.

    (a) where, $consumesmore(p, k)$ represents $\exists p : q = M_k(p)$,
    $q1 = \sum_{t \in T_k, p \in \bullet t} W(p, t) + \sum_{t \in T_k, p \in R(t)} M_k(p), q1 > q$, indicating place p that will be over consumed if T_k is fired, as defined in definition 23 (conflicting transitions),

    (b) $holds(p, q, k+1)$ represents $q = M_{k+1}(p)$ – by construction of Π^3

    (c) and $could\_not\_have(t, k)$ represents enabled transition t in T_k that could not be fired due to insufficient tokens

    (d) X_{7k+7} does not contain $could\_not\_have(t, k)$, when $enabled(t, k) \in X_0 \cup \dots \cup X_{7k+6}$ and $fires(t, k) \notin X_0 \cup \dots \cup X_{7k+6}$ due to construction of A, encoding of $a7$ and its body atoms. As a result it is not eliminated by the constraint $a6$

16. $eval_{U_{7k+7}}(bot_{U_{7k+8}}(\Pi^3) \setminus bot_{U_{7k+7}}(\Pi^3), X_0 \cup \dots \cup X_{7k+7}) = \{consumesmore\text{:-.} | \{consumesmore(p, 0), \dots, consumesmore(p, k)\} \cap (X_0 \cup \dots \cup X_{7k+7}) \neq \emptyset\}$. It's answer set is $X_{7k+8} = A_{16}$ – using forced atom proposition

    (a) X_{7k+8} will be empty since none of $consumesmore(p, 0), \dots,$ $consumesmore(p, k)$ hold in $X_0 \cup \dots \cup X_{7k+7}$ due to the construction of A, encoding of $a2$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \dots \cup X_{7k+8}$ is the answer set of Π^3 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 7k + 8$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^3.

Next we show (II): Given Π^3 be the encoding of a Petri Net $PN(P, T, E, W, R, I)$ with initial marking M_0, and A be an answer set of Π^3 that satisfies (A.7) and (A.8), then we can construct $X = M_0, T_0, \dots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \dots, M_i(p_n))$, where $\{holds(p_0, M_i(p_0), i), \dots holds(p_n, M_i(p_n), i)\} \subseteq A$, for $0 \leq i \leq k + 1$

2. $T_i = \{t : fires(t,i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) M_1 is T_0's target marking w.r.t. M_0.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0,0), \dots, fires(t_x,0)\}$ be the set of all $fires(\dots,0)$ atoms in A,
   (a) Then for each $fires(t_i,0) \in A$
      i. $enabled(t_i,0) \in A$ – from rule $a1$ and supported rule proposition
      ii. Then $notenabled(t_i,0) \notin A$ – from rule $e2$ and supported rule proposition
      iii. Then $body(e3)$ must not hold in A and $body(e4)$ must not hold in A – from rules $e3, e4$ and forced atom proposition
      iv. Then $q \not< n_i \equiv q \geq n_i$ in $e3$ for all $\{holds(p,q,0), ptarc(p,t_i,n_i,0)\} \subseteq A$ – from $e3$, forced atom proposition, and given facts ($holds(p,q,0) \in A, ptarc(p,t_i,n_i,0) \in A$)
      v. And $q \not\geq n_i \equiv q < n_i$ in $e4$ for all $\{holds(p,q,0), iptarc(p,t_i,n_i,0)\} \subseteq A, n_i = 1$; $q > n_i \equiv q = 0$ – from $e4$, forced atom proposition, given facts ($holds(p,q,0) \in A, iptarc(p,t_i,1,0) \in A$), and q is a positive integer
      vi. Then $(\forall p \in \bullet t_i, M_0(p) \geq W(p,t_i)) \wedge (\forall p \in I(t_i), M_0(p) = 0)$ – from
         A. $holds(p,q,0) \in A$ represents $q = M_0(p)$ – rule $i1$ construction
         B. $ptarc(p,t_i,n_i,0) \in A$ represents $n_i = W(p,t_i)$ – rule $f7$ construction; or it represents $n_i = M_0(p)$ – rule $f9$ construction; the construction of $f9$ ensures that $notenabled(t,0)$ is never true due to the reset arc
         C. definition 11 of preset $\bullet t_i$ in PN
      vii. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 26 of enabled transition
   (b) And $consumesmore \notin A$, since A is an answer set of Π^3 – from rule $a4$ and supported rule proposition
      i. Then $\nexists consumesmore(p,0) \in A$ – from rule $a3$ and supported rule proposition
      ii. Then $\nexists \{holds(p,q,0), tot\_decr(p,q1,0)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition
      iii. Then $\nexists p : (\sum_{t_i \in \{t_0,\dots,t_x\}, p \in \bullet t_i} W(p,t_i) + \sum_{t_i \in \{t_0,\dots,t_x\}, p \in R(t_i)} M_0(p)) > M_0(p)$ – from the following
         A. $holds(p,q,0)$ represents $q = M_0(p)$ – from rule $i1$ encoding, given
         B. $tot\_decr(p,q1,0) \in A$ if $\{del(p,q1_0,t_0,0), \dots, del(p,q1_x,t_x,0)\} \subseteq A$, where $q1 = q1_0 + \cdots + q1_x$ – from $r4$ and forced atom proposition

C. $del(p, q1_i, t_i, 0) \in A$ if $\{fires(t_i, 0), ptarc(p, t_i, q1_i, 0)\} \subseteq A$ – from $r7$ and supported rule proposition

D. $del(p, q1_i, t_i, 0)$ either represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_i = M_0(p)$ tokens from $p \in R(t_i)$– from rule $r7$, supported rule proposition, and definition 22 of transition execution in PN

iv. Then the set of transitions in T_0 do not conflict – by the definition 23 of conflicting transitions

(c) And for each $enabled(t_j, 0) \in A$ and $fires(t_j, 0) \notin A$, $could\_not\_have(t_j, 0) \in A$, since A is an answer set of Π^3 - from rule $a6$ and supported rule proposition

i. Then $\{enabled(t_j, 0), holds(s, qq, 0), ptarc(s, t_j, q, 0), tot\_decr(s, qqq, 0)\} \subseteq A$, such that $q > qq - qqq$ and $fires(t_j, 0) \notin A$ - from rule $a7$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q > M_0(s) - (\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_0, s \in R(t_i)} M_0(s))$, where $q = W(s, t_j)$ if $s \in \bullet t_j$, or $M_0(s)$ otherwise - from the following:

A. $ptarc(s, t_i, q, 0)$ represents $q = W(s, t_i)$ if $(s, t_i) \in E^-$ or $q = M_0(s)$ if $s \in R(t_i)$ – from rule $f7, f9$ construction

B. $holds(s, qq, 0)$ represents $qq = M_0(s)$ – from $i1$ construction

C. $tot\_decr(s, qqq, 0) \in A$ if $\{del(s, qqq_0, t_0, 0), \ldots, del(s, qqq_x, t_x, 0)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

D. $del(s, qqq_i, t_i, 0) \in A$ if $\{fires(t_i, 0), ptarc(s, t_i, qqq_i, 0)\} \subseteq A$ – from rule $r7$ and supported rule proposition

E. $del(s, qqq_i, t_i, 0)$ represents $qqq_i = W(s, t_i) : t_i \in T_0, (s, t_i) \in E^-$, or $qqq_i = M_0(t_i) : t_i \in T_0, s \in R(t_i)$ – from rule $f7, f9$ construction

F. $tot\_decr(q, qqq, 0)$ represents $\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_0, s \in R(t_i)} M_0(s)$ – from (C,D,E) above

iii. Then firing $T_0 \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_0 is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $enabled(t_r, 0) \in A$, $fires(t_r, 0) \in A$, since A is an answer set of Π^2 - from rule $f10$ and supported rule proposition

i. Then, the firing set T_0 satisfies the reset-transition requirement of definition 21 (firing set)

(e) Then $\{t_0, \ldots, t_x\} = T_0$ – using 1(a),1(b),1(d) above; and using 1(c) it is a maximal firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $holds(p, q, 1) \in A$

(a) Then $\{holds(p, q1, 0), tot\_incr(p, q2, 0), tot\_decr(p, q3, 0)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) Then $holds(p, q1, 0) \in A$ represents $q1 = M_0(p)$ – given, rule $i1$ construction; and $\{add(p, q2_0, t_0, 0), \dots, add(p, q2_j, t_j, 0)\} \subseteq A : q2_0 + \dots + q2_j = q2$ and $\{del(p, q3_0, t_0, 0), \dots, del(p, q3_l, t_l, 0)\} \subseteq A : q3_0 + \dots + q3_l = q3$ – rules $r3, r4$ and supported rule proposition, respectively

(c) Then $\{fires(t_0, 0), \dots, fires(t_j, 0)\} \subseteq A$ and $\{fires(t_0, 0), \dots, fires(t_l, 0)\} \subseteq A$ – rules $r6, r7$ and supported rule proposition, respectively

(d) Then $\{fires(t_0, 0), \dots, fires(t_j, 0)\} \cup \{fires(t_0, 0), \dots, fires(t_l, 0)\} \subseteq A = \{fires(t_0, 0), \dots, fires(t_x, 0)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

(f) Then $q = M_0(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - (\sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} M_0(p))$ – from (2b) above and the following

   i. Each $add(p, q_j, t_j, 0) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r6$ encoding, and definition 22 of transition execution in PN

   ii. Each $del(p, t_y, q_y, 0) \in A$ represents either $q_y = W(p, t_y)$ for $p \in \bullet t_y$, or $q_y = M_0(p)$ for $p \in R(t_y)$ – from rule $r7, f7$ encoding and definition 22 of transition execution in PN; or from rule $r7, f9$ encoding and definition of reset arc in PN

   iii. Each $tot\_incr(p, q2, 0) \in A$ represents $q2 = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

   iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} M_0(p)$ – aggregate assignment atom semantics in rule $r4$

(g) Then, $M_1(p) = q$ – since $holds(p, q, 1) \in A$ encodes $q = M_1(p)$ – from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set in M_k. Let $\{fires(t_0, k), \dots, fires(t_x, k)\}$ be the set of all $fires(\dots, k)$ atoms in A,

   (a) Then for each $fires(t_i, k) \in A$

      i. $enabled(t_i, k) \in A$ – from rule $a1$ and supported rule proposition

      ii. Then $notenabled(t_i, k) \notin A$ – from rule $e2$ and supported rule proposition

      iii. Then $body(e3)$ must not hold in A and $body(e4)$ must not hold in A – from rule $e3, e4$ and forced atom proposition

      iv. Then $q \not< n_i \equiv q \geq n_i$ in $e3$ for all $\{holds(p, q, k), ptarc(p, t_i, n_i, k)\} \subseteq A$ – from $e3$, forced atom proposition, and given facts ($holds(p, q, k) \in A, ptarc(p, t_i, n_i, k) \in A$)

v. And $q \not\geq n_i \equiv q < n_i$ in $e4$ for all $\{holds(p,q,k), iptarc(p,t_i,n_i,k)\} \subseteq A, n_i = 1; q > n_i \equiv q = 0$ – from $e4$, forced atom proposition, given facts $(holds(p,q,k) \in A, iptarc(p,t_i,1,k) \in A)$, and q is a positive integer

vi. Then $(\forall p \in \bullet t_i, M_k(p) \geq W(p,t_i)) \wedge (\forall p \in I(t_i), M_k(p) = 0)$ – from

A. $holds(p,q,k) \in A$ represents $q = M_k(p)$ – inductive assumption, given

B. $ptarc(p,t_i,n_i,k) \in A$ represents $n_i = W(p,t_i)$ – rule $f7$ construction; or it represents $n_i = M_k(p)$ – rule $f9$ construction; the construction of $f9$ ensures that $notenabled(t,0)$ is never true due to the reset arc

C. definition 11 of preset $\bullet t_i$ in PN

vii. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 26 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^3 – from rule $a4$ and supported rule proposition

i. Then $\nexists consumesmore(p,k) \in A$ – from rule $a3$ and supported rule proposition

ii. Then $\nexists\{holds(p,q,k), tot\_decr(p,q1,k)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition

iii. Then $\nexists p : (\sum_{t_i \in \{t_0,\dots,t_x\}, p \in \bullet t_i} W(p,t_i) + \sum_{t_i \in \{t_0,\dots,t_x\}, p \in R(t_i)} M_k(p)) > M_k(p)$ – from the following

A. $holds(p,q,k)$ represents $q = M_k(p)$ – inductive assumption, construction

B. $tot\_decr(p,q1,k) \in A$ if $\{del(p,q1_0,t_0,k), \dots, del(p,q1_x,t_x,k)\} \subseteq A$, where $q1 = q1_0 + \dots + q1_x$ – from $r4$ and forced atom proposition

C. $del(p,q1_i,t_i,k) \in A$ if $\{fires(t_i,k), ptarc(p,t_i,q1_i,k)\} \subseteq A$ – from $r7$ and supported rule proposition

D. $del(p,q1_i,t_i,k)$ either represents removal of $q1_i = W(p,t_i)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_i = M_k(p)$ tokens from $p \in R(t_i)$– from rule $r7$, supported rule proposition, and definition 22 of transition execution in PN

iv. Then T_k does not contain conflicting transitions – by the definition 23 of conflicting transitions

(c) And for each $enabled(t_j,k) \in A$ and $fires(t_j,k) \notin A$, $could\_not\_have(t_j,k) \in A$, since A is an answer set of Π^3 - from rule $a6$ and supported rule proposition

i. Then $\{enabled(t_j,k), holds(s,qq,k), ptarc(s,t_j,q,k), tot\_decr(s,qqq,k)\} \subseteq A$, such that $q > qq - qqq$ and $fires(t_j,k) \notin A$ - from rule $a7$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q > M_k(s) - (\sum_{t_i \in T_k, s \in \bullet t_i} W(s,t_i) + \sum_{t_i \in T_k, s \in R(t_i)} M_k(s))$, where $q = W(s,t_j)$ if $s \in \bullet t_j$, or $M_k(s)$ otherwise - from the following:

A. $ptarc(s, t_i, q, k)$ represents $q = W(s, t_i)$ if $(s, t_i) \in E^-$ or $q = M_k(s)$ if $s \in R(t_i)$ – from rule $f7, f9$ construction

B. $holds(s, qq, k)$ represents $qq = M_k(s)$ – construction

C. $tot\_decr(s, qqq, k) \in A$ if $\{del(s, qqq_0, t_0, k), \ldots, del(s, qqq_x, t_x, k)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

D. $del(s, qqq_i, t_i, k) \in A$ if $\{fires(t_i, k), ptarc(s, t_i, qqq_i, k)\} \subseteq A$ – from rule $r7$ and supported rule proposition

E. $del(s, qqq_i, t_i, k)$ represents $qqq_i = W(s, t_i) : t_i \in T_k, (s, t_i) \in E^-$, or $qqq_i = M_k(t_i) : t_i \in T_k, s \in R(t_i)$ – from rule $f7, f9$ construction

F. $tot\_decr(q, qqq, k)$ represents $\sum_{t_i \in T_k, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_k, s \in R(t_i)} M_k(s)$ – from (C,D,E) above

iii. Then firing $T_k \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_k is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $enabled(t_r, k) \in A$, $fires(t_r, k) \in A$, since A is an answer set of Π^2 - from rule $f10$ and supported rule proposition

i. Then the firing set T_k satisfies the reset transition requirement of definition 21 (firing set)

(e) Then $\{t_0, \ldots, t_x\} = T_k$ – using 1(a),1(b), 1(d) above; and using 1(c) it is a maximal firing set

2. We show that M_{k+1} is produced by firing T_k in M_k. Let $holds(p, q, k+1) \in A$

(a) Then $\{holds(p, q1, k), tot\_incr(p, q2, k), tot\_decr(p, q3, k)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) Then, $holds(p, q1, k) \in A$ represents $q1 = M_k(p)$ – inductive assumption, construction ; and $\{add(p, q2_0, t_0, k), \ldots, add(p, q2_j, t_j, k)\} \subseteq A : q2_0 + \cdots + q2_j = q2$ and $\{del(p, q3_0, t_0, k), \ldots, del(p, q3_l, t_l, k)\} \subseteq A : q3_0 + \cdots + q3_l = q3$ – rules $r3, r4$ and supported rule proposition, respectively

(c) Then $\{fires(t_0, k), \ldots, fires(t_j, k)\} \subseteq A$ and $\{fires(t_0, k), \ldots, fires(t_l, k)\} \subseteq A$ – rules $r6, r7$ and supported rule proposition, respectively

(d) Then $\{fires(t_0, k), \ldots, fires(t_j, k)\} \cup \{fires(t_0, k), \ldots, fires(t_l, k)\} \subseteq A = \{fires(t_0, k), \ldots, fires(t_x, k)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x, k) \in A$ we have $t_x \in T_k$ – already shown in item 1 above

(f) Then $q = M_k(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - (\sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_k \wedge p \in R(t_x)} M_k(p))$ – from (2b) above and the following

i. Each $add(p, q_j, t_j, k) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r6$ encoding, and definition 22 of transition execution in PN

ii. Each $del(p, t_y, q_y, k) \in A$ represents either $q_y = W(p, t_y)$ for $p \in \bullet t_y$, or $q_y = M_k(p)$ for $p \in R(t_y)$ – from rule $r7, f7$ encoding and definition 22 of transition execution in PN; or from rule $r7, f9$ encoding and definition of reset arc in PN

iii. Each $tot\_incr(p, q2, k) \in A$ represents $q2 = \sum_{t_x \in T_k \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_k \wedge p \in R(t_x)} M_k(p)$ – aggregate assignment atom semantics in rule $r4$

(g) Then, $M_{k+1}(p) = q$ – since $holds(p, q, k+1) \in A$ encodes $q = M_{k+1}(p)$ – from construction

As a result, for any $n > k$, T_n will be a valid firing set for M_n and M_{n+1} will be its target marking.
Conclusion: Since both (I) and (II) hold, $X = M_0, T_0, M_1, \dots, M_k, T_{k+1}$ is an execution sequence of $PN(P, T, E, W, R)$ (w.r.t M_0) iff there is an answer set A of $\Pi^3(PN, M_0, k, ntok)$ such that (A.5) and (A.6) hold.

A.5 Proof of Proposition 5

Let $PN = (P, T, E, W, R, I, Q, QW)$ be a Petri Net, M_0 be its initial marking and let $\Pi^4(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.7. Then $X = M_0, T_0, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t. M_0) iff there is an answer set A of $\Pi^4(PN, M_0, k, ntok)$ such that:

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\} \tag{A.9}$$

$$\begin{aligned} \{holds(p, q, ts) : p \in P, q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ = \{holds(p, q, ts) : holds(p, q, ts) \in A\} \end{aligned} \tag{A.10}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.9) and (A.10) and show that A is an answer set of Π^4

(II) Given an answer set A of Π^4, we create an execution sequence X such that (A.9) and (A.10) are satisfied.

First we show (I): Given PN and an execution sequence X of PN, we create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$
2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$
3. $A_3 = \{place(p) : p \in P\}$
4. $A_4 = \{trans(t) : t \in T\}$
5. $A_5 = \{ptarc(p, t, n, ts) : (p, t) \in E^-, n = W(p, t), 0 \leq ts \leq k\}$, where $E^- \subseteq E$
6. $A_6 = \{tparc(t, p, n, ts) : (t, p) \in E^+, n = W(t, p), 0 \leq ts \leq k\}$, where $E^+ \subseteq E$

7. $A_7 = \{holds(p, q, 0) : p \in P, q = M_0(p)\}$

8. $A_8 = \{notenabled(t, ts) : t \in T, 0 \leq ts \leq k, (\exists p \in \bullet t, M_{ts}(p) < W(p, t)) \vee (\exists p \in I(t), M_{ts}(p) \neq 0) \vee (\exists (p, t) \in Q, M_{ts}(p) < QW(p, t))\}$
   per definition 28 (enabled transition)

9. $A_9 = \{enabled(t, ts) : t \in T, 0 \leq ts \leq k, (\forall p \in \bullet t, W(p, t) \leq M_{ts}(p)) \wedge (\forall p \in I(t), M_{ts}(p) = 0) \wedge (\forall (p, t) \in Q, M_{ts}(p) \geq QW(p, t))\}$
   per definition 28 (enabled transition)

10. $A_{10} = \{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
    per definition 28 (enabled transitions), only an enabled transition may fire

11. $A_{11} = \{add(p, q, t, ts) : t \in T_{ts}, p \in t\bullet, q = W(t, p), 0 \leq ts \leq k\}$
    per definition 22 (transition execution)

12. $A_{12} = \{del(p, q, t, ts) : t \in T_{ts}, p \in \bullet t, q = W(p, t), 0 \leq ts \leq k\} \cup \{del(p, q, t, ts) : t \in T_{ts}, p \in R(t), q = M_{ts}(p), 0 \leq ts \leq k\}$
    per definition 22 (transition execution)

13. $A_{13} = \{tot\_incr(p, q, ts) : p \in P, q = \sum_{t \in T_{ts}, p \in t\bullet} W(t, p), 0 \leq ts \leq k\}$
    per definition 24 (firing set execution)

14. $A_{14} = \{tot\_decr(p, q, ts) : p \in P, q = \sum_{t \in T_{ts}, p \in \bullet t} W(p, t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), 0 \leq ts \leq k\}$
    per definition 24 (firing set execution)

15. $A_{15} = \{consumesmore(p, ts) : p \in P, q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p, t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p), q1 > q, 0 \leq ts \leq k\}$
    per definition 23 (conflicting transitions)

16. $A_{16} = \{consumesmore : \exists p \in P : q = M_{ts}(p), q1 = \sum_{t \in T_{ts}, p \in \bullet t} W(p, t) + \sum_{t \in T_{ts}, p \in R(t)}(M_{ts}(p)), q1 > q, 0 \leq ts \leq k\}$
    per definition 23 (conflicting transitions)

17. $A_{17} = \{could\_not\_have(t, ts) : t \in T, (\forall p \in \bullet t, W(p, t) \leq M_{ts}(p)), t \not\in T_{ts}, (\exists p \in \bullet t : W(p, t) > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p, t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), 0 \leq ts \leq k\}$
    per the maximal firing set semantics

18. $A_{18} = \{holds(p, q, ts+1) : p \in P, q = M_{ts+1}(p), 0 \leq ts < k\}$,
    where $M_{ts+1}(p) = M_{ts}(p) - (\sum_{t \in T_{ts}, p \in \bullet t} W(p, t) +$
    $\sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p)) + \sum_{t \in T_{ts}, p \in t\bullet} W(t, p)$
    according to definition 21 (firing set execution)

19. $A_{19} = \{ptarc(p, t, n, ts) : p \in R(t), n = M_{ts}(p), n > 0, 0 \leq ts \leq k\}$

20. $A_{20} = \{iptarc(p, t, 1, ts) : p \in P, 0 \leq ts < k\}$

21. $A_{21} = \{\mathit{tptarc}(p,t,n,ts) : (p,t) \in Q, n = QW(p,t), 0 \leq ts \leq k\}$

We show that A satisfies (A.7) and (A.8), and A is an answer set of Π^4.

A satisfies (A.7) and (A.8) by its construction above. We show A is an answer set of Π^4 by splitting. We split $\mathit{lit}(\Pi^4)$ into a sequence of $7k+9$ sets:

- $U_0 = \mathit{head}(f1) \cup \mathit{head}(f2) \cup \mathit{head}(f5) \cup \mathit{head}(f6) \cup \mathit{head}(i1) = \{\mathit{place}(p) : p \in P\} \cup \{\mathit{trans}(t) : t \in T\} \cup \{\mathit{time}(0), \ldots, \mathit{time}(k)\} \cup \{\mathit{num}(0), \ldots, \mathit{num}(\mathit{ntok})\} \cup \{\mathit{holds}(p,q,0) : p \in P, q = M_0(p)\}$

- $U_{7k+1} = U_{7k+0} \cup \mathit{head}(f7)^{ts=k} \cup \mathit{head}(f8)^{ts=k} \cup \mathit{head}(f9)^{ts=k} \cup \mathit{head}(f11)^{ts=k} \cup \mathit{head}(f12)^{ts=k} = U_{7k+0} \cup \{\mathit{ptarc}(p,t,n,k) : (p,t) \in E^-, n = W(p,t)\} \cup \{\mathit{tparc}(t,p,n,k) : (t,p) \in E^+, n = W(t,p)\} \cup \{\mathit{ptarc}(p,t,n,k) : p \in R(t), n = M_k(p), n > 0\} \cup \{\mathit{iptarc}(p,t,1,k) : p \in I(t)\} \cup \{\mathit{tptarc}(p,t,n,k) : (p,t) \in Q, n = QW(p,t)\}$

- $U_{7k+2} = U_{7k+1} \cup \mathit{head}(e3)^{ts=k} = U_{7k+1} \cup \{\mathit{notenabled}(t,k) : t \in T\}$

- $U_{7k+3} = U_{7k+2} \cup \mathit{head}(e2)^{ts=k} = U_{7k+2} \cup \{\mathit{enabled}(t,k) : t \in T\}$

- $U_{7k+4} = U_{7k+3} \cup \mathit{head}(a1)^{ts=k} = U_{7k+3} \cup \{\mathit{fires}(t,k) : t \in T\}$

- $U_{7k+5} = U_{7k+4} \cup \mathit{head}(r6)^{ts=k} \cup \mathit{head}(r7)^{ts=k} = U_{7k+4} \cup \{\mathit{add}(p,q,t,k) : p \in P, t \in T, q = W(t,p)\} \cup \{\mathit{del}(p,q,t,k) : p \in P, t \in T, q = W(p,t)\} \cup \{\mathit{del}(p,q,t,k) : p \in P, t \in T, q = M_k(p)\}$

- $U_{7k+6} = U_{7k+5} \cup \mathit{head}(r3)^{ts=k} \cup \mathit{head}(r4)^{ts=k} = U_{7k+5} \cup \{\mathit{tot\_incr}(p,q,k) : p \in P, 0 \leq q \leq \mathit{ntok}\} \cup \{\mathit{tot\_decr}(p,q,k) : p \in P, 0 \leq q \leq \mathit{ntok}\}$

- $U_{7k+7} = U_{7k+6} \cup \mathit{head}(r5)^{ts=k} \cup \mathit{head}(a2)^{ts=k} \cup \mathit{head}(a7)^{ts=k} = U_{7k+6} \cup \{\mathit{consumesmore}(p,k) : p \in P\} \cup \{\mathit{holds}(p,q,k+1) : p \in P, 0 \leq q \leq \mathit{ntok}\} \cup \{\mathit{could\_not\_have}(t,k) : t \in T\}$

- $U_{7k+8} = U_{7k+7} \cup \mathit{head}(a3) = U_{7k+7} \cup \{\mathit{consumesmore}\}$

where $\mathit{head}(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\ldots,ts) : a(\ldots,ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 7k+8$ form a splitting sequence, since each U_i is a splitting set of Π^4, and $\langle U_\alpha \rangle_{\alpha < \mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{8(k+1)}$ and $\bigcup_{\alpha < \mu} U_\alpha = \mathit{lit}(\Pi^4)$.

We compute the answer set of Π^4 using the splitting sets as follows:

1. $\mathit{bot}_{U_0}(\Pi^4) = f1 \cup f2 \cup f5 \cup i1 \cup f6$ and $X_0 = A_1 \cup \cdots \cup A_4 \cup A_7$ $(= U_0)$ is its answer set – using forced atom proposition

2. $\mathit{eval}_{U_0}(\mathit{bot}_{U_1}(\Pi^4) \setminus \mathit{bot}_{U_0}(\Pi^4), X_0) = \{\mathit{ptarc}(p,t,q,0)\text{:-.}|q = W(p,t)\} \cup \{\mathit{tparc}(t,p,q,0)\text{:-.}|q = W(t,p)\} \cup \{\mathit{ptarc}(p,t,q,0)\text{:-.}|q = M_0(p)\} \cup \{\mathit{iptarc}(p,t,1,0)\text{:-.}\} \cup \{\mathit{tptarc}(p,t,q,0)\text{:-.}|q = QW(p,t)\}$. Its answer set $X_1 = A_5^{ts=0} \cup A_6^{ts=0} \cup A_{19}^{ts=0} \cup A_{20}^{ts=0} \cup A_{21}^{ts=0}$ – using forced atom proposition and construction of $A_5, A_6, A_{19}, A_{20}, A_{21}$.

3. $eval_{U_1}(bot_{U_2}(\Pi^4) \setminus bot_{U_1}(\Pi^4), X_0 \cup X_1) = \{notenabled(t,0)$:-.$|(\{trans(t),$ $ptarc(p,t,n,0), holds(p,q,0)\} \subseteq X_0 \cup X_1,$ where $q < n)$ or $(\{notenabled(t,0)$:-.$|$ $(\{trans(t), iptarc(p,t,n2,0), holds(p,q,0)\} \subseteq X_0 \cup X_1,$ where $q \geq n2\})$ or $(\{trans(t), tptarc(p,t,n3,0), holds(p,q,0)\} \subseteq X_0 \cup X_1,$ where $q < n3)\}$. Its answer set $X_2 = A_8^{ts=0}$ – using forced atom proposition and construction of A_8.

   (a) where, $q = M_0(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of $i1$ and $f7$ in Π^4, and

   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

   (c) $n2 = 1$ – by construction of *iptarc* predicates in Π^4, meaning $q \geq n2 \equiv q \geq 1 \equiv q > 0$,

   (d) $tptarc(p,t,n3,0)$ represents $n3 = QW(p,t)$, where $(p,t) \in Q$

   (e) thus, $notenabled(t,0) \in X_1$ represents $(\exists p \in \bullet t : M_0(p) < W(p,t)) \vee (\exists p \in I(t) : M_0(p) > 0) \vee (\exists (p,t) \in Q : M_{ts}(p) < QW(p,t))$.

4. $eval_{U_2}(bot_{U_3}(\Pi^4) \setminus bot_{U_2}(\Pi^4), X_0 \cup \cdots \cup X_2) = \{enabled(t,0)$:-.$|trans(t) \in X_0 \cup \cdots \cup X_2, notenabled(t,0) \notin X_0 \cup \cdots \cup X_2\}$. Its answer set is $X_3 = A_9^{ts=0}$ – using forced atom proposition and construction of A_9.

   (a) since an $enabled(t,0) \in X_3$ if $\nexists$ $notenabled(t,0) \in X_0 \cup \cdots \cup X_2$; which is equivalent to $(\nexists p \in \bullet t : M_0(p) < W(p,t)) \wedge (\nexists p \in I(t) : M_0(p) > 0) \wedge (\nexists (p,t) \in Q : M_0(p) < QW(p,t)) \equiv (\forall p \in \bullet t : M_0(p) \geq W(p,t)) \wedge (\forall p \in I(t) : M_0(p) = 0)$.

5. $eval_{U_3}(bot_{U_4}(\Pi^4) \setminus bot_{U_3}(\Pi^4), X_0 \cup \cdots \cup X_3) = \{\{fires(t,0)\}$:-.$|enabled(t,0)$ holds in $X_0 \cup \cdots \cup X_3\}$. It has multiple answer sets $X_{4.1}, \ldots, X_{4.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $eval_{U_3}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^4 determined using splitting is equal to A, we only consider the set that matches the $fires(t,0)$ elements in A and call it X_4, ignoring the rest. Thus, $X_4 = A_{10}^{ts=0}$, representing T_0.

   (a) in addition, for every t such that $enabled(t,0) \in X_0 \cup \cdots \cup X_3, R(t) \neq \emptyset$; $fires(t,0) \in X_4$ – per definition 21 (firing set); requiring that a reset transition is fired when enabled

   (b) thus, the firing set T_0 will not be eliminated by the constraint $f10$

6. $eval_{U_4}(bot_{U_5}(\Pi^4) \setminus bot_{U_4}(\Pi^4), X_0 \cup \cdots \cup X_4) = \{add(p,n,t,0)$:-.$|\{fires(t,0),$ $tparc(t,p,n,0)\} \subseteq X_0 \cup \cdots \cup X_4\} \cup \{del(p,n,t,0)$:-.$|\{fires(t,0), ptarc(p,t,n,0)\}$ $\subseteq X_0 \cup \cdots \cup X_4\}$. It's answer set is $X_5 = A_{11}^{ts=0} \cup A_{12}^{ts=0}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

   (a) where, each *add* atom is equivalent to $n = W(t,p), p \in t\bullet$,

   (b) and each *del* atom is equivalent to $n = W(p,t), p \in \bullet t$; or $n = M_k(p), p \in R(t)$,

   (c) representing the effect of transitions in T_0 – by construction

7. $eval_{U_5}(bot_{U_6}(\Pi^4) \setminus bot_{U_5}(\Pi^4), X_0 \cup \cdots \cup X_5) = \{tot\_incr(p,qq,0)\text{:-}.| qq = \sum_{add(p,q,t,0) \in X_0 \cup \cdots \cup X_5} q\} \cup \{tot\_decr(p,qq,0)\text{:-}.|qq = \sum_{del(p,q,t,0) \in X_0 \cup \cdots \cup X_5} q\}$. It's answer set is $X_6 = A_{13}^{ts=0} \cup A_{14}^{ts=0}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

   (a) where, each $tot\_incr(p,qq,0)$, $qq = \sum_{add(p,q,t,0) \in X_0 \cup \ldots X_5} q$ $\equiv qq = \sum_{t \in X_4, p \in t\bullet} W(p,t)$,

   (b) and each $tot\_decr(p,qq,0)$, $qq = \sum_{del(p,q,t,0) \in X_0 \cup \ldots X_5} q$ $\equiv qq = \sum_{t \in X_4, p \in \bullet t} W(t,p) + \sum_{t \in X_4, p \in R(t)} M_k(p)$,

   (c) represent the net effect of transitions in T_0 – by construction

8. $eval_{U_6}(bot_{U_7}(\Pi^4) \setminus bot_{U_6}(\Pi^4), X_0 \cup \cdots \cup X_6) = \{consumesmore(p,0)\text{:-}.| \{holds(p,q,0), tot\_decr(p,q1,0)\} \subseteq X_0 \cup \cdots \cup X_6, q1 > q\} \cup \{holds(p,q,1)\text{:-}.,| \{holds(p,q1,0), tot\_incr(p,q2,0), tot\_decr(p,q3,0)\} \subseteq X_0 \cup \cdots \cup X_6, q = q1 + q2 - q3\} \cup \{could\_not\_have(t,0)\text{:-}.|\{enabled(t,0), ptarc(s,t,q), holds(s,qq,0), tot\_decr(s,qqq,0)\} \subseteq X_0 \cup \cdots \cup X_6, fires(t,0) \notin (X_0 \cup \cdots \cup X_6), q > qq - qqq\}$. It's answer set is $X_7 = A_{15}^{ts=0} \cup A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of $A_{15}, A_{17}, A_{18}, A_9$.

   (a) where, $consumesmore(p,0)$ represents $\exists p : q = M_0(p), q1 = \sum_{t \in T_0, p \in \bullet t} W(p,t) + \sum_{t \in T_0, p \in R(t)} M_0(p), q1 > q$, indicating place p will be over consumed if T_0 is fired, as defined in definition 23 (conflicting transitions),

   (b) $holds(p,q,1)$ represents $q = M_1(p)$ – by construction of Π^4,

   (c) and $could\_not\_have(t,0)$ represents enabled transition $t \in T_0$ that could not fire due to insufficient tokens

   (d) X_7 does not contain $could\_not\_have(t,0)$, when $enabled(t,0) \in X_0 \cup \cdots \cup X_6$ and $fires(t,0) \notin X_0 \cup \cdots \cup X_6$ due to construction of A, encoding of $a7$ and its body atoms. As a result it is not eliminated by the constraint $a6$

$\vdots$

9. $eval_{U_{7k+0}}(bot_{U_{7k+1}}(\Pi^4) \setminus bot_{U_{7k+0}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+0}) = \{ptarc(p,t,q,k)\text{:-}.|q = W(p,t)\} \cup \{tparc(t,p,q,k)\text{:-}.|q = W(t,p)\} \cup \{ptarc(p,t,q,k)\text{:-}.|q = M_k(p)\} \cup \{iptarc(p,t,1,k)\text{:-}.\}$. Its answer set $X_{7k+1} = A_5^{ts=k} \cup A_6^{ts=k} \cup A_{19}^{ts=k} \cup A_{20}^{ts=k}$ – using forced atom proposition and construction of A_5, A_6, A_{19}, A_{20}.

10. $eval_{U_{7k+1}}(bot_{U_{7k+2}}(\Pi^4) \setminus bot_{U_{7k+1}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+1}) = \{notenabled(t,k)\text{:-}.| (\{trans(t), ptarc(p,t,n,k), holds(p,q,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q < n)$ or $\{notenabled(t,k)\text{:-}.|(\{trans(t), iptarc(p,t,n2,k), holds(p,q,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q \geq n2\}\}$. Its answer set $X_{7k+2} = A_8^{ts=k}$ – using forced atom proposition and construction of A_8.

    (a) where, $q = M_k(p)$, and $n = W(p,t)$ for an arc $(p,t) \in E^-$ – by construction of $holds$ and $ptarc$ predicates in Π^4, and

(b) in an arc $(p, t) \in E^-$, $p \in \bullet t$ (by definition 11 of preset)

(c) $n2 = 1$ – by construction of *iptarc* predicates in Π^4, meaning $q \geq n2 \equiv q \geq 1 \equiv q > 0$,

(d) thus, $notenabled(t, k) \in X_{7k+1}$ represents $(\exists p \in \bullet t : M_k(p) < W(p, t)) \vee (\exists p \in I(t) : M_k(p) > k)$.

11. $eval_{U_{7k+2}}(bot_{U_{7k+3}}(\Pi^4) \setminus bot_{U_{7k+2}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+2}) = \{enabled(t, k)\text{:-.}| trans(t) \in X_0 \cup \cdots \cup X_{7k+2} \wedge notenabled(t, k) \notin X_0 \cup \cdots \cup X_{7k+2}\}$. Its answer set is $X_{7k+3} = A_9^{ts=k}$ – using forced atom proposition and construction of A_9.

    (a) since an $enabled(t, k) \in X_{7k+3}$ if $\nexists$ $notenabled(t, k) \in X_0 \cup \cdots \cup X_{7k+2}$; which is equivalent to $(\nexists p \in \bullet t : M_k(p) < W(p, t)) \wedge (\nexists p \in I(t) : M_k(p) > k) \equiv (\forall p \in \bullet t : M_k(p) \geq W(p, t)) \wedge (\forall p \in I(t) : M_k(p) = k)$.

12. $eval_{U_{7k+3}}(bot_{U_{7k+4}}(\Pi^4) \setminus bot_{U_{7k+3}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+3}) = \{\{fires(t, k)\}\text{:-.}|$
$enabled(t, k)$
 holds in $X_0 \cup \cdots \cup X_{7k+3}\}$. It has multiple answer sets $X_{7k+4.1}, \ldots, X_{7k+4.n}$, corresponding to elements of power set of $fires(t, k)$ atoms in $eval_{U_{7k+3}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^4 determined using splitting is equal to A, we only consider the set that matches the $fires(t, k)$ elements in A and call it X_{7k+4}, ignoring the rest. Thus, $X_{7k+4} = A_{10}^{ts=k}$, representing T_k.

    (a) in addition, for every t such that $enabled(t, k) \in X_0 \cup \cdots \cup X_{7k+3}$, $R(t) \neq \emptyset$; $fires(t, k) \in X_{7k+4}$ – per definition 21 (firing set); requiring that a reset transition is fired when enabled

    (b) thus, the firing set T_k will not be eliminated by the constraint $f10$

13. $eval_{U_{7k+4}}(bot_{U_{7k+5}}(\Pi^4) \setminus bot_{U_{7k+4}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+4}) = \{add(p, n, t, k)\text{:-.}| \{fires(t, k), tparc(t, p, n, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+4}\} \cup \{del(p, n, t, k)\text{:-.}|\{fires(t, k), ptarc(p, t, n, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+4}\}$. It's answer set is $X_{7k+5} = A_{11}^{ts=k} \cup A_{12}^{ts=k}$ – using forced atom proposition and definitions of A_{11} and A_{12}.

    (a) where, each *add* atom is equivalent to $n = W(t, p) : p \in t\bullet$,

    (b) and each *del* atom is equivalent to $n = W(p, t) : p \in \bullet t$; or $n = M_k(p) : p \in R(t)$,

    (c) representing the effect of transitions in T_k

14. $eval_{U_{7k+5}}(bot_{U_{7k+6}}(\Pi^4) \setminus bot_{U_{7k+5}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+5}) =$
$\{tot\_incr(p, qq, k)\text{:-.}|qq = \sum_{add(p,q,t,k) \in X_0 \cup \cdots \cup X_{7k+5}} q\} \cup$
$\{tot\_decr(p, qq, k)\text{:-.}|qq = \sum_{del(p,q,t,k) \in X_0 \cup \cdots \cup X_{7k+5}} q\}$. It's answer set is $X_{7k+6} = A_{13}^{ts=k} \cup A_{14}^{ts=k}$ – using forced atom proposition and definitions of A_{13} and A_{14}.

    (a) where, each $tot\_incr(p, qq, k)$, $qq = \sum_{add(p,q,t,k) \in X_0 \cup \ldots X_{7k+5}} q$
    $\equiv qq = \sum_{t \in X_{7k+4}, p \in t\bullet} W(p, t)$,

    (b) and each $tot\_decr(p, qq, k)$, $qq = \sum_{del(p,q,t,k) \in X_0 \cup \ldots X_{7k+5}} q$
    $\equiv qq = \sum_{t \in X_{7k+4}, p \in \bullet t} W(t, p) + \sum_{t \in X_{7k+4}, p \in R(t)} M_k(p)$,

(c) represent the net effect of transition in T_k

15. $eval_{U_{7k+6}}(bot_{U_{7k+7}}(\Pi^4) \setminus bot_{U_{7k+6}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+6}) = \{consumesmore(p,k){:}\text{-}.| \{holds(p,q,k), tot\_decr(p,q1,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+6}, q1 > q\} \cup \{holds(p,q,1){:}\text{-}.,| \{holds(p,q1,k), tot\_incr(p,q2,k), tot\_decr(p,q3,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+6}, q = q1 + q2 - q3\} \cup \{could\_not\_have(t,k){:}\text{-}.|\{enabled(t,k), ptarc(s,t,q), holds(s,qq,k), tot\_decr(s,qqq,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+6}, fires(t,k) \notin (X_0 \cup \cdots \cup X_{7k+6}), q > qq - qqq\}$. It's answer set is $X_{7k+7} = A_{15}^{ts=k} \cup A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition and definitions of $A_{15}, A_{17}, A_{18}, A_9$.

    (a) where, $consumesmore(p,k)$ represents $\exists p : q = M_k(p), q1 = \sum_{t \in T_k, p \in \bullet t} W(p,t) + \sum_{t \in T_k, p \in R(t)} M_k(p), q1 > q$, indicating place p that will be over consumed if T_k is fired, as defined in definition 23 (conflicting transitions),

    (b) $holds(p,q,k+1)$ represents $q = M_{k+1}(p)$ – by construction of Π^4,

    (c) and $could\_not\_have(t,k)$ represents enabled transition t in T_k that could not be fired due to insufficient tokens

    (d) X_{7k+7} does not contain $could\_not\_have(t,k)$, when $enabled(t,k) \in X_0 \cup \cdots \cup X_{7k+6}$ and $fires(t,k) \notin X_0 \cup \cdots \cup X_{7k+6}$ due to construction of A, encoding of $a7$ and its body atoms. As a result it is not eliminated by the constraint $a6$

16. $eval_{U_{7k+7}}(bot_{U_{7k+8}}(\Pi^4) \setminus bot_{U_{7k+7}}(\Pi^4), X_0 \cup \cdots \cup X_{7k+7}) = \{consumesmore{:}\text{-}.| \{consumesmore(p,0), \ldots, consumesmore(p,k)\} \cap (X_0 \cup \cdots \cup X_{7k+7}) \neq \emptyset\}$. It's answer set is $X_{7k+8} = A_{16}$ – using forced atom proposition

    (a) X_{7k+8} will be empty since none of $consumesmore(p,0), \ldots,$ $consumesmore(p,k)$ hold in $X_0 \cup \cdots \cup X_{7k+8}$ due to the construction of A, encoding of $a2$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \cdots \cup X_{7k+8}$ is the answer set of Π^4 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 7k+8$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^4.

Next we show (II): Given Π^4 be the encoding of a Petri Net $PN(P,T,E,W,R,I)$ with initial marking M_0, and A be an answer set of Π^4 that satisfies (A.7) and (A.8), then we can construct $X = M_0, T_0, \ldots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \ldots, M_i(p_n))$, where $\{holds(p_0, M_i(p_0), i), \ldots holds(p_n, M_i(p_n), i)\} \subseteq A$, for $0 \leq i \leq k+1$

2. $T_i = \{t : fires(t,i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) firing of T_0 in M_0 produces marking M_1.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0, 0), \ldots, fires(t_x, 0)\}$ be the set of all $fires(\ldots, 0)$ atoms in A,

    (a) Then for each $fires(t_i, 0) \in A$

    i. $enabled(t_i, 0) \in A$ – from rule $a1$ and supported rule proposition

    ii. Then $notenabled(t_i, 0) \notin A$ – from rule $e2$ and supported rule proposition

    iii. Then either of $body(e3)$, $body(e4)$, or $body(e5)$ must not hold in A – from rules $e3, e4, e5$ and forced atom proposition

    iv. Then $q \not< n_i \equiv q \geq n_i$ in $e3$ for all $\{holds(p, q, 0), ptarc(p, t_i, n_i, 0)\} \subseteq A$ – from $e3$, forced atom proposition, and given facts ($holds(p, q, 0) \in A, ptarc(p, t_i, n_i, 0) \in A$)

    v. And $q \not> n_i \equiv q < n_i$ in $e4$ for all $\{holds(p, q, 0), iptarc(p, t_i, n_i, 0)\} \subseteq A, n_i = 1$; $q > n_i \equiv q = 0$ – from $e4$, forced atom proposition, given facts ($holds(p, q, 0) \in A, iptarc(p, t_i, 1, 0) \in A$), and q is a positive integer

    vi. And $q \not< n_i \equiv q \geq n_i$ in $e5$ for all $\{holds(p, q, 0), tptarc(p, t_i, n_i, 0)\} \subseteq A$ – from $e5$, forced atom proposition, and given facts

    vii. Then $(\forall p \in \bullet t_i, M_0(p) \geq W(p, t_i)) \wedge (\forall p \in I(t_i), M_0(p) = 0) \wedge (\forall (p, t_i) \in Q, M_0(p) \geq QW(p, t_i))$ – from the following

    A. $holds(p, q, 0) \in A$ represents $q = M_0(p)$ – rule $i1$ construction

    B. $ptarc(p, t_i, n_i, 0) \in A$ represents $n_i = W(p, t_i)$ – rule $f7$ construction; or it represents $n_i = M_0(p)$ – rule $f9$ construction; the construction of $f9$ ensures that $notenabled(t, 0)$ is never true due to the reset arc

    C. definition 11 of preset $\bullet t_i$ in PN

    D. definition 28 of enabled transition in PN

    viii. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 28 of enabled transition

    (b) And $consumesmore \notin A$, since A is an answer set of Π^4 – from rule $a4$ and supported rule proposition

    i. Then $\nexists consumesmore(p, 0) \in A$ – from rule $a3$ and supported rule proposition

    ii. Then $\nexists \{holds(p, q, 0), tot\_decr(p, q1, 0)\} \subseteq A : q1 > q$ in $body(a2)$ – from $a2$ and forced atom proposition

    iii. Then $\nexists p : (\sum_{t_i \in \{t_0, \ldots, t_x\}, p \in \bullet t_i} W(p, t_i) + \sum_{t_i \in \{t_0, \ldots, t_x\}, p \in R(t_i)} M_0(p)) > M_0(p)$ – from the following

        A. $\mathit{holds}(p, q, 0)$ represents $q = M_0(p)$ – from rule $i1$ construction, given
        B. $\mathit{tot\_decr}(p, q1, 0) \in A$ if $\{\mathit{del}(p, q1_0, t_0, 0), \dots, \mathit{del}(p, q1_x, t_x, 0)\} \subseteq A$, where $q1 = q1_0 + \dots + q1_x$ – from $r4$ and forced atom proposition
        C. $\mathit{del}(p, q1_i, t_i, 0) \in A$ if $\{\mathit{fires}(t_i, 0), \mathit{ptarc}(p, t_i, q1_i, 0)\} \subseteq A$ – from $r7$ and supported rule proposition
        D. $\mathit{del}(p, q1_i, t_i, 0)$ represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_i = M_0(p)$ tokens from $p \in R(t_i)$– from rule $r7$, supported rule proposition, and definition 22 of transition execution in PN
    iv. Then the set of transitions in T_0 do not conflict – by the definition 23 of conflicting transitions

(c) And for each $\mathit{enabled}(t_j, 0) \in A$ and $\mathit{fires}(t_j, 0) \notin A$, $\mathit{could\_not\_have}(t_j, 0) \in A$, since A is an answer set of Π^4 - from rule $a6$ and supported rule proposition
    i. Then $\{\mathit{enabled}(t_j, 0), \mathit{holds}(s, qq, 0), \mathit{ptarc}(s, t_j, q, 0), \mathit{tot\_decr}(s, qqq, 0)\} \subseteq A$, such that $q > qq - qqq$ and $\mathit{fires}(t_j, 0) \notin A$ - from rule $a7$ and supported rule proposition
    ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q > M_0(s) - (\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_0, s \in R(t_i)} M_0(s))$, where $q = W(s, t_j)$ if $s \in \bullet t_j$, or $M_0(s)$ otherwise - from the following:
        A. $\mathit{ptarc}(s, t_i, q, 0)$ represents $q = W(s, t_i)$ if $(s, t_i) \in E^-$ or $q = M_0(s)$ if $s \in R(t_i)$ – from rule $f7, f9$ construction
        B. $\mathit{holds}(s, qq, 0)$ represents $qq = M_0(s)$ – from $i1$ construction
        C. $\mathit{tot\_decr}(s, qqq, 0) \in A$ if $\{\mathit{del}(s, qqq_0, t_0, 0), \dots, \mathit{del}(s, qqq_x, t_x, 0)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition
        D. $\mathit{del}(s, qqq_i, t_i, 0) \in A$ if $\{\mathit{fires}(t_i, 0), \mathit{ptarc}(s, t_i, qqq_i, 0)\} \subseteq A$ – from rule $r7$ and supported rule proposition
        E. $\mathit{del}(s, qqq_i, t_i, 0)$ represents $qqq_i = W(s, t_i) : t_i \in T_0, (s, t_i) \in E^-$, or $qqq_i = M_0(t_i) : t_i \in T_0, s \in R(t_i)$ – from rule $f7, f9$ construction
        F. $\mathit{tot\_decr}(q, qqq, 0)$ represents $\sum_{t_i \in T_0, s \in \bullet t_i} W(s, t_i) + \sum_{t_i \in T_0, s \in R(t_i)} M_0(s)$ – from (C,D,E) above
    iii. Then firing $T_0 \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_0 is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $\mathit{enabled}(t_r, 0) \in A$, $\mathit{fires}(t_r, 0) \in A$, since A is an answer set of Π^2 - from rule $f10$ and supported rule proposition
    i. Then, the firing set T_0 satisfies the reset-transition requirement of definition 21 (firing set)

(e) Then $\{t_0, \dots, t_x\} = T_0$ – using 1(a),1(b),1(d) above; and using 1(c) it is a maximal firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $holds(p, q, 1) \in A$

   (a) Then $\{holds(p, q1, 0), tot\_incr(p, q2, 0), tot\_decr(p, q3, 0)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

   (b) Then $holds(p, q1, 0) \in A$ represents $q1 = M_0(p)$ – given, rule $i1$ construction;

   (c) Then $\{add(p, q2_0, t_0, 0), \ldots, add(p, q2_j, t_j, 0)\} \subseteq A : q2_0 + \cdots + q2_j = q2$ and $\{del(p, q3_0, t_0, 0), \ldots, del(p, q3_l, t_l, 0)\} \subseteq A : q3_0 + \cdots + q3_l = q3$ – rules $r3, r4$ and supported rule proposition, respectively

   (d) Then $\{fires(t_0, 0), \ldots, fires(t_j, 0)\} \subseteq A$ and $\{fires(t_0, 0), \ldots, fires(t_l, 0)\} \subseteq A$ – rules $r6, r7$ and supported rule proposition, respectively

   (e) Then $\{fires(t_0, 0), \ldots, fires(t_j, 0)\} \cup \{fires(t_0, 0), \ldots, fires(t_l, 0)\} \subseteq A = \{fires(t_0, 0), \ldots, fires(t_x, 0)\} \subseteq A$ – set union of subsets

   (f) Then for each $fires(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

   (g) Then $q = M_0(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - (\sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} M_0(p))$ – from (2b) above and the following

      i. Each $add(p, q_j, t_j, 0) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j\bullet$ – rule $r6$ encoding, and definition 22 of transition execution in PN

      ii. Each $del(p, t_y, q_y, 0) \in A$ represents either $q_y = W(p, t_y)$ for $p \in \bullet t_y$, or $q_y = M_0(p)$ for $p \in R(t_y)$ – from rule $r7, f7$ encoding and definition 22 of transition execution in PN; or from rule $r7, f9$ encoding and definition of reset arc in PN

      iii. Each $tot\_incr(p, q2, 0) \in A$ represents $q2 = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

      iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} M_0(p)$ – aggregate assignment atom semantics in rule $r4$

   (h) Then, $M_1(p) = q$ – since $holds(p, q, 1) \in A$ encodes $q = M_1(p)$ – from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set in M_k. Let $\{fires(t_0, k), \ldots, fires(t_x, k)\}$ be the set of all $fires(\ldots, k)$ atoms in A,

   (a) Then for each $fires(t_i, k) \in A$

      i. $enabled(t_i, k) \in A$ – from rule $a1$ and supported rule proposition

      ii. Then $notenabled(t_i, k) \notin A$ – from rule $e2$ and supported rule proposition

      iii. Then either of $body(e3)$, $body(e4)$, or $body(e5)$ must not hold in A – from rule $e3, e4, e5$ and forced atom proposition

iv. Then $q \not< n_i \equiv q \geq n_i$ in $e3$ for all $\{holds(p, q, k), ptarc(p, t_i, n_i, k)\} \subseteq A$ – from $e3$, forced atom proposition, and given facts ($holds(p, q, k) \in A, ptarc(p, t_i, n_i, k) \in A$)

v. And $q \not> n_i \equiv q < n_i$ in $e4$ for all $\{holds(p, q, k), iptarc(p, t_i, n_i, k)\} \subseteq A, n_i = 1; q > n_i \equiv q = 0$ – from $e4$, forced atom proposition, given facts ($holds(p, q, k) \in A, iptarc(p, t_i, 1, k) \in A$), and q is a positive integer

vi. And $q \not< n_i \equiv q \geq n_i$ in $e5$ for all $\{holds(p, q, k), tptarc(p, t_i, n_i, k)\} \subseteq A$ – from $e5$, forced atom proposition, and given facts

vii. Then $(\forall p \in \bullet t_i, M_k(p) \geq W(p, t_i)) \wedge (\forall p \in I(t_i), M_k(p) = 0) \wedge (\forall (p, t_i) \in Q, M_k(p) \geq QW(p, t_i))$ – from

   A. $holds(p, q, k) \in A$ represents $q = M_k(p)$ – inductive assumption, given

   B. $ptarc(p, t_i, n_i, k) \in A$ represents $n_i = W(p, t_i)$ – rule $f7$ construction; or it represents $n_i = M_k(p)$ – rule $f9$ construction; the construction of $f9$ ensures that $notenabled(t, k)$ is never true due to the reset arc

   C. definition 11 of preset $\bullet t_i$ in PN

   D. definition 28 of enabled transition in PN

viii. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 28 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^4 – from rule $a4$ and supported rule proposition

i. Then $\nexists consumesmore(p, k) \in A$ – from rule $a3$ and supported rule proposition

ii. Then $\nexists \{holds(p, q, k), tot\_decr(p, q1, k)\} \subseteq A : q1 > q$ in $body(e3)$ – from $a2$ and forced atom proposition

iii. Then $\nexists p : (\sum_{t_i \in \{t_0, \ldots, t_x\}, p \in \bullet t_i} W(p, t_i) + \sum_{t_i \in \{t_0, \ldots, t_x\}, p \in R(t_i)} M_k(p)) > M_k(p)$ – from the following

   A. $holds(p, q, k)$ represents $q = M_k(p)$ – inductive assumption, given

   B. $tot\_decr(p, q1, k) \in A$ if $\{del(p, q1_0, t_0, k), \ldots, del(p, q1_x, t_x, k)\} \subseteq A$, where $q1 = q1_0 + \cdots + q1_x$ – from $r4$ and forced atom proposition

   C. $del(p, q1_i, t_i, k) \in A$ if $\{fires(t_i, k), ptarc(p, t_i, q1_i, k)\} \subseteq A$ – from $r7$ and supported rule proposition

   D. $del(p, q1_i, t_i, k)$ either represents removal of $q1_i = W(p, t_i)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_i = M_k(p)$ tokens from $p \in R(t_i)$– from rule $r7$, supported rule proposition, and definition 22 of transition execution in PN

iv. Then T_k does not contain conflicting transitions – by the definition 23 of conflicting transitions

(c) And for each $enabled(t_j, k) \in A$ and $fires(t_j, k) \notin A$, $could\_not\_have(t_j, k) \in A$, since A is an answer set of Π^4 - from rule $a6$ and supported rule proposition

i. Then $\{enabled(t_j,k), holds(s,qq,k), ptarc(s,t_j,q,k), tot\_decr(s,qqq,k)\} \subseteq A$, such that $q > qq - qqq$ and $fires(t_j,k) \notin A$ - from rule $a7$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q > M_k(s) - (\sum_{t_i \in T_k, s \in \bullet t_i} W(s,t_i) + \sum_{t_i \in T_k, s \in R(t_i)} M_k(s))$, where $q = W(s,t_j)$ if $s \in \bullet t_j$, or $M_k(s)$ otherwise - from the following:

A. $ptarc(s,t_i,q,k)$ represents $q = W(s,t_i)$ if $(s,t_i) \in E^-$ or $q = M_k(s)$ if $s \in R(t_i)$ – from rule $f7, f9$ construction

B. $holds(s,qq,k)$ represents $qq = M_k(s)$ – construction

C. $tot\_decr(s,qqq,k) \in A$ if $\{del(s,qqq_0,t_0,k), \dots, del(s,qqq_x,t_x,k)\} \subseteq A$ – from rule $r4$ construction and supported rule proposition

D. $del(s,qqq_i,t_i,k) \in A$ if $\{fires(t_i,k), ptarc(s,t_i,qqq_i,k)\} \subseteq A$ – from rule $r7$ and supported rule proposition

E. $del(s,qqq_i,t_i,k)$ represents $qqq_i = W(s,t_i) : t_i \in T_k, (s,t_i) \in E^-$, or $qqq_i = M_k(t_i) : t_i \in T_k, s \in R(t_i)$ – from rule $f7, f9$ construction

F. $tot\_decr(q,qqq,k)$ represents $\sum_{t_i \in T_k, s \in \bullet t_i} W(s,t_i) + \sum_{t_i \in T_k, s \in R(t_i)} M_k(s)$ – from (C,D,E) above

iii. Then firing $T_k \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_k is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $enabled(t_r,k) \in A$, $fires(t_r,k) \in A$, since A is an answer set of Π^2 - from rule $f10$ and supported rule proposition

i. Then the firing set T_k satisfies the reset transition requirement of definition 21 (firing set)

(e) Then $\{t_0,\dots,t_x\} = T_k$ – using 1(a),1(b), 1(d) above; and using 1(c) it is a maximal firing set

2. We show that M_{k+1} is produced by firing T_k in M_k. Let $holds(p,q,k+1) \in A$

(a) Then $\{holds(p,q1,k), tot\_incr(p,q2,k), tot\_decr(p,q3,k)\} \subseteq A : q = q1 + q2 - q3$ – from rule $r5$ and supported rule proposition

(b) Then $holds(p,q1,k) \in A$ represents $q1 = M_k(p)$ – construction, inductive assumption; and $\{add(p,q2_0,t_0,k), \dots, add(p,q2_j,t_j,k)\} \subseteq A : q2_0 + \dots + q2_j = q2$ and $\{del(p,q3_0,t_0,k), \dots, del(p,q3_l,t_l,k)\} \subseteq A : q3_0 + \dots + q3_l = q3$ – rules $r3, r4$ and supported rule proposition, respectively

(c) Then $\{fires(t_0,k), \dots, fires(t_j,k)\} \subseteq A$ and $\{fires(t_0,k), \dots, fires(t_l,k)\} \subseteq A$ – rules $r6, r7$ and supported rule proposition, respectively

(d) Then $\{fires(t_0,k), \dots, fires(t_j,k)\} \cup \{fires(t_0,k), \dots, fires(t_l,k)\} \subseteq A = \{fires(t_0,k), \dots, fires(t_x,k)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x,k) \in A$ we have $t_x \in T_k$ – already shown in item 1 above

(f) Then $q = M_k(p) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} W(t_x, p) - (\sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_k \wedge p \in R(t_x)} M_k(p))$ – from (2b) above and the following

   i. Each $add(p, q_j, t_j, k) \in A$ represents $q_j = W(t_j, p)$ for $p \in t_j \bullet$ – rule $r6$ encoding, and definition 22 of transition execution in PN

   ii. Each $del(p, t_y, q_y, k) \in A$ represents either $q_y = W(p, t_y)$ for $p \in \bullet t_y$, or $q_y = M_k(p)$ for $p \in R(t_y)$ – from rule $r7, f7$ encoding and definition 22 of transition execution in PN; or from rule $r7, f9$ encoding and definition of reset arc in PN

   iii. Each $tot\_incr(p, q2, k) \in A$ represents $q2 = \sum_{t_x \in T_k \wedge p \in t_x \bullet} W(t_x, p)$ – aggregate assignment atom semantics in rule $r3$

   iv. Each $tot\_decr(p, q3, 0) \in A$ represents $q3 = \sum_{t_x \in T_k \wedge p \in \bullet t_x} W(p, t_x) + \sum_{t_x \in T_k \wedge p \in R(t_x)} M_0(p)$ – aggregate assignment atom semantics in rule $r4$

(g) Then, $M_{k+1}(p) = q$ – since $holds(p, q, k+1) \in A$ encodes $q = M_{k+1}(p)$ – from construction

As a result, for any $n > k$, T_n will be a valid firing set for M_n and M_{n+1} will be its target marking.
Conclusion: Since both (I) and (II) hold, $X = M_0, T_0, M_1, \dots, M_k, T_{k+1}$ is an execution sequence of $PN(P, T, E, W, R)$ (w.r.t M_0) iff there is an answer set A of $\Pi^4(PN, M_0, k, ntok)$ such that (A.5) and (A.6) hold.

A.6 Proof of Proposition 6

Let $PN = (P, T, E, C, W, R, I, Q, QW)$ be a Petri Net, M_0 be its initial marking and let $\Pi^5(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.8. Then $X = M_0, T_k, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t. M_0) iff there is an answer set A of $\Pi^5(PN, M_0, k, ntok)$ such that:

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\} \tag{A.11}$$

$$\begin{aligned} \{holds(p, q, c, ts) : p \in P, c/q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ = \{holds(p, q, c, ts) : holds(p, q, c, ts) \in A\} \end{aligned} \tag{A.12}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.11) and (A.12) and show that A is an answer set of Π^5

(II) Given an answer set A of Π^5, we create an execution sequence X such that (A.11) and (A.12) are satisfied.

First we show (I): Given PN and an execution sequence X of PN, we create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$

2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$

3. $A_3 = \{place(p) : p \in P\}$

4. $A_4 = \{trans(t) : t \in T\}$

5. $A_5 = \{color(c) : c \in C\}$

6. $A_6 = \{ptarc(p, t, n_c, c, ts) : (p, t) \in E^-, c \in C, n_c = m_{W(p,t)}(c), n_c > 0, 0 \leq ts \leq k\}$, where $E^- \subseteq E$

7. $A_7 = \{tparc(t, p, n_c, c, ts) : (t, p) \in E^+, c \in C, n_c = m_{W(t,p)}(c), n_c > 0, 0 \leq ts \leq k\}$, where $E^+ \subseteq E$

8. $A_8 = \{holds(p, q_c, c, 0) : p \in P, c \in C, q_c = m_{M_0(p)}(c)\}$

9. $A_9 = \{ptarc(p, t, n_c, c, ts) : p \in R(t), c \in C, n_c = m_{M_{ts}(p)}, n_c > 0, 0 \leq ts \leq k\}$

10. $A_{10} = \{iptarc(p, t, 1, c, ts) : p \in I(t), c \in C, 0 \leq ts < k\}$

11. $A_{11} = \{tptarc(p, t, n_c, c, ts) : (p, t) \in Q, c \in C, n_c = m_{QW(p,t)}(c), n_c > 0, 0 \leq ts \leq k\}$

12. $A_{12} = \{notenabled(t, ts) : t \in T, 0 \leq ts \leq k, \exists c \in C, (\exists p \in \bullet t, m_{M_{ts}(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t), m_{M_{ts}(p)}(c) > 0) \vee (\exists (p, t) \in Q, m_{M_{ts}(p)}(c) < m_{QW(p,t)}(c))\}$
    per definition 32 (enabled transition)

13. $A_{13} = \{enabled(t, ts) : t \in T, 0 \leq ts \leq k, \forall c \in C, (\forall p \in \bullet t, m_{W(p,t)}(c) \leq m_{M_{ts}(p)}(c)) \wedge (\forall p \in I(t), m_{M_{ts}(p)}(c) = 0) \wedge (\forall (p, t) \in Q, m_{M_{ts}(p)}(c) \geq m_{QW(p,t)}(c))\}$
    per definition 32 (enabled transition)

14. $A_{14} = \{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
    per definition 35 (firing set), only an enabled transition may fire

15. $A_{15} = \{add(p, q_c, t, c, ts) : t \in T_{ts}, p \in t\bullet, c \in C, q_c = m_{W(t,p)}(c), 0 \leq ts \leq k\}$
    per definition 33 (transition execution)

16. $A_{16} = \{del(p, q_c, t, c, ts) : t \in T_{ts}, p \in \bullet t, c \in C, q_c = m_{W(p,t)}(c), 0 \leq ts \leq k\} \cup \{del(p, q_c, t, c, ts) : t \in T_{ts}, p \in R(t), c \in C, q_c = m_{M_{ts}(p)}(c), 0 \leq ts \leq k\}$
    per definition 33 (transition execution)

17. $A_{17} = \{tot\_incr(p, q_c, c, ts) : p \in P, c \in C, q_c = \sum_{t \in T_{ts}, p \in t\bullet} m_{W(t,p)}(c), 0 \leq ts \leq k\}$
    per definition 36 (firing set execution)

18. $A_{18} = \{tot\_decr(p, q_c, c, ts) : p \in P, c \in C, q_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} m_{M_{ts}(p)}(c), 0 \leq ts \leq k\}$
    per definition 36 (firing set execution)

19. $A_{19} = \{consumesmore(p, ts) : p \in P, c \in C, q_c = m_{M_{ts}(p)}(c),$
$q1_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} m_{M_{ts}(p)}(c), q1 > q, 0 \leq ts \leq k\}$
per definition 34 (conflicting transitions)

20. $A_{20} = \{consumesmore : \exists p \in P, c \in C, q_c = m_{M_{ts}(p)}(c),$
$q1_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)}(m_{M_{ts}(p)}(c)), q1 > q, 0 \leq ts \leq k\}$
per definition 34 (conflicting transitions) per the maximal firing set semantics

21. $A_{21} = \{holds(p, q_c, c, ts + 1) : p \in P, c \in C, q_c = m_{M_{ts+1}(p)}(c), 0 \leq ts < k\}$,
where $M_{ts+1}(p) = M_{ts}(p) - (\sum_{t \in T_{ts}, p \in \bullet t} W(p,t) +$
$\sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p)) + \sum_{t \in T_{ts}, p \in t \bullet} W(t, p)$
according to definition 36 (firing set execution)

We show that A satisfies (A.11) **and** (A.12)**, and A is an answer set of Π^5.**

A satisfies (A.11) and (A.12) by its construction above. We show A is an answer set of Π^5 by splitting. We split $lit(\Pi^5)$ into a sequence of $7k + 9$ sets:

- $U_0 = head(f1) \cup head(f2) \cup head(f13) \cup head(f5) \cup head(f6) \cup head(i2) = \{place(p) : p \in P\} \cup \{trans(t) : t \in T\} \cup \{col(c) : c \in C\} \cup \{time(0), \dots, time(k)\} \cup \{num(0), \dots, num(ntok)\} \cup \{holds(p, q_c, c, 0) : p \in P, c \in C, q_c = m_{M_0(p)}(c)\}$

- $U_{7k+1} = U_{7k+0} \cup head(f14)^{ts=k} \cup head(f15)^{ts=k} \cup head(f16)^{ts=k} \cup head(f17)^{ts=k} \cup head(f18)^{ts=k} = U_{7k+0} \cup \{ptarc(p, t, n_c, c, k) : (p, t) \in E^-, c \in C, n_c = m_{W(p,t)}(c)\} \cup \{tparc(t, p, n_c, c, k) : (t, p) \in E^+, c \in C, n_c = m_{W(t,p)}(c)\} \cup \{ptarc(p, t, n_c, c, k) : p \in R(t), c \in C, n_c = m_{M_k(p)}(c), n > 0\} \cup \{iptarc(p, t, 1, c, k) : p \in I(t), c \in C\} \cup \{tptarc(p, t, n_c, c, k) : (p, t) \in Q, c \in C, n_c = m_{QW(p,t)}(c)\}$

- $U_{7k+2} = U_{7k+1} \cup head(e6)^{ts=k} \cup head(e7)^{ts=k} \cup head(e8)^{ts=k} = U_{7k+1} \cup \{notenabled(t, k) : t \in T\}$

- $U_{7k+3} = U_{7k+2} \cup head(e9)^{ts=k} = U_{7k+2} \cup \{enabled(t, k) : t \in T\}$

- $U_{7k+4} = U_{7k+3} \cup head(a1)^{ts=k} = U_{7k+3} \cup \{fires(t, k) : t \in T\}$

- $U_{7k+5} = U_{7k+4} \cup head(r8)^{ts=k} \cup head(r9)^{ts=k} = U_{7k+4} \cup \{add(p, q_c, t, c, k) : p \in P, t \in T, c \in C, q_c = m_{W(t,p)}(c)\} \cup \{del(p, q_c, t, c, k) : p \in P, t \in T, c \in C, q_c = m_{W(p,t)}(c)\} \cup \{del(p, q_c, t, c, k) : p \in P, t \in T, c \in C, q_c = m_{M_k(p)}(c)\}$

- $U_{7k+6} = U_{7k+5} \cup head(r10)^{ts=k} \cup head(r11)^{ts=k} = U_{7k+5} \cup \{tot\_incr(p, q_c, c, k) : p \in P, c \in C, 0 \leq q_c \leq ntok\} \cup \{tot\_decr(p, q_c, c, k) : p \in P, c \in C, 0 \leq q_c \leq ntok\}$

- $U_{7k+7} = U_{7k+6} \cup head(a9)^{ts=k} \cup head(r12)^{ts=k} = U_{7k+6} \cup \{consumesmore(p, k) : p \in P\} \cup \{holds(p, q, k + 1) : p \in P, 0 \leq q \leq ntok\}$

- $U_{7k+8} = U_{7k+7} \cup head(a3) = U_{7k+7} \cup \{consumesmore\}$

where $head(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\dots,ts) : a(\dots,ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 7k+8$ form a splitting sequence, since each U_i is a splitting set of Π^5, and $\langle U_\alpha \rangle_{\alpha<\mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{7k+8}$ and $\bigcup_{\alpha<\mu} U_\alpha = lit(\Pi^5)$.

We compute the answer set of Π^5 using the splitting sets as follows:

1. $bot_{U_0}(\Pi^5) = f1 \cup f2 \cup f14 \cup f9 \cup f18 \cup i2$ and $X_0 = A_1 \cup \cdots \cup A_5 \cup A_8$ $(= U_0)$ is its answer set – using forced atom proposition

2. $eval_{U_0}(bot_{U_1}(\Pi^5) \backslash bot_{U_0}(\Pi^5), X_0) = \{ptarc(p,t,q_c,c,0)\text{:-}.|c \in C, q_c = m_{W(p,t)}(c), q_c > 0\} \cup \{tparc(t,p,q_c,c,0)\text{:-}.|c \in C, q_c = m_{W(t,p)}(c), q_c > 0\} \cup \{ptarc(p,t,q_c,c,0)\text{:-}.|c \in C, q_c = m_{M_0(p)}(c), q_c > 0\} \cup \{iptarc(p,t,1,c,0)\text{:-}.|c \in C\} \cup \{tptarc(p,t,q_c,c,0)\text{:-}.|c \in C, q_c = m_{QW(p,t)}(c), q_c > 0\}$. Its answer set $X_1 = A_6^{ts=0} \cup A_7^{ts=0} \cup A_9^{ts=0} \cup A_{10}^{ts=0} \cup A_{11}^{ts=0}$ – using forced atom proposition and construction of $A_6, A_7, A_9, A_{10}, A_{11}$.

3. $eval_{U_1}(bot_{U_2}(\Pi^5) \setminus bot_{U_1}(\Pi^5), X_0 \cup X_1) = \{notenabled(t,0)\text{:-}.|(\{trans(t), ptarc(p,t,n_c,c,0), holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c < n_c)$ or $(\{notenabled(t,0)\text{:-}.|(\{trans(t), iptarc(p,t,n2_c,c,0), holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c \geq n2_c\})$ or $(\{trans(t), tptarc(p,t,n3_c,c,0), holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c < n3_c)\}$. Its answer set $X_2 = A_{12}^{ts=0}$ – using forced atom proposition and construction of A_{12}.

   (a) where, $q_c = m_{M_0(p)}(c)$, and $n_c = m_{W(p,t)}(c)$ for an arc $(p,t) \in E^-$ – by construction of $i2$ and $f14$ in Π^5, and

   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 31 of preset)

   (c) $n2_c = 1$ – by construction of $iptarc$ predicates in Π^5, meaning $q_c \geq n2_c \equiv q_c \geq 1 \equiv q_c > 0$,

   (d) $tptarc(p,t,n3_c,c,0)$ represents $n3_c = m_{QW(p,t)}(c)$, where $(p,t) \in Q$

   (e) thus, $notenabled(t,0) \in X_1$ represents $\exists c \in C, (\exists p \in \bullet t : m_{M_0(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t) : m_{M_0(p)}(c) > 0) \vee (\exists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c))$.

4. $eval_{U_2}(bot_{U_3}(\Pi^5) \setminus bot_{U_2}(\Pi^5), X_0 \cup \cdots \cup X_2) = \{enabled(t,0)\text{:-}.|trans(t) \in X_0 \cup \cdots \cup X_2, notenabled(t,0) \notin X_0 \cup \cdots \cup X_2\}$. Its answer set is $X_3 = A_{13}^{ts=0}$ – using forced atom proposition and construction of A_{13}.

   (a) where, an $enabled(t,0) \in X_3$ if $\nexists$ $notenabled(t,0) \in X_0 \cup \cdots \cup X_2$; which is equivalent to $\forall c \in C, (\nexists p \in \bullet t : m_{M_0(p)}(c) < m_{W(p,t)}(c)) \wedge (\nexists p \in I(t) : m_{M_0(p)}(c) > 0) \wedge (\nexists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c)) \equiv \forall c \in C, (\forall p \in \bullet t : m_{M_0(p)}(c) \geq m_{W(p,t)}(c)) \wedge (\forall p \in I(t) : m_{M_0(p)}(c) = 0)$.

5. $eval_{U_3}(bot_{U_4}(\Pi^5) \setminus bot_{U_3}(\Pi^5), X_0 \cup \cdots \cup X_3) = \{\{fires(t,0)\}\text{:-}.|enabled(t,0)$ holds in $X_0 \cup \cdots \cup X_3\}$. It has multiple answer sets $X_{4.1}, \dots, X_{4.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $eval_{U_3}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^5 determined using splitting is equal to A, we only consider the set that matches

the $fires(t,0)$ elements in A and call it X_4, ignoring the rest. Thus, $X_4 = A_{14}^{ts=0}$, representing T_0.

(a) in addition, for every t such that $enabled(t,0) \in X_0 \cup \dots \cup X_3, R(t) \neq \emptyset$; $fires(t,0) \in X_4$ – per definition 35 (firing set); requiring that a reset transition is fired when enabled

(b) thus, the firing set T_0 will not be eliminated by the constraint $f10$

6. $eval_{U_4}(bot_{U_5}(\Pi^5) \setminus bot_{U_4}(\Pi^5), X_0 \cup \dots \cup X_4) = \{add(p,n_c,t,c,0)\text{:-}.|\{fires(t,0), tparc(t,p,n_c,c,0)\} \subseteq X_0 \cup \dots \cup X_4\} \cup \{del(p,n_c,t,c,0)\text{:-}.|\{fires(t,0), ptarc(p,t,n_c,c,0)\} \subseteq X_0 \cup \dots \cup X_4\}$. It's answer set is $X_5 = A_{15}^{ts=0} \cup A_{16}^{ts=0}$ – using forced atom proposition and definitions of A_{15} and A_{16}.

   (a) where, each add atom is equivalent to $n_c = m_{W(t,p)}(c), c \in C, p \in t\bullet$,

   (b) and each del atom is equivalent to $n_c = m_{W(p,t)}(c), c \in C, p \in \bullet t$; or $n_c = m_{M_0(p)}(c), c \in C, p \in R(t)$,

   (c) representing the effect of transitions in T_0

7. $eval_{U_5}(bot_{U_6}(\Pi^5) \setminus bot_{U_5}(\Pi^5), X_0 \cup \dots \cup X_5) = \{tot\_incr(p,qq_c,c,0)\text{:-}.|qq_c = \sum_{add(p,q_c,t,c,0) \in X_0 \cup \dots \cup X_5} q_c\} \cup \{tot\_decr(p,qq_c,c,0)\text{:-}.|qq_c = \sum_{del(p,q_c,t,c,0) \in X_0 \cup \dots \cup X_5} q_c\}$. It's answer set is $X_6 = A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of A_{17} and A_{18}.

   (a) where, each $tot\_incr(p,qq_c,c,0)$, $qq_c = \sum_{add(p,q_c,t,c,0) \in X_0 \cup \dots X_5} q_c$ $\equiv qq_c = \sum_{t \in X_4, p \in t\bullet} m_{W(p,t)}(c)$,

   (b) and each $tot\_decr(p,qq_c,c,0)$, $qq_c = \sum_{del(p,q_c,t,c,0) \in X_0 \cup \dots X_5} q_c$ $\equiv qq = \sum_{t \in X_4, p \in \bullet t} m_{W(t,p)}(c) + \sum_{t \in X_4, p \in R(t)} m_{M_0(p)}(c)$,

   (c) represent the net effect of transitions in T_0

8. $eval_{U_6}(bot_{U_7}(\Pi^5) \setminus bot_{U_6}(\Pi^5), X_0 \cup \dots \cup X_6) = \{consumesmore(p,0)\text{:-}.| \{holds(p,q_c,c,0), tot\_decr(p,q1_c,c,0)\} \subseteq X_0 \cup \dots \cup X_6, q1_c > q_c\} \cup \{holds(p,q_c,c,1)\text{:-}., |\{holds(p,q1_c,c,0), tot\_incr(p,q2_c,c,0), tot\_decr(p,q3_c,c,0)\} \subseteq X_0 \cup \dots \cup X_6, q_c = q1_c + q2_c - q3_c\}$. It's answer set is $X_7 = A_{19}^{ts=0} \cup A_{21}^{ts=0}$ – using forced atom proposition and definitions of A_{19}, A_{21}.

   (a) where, $consumesmore(p,0)$ represents $\exists p : q_c = m_{M_0(p)}(c), q1_c = \sum_{t \in T_0, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_0, p \in R(t)} m_{M_0(p)}(c), q1_c > q_c, c \in C$, indicating place p will be over consumed if T_0 is fired, as defined in definition 34 (conflicting transitions),

   (b) $holds(p,q_c,c,1)$ represents $q_c = m_{M_1(p)}(c)$ – by construction of Π^5

$\vdots$

9. $eval_{U_{7k+0}}(bot_{U_{7k+1}}(\Pi^5) \backslash bot_{U_{7k+0}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+0}) = \{ptarc(p,t,q_c,c,k)\text{:-.}|c \in C, q_c = mW(p,t)(c), q_c > 0\} \cup \{tparc(t,p,q_c,c,k)\text{:-.}|c \in C, q_c = m_{W(t,p)}(c), q_c > 0\} \cup \{ptarc(p,t,q_c,c,k)\text{:-.}|\, c \in C, q_c = m_{M_k(p)}(c), q_c > 0\} \cup \{iptarc(p,t,1,c,k)\text{:-.}|c \in C\} \cup \{tptarc(p,t,q_c,c,k)\text{:-.}|c \in C, q_c = m_{QW(p,t)}(c), q_c > 0\}$. Its answer set $X_{7k+1} = A_6^{ts=k} \cup A_7^{ts=k} \cup A_9^{ts=k} \cup A_{10}^{ts=k} \cup A_{11}^{ts=k}$ – using forced atom proposition and construction of $A_6, A_7, A_9, A_{10}, A_{11}$.

10. $eval_{U_{7k+1}}(bot_{U_{7k+2}}(\Pi^5) \setminus bot_{U_{7k+1}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+1}) = \{notenabled(t,k)\text{:-.}|$ $(\{trans(t), ptarc(p,t,n_c,c,k), holds(p,q_c,c,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q_c < n_c)$ or $(\{notenabled(t,k)\text{:-.}|(\{trans(t), iptarc(p,t,n2_c,c,k), holds(p,q_c,c,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q_c \geq n2_c\})$ or $(\{trans(t), tptarc(p,t,n3_c,c,k),$ $holds(p,q_c,c,k)\} \subseteq X_0 \cup \cdots \cup X_{7k+1},$ where $q_c < n3_c)\}$. Its answer set $X_{7k+2} = A_{12}^{ts=k}$ – using forced atom proposition and construction of A_{12}.

    (a) since $q_c = m_{M_k(p)}(c)$, and $n_c = m_{W(p,t)}(c)$ for an arc $(p,t) \in E^-$ – by construction of *holds* and *ptarc* predicates in Π^5, and

    (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 31 of preset)

    (c) $n2_c = 1$ – by construction of *iptarc* predicates in Π^5, meaning $q_c \geq n2_c \equiv q_c \geq 1 \equiv q_c > 0$,

    (d) $tptarc(p,t,n3_c,c,k)$ represents $n3_c = m_{QW(p,t)}(c)$, where $(p,t) \in Q$

    (e) thus, $notenabled(t,k) \in X_{8k+1}$ represents $\exists c \in C(\exists p \in \bullet t : m_{M_k(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t) : m_{M_k(p)}(c) > 0) \vee (\exists (p,t) \in Q : m_{M_{ts}(p)}(c) < m_{QW(p,t)}(c))$.

11. $eval_{U_{7k+2}}(bot_{U_{7k+3}}(\Pi^5) \backslash bot_{U_{7k+2}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+2}) = \{enabled(t,k)\text{:-.}|trans(t) \in X_0 \cup \cdots \cup X_{7k+2} \wedge notenabled(t,k) \notin X_0 \cup \cdots \cup X_{7k+2}\}$. Its answer set is $X_{7k+3} = A_{13}^{ts=k}$ – using forced atom proposition and construction of A_{13}.

    (a) since an $enabled(t,k) \in X_{7k+3}$ if $\nexists\ notenabled(t,k) \in X_0 \cup \cdots \cup X_{7k+2}$; which is equivalent to $\forall c \in C, (\nexists p \in \bullet t : m_{M_k(p)}(c) < m_{W(p,t)}(c)) \wedge (\nexists p \in I(t) : m_{M_k(p)}(c) > 0) \wedge (\nexists (p,t) \in Q : m_{M_k(p)}(c) < m_{QW(p,t)}(c)) \equiv \forall c \in C, (\forall p \in \bullet t : m_{M_k(p)}(c) \geq m_{W(p,t)}(c)) \wedge (\forall p \in I(t) : m_{M_k(p)}(c) = 0)$.

12. $eval_{U_{7k+3}}(bot_{U_{7k+4}}(\Pi^5) \setminus bot_{U_{7k+3}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+3}) = \{\{fires(t,k)\}\text{:-.}|$ $enabled(t,k)$ holds in $X_0 \cup \cdots \cup X_{7k+3}\}$. It has multiple answer sets $X_{7k+}4.1, \ldots, X_{7k+}4.n$, corresponding to elements of power set of $fires(t,k)$ atoms in $eval_{U_3}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^5 determined using splitting is equal to A, we only consider the set that matches the $fires(t,k)$ elements in A and call it X_{7k+4}, ignoring the rest. Thus, $X_{7k+4} = A_{14}^{ts=k}$, representing T_k.

    (a) in addition, for every t such that $enabled(t,k) \in X_0 \cup \cdots \cup X_{7k+3}, R(t) \neq \emptyset$; $fires(t,k) \in X_{7k+4}$ – per definition 35 (firing set); requiring that a reset transition is fired when enabled

    (b) thus, the firing set T_k will not be eliminated by the constraint $f10$

13. $eval_{U_{7k+4}}(bot_{U_{7k+5}}(\Pi^5) \setminus bot_{U_{7k+4}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+4}) = \{add(p, n_c, t, c, k){:}\text{-}.| \{fires(t, k),$ $tparc(t, p, n_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+4}\} \cup \{del(p, n_c, t, c, k){:}\text{-}.| \{fires(t, k),$ $ptarc(p, t, n_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+4}\}$. It's answer set is $X_{7k+5} = A_{15}^{ts=k} \cup A_{16}^{ts=k}$ – using forced atom proposition and definitions of A_{15} and A_{16}.

    (a) where each *add* atom is equivalent to $n_c = m_{W(t,p)}(c), c \in C, p \in t\bullet$,

    (b) and each *del* atom is equivalent to $n_c = m_{W(p,t)}(c), c \in C, p \in \bullet t$; or $n_c = m_{M_k(p)}(c), c \in C, p \in R(t)$,

    (c) representing the effect of transitions in T_k

14. $eval_{U_{7k+5}}(bot_{U_{7k+6}}(\Pi^5) \setminus bot_{U_{7k+5}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+5}) = \{tot\_incr(p, qq_c, c, k){:}\text{-}.|$ $qq = \sum_{add(p,q_c,t,c,k) \in X_0 \cup \cdots \cup X_{7k+5}} q_c\} \cup \{tot\_decr(p, qq_c, c, k){:}\text{-}.|$ $qq_c = \sum_{del(p,q_c,t,c,k) \in X_0 \cup \cdots \cup X_{7k+5}} q_c\}$. It's answer set is $X_{7k+6} = A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition and definitions of A_{17} and A_{18}.

    (a) where, each $tot\_incr(p, qq_c, c, k)$, $qq_c = \sum_{add(p,q_c,t,c,k) \in X_0 \cup \ldots X_{8k+5}} q_c$ $\equiv qq_c = \sum_{t \in X_{7k+4}, p \in t\bullet} m_{W(p,t)}(c)$,

    (b) and each $tot\_decr(p, qq_c, c, k)$, $qq_c = \sum_{del(p,q_c,t,c,k) \in X_0 \cup \ldots X_{8k+5}} q_c$ $\equiv qq = \sum_{t \in X_{7k+4}, p \in \bullet t} m_{W(t,p)}(c) + \sum_{t \in X_{7k+4}, p \in R(t)} m_{M_k(p)}(c)$,

    (c) represent the net effect of transitions in T_k

15. $eval_{U_{7k+6}}(bot_{U_{7k+7}}(\Pi^5) \setminus bot_{U_{7k+6}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+6}) = \{consumesmore(p, k){:}\text{-}.|$ $\{holds(p, q_c, c, k), tot\_decr(p, q1_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+6}, q1 > q\} \cup$ $\{holds(p, q_c, c, k+1){:}\text{-}.|\{holds(p, q1_c, c, k), tot\_incr(p, q2_c, c, k),$ $tot\_decr(p, q3_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{7k+6}, q_c = q1_c + q2_c - q3_c\}$. It's answer set is $X_{7k+7} = A_{19}^{ts=k} \cup A_{21}^{ts=k}$ – using forced atom proposition and definitions of A_{19}, A_{21}.

    (a) where, $consumesmore(p, k)$ represents $\exists p : q_c = m_{M_k(p)}(c), q1_c = \sum_{t \in T_k, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_k, p \in R(t)} m_{M_k(p)}(c), q1_c > q_c, c \in C$, indicating place p that will be over consumed if T_k is fired, as defined in definition 34 (conflicting transitions),

    (b) and $holds(p, q_c, c, k+1)$ represents $q_c = m_{M_{k+1}(p)}(c)$ – by construction of Π^5

16. $eval_{U_{7k+7}}(bot_{U_{7k+8}}(\Pi^5) \setminus bot_{U_{7k+7}}(\Pi^5), X_0 \cup \cdots \cup X_{7k+7}) = \{consumesmore{:}\text{-}.|$ $\{consumesmore(p, 0), \ldots, consumesmore(p, k)\} \cap (X_0 \cup \cdots \cup X_{7k+7}) \neq \emptyset\}$. It's answer set is $X_{7k+8} = A_{20}^{ts=k}$ – using forced atom proposition and the definition of A_{20}

    (a) X_{7k+7} will be empty since none of $consumesmore(p, 0), \ldots,$ $consumesmore(p, k)$ hold in $X_0 \cup \cdots \cup X_{7k+7}$ due to the construction of A, encoding of $a9$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \cdots \cup X_{7k+8}$ is the answer set of Π^5 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 7k+8$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^5.

Next we show (II): Given Π^5 be the encoding of a Petri Net $PN(P,T,E,C,W,R,I,Q,WQ)$ with initial marking M_0, and A be an answer set of Π^5 that satisfies (A.11) and (A.12), then we can construct $X = M_0, T_k, \ldots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \ldots, M_i(p_n))$, where $\{holds(p_0, m_{M_i(p_0)}(c), c, i), \ldots holds(p_n, m_{M_i(p_n)}(c), c, i)\} \subseteq A$, for $c \in C, 0 \leq i \leq k+1$
2. $T_i = \{t : fires(t, i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) firing T_0 in M_0 results in marking M_1.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0, 0), \ldots, fires(t_x, 0)\}$ be the set of all $fires(\ldots, 0)$ atoms in A,
   (a) Then for each $fires(t_i, 0) \in A$
      i. $enabled(t_i, 0) \in A$ – from rule $a1$ and supported rule proposition
      ii. Then $notenabled(t_i, 0) \notin A$ – from rule $e9$ and supported rule proposition
      iii. Then either of $body(e6)$, $body(e7)$, or $body(e8)$ must not hold in A – from rules $e6, e7, e8$ and forced atom proposition
      iv. Then $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e6$ for all $\{holds(p, q_c, c, 0), ptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A$ – from $e6$, forced atom proposition, and given facts $(holds(p, q_c, c, 0) \in A, ptarc(p, t_i, n_{ic}, 0) \in A)$
      v. And $q_c \not\geq n_{ic} \equiv q_c < n_{ic}$ in $e7$ for all $\{holds(p, q_c, c, 0), iptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A, n_{ic} = 1$; $q_c > n_{ic} \equiv q_c = 0$ – from $e7$, forced atom proposition, given facts $(holds(p, q_c, c, 0) \in A, iptarc(p, t_i, 1, c, 0) \in A)$, and q_c is a positive integer
      vi. And $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e8$ for all $\{holds(p, q_c, c, 0), tptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A$ – from $e8$, forced atom proposition, and given facts
      vii. Then $\forall c \in C, (\forall p \in \bullet t_i, m_{M_0(p)}(c) \geq m_{W(p,t_i)}(c)) \wedge (\forall p \in I(t_i), m_{M_0(p)}(c) = 0) \wedge (\forall (p, t_i) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t_i)}(c))$ – from the following
         A. $holds(p, q_c, c, 0) \in A$ represents $q_c = m_{M_0(p)}(c)$ – rule $i2$ construction

B. $ptarc(p, t_i, n_{ic}, 0) \in A$ represents $n_{ic} = m_{W(p,t_i)}(c)$ – rule $f14$ construction; or it represents $n_{ic} = m_{M_0(p)}(c)$ – rule $f16$ construction; the construction of $f16$ ensures that $notenabled(t, 0)$ is never true for a reset arc

C. definition 31 of preset $\bullet t_i$ in PN

D. definition 32 of enabled transition in PN

viii. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 32 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^5 – from rule $a3$ and supported rule proposition

i. Then $\nexists consumesmore(p, 0) \in A$ – from rule $a9$ and supported rule proposition

ii. Then $\nexists \{holds(p, q_c, c, 0), tot\_decr(p, q1_c, c, 0)\} \subseteq A, q1_c > q_c$ in $body(a9)$ – from $a9$ and forced atom proposition

iii. Then $\nexists c \in C \nexists p \in P, (\sum_{t_i \in \{t_0, \dots, t_x\}, p \in \bullet t_i} m_{W(p,t_i)}(c) + \sum_{t_i \in \{t_0, \dots, t_x\}, p \in R(t_i)} m_{M_0(p)}(c)) > m_{M_0(p)}(c)$ – from the following

A. $holds(p, q_c, c, 0)$ represents $q_c = m_{M_0(p)}(c)$ – from rule $i2$ construction, given

B. $tot\_decr(p, q1_c, c, 0) \in A$ if $\{del(p, q1_{0_c}, t_0, c, 0), \dots, del(p, q1_{x_c}, t_x, c, 0)\} \subseteq A$, where $q1_c = q1_{0_c} + \dots + q1_{x_c}$ – from $r11$ and forced atom proposition

C. $del(p, q1_{i_c}, t_i, c, 0) \in A$ if $\{fires(t_i, 0), ptarc(p, t_i, q1_{i_c}, c, 0)\} \subseteq A$ – from $r9$ and supported rule proposition

D. $del(p, q1_{i_c}, t_i, c, 0)$ either represents removal of $q1_{i_c} = m_{W(p,t_i)}(c)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_{i_c} = m_{M_0(p)}(c)$ tokens from $p \in R(t_i)$– from rules $r9, f14, f16$, supported rule proposition, and definition 33 of transition execution in PN

iv. Then the set of transitions in T_0 do not conflict – by the definition 34 of conflicting transitions

(c) And for each reset transition t_r with $enabled(t_r, 0) \in A$, $fires(t_r, 0) \in A$, since A is an answer set of Π^5 - from rule $f10$ and supported rule proposition

i. Then, the firing set T_0 satisfies the reset-transition requirement of definition 35 (firing set)

(d) Then $\{t_0, \dots, t_x\} = T_0$ – using 1(a),1(b),1(d) above; and using 1(c) it is a maximal firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $holds(p, q_c, c, 1) \in A$

(a) Then $\{holds(p, q1_c, c, 0), tot\_incr(p, q2_c, c, 0), tot\_decr(p, q3_c, c, 0)\} \subseteq A$: $q_c = q1_c + q2_c - q3_c$ – from rule $r12$ and supported rule proposition

(b) Then, $holds(p, q1_c, c, 0) \in A$ represents $q1_c = m_{M_0(p)}(c)$ – given, rule $i2$ construction; and $\{add(p, q2_{0_c}, t_0, c, 0), \dots, add(p, q2_{j_c}, t_j, c, 0)\} \subseteq A$:

$q2_{0_c} + \cdots + q2_{j_c} = q2_c$ and $\{del(p, q3_{0_c}, t_0, c, 0), \ldots, del(p, q3_{l_c}, t_l, c, 0)\} \subseteq A : q3_{0_c} + \cdots + q3_{l_c} = q3_c$ – rules $r10, r11$ and supported rule proposition, respectively

(c) Then $\{fires(t_0, 0), \ldots, fires(t_j, 0)\} \subseteq A$ and $\{fires(t_0, 0), \ldots, fires(t_l, 0)\} \subseteq A$ – rules $r8, r9$ and supported rule proposition, respectively

(d) Then $\{fires(t_0, 0), \ldots, fires(t_j, 0)\} \cup \{fires(t_0, 0), \ldots, fires(t_l, 0)\} \subseteq A = \{fires(t_0, 0), \ldots, fires(t_x, 0)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

(f) Then $q_c = m_{M_0(p)}(c) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} m_{W(t_x,p)}(c) - (\sum_{t_x \in T_0 \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} m_{M_0(p)}(c))$ – from (2b) above and the following

   i. Each $add(p, q_{j_c}, t_j, c, 0) \in A$ represents $q_{j_c} = m_{W(t_j,p)}(c)$ for $p \in t_j \bullet$ – rule $r8, f15$ encoding, and definition 33 of transition execution in PN

   ii. Each $del(p, t_y, q_{y_c}, c, 0) \in A$ represents either $q_{y_c} = m_{W(p,t_y)}(c)$ for $p \in \bullet t_y$, or $q_{y_c} = m_{M_0(p)}(c)$ for $p \in R(t_y)$ – from rule $r9, f14$ encoding and definition 33 of transition execution in PN; or from rule $r9, f16$ encoding and definition of reset arc in PN

   iii. Each $tot\_incr(p, q2_c, c, 0) \in A$ represents $q2_c = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} m_{W(t_x,p)}(c)$ – aggregate assignment atom semantics in rule $r10$

   iv. Each $tot\_decr(p, q3_c, c, 0) \in A$ represents $q3_c = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} m_{M_0(p)}(c)$ – aggregate assignment atom semantics in rule $r11$

(g) Then, $m_{M_1(p)}(c) = q_c$ – since $holds(p, q_c, c, 1) \in A$ encodes $q_c = m_{M_1(p)}(c)$ – from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set in M_k. Let $\{fires(t_0, k), \ldots, fires(t_x, k)\}$ be the set of all $fires(\ldots, k)$ atoms in A,

   (a) Then for each $fires(t_i, k) \in A$

      i. $enabled(t_i, k) \in A$ – from rule $a1$ and supported rule proposition

      ii. Then $notenabled(t_i, k) \notin A$ – from rule $e9$ and supported rule proposition

      iii. Then either of $body(e6)$, $body(e7)$, or $body(e8)$ must not hold in A – from rules $e6, e7, e8$ and forced atom proposition

      iv. Then $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e6$ for all $\{holds(p, q_c, c, k), ptarc(p, t_i, n_{ic}, c, k)\} \subseteq A$ – from $e6$, forced atom proposition, given facts, and the inductive assumption

      v. And $q_c \not\geq n_{ic} \equiv q_c < n_{ic}$ in $e7$ for all $\{holds(p, q_c, c, k), iptarc(p, t_i, n_{ic}, c, k)\} \subseteq A, n_{ic} = 1$; $q_c > n_{ic} \equiv q_c = 0$ – from $e7$, forced atom proposition, given facts ($holds(p, q_c, c, k) \in A, iptarc(p, t_i, 1, c, k) \in A$), and q_c is a positive integer

vi. And $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e8$ for all $\{holds(p, q_c, c, k),$ $tptarc(p, t_i, n_{ic}, c, k)\} \subseteq A$ – from $e8$, forced atom proposition, and inductive assumption

vii. Then $(\forall p \in \bullet t_i, m_{M_k(p)}(c) \geq m_{W(p,t_i)}(c)) \wedge (\forall p \in I(t_i), m_{M_k(p)}(c) = 0) \wedge (\forall (p, t_i) \in Q, m_{M_k(p)}(c) \geq m_{QW(p,t_i)}(c))$ – from

A. $holds(p, q_c, c, k) \in A$ represents $q_c = m_{M_k(p)}(c)$ – construction of Π^5

B. $ptarc(p, t_i, n_{ic}, k) \in A$ represents $n_{ic} = m_{W(p,t_i)}(c)$ – rule $f14$ construction; or it represents $n_{ic} = m_{M_k(p)}(c)$ – rule $f16$ construction; the construction of $f16$ ensures that $notenabled(t, k)$ is never true for a reset arc

C. definition 31 of preset $\bullet t_i$ in PN

D. definition 32 of enabled transition in PN

viii. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 32 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^5 – from rule $a3$ and supported rule proposition

i. Then $\nexists consumesmore(p, k) \in A$ – from rule $a9$ and supported rule proposition

ii. Then $\nexists\{holds(p, q_c, c, k), tot\_decr(p, q1_c, c, k)\} \subseteq A, q1_c > q_c$ in $body(a9)$ – from $a9$ and forced atom proposition

iii. Then $\nexists c \in C, \nexists p \in P, (\sum_{t_i \in \{t_0,\ldots,t_x\}, p \in \bullet t_i} m_{W(p,t_i)}(c) + \sum_{t_i \in \{t_0,\ldots,t_x\}, p \in R(t_i)} m_{M_k(p)}(c)) > m_{M_k(p)}(c)$ – from the following

A. $holds(p, q_c, c, k)$ represents $q_c = m_{M_k(p)}(c)$ – construction of Π^5, inductive assumption

B. $tot\_decr(p, q1_c, c, k) \in A$ if $\{del(p, q1_{0c}, t_0, c, k), \ldots,$ $del(p, q1_{xc}, t_x, c, k)\} \subseteq A$, where $q1_c = q1_{0c} + \cdots + q1_{xc}$ – from $r11$ and forced atom proposition

C. $del(p, q1_{ic}, t_i, c, k) \in A$ if $\{fires(t_i, k), ptarc(p, t_i, q1_{ic}, c, k)\} \subseteq A$ – from $r9$ and supported rule proposition

D. $del(p, q1_{ic}, t_i, c, k)$ either represents removal of $q1_{ic} = m_{W(p,t_i)}(c)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_{ic} = m_{M_k(p)}(c)$ tokens from $p \in R(t_i)$– from rules $r9, f14, f16$, supported rule proposition, and definition 33 of transition execution in PN

iv. Then the set of transitions in T_k do not conflict – by the definition 34 of conflicting transitions

(c) And for each reset transition t_r with $enabled(t_r, k) \in A$, $fires(t_r, k) \in A$, since A is an answer set of Π^5 - from rule $f10$ and supported rule proposition

i. Then the firing set T_k satisfies the reset transition requirement of definition 35 (firing set)

(d) Then $\{t_0, \ldots, t_x\} = T_k$ – using 1(a),1(b), 1(d) above; and using 1(c) it is a maximal firing set

2. We show that M_{k+1} is produced by firing T_k in M_k. Let $holds(p, q_c, c, k+1) \in A$
   (a) Then $\{holds(p, q1_c, c, k), tot\_incr(p, q2_c, c, k), tot\_decr(p, q3_c, c, k)\} \subseteq A : q_c = q1_c + q2_c - q3_c$ – from rule $r12$ and supported rule proposition
   (b) Then, $holds(p, q1_c, c, k) \in A$ represents $q1_c = m_{M_k(p)}(c)$ – construction, inductive assumption; and $\{add(p, q2_{0_c}, t_0, c, k), \dots, add(p, q2_{j_c}, t_j, c, k)\} \subseteq A : q2_{0_c} + \dots + q2_{j_c} = q2_c$ and $\{del(p, q3_{0_c}, t_0, c, k), \dots, del(p, q3_{l_c}, t_l, c, k)\} \subseteq A : q3_{0_c} + \dots + q3_{l_c} = q3_c$ – rules $r10, r11$ and supported rule proposition, respectively
   (c) Then $\{fires(t_0, k), \dots, fires(t_j, k)\} \subseteq A$ and $\{fires(t_0, k), \dots, fires(t_l, k)\} \subseteq A$ – rules $r8, r9$ and supported rule proposition, respectively
   (d) Then $\{fires(t_0, k), \dots, fires(t_j, k)\} \cup \{fires(t_0, k), \dots, fires(t_l, k)\} \subseteq A = \{fires(t_0, k), \dots, fires(t_x, k)\} \subseteq A$ – set union of subsets
   (e) Then for each $fires(t_x, k) \in A$ we have $t_x \in T_k$ – already shown in item 1 above
   (f) Then $q_c = m_{M_k(p)}(c) + \sum_{t_x \in T_k \wedge p \in t_x \bullet} m_{W(t_x,p)}(c) - (\sum_{t_x \in T_k \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_k \wedge p \in R(t_x)} m_{M_k(p)}(c))$ – from (2b) above and the following
      i. Each $add(p, q_{j_c}, t_j, c, k) \in A$ represents $q_{j_c} = m_{W(t_j,p)}(c)$ for $p \in t_j \bullet$ – rule $r8, f15$ encoding, and definition 33 of transition execution in PN
      ii. Each $del(p, t_y, q_{y_c}, c, k) \in A$ represents either $q_{y_c} = m_{W(p,t_y)}(c)$ for $p \in \bullet t_y$, or $q_{y_c} = m_{M_k(p)}(c)$ for $p \in R(t_y)$ – from rule $r9, f14$ encoding and definition 33 of transition execution in PN; or from rule $r9, f16$ encoding and definition of reset arc in PN
      iii. Each $tot\_incr(p, q2_c, c, k) \in A$ represents $q2_c = \sum_{t_x \in T_k \wedge p \in t_x \bullet} m_{W(t_x,p)}(c)$ – aggregate assignment atom semantics in rule $r10$
      iv. Each $tot\_decr(p, q3_c, c, k) \in A$ represents $q3_c = \sum_{t_x \in T_k \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_k \wedge p \in R(t_x)} m_{M_k(p)}(c)$ – aggregate assignment atom semantics in rule $r11$
   (g) Then, $m_{M_{k+1}(p)}(c) = q_c$ – since $holds(p, q_c, c, k+1) \in A$ encodes $q_c = m_{M_{k+1}(p)}(c)$ – from construction

As a result, for any $n > k$, T_n will be a valid firing set for M_n and M_{n+1} will be its target marking.
Conclusion: Since both (I) and (II) hold, $X = M_0, T_k, M_1, \dots, M_k, T_{k+1}$ is an execution sequence of $PN(P, T, E, C, W, R, I, Q, QW)$ (w.r.t M_0) iff there is an answer set A of $\Pi^5(PN, M_0, k, ntok)$ such that (A.11) and (A.12) hold.

A.7 Proof of Proposition 7

Let $PN = (P, T, E, C, W, R, I, Q, QW, Z)$ be a Petri Net, M_0 be its initial marking and let $\Pi^6(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.10. Then $X = M_0, T_k, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t. M_0) iff there is an answer set A of $\Pi^6(PN, M_0, k, ntok)$ such that:

$$\{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t, ts) : fires(t, ts) \in A\} \tag{A.13}$$

$$\begin{aligned}\{holds(p,q,c,ts) : p \in P, c/q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ = \{holds(p,q,c,ts) : holds(p,q,c,ts) \in A\}\end{aligned} \tag{A.14}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.13) and (A.14) and show that A is an answer set of Π^6

(II) Given an answer set A of Π^6, we create an execution sequence X such that (A.13) and (A.14) are satisfied.

First we show (I): We create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$
2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$
3. $A_3 = \{place(p) : p \in P\}$
4. $A_4 = \{trans(t) : t \in T\}$
5. $A_5 = \{color(c) : c \in C\}$
6. $A_6 = \{ptarc(p,t,n_c,c,ts) : (p,t) \in E^-, c \in C, n_c = m_{W(p,t)}(c), n_c > 0, 0 \leq ts \leq k\}$, where $E^- \subseteq E$
7. $A_7 = \{tparc(t,p,n_c,c,ts) : (t,p) \in E^+, c \in C, n_c = m_{W(t,p)}(c), n_c > 0, 0 \leq ts \leq k\}$, where $E^+ \subseteq E$
8. $A_8 = \{holds(p,q_c,c,0) : p \in P, c \in C, q_c = m_{M_0(p)}(c)\}$
9. $A_9 = \{ptarc(p,t,n_c,c,ts) : p \in R(t), c \in C, n_c = m_{M_{ts}(p)}, n_c > 0, 0 \leq ts \leq k\}$
10. $A_{10} = \{iptarc(p,t,1,c,ts) : p \in I(t), c \in C, 0 \leq ts < k\}$
11. $A_{11} = \{tptarc(p,t,n_c,c,ts) : (p,t) \in Q, c \in C, n_c = m_{QW(p,t)}(c), n_c > 0, 0 \leq ts \leq k\}$
12. $A_{12} = \{notenabled(t,ts) : t \in T, 0 \leq ts \leq k, \exists c \in C, (\exists p \in \bullet t, m_{M_{ts}(p)}(c) < m_{W(p,t)}(c)) \lor (\exists p \in I(t), m_{M_{ts}(p)}(c) > 0) \lor (\exists (p,t) \in Q, m_{M_{ts}(p)}(c) < m_{QW(p,t)}(c))\}$
per definition 40 (enabled transition)
13. $A_{13} = \{enabled(t,ts) : t \in T, 0 \leq ts \leq k, \forall c \in C, (\forall p \in \bullet t, m_{W(p,t)}(c) \leq m_{M_{ts}(p)}(c)) \land (\forall p \in I(t), m_{M_{ts}(p)}(c) = 0) \land (\forall (p,t) \in Q, m_{M_{ts}(p)}(c) \geq m_{QW(p,t)}(c))\}$
per definition 40 (enabled transition)
14. $A_{14} = \{fires(t,ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
per definition 41 (firing set), only an enabled transition may fire
15. $A_{15} = \{add(p,q_c,t,c,ts) : t \in T_{ts}, p \in t\bullet, c \in C, q_c = m_{W(t,p)}(c), 0 \leq ts \leq k\}$
per definition 33 (transition execution)

16. $A_{16} = \{\mathit{del}(p, q_c, t, c, ts) : t \in T_{ts}, p \in \bullet t, c \in C, q_c = m_{W(p,t)}(c), 0 \leq ts \leq k\} \cup \{\mathit{del}(p, q_c, t, c, ts) : t \in T_{ts}, p \in R(t), c \in C, q_c = m_{M_{ts}(p)}(c), 0 \leq ts \leq k\}$
    per definition 33 (transition execution)

17. $A_{17} = \{\mathit{tot\_incr}(p, q_c, c, ts) : p \in P, c \in C, q_c = \sum_{t \in T_{ts}, p \in t\bullet} m_{W(t,p)}(c), 0 \leq ts \leq k\}$
    per definition 36 (firing set execution)

18. $A_{18} = \{\mathit{tot\_decr}(p, q_c, c, ts) : p \in P, c \in C, q_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} m_{M_{ts}(p)}(c), 0 \leq ts \leq k\}$
    per definition 36 (firing set execution)

19. $A_{19} = \{\mathit{consumesmore}(p, ts) : p \in P, c \in C, q_c = m_{M_{ts}(p)}(c),$
    $q1_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} m_{M_{ts}(p)}(c), q1_c > q_c, 0 \leq ts \leq k\}$
    per definition 34 (conflicting transitions)

20. $A_{20} = \{\mathit{consumesmore} : \exists p \in P, c \in C, q_c = m_{M_{ts}(p)}(c),$
    $q1_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} (m_{M_{ts}(p)}(c)), q1_c > q_c, 0 \leq ts \leq k\}$
    per definition 34 (conflicting transitions)

21. $A_{21} = \{\mathit{holds}(p, q_c, c, ts + 1) : p \in P, c \in C, q_c = m_{M_{ts+1}(p)}(c), 0 \leq ts < k\}$,
    where $M_{ts+1}(p) = M_j(p) - (\sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p)) + \sum_{t \in T_{ts}, p \in t\bullet} W(t,p)$
    according to definition 36 (firing set execution)

22. $A_{22} = \{\mathit{transpr}(t, pr) : t \in T, pr = Z(t)\}$

23. $A_{23} = \{\mathit{notprenabled}(t, ts) : t \in T, \mathit{enabled}(t, ts) \in A_{13}, (\exists tt \in T, \mathit{enabled}(tt, ts) \in A_{13}, Z(tt) < Z(t)), 0 \leq ts \leq k\}$
    $= \{\mathit{notprenabled}(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), (\forall p \in I(t), M_{ts}(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq WQ(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)),$
    $(\forall pp \in I(tt), M_{ts}(pp) = 0), (\forall (pp, tt) \in Q, M_{ts}(pp) \geq QW(pp, tt)), Z(tt) < Z(t), 0 \leq ts \leq k\}$

24. $A_{24} = \{\mathit{prenabled}(t, ts) : t \in T, \mathit{enabled}(t, ts) \in A_{13}, (\nexists tt \in T : \mathit{enabled}(tt, ts) \in A_{13}, Z(tt) < Z(t)), 0 \leq ts \leq k\} = \{\mathit{prenabled}(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), (\forall p \in I(t), M_{ts}(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq WQ(p,t)), \nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_{ts}(pp) = 0), (\forall (pp, tt) \in Q, M_{ts}(pp) \geq QW(pp, tt)), Z(tt) < Z(t), 0 \leq ts \leq k\}$

25. $A_{25} = \{\mathit{could\_not\_have}(t, ts) : t \in T, \mathit{prenabled}(t, ts) \in A_{24}, \mathit{fires}(t, ts) \notin A_{14}, (\exists p \in \bullet t \cup R(t) : q > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p, t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), q = W(p,t)$ if $(p,t) \in E^-$ or $R(t)$ otherwise$), 0 \leq ts \leq k\} = \{\mathit{could_not_have}(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), (\forall p \in I(t), M_{ts}(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), (\nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_{ts}(pp) = 0), (\forall (pp, tt) \in Q, M_{ts}(pp) \geq QW(pp, tt)), Z(tt) < Z(t)), t \notin T_{ts}, (\exists p \in \bullet t \cup R(t) : q > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p, t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), q = W(p,t)$ if $(p,t) \in E^-$ or $R(t)$ otherwise$), 0 \leq ts \leq k\}$
    per the maximal firing set semantics

We show that A satisfies (A.13) and (A.14), and A is an answer set of Π^6.

A satisfies (A.13) and (A.14) by its construction above. We show A is an answer set of Π^6 by splitting. We split $lit(\Pi^6)$ into a sequence of $9k + 11$ sets:

- $U_0 = head(f1) \cup head(f2) \cup head(f13) \cup head(f5) \cup head(f6) \cup head(i2) \cup head(f19) = \{place(p) : p \in P\} \cup \{trans(t) : t \in T\} \cup \{col(c) : c \in C\} \cup \{time(0), \dots, time(k)\} \cup \{num(0), \dots, num(ntok)\} \cup \{holds(p, q_c, c, 0) : p \in P, c \in C, q_c = m_{M_0(p)}(c)\} \cup \{transpr(t, pr) : t \in T, pr = Z(t)\}$

- $U_{9k+1} = U_{9k+0} \cup head(f14)^{ts=k} \cup head(f15)^{ts=k} \cup head(f16)^{ts=k} \cup head(f17)^{ts=k} \cup head(f18)^{ts=k} = U_{10k+0} \cup \{ptarc(p, t, n_c, c, k) : (p, t) \in E^-, c \in C, n_c = m_{W(p,t)}(c)\}$ $\cup \{tparc(t, p, n_c, c, k) : (t, p) \in E^+, c \in C, n_c = m_{W(t,p)}(c)\} \cup \{ptarc(p, t, n_c, c, k) : p \in R(t), c \in C, n_c = m_{M_k(p)}(c), n > 0\} \cup \{iptarc(p, t, 1, c, k) : p \in I(t), c \in C\} \cup \{tptarc(p, t, n_c, c, k) : (p, t) \in Q, c \in C, n_c = m_{QW(p,t)}(c)\}$

- $U_{9k+2} = U_{9k+1} \cup head(e6)^{ts=k} \cup head(e7)^{ts=k} \cup head(e8)^{ts=k} = U_{9k+1}$ $\cup \{notenabled(t, k) : t \in T\}$

- $U_{9k+3} = U_{9k+2} \cup head(e9)^{ts=k} = U_{9k+2} \cup \{enabled(t, k) : t \in T\}$

- $U_{9k+4} = U_{9k+3} \cup head(a11)^{ts=k} = U_{9k+3} \cup \{notprenabled(t, k) : t \in T\}$

- $U_{9k+5} = U_{9k+4} \cup head(a12)^{ts=k} = U_{9k+4} \cup \{prenabled(t, k) : t \in T\}$

- $U_{9k+6} = U_{9k+5} \cup head(a13)^{ts=k} = U_{9k+5} \cup \{fires(t, k) : t \in T\}$

- $U_{9k+7} = U_{9k+6} \cup head(r8)^{ts=k} \cup head(r9)^{ts=k} = U_{9k+6} \cup \{add(p, q_c, t, c, k) : p \in P, t \in T, c \in C, q_c = m_{W(t,p)}(c)\} \cup \{del(p, q_c, t, c, k) : p \in P, t \in T, c \in C, q_c = m_{W(p,t)}(c)\} \cup \{del(p, q_c, t, c, k) : p \in P, t \in T, c \in C, q_c = m_{M_k(p)}(c)\}$

- $U_{9k+8} = U_{9k+7} \cup head(r10)^{ts=k} \cup head(r11)^{ts=k} = U_{9k+7} \cup \{tot\_incr(p, q_c, c, k) : p \in P, c \in C, 0 \leq q_c \leq ntok\} \cup \{tot\_decr(p, q_c, c, k) : p \in P, c \in C, 0 \leq q_c \leq ntok\}$

- $U_{9k+9} = U_{9k+8} \cup head(a9)^{ts=k} \cup head(r12)^{ts=k} \cup head(a14)^{ts=k} = U_{9k+8} \cup \{consumesmore(p, k) : p \in P\} \cup \{holds(p, q, k + 1) : p \in P, 0 \leq q \leq ntok\} \cup \{could\_not\_have(t, k) : t \in T\}$

- $U_{9k+10} = U_{9k+9} \cup head(a3)^{ts=k} = U_{9k+9} \cup \{consumesmore\}$

where $head(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\dots, ts) : a(\dots, ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 9k + 10$ form a splitting sequence, since each U_i is a splitting set of Π^6, and $\langle U_\alpha \rangle_{\alpha < \mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{9k+10}$ and $\bigcup_{\alpha < \mu} U_\alpha = lit(\Pi^6)$.

We compute the answer set of Π^6 using the splitting sets as follows:

1. $bot_{U_0}(\Pi^6) = f1 \cup f2 \cup f13 \cup f5 \cup f6 \cup i2 \cup f19$ and $X_0 = A_1 \cup \cdots \cup A_5 \cup A_8$ $(= U_0)$ is its answer set – using forced atom proposition

2. $eval_{U_0}(bot_{U_1}(\Pi^6) \backslash bot_{U_0}(\Pi^6), X_0) = \{ptarc(p,t,q_c,c,0){:}\text{-}.|c \in C, q_c = m_{W(p,t)}(c), q_c > 0\} \cup \{tparc(t,p,q_c,c,0){:}\text{-}.|c \in C, q_c = m_{W(t,p)}(c), q_c > 0\} \cup \{ptarc(p,t,q_c,c,0){:}\text{-}.|c \in C, q_c = m_{M_0(p)}(c), q_c > 0\} \cup \{iptarc(p,t,1,c,0){:}\text{-}.|c \in C\} \cup \{tptarc(p,t,q_c,c,0){:}\text{-}.|c \in C, q_c = m_{QW(p,t)}(c), q_c > 0\}$. Its answer set $X_1 = A_6^{ts=0} \cup A_7^{ts=0} \cup A_9^{ts=0} \cup A_{10}^{ts=0} \cup A_{11}^{ts=0}$ – using forced atom proposition and construction of $A_6, A_7, A_9, A_{10}, A_{11}$.

3. $eval_{U_1}(bot_{U_2}(\Pi^6) \setminus bot_{U_1}(\Pi^6), X_0 \cup X_1) = \{notenabled(t,0){:}\text{-}.|(\{trans(t), ptarc(p,t,n_c,c,0), holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c < n_c)$ or $(\{notenabled(t,0){:}\text{-}.|(\{trans(t), iptarc(p,t,n2_c,c,0), holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c \geq n2_c\})$ or $(\{trans(t), tptarc(p,t,n3_c,c,0), holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c < n3_c)\}$. Its answer set $X_2 = A_{12}^{ts=0}$ – using forced atom proposition and construction of A_{12}.

   (a) where, $q_c = m_{M_0(p)}(c)$, and $n_c = m_{W(p,t)}(c)$ for an arc $(p,t) \in E^-$ – by construction of $i2$ and $f14$ in Π^6, and

   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 31 of preset)

   (c) $n2_c = 1$ – by construction of $iptarc$ predicates in Π^6, meaning $q_c \geq n2_c \equiv q_c \geq 1 \equiv q_c > 0$,

   (d) $tptarc(p,t,n3_c,c,0)$ represents $n3_c = m_{QW(p,t)}(c)$, where $(p,t) \in Q$

   (e) thus, $notenabled(t,0) \in X_1$ represents $\exists c \in C, (\exists p \in \bullet t : m_{M_0(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t) : m_{M_0(p)}(c) > 0) \vee (\exists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c))$.

4. $eval_{U_2}(bot_{U_3}(\Pi^6) \setminus bot_{U_2}(\Pi^6), X_0 \cup \cdots \cup X_2) = \{enabled(t,0){:}\text{-}.|trans(t) \in X_0 \cup \cdots \cup X_2, notenabled(t,0) \notin X_0 \cup \cdots \cup X_2\}$. Its answer set is $X_3 = A_{13}^{ts=0}$ – using forced atom proposition and construction of A_{13}.

   (a) since an $enabled(t,0) \in X_3$ if $\nexists$ $notenabled(t,0) \in X_0 \cup \cdots \cup X_2$; which is equivalent to $\nexists t, \forall c \in C, (\nexists p \in \bullet t, m_{M_0(p)}(c) < m_{W(p,t)}(c)), (\nexists p \in I(t), m_{M_0(p)}(c) > 0), (\nexists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c)), \forall c \in C, (\forall p \in \bullet t : m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t) : m_{M_0(p)}(c) = 0)$.

5. $eval_{U_3}(bot_{U_4}(\Pi^6) \setminus bot_{U_3}(\Pi^6), X_0 \cup \cdots \cup X_3) = \{notprenabled(t,0){:}\text{-}.| \{enabled(t,0), transpr(t,p), enabled(tt,0), transpr(tt,pp)\} \subseteq X_0 \cup \cdots \cup X_3, pp < p\}$. Its answer set is $X_4 = A_{23}^{ts=k}$ – using forced atom proposition and construction of A_{23}.

   (a) $enabled(t,0)$ represents $\exists t \in T, \forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_0(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c))$

   (b) $enabled(tt,0)$ represents $\exists tt \in T, \forall c \in C, (\forall pp \in \bullet tt, m_{M_0(pp)}(c) \geq m_{W(pp,tt)}(c)) \wedge (\forall pp \in I(tt), m_{M_0(pp)}(c) = 0), (\forall (pp,tt) \in Q, m_{M_0(pp)}(c) \geq m_{QW(pp,tt)}(c))$

   (c) $transpr(t,p)$ represents $p = Z(t)$ – by construction

   (d) $transpr(tt,pp)$ represents $pp = Z(tt)$ – by construction

(e) thus, $notprenabled(t,0)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)) \wedge (\forall p \in I(t), m_{M_0(p)}(c) = 0), \exists tt \in T, (\forall pp \in \bullet tt, m_{M_0(pp)}(c) \geq m_{W(pp,tt)}(c)) \wedge (\forall pp \in I(tt), m_{M_0(pp)}(c) = 0), Z(tt) < Z(t)$

(f) which is equivalent to $(\forall p \in \bullet t : M_0(p) \geq W(p,t)) \wedge (\forall p \in I(t), M_0(p) = 0), \exists tt \in T, (\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = 0), Z(tt) < Z(t)$ – assuming multiset domain C for all operations

6. $eval_{U_4}(bot_{U_5}(\Pi^6) \setminus bot_{U_4}(\Pi^6), X_0 \cup \cdots \cup X_4) = \{prenabled(t,0)\text{:-}.|enabled(t,0) \in X_0 \cup \cdots \cup X_4, notprenabled(t,0) \notin X_0 \cup \cdots \cup X_4\}$. Its answer set is $X_5 = A_{24}^{ts=k}$ – using forced atom proposition and construction of A_{24}

   (a) $enabled(t,0)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_0(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c)) \equiv (\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t))$ – from 4 above and assuming multiset domain C for all operations

   (b) $notprenabled(t,0)$ represents $(\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = 0), (\forall (pp,tt) \in Q, M_0(pp) \geq W(pp,tt)), Z(tt) < Z(t)$ – from 5 above and assuming multiset domain C for all operations

   (c) then, $prenabled(t,0)$ represents $(\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \nexists tt \in T, ((\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = 0), (\forall (pp,tt) \in Q, M_0(pp) \geq W(pp,tt)), Z(tt) < Z(t))$ – from (a), (b) and $enabled(t,0) \in X_0 \cup \cdots \cup X_4$

7. $eval_{U_5}(bot_{U_6}(\Pi^6) \setminus bot_{U_5}(\Pi^6), X_0 \cup \cdots \cup X_5) = \{\{fires(t,0)\}\text{:-}.|prenabled(t,0)$ holds in $X_0 \cup \cdots \cup X_5\}$. It has multiple answer sets $X_{6.1}, \dots, X_{6.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $eval_{U_5}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^6 determined using splitting is equal to A, we only consider the set that matches the $fires(t,0)$ elements in A and call it X_6, ignoring the rest. Thus, $X_6 = A_{14}^{ts=0}$, representing T_0.

   (a) in addition, for every t such that $prenabled(t,0) \in X_0 \cup \cdots \cup X_5, R(t) \neq \emptyset$; $fires(t,0) \in X_6$ – per definition 41 (firing set); requiring that a reset transition is fired when enabled

   (b) thus, the firing set T_0 will not be eliminated by the constraint $f20$

8. $eval_{U_6}(bot_{U_7}(\Pi^6) \setminus bot_{U_6}(\Pi^6), X_0 \cup \cdots \cup X_6) = \{add(p,n_c,t,c,0)\text{:-}.|\{fires(t,0), tparc(t,p,n_c,c,0)\} \subseteq X_0 \cup \cdots \cup X_6\} \cup \{del(p,n_c,t,c,0)\text{:-}.|\{fires(t,0), ptarc(p,t,n_c,c,0)\} \subseteq X_0 \cup \cdots \cup X_6\}$. It's answer set is $X_7 = A_{15}^{ts=0} \cup A_{16}^{ts=0}$ – using forced atom proposition and definitions of A_{15} and A_{16}.

   (a) where, each add atom is equivalent to $n_c = m_{W(t,p)}(c), c \in C, p \in t\bullet$,

   (b) and each del atom is equivalent to $n_c = m_{W(p,t)}(c), c \in C, p \in \bullet t$; or $n_c = m_{M_0(p)}(c), c \in C, p \in R(t)$,

   (c) representing the effect of transitions in T_0 – by construction

9. $eval_{U_7}(bot_{U_8}(\Pi^6) \setminus bot_{U_7}(\Pi^6), X_0 \cup \cdots \cup X_7) = \{tot\_incr(p, qq_c, c, 0)\text{:-}.|qq_c = \sum_{add(p,q_c,t,c,0) \in X_0 \cup \cdots \cup X_7} q_c\} \cup \{tot\_decr(p, qq_c, c, 0)\text{:-}.|qq_c = \sum_{del(p,q_c,t,c,0) \in X_0 \cup \cdots \cup X_7} q_c\}$. It's answer set is $X_8 = A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of A_{17} and A_{18}.

   (a) where, each $tot\_incr(p, qq_c, c, 0)$, $qq_c = \sum_{add(p,q_c,t,c,0) \in X_0 \cup \ldots X_7} q_c$ $\equiv qq_c = \sum_{t \in X_6, p \in t\bullet} m_{W(p,t)}(c)$,

   (b) where, each $tot\_decr(p, qq_c, c, 0)$, $qq_c = \sum_{del(p,q_c,t,c,0) \in X_0 \cup \ldots X_7} q_c$ $\equiv qq = \sum_{t \in X_6, p \in \bullet t} m_{W(t,p)}(c) + \sum_{t \in X_6, p \in R(t)} m_{M_0(p)}(c)$,

   (c) represent the net effect of transitions in T_0

10. $eval_{U_8}(bot_{U_9}(\Pi^6) \setminus bot_{U_8}(\Pi^6), X_0 \cup \cdots \cup X_8) = \{consumesmore(p, 0)\text{:-}.|$ $\{holds(p, q_c, c, 0), tot\_decr(p, q1_c, c, 0)\} \subseteq X_0 \cup \cdots \cup X_8, q1_c > q_c\} \cup$ $\{holds(p, q_c, c, 1)\text{:-}.|\{holds(p, q1_c, c, 0), tot\_incr(p, q2_c, c, 0), tot\_decr(p, q3_c, c, 0)\}$ $\subseteq X_0 \cup \cdots \cup X_6, q_c = q1_c + q2_c - q3_c\} \cup \{could\_not\_have(t, 0)\text{:-}.|\{prenabled(t, 0),$ $ptarc(s, t, q, c, 0), holds(s, qq, c, 0), tot\_decr(s, qqq, c, 0)\} \subseteq X_0 \cup \cdots \cup X_8,$ $fires(t, 0) \notin X_0 \cup \cdots \cup X_8, q > qq - qqq\}$. It's answer set is $X_9 = A_{19}^{ts=0} \cup A_{21}^{ts=0} \cup A_{25}^{ts=0}$ – using forced atom proposition and definitions of A_{19}, A_{21}, A_{25}.

   (a) where, $consumesmore(p, 0)$ represents $\exists p : q_c = m_{M_0(p)}(c), q1_c = \sum_{t \in T_0, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_0, p \in R(t)} m_{M_0(p)}(c), q1_c > q_c, c \in C$, indicating place p will be over consumed if T_0 is fired, as defined in definition 34 (conflicting transitions),

   (b) $holds(p, q_c, c, 1)$ if $q_c = m_{M_0(p)}(c) + \sum_{t \in T_0, p \in t\bullet} m_{W(t,p)}(c) -$ $(\sum_{t \in T_0, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_0, p \in R(t)} m_{M_0(p)}(c))$ represented by $q_c = m_{M_1(p)}(c)$ for some $c \in C$ – by construction of Π^6

   (c) $could\_not\_have(t, 0)$ if

      i. $(\forall p \in \bullet t, W(p, t) \leq M_0(p)), (\forall p \in I(t), M_0(p) = 0), (\forall (p, t) \in Q,$ $M_0(p) \geq WQ(p, t)), \nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_0(pp) = 0), (\forall (pp, tt) \in Q, M_0(pp) \geq QW(pp, tt)), Z(tt) < Z(t)$,

      ii. and $q_c > m_{M_0(s)}(c) - (\sum_{t' \in T_0, s \in \bullet t'} m_{W(s,t')}(c) + \sum_{t' \in T_0, s \in R(t)} m_{M_0(s)}(c))$, $q_c = m_{W(s,t)}(c)$ if $s \in \bullet t$ or $m_{M_0(s)}(c)$ otherwise for some $c \in C$, which becomes $q > M_0(s) - (\sum_{t' \in T_0, s \in \bullet t'} W(s, t') + \sum_{t' \in T_0, s \in R(t)} M_0(s)), q = W(s, t)$ if $s \in \bullet t$ or $M_0(s)$ otherwise for all $c \in C$

      iii. (i), (ii) above combined match the definition of A_{25}

   (d) X_9 does not contain $could\_not\_have(t, 0)$, when $prenabled(t, 0) \in X_0 \cup \cdots \cup X_8$ and $fires(t, 0) \notin X_0 \cup \cdots \cup X_8$ due to construction of A, encoding of $a9$ and its body atoms. As a result, it is not eliminated by the constraint $a6$

$\vdots$

11. $eval_{U_{9k+0}}(bot_{U_{9k+1}}(\Pi^6) \backslash bot_{U_{9k+0}}(\Pi^6), X_0 \cup \dots \cup X_{9k+0}) = \{ptarc(p,t,q_c,c,k)\text{:-.}|c \in C, q_c = m_{W(p,t)}(c), q_c > 0\} \cup \{tparc(t,p,q_c,c,k)\text{:-.}|c \in C, q_c = m_{W(t,p)}(c), q_c > 0\} \cup \{ptarc(p,t,q_c,c,k)\text{:-.}|c \in C, q_c = m_{M_0(p)}(c), q_c > 0\} \cup \{iptarc(p,t,1,c,k)\text{:-.}|c \in C\} \cup \{tptarc(p,t,q_c,c,k)\text{:-.}|c \in C, q_c = m_{QW(p,t)}(c), q_c > 0\}$. Its answer set $X_{9k+1} = A_6^{ts=k} \cup A_7^{ts=k} \cup A_9^{ts=k} \cup A_{10}^{ts=k} \cup A_{11}^{ts=k}$ – using forced atom proposition and construction of $A_6, A_7, A_9, A_{10}, A_{11}$.

12. $eval_{U_{9k+1}}(bot_{U_{9k+2}}(\Pi^6) \setminus bot_{U_{9k+1}}(\Pi^6), X_0 \cup \dots \cup X_{9k+1}) = \{notenabled(t,k)\text{:-.}| (\{trans(t), ptarc(p,t,n_c,c,k), holds(p,q_c,c,k)\} \subseteq X_0 \cup \dots \cup X_{10k+1},$ where $q_c < n_c)$ or $(\{notenabled(t,k)\text{:-.}| (\{trans(t), iptarc(p,t,n2_c,c,k), holds(p,q_c,c,k)\} \subseteq X_0 \cup \dots \cup X_{10k+1},$ where $q_c \geq n2_c\})$ or $(\{trans(t), tptarc(p,t,n3_c,c,k),$ $holds(p,q_c,c,k)\} \subseteq X_0 \cup X_{9k+1},$ where $q_c < n3_c)\}$. Its answer set $X_{9k+2} = A_{12}^{ts=k}$ – using forced atom proposition and construction of A_{12}.

    (a) where, $q_c = m_{M_0(p)}(c)$, and $n_c = m_{W(p,t)}(c)$ for an arc $(p,t) \in E^-$ – by construction of $i2$ and $f14$ predicates in Π^6, and

    (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 31 of preset)

    (c) $n2_c = 1$ – by construction of $iptarc$ predicates in Π^6, meaning $q_c \geq n2_c \equiv q_c \geq 1 \equiv q_c > 0$,

    (d) $tptarc(p,t,n3_c,c,k)$ represents $n3_c = m_{QW(p,t)}(c)$, where $(p,t) \in Q$

    (e) thus, $notenabled(t,k) \in X_{9k+1}$ represents $\exists c \in C, (\exists p \in \bullet t : m_{M_0(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t) : m_{M_0(p)}(c) > k) \vee (\exists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c))$.

13. $eval_{U_{9k+2}}(bot_{U_{9k+3}}(\Pi^6) \backslash bot_{U_{9k+2}}(\Pi^6), X_0 \cup \dots \cup X_{9k+2}) = \{enabled(t,k)\text{:-.}|trans(t) \in X_0 \cup \dots \cup X_{9k+2}, notenabled(t,k) \notin X_0 \cup \dots \cup X_{9k+2}\}$. Its answer set is $X_{9k+3} = A_{13}^{ts=k}$ – using forced atom proposition and construction of A_{13}.

    (a) since an $enabled(t,k) \in X_{9k+3}$ if $\nexists$ $notenabled(t,k) \in X_0 \cup \dots \cup X_{9k+2}$; which is equivalent to $\nexists t, \forall c \in C, (\nexists p \in \bullet t, m_{M_0(p)}(c) < m_{W(p,t)}(c)), (\nexists p \in I(t), m_{M_0(p)}(c) > k), (\nexists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c)), \forall c \in C, (\forall p \in \bullet t : m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t) : m_{M_0(p)}(c) = k)$.

14. $eval_{U_{9k+3}}(bot_{U_{9k+4}}(\Pi^6) \setminus bot_{U_{9k+3}}(\Pi^6), X_0 \cup \dots \cup X_{9k+3}) = \{notprenabled(t,k)\text{:-.}| \{enabled(t,k), transpr(t,p), enabled(tt,k), transpr(tt,pp)\} \subseteq X_0 \cup \dots \cup X_{9k+3}, pp < p\}$. Its answer set is $X_{9k+4} = A_{23}^{ts=k}$ – using forced atom proposition and construction of A_{23}.

    (a) $enabled(t,k)$ represents $\exists t \in T, \forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)),$ $(\forall p \in I(t), m_{M_0(p)}(c) = k), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c))$

    (b) $enabled(tt,k)$ represents $\exists tt \in T, \forall c \in C, (\forall pp \in \bullet tt, m_{M_0(pp)}(c) \geq m_{W(pp,tt)}(c)) \wedge (\forall pp \in I(tt), m_{M_0(pp)}(c) = k), (\forall (pp,tt) \in Q, m_{M_0(pp)}(c) \geq m_{QW(pp,tt)}(c))$

    (c) $transpr(t,p)$ represents $p = Z(t)$ – by construction

    (d) $transpr(tt,pp)$ represents $pp = Z(tt)$ – by construction

(e) thus, $\mathit{notprenabled}(t,k)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)) \wedge (\forall p \in I(t), m_{M_0(p)}(c) = k), \exists tt \in T, (\forall pp \in \bullet tt, m_{M_0(pp)}(c) \geq m_{W(pp,tt)}(c)) \wedge (\forall pp \in I(tt), m_{M_0(pp)}(c) = k), Z(tt) < Z(t)$

(f) which is equivalent to $(\forall p \in \bullet t : M_0(p) \geq W(p,t)) \wedge (\forall p \in I(t), M_0(p) = k), \exists tt \in T, (\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = k), Z(tt) < Z(t)$ – assuming multiset domain C for all operations

15. $\mathit{eval}_{U_{9k+4}}(\mathit{bot}_{U_{9k+5}}(\Pi^6) \setminus \mathit{bot}_{U_{9k+4}}(\Pi^6), X_0 \cup \cdots \cup X_{9k+4}) = \{\mathit{prenabled}(t,k){:}\text{-}.| \mathit{enabled}(t,k) \in X_0 \cup \cdots \cup X_{9k+4}, \mathit{notprenabled}(t,k) \notin X_0 \cup \cdots \cup X_{9k+4}\}$. Its answer set is $X_{9k+5} = A_{24}^{ts=k}$ – using forced atom proposition and construction of A_{24}

    (a) $\mathit{enabled}(t,k)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_0(p)}(c) = k), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c)) \equiv (\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = k), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t))$ – from 13 above and assuming multiset domain C for all operations

    (b) $\mathit{notprenabled}(t,k)$ represents $(\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = k), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = k), (\forall (pp,tt) \in Q, M_0(pp) \geq W(pp,tt)), Z(tt) < Z(t)$ – from 14 above and assuming multiset domain C for all operations

    (c) then, $\mathit{prenabled}(t,k)$ represents $(\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = k), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \nexists tt \in T, ((\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = k), (\forall (pp,tt) \in Q, M_0(pp) \geq W(pp,tt)), Z(tt) < Z(t))$ – from (a), (b) and $\mathit{enabled}(t,k) \in X_0 \cup \cdots \cup X_{9k+4}$

16. $\mathit{eval}_{U_{9k+5}}(\mathit{bot}_{U_{9k+6}}(\Pi^6) \setminus \mathit{bot}_{U_{9k+5}}(\Pi^6), X_0 \cup \cdots \cup X_{9k+5}) = \{\{fires(t,k)\}{:}\text{-}.| \mathit{prenabled}(t,k)$ holds in $X_0 \cup \cdots \cup X_{9k+5}\}$. It has multiple answer sets $X_{9k+6.1}, \dots, X_{9k+6.n}$, corresponding to elements of power set of $fires(t,k)$ atoms in $\mathit{eval}_{U_{9k+5}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^6 determined using splitting is equal to A, we only consider the set that matches the $fires(t,k)$ elements in A and call it X_{9k+6}, ignoring the rest. Thus, $X_{9k+6} = A_{14}^{ts=k}$, representing T_k.

    (a) in addition, for every t such that $\mathit{prenabled}(t,k) \in X_0 \cup \cdots \cup X_{9k+5}, R(t) \neq \emptyset$; $fires(t,k) \in X_{9k+6}$ – per definition 41 (firing set); requiring that a reset transition is fired when enabled

    (b) thus, the firing set T_k will not be eliminated by the constraint $f20$

17. $\mathit{eval}_{U_{9k+6}}(\mathit{bot}_{U_{9k+7}}(\Pi^6) \setminus \mathit{bot}_{U_{9k+6}}(\Pi^6), X_0 \cup \cdots \cup X_{9k+6}) = \{\mathit{add}(p,n_c,t,c,k){:}\text{-}.| \{fires(t,k), \mathit{tparc}(t,p,n_c,c,k)\} \subseteq X_0 \cup \cdots \cup X_{9k+6}\} \cup \{\mathit{del}(p,n_c,t,c,k){:}\text{-}.| \{fires(t,k), \mathit{ptarc}(p,t,n_c,c,k)\} \subseteq X_0 \cup \cdots \cup X_{9k+6}\}$. It's answer set is $X_{9k+7} = A_{15}^{ts=k} \cup A_{16}^{ts=k}$ – using forced atom proposition and definitions of A_{15} and A_{16}.

    (a) where, each add atom is equivalent to $n_c = m_{W(t,p)}(c), c \in C, p \in t\bullet$,

(b) and each del atom is equivalent to $n_c = m_{W(p,t)}(c), c \in C, p \in \bullet t$; or $n_c = m_{M_0(p)}(c), c \in C, p \in R(t)$,

(c) representing the effect of transitions in T_k.

18. $eval_{U_{9k+7}}(bot_{U_{9k+8}}(\Pi^6) \setminus bot_{U_{9k+7}}(\Pi^6), X_0 \cup \cdots \cup X_{9k+7}) = \{tot\_incr(p, qq_c, c, k){:}\text{-}.| qq_c = \sum_{add(p,q_c,t,c,k) \in X_0 \cup \cdots \cup X_{10k+7}} q_c\} \cup \{tot\_decr(p, qq_c, c, k){:}\text{-}.| qq_c = \sum_{del(p,q_c,t,c,k) \in X_0 \cup \cdots \cup X_{9k+7}} q_c\}$. It's answer set is $X_{10k+8} = A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition and definitions of A_{17} and A_{18}.

    (a) where, each $tot\_incr(p, qq_c, c, k)$, $qq_c = \sum_{add(p,q_c,t,c,k) \in X_0 \cup \ldots X_{10k+7}} q_c$
    $\equiv qq_c = \sum_{t \in X_{9k+6}, p \in t\bullet} m_{W(p,t)}(c)$,

    (b) and each $tot\_decr(p, qq_c, c, k)$, $qq_c = \sum_{del(p,q_c,t,c,k) \in X_0 \cup \ldots X_{10k+7}} q_c$
    $\equiv qq = \sum_{t \in X_{10k+6}, p \in \bullet t} m_{W(t,p)}(c) + \sum_{t \in X_{9k+6}, p \in R(t)} m_{M_0(p)}(c)$,

    (c) represent the net effect of transitions in T_k

19. $eval_{U_{9k+8}}(bot_{U_{9k+9}}(\Pi^6) \setminus bot_{U_{9k+8}}(\Pi^6), X_0 \cup \cdots \cup X_{9k+8}) = \{consumesmore(p, k){:}\text{-}.| \{holds(p, q_c, c, k), tot\_decr(p, q1_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+8}, q1_c > q_c\} \cup \{holds(p, q_c, c, 1){:}\text{-}.| \{holds(p, q1_c, c, k), tot\_incr(p, q2_c, c, k), tot\_decr(p, q3_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+6}, q_c = q1_c + q2_c - q3_c\} \cup \{could\_not\_have(t, k){:}\text{-}.| \{prenabled(t, k), ptarc(s, t, q, c, k), holds(s, qq, c, k), tot\_decr(s, qqq, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+8}, fires(t, k) \notin X_0 \cup \cdots \cup X_{9k+8}, q > qq - qqq\}$. It's answer set is $X_{10k+9} = A_{19}^{ts=k} \cup A_{21}^{ts=k} \cup A_{25}^{ts=k}$ – using forced atom proposition and definitions of A_{19}, A_{21}, A_{25}.

    (a) where, $consumesmore(p, k)$ represents $\exists p : q_c = m_{M_0(p)}(c)$, $q1_c = \sum_{t \in T_k, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_k, p \in R(t)} m_{M_0(p)}(c), q1_c > q_c, c \in C$, indicating place p that will be over consumed if T_k is fired, as defined in definition 34 (conflicting transition)

    (b) $holds(p, q_c, c, k+1)$ if $q_c = m_{M_0(p)}(c) + \sum_{t \in T_k, p \in t\bullet} m_{W(t,p)}(c) - (\sum_{t \in T_k, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_k, p \in R(t)} m_{M_0(p)}(c))$ represented by $q_c = m_{M_1(p)}(c)$ for some $c \in C$ – by construction of Π^6

    (c) $could\_not\_have(t, k)$ if

        i. $(\forall p \in \bullet t, W(p,t) \leq M_0(p)), (\forall p \in I(t), M_0(p) = k), (\forall (p,t) \in Q, M_0(p) \geq WQ(p,t)), \nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_0(pp) = k), (\forall (pp, tt) \in Q,\ M_0(pp) \geq QW(pp, tt)), Z(tt) < Z(t)$,

        ii. and $q_c > m_{M_0(s)}(c) - (\sum_{t' \in T_k, s \in \bullet t'} m_{W(s,t')}(c) + \sum_{t' \in T_k, s \in R(t)} m_{M_0(s)}(c))$, $q_c = m_{W(s,t)}(c)$ if $s \in \bullet t$ or $m_{M_0(s)}(c)$ otherwise for some $c \in C$, which becomes $q > M_0(s) - (\sum_{t' \in T_k, s \in \bullet t'} W(s, t') + \sum_{t' \in T_k, s \in R(t)} M_0(s)), q = W(s,t)$ if $s \in \bullet t$ or $M_0(s)$ otherwise for all $c \in C$

        iii. (i), (ii) above combined match the definition of A_{25}

    (d) X_{9k+9} does not contain $could\_not\_have(t, k)$, when $prenabled(t, k) \in X_0 \cup \cdots \cup X_{9k+6}$ and $fires(t, k) \notin X_0 \cup \cdots \cup X_{9k+5}$due to construction of A, encoding of $a10$ and its body atoms. As a result it is not eliminated by the constraint $a6$

20. $eval_{U_{9k+9}}(bot_{U_{9k+10}}(\Pi^6) \setminus bot_{U_{9k+9}}(\Pi^6), X_0 \cup \dots \cup X_{9k+9}) = \{consumesmore\text{:-.}| \{consumesmore(p,k)\} \subseteq A\}$. It's answer set is $X_{9k+10} = A_{20}$ – using forced atom proposition and the definition of A_{20}

    (a) X_{9k+10} will be empty since none of $consumesmore(p,0), \dots, consumesmore(p,k)$ hold in $X_0 \cup \dots \cup X_{9k+9}$ due to the construction of A, encoding of $a9$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \dots \cup X_{9k+10}$ is the answer set of Π^6 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 9k+10$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^6.

Next we show (II): Given Π^6 be the encoding of a Petri Net $PN(P,T,E,C,W,R,I,Q,QW,Z)$ with initial marking M_0, and A be an answer set of Π^6 that satisfies (A.13) and (A.14), then we can construct $X = M_0, T_k, \dots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \dots, M_i(p_n))$, where $\{holds(p_0, m_{M_i(p_0)}(c), c, i), \dots, holds(p_n, m_{M_i(p_n)}(c), c, i)\} \subseteq A$, for $c \in C, 0 \leq i \leq k+1$
2. $T_i = \{t : fires(t,i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) firing T_0 in M_0 produces marking M_1.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0,0), \dots, fires(t_x,0)\}$ be the set of all $fires(\dots,0)$ atoms in A,

    (a) Then for each $fires(t_i,0) \in A$

        i. $prenabled(t_i,0) \in A$ – from rule $a13$ and supported rule proposition
        ii. Then $enabled(t_i,0) \in A$ – from rule $a12$ and supported rule proposition
        iii. And $notprenabled(t_i,0) \notin A$ – from rule $a12$ and supported rule proposition
        iv. For $enabled(t_i,0) \in A$

            A. $notenabled(t_i,0) \notin A$ – from rule $e9$ and supported rule proposition
            B. Then either of $body(e6)$, $body(e7)$, or $body(e8)$ must not hold in A for t_i – from rules $e6, e7, e8$ and forced atom proposition
            C. Then $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e6$ for all $\{holds(p,q_c,c,0), ptarc(p,t_i,n_{ic},c,0)\} \subseteq A$ – from $e:6$, forced atom proposition, and given facts $(holds(p,q_c,c,0) \in A, ptarc(p,t_i,n_{ic},0) \in A)$

D. And $q_c \not\geq n_{ic} \equiv q_c < n_{ic}$ in $e7$ for all $\{holds(p, q_c, c, 0), iptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A, n_{ic} = 1$; $q_c > n_{ic} \equiv q_c = 0$ – from $e7$, forced atom proposition, given facts ($holds(p, q_c, c, 0) \in A$, $iptarc(p, t_i, 1, c, 0) \in A$), and q_c is a positive integer

E. And $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e8$ for all $\{holds(p, q_c, c, 0), tptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A$ – from $e8$, forced atom proposition, and given facts

F. Then $\forall c \in C, (\forall p \in \bullet t_i, m_{M_0(p)}(c) \geq m_{W(p,t_i)}(c)) \wedge (\forall p \in I(t_i), m_{M_0(p)}(c) = 0) \wedge (\forall (p, t_i) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t_i)}(c))$ – from $i2, f14, f17, f18$ construction, definition 31 of preset $\bullet t_i$ in PN, definition 40 of enabled transition in PN, and that the construction of reset arcs by $f16$ ensures $notenabled(t, 0)$ is never true for a reset arc, where $holds(p, q_c, c, 0) \in A$ represents $q_c = m_{M_0(p)}(c)$, $ptarc(p, t_i, n_{ic}, 0) \in A$ represents $n_{ic} = m_{W(p,t_i)}(c)$, $n_{ic} = m_{M_0(p)}(c)$.

G. Which is equivalent to $(\forall p \in \bullet t_i, M_0(p) \geq W(p, t_i)) \wedge (\forall p \in I(t_i), M_0(p) = 0) \wedge (\forall (p, t_i) \in Q, M_0(p) \geq QW(p, t_i))$ – assuming multiset domain C

v. For $notprenabled(t_i, 0) \notin A$

A. Either $(\nexists enabled(tt, 0) \in A : pp < p_i)$ or $(\forall enabled(tt, 0) \in A : pp \not< p_i)$ where $pp = Z(tt), p_i = Z(t_i)$ – from rule $a11, f19$ and forced atom proposition

B. This matches the definition of an enabled priority transition

vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 40 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^6 – from rule $a3$ and supported rule proposition

i. Then $\nexists consumesmore(p, 0) \in A$ – from rule $a9$ and supported rule proposition

ii. Then $\nexists\{holds(p, q_c, c, 0), tot\_decr(p, q1_c, c, 0)\} \subseteq A, q1_c > q_c$ in $body(a9)$ – from $a9$ and forced atom proposition

iii. Then $\nexists c \in C \nexists p \in P, (\sum_{t_i \in \{t_0,\ldots,t_x\}, p \in \bullet t_i} m_{W(p,t_i)}(c) + \sum_{t_i \in \{t_0,\ldots,t_x\}, p \in R(t_i)} m_{M_0(p)}(c)) > m_{M_0(p)}(c)$ – from the following

A. $holds(p, q_c, c, 0)$ represents $q_c = m_{M_0(p)}(c)$ – from rule $i2$ encoding, given

B. $tot\_decr(p, q1_c, c, 0) \in A$ if $\{del(p, q1_{0_c}, t_0, c, 0), \ldots, del(p, q1_{x_c}, t_x, c, 0)\} \subseteq A$, where $q1_c = q1_{0_c} + \cdots + q1_{x_c}$ – from $r11$ and forced atom proposition

C. $del(p, q1_{i_c}, t_i, c, 0) \in A$ if $\{fires(t_i, 0), ptarc(p, t_i, q1_{i_c}, c, 0)\} \subseteq A$ – from $r9$ and supported rule proposition

D. $del(p, q1_{i_c}, t_i, c, 0)$ either represents removal of $q1_{i_c} = m_{W(p,t_i)}(c)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_{i_c} = m_{M_0(p)}(c)$ tokens from $p \in R(t_i)$– from rules $r9, f14, f16$, supported rule proposition, and definition 33 of transition execution in PN

iv. Then the set of transitions in T_0 do not conflict – by the definition 34 of conflicting transitions

(c) And for each $\mathit{prenabled}(t_j, 0) \in A$ and $fires(t_j, 0) \notin A$, $\mathit{could\_not\_have}(t_j, 0) \in A$, since A is an answer set of Π^6 - from rule $a15$ and supported rule proposition

i. Then $\{\mathit{prenabled}(t_j, 0), \mathit{holds}(s, qq_c, c, 0), \mathit{ptarc}(s, t_j, q_c, c, 0), \mathit{tot\_decr}(s, qqq_c, c, 0)\} \subseteq A$, such that $q_c > qq_c - qqq_c$ and $fires(t_j, 0) \notin A$ - from rule $a14$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q_c > m_{M_0(s)}(c) - (\sum_{t_i \in T_0, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_0, s \in R(t_i)} m_{M_0(s)}(c))$, where $q_c = m_{W(s,t_j)}(c)$ if $s \in \bullet t_j$, or $m_{M_0(s)}(c)$ otherwise – from the following

A. $\mathit{ptarc}(s, t_i, q_c, c, 0)$ represents $q_c = m_{W(s,t_i)}(c)$ if $(s, t_i) \in E^-$ or $q_c = m_{M_0(s)}(c)$ if $s \in R(t_i)$ – from rule $f14, f16$ construction

B. $\mathit{holds}(s, qq_c, c, 0)$ represents $qq_c = m_{M_0(s)}(c)$ – from $i2$ construction

C. $\mathit{tot\_decr}(s, qqq_c, c, 0) \in A$ if $\{\mathit{del}(s, qqq_{0_c}, t_0, c, 0), \dots, \mathit{del}(s, qqq_{x_c}, t_x, c, 0)\} \subseteq A$ – from rule $r11$ construction and supported rule proposition

D. $\mathit{del}(s, qqq_{i_c}, t_i, c, 0) \in A$ if $\{fires(t_i, 0), \mathit{ptarc}(s, t_i, qqq_{i_c}, c, 0)\} \subseteq A$ – from rule $r9$ and supported rule proposition

E. $\mathit{del}(s, qqq_{i_c}, t_i, c, 0)$ either represents $qqq_{i_c} = m_{W(s,t_i)}(c) : t_i \in T_0, (s, t_i) \in E^-$, or $qqq_{i_c} = m_{M_0(t_i)}(c) : t_i \in T_0, s \in R(t_i)$ – from rule $f14, f16$ construction

F. $\mathit{tot\_decr}(q, qqq_c, c, 0)$ represents $\sum_{t_i \in T_0, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_0, s \in R(t_i)} m_{M_0(s)}(c)$ – from (C,D,E) above

iii. Then firing $T_0 \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_0 is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $\mathit{prenabled}(t_r, 0) \in A$, $fires(t_r, 0) \in A$, since A is an answer set of Π^6 - from rule $f20$ and supported rule proposition

i. Then, the firing set T_0 satisfies the reset-transition requirement of definition 41 (firing set)

(e) Then $\{t_0, \dots, t_x\} = T_0$ – using 1(a),1(b),1(d) above; and using 1(c) it is a maximal firing set

2. We show M_1 is produced by firing T_0 in M_0. Let $\mathit{holds}(p, q_c, c, 1) \in A$

(a) Then $\{\mathit{holds}(p, q1_c, c, 0), \mathit{tot\_incr}(p, q2_c, c, 0), \mathit{tot\_decr}(p, q3_c, c, 0)\} \subseteq A : q_c = q1_c + q2_c - q3_c$ – from rule $r12$ and supported rule proposition

(b) Then, $\mathit{holds}(p, q1_c, c, 0) \in A$ represents $q1_c = m_{M_0(p)}(c)$ – given, construction;

and $\{\mathit{add}(p, q2_{0_c}, t_0, c, 0), \dots, \mathit{add}(p, q2_{j_c}, t_j, c, 0)\} \subseteq A : q2_{0_c} + \cdots + q2_{j_c} = q2_c$ and $\{\mathit{del}(p, q3_{0_c}, t_0, c, 0), \dots, \mathit{del}(p, q3_{l_c}, t_l, c, 0)\} \subseteq A : q3_{0_c} + \cdots + q3_{l_c} = q3_c$ – rules $r10, r11$ and supported rule proposition, respectively

(c) Then $\{fires(t_0,0),\dots,fires(t_j,0)\} \subseteq A$ and $\{fires(t_0,0),\dots,fires(t_l,0)\} \subseteq A$ – rules $r8, r9$ and supported rule proposition, respectively

(d) Then $\{fires(t_0,0),\dots,fires(t_j,0)\} \cup \{fires(t_0,0),\dots,fires(t_l,0)\} \subseteq A = \{fires(t_0,0),\dots, fires(t_x,0)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x,0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

(f) Then $q_c = m_{M_0(p)}(c) + \sum_{t_x \in T_0 \wedge p \in t_x \bullet} m_{W(t_x,p)}(c) - (\sum_{t_x \in T_0 \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} m_{M_0(p)}(c))$ – from (2b) above and the following

   i. Each $add(p, q_{j_c}, t_j, c, 0) \in A$ represents $q_{j_c} = m_{W(t_j,p)}(c)$ for $p \in t_j \bullet$ – rule $r8, f15$ encoding, and definition 33 of transition execution in PN

   ii. Each $del(p, t_y, q_{y_c}, c, 0) \in A$ represents either $q_{y_c} = m_{W(p,t_y)}(c)$ for $p \in \bullet t_y$, or $q_{y_c} = m_{M_0(p)}(c)$ for $p \in R(t_y)$ – from rule $r9, f14$ encoding and definition 33 of transition execution in PN; or from rule $r9, f16$ encoding and definition of reset arc in PN

   iii. Each $tot\_incr(p, q2_c, c, 0) \in A$ represents $q2_c = \sum_{t_x \in T_0 \wedge p \in t_x \bullet} m_{W(t_x,p)}(c)$ – aggregate assignment atom semantics in rule $r10$

   iv. Each $tot\_decr(p, q3_c, c, 0) \in A$ represents $q3_c = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} m_{M_0(p)}(c)$ – aggregate assignment atom semantics in rule $r11$

(g) Then, $m_{M_1(p)}(c) = q_c$ – since $holds(p, q_c, c, 1) \in A$ encodes $q_c = m_{M_1(p)}(c)$ – from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set in M_k. Let $\{fires(T_k,k),\dots,fires(t_x,k)\}$ be the set of all $fires(\dots,k)$ atoms in A,

   (a) Then for each $fires(t_i,k) \in A$

      i. $prenabled(t_i,k) \in A$ – from rule $a13$ and supported rule proposition

      ii. Then $enabled(t_i,k) \in A$ – from rule $a12$ and supported rule proposition

      iii. And $notprenabled(t_i,k) \notin A$ – from rule $a12$ and supported rule proposition

      iv. For $enabled(t_i,k) \in A$

         A. $notenabled(t_i,k) \notin A$ – from rule $e9$ and supported rule proposition

         B. Then either of $body(e6)$, $body(e7)$, or $body(e8)$ must not hold in A for t_i – from rules $e6, e7, e8$ and forced atom proposition

         C. Then $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e6$ for all $\{holds(p,q_c,c,k), ptarc(p,t_i,n_{ic},c,k)\} \subseteq A$ – from $e6$, forced atom proposition, and given facts $(holds(p,q_c,c,k) \in A, ptarc(p,t_i,n_{ic},k) \in A)$

D. And $q_c \not\geq n_{ic} \equiv q_c < n_{ic}$ in $e7$ for all $\{\mathit{holds}(p, q_c, c, k), \mathit{iptarc}(p, t_i, n_{ic}, c, k)\} \subseteq A, n_{ic} = 1$; $q_c > n_{ic} \equiv q_c = 0$ – from $e7$, forced atom proposition, given facts ($\mathit{holds}(p, q_c, c, k) \in A$, $\mathit{iptarc}(p, t_i, 1, c, k) \in A$), and q_c is a positive integer

E. And $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e8$ for all $\{\mathit{holds}(p, q_c, c, k), \mathit{tptarc}(p, t_i, n_{ic}, c, k)\} \subseteq A$ – from $e8$, forced atom proposition, and given facts

F. Then $\forall c \in C, (\forall p \in \bullet t_i, m_{M_k(p)}(c) \geq m_{W(p,t_i)}(c)) \wedge (\forall p \in I(t_i), m_{M_k(p)}(c) = k) \wedge (\forall (p, t_i) \in Q, m_{M_k(p)}(c) \geq m_{QW(p,t_i)}(c))$ – from $f14, f17, f18$ construction, inductive assumption, definition 31 of preset $\bullet t_i$ in PN, definition 40 of enabled transition in PN, and that the construction of reset arcs by $f16$ ensures $\mathit{notenabled}(t, k)$ is never true for a reset arc, where $\mathit{holds}(p, q_c, c, k) \in A$ represents $q_c = m_{M_k(p)}(c)$, $\mathit{ptarc}(p, t_i, n_{ic}, k) \in A$ represents $n_{ic} = m_{W(p,t_i)}(c)$, $n_{ic} = m_{M_k(p)}(c)$.

G. Which is equivalent to $(\forall p \in \bullet t_i, M_k(p) \geq W(p, t_i)) \wedge (\forall p \in I(t_i), M_k(p) = k) \wedge (\forall (p, t_i) \in Q, M_k(p) \geq QW(p, t_i))$ – assuming multiset domain C

v. For $\mathit{notprenabled}(t_i, k) \notin A$

A. Either $(\nexists \mathit{enabled}(tt, k) \in A : pp < p_i)$ or $(\forall \mathit{enabled}(tt, k) \in A : pp \not< p_i)$ where $pp = Z(tt), p_i = Z(t_i)$ – from rule $a11, f19$ and forced atom proposition

B. This matches the definition of an enabled priority transition

vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 40 of enabled transition

(b) And $\mathit{consumesmore} \notin A$, since A is an answer set of Π^6 – from rule $a4$ and supported rule proposition

i. Then $\nexists \mathit{consumesmore}(p, k) \in A$ – from rule $a4$ and supported rule proposition

ii. Then $\nexists\{\mathit{holds}(p, q_c, c, k), \mathit{tot\_decr}(p, q1_c, c, k)\} \subseteq A, q1_c > q_c$ in $\mathit{body}(a9)$ – from $a9$ and forced atom proposition

iii. Then $\nexists c \in C \nexists p \in P, (\sum_{t_i \in \{T_k, \ldots, t_x\}, p \in \bullet t_i} m_{W(p,t_i)}(c) + \sum_{t_i \in \{T_k, \ldots, t_x\}, p \in R(t_i)} m_{M_k(p)}(c)) > m_{M_k(p)}(c)$ – from the following

A. $\mathit{holds}(p, q_c, c, k)$ represents $q_c = m_{M_k(p)}(c)$ – from inductive assumption and construction, given

B. $\mathit{tot\_decr}(p, q1_c, c, k) \in A$ if $\{\mathit{del}(p, q1_{0c}, T_k, c, k), \ldots, \mathit{del}(p, q1_{xc}, t_x, c, k)\} \subseteq A$, where $q1_c = q1_{0c} + \cdots + q1_{xc}$ – from $r11$ and forced atom proposition

C. $\mathit{del}(p, q1_{ic}, t_i, c, k) \in A$ if $\{\mathit{fires}(t_i, k), \mathit{ptarc}(p, t_i, q1_{ic}, c, k)\} \subseteq A$ – from $r9$ and supported rule proposition

D. $\mathit{del}(p, q1_{ic}, t_i, c, k)$ either represents removal of $q1_{ic} = m_{W(p,t_i)}(c)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_{ic} = m_{M_k(p)}(c)$ tokens from $p \in R(t_i)$– from rules $r9, f14, f16$, supported rule proposition, and definition 33 of transition execution in PN

iv. Then the set of transitions in T_k do not conflict – by the definition 34 of conflicting transitions

(c) And for each $prenabled(t_j, k) \in A$ and $fires(t_j, k) \notin A$, $could\_not\_have(t_j, k) \in A$, since A is an answer set of Π^6 - from rule $a15$ and supported rule proposition

i. Then $\{prenabled(t_j, k), holds(s, qq_c, c, k), ptarc(s, t_j, q_c, c, k), tot\_decr(s, qqq_c, c, k)\} \subseteq A$, such that $q_c > qq_c - qqq_c$ and $fires(t_j, k) \notin A$ - from rule $a14$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q_c > m_{M_k(s)}(c) - (\sum_{t_i \in T_k, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_k, s \in R(t_i)} m_{M_k(s)}(c))$, where $q_c = m_{W(s,t_j)}(c)$ if $s \in \bullet t_j$, or $m_{M_k(s)}(c)$ otherwise.

A. $ptarc(s, t_i, q_c, c, k)$ represents $q_c = m_{W(s,t_i)}(c)$ if $(s, t_i) \in E^-$ or $q_c = m_{M_k(s)}(c)$ if $s \in R(t_i)$ – from rule $f14, f16$ construction

B. $holds(s, qq_c, c, k)$ represents $qq_c = m_{M_k(s)}(c)$ – from inductive assumption and construction

C. $tot\_decr(s, qqq_c, c, k) \in A$ if $\{del(s, qqq_{0c}, T_k, c, k), \ldots, del(s, qqq_{xc}, t_x, c, k)\} \subseteq A$ – from rule $r11$ construction and supported rule proposition

D. $del(s, qqq_{ic}, t_i, c, k) \in A$ if $\{fires(t_i, k), ptarc(s, t_i, qqq_{ic}, c, k)\} \subseteq A$ – from rule $r9$ and supported rule proposition

E. $del(s, qqq_{ic}, t_i, c, k)$ either represents $qqq_{ic} = m_{W(s,t_i)}(c) : t_i \in T_k, (s, t_i) \in E^-$, or $qqq_{ic} = m_{M_k(t_i)}(c) : t_i \in T_k, s \in R(t_i)$ – from rule $f14, f16$ construction

F. $tot\_decr(q, qqq_c, c, k)$ represents $\sum_{t_i \in T_k, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_k, s \in R(t_i)} m_{M_k(s)}(c)$ – from (C,D,E) above

iii. Then firing $T_k \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_k is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $prenabled(t_r, k) \in A$, $fires(t_r, k) \in A$, since A is an answer set of Π^6 - from rule $f20$ and supported rule proposition

i. Then the firing set T_k satisfies the reset transition requirement of definition 41 (firing set)

(e) Then $\{t_0, \ldots, t_x\} = T_k$ – using 1(a),1(b), 1(d) above; and using 1(c) it is a maximal firing set

2. We show that M_{k+1} is produced by firing T_k in M_k. Let $holds(p, q_c, c, k+1) \in A$

(a) Then $\{holds(p, q1_c, c, k), tot\_incr(p, q2_c, c, k), tot\_decr(p, q3_c, c, k)\} \subseteq A : q_c = q1_c + q2_c - q3_c$ – from rule $r12$ and supported rule proposition

(b) Then, $holds(p, q1_c, c, k) \in A$ represents $q1_c = m_{M_k(p)}(c)$ – inductive assumption;

and $\{add(p,q2_{0_c},T_k,c,k),\dots,add(p,q2_{j_c},t_j,c,k)\} \subseteq A : q2_{0_c}+\dots+q2_{j_c} = q2_c$ and $\{del(p,q3_{0_c},T_k,c,k),\dots,\ del(p,q3_{l_c},t_l,c,k)\} \subseteq A : q3_{0_c}+\dots+q3_{l_c} = q3_c$ – rules $r10, r11$ and supported rule proposition, respectively

(c) Then $\{fires(T_k,k),\dots,fires(t_j,k)\} \subseteq A$ and $\{fires(T_k,k),\dots, fires(t_l,k)\} \subseteq A$ – rules $r8, r9$ and supported rule proposition, respectively

(d) Then $\{fires(T_k,k),\dots,\ fires(t_j,k)\} \cup \{fires(T_k,k),\dots,\ fires(t_l,k)\} \subseteq A = \{fires(T_k,k),\dots,fires(t_x,k)\} \subseteq A$ – set union of subsets

(e) Then for each $fires(t_x,k) \in A$ we have $t_x \in T_k$ – already shown in item 1 above

(f) Then $q_c = m_{M_k(p)}(c) + \sum_{t_x \in T_k \wedge p \in t_x \bullet} m_{W(t_x,p)}(c) - (\sum_{t_x \in T_k \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_k \wedge p \in R(t_x)} m_{M_k(p)}(c))$ – from (2b) above and the following

  i. Each $add(p,q_{j_c},t_j,c,k) \in A$ represents $q_{j_c} = m_{W(t_j,p)}(c)$ for $p \in t_j \bullet$ – rule $r8, f15$ encoding, and definition 33 of transition execution in PN

  ii. Each $del(p,t_y,q_{y_c},c,k) \in A$ represents either $q_{y_c} = m_{W(p,t_y)}(c)$ for $p \in \bullet t_y$, or $q_{y_c} = m_{M_k(p)}(c)$ for $p \in R(t_y)$ – from rule $r9, f14$ encoding and definition 33 of transition execution in PN; or from rule $r9, f16$ encoding and definition of reset arc in PN

  iii. Each $tot\_incr(p,q2_c,c,k) \in A$ represents $q2_c = \sum_{t_x \in T_k \wedge p \in t_x \bullet} m_{W(t_x,p)}(c)$ – aggregate assignment atom semantics in rule $r10$

  iv. Each $tot\_decr(p,q3_c,c,k) \in A$ represents $q3_c = \sum_{t_x \in T_k \wedge p \in \bullet t_x} m_{W(p,t_x)}(c) + \sum_{t_x \in T_k \wedge p \in R(t_x)} m_{M_k(p)}(c)$ – aggregate assignment atom semantics in rule $r11$

(g) Then, $m_{M_{k+1}(p)}(c) = q_c$ – since $holds(p,q_c,c,k+1) \in A$ encodes $q_c = m_{M_{k+1}(p)}(c)$ – from construction

As a result, for any $n > k$, T_n will be a valid firing set for M_n and M_{n+1} will be its target marking.
Conclusion: Since both (I) and (II) hold, $X = M_0, T_k, M_1, \dots, M_k, T_{k+1}$ is an execution sequence of $PN(P,T,E,C,W,R,I,Q,QW,Z)$ (w.r.t M_0) iff there is an answer set A of $\Pi^6(PN, M_0, k, ntok)$ such that (A.13) and (A.14) hold.

A.8 Proof of Proposition 8

Let $PN = (P,T,E,C,W,R,I,Q,QW,Z,D)$ be a Petri Net, M_0 be its initial marking and let $\Pi^7(PN, M_0, k, ntok)$ be the ASP encoding of PN and M_0 over a simulation length k, with maximum $ntok$ tokens on any place node, as defined in section 2.11. Then $X = M_0, T_k, M_1, \dots, M_k, T_k, M_{k+1}$ is an execution sequence of PN (w.r.t. M_0) iff there is an answer set A of $\Pi^7(PN, M_0, k, ntok)$ such that:

$$\{fires(t,ts) : t \in T_{ts}, 0 \leq ts \leq k\} = \{fires(t,ts) : fires(t,ts) \in A\} \tag{A.15}$$

$$\begin{aligned} \{holds(p,q,c,ts) : p \in P, c/q = M_{ts}(p), 0 \leq ts \leq k+1\} \\ = \{holds(p,q,c,ts) : holds(p,q,c,ts) \in A\} \end{aligned} \tag{A.16}$$

We prove this by showing that:

(I) Given an execution sequence X, we create a set A such that it satisfies (A.15) and (A.16) and show that A is an answer set of Π^7

(II) Given an answer set A of Π^7, we create an execution sequence X such that (A.15) and (A.16) are satisfied.

First we show (I): We create a set A as a union of the following sets:

1. $A_1 = \{num(n) : 0 \leq n \leq ntok\}$
2. $A_2 = \{time(ts) : 0 \leq ts \leq k\}$
3. $A_3 = \{place(p) : p \in P\}$
4. $A_4 = \{trans(t) : t \in T\}$
5. $A_5 = \{color(c) : c \in C\}$
6. $A_6 = \{ptarc(p, t, n_c, c, ts) : (p, t) \in E^-, c \in C, n_c = m_{W(p,t)}(c), n_c > 0, 0 \leq ts \leq k\}$, where $E^- \subseteq E$
7. $A_7 = \{tparc(t, p, n_c, c, ts, d) : (t, p) \in E^+, c \in C, n_c = m_{W(t,p)}(c), n_c > 0, d = D(t), 0 \leq ts \leq k\}$, where $E^+ \subseteq E$
8. $A_8 = \{holds(p, q_c, c, 0) : p \in P, c \in C, q_c = m_{M_0(p)}(c)\}$
9. $A_9 = \{ptarc(p, t, n_c, c, ts) : p \in R(t), c \in C, n_c = m_{M_{ts}(p)}, n_c > 0, 0 \leq ts \leq k\}$
10. $A_{10} = \{iptarc(p, t, 1, c, ts) : p \in I(t), c \in C, 0 \leq ts < k\}$
11. $A_{11} = \{tptarc(p, t, n_c, c, ts) : (p, t) \in Q, c \in C, n_c = m_{QW(p,t)}(c), n_c > 0, 0 \leq ts \leq k\}$
12. $A_{12} = \{notenabled(t, ts) : t \in T, 0 \leq ts \leq k, \exists c \in C, (\exists p \in \bullet t, m_{M_{ts}(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t), m_{M_{ts}(p)}(c) > 0) \vee (\exists (p, t) \in Q, m_{M_{ts}(p)}(c) < m_{QW(p,t)}(c))\}$
per definition 40 (enabled transition)
13. $A_{13} = \{enabled(t, ts) : t \in T, 0 \leq ts \leq k, \forall c \in C, (\forall p \in \bullet t, m_{W(p,t)}(c) \leq m_{M_{ts}(p)}(c)) \wedge (\forall p \in I(t), m_{M_{ts}(p)}(c) = 0) \wedge (\forall (p, t) \in Q, m_{M_{ts}(p)}(c) \geq m_{QW(p,t)}(c))\}$
per definition 40 (enabled transition)
14. $A_{14} = \{fires(t, ts) : t \in T_{ts}, 0 \leq ts \leq k\}$
per definition 40 (enabled transitions), only an enabled transition may fire
15. $A_{15} = \{add(p, q_c, t, c, ts + d - 1) : t \in T_{ts}, p \in t\bullet, c \in C, q_c = m_{W(t,p)}(c),$
$d = D(t), 0 \leq ts \leq k\}$
per definition 43 (transition execution)
16. $A_{16} = \{del(p, q_c, t, c, ts) : t \in T_{ts}, p \in \bullet t, c \in C, q_c = m_{W(p,t)}(c), 0 \leq ts \leq k\} \cup \{del(p, q_c, t, c, ts) : t \in T_{ts}, p \in R(t), c \in C, q_c = m_{M_{ts}(p)}(c), 0 \leq ts \leq k\}$
per definition 43 (transition execution)

17. $A_{17} = \{tot\_incr(p, q_c, c, ts) : p \in P, c \in C,$ $q_c = \sum_{t \in T_l, p \in t\bullet, l \leq ts, l+D(t)=ts+1} m_{W(t,p)}(c), 0 \leq ts \leq k\}$
    per definition 44 (firing set execution)

18. $A_{18} = \{tot\_decr(p, q_c, c, ts) : p \in P, c \in C, q_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} m_{M_{ts}(p)}(c), 0 \leq ts \leq k\}$
    per definition 44 (firing set execution)

19. $A_{19} = \{consumesmore(p, ts) : p \in P, c \in C, q_c = m_{M_{ts}(p)}(c),$ $q1_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} m_{M_{ts}(p)}(c), q1_c > q_c, 0 \leq ts \leq k\}$
    per definition 34 (conflicting transitions)

20. $A_{20} = \{consumesmore : \exists p \in P, c \in C, q_c = m_{M_{ts}(p)}(c),$ $q1_c = \sum_{t \in T_{ts}, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_{ts}, p \in R(t)} (m_{M_{ts}(p)}(c)), q1_c > q_c, 0 \leq ts \leq k\}$
    per definition 34 (conflicting transitions)

21. $A_{21} = \{holds(p, q_c, c, ts+1) : p \in P, c \in C, q_c = m_{M_{ts+1}(p)}(c), 0 \leq ts < k\}$, where $M_{ts+1}(p) = M_{ts}(p) - (\sum_{t \in T_{ts}, p \in \bullet t} W(p,t) + \sum_{t \in T_{ts}, p \in R(t)} M_{ts}(p)) + \sum_{t \in T_l, p \in t\bullet, l \leq ts, l+D(t)-1=ts} W(t,p)$
    according to definition 44 (firing set execution)

22. $A_{22} = \{transpr(t, pr) : t \in T, pr = Z(t)\}$

23. $A_{23} = \{notprenabled(t, ts) : t \in T, enabled(t, ts) \in A_{13}, (\exists tt \in T, enabled(tt, ts) \in A_{13}, Z(tt) < Z(t)), 0 \leq ts \leq k\}$
    $= \{notprenabled(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), (\forall p \in I(t), M_{ts}(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq WQ(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_{ts}(pp) = 0), (\forall (pp, tt) \in Q, M_{ts}(pp) \geq QW(pp, tt)), Z(tt) < Z(t), 0 \leq ts \leq k\}$

24. $A_{24} = \{prenabled(t, ts) : t \in T, enabled(t, ts) \in A_{13}, (\nexists tt \in T : enabled(tt, ts) \in A_{13}, Z(tt) < Z(t)), 0 \leq ts \leq k\}$
    $= \{prenabled(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), (\forall p \in I(t), M_{ts}(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq WQ(p,t)), \nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_{ts}(pp) = 0), (\forall (pp, tt) \in Q, M_{ts}(pp) \geq QW(pp, tt)), Z(tt) < Z(t), 0 \leq ts \leq k\}$

25. $A_{25} = \{could\_not\_have(t, ts) : t \in T, prenabled(t, ts) \in A_{24}, fires(t, ts) \notin A_{14}, (\exists p \in \bullet t \cup R(t) : q > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p, t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), q = W(p,t)$ if $(p,t) \in E^-$ or $R(t)$ otherwise$), 0 \leq ts \leq k\}$
    $= \{could\_not\_have(t, ts) : t \in T, (\forall p \in \bullet t, W(p,t) \leq M_{ts}(p)), (\forall p \in I(t), M_{ts}(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), (\nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_{ts}(pp) = 0), (\forall (pp, tt) \in Q, M_{ts}(pp) \geq QW(pp, tt)), Z(tt) < Z(t)), t \notin T_{ts}, (\exists p \in \bullet t \cup R(t) : q > M_{ts}(p) - (\sum_{t' \in T_{ts}, p \in \bullet t'} W(p, t') + \sum_{t' \in T_{ts}, p \in R(t')} M_{ts}(p)), q = W(p,t)$ if $(p,t) \in E^-$ or $R(t)$ otherwise$), 0 \leq ts \leq k\}$
    per the maximal firing set semantics

We show that A satisfies (A.15) and (A.16), and A is an answer set of Π^7.

A satisfies (A.15) and (A.16) by its construction above. We show A is an answer set of Π^7 by splitting. We split $\mathit{lit}(\Pi^7)$ into a sequence of $9k+11$ sets:

- $U_0 = \mathit{head}(f1) \cup \mathit{head}(f2) \cup \mathit{head}(f13) \cup \mathit{head}(f5) \cup \mathit{head}(f6) \cup \mathit{head}(i2) \cup \mathit{head}(f19) = \{\mathit{place}(p) : p \in P\} \cup \{\mathit{trans}(t) : t \in T\} \cup \{\mathit{col}(c) : c \in C\} \cup \{\mathit{time}(0), \ldots, \mathit{time}(k)\} \cup \{\mathit{num}(0), \ldots, \mathit{num}(\mathit{ntok})\} \cup \{\mathit{holds}(p, q_c, c, 0) : p \in P, c \in C, q_c = m_{M_0(p)}(c)\} \cup \{\mathit{transpr}(t, pr) : t \in T, pr = Z(t)\}$

- $U_{9k+1} = U_{9k+0} \cup \mathit{head}(f14)^{ts=k} \cup \mathit{head}(f15)^{ts=k} \cup \mathit{head}(f16)^{ts=k} \cup \mathit{head}(f17)^{ts=k} \cup \mathit{head}(f18)^{ts=k} = U_{9k+0} \cup \{\mathit{ptarc}(p,t,n_c,c,k) : (p,t) \in E^-, c \in C, n_c = m_{W(p,t)}(c)\} \cup \{\mathit{tparc}(t,p,n_c,c,k,d) : (t,p) \in E^+, c \in C, n_c = m_{W(t,p)}(c), d = D(t)\} \cup \{\mathit{ptarc}(p,t,n_c,c,k) : p \in R(t), c \in C, n_c = m_{M_k(p)}(c), n > 0\} \cup \{\mathit{iptarc}(p,t,1,c,k) :$

  $p \in I(t), c \in C\} \cup \{\mathit{tptarc}(p,t,n_c,c,k) : (p,t) \in Q, c \in C, n_c = m_{QW(p,t)}(c)\}$

- $U_{9k+2} = U_{9k+1} \cup \mathit{head}(e6)^{ts=k} \cup \mathit{head}(e7)^{ts=k} \cup \mathit{head}(e8)^{ts=k} = U_{10k+1} \cup \{\mathit{notenabled}(t,k) : t \in T\}$

- $U_{9k+3} = U_{9k+2} \cup \mathit{head}(e9)^{ts=k} = U_{9k+2} \cup \{\mathit{enabled}(t,k) : t \in T\}$

- $U_{9k+4} = U_{9k+3} \cup \mathit{head}(a11)^{ts=k} = U_{9k+3} \cup \{\mathit{notprenabled}(t,k) : t \in T\}$

- $U_{9k+5} = U_{9k+4} \cup \mathit{head}(a12)^{ts=k} = U_{9k+4} \cup \{\mathit{prenabled}(t,k) : t \in T\}$

- $U_{9k+6} = U_{9k+5} \cup \mathit{head}(a13)^{ts=k} = U_{9k+5} \cup \{\mathit{fires}(t,k) : t \in T\}$

- $U_{9k+7} = U_{9k+6} \cup \mathit{head}(r13)^{ts=k} \cup \mathit{head}(r9)^{ts=k} = U_{9k+6} \cup \{\mathit{add}(p,q_c,t,c,k) : p \in P, t \in T, c \in C, q_c = m_{W(t,p)}(c)\} \cup \{\mathit{del}(p,q_c,t,c,k) : p \in P, t \in T, c \in C, q_c = m_{W(p,t)}(c)\} \cup \{\mathit{del}(p,q_c,t,c,k) : p \in P, t \in T, c \in C, q_c = m_{M_k(p)}(c)\}$

- $U_{9k+8} = U_{9k+7} \cup \mathit{head}(r10)^{ts=k} \cup \mathit{head}(r11)^{ts=k} = U_{9k+7} \cup \{\mathit{tot\_incr}(p,q_c,c,k) : p \in P, c \in C, 0 \leq q_c \leq \mathit{ntok}\} \cup \{\mathit{tot\_decr}(p,q_c,c,k) : p \in P, c \in C, 0 \leq q_c \leq \mathit{ntok}\}$

- $U_{9k+9} = U_{9k+8} \cup \mathit{head}(a9)^{ts=k} \cup \mathit{head}(r12)^{ts=k} \cup \mathit{head}(a14)^{ts=k} = U_{9k+8} \cup \{\mathit{consumesmore}(p,k) : p \in P\} \cup \{\mathit{holds}(p,q,k+1) : p \in P, 0 \leq q \leq \mathit{ntok}\} \cup \{\mathit{could\_not\_have}(t,k) : t \in T\}$

- $U_{9k+10} = U_{9k+9} \cup \mathit{head}(a3) = U_{9k+9} \cup \{\mathit{consumesmore}\}$

where $\mathit{head}(r_i)^{ts=k}$ are head atoms of ground rule r_i in which $ts = k$. We write $A_i^{ts=k} = \{a(\ldots,ts) : a(\ldots,ts) \in A_i, ts = k\}$ as short hand for all atoms in A_i with $ts = k$. $U_\alpha, 0 \leq \alpha \leq 9k+10$ form a splitting sequence, since each U_i is a splitting set of Π^7, and $\langle U_\alpha \rangle_{\alpha<\mu}$ is a monotone continuous sequence, where $U_0 \subseteq U_1 \cdots \subseteq U_{9k+10}$ and $\bigcup_{\alpha<\mu} U_\alpha = \mathit{lit}(\Pi^7)$.

We compute the answer set of Π^7 using the splitting sets as follows:

1. $\mathit{bot}_{U_0}(\Pi^7) = f1 \cup f2 \cup f13 \cup f5 \cup f6 \cup i2 \cup f19$ and $X_0 = A_1 \cup \cdots \cup A_5 \cup A_8$ $(= U_0)$ is its answer set – using forced atom proposition

2. $eval_{U_0}(bot_{U_1}(\Pi^7)\backslash bot_{U_0}(\Pi^7), X_0) = \{ptarc(p,t,q_c,c,0)\text{:-}.|c \in C, q_c = m_{W(p,t)}(c), q_c > 0\} \cup$ $\{tparc(t,p,q_c,c,0,d)\text{:-}.|c \in C, q_c = m_{W(t,p)}(c), q_c > 0, d = D(t)\} \cup$ $\{ptarc(p,t,q_c,c,0)\text{:-}.|c \in C, q_c = m_{M_0(p)}(c), q_c > 0\} \cup$ $\{iptarc(p,t,1,c,0)\text{:-}.|c \in C\} \cup$ $\{tptarc(p,t,q_c,c,0)\text{:-}.|c \in C, q_c = m_{QW(p,t)}(c), q_c > 0\}$. Its answer set $X_1 = A_6^{ts=0} \cup A_7^{ts=0} \cup A_9^{ts=0} \cup A_{10}^{ts=0} \cup A_{11}^{ts=0}$ – using forced atom proposition and construction of $A_6, A_7, A_9, A_{10}, A_{11}$.

3. $eval_{U_1}(bot_{U_2}(\Pi^7) \setminus bot_{U_1}(\Pi^7), X_0 \cup X_1) = \{notenabled(t,0)\text{:-}.|(\{trans(t),$ $ptarc(p,t,n_c,c,0),$ $holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c < n_c)$ or $(\{notenabled(t,0)\text{:-}.|$ $(\{trans(t),$ $iptarc(p,t,n2_c,c,0),$ $holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c \geq n2_c\})$ or $(\{trans(t),$ $tptarc(p,t,n3_c,c,0),$ $holds(p,q_c,c,0)\} \subseteq X_0 \cup X_1,$ where $q_c < n3_c)\}$. Its answer set $X_2 = A_{12}^{ts=0}$ – using forced atom proposition and construction of A_{12}.

   (a) where, $q_c = m_{M_0(p)}(c)$, and $n_c = m_{W(p,t)}(c)$ for an arc $(p,t) \in E^-$ – by construction of $i2$ and $f14$ in Π^7, and

   (b) in an arc $(p,t) \in E^-$, $p \in \bullet t$ (by definition 31 of preset)

   (c) $n2_c = 1$ – by construction of $iptarc$ predicates in Π^7, meaning $q_c \geq n2_c \equiv q_c \geq 1 \equiv q_c > 0$,

   (d) $tptarc(p,t,n3_c,c,0)$ represents $n3_c = m_{QW(p,t)}(c)$, where $(p,t) \in Q$

   (e) thus, $notenabled(t,0) \in X_1$ means $\exists c \in C, (\exists p \in \bullet t : m_{M_0(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t) : m_{M_0(p)}(c) > 0) \vee (\exists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c))$.

4. $eval_{U_2}(bot_{U_3}(\Pi^7) \setminus bot_{U_2}(\Pi^7), X_0 \cup \dots \cup X_2) = \{enabled(t,0)\text{:-}.|trans(t) \in X_0 \cup \dots \cup X_2, notenabled(t,0) \notin X_0 \cup \dots \cup X_2\}$. Its answer set is $X_3 = A_{13}^{ts=0}$ – using forced atom proposition and construction of A_{13}.

   (a) since an $enabled(t,0) \in X_3$ if $\nexists$ $notenabled(t,0) \in X_0 \cup \dots \cup X_2$; which is equivalent to $\nexists t, \forall c \in C, (\nexists p \in \bullet t, m_{M_0(p)}(c) < m_{W(p,t)}(c)), (\nexists p \in I(t),$ $m_{M_0(p)}(c) > 0), (\nexists (p,t) \in Q : m_{M_0(p)}(c) < m_{QW(p,t)}(c)), \forall c \in C, (\forall p \in \bullet t :$ $m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t) : m_{M_0(p)}(c) = 0)$.

5. $eval_{U_3}(bot_{U_4}(\Pi^7) \setminus bot_{U_3}(\Pi^7), X_0 \cup \dots \cup X_3) = \{notprenabled(t,0)\text{:-}.|$ $\{enabled(t,0), transpr(t,p), enabled(tt,0), transpr(tt,pp)\} \subseteq X_0 \cup \dots \cup X_3, pp < p\}$. Its answer set is $X_4 = A_{23}^{ts=k}$ – using forced atom proposition and construction of A_{23}.

   (a) $enabled(t,0)$ represents $\exists t \in T, \forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)),$ $(\forall p \in I(t), m_{M_0(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c))$

   (b) $enabled(tt,0)$ represents $\exists tt \in T, \forall c \in C,$ $(\forall pp \in \bullet tt, m_{M_0(pp)}(c) \geq$ $m_{W(pp,tt)}(c)) \wedge$ $(\forall pp \in I(tt), m_{M_0(pp)}(c) = 0), (\forall (pp,tt) \in Q, m_{M_0(pp)}(c) \geq$ $m_{QW(pp,tt)}(c))$

   (c) $transpr(t,p)$ represents $p = Z(t)$ – by construction

   (d) $transpr(tt,pp)$ represents $pp = Z(tt)$ – by construction

(e) thus, $\mathit{notprenabled}(t,0)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_0(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c)), \exists tt \in T, (\forall pp \in \bullet tt, m_{M_0(pp)}(c) \geq m_{W(pp,tt)}(c)), (\forall pp \in I(tt), m_{M_0(pp)}(c) = 0), (\forall (pp,tt) \in Q, m_{M_0(pp)}(c) \geq m_{QW(pp,tt)}(c)), Z(tt) < Z(t)$

(f) which is equivalent to $(\forall p \in \bullet t : M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = 0), (\forall (pp,tt) \in Q, M_0 \geq QW(pp,tt)), Z(tt) < Z(t)$ – assuming multiset domain C for all operations

6. $\mathit{eval}_{U_4}(\mathit{bot}_{U_5}(\Pi^7) \setminus \mathit{bot}_{U_4}(\Pi^7), X_0 \cup \cdots \cup X_4) = \{\mathit{prenabled}(t,0){:}\text{-}.|\mathit{enabled}(t,0) \in X_0 \cup \cdots \cup X_4, \mathit{notprenabled}(t,0) \notin X_0 \cup \cdots \cup X_4\}$. Its answer set is $X_5 = A_{24}^{ts=k}$ – using forced atom proposition and construction of A_{24}

   (a) $\mathit{enabled}(t,0)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_0(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_0(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t)}(c)) \equiv (\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t))$ – from 4 above and assuming multiset domain C for all operations

   (b) $\mathit{notprenabled}(t,0)$ represents $(\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = 0), (\forall (pp,tt) \in Q, M_0(pp) \geq QW(pp,tt)), Z(tt) < Z(t)$ – from 5 above and assuming multiset domain C for all operations

   (c) then, $\mathit{prenabled}(t,0)$ represents $(\forall p \in \bullet t, M_0(p) \geq W(p,t)), (\forall p \in I(t), M_0(p) = 0), (\forall (p,t) \in Q, M_0(p) \geq QW(p,t)), \nexists tt \in T, ((\forall pp \in \bullet tt, M_0(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_0(pp) = 0), (\forall (pp,tt) \in Q, M_0(pp) \geq QW(pp,tt)), Z(tt) < Z(t))$ – from (a), (b) and $\mathit{enabled}(t,0) \in X_0 \cup \cdots \cup X_4$

7. $\mathit{eval}_{U_5}(\mathit{bot}_{U_6}(\Pi^7) \setminus \mathit{bot}_{U_5}(\Pi^7), X_0 \cup \cdots \cup X_5) = \{\{\mathit{fires}(t,0)\}{:}\text{-}.|\mathit{prenabled}(t,0)$ holds in $X_0 \cup \cdots \cup X_5\}$. It has multiple answer sets $X_{6.1}, \ldots, X_{6.n}$, corresponding to elements of power set of $fires(t,0)$ atoms in $\mathit{eval}_{U_5}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^7 determined using splitting is equal to A, we only consider the set that matches the $fires(t,0)$ elements in A and call it X_6, ignoring the rest. Thus, $X_6 = A_{14}^{ts=0}$, representing T_0.

   (a) in addition, for every t such that $\mathit{prenabled}(t,0) \in X_0 \cup \cdots \cup X_5, R(t) \neq \emptyset$; $fires(t,0) \in X_6$ – per definition 41 (firing set); requiring that a reset transition is fired when enabled

   (b) thus, the firing set T_0 will not be eliminated by the constraint $f20$

8. $\mathit{eval}_{U_6}(\mathit{bot}_{U_7}(\Pi^7) \setminus \mathit{bot}_{U_6}(\Pi^7), X_0 \cup \cdots \cup X_6) = \{\mathit{add}(p,n_c,t,c,0){:}\text{-}.|\ \{fires(t,0-d+1), \mathit{tparc}(t,p,n_c,c,0,d)\} \subseteq X_0 \cup \cdots \cup X_6\} \cup \{\mathit{del}(p,n_c,t,c,0){:}\text{-}.|\ \{fires(t,0), \mathit{ptarc}(p,t,n_c,c,0)\} \subseteq X_0 \cup \cdots \cup X_6\}$. It's answer set is $X_7 = A_{15}^{ts=0} \cup A_{16}^{ts=0}$ – using forced atom proposition and definitions of A_{15} and A_{16}.

   (a) where, each add atom is equivalent to $n_c = m_{W(t,p)}(c), c \in C, p \in t\bullet$,

(b) and each del atom is equivalent to $n_c = m_{W(p,t)}(c), c \in C, p \in \bullet t$; or $n_c = m_{M_0(p)}(c), c \in C, p \in R(t)$,

(c) representing the effect of transitions in T_0

9. $eval_{U_7}(bot_{U_8}(\Pi^7) \setminus bot_{U_7}(\Pi^7), X_0 \cup \cdots \cup X_7) =$ $\{tot\_incr(p, qq_c, c, 0)$:-.$| qq_c = \sum_{add(p,q_c,t,c,0) \in X_0 \cup \cdots \cup X_7} q_c\} \cup$ $\{tot\_decr(p, qq_c, c, 0)$:-.$| qq_c = \sum_{del(p,q_c,t,c,0) \in X_0 \cup \cdots \cup X_7} q_c\}$. It's answer set is $X_8 = A_{17}^{ts=0} \cup A_{18}^{ts=0}$ – using forced atom proposition and definitions of A_{17} and A_{18}.

   (a) where, each $tot\_incr(p, qq_c, c, 0)$, $qq_c = \sum_{add(p,q_c,t,c,0) \in X_0 \cup \dots X_7} q_c$ $\equiv qq_c = \sum_{t \in X_6, p \in t\bullet, 0+D(t)-1=0} m_{W(p,t)}(c)$,

   (b) $tot\_decr(p, qq_c, c, 0)$, $qq_c = \sum_{del(p,q_c,t,c,0) \in X_0 \cup \dots X_7} q_c$ $\equiv qq = \sum_{t \in X_6, p \in \bullet t} m_{W(t,p)}(c) + \sum_{t \in X_6, p \in R(t)} m_{M_0(p)}(c)$,

   (c) represent the net effect of transitions in T_0

10. $eval_{U_8}(bot_{U_9}(\Pi^7) \setminus bot_{U_8}(\Pi^7), X_0 \cup \cdots \cup X_8) = \{consumesmore(p, 0)$:-.$|$ $\{holds(p, q_c, c, 0), tot\_decr(p, q1_c, c, 0)\} \subseteq X_0 \cup \cdots \cup X_8, q1_c > q_c\} \cup$ $\{holds(p, q_c, c, 1)$:-.$|\{holds(p, q1_c, c, 0), tot\_incr(p, q2_c, c, 0),$ $tot\_decr(p, q3_c, c, 0)\} \subseteq X_0 \cup \cdots \cup X_6, q_c = q1_c + q2_c - q3_c\} \cup \{could\_not\_have(t, 0)$:-.$|$ $\{prenabled(t, 0), ptarc(s, t, q, c, 0), holds(s, qq, c, 0), tot\_decr(s, qqq, c, 0)\} \subseteq X_0 \cup$ $\cdots \cup X_8, fires(t, 0) \notin X_0 \cup \cdots \cup X_8, q > qq - qqq\}$. It's answer set is $X_9 = A_{19}^{ts=0} \cup$ $A_{21}^{ts=0} \cup A_{25}^{ts=0}$ – using forced atom proposition and definitions of A_{19}, A_{21}, A_{25}.

   (a) where $consumesmore(p, 0)$ represents $\exists p : q_c = m_{M_0(p)}(c), q1_c = \sum_{t \in T_0, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_0, p \in R(t)} m_{M_0(p)}(c), q1_c > q_c, c \in C$, indicating place p will be over consumed if T_0 is fired, as defined in definition 34 (conflicting transitions),

   (b) $holds(p, q_c, c, 1)$ if $q_c = m_{M_0(p)}(c) + \sum_{t \in T_0, p \in t\bullet, 0+D(t)-1=0} m_{W(t,p)}(c) -$ $(\sum_{t \in T_0, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_0, p \in R(t)} m_{M_0(p)}(c))$ represented by $q_c = m_{M_1(p)}(c)$ for some $c \in C$ – by construction of Π^7

   (c) and $consumesmore(p, 0)$ if $\sum_{t \in T_0, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_0, p \in R(t)} m_{M_0(p)}(c) >$ $m_{M_0(p)}(c)$ for any $c \in C$

   (d) $could\_not\_have(t, 0)$ if

      i. $(\forall p \in \bullet t, W(p, t) \leq M_0(p)), (\forall p \in I(t), M_0(p) = 0), (\forall (p, t) \in Q,$ $M_0(p) \geq WQ(p, t)), \nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in$ $I(tt), M_0(pp) = 0), (\forall (pp, tt) \in Q, M_0(pp) \geq QW(pp, tt)), Z(tt) <$ $Z(t)$,

      ii. and $q_c > m_{M_0(s)}(c) - (\sum_{t' \in T_0, s \in \bullet t'} m_{W(s,t')}(c) + \sum_{t' \in T_0, s \in R(t)} m_{M_0(s)}(c))$, $q_c = m_{W(s,t)}(c)$ if $s \in \bullet t$ or $m_{M_0(s)}(c)$ otherwise for some $c \in C$, which becomes $q > M_0(s) - (\sum_{t' \in T_0, s \in \bullet t'} W(s, t') + \sum_{t' \in T_0, s \in R(t)} M_0(s)), q =$ $W(s, t)$ if $s \in \bullet t$ or $M_0(s)$ otherwise for all $c \in C$

      iii. (i), (ii) above combined match the definition of A_{25}

(e) X_9 does not contain $could\_not\_have(t, 0)$, when $prenabled(t, 0) \in X_0 \cup \cdots \cup X_6$ and $fires(t, 0) \notin X_0 \cup \cdots \cup X_5$ due to construction of A, encoding of $a10$ and its body atoms. As a result, it is not eliminated by the constraint $a15$

$$\vdots$$

11. $eval_{U_{9k+0}}(bot_{U_{9k+1}}(\Pi^7) \setminus bot_{U_{9k+0}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+0}) = \{ptarc(p, t, q_c, c, k)\text{:-.}|c \in C, q_c = m_{W(p,t)}(c), q_c > 0\} \cup \{tparc(t, p, q_c, c, k, d)\text{:-.}|c \in C, q_c = m_{W(t,p)}(c), q_c > 0, d = D(t)\} \cup \{ptarc(p, t, q_c, c, k)\text{:-.}|c \in C, q_c = m_{M_k(p)}(c), q_c > 0\} \cup \{iptarc(p, t, 1, c, k)\text{:-.}|c \in C\} \cup \{tptarc(p, t, q_c, c, k)\text{:-.}|c \in C, q_c = m_{QW(p,t)}(c), q_c > 0\}$. Its answer set $X_{9k+1} = A_6^{ts=k} \cup A_7^{ts=k} \cup A_9^{ts=k} \cup A_{10}^{ts=k} \cup A_{11}^{ts=k}$ – using forced atom proposition and construction of $A_6, A_7, A_9, A_{10}, A_{11}$.

12. $eval_{U_{9k+1}}(bot_{U_{10k+2}}(\Pi^7) \setminus bot_{U_{9k+1}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+1}) = \{notenabled(t, k)\text{:-.}| (\{trans(t), ptarc(p, t, n_c, c, k), holds(p, q_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+1}, \text{ where } q_c < n_c)$ or $(\{notenabled(t, k)\text{:-.}|(\{trans(t), iptarc(p, t, n2_c, c, k), holds(p, q_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+1}, \text{ where } q_c \geq n2_c\})$ or $(\{trans(t), tptarc(p, t, n3_c, c, k), holds(p, q_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+1}, \text{ where } q_c < n3_c)\}$. Its answer set $X_{9k+2} = A_{12}^{ts=k}$ – using forced atom proposition and construction of A_{12}.

    (a) where, $q_c = m_{M_k(p)}(c)$, and $n_c = m_{W(p,t)}(c)$ for an arc $(p, t) \in E^-$ – by construction of $i2$ and $f14$ predicates in Π^7, and

    (b) in an arc $(p, t) \in E^-$, $p \in \bullet t$ (by definition 31 of preset)

    (c) $n2_c = 1$ – by construction of $iptarc$ predicates in Π^7, meaning $q_c \geq n2_c \equiv q_c \geq 1 \equiv q_c > 0$,

    (d) $tptarc(p, t, n3_c, c, k)$ represents $n3_c = m_{QW(p,t)}(c)$, where $(p, t) \in Q$

    (e) thus, $notenabled(t, k) \in X_{9k+1}$ represents $\exists c \in C, (\exists p \in \bullet t : m_{M_k(p)}(c) < m_{W(p,t)}(c)) \vee (\exists p \in I(t) : m_{M_k(p)}(c) > 0) \vee (\exists (p, t) \in Q : m_{M_k(p)}(c) < m_{QW(p,t)}(c))$.

13. $eval_{U_{9k+2}}(bot_{U_{9k+3}}(\Pi^7) \setminus bot_{U_{9k+2}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+2}) = \{enabled(t, k)\text{:-.}|trans(t) \in X_0 \cup \cdots \cup X_{9k+2}, notenabled(t, k) \notin X_0 \cup \cdots \cup X_{9k+2}\}$. Its answer set is $X_{9k+3} = A_{13}^{ts=k}$ – using forced atom proposition and construction of A_{13}.

    (a) Since an $enabled(t, k) \in X_{9k+3}$ if $\nexists$ $notenabled(t, k) \in X_0 \cup \cdots \cup X_{9k+2}$; which is equivalent to $\nexists t, \forall c \in C, (\nexists p \in \bullet t, m_{M_k(p)}(c) < m_{W(p,t)}(c)), (\nexists p \in I(t), m_{M_k(p)}(c) > 0), (\nexists (p, t) \in Q : m_{M_k(p)}(c) < m_{QW(p,t)}(c)), \forall c \in C, (\forall p \in \bullet t : m_{M_k(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t) : m_{M_k(p)}(c) = 0)$.

14. $eval_{U_{9k+3}}(bot_{U_{9k+4}}(\Pi^7) \setminus bot_{U_{9k+3}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+3}) = \{notprenabled(t, k)\text{:-.}| \{enabled(t, k), transpr(t, p), enabled(tt, k), transpr(tt, pp)\} \subseteq X_0 \cup \cdots \cup X_{9k+3}, pp < p\}$. Its answer set is $X_{9k+4} = A_{23}^{ts=k}$ – using forced atom proposition and construction of A_{23}.

    (a) $enabled(t, k)$ represents $\exists t \in T, \forall c \in C, (\forall p \in \bullet t, m_{M_k(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_k(p)}(c) = 0), (\forall (p, t) \in Q, m_{M_k(p)}(c) \geq m_{QW(p,t)}(c))$

   (b) $\mathit{enabled}(tt,k)$ represents $\exists tt \in T, \forall c \in C, (\forall pp \in \bullet tt, m_{M_k(pp)}(c) \geq m_{W(pp,tt)}(c)) \wedge (\forall pp \in I(tt), m_{M_k(pp)}(c) = 0), (\forall (pp,tt) \in Q, m_{M_k(pp)}(c) \geq m_{QW(pp,tt)}(c))$

   (c) $\mathit{transpr}(t,p)$ represents $p = Z(t)$ – by construction

   (d) $\mathit{transpr}(tt,pp)$ represents $pp = Z(tt)$ – by construction

   (e) thus, $\mathit{notprenabled}(t,k)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_k(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_k(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_k(p)}(c) \geq m_{QW(p,t)}(c)), \exists tt \in T, (\forall pp \in \bullet tt, m_{M_k(pp)}(c) \geq m_{W(pp,tt)}(c)), (\forall pp \in I(tt), m_{M_k(pp)}(c) = 0), (\forall (pp,tt) \in Q, m_{M_k(pp)}(c) \geq m_{QW(pp,tt)}(c)), Z(tt) < Z(t)$

   (f) which is equivalent to $(\forall p \in \bullet t : M_k(p) \geq W(p,t)), (\forall p \in I(t), M_k(p) = 0), (\forall (p,t) \in Q, M_k(p) \geq QW(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, M_k(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_k(pp) = 0), (\forall (pp,tt) \in Q, M_k(pp) \geq QW(pp,tt)), Z(tt) < Z(t)$ – assuming multiset domain C for all operations

15. $\mathit{eval}_{U_{9k+4}}(\mathit{bot}_{U_{9k+5}}(\Pi^7) \setminus \mathit{bot}_{U_{9k+4}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+4}) = \{\mathit{prenabled}(t,k){:}\text{-}.| \mathit{enabled}(t,k) \in X_0 \cup \cdots \cup X_{9k+4}, \mathit{notprenabled}(t,k) \notin X_0 \cup \cdots \cup X_{9k+4}\}$. Its answer set is $X_{9k+5} = A_{24}^{ts=k}$ – using forced atom proposition and construction of A_{24}

    (a) $\mathit{enabled}(t,k)$ represents $\forall c \in C, (\forall p \in \bullet t, m_{M_k(p)}(c) \geq m_{W(p,t)}(c)), (\forall p \in I(t), m_{M_k(p)}(c) = 0), (\forall (p,t) \in Q, m_{M_k(p)}(c) \geq m_{QW(p,t)}(c)) \equiv (\forall p \in \bullet t, M_k(p) \geq W(p,t)), (\forall p \in I(t), M_k(p) = 0), (\forall (p,t) \in Q, M_k(p) \geq QW(p,t))$ – from 13 above and assuming multiset domain C for all operations

    (b) $\mathit{notprenabled}(t,k)$ represents $(\forall p \in \bullet t, M_k(p) \geq W(p,t)), (\forall p \in I(t), M_k(p) = 0), (\forall (p,t) \in Q, M_k(p) \geq QW(p,t)), \exists tt \in T, (\forall pp \in \bullet tt, M_k(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_k(pp) = 0), (\forall (pp,tt) \in Q, M_k(pp) \geq QW(pp,tt)), Z(tt) < Z(t)$ – from 14 above and assuming multiset domain C for all operations

    (c) then, $\mathit{prenabled}(t,k)$ represents $(\forall p \in \bullet t, M_k(p) \geq W(p,t)), (\forall p \in I(t), M_k(p) = 0), (\forall (p,t) \in Q, M_k(p) \geq QW(p,t)), \nexists tt \in T, ((\forall pp \in \bullet tt, M_k(pp) \geq W(pp,tt)), (\forall pp \in I(tt), M_k(pp) = 0), (\forall (pp,tt) \in Q, M_k(pp) \geq QW(pp,tt)), Z(tt) < Z(t))$ – from (a), (b) and $\mathit{enabled}(t,k) \in X_0 \cup \cdots \cup X_{9k+4}$

16. $\mathit{eval}_{U_{9k+5}}(\mathit{bot}_{U_{9k+6}}(\Pi^7) \setminus \mathit{bot}_{U_{9k+5}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+5}) = \{\{\mathit{fires}(t,k)\}{:}\text{-}.| \mathit{prenabled}(t,k) \text{ holds in } X_0 \cup \cdots \cup X_{9k+5}\}$. It has multiple answer sets $X_{9k+6.1}, \ldots, X_{9k+6.n}$, corresponding to elements of power set of $\mathit{fires}(t,k)$ atoms in $\mathit{eval}_{U_{9k+5}}(...)$ – using supported rule proposition. Since we are showing that the union of answer sets of Π^7 determined using splitting is equal to A, we only consider the set that matches the $\mathit{fires}(t,k)$ elements in A and call it X_{9k+6}, ignoring the rest. Thus, $X_{9k+6} = A_{14}^{ts=k}$, representing T_k.

    (a) in addition, for every t such that $\mathit{prenabled}(t,k) \in X_0 \cup \cdots \cup X_{9k+5}, R(t) \neq \emptyset$; $\mathit{fires}(t,k) \in X_{9k+6}$ – per definition 41 (firing set); requiring that a reset transition is fired when enabled

   (b) thus, the firing set T_k will not be eliminated by the constraint $f20$

17. $eval_{U_{9k+6}}(bot_{U_{9k+7}}(\Pi^7) \setminus bot_{U_{9k+6}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+6}) = \{add(p, n_c, t, c, k){:}\text{-}.| \{fires(t, k-d+1), tparc(t, p, n_c, c, 0, d)\} \subseteq X_0 \cup \cdots \cup X_{9k+6}\} \cup \{del(p, n_c, t, c, k){:}\text{-}.|\{fires(t, k), ptarc(p, t, n_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+6}\}$. It's answer set is $X_{9k+7} = A_{15}^{ts=k} \cup A_{16}^{ts=k}$ – using forced atom proposition and definitions of A_{15} and A_{16}.
    (a) where, each add atom is equivalent to $n_c = m_{W(t,p)}(c), c \in C, p \in t\bullet$,
    (b) and each del atom is equivalent to $n_c = m_{W(p,t)}(c), c \in C, p \in \bullet t$; or $n_c = m_{M_k(p)}(c), c \in C, p \in R(t)$,
    (c) representing the effect of transitions in T_k

18. $eval_{U_{9k+7}}(bot_{U_{9k+8}}(\Pi^7) \setminus bot_{U_{9k+7}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+7}) = \{tot\_incr(p, qq_c, c, k){:}\text{-}.| qq_c = \sum_{add(p,q_c,t,c,k) \in X_0 \cup \cdots \cup X_{9k+7}} q_c\} \cup \{tot\_decr(p, qq_c, c, k){:}\text{-}.| qq_c = \sum_{del(p,q_c,t,c,k) \in X_0 \cup \cdots \cup X_{9k+7}} q_c\}$. It's answer set is $X_{9k+8} = A_{17}^{ts=k} \cup A_{18}^{ts=k}$ – using forced atom proposition and definitions of A_{17} and A_{18}.
    (a) where, each $tot\_incr(p, qq_c, c, k)$, $qq_c = \sum_{add(p,q_c,t,c,k) \in X_0 \cup \ldots X_{9k+7}} q_c$ $\equiv qq_c = \sum_{t \in X_{9k+6}, p \in t\bullet, 0 \leq l \leq k, l+D(t)-1=k} m_{W(p,t)}(c)$,
    (b) and each $tot\_decr(p, qq_c, c, k)$, $qq_c = \sum_{del(p,q_c,t,c,k) \in X_0 \cup \ldots X_{9k+7}} q_c$ $\equiv qq = \sum_{t \in X_{9k+6}, p \in \bullet t} m_{W(t,p)}(c) + \sum_{t \in X_{9k+6}, p \in R(t)} m_{M_k(p)}(c)$,
    (c) represent the net effect of transitions in T_k

19. $eval_{U_{9k+8}}(bot_{U_{9k+9}}(\Pi^7) \setminus bot_{U_{9k+8}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+8}) = \{consumesmore(p, k){:}\text{-}.| \{holds(p, q_c, c, k), tot\_decr(p, q1_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+8}, q1_c > q_c\} \cup \{holds(p, q_c, c, k+1){:}\text{-}., |\{holds(p, q1_c, c, k), tot\_incr(p, q2_c, c, k), tot\_decr(p, q3_c, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+6}, q_c = q1_c + q2_c - q3_c\} \cup \{could\_not\_have(t, k){:}\text{-}.|\{prenabled(t, k), ptarc(s, t, q, c, k), holds(s, qq, c, k), tot\_decr(s, qqq, c, k)\} \subseteq X_0 \cup \cdots \cup X_{9k+8}, fires(t, k) \notin X_0 \cup \cdots \cup X_{10k+8}, q > qq - qqq\}$. It's answer set is $X_{9k+9} = A_{19}^{ts=k} \cup A_{21}^{ts=k} \cup A_{25}^{ts=k}$ – using forced atom proposition and definitions of A_{19}, A_{21}, A_{25}.
    (a) where, $consumesmore(p, k)$ represents $\exists p : q_c = m_{M_k(p)}(c)$, $q1_c = \sum_{t \in T_k, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_k, p \in R(t)} m_{M_k(p)}(c), q1_c > q_c, c \in C$, indicating place p that will be over consumed if T_k is fired, as defined in definition 34 (conflicting transitions),
    (b) $holds(p, q_c, c, k+1)$ if $q_c = m_{M_k(p)}(c) + \sum_{t \in T_l, p \in t\bullet, 0 \leq l \leq k, l+D(t)-1=k} m_{W(t,p)}(c) - (\sum_{t \in T_k, p \in \bullet t} m_{W(p,t)}(c) + \sum_{t \in T_k, p \in R(t)} m_{M_k(p)}(c))$ represented by $q_c = m_{M_1(p)}(c)$ for some $c \in C$ – by construction of Π^7,
    (c) and $could\_not\_have(t, k)$ if
        i. $(\forall p \in \bullet t, W(p,t) \leq M_k(p)), (\forall p \in I(t), M_k(p) = 0), (\forall (p,t) \in Q, M_k(p) \geq WQ(p,t)), \nexists tt \in T, (\forall pp \in \bullet tt, W(pp, tt) \leq M_{ts}(pp)), (\forall pp \in I(tt), M_k(pp) = 0), (\forall (pp, tt) \in Q, M_k(pp) \geq QW(pp, tt)), Z(tt) < Z(t)$,

ii. and $q_c > m_{M_k(s)}(c) - (\sum_{t' \in T_k, s \in \bullet t'} m_{W(s,t')}(c) + \sum_{t' \in T_k, s \in R(t)} m_{M_k(s)}(c))$, $q_c = m_{W(s,t)}(c)$ if $s \in \bullet t$ or $m_{M_k(s)}(c)$ otherwise for some $c \in C$, which becomes $q > M_k(s) - (\sum_{t' \in T_k, s \in \bullet t'} W(s,t') + \sum_{t' \in T_k, s \in R(t)} M_k(s)), q = W(s,t)$ if $s \in \bullet t$ or $M_k(s)$ otherwise for all $c \in C$

iii. (i), (ii) above combined match the definition of A_{25}

(d) X_{9k+9} does not contain $could\_not\_have(t,k)$, when $prenabled(t,k) \in X_0 \cup \cdots \cup X_{9k+6}$ and $fires(t,k) \notin X_0 \cup \cdots \cup X_{9k+5}$ due to construction of A, encoding of $a10$ and its body atoms. As a result it is not eliminated by the constraint $a15$

20. $eval_{U_{9k+9}}(bot_{U_{9k+10}}(\Pi^7) \setminus bot_{U_{9k+9}}(\Pi^7), X_0 \cup \cdots \cup X_{9k+9}) = \{consumesmore\text{:-}.| \{consumesmore(p,0), \ldots, consumesmore(p,k)\} \cap (X_0 \cup \cdots \cup X_{9k+9}) \neq \emptyset\}$. It's answer set is $X_{9k+10} = A_{20}$ – using forced atom proposition and the definition of A_{20}

    (a) X_{9k+10} will be empty since none of $consumesmore(p,0), \ldots, consumesmore(p,k)$ hold in $X_0 \cup \cdots \cup X_{9k+10}$ due to the construction of A, encoding of $a2$ and its body atoms. As a result, it is not eliminated by the constraint $a4$

The set $X = X_0 \cup \cdots \cup X_{9k+10}$ is the answer set of Π^7 by the splitting sequence theorem 15. Each $X_i, 0 \leq i \leq 9k+10$ matches a distinct portion of A, and $X = A$, thus A is an answer set of Π^7.

Next we show (II): Given Π^7 be the encoding of a Petri Net $PN(P,T,E,C,W,R,I,Q,QW,Z,D)$ with initial marking M_0, and A be an answer set of Π^7 that satisfies (A.15) and (A.16), then we can construct $X = M_0, T_k, \ldots, M_k, T_k, M_{k+1}$ from A, such that it is an execution sequence of PN.

We construct the X as follows:

1. $M_i = (M_i(p_0), \ldots, M_i(p_n))$, where $\{holds(p_0, m_{M_i(p_0)}(c), c, i), \ldots, holds(p_n, m_{M_i(p_n)}(c), c, i)\} \subseteq A$, for $c \in C, 0 \leq i \leq k+1$
2. $T_i = \{t : fires(t,i) \in A\}$, for $0 \leq i \leq k$

and show that X is indeed an execution sequence of PN. We show this by induction over k (i.e. given M_k, T_k is a valid firing set and its firing produces marking M_{k+1}).

Base case: Let $k = 0$, and M_0 is a valid marking in X for PN, show (1) T_0 is a valid firing set for M_0, and (2) firing T_0 in M_0 produces marking M_1.

1. We show T_0 is a valid firing set for M_0. Let $\{fires(t_0,0), \ldots, fires(t_x,0)\}$ be the set of all $fires(\ldots,0)$ atoms in A,

    (a) Then for each $fires(t_i,0) \in A$

    i. $prenabled(t_i,0) \in A$ – from rule $a13$ and supported rule proposition

ii. Then $enabled(t_i, 0) \in A$ – from rule $a12$ and supported rule proposition

iii. And $notprenabled(t_i, 0) \notin A$ – from rule $a12$ and supported rule proposition

iv. For $enabled(t_i, 0) \in A$

A. $notenabled(t_i, 0) \notin A$ – from rule $e9$ and supported rule proposition

B. Then either of $body(e6)$, $body(e7)$, or $body(e8)$ must not hold in A for t_i – from rules $body(e6), body(e7), body(e8)$ and forced atom proposition

C. Then $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e6$ for all $\{holds(p, q_c, c, 0), ptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A$ – from $e6$, forced atom proposition, and given facts $(holds(p, q_c, c, 0) \in A, ptarc(p, t_i, n_{ic}, 0) \in A)$

D. And $q_c \not> n_{ic} \equiv q_c < n_{ic}$ in $e7$ for all $\{holds(p, q_c, c, 0), iptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A, n_{ic} = 1$; $q_c > n_{ic} \equiv q_c = 0$ – from $e7$, forced atom proposition, given facts $(holds(p, q_c, c, 0) \in A, iptarc(p, t_i, 1, c, 0) \in A)$, and q_c is a positive integer

E. And $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e8$ for all $\{holds(p, q_c, c, 0), tptarc(p, t_i, n_{ic}, c, 0)\} \subseteq A$ – from $e8$, forced atom proposition, and given facts

F. Then $\forall c \in C, (\forall p \in \bullet t_i, m_{M_0(p)}(c) \geq m_{W(p,t_i)}(c)) \wedge (\forall p \in I(t_i), m_{M_0(p)}(c) = 0) \wedge (\forall (p, t_i) \in Q, m_{M_0(p)}(c) \geq m_{QW(p,t_i)}(c))$ – from $i2, f14, f16$ construction, definition 31 of preset $\bullet t_i$ in PN, definition 40 of enabled transition in PN, and that the construction of reset arcs by $f16$ ensures $notenabled(t, 0)$ is never true for a reset arc, where $holds(p, q_c, c, 0) \in A$ represents $q_c = m_{M_0(p)}(c)$, $ptarc(p, t_i, n_{ic}, 0) \in A$ represents $n_{ic} = m_{W(p,t_i)}(c), n_{ic} = m_{M_0(p)}(c)$.

G. Which is equivalent to $(\forall p \in \bullet t_i, M_0(p) \geq W(p, t_i)) \wedge (\forall p \in I(t_i), M_0(p) = 0) \wedge (\forall (p, t_i) \in Q, M_0(p) \geq QW(p, t_i))$ – assuming multiset domain C

v. For $notprenabled(t_i, 0) \notin A$

A. Either $(\nexists enabled(tt, 0) \in A : pp < p_i)$ or $(\forall enabled(tt, 0) \in A : pp \not< p_i)$ where $pp = Z(tt), p_i = Z(t_i)$ – from rule $a11, f19$ and forced atom proposition

B. This matches the definition of an enabled priority transition

vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_0 – from definition 40 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^7 – from rule $a4$ and supported rule proposition

i. Then $\nexists consumesmore(p, 0) \in A$ – from rule $a3$ and supported rule proposition

ii. Then $\nexists \{holds(p, q_c, c, 0), tot\_decr(p, q1_c, c, 0)\} \subseteq A, q1_c > q_c$ in $body(a9)$ – from $a9$ and forced atom proposition

iii. Then $\nexists c \in C \nexists p \in P, (\sum_{t_i \in \{t_0,\ldots,t_x\}, p \in \bullet t_i} m_{W(p,t_i)}(c) + \sum_{t_i \in \{t_0,\ldots,t_x\}, p \in R(t_i)} m_{M_0(p)}(c)) > m_{M_0(p)}(c)$ – from the following

A. $holds(p, q_c, c, 0)$ represents $q_c = m_{M_0(p)}(c)$ – from rule $i2$ encoding, given

B. $tot\_decr(p, q1_c, c, 0) \in A$ if $\{del(p, q1_{0_c}, t_0, c, 0), \dots, del(p, q1_{x_c}, t_x, c, 0)\} \subseteq A$, where $q1_c = q1_{0_c} + \cdots + q1_{x_c}$ – from $r11$ and forced atom proposition

C. $del(p, q1_{i_c}, t_i, c, 0) \in A$ if $\{fires(t_i, 0), ptarc(p, t_i, q1_{i_c}, c, 0)\} \subseteq A$ – from $r9$ and supported rule proposition

D. $del(p, q1_{i_c}, t_i, c, 0)$ represents removal of $q1_{i_c} = m_{W(p,t_i)}(c)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_{i_c} = m_{M_0(p)}(c)$ tokens from $p \in R(t_i)$– from rules $r9, f14, f16$, supported rule proposition, and definition 31 of preset in PN

iv. Then the set of transitions in T_0 do not conflict – by the definition 34 of conflicting transitions

(c) And for each $prenabled(t_j, 0) \in A$ and $fires(t_j, 0) \notin A$, $could\_not\_have(t_j, 0) \in A$, since A is an answer set of Π^7 - from rule $a15$ and supported rule proposition

i. Then $\{prenabled(t_j, 0), holds(s, qq_c, c, 0), ptarc(s, t_j, q_c, c, 0), tot\_decr(s, qqq_c, c, 0)\} \subseteq A$, such that $q_c > qq_c - qqq_c$ and $fires(t_j, 0) \notin A$ - from rule $a14$ and supported rule proposition

ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q_c > m_{M_0(s)}(c) - (\sum_{t_i \in T_0, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_0, s \in R(t_i)} m_{M_0(s)}(c))$, where $q_c = m_{W(s,t_j)}(c)$ if $s \in \bullet t_j$, or $m_{M_0(s)}(c)$ otherwise.

A. $ptarc(s, t_i, q_c, c, 0)$ represents $q_c = m_{W(s,t_i)}(c)$ if $(s, t_i) \in E^-$ or $q_c = m_{M_0(s)}(c)$ if $s \in R(t_i)$ – from rule $f14, f16$ construction

B. $holds(s, qq_c, c, 0)$ represents $qq_c = m_{M_0(s)}(c)$ – from $i2$ construction

C. $tot\_decr(s, qqq_c, c, 0) \in A$ if $\{del(s, qqq_{0_c}, t_0, c, 0), \dots, del(s, qqq_{x_c}, t_x, c, 0)\} \subseteq A$ – from rule $r11$ construction and supported rule proposition

D. $del(s, qqq_{i_c}, t_i, c, 0) \in A$ if $\{fires(t_i, 0), ptarc(s, t_i, qqq_{i_c}, c, 0)\} \subseteq A$ – from rule $r9$ and supported rule proposition

E. $del(s, qqq_{i_c}, t_i, c, 0)$ either represents $qqq_{i_c} = m_{W(s,t_i)}(c) : t_i \in T_0, (s, t_i) \in E^-$, or $qqq_{i_c} = m_{M_0(t_i)}(c) : t_i \in T_0, s \in R(t_i)$ – from rule $f14, f16$ construction

F. $tot\_decr(q, qqq_c, c, 0)$ represents $\sum_{t_i \in T_0, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_0, s \in R(t_i)} m_{M_0(s)}(c)$ – from (C,D,E) above

iii. Then firing $T_0 \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_0 is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $prenabled(t_r, 0) \in A$, $fires(t_r, 0) \in A$, since A is an answer set of Π^7 - from rule $f20$ and supported rule proposition

i. Then, the firing set T_0 satisfies the reset-transition requirement of definition 41 (firing set)

(e) Then $\{t_0, \dots, t_x\} = T_0$ – using 1(a),1(b),1(d) above; and using 1(c) it is a maximal firing set

2. Let $\mathit{holds}(p, q_c, c, 1) \in A$

   (a) Then $\{\mathit{holds}(p, q1_c, c, 0), \mathit{tot\_incr}(p, q2_c, c, 0), \mathit{tot\_decr}(p, q3_c, c, 0)\} \subseteq A$: $q_c = q1_c + q2_c - q3_c$ – from rule $r12$ and supported rule proposition

   (b) Then, $\mathit{holds}(p, q1_c, c, 0) \in A$ represents $q1_c = m_{M_0(p)}(c)$ – given ; and $\{\mathit{add}(p, q2_{0_c}, t_0, c, 0), \dots, \mathit{add}(p, q2_{j_c}, t_j, c, 0)\} \subseteq A : q2_{0_c} + \cdots + q2_{j_c} = q2_c$ and $\{\mathit{del}(p, q3_{0_c}, t_0, c, 0), \dots, \mathit{del}(p, q3_{l_c}, t_l, c, 0)\} \subseteq A : q3_{0_c} + \cdots + q3_{l_c} = q3_c$ – rules $r10, r11$ and supported rule proposition, respectively

   (c) Then $\{fires(t_0, 0), \dots, fires(t_j, 0)\} \subseteq A$ and $\{fires(t_0, 0), \dots, fires(t_l, 0)\} \subseteq A$ – rules $r13, r9$ and supported rule proposition, respectively

   (d) Then $\{fires(t_0, 0), \dots, fires(t_j, 0)\} \cup \{fires(t_0, 0), \dots, fires(t_l, 0)\} \subseteq A = \{fires(t_0, 0), \dots, fires(t_x, 0)\} \subseteq A$ – set union of subsets

   (e) Then for each $fires(t_x, 0) \in A$ we have $t_x \in T_0$ – already shown in item 1 above

   (f) Then $q_c = m_{M_0(p)}(c) + \sum_{t_x \in T_0, p \in t_x \bullet, 0 + D(t_x) - 1 = 0} m_{W(t_x, p)}(c) - (\sum_{t_x \in T_0 \wedge p \in \bullet t_x} m_{W(p, t_x)}(c) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} m_{M_0(p)}(c))$ – from (2b) above and the following

      i. Each $\mathit{add}(p, q_{j_c}, t_j, c, 0) \in A$ represents $q_{j_c} = m_{W(t_j, p)}(c)$ for $p \in t_j \bullet$ – rule $r13, f21$ encoding, and definition 43 of transition execution in PN

      ii. Each $\mathit{del}(p, t_y, q_{y_c}, c, 0) \in A$ represents either $q_{y_c} = m_{W(p, t_y)}(c)$ for $p \in \bullet t_y$, or $q_{y_c} = m_{M_0(p)}(c)$ for $p \in R(t_y)$ – from rule $r9, f14$ encoding and definition 43 of transition execution in PN; or from rule $r9, f16$ encoding and definition of reset arc in PN

      iii. Each $\mathit{tot\_incr}(p, q2_c, c, 0) \in A$ represents $q2_c = \sum_{t_x \in T_0 \wedge p \in t_x \bullet, 0 + D(t_x) - 1 = 0} m_{W(t_x, p)}(c)$ – aggregate assignment atom semantics in rule $r10$

      iv. Each $\mathit{tot\_decr}(p, q3_c, c, 0) \in A$ represents $q3_c = \sum_{t_x \in T_0 \wedge p \in \bullet t_x} m_{W(p, t_x)}(c) + \sum_{t_x \in T_0 \wedge p \in R(t_x)} m_{M_0(p)}(c)$ – aggregate assignment atom semantics in rule $r11$

   (g) Then, $m_{M_1(p)}(c) = q_c$ – since $\mathit{holds}(p, q_c, c, 1) \in A$ encodes $q_c = m_{M_1(p)}(c)$ – from construction

Inductive Step: Let $k > 0$, and M_k is a valid marking in X for PN, show (1) T_k is a valid firing set for M_k, and (2) firing T_k in M_k produces marking M_{k+1}.

1. We show that T_k is a valid firing set in M_k. Let $\{fires(t_0, k), \dots, fires(t_x, k)\}$ be the set of all $fires(\dots, k)$ atoms in A,

   (a) We show that T_k is a valid firing set in M_k. Then for each $fires(t_i, k) \in A$

      i. $\mathit{prenabled}(t_i, k) \in A$ – from rule $a13$ and supported rule proposition

ii. Then $enabled(t_i, k) \in A$ – from rule $a12$ and supported rule proposition

iii. And $notprenabled(t_i, k) \notin A$ – from rule $a12$ and supported rule proposition

iv. For $enabled(t_i, k) \in A$

A. $notenabled(t_i, k) \notin A$ – from rule $e9$ and supported rule proposition

B. Then either of $body(e6)$, $body(e7)$, or $body(e8)$ must not hold in A for t_i – from rules $body(e6), body(e7), body(e8)$ and forced atom proposition

C. Then $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e6$ for all $\{holds(p, q_c, c, k), ptarc(p, t_i, n_{ic}, c, k)\} \subseteq A$ – from $e6$, forced atom proposition, and given facts $(holds(p, q_c, c, k) \in A, ptarc(p, t_i, n_{ic}, k) \in A)$

D. And $q_c \not\geq n_{ic} \equiv q_c < n_{ic}$ in $e7$ for all $\{holds(p, q_c, c, k), iptarc(p, t_i, n_{ic}, c, k)\} \subseteq A, n_{ic} = 1$; $q_c > n_{ic} \equiv q_c = 0$ – from $e7$, forced atom proposition, given facts $(holds(p, q_c, c, k) \in A, iptarc(p, t_i, 1, c, k) \in A)$, and q_c is a positive integer

E. And $q_c \not< n_{ic} \equiv q_c \geq n_{ic}$ in $e8$ for all $\{holds(p, q_c, c, k), tptarc(p, t_i, n_{ic}, c, k)\} \subseteq A$ – from $e8$, forced atom proposition, and given facts

F. Then $\forall c \in C, (\forall p \in \bullet t_i, m_{M_k(p)}(c) \geq m_{W(p,t_i)}(c)) \wedge (\forall p \in I(t_i), m_{M_k(p)}(c) = 0) \wedge (\forall (p, t_i) \in Q, m_{M_k(p)}(c) \geq m_{QW(p,t_i)}(c))$ – from the inductive assumption, $f14, f16$ construction, definition 31 of preset $\bullet t_i$ in PN, definition 40 of enabled transition in PN, and that the construction of reset arcs by $f16$ ensures $notenabled(t, k)$ is never true for a reset arc, where $holds(p, q_c, c, k) \in A$ represents $q_c = m_{M_k(p)}(c)$, $ptarc(p, t_i, n_{ic}, k) \in A$ represents $n_{ic} = m_{W(p,t_i)}(c)$, $n_{ic} = m_{M_k(p)}(c)$.

G. Which is equivalent to $(\forall p \in \bullet t_i, M_k(p) \geq W(p, t_i)) \wedge (\forall p \in I(t_i), M_k(p) = 0) \wedge (\forall (p, t_i) \in Q, M_k(p) \geq QW(p, t_i))$ – assuming multiset domain C

v. For $notprenabled(t_i, k) \notin A$

A. Either $(\nexists enabled(tt, k) \in A : pp < p_i)$ or $(\forall enabled(tt, k) \in A : pp \not< p_i)$ where $pp = Z(tt), p_i = Z(t_i)$ – from rule $a11, f19$ and forced atom proposition

B. This matches the definition of an enabled priority transition

vi. Then t_i is enabled and can fire in PN, as a result it can belong to T_k – from definition 40 of enabled transition

(b) And $consumesmore \notin A$, since A is an answer set of Π^7 – from rule $a4$ and supported rule proposition

i. Then $\nexists consumesmore(p, k) \in A$ – from rule $a3$ and supported rule proposition

ii. Then $\nexists\{holds(p, q_c, c, k), tot\_decr(p, q1_c, c, k)\} \subseteq A, q1_c > q_c$ in $body(a9)$ – from $a9$ and forced atom proposition

iii. Then $\nexists c \in C \nexists p \in P, (\sum_{t_i \in \{t_0,\dots,t_x\}, p \in \bullet t_i} m_{W(p,t_i)}(c) + \sum_{t_i \in \{t_0,\dots,t_x\}, p \in R(t_i)} m_{M_k(p)}(c)) > m_{M_k(p)}(c)$ – from the following

  A. $\mathit{holds}(p, q_c, c, k)$ represents $q_c = m_{M_k(p)}(c)$ – from rule PN encoding, given

  B. $\mathit{tot\_decr}(p, q1_c, c, k) \in A$ if $\{\mathit{del}(p, q1_{0c}, t_0, c, k), \dots, \mathit{del}(p, q1_{xc}, t_x, c, k)\} \subseteq A$, where $q1_c = q1_{0c} + \cdots + q1_{xc}$ – from $r11$ and forced atom proposition

  C. $\mathit{del}(p, q1_{ic}, t_i, c, k) \in A$ if $\{\mathit{fires}(t_i, k), \mathit{ptarc}(p, t_i, q1_{ic}, c, k)\} \subseteq A$ – from $r9$ and supported rule proposition

  D. $\mathit{del}(p, q1_{ic}, t_i, c, k)$ either represents removal of $q1_{ic} = m_{W(p,t_i)}(c)$ tokens from $p \in \bullet t_i$; or it represents removal of $q1_{ic} = m_{M_k(p)}(c)$ tokens from $p \in R(t_i)$– from rules $r9, f14, f16$, supported rule proposition, and definition 43 of transition execution in PN

iv. Then the set of transitions in T_k do not conflict – by the definition 34 of conflicting transitions

(c) And for each $\mathit{prenabled}(t_j, k) \in A$ and $\mathit{fires}(t_j, k) \notin A$, $\mathit{could\_not\_have}(t_j, k) \in A$, since A is an answer set of Π^7 - from rule $a15$ and supported rule proposition

  i. Then $\{\mathit{prenabled}(t_j, k), \mathit{holds}(s, qq_c, c, k), \mathit{ptarc}(s, t_j, q_c, c, k), \mathit{tot\_decr}(s, qqq_c, c, k)\} \subseteq A$, such that $q_c > qq_c - qqq_c$ and $\mathit{fires}(t_j, k) \notin A$ - from rule $a14$ and supported rule proposition

  ii. Then for an $s \in \bullet t_j \cup R(t_j)$, $q_c > m_{M_k(s)}(c) - (\sum_{t_i \in T_k, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_k, s \in R(t_i)} m_{M_k(s)}(c))$, where $q_c = m_{W(s,t_j)}(c)$ if $s \in \bullet t_j$, or $m_{M_k(s)}(c)$ otherwise.

    A. $\mathit{ptarc}(s, t_i, q_c, c, k)$ represents $q_c = m_{W(s,t_i)}(c)$ if $(s, t_i) \in E^-$ or $q_c = m_{M_k(s)}(c)$ if $s \in R(t_i)$ – from rule $f14, f16$ construction

    B. $\mathit{holds}(s, qq_c, c, k)$ represents $qq_c = m_{M_k(s)}(c)$ – from $i2$ construction

    C. $\mathit{tot\_decr}(s, qqq_c, c, k) \in A$ if $\{\mathit{del}(s, qqq_{0c}, t_0, c, k), \dots, \mathit{del}(s, qqq_{xc}, t_x, c, k)\} \subseteq A$ – from rule $r11$ construction and supported rule proposition

    D. $\mathit{del}(s, qqq_{ic}, t_i, c, k) \in A$ if $\{\mathit{fires}(t_i, k), \mathit{ptarc}(s, t_i, qqq_{ic}, c, k)\} \subseteq A$ – from rule $r9$ and supported rule proposition

    E. $\mathit{del}(s, qqq_{ic}, t_i, c, k)$ represents $qqq_{ic} = m_{W(s,t_i)}(c) : t_i \in T_k, (s, t_i) \in E^-$, or $qqq_{ic} = m_{M_k(t_i)}(c) : t_i \in T_k, s \in R(t_i)$ – from rule $f14, f16$ construction

    F. $\mathit{tot\_decr}(q, qqq_c, c, k)$ represents $\sum_{t_i \in T_k, s \in \bullet t_i} m_{W(s,t_i)}(c) + \sum_{t_i \in T_k, s \in R(t_i)} m_{M_k(s)}(c)$ – from (C,D,E) above

  iii. Then firing $T_k \cup \{t_j\}$ would have required more tokens than are present at its source place $s \in \bullet t_j \cup R(t_j)$. Thus, T_k is a maximal set of transitions that can simultaneously fire.

(d) And for each reset transition t_r with $\mathit{enabled}(t_r, k) \in A$, $\mathit{fires}(t_r, k) \in A$, since A is an answer set of Π^7 - from rule $f20$ and supported rule proposition

   i. Then the firing set T_k satisfies the reset transition requirement of definition 41 (firing set)

(e) Then $\{t_0, \dots, t_x\} = T_k$ – using 1(a),1(b), 1(d) above; and using 1(c) it is a maximal firing set

2. We show that M_{k+1} is produced by firing T_k in M_k. Let $holds(p, q_c, c, k+1) \in A$

   (a) Then $\{holds(p, q1_c, c, k), tot\_incr(p, q2_c, c, k), tot\_decr(p, q3_c, c, k)\} \subseteq A : q_c = q1_c + q2_c - q3_c$ – from rule $r12$ and supported rule proposition

   (b) Then $holds(p, q1_c, c, k) \in A$ represents $q1_c = m_{M_k(p)}(c)$ – inductive assumption; and $\{add(p, q2_{0_c}, t_0, c, k), \dots, add(p, q2_{j_c}, t_j, c, k)\} \subseteq A : q2_{0_c} + \cdots + q2_{j_c} = q2_c$ and $\{del(p, q3_{0_c}, t_0, c, k), \dots, del(p, q3_{l_c}, t_l, c, k)\} \subseteq A : q3_{0_c} + \cdots + q3_{l_c} = q3_c$ – rules $r10, r11$ and supported rule proposition, respectively

   (c) Then $\{fires(t_0, k), \dots, fires(t_j, k)\} \subseteq A$ and $\{fires(t_0, k), \dots, fires(t_l, k)\} \subseteq A$ – rules $r13, r9$ and supported rule proposition, respectively

   (d) Then $\{fires(t_0, k), \dots, fires(t_j, k)\} \cup \{fires(t_0, k), \dots, fires(t_l, k)\} \subseteq A = \{fires(t_0, k), \dots, fires(t_x, k)\} \subseteq A$ – set union of subsets

   (e) Then for each $fires(t_x, k) \in A$ we have $t_x \in T_k$ – already shown in item 1 above

   (f) Then $q_c = m_{M_k(p)}(c) + \sum_{t_x \in T_l, p \in t_x \bullet, 0 \le l \le k, l + D(t_x) - 1 = k} m_{W(t_x, p)}(c) - (\sum_{t_x \in T_k \wedge p \in \bullet t_x} m_{W(p, t_x)}(c) + \sum_{t_x \in T_k \wedge p \in R(t_x)} m_{M_k(p)}(c))$ – from (2b) above and the following

      i. Each $add(p, q_{j_c}, t_j, c, k) \in A$ represents $q_{j_c} = m_{W(t_j, p)}(c)$ for $p \in t_j \bullet$ – rule $r13, f21$ encoding, and definition 31 of postset in PN

      ii. Each $del(p, t_y, q_{y_c}, c, k) \in A$ represents either $q_{y_c} = m_{W(p, t_y)}(c)$ for $p \in \bullet t_y$, or $q_{y_c} = m_{M_k(p)}(c)$ for $p \in R(t_y)$ – from rule $r9, f14$ encoding and definition 31 of preset in PN; or from rule $r9, f16$ encoding and definition of reset arc in PN

      iii. Each $tot\_incr(p, q2_c, c, k) \in A$ represents $q2_c = \sum_{t_x \in T_l, p \in t_x \bullet, 0 \le l \le k, l + D(t_x) - 1 = k} m_{W(t_x, p)}(c)$ – aggregate assignment atom semantics in rule $r10$

      iv. Each $tot\_decr(p, q3_c, c, k) \in A$ represents $q3_c = \sum_{t_x \in T_k \wedge p \in \bullet t_x} m_{W(p, t_x)}(c) + \sum_{t_x \in T_k \wedge p \in R(t_x)} m_{M_k(p)}(c)$ – aggregate assignment atom semantics in rule $r11$

   (g) Then, $m_{M_{k+1}(p)}(c) = q_c$ – since $holds(p, q_c, c, k+1) \in A$ encodes $q_c = m_{M_{k+1}(p)}(c)$ – from construction

As a result, for any $n > k$, T_n will be a valid firing set for M_n and M_{n+1} will be its target marking.

Conclusion: Since both (I) and (II) hold, $X = M_0, T_k, M_1, \dots, M_k, T_{k+1}$ is an execution sequence of $PN(P, T, E, C, W, R, I, Q, QW, Z, D)$ (w.r.t M_0) iff there is an answer set A of $\Pi^7(PN, M_0, k, ntok)$ such that (A.15) and (A.16) hold.

APPENDIX B

COMPLETE SET OF QUERIES USED FOR DRUG-DRUG INTERACTION

B.1 Drug Activates Gene

```
//S{/NP{//?[Value='activation'](kw1)=>//?[Value='of'](kw2)=>//?[Tag='
    GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='DRUG'](kw3)}} :::
    distinct sent.cid, kw3.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw3)=>/?[Value='activation'](kw1)}=>/PP{/?[Value
    ='of'](kw2)=>//?[Tag='GENE'](kw0)}} ::: distinct sent.cid, kw3.value
    , kw1.value, kw0.value, sent.value
//NP{/?[Value='activation'](kw1)=>/PP{/?[Value='of'](kw2)=>//?[Tag='GENE
    '](kw0)=>//?[Value='by'](kw4)=>//?[Tag='DRUG'](kw3)}} ::: distinct
    sent.cid, kw3.value, kw1.value, kw0.value, sent.value
//S{/NP{//?[Tag='GENE'](kw2)}=>/VP{//?[Value='activation'](kw1)=>//?[Tag
    ='DRUG'](kw0)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/NP{//?[Tag='DRUG'](kw0)}=>/VP{//?[Value='activation'](kw1)=>//?[Tag
    ='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw0)=>/?[Value='activation'](kw1)}=>/PP{//?[Tag
    ='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/NP{//?[Value='activation'](kw1)=>//?[Tag='GENE'](kw2)}=>/VP{//?[Tag
    ='DRUG'](kw0)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/NP{//?[Value='activation'](kw1)=>//?[Value='of'](kw2)=>//?[Tag='
    GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='DRUG'](kw3)}} :::
    distinct sent.cid, kw3.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw3)=>/?[Value='activation'](kw1)}=>/PP{/?[Value
    ='of'](kw2)=>//?[Tag='GENE'](kw0)}} ::: distinct sent.cid, kw3.value
    , kw1.value, kw0.value, sent.value
//NP{/?[Value='activation'](kw1)=>/PP{/?[Value='of'](kw2)=>//?[Tag='GENE
    '](kw0)=>//?[Value='by'](kw4)=>//?[Tag='DRUG'](kw3)}} ::: distinct
    sent.cid, kw3.value, kw1.value, kw0.value, sent.value
```

B.2 Gene Induces Gene

```
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='induced'](kw1)=>//?[Value
    ='by'](kw3)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value
    , kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{//?[Value='induced'](kw1)=>//?[Value='
    by'](kw3)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value,
    kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value IN {'increase','increased
    '}](kw1)=>//?[Tag='GENE'](kw2)=>//?[Value='activity'](kw3)}} :::
    distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)=>//?[Value='activity'](kw3)}=>/VP{//?[Value
    ='increase'](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0
    .value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value IN {'stimulates','stimulate
    ', 'stimulated'}](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid
    , kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'stimulates','stimulate
    '}](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value,
    kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value='stimulates'](kw1)=>//?[Tag='
    GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value
    , sent.value
```

```
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{/?[Value='activated'](kw1)=>//?[Tag='
    GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value
    , sent.value
//NP{/NP{/?[Value='induction'](kw1)}=>/PP{/?[Value='of'](kw3)=>/NP{//?[
    Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}} :::
    distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='stimulation'](kw1)}=>/PP{/?[Value='of'](kw3)=>/NP
    {//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}}
     ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='activation'](kw1)}=>/PP{/?[Value IN {'of'}](kw3)=>/NP
    {//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}}
     ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='inducible'](kw1)=>//?[Tag
    ='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='induced'](kw1)=>//?[Value
    ='by'](kw3)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value
    , kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{//?[Value='induced'](kw1)=>//?[Value='
    by'](kw3)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value,
    kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value IN {'increase','increased
    '}](kw1)=>//?[Tag='GENE'](kw2)=>//?[Value='activity'](kw3)}} :::
    distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)=>//?[Value='activity'](kw3)}=>/VP{//?[Value
    ='increase'](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0
    .value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='stimulated'](kw1)=>//?[Tag
    ='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'stimulates','stimulate
    '}](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value,
    kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value='stimulates'](kw1)=>//?[Tag='
    GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value
    , sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{/?[Value='activated'](kw1)=>//?[Tag='
    GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value
    , sent.value
//VP{/?[Value='activated'](kw1)=>/PP{//?[Tag='GENE'](kw0)=>//?[Tag='GENE
    '](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value,
    sent.value
//NP{/NP{/?[Value='induction'](kw1)}=>/PP{/?[Value='of'](kw3)=>/NP{//?[
    Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}} :::
    distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='stimulation'](kw1)}=>/PP{/?[Value='of'](kw3)=>/NP
    {//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}}
     ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='activation'](kw1)}=>/PP{/?[Value IN {'of'}](kw3)=>/NP
    {//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}}
     ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='inducible'](kw1)=>//?[Tag
    ='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
```

B.3 Gene Inhibits Gene

```
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='induced'](kw1)=>//?[Value='by'](kw3)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{//?[Value='induced'](kw1)=>//?[Value='by'](kw3)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value IN {'increase','increased'}](kw1)=>//?[Tag='GENE'](kw2)=>//?[Value='activity'](kw3)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)=>//?[Value='activity'](kw3)}=>/VP{//?[Value='increase'](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value IN {'stimulates','stimulate', 'stimulated'}](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'stimulates','stimulate'}](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value='stimulates'](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{/?[Value='activated'](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='induction'](kw1)}=>/PP{/?[Value='of'](kw3)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='stimulation'](kw1)}=>/PP{/?[Value='of'](kw3)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='activation'](kw1)}=>/PP{/?[Value IN {'of'}](kw3)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value='by'](kw4)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{//?[Tag='GENE'](kw0)}=>/VP{//?[Value='inducible'](kw1)=>//?[Tag='GENE'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

B.4 Drug Changes Gene Expression/Activity

```
//S{/?[Tag='DRUG'](kw2)=>/?[Value IN {'increased','increase','increases'}](kw1)=>/?[Value IN {'levels','level'}](kw3)=>/?[Tag='GENE'](kw0)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value IN {'increased','increases'}](kw1)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'expression','level','activity','activities','levels'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Value IN {'increases','increase'}](kw1)=>//?[Tag='GENE'](kw0)=>//?[Value IN {'levels','activity'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'increased','increases'}](kw1)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'activity','activities','levels'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value='increased'](kw1)=>/NP{//?[Value IN {'activity','levels','expression'}](kw3)=>//?[Tag='GENE'](
kw0)}}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value IN {'increased','increases','increase'}](kw1)=>//?[Tag='GENE'](kw0)=>//?[Value IN {'activity','levels'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value='increased'](kw1)=>//?[Tag='GENE'](kw0)=>//?[Value IN {'expression','activity'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'increased','increases'}](kw1)=>/NP{//?[Value IN {'expression','activity','activities','level','levels'}](kw3)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'decreased','decreases'}](kw1)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'activity','expression'}](kw3)}}} ::: distinct sent.cid, kw2.value, kw1,value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/?[Value IN {'increased','increase','increases'}](kw1)=>/?[Value IN {'levels','level'}](kw3)=>/?[Tag='GENE'](kw0)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value IN {'increased','increases'}](kw1)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'expression','level','activity','activities','levels'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Value IN {'increases','increase'}](kw1)=>//?[Tag='GENE'](kw0)=>//?[Value IN {'levels','activity'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'increased','increases'}](kw1)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'activity','activities','levels'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value='increased'](kw1)=>/NP{//?[Value IN {'activity','levels','expression'}](kw3)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value IN {'increased','increases','increase'}](kw1)=>//?[Tag='GENE'](kw0)=>//?[Value IN {'activity','levels'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value='increased'](kw1)=>//?[Tag='GENE'](kw0)=>//?[Value IN {'expression','activity'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'increased','increases'}](kw1)=>/NP{//?[Value IN {'expression','activity','activities','level','levels'}](kw3)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'decreased','decreases'}](kw1)=>/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'activity','expression'}](kw3)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
```

B.5 Drug Induces/Stimulates Gene

```
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//VP{/?[Value IN {'stimulated','induced'}](kw1)=>/PP{/NP{//?[Tag='DRUG'](kw2)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/PP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'stimulated','induced'}](kw1)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{/?[Value IN {'stimulated','induced'}](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='induced'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value IN {'stimulate','induce'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Value IN {'induction','stimulation'}](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Value IN {'stimulation','induction'}](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/VP{/VP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/NP{//?[Value IN {'stimulation','induction'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/NP{/?[Value='induction'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value IN {'stimulation','induction'}](kw1)}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/?[Value IN {'stimulation','induction'}](kw1)=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value IN {'stimulates','induces'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'stimulates','induces'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
//VP{/?[Value IN {'stimulated','induced'}](kw1)=>/PP{/NP{//?[Tag='DRUG'](kw2)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/PP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Tag='GENE'](kw0)=>//?[Value IN {'stimulated','induced'}](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{/?[Value IN {'stimulated','induced'}](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='induced'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'stimulated','induced'}](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value IN {'stimulate','induce'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value IN {'induction','stimulation'}](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Value IN {'stimulation','induction'}](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/VP{/VP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/NP{//?[Value IN {'stimulation','induction'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/NP{/?[Value='induction'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value IN {'stimulation','induction'}](kw1)}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Value IN {'stimulation','induction'}](kw1)=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value IN {'stimulates','induces'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value IN {'stimulates','induces'}](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

B.6 Drug Inhibits Gene

```
//NP{/NP{/NP{/?[Value='inhibitory'](kw1)}=>/PP{//?[Tag='DRUG'](kw2)}}=>/PP{/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/NP{/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)}}=>/?[Value='inhibitor'](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Value='inhibitor'](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Tag='GENE'](kw0)=>//?[Value='inhibitor'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Value='inhibitor'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Tag='GENE'](kw0)}}=>/?[Value='inhibitor'](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/?[Value='inhibitor'](kw1)=>/?[Tag='GENE'](kw0)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='inhibitor'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Value='inhibitor'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value='inhibitor'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/NP{//?[Value='inhibitor'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/PP{//?[Tag='DRUG'](kw2)=>//?[Tag='GENE'](kw0)}=>/NP{/?[Value='inhibitor'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)=>/?[Tag='DRUG'](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
//NP{/NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/PP{//?[Tag='DRUG'](kw2)}=>/?[Tag='GENE'](kw0)=>/?[Value='inhibitor'](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibited'](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/?[Value='inhibited'](kw1)=>/?[Tag='GENE'](kw0)} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value='inhibited'](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/PP{//?[Tag='GENE'](kw0)}}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Tag='GENE'](kw0)}}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Tag='GENE'](kw0)}}=>/VP{/?[Value='inhibited'](kw1)=>/PP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibited'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
```

```
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibit'](kw1)}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{//?[Value='inhibit'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value='inhibit'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/PP{//?[Value='inhibition'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/NP{//?[Value='inhibition'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/?[Value='inhibition'](kw1)=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Value='inhibition'](kw1)=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/?[Value='inhibition'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/VP{/PP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/VP{/NP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Value IN {'inhibition','Inhibition'}](kw1)=>/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Value='inhibition'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
//S{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
```

```
60 //NP{/NP{/?[Value='inhibition '](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
61 //NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='inhibition '](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
62 //NP{/NP{/?[Value='inhibition '](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
63 //NP{/NP{/NP{/?[Value='inhibition '](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
64 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value='inhibits '](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
65 /S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibits '](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
66 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value='inhibits '](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
67 //S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibits '](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw2.value, kw1.value, kw0.value, sent.value
68 //NP{/NP{/NP{/?[Value='inhibitory '](kw1)}=>/PP{//?[Tag='DRUG'](kw2)}}=>/PP{/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
69 //S{/?[Tag='DRUG'](kw2)=>/VP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
70 //NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
71 //NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)}=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
72 //NP{/NP{/NP{/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)}}=>/?[Value='inhibitor '](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
73 //NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
74 //NP{/NP{/?[Value='inhibitor '](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
75 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Tag='GENE'](kw0)=>//?[Value='inhibitor '](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
76 //S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Value='inhibitor '](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
77 //NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Tag='GENE'](kw0)}}=>/?[Value='inhibitor '](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
78 //S{/?[Tag='DRUG'](kw2)=>/?[Value='inhibitor '](kw1)=>/?[Tag='GENE'](kw0)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
79 //NP{/NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='inhibitor '](kw1)}=>/PP{//?[
     Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
80 //S{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)=>/?[Tag='DRUG'](kw2)
     } ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
81 //NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Value='inhibitor '](kw1)}=>/PP{//?[
     Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
82 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value='inhibitor '](kw1)=>//?[Tag='
     GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
     value, sent.value
83 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/NP{//?[Value='inhibitor '](kw1)=>//?[
     Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
84 //NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='
     inhibitor '](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
85 //NP{/NP{/PP{//?[Tag='DRUG'](kw2)=>//?[Tag='GENE'](kw0)}=>/NP{/?[Value='
     inhibitor '](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
86 //NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)=>/?[Tag='DRUG
     '](kw2)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value,
     sent.value
87 //S{/?[Tag='DRUG'](kw2)=>/?[Tag='GENE'](kw0)=>/?[Value='inhibitor '](kw1)
     } ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
88 //NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Tag='GENE'](kw0)=>/?[Value='inhibitor
     '](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value,
     sent.value
89 //NP{/NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Tag='GENE'](kw0)=>/?[Value='
     inhibitor '](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
90 //NP{/NP{/PP{//?[Tag='DRUG'](kw2)}=>/?[Tag='GENE'](kw0)=>/?[Value='
     inhibitor '](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
     value, sent.value
91 //S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibited '](kw1)=>/NP{//?[Tag='
     GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
     value, sent.value
92 //S{/NP{/NP{/?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited '](kw1)=>/NP
     {//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value
     , kw2.value, sent.value
93 //S{/?[Tag='DRUG'](kw2)=>/?[Value='inhibited '](kw1)=>/?[Tag='GENE'](kw0)
     } ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
94 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value='inhibited '](kw1)=>/NP{//?[
     Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
     .value, sent.value
95 //S{/NP{/NP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited '](kw1)=>/NP
     {/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value,
      kw2.value, sent.value
96 //S{/NP{/PP{//?[Tag='GENE'](kw0)}}=>/VP{/VP{/?[Value='inhibited '](kw1)
     =>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.
     value, kw2.value, sent.value
97 //S{/NP{/NP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited '](kw1)=>/NP
     {//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value
     , kw2.value, sent.value
98 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value='inhibited '](kw1)=>//?[Tag='
     GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
     value, sent.value
```

```
99 //S{/NP{/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
100 //S{/NP{/NP{/?[Tag='GENE'](kw0)}}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
101 //S{/NP{/NP{/?[Tag='GENE'](kw0)}}=>/VP{/?[Value='inhibited'](kw1)=>/PP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
102 //S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibited'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
103 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
104 //S{/NP{/NP{/?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
105 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{/?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
106 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
107 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='inhibited'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
108 //S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibited'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
109 //S{/NP{/NP{/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)}}=>/VP{/?[Value='inhibit'](kw1)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
110 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{//?[Value='inhibit'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
111 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value='inhibit'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
112 //NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
113 //S{/?[Tag='DRUG'](kw2)=>/VP{/PP{//?[Value='inhibition'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
114 //S{/NP{/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/NP{//?[Value='inhibition'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
115 //NP{/?[Value='inhibition'](kw1)=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
116 //S{/?[Value='inhibition'](kw1)=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
117 //NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
118 //NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
119 //S{/NP{/?[Value='inhibition'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/VP{/PP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
120 //S{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/VP{/NP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
121 //S{/?[Value IN {'inhibition','Inhibition'}](kw1)=>/?[Tag='GENE'](kw0)=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
122 //NP{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
123 //NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='inhibition'](kw1)}=>/PP{/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
124 //NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
125 //S{/?[Tag='DRUG'](kw2)=>/VP{/NP{//?[Value='inhibition'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
126 //S{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='DRUG'](kw2)}}=>/VP{/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
127 //NP{/NP{/?[Value='inhibition'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
128 //NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='inhibition'](kw1)}=>/PP{/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
129 //NP{/NP{/?[Value='inhibition'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
130 //NP{/NP{/NP{/?[Value='inhibition'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
131 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/?[Value='inhibits'](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
132 /S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibits'](kw1)=>/NP{/?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
133 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value='inhibits'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
134 //S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='inhibits'](kw1)=>/NP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

B.7 Gene Metabolized Drug

```
1 //S{/NP{/NP{/?[Tag='DRUG'](kw2)}}=>/VP{/VP{/?[Value IN {'metabolised','metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value IN {'metabolised','
    metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid,
    kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/SBAR{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value IN {'metabolised','
    metabolized'}](kw1)=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.
    cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{/?[Value IN {'metabolised','
    metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid,
    kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{//?[Value IN {'metabolized','
    metabolised'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid,
    kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value IN {'metabolised','metabolized
    '}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value,
    kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='metabolized'](kw1)=>/PP{//?[Tag='
    GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='metabolism'](kw1)}=>/VP{/VP{//?[
    Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
    .value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/NP{//?[Tag='DRUG'](kw2)=>//?[Value='
    metabolism'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value,
    kw2.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='metabolism'](kw1)}=>/PP{/NP{/?[
    Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
    .value, sent.value
//NP{/?[Value='metabolism'](kw1)=>/PP{/NP{//?[Tag='DRUG'](kw2)=>//?[Tag
    ='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
    value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='metabolism'](kw1)=>//?[
    Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
    .value, sent.value
//S{/NP{/NP{/?[Tag='GENE'](kw0)}}=>/VP{/NP{//?[Value='metabolism'](kw1)
    =>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.
    value, kw2.value, sent.value
//NP{/NP{/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)=>/?[
    Value='metabolism'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.
    value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/PP{//?[Value='metabolism'](kw1)=>//?[
    Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
    .value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='metabolism'](kw1)}=>/PP{/NP{//?[
    Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
    .value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='metabolism'](kw1)}=>/PP{/NP{//?[
    Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
    .value, sent.value
//S{/NP{/NP{/?[Value='metabolism'](kw1)}=>/PP{//?[Tag='DRUG'](kw2)}}=>/
    VP{/VP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1
    .value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/PP{//?[Tag='DRUG'](kw2)=>//?[Value='
    metabolism'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value,
    kw2.value, sent.value
//NP{/NP{/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{//?[Value='metabolism'](kw1
    )=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.
    value, kw2.value, sent.value
```

```
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/NP{//?[Value='substrate'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='substrate'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value='substrate'](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='GENE'](kw0)=>/?[Value='substrate'](kw1)=>/?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Value='substrate'](kw1)}=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/NP{/?[Tag='DRUG'](kw2)}}=>/VP{/VP{/?[Value IN {'metabolised','metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value IN {'metabolised','metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/SBAR{//?[Tag='DRUG'](kw2)}}=>/VP{/?[Value IN {'metabolised','metabolized'}](kw1)=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{/?[Value IN {'metabolised','metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/VP{//?[Value IN {'metabolized','metabolised'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/VP{/?[Value IN {'metabolised','metabolized'}](kw1)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/VP{/?[Value='metabolized'](kw1)=>/PP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='metabolism'](kw1)}=>/VP{/VP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/NP{//?[Tag='DRUG'](kw2)=>//?[Value='metabolism'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='metabolism'](kw1)}=>/PP{/NP{/?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/?[Value='metabolism'](kw1)=>/PP{/NP{//?[Tag='DRUG'](kw2)=>//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='metabolism'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//S{/NP{/NP{/?[Tag='GENE'](kw0)}}=>/VP{/NP{//?[Value='metabolism'](kw1)=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
//NP{/NP{/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='metabolism'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.value
```

```
40 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/PP{//?[Value='metabolism'](kw1)=>//?[
       Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
       .value, sent.value
41 //NP{/NP{/?[Tag='DRUG'](kw2)=>/?[Value='metabolism'](kw1)}=>/PP{/NP{//?[
       Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
       .value, sent.value
42 //NP{/NP{/?[Tag='GENE'](kw0)=>/?[Value='metabolism'](kw1)}=>/PP{/NP{//?[
       Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
       .value, sent.value
43 //S{/NP{/NP{/?[Value='metabolism'](kw1)}=>/PP{//?[Tag='DRUG'](kw2)}}=>/
       VP{/VP{//?[Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1
       .value, kw2.value, sent.value
44 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/PP{//?[Tag='DRUG'](kw2)=>//?[Value='
       metabolism'](kw1)}}} ::: distinct sent.cid, kw0.value, kw1.value,
       kw2.value, sent.value
45 //NP{/NP{/PP{//?[Tag='GENE'](kw0)}}=>/PP{/NP{//?[Value='metabolism'](kw1
       )=>//?[Tag='DRUG'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.
       value, kw2.value, sent.value
46 //S{/NP{/?[Tag='DRUG'](kw2)}=>/VP{/NP{//?[Value='substrate'](kw1)=>//?[
       Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
       .value, sent.value
47 //NP{/NP{/?[Tag='DRUG'](kw2)=>/NP{/?[Value='substrate'](kw1)}=>/PP{//?[
       Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
       .value, sent.value
48 //S{/?[Tag='DRUG'](kw2)=>/VP{/VP{//?[Value='substrate'](kw1)=>//?[Tag='
      GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.
       value, sent.value
49 //NP{/?[Tag='GENE'](kw0)=>/?[Value='substrate'](kw1)=>/?[Tag='DRUG'](kw2
       )} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, sent.
       value
50 //NP{/?[Tag='DRUG'](kw2)=>/NP{/NP{/?[Value='substrate'](kw1)}=>/PP{//?[
       Tag='GENE'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2
       .value, sent.value
```

B.8 Gene Regulates Gene

```
1 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value IN {'regulated', '
      upregulated', 'downregulated', 'up-regulated', 'down-regulated'}](
      kw1)=>//?[Value='by'](kw3)=>//?[Tag='GENE'](kw2)}}} ::: distinct
      sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
2 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{/?[Value IN {'regulated', '
      upregulated', 'downregulated', 'up-regulated', 'down-regulated'}](
      kw1)=>//?[Value='by'](kw3)=>//?[Tag='GENE'](kw2)}}} ::: distinct
      sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
3 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'regulated','down-
      regulated'}](kw1)=>/NP{//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid
      , kw0.value, kw1.value, kw2.value, sent.value
4 //NP{/NP{/?[Value IN {'regulation', 'upregulation', 'downregulation', '
      up-regulation', 'down-regulation'}](kw1)}=>/PP{/NP{//?[Tag='GENE'](
      kw0)=>//?[Value='by'](kw3)=>//?[Tag='GENE'](kw2)}}} ::: distinct
      sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
5 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/?[Value IN {'regulates', 'upregulates
      ', 'downregulates', 'up-regulates', 'down-regulates'}](kw1)=>/NP
      {//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value
      , kw2.value, sent.value
```

```
6 //S{/NP{/?[Tag='GENE'](kw0)}=>/VP{/VP{//?[Value='in'](kw3)=>//?[Value='regulating'](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
```

B.9 Gene Regulate Gene (Xenobiotic Metabolism)

```
1 //S{//?[Tag='GENE' AND Canonical LIKE 'CYP\%'](kw0)<=>//?[Tag='GENE' AND Canonical IN {'AhR', 'CASR', 'CAR', 'PXR', 'NR1I2', 'NR1I3'}](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, sent.value
2 //S{//?[Tag='GENE' AND Value LIKE 'cytochrome\%'](kw0)<=>//?[Tag='GENE' AND Canonical IN {'AhR', 'CASR', 'CAR', 'PXR', 'NR1I2', 'NR1I3'}](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, sent.value
```

B.10 Negative Drug Induces/Metabolizes/Inhibits Gene

```
1 //S{/?[Tag='DRUG'](kw0)=>/VP{/VP{/?[Value IN {'induced', 'inhibited', 'metabolized', 'metabolised'}](kw1)=>//?[Tag='GENE'](kw2)=>//?[Value='not'](kw3)=>//?[Tag='GENE'](kw4)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw4.value, kw3.value, sent.value
2 //S{/?[Tag='DRUG'](kw0)=>/?[Value='not'](kw3)=>/?[Tag='DRUG'](kw4)=>/?[Value='inhibited'](kw1)=>/?[Tag='GENE'](kw2)} ::: distinct sent.cid, kw4.value, kw1.value, kw2.value, kw3.value, sent.value
3 //S{/SBAR{/S{//?[Tag='DRUG'](kw0)}}=>/S{/S{//?[Value='metabolized'](kw1)=>//?[Tag='GENE'](kw2)=>//?[Value='not'](kw3)=>//?[Tag='GENE'](kw4)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw4.value, kw3.value, sent.value
4 //S{/NP{/PP{//?[Tag='GENE'](kw2)}}=>/VP{/VP{/?[Value IN {'induced', 'inhibited', 'metabolized', 'metabolised'}](kw1)=>//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw3)=>//?[Tag='DRUG'](kw4)}}} ::: distinct sent.cid, kw4.value, kw1.value, kw2.value, kw3.value, sent.value
5 //S{/NP{/PP{//?[Tag='DRUG'](kw0)}}=>/VP{/VP{/?[Value IN {'induced', 'inhibited', 'metabolized', 'metabolised'}](kw1)=>//?[Tag='GENE'](kw2)=>//?[Value IN {'not','no'}](kw3)=>//?[Tag='GENE'](kw4)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw4.value, kw3.value, sent.value
6 //S{/?[Tag='DRUG'](kw0)=>/VP{/?[Value='not'](kw3)=>/VP{//?[Value IN {'induced', 'inhibited', 'metabolized', 'metabolised'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
7 //S{/NP{/?[Tag='GENE'](kw2)}=>/VP{/VP{/?[Value IN {'induced', 'inhibited', 'metabolized', 'metabolised'}](kw1)=>//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw3)=>//?[Tag='DRUG'](kw4)}}} ::: distinct sent.cid, kw4.value, kw1.value, kw2.value, kw3.value, sent.value
8 //S{/NP{/NP{/?[Value='not'](kw3)=>/?[Tag='DRUG'](kw0)}}=>/VP{/?[Value IN {'inhibited','induced'}](kw1)=>/NP{//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
9 //S{/NP{/NP{/?[Value IN {'no','not'}](kw3)=>/?[Tag='DRUG'](kw0)}}=>/VP{/?[Value IN {'inhibited','induced'}](kw1)=>/NP{/?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
10 //S{/?[Tag='DRUG'](kw0)=>/S{/S{//?[Value='not'](kw3)=>//?[Value IN {'induce', 'inhibit'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
11 //S{/?[Tag='DRUG'](kw0)=>/?[Tag='GENE'](kw2)=>/?[Value IN {'not'}](kw3)=>/?[Value IN {'inhibit','induce'}](kw1)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
```

```
12 //S{/?[Tag='DRUG'](kw0)=>/VP{/?[Value='not'](kw3)=>/VP{/?[Value IN {'inhibit','induce'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
13 //S{/NP{/?[Tag='DRUG'](kw0)}=>/VP{/VP{/?[Value='not'](kw3)=>//?[Value='inhibit'](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
14 //S{/NP{/NP{/?[Tag='GENE'](kw2)}}=>/VP{/?[Value='not'](kw3)=>/VP{/?[Value='metabolize'](kw1)=>//?[Tag='DRUG'](kw0)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
15 //S{/NP{/NP{//?[Tag='DRUG'](kw0)}}=>/VP{/?[Value='not'](kw3)=>/VP{/?[Value IN {'inhibit','induce'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
16 //S{/NP{/?[Tag='DRUG'](kw0)}=>/VP{/?[Value='not'](kw3)=>/VP{/?[Value='inhibit'](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
17 //S{/NP{/?[Tag='DRUG'](kw0)}=>/VP{/?[Value IN {'induces', 'inhibits'}](kw1)=>/NP{//?[Tag='GENE'](kw2)=>//?[Value='not'](kw3)=>//?[Tag='GENE'](kw4)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw4.value, kw3.value, sent.value
18 //S{/?[Tag='DRUG'](kw0)=>/VP{/?[Value='inhibits'](kw1)=>/NP{//?[Tag='GENE'](kw2)=>//?[Value='not'](kw3)=>//?[Tag='GENE'](kw4)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw4.value, kw3.value, sent.value
19 //S{/?[Tag='DRUG'](kw0)=>/VP{/NP{//?[Value='no'](kw3)=>//?[Value IN {'induction','metabolism','inhibition'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
20 //S{/NP{/NP{//?[Tag='DRUG'](kw0)=>//?[Tag='GENE'](kw2)=>//?[Value IN {'induction','metabolism','inhibition'}](kw1)}}=>/VP{/?[Value='not'](kw3)}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
21 //S{/?[Tag='DRUG'](kw0)=>/VP{/?[Value='not'](kw3)=>/VP{//?[Value IN {'induction','metabolism','inhibition'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
22 //S{/NP{/PP{//?[Tag='DRUG'](kw0)}}=>/VP{/?[Value='not'](kw3)=>/VP{//?[Value IN {'induction','metabolism','inhibition'}](kw1)=>//?[Tag='GENE'](kw2)}}} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw3.value, sent.value
```

B.11 Negative Drug Induces Gene

```
1 //S{//?[Tag='GENE'](kw0)=>//?[Value IN {'induced','increased','stimulated'}](kw1)=>//?[Value='by'](kw3)=>//?[Tag='DRUG'](kw2)=>//?[Value='not'](kw4)=>//?[Tag='DRUG'](kw5)} ::: distinct sent.cid, kw5.value, kw1.value, kw0.value, kw4.value, sent.value
2 //S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)=>//?[Value IN {'induced','increased','stimulated'}](kw1)=>//?[Value='by'](kw3)=>//?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw2.value, kw1.value, kw5.value, kw4.value, sent.value
3 //S{//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw4)=>//?[Value IN {'induce','induced','increase','increased','stimulate','stimulated'}](kw1)=>//?[Tag='GENE'](kw2)} ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw4.value, sent.value
4 //S{//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='DRUG'](kw5)=>//?[Value IN {'induces','increases','stimulates','induced','
```

```
    increased','stimulated'}](kw1)=>//?[Tag='GENE'](kw2)} ::: distinct
    sent.cid, kw5.value, kw1.value, kw2.value, kw4.value, sent.value
//S{//?[Value='no'](kw5)=>//?[value IN {'induction','stimulation'}](kw1)
    =>//?[Value IN {'of'}](kw2)=>//?[Tag='GENE'](kw0)=>//?[value IN {'by
    '}](kw3)=>//?[Tag='DRUG'](kw4)} ::: distinct sent.cid, kw4.value,
    kw1.value, kw0.value, kw5.value, sent.value
```

B.12 Negative Drug Inhibits Gene

```
//S{//?[Tag='GENE'](kw0)=>//?[Value IN {'inhibited','decreased'}](kw1)
    =>//?[Value='by'](kw3)=>//?[Tag='DRUG'](kw2)=>//?[Value='not'](kw4)
    =>//?[Tag='DRUG'](kw5)} ::: distinct sent.cid, kw5.value, kw1.value,
     kw0.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)
    =>//?[Value IN {'inhibited','decreased'}](kw1)=>//?[Value='by'](kw3)
    =>//?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw2.value, kw1.value,
     kw5.value, kw4.value, sent.value
//S{//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw4)=>//?[Value IN {'inhibit
    ','inhibited','decrease','decreased'}](kw1)=>//?[Tag='GENE'](kw2)}
    ::: distinct sent.cid, kw0.value, kw1.value, kw2.value, kw4.value,
    sent.value
//S{//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='DRUG'](kw5)
    =>//?[Value IN {'inhibits','decreases','inhibited','decreased'}](kw1
    )=>//?[Tag='GENE'](kw2)} ::: distinct sent.cid, kw5.value, kw1.value
    , kw2.value, kw4.value, sent.value
//S{//?[Value='no'](kw5)=>//?[value IN {'inhibition'}](kw1)=>//?[Value
    IN {'of'}](kw2)=>//?[Tag='GENE'](kw0)=>//?[value IN {'by'}](kw3)
    =>//?[Tag='DRUG'](kw4)} ::: distinct sent.cid, kw4.value, kw1.value,
     kw0.value, kw5.value, sent.value
```

B.13 Negative Gene Metabolizes Drug

```
//S{//?[Tag='DRUG'](kw0)=>//?[Value IN {'metabolized','metabolised'}](
    kw1)=>//?[Value='by'](kw3)=>//?[Tag='GENE'](kw2)=>//?[Value='not'](
    kw4)=>//?[Tag='GENE'](kw5)} ::: distinct sent.cid, kw0.value, kw1.
    value, kw5.value, kw4.value, sent.value
//S{//?[Tag='DRUG'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='DRUG'](kw5)
    =>//?[Value IN {'metabolized','metabolised'}](kw1)=>//?[Value='by'](
    kw3)=>//?[Tag='GENE'](kw2)} ::: distinct sent.cid, kw5.value, kw1.
    value, kw2.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Value IN {'
    metabolize','metabolise'}](kw1)=>//?[Tag='DRUG'](kw2)} ::: distinct
    sent.cid, kw2.value, kw1.value, kw0.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)
    =>//?[Value IN {'metabolize','metabolise','metabolizes','metabolises
    '}](kw1)=>//?[Tag='DRUG'](kw2)} ::: distinct sent.cid, kw2.value,
    kw1.value, kw5.value, kw4.value, sent.value
```

B.14 Negative Gene Downregulates Gene

```
//S{//?[Tag='GENE'](kw0)=>//?[Value IN {'suppressed','downregulated','
    inhibited','down-regulated','repressed','disrupted'}](kw1)=>//?[
    Value='by'](kw3)=>//?[Tag='GENE'](kw2)=>//?[Value='not'](kw4)=>//?[
    Tag='GENE'](kw5)} ::: distinct sent.cid, kw5.value, kw1.value, kw0.
    value, kw4.value, sent.value
```

```
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)
    =>//?[Value IN {'suppressed','downregulated','inhibited','down-
    regulated','repressed','disrupted'}](kw1)=>//?[Value='by'](kw3)
    =>//?[Tag='GENE'](kw2)} ::: distinct sent.cid, kw2.value, kw1.value,
     kw5.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Value IN {'
    suppressed','suppress','downregulated','downregulate','inhibited','
    inhibit','down-regulated','down-regulate','repressed','repress','
    disrupted','disrupt'}](kw1)=>//?[Tag='GENE'](kw2)} ::: distinct sent
    .cid, kw0.value, kw1.value, kw2.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)
    =>//?[Value IN {'suppresses','downregulates','inhibits','down-
    regulates','represses','disrupts','suppressed','downregulated','
    inhibited','down-regulated','repressed','disrupted'}](kw1)=>//?[Tag
    ='GENE'](kw2)} ::: distinct sent.cid, kw5.value, kw1.value, kw2.
    value, kw4.value, sent.value
//S{//?[Value='no'](kw5)=>//?[value IN {'inhibition','downregulation','
    down-regulation'}](kw1)=>//?[Value IN {'of'}](kw2)=>//?[Tag='GENE'](
    kw0)=>//?[value IN {'on'}](kw3)=>//?[Tag='GENE'](kw4)} ::: distinct
    sent.cid, kw0.value, kw1.value, kw4.value, kw5.value, sent.value
```

B.15 Negative Gene Upregulates Gene

```
//S{//?[Tag='GENE'](kw0)=>//?[Value IN {'activated','induced','
    stimulated','regulated','upregulated','up-regulated'}](kw1)=>//?[
    Value='by'](kw3)=>//?[Tag='GENE'](kw2)=>//?[Value='not'](kw4)=>//?[
    Tag='GENE'](kw5)} ::: distinct sent.cid, kw5.value, kw1.value, kw0.
    value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)
    =>//?[Value IN {'activated','induced','stimulated','regulated','
    upregulated','up-regulated'}](kw1) ::: distinct sent.cid, kw2.value,
     kw1.value, kw5.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Value IN {'
    activated','induced','stimulated','regulated','upregulated','up-
    regulated','activate','induce','stimulate','regulate','upregulate','
    up-regulate'}](kw1)=>//?[Tag='GENE'](kw2)} ::: distinct sent.cid,
    kw0.value, kw1.value, kw2.value, kw4.value, sent.value
//S{//?[Tag='GENE'](kw0)=>//?[Value='not'](kw4)=>//?[Tag='GENE'](kw5)
    =>//?[Value IN {'activates','induces','stimulates','regulates','
    upregulates','up-regulates','activate','induce','stimulate','
    regulate','upregulate','up-regulate'}](kw1)=>//?[Tag='GENE'](kw2)}
    ::: distinct sent.cid, kw5.value, kw1.value, kw2.value, kw4.value,
    sent.value
//S{//?[Value='no'](kw5)=>//?[value IN {'induction','activation','
    stimulation','regulation','upregulation','up-regulation'}](kw1)
    =>//?[Value IN {'of'}](kw2)=>//?[Tag='GENE'](kw0)=>//?[value IN {'by
    '}](kw3)=>//?[Tag='GENE'](kw4)} ::: distinct sent.cid, kw4.value,
    kw1.value, kw0.value, kw5.value, sent.value
```

B.16 Drug Gene Co-Occurrence

```
//S{//?[Tag='DRUG'](kw0)<=>//?[Tag='GENE'](kw1)} ::: distinct sent.cid,
    kw0.value, kw1.value, kw0.type, kw1.type, sent.value
```