Reasoning about Non-immediate Triggers in Biological Networks

Nam Tran
Pathology Informatics
School of Medicine
Yale University
New Haven, CT 06510, USA
nam.tran@yale.edu

Chitta Baral
Computer Science and Engineering
School of Computing & Informatics
Arizona State University
Tempe, AZ 85281, USA
chitta@asu.edu

Abstract

Modeling molecular interactions in biological networks is important from various perspectives such as predicting side effects of drugs, explaining unusual cellular behavior and drug and therapy design. Various formal languages have been proposed for representing and reasoning about molecular interactions. The interactions are modeled as triggered events in most of the approaches. The triggering of events is assumed to be immediate: once an interaction is triggered, it should occur immediately. Although working well for engineering systems, this assumption poses a serious problem in modeling biological systems. Our knowledge about biological systems is inherently incomplete, thus molecular interactions are constantly elaborated and refined at different granularity of abstraction. The model of immediate triggers can not consistently deal with this refinement. In this paper we propose an action language to address this problem. We show that the language allows for refinements of biological knowledge, although at a higher cost in terms of complexity.

1 Introduction

The living cell constantly receives and responds to signals from its environment. A signal is initiated when some extracellular molecules are sensed by their respective cell-surface receptor. Signaling molecules inside the cell then interact with one another to transduce the signal into cellular responses that regulate the introduction of different proteins, thus controlling various functions of the cell. Specific collections of interactions with a common theme are often referred to as *biological signaling networks*. Almost any disease can be described in terms of aberrations in these signaling networks. For example, most cancers are caused by a breakdown in networks regulating cell growth. Modeling biological networks is thus essential for building hypotheses about functions of the cell. After experimental verifications, these hypotheses become new knowledge that can lead to effective therapeutic strategies that correct or alter abnormal cell behaviors.

In the past action languages have been developed and applied to domains such as robots [16, 29, 37, 33, 5], agents [21], helicopters [9], and space shuttles [2, 3, 23]. Recently, an extension of action language $\mathcal{A}$ with triggered actions - the language $\mathcal{A}_T^0$ - was proposed by [38, 4] to model cellular interactions. In $\mathcal{A}_T^0$, an action theory of the cell specifies effects of actions and how actions are triggered or inhibited. Such a specification dictates how the cell evolves (i.e. changes through time). An evolution of the cell starts from a state which triggers certain actions. These actions change the state, which may trigger further actions and so on. Reasoning abilities in $\mathcal{A}_T^0$ include (i) predicting the impact of a particular action, (ii) explaining observations, (iii) planning to make certain components of the cell behave in a particular way. Each of the

above has ultimate significance to cell biology and medical science. For example, one might want to know if taking a drug has a side effect in terms of if it can prevent a particular hormone from being produced thus disrupting certain cellular and biological mechanisms. This corresponds to prediction. Another example is that when a cell is observed to behave in an abnormal way (eg. it keeps proliferating instead of dying), one may want to find out why that is the case. This corresponds to explanation or medical diagnosis. One might then want to figure out a way, perhaps by introducing particular drug elements to the cell or cell membrane at particular time instances, to make the cell behave in a particular way. This would correspond to drug design and drug therapy.

In this paper, we address an important limitation of the model of triggered actions in $\mathcal{A}_T^0$. Triggered actions in $\mathcal{A}_T^0$ are assumed to occur immediately as soon as they are enabled. This modeling assumption works well in artificial engineering systems such as robots or automatic controllers. It is also reasonable, since we usually possess the complete knowledge about artificial systems. However, knowledge about biological systems such as the cell is inherently incomplete. We often need to model biological triggers at different granularity of time and abstraction. Finer granularity usually arises when more detailed knowledge becomes available. For example, a biological action or trigger would be shown to consist of other actions and triggers as sub-components. These new sub-components would change the order of events. An alternative ordering of events would result in a different final outcome, thus leading to contradictions to a previously formed conclusion. An action language for biological domains should allow for this type of knowledge refinement without compromising the consistency of knowledge bases. This limitation of $\mathcal{A}_T^0$ is also inherent in related models of biological networks [27, 24, 28, 13, 19, 35, 8, 14, 7, 11, 30] as well as in related action formalisms of triggers [31, 26, 34, 20, 25, 30, 18, 14, 17, 37].

The paper is organized as follows. In Section 2 we give a brief overview of the action language $\mathcal{A}_T^0$ then motivate the our language with an abstract example. In Section 3, we present the syntax and semantics of the new language $\mathcal{A}_T^\infty$. We study monotic refinements of rules and the complexity of the language $\mathcal{A}_T^\infty$ in Section 4 and 5, respectively. The Appendix includes technical background on complexity theory (for the analysis of the complexity of $\mathcal{A}_T^\infty$) and the linear temporal logic (for representing queries in $\mathcal{A}_T^\infty$).

2 Background and Motivating Example

In dynamic domains, reasoning about actions and changes is of imperative importance [22]. The objective of action languages is to provide succinct and elaboration tolerant representation of an agent acting in a dynamic world. Research issues in the field of reasoning about actions include:

- developing languages that allow for (a) succinct representation of the world, (b) succinct representation of causal relations between properties of the world, (c) succinct representation of the impact of actions (and their executability) on the world, (d) succinct representation of the way the world evolves, or the way we want the world to evolve, and (e) representation of action-plans; and
- reasoning algorithms that can answer a query expressed using (d) and (e) with respect to an action theory composed of (a), (b) and (c).

Different kinds of queries mentioned above lead to different kinds of reasoning. For example, given a sequence of actions $a_1, \ldots, a_n$ and a property p, asking if p will be true during (or after) the execution of $a_1, \ldots, a_n$ is referred to as *prediction*. Now if p is given, then finding the appropriate $a_1, \ldots, a_n$ so that p is achieved is called *planning*. Finally, if a set of observations is given and one needs to find particular action occurrences and/or facts about intermediate states of the world that explain the observations, then one does

explanation or diagnosis. All the above kinds of reasoning may have to be done with partial or incomplete knowledge about the world.

The action language $\mathcal{A}$ [12] was developed to be a high-level language for reasoning about actions and their effects without involving details of any particular logic. Due to the simplicity and intuitiveness of $\mathcal{A}$, it has been used as the basis of the language $\mathcal{A}_T^0$ for triggered actions. Understanding this basis is important for further discussion, so we will now briefly review the language $\mathcal{A}$.

2.1 Action language $\mathcal{A}$

An *action theory* in $\mathcal{A}$ [12] is defined over two disjoint sets, a set of actions $\mathbf{A}$ and a set of fluents $\mathbf{F}$.

A *fluent literal* is a *fluent* (eg. f) or the negation of a fluent (eg. $\neg f$). A set of fluent literals is said to be *consistent* if it does not contain both f and $\neg f$ for some fluent f. An *interpretation* I of the fluents in $\mathcal{D}$ is a maximal consistent set of fluent literals of $\mathcal{D}$. A fluent f is said to be true (resp. false) in I if $f \in I$ (resp. $\neg f \in I$). The truth value of a fluent formula in I is defined recursively over the propositional connectives in the usual way. For example, $f \wedge g$ is true in I if f is true in I and g is true in I. We say that a formula φ holds in I (or I satisfies φ), denoted by $I \models \varphi$, if φ is true in I. A *state* is an interpretation of fluents.

A *domain description* is a set of statements of the form:

$$a \textbf{ causes } f \textbf{ if } f_1, \ldots, f_n \tag{1}$$

When $n = 0$, the above statement is simply written as a **causes** f.

Observations are statements about the initial state, which are of the form:

$$\textbf{initially } f$$

A set of observations O is said to be complete if for any fluent f, either **initially** $f \in O$ or **initially** $\neg f \in O$.

Queries in $\mathcal{A}$ are statements of the form:

$$Q = f \textbf{ after } a_1, \ldots, a_n \tag{2}$$

where f is a fluent literal, and $a_1, \ldots, a_n$ are actions. Intuitively, this statement queries whether f is true after the sequence of actions $a_1, \ldots, a_n$.

A *state transition* is a change of one state to another state due to effects of some actions. The effect of an action a in a state s is the set

$$E(a, s) = \{\ f \mid a \textbf{ causes } f \textbf{ if } f_1, \ldots f_n \in \mathcal{D} \text{ and } \{f_1, \ldots, f_n\} \subseteq s\ \}. \tag{3}$$

Let $\neg\neg g = g$ and $\neg E(a, s) = \{\neg g | g \in E(a, s)\}$. State transitions are computed by the transition function defined as follows.

Definition 2.1. A transition function of a domain $\mathcal{D}$ is a function Φ from pairs of actions and states into states such that:

- if $E(a, s)$ is consistent, then

$$\Phi(a, s) = (s \setminus \neg E(a, s)) \cup E(a, s)\ ;$$

- otherwise $\Phi(a, s)$ is undefined.

An *action theory* is a pair (D, O), where D is a domain description and O is a set of observations. A state s_0 is an initial state corresponding to an action theory (D, O) if for every fluent literal g, $g \in s_0$ iff **initially** $g \in O$. We then say that $\langle s_0, \Phi \rangle$ is a model of (D, O).

An action theory (D, O) entails a query Q of the form (2), if for all models $\langle s_0, \Phi \rangle$ of (D, O), f holds in the state $\Phi(a_n, \Phi(a_{n-1}, \ldots \Phi(a_1, s_0) \ldots)$. The entailment is denoted $(D, O) \models Q$.

2.2 Language $\mathcal{A}_T^0$ for triggered actions

The action language $\mathcal{A}_T^0$ extends $\mathcal{A}$ with statements representing triggered actions. A domain description $\mathcal{D}$ in $\mathcal{A}_T^0$ is a set of statements of the form (1) and of the following forms:

$$g_1, \ldots, g_m \textbf{ triggers } b \tag{4}$$

$$h_1, \ldots, h_l \textbf{ inhibits } c \tag{5}$$

where g_j, h_k are fluent literals and b, c are individual actions. (4) is called a *trigger* rule (or simply trigger), which says that action b is to occur if it is not inhibited and if all the literals $g_1, \ldots, g_m$ hold. (5) is an *inhibition* rule, which says that action c can not happen if all the literals $h_1, \ldots, h_l$ hold.

An action a is said to be *triggered* by a state s, if there exists a trigger rule (4) such that all the literals $g_1, \ldots, g_m$ are true in s. An action a is said to be *inhibited* by a state s, if there exists a inhibition rule (5) such that all the literals $h_1, \ldots, h_l$ are true in s.

Since a state can trigger multiple actions, *state transitions* in $\mathcal{A}_T^0$ are extended to pairs of sets of actions and states. The direct effect of a set A of actions in a state s is the set

$$E(A, s) = \bigcup_{a \in A} E(a, s).$$

where $E(a, s)$ is defined by (3). Let us denote $\neg E(A, s) = \{\neg f \mid f \in E(A, s)\}$. The state $\Phi(A, s)$ resulting from the occurrence of A in s is defined as follows.

- $\Phi(\emptyset, s) = s$;
- if $A \neq \emptyset$ and $E(A, s)$ is consistent, then

$$\Phi(A, s) = (s \setminus \neg E(A, s)) \cup E(A, s) \ ;$$

- otherwise $\Phi(A, s)$ is undefined.

A *transition sequence* τ is a sequence of the form $\tau = \langle s_0, A_0, s_1, A_1, \ldots \rangle$; where s_i's are states and A_j's are sets of actions in $\mathcal{D}$, such that $s_{i+1} = \Phi(A_i, s_i)$ for all i, and $A_j = \emptyset$ for all $j > k$ if $A_k = \emptyset$.

A *trajectory* is a transition sequence $\tau = \langle s_0, A_0, s_1, A_1, \ldots \rangle$ where A_i is a the set of all actions that are triggered but not inhibited by the state s_i (for all $i \geq 0$). *Observations* are statements of the form " f **at** i " or of the form " a **occurs_at** j ", where i and j are non-negative integers. The former statement means that the fluent literal f is observed to be true at time i. The latter means that the action a is observed to occur at time j. A trajectory $\tau = \langle s_0, A_0, s_1, A_1, \ldots \rangle$ satisfies " f **at** i " iff $f \in s_i$. Also, τ satisfies " a **occurs_at** j " iff $a \in A_j$.

A *query* in $\mathcal{A}_T^0$ has the form

$$Q = f \textbf{ after } A_1 \textbf{ at } t_1, \ldots, A_n \textbf{ at } t_n \tag{6}$$

where f is a fluent, $A_1, \ldots, A_n$ are sets of actions and $t_1 < \ldots < t_n$ are time points. When $n = 0$, we simply write $Q = f$.

A *action theory* is a pair $(\mathcal{D}, \mathcal{O})$ where $\mathcal{D}$ is a domain description and $\mathcal{O}$ is a set of observations. A *model* of an action theory $(\mathcal{D}, \mathcal{O})$ is a trajectory of $\mathcal{D}$ that satisfies all the observations of $\mathcal{O}$.

Let $\mathcal{T} = (\mathcal{D}, \mathcal{O})$ be an action theory and Q be the query (6). Let $\mathcal{O}'$ be the set of observations $\mathcal{O}' = \mathcal{O} \cup \{A_1 \textbf{ occurs_at } t_1, \ldots, A_n \textbf{ occurs_at } t_n\}$. Then $\mathcal{T}$ entails Q, written as $\mathcal{T} \models Q$, iff

(i) $(\mathcal{D}, \mathcal{O}')$ has at least one model; and

(ii) for all trajectory models $\tau = \langle s_0, A'_1, s_1, A'_2 \ldots A'_m, s_m \ldots \rangle$ of the theory $(\mathcal{D}, \mathcal{O}')$, there exists N such that f is true in all the states s_k, $k > N$.

It follows from the definition of trajectories and models that any action a that is triggered but not inhibited by s must occur in s. We say that the triggering of actions in $\mathcal{A}^0_T$ is *immediate*.

2.3 Motivating example

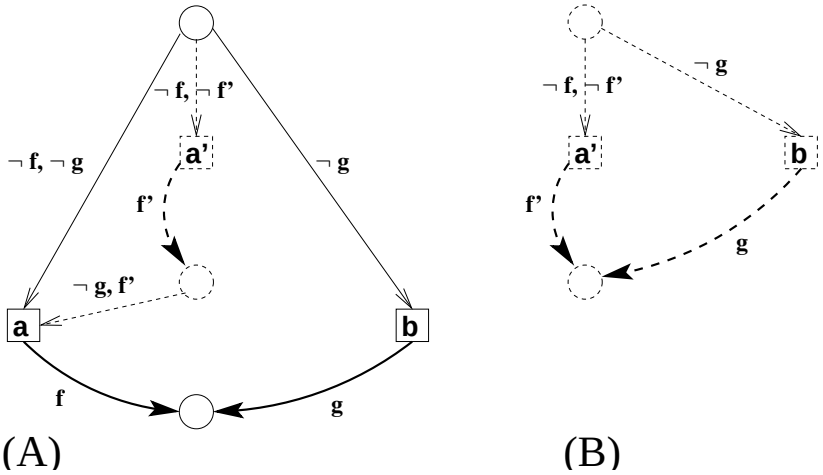

Figure 1: More detailed knowledge leads to a different interpretation of the same process. (A) Original (assumingly) correct interpretation, regardless the details of $\neg f, \neg g$ **triggers** a in dashed lines. (B) New interpretation based on $\mathcal{A}^0_T$ semantics.

The following hypothetical example illustrates a serious limitation of the immdediate tringgering of actions. Let us consider the $\mathcal{A}^0_T$ domain description:

$$\mathcal{D}_{afg} = \begin{cases} \neg f, \neg g \textbf{ triggers } a \\ a \textbf{ causes } f \\ \neg g \textbf{ triggers } b \\ b \textbf{ causes } g \end{cases}$$

The semantics of $\mathcal{A}^0_T$ dictates that if an action is triggered at time t then it has to occur at time t. Assume that f and g are false at time 0, then both a and b occur at time 0. The occurrence of a makes f *become true* at time 1. No other actions after time 1, so f remains true. Thus we have informally proved that:

$$(\mathcal{D}_{afg}, \{\neg f \textbf{ at } 0, \neg g \textbf{ at } 0\}) \models f$$

Now imagine that the rule $\neg f, \neg g$ **triggers** a is replaced by the following rules, which capture more detailed knowledge:

$$\neg f \ , \neg f' \textbf{ triggers } a' \tag{7}$$

$$a' \textbf{ causes } f' \tag{8}$$

$$f', \neg g \textbf{ triggers } a \tag{9}$$

Let $\mathcal{D}'_{afg}$ be the modified domain description. Intuitively, the triggering of the action a by $\neg f \wedge \neg g$ is mediated by some new action a' and fluent f' (dashed lines in Fig. 1A). Given the same initial condition of f and g being false at time 0, a different conclusion about the final value of f must be drawn from the refined interaction specification (Fig. 1B). The argument goes as follows. The action b is triggered and occurs at time 0, regardless of what value f' has. If f' is true at time 0, then a' is not triggered. Consequently, a' and a will not occur, so f is unchanged (remaining false). On the other hand, if f' is false at time 0 then the actions a' and b occur at time 0. Thus a' and b cause $\neg f$ and g at time 1. No additional action can be triggered from time 1 onwards, so f *is always false*. We have informally proved that:

$$(\mathcal{D}'_{afg}, \{\neg f \textbf{ at } 0, \neg g \textbf{ at } 0\}) \models \neg f$$

Hence, the new conclusion about f contradicts the previous prediction based on the original domain description $\mathcal{D}_{afg}$. Intuitively, this non-monotonicity with respect to refinement is not acceptable. Since the detailed knowledge (i.e., (7)-(9)) is about what happens "inside" the trigger of a, the consequence of the trigger itself should not be affected, provided everything "outside" remains intact.

We propose a new action language $\mathcal{A}^{\infty}_{T}$ as a solution to the problem of immediate triggered actions. In $\mathcal{A}^{\infty}_{T}$, an action will occur (i.e be fired) with some delay after it is triggered. We would still encounter the same problem if this firing delay is bounded. For example, let us assume the firing delays are bounded by N. In the domain D_{afg}, if the rule $\neg f, \neg f'$ **triggers** a' (see (7)) is refined further by means of $N+1$ immediate actions, then a would not be triggered thus cannot occur. Nevertheless, allowing unbounded delays would lead to an infinite number of possible sequences of action occurrences. This will be a major issue in defining the semantics of $\mathcal{A}^{\infty}_{T}$. Another issue is that domain descriptions contain no information about ordering of action occurrences. This ordering information is important, as the above example illustrates. In the new language, we will augment a domain description with information about ordering of events. We use linear temporal logic - a natural candidate formalism - to represent the information about ordering.

To sum up, the new language would representing triggers with the following characteristics: (1) the non-immediacy of triggering; (2) unbounded firing delays; and (3) information about ordering of events in a linear temporal logic.

3 Language $\mathcal{A}^{\infty}_{T}$ for Non-immediate Triggers

Similar to $\mathcal{A}^{0}_{T}$, the language $\mathcal{A}^{\infty}_{T}$ is built upon the high level language $\mathcal{A}$. We extend $\mathcal{A}$ with new features to represent triggers and inhibitions. Note that the domain description sub-language of $\mathcal{A}^{\infty}_{T}$ is syntactically similar to that of $\mathcal{A}^{0}_{T}$. But the similarity between the two languages stops there.

3.1 Syntax

3.1.1 Domain description

The mechanism of changes in the cell are captured by a *domain description* in $\mathcal{A}^{\infty}_{T}$. A domain description $\mathcal{D}$ in $\mathcal{A}^{\infty}_{T}$ is a set of statements called *causal rules*, *triggers* and *inhibitors*.

A causal rule is a statement of the form

$$a \textbf{ causes } f \textbf{ if } f_1, \ldots, f_n \tag{10}$$

where a is an action and f, $f_1, \ldots, f_n$ are fluent literals. The conjunction of $f_1, \ldots, f_n$ is referred to as the precondition of the causal rule, and f is called an effect of the action a with respect to the precondition

$f, f_1, \ldots, f_n$. Intuitively, the causal rule encodes the information that f will become true if the action a happens in a state where all the properties $f_1, \ldots, f_n$ hold.

A trigger rule is a statement of the form

$$f_1, \ldots, f_n \textbf{ triggers } a \tag{11}$$

where $f_1, \ldots, f_n$ are fluent literals and a is an action. The conjunction of $f_1, \ldots, f_n$ is referred to as the precondition of the trigger, and a is called a triggered action. Intuitively, the trigger encodes the information that the action a will happen if all the properties $f_1, \ldots, f_n$ are true in the current state and a is not inhibited. The inhibitions of actions are dictated by inhibitor rules.

An inhibitor rule is a statement of the form

$$f_1, \ldots, f_n \textbf{ inhibits } a \tag{12}$$

where $f_1, \ldots, f_n$ are fluent literals and a is an action. The conjunction of $f_1, \ldots, f_n$ is referred to as the precondition of the trigger. Intuitively, the inhibitor dictates that the action a is prevented from happening in a state where all the properties $f_1, \ldots, f_n$ are true.

3.1.2 Observations

In reasoning about cellular behaviors, we usually start with some initial state. We can have complete or incomplete observations about properties of the initial cellular state. A fluent observation is written in a statement of the form

$$\textbf{initially } f$$

where each f is a fluent literal. An action observation is written in a statement of the form

$$\textbf{initially } a$$

A set of observations **initially** ω_i $(i = 1, \ldots, n)$ will usually be shorten as

$$\textbf{initially } \omega_1, \ldots, \omega_n$$

Intuitively, the statement "**initially** f" means that f is true in the initial state. The statement "**initially** a" means that the action a occurs at the initial state. A set $\mathcal{O}$ of observations is *complete*, if for any fluent f, either "**initially** $\neg f$" or "**initially** f" belongs to $\mathcal{O}$.

Definition 3.1. Let $\mathcal{D}$ be a domain description and $\mathcal{O}$ be a set of observations. A state s is an *initial state* in $\mathcal{D}$ corresponding to $\mathcal{O}$ iff for all observation "**initially** f " of the set $\mathcal{O}$, f is in s.

3.1.3 Event orderings

Let $\mathcal{D}$ be a domain description. An *event* is a set of fluent literals (i.e., *state event*), or a set of actions (i.e., *action event*). A state event is said to happen when its elements hold. An action event is said to happen when its elements occur. An *event ordering* is a statement of the form:

$$e \quad \textbf{restricts } e_1 \ \ \mathbf{op} \ \ e_2$$

where e, e_1 and e_2 are events, and $\mathbf{op} \in \{\prec, \parallel, \preceq\}$. Intuitively, the statement encodes that if e happens, then the earliest happening of e_1 and e_2 after e must obey the ordering $\mathbf{op}$. Here $\prec$ means *earlier*, $\parallel$ means *at the same time* and $\preceq$ means either $\prec$ or $\parallel$.

3.1.4 Queries

We are interested in queries about properties of cellular changes through time. It is natural to formulate these queries in linear temporal logic (see Appendix *Technical Background*).

Definition 3.2 (Action theory)**.** A *action theory* in $\mathcal{A}_T^\infty$ is a triple $(\mathcal{D}, \mathcal{E}, \mathcal{I})$, where $\mathcal{D}$ is an *domain description*, $\mathcal{E}$ is an *event ordering specification*, and $\mathcal{I}$ is a set of observations.

3.2 Semantics

We define the semantics of $\mathcal{A}_T^\infty$ in the following steps. First, we define *trajectories* of a domain description in $\mathcal{A}_T^\infty$. Next, we define *interpretations* of a theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$: an interpretation is a trajectory with respect to $\mathcal{D}$ that satisfies certain properties with respect to $\mathcal{E}$ and $\mathcal{I}$. Finally, we show how a *model* of a theory is chosen among its interpretations.

Intuitively, a domain description determines the possible trajectories, along which the cell change from a state to another due to actions. A trajectory is a sequence of states and action occurrences. Causal laws (statements of the form (10)) define how the cell changes from one state to another state due to an action occurrence. Trigger and inhibitor statements (of the forms (11)-(12)) define what actions are triggered to happen.

3.2.1 Trajectories of a domain description

A state transition is a change of one state to another state due to effects of some actions. The effect of an action a in a state s is the set

$$E_{\mathcal{D}}(a, s) = \{\ f \mid a \textbf{ causes } f \textbf{ if } \ f_1, \ldots f_n \in \mathcal{D} \text{ and } \{f_1, \ldots, f_n\} \subseteq s\ \}.$$

The effect of a set A of actions in a state s is the set

$$E_{\mathcal{D}}(A, s) = \bigcup_{a \in A} E_{\mathcal{D}}(a, s).$$

Let $\neg\neg g = g$ and $\neg E_{\mathcal{D}}(A, s) = \{\neg g | g \in E(A, s)\}$.

State transitions of a domain description $\mathcal{D}$ are computed by transition function $\Phi_{\mathcal{D}}$.

Definition 3.3 (Transition function)**.** A transition function of a domain $\mathcal{D}$ is a function $\Phi_{\mathcal{D}}$ from pairs of a set of actions and a state into states such that:

- if $E_{\mathcal{D}}(A, s)$ is consistent, then

$$\Phi_{\mathcal{D}}(A, s) = (s \setminus \neg E_{\mathcal{D}}(A, s)) \cup E_{\mathcal{D}}(A, s) \tag{13}$$

- otherwise $\Phi_{\mathcal{D}}(A, s)$ is undefined. □

We shall use the notation $\Phi(A, s)$ instead of $\Phi_{\mathcal{D}}(A, s)$ if there is only one domain description $\mathcal{D}$ in discussion.

A *transition sequence* σ is a sequence of the form $\sigma = \langle s_0, A_0, s_1, A_1, \ldots\rangle$, where s_i's are states and A_j's are sets of actions in $\mathcal{D}$, such that $s_{i+1} = \Phi(A_i, s_i)$ for all i ; and $\mathsf{A}_j(\sigma) = \emptyset$ for all $j > i$ if $\mathsf{A}_i(\sigma) = \emptyset$. Given any transition sequence $\sigma = \langle s_0, A_0, s_1, A_1, \ldots\rangle$, let us define that $\mathsf{s}_i(\sigma) = s_i$, $\mathsf{A}_i(\sigma) = A_i$ and $\sigma[m, n] = \langle s_m, A_m, s_{m+1}, A_{m+1}, \ldots, A_{n-1}, s_n\rangle$.

Definition 3.4. (Length of finite transition sequence) Let $\mathcal{D}$ be a domain description. A transition sequence σ of $\mathcal{D}$ is *finite* iff there exists $\mathsf{A}_i(\sigma) = \emptyset$. The length of a finite transition sequence σ, denoted $|\sigma|$, is the smallest i such that $\mathsf{A}_i(\sigma) = \emptyset$. □

Example 3.1. Let D be the domain description consisting of the rules:

$$f \textbf{ triggers } a$$
$$f \textbf{ triggers } b$$
$$b \textbf{ causes } \neg f$$

Then $\Phi(a, \{\neg f\}) = \{\neg f\}$, $\Phi(a, \{f\}) = \{f\}$ and $\Phi(b, \{\neg f\}) = \Phi(b, \{f\}) = \{\neg f\}$. Some infinite transition sequences of $\mathcal{D}$ includes:

$$\langle \{\neg f\}, \{a\}, \{\neg f\}, \{a\}, \ldots \rangle$$
$$\langle \{f\}, \{b\}, \{\neg f\}, \{b\}, \{\neg f\}, \{b\}, \ldots \rangle$$
$$\langle \{f\}, \{a, b\}, \{\neg f\}, \{a\}, \{\neg f\}, \{b\}, \{\neg f\}, \{a\}, \ldots \rangle$$

Some finite transition sequence of $\mathcal{D}$ includes:

$$\langle \{f\}, \{a\}, \{f\}, \{a, b\}, \{\neg f\}, \emptyset, \ldots \rangle$$
$$\langle \{f\}, \{b\}, \{\neg f\}, \{a\}, \{\neg f\}, \emptyset, \ldots \rangle$$
$$\langle \{f\}, \{a, b\}, \{\neg f\}, \{a\}, \emptyset, \ldots \rangle$$

All these transition sequences has length 2. □

Definition 3.5 (Triggering/inhibition by state)**.** Let $\mathcal{D}$ be a domain description and σ be a transition sequence of $\mathcal{D}$. An action a is said to be *inhibited* by a state $\mathsf{s}_i(\sigma)$ in σ, if there exists an inhibitor rule "$f_1, \ldots, f_n$ **inhibits** a " such that all the fluent literals $f_1, \ldots, f_n$ hold in $\mathsf{s}_i(\sigma)$. An action b is said to be *triggered* by a state $\mathsf{s}_j(\sigma)$ in σ, if b is not inhibited by $\mathsf{s}_j(\sigma)$ in σ and if there exists a trigger rule "$g_1, \ldots, g_m$ **triggers** b " such that all the fluent literals $g_1, \ldots, g_m$ hold in $\mathsf{s}_j(\sigma)$. □

Intuitively, transition sequences of a domain can be considered as its interpretations. Trajectories of a domain can be considered as models of the domain. Similarly as with $\mathcal{A}_T^0$, we shall define trajectories to be transition sequences in which an action occurs iff it is triggered. The difference in $\mathcal{A}_T^\infty$ is when an action will occur if it has been triggered. Moreover, since trajectories are models, they should be minimal in some sense.

Definition 3.6 (Trajectory)**.** A trajectory is a transition sequence σ that satisfies all the following conditions:

(i) For all i, for all actions a triggered by the state $\mathsf{s}_i(\sigma)$, there exists $j \geq i$ such that $a \in \mathsf{A}_j(\sigma)$.

(ii) For all i and for all actions a, the action a is triggered by states among the states $\mathsf{s}_j(\sigma)$ $(0 \leq j \leq i)$ at least as many times as it occurs in the sets $\mathsf{A}_k(\sigma)$ $(0 \leq k \leq i)$. Formally, for all i and a:

$$\sum_{j=0}^{i} \delta(\mathsf{s}_j(\sigma) \text{ triggers } a) \geq \sum_{j=0}^{i} \delta(a \in \mathsf{A}_j(\sigma))$$

where $\delta(P) = 1$ if the proposition P is true, otherwise $\delta(P) = 0$. The condition *(i)* and *(ii)* are respectively called the *occurrence criterion* and *prefix criterion* of trajectories. □

We shall attempt to provide some more intuitions for the conditions *(i)-(ii)* in Definition 3.6. The *trigger* criterion is reasonable: if an action is triggered at some state then it must occur either immediately or at some later state. The condition *(ii)* captures the intuition that if an action occurs then it must have been triggered. It is called prefix criterion, since we will show later on that any finite transition sequence σ is a prefix of some trajectory iff σ satisfies the condition *(ii)*. Furthermore, since we now deal with non-immediate triggering, the prefix criterion *can not* be substituted with the seemingly simpler and straightforward condition:

(II) *For all i and for all actions a in $\mathsf{A}_i(\sigma)$: there exists some $0 \leq j \leq i$ such that the action a is triggered by the state $\mathsf{s}_j(\sigma)$.*

The difference between the conditions *(ii)* and (II) is illustrated in the next example.

Example 3.2. Let D be a domain description of actions a, b, c, d and fluents f_a, f_b, f_c, f_d and g.

$$\begin{aligned} D = \{ & \neg f_a \textbf{ triggers } a; \quad a \textbf{ causes } f_a; \\ & a \textbf{ causes } \neg g \textbf{ if } g; \quad a \textbf{ causes } g \textbf{ if } \neg g; \\ & g, f_a, \neg f_b \textbf{ triggers } b; \quad b \textbf{ causes } f_b; \\ & g, f_a, f_b, \neg f_c \textbf{ triggers } c; \quad c \textbf{ causes } f_c; \\ & \neg g, f_a, f_b, \neg f_d \textbf{ triggers } d; \quad d \textbf{ causes } f_d \ \} \end{aligned}$$

To simplify notation, we represent states by their subsets of positive fluent literals. Let us consider the transition sequences σ_1 and σ_2 as follows.

$$\begin{aligned} \sigma_1 =& \langle \{\}, \{a\}, \{f_a, g\}, \{b\}, \{f_a, f_b, g\}, \{c\}, \{f_a, f_b, f_c, g\} \rangle \\ \sigma_2 =& \langle \{\}, \{a\}, \{f_a, g\}, \{a, b\}, \{f_a, f_b\}, \{d\}, \{f_a, f_b, f_d\} \rangle \end{aligned}$$

By Definition 3.6, the sequence σ_1 is a trajectory but σ_2 is not. If the condition (II) were used in place of the condition *(ii)*, then both σ_1 and σ_2 would be considered to be trajectories. However, the second occurrence of a in the "trajectory" σ_2 would not be supported (i.e., not be triggered). It is the first occurrence of a, not the second occurrence that can be said to be triggered by the initial state $\{\}$. The second state $\{f_a, g\}$ does not trigger a, so the second occurrence of a has not been triggered. □

A domain description can have an infinite number of trajectories which are all finite, as shown in the following example.

Example 3.3. Consider the domain description D in Example 3.1. There is an infinite number of trajectories of D. The only trajectory starting from the initial state $\{\neg f\}$ is $\langle \{\neg f\} \rangle$. The trajectories starting from the initial state $\{f\}$ include:

$$\begin{aligned} \sigma =& \langle \{f\}, \{b\}, \{\neg f\}, \{a\}, \{\neg f\} \rangle \\ \sigma_0 =& \langle \{f\}, \{a, b\}, \{\neg f\} \rangle \\ \sigma_1 =& \langle \{f\}, \{a\}, \{f\}, \{a, b\}, \{\neg f\} \rangle \\ & \cdots\cdots \\ \sigma_n =& \langle \{f\}, \{a\}, \ldots, \{f\}, \{a\}, \{f\}, \{a, b\}, \{\neg f\} \rangle \end{aligned}$$

for any number n. □

3.2.2 Interpretations and models

Intuitively, an interpretation of a theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is a trajectory σ that starts from an initial state described by $\mathcal{I}$ and that satisfies the set $\mathcal{E}$ of event orderings. Formally, an interpretation of a theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is defined as follows.

Let σ be a trajectory of a domain description $\mathcal{D}$. An event e is said to happen at the i^{th} state of the trajectory σ if $e \subseteq \mathsf{s}_i(\sigma) \cup \mathsf{A}_i(\sigma)$. Given a fixed j, the smallest index $i > j$ such that an event e happens at the i^{th} state of the trajectory σ is denoted $first(e, \sigma, j)$.

Definition 3.7 (Interpretation)**.** An interpretation of a theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is a trajectory σ such that:

- For all observation "**initially** f " of the set $\mathcal{I}$ where f is a fluent literal: $f \in \mathsf{s}_0(\sigma)$.
- For all observation "**initially** a " of the set $\mathcal{I}$ where a is an action: $a \in \mathsf{A}_0(\sigma)$.
- For all event ordering " e **restricts** $e_1 \prec e_2$ " in the set $\mathcal{E}$, if e happens at the i^{th} state of τ, and $first(e_j, \tau, i)$ exists $(j = 1, 2)$, then $first(e_1, \tau, i) < first(e_2, \tau, i)$.
- For all event ordering " e **restricts** $e_1 \parallel e_2$ " in the set $\mathcal{E}$, if e happens at the i^{th} state of τ, and $first(e_j, \tau, i)$ exists $(j = 1, 2)$, then $first(e_1, \tau, i) = first(e_2, \tau, i)$.
- For all event ordering " e **restricts** $e_1 \preceq e_2$ " in the set $\mathcal{E}$, if e happens at the i^{th} state of τ, and $first(e_j, \tau, i)$ exists $(j = 1, 2)$, then $first(e_1, \tau, i) \leq first(e_2, \tau, i)$.

The set of all interpretations of $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is denoted $int(\mathcal{T})$. □

Example 3.4. Let us continue with Example 3.3. Let $I = \{$**initially** $f\}$. Then all the trajectories σ, σ_i $(i \geq 0)$ are interpretation of the action theory $(D, \emptyset, I)$. For all $i \geq 1$, the trajectories σ_i is an interpretations of $(D, \{f$ **restricts** $a \prec b\}, I)$. The trajectory σ and σ_0 is the unique interpretation of $(D, \{f$ **restricts** $b \prec a\}, I)$ and of $(D, \{f$ **restricts** $a \parallel b\}, I)$, respectively. Finally, the action theory $(D, \emptyset, I \cup \{$**initially** $b\})$ has the unique interpretation σ. □

Recall that one of our goals in having the new language $\mathcal{A}_T^\infty$ is to capture the non-immediacy of triggering. Assume that an action theory in $\mathcal{A}_T^\infty$ contains a trigger rule "f **triggers** a ". If f holds in some state s, then the occurrence of a can be delayed at length after the state s. Non-immediacy of triggering may lead to infinite number of interpretations of the occurrence of a.

Example 3.5. Consider the action theory $T = (D, \emptyset, I)$ where D is the domain description in Example 3.1 and $I = \{$**initially** $f\}$. Then all the trajectories described in Example 3.3 are interpretations of T. Thus T has an infinite number of interpretations.

A closer look into Example 3.5 reveals that the infinite number of trajectories is caused by delays of the occurrence of the action b. Although b is triggered at the initial state, it is allowed to occur at any time later. A trajectory ends as soon as b occurs. Among these trajectories, σ_0 seems to be the best candidate for a model. The reason is that σ_0 corresponds to the case where b occurs at the earliest possible time. Thus we prompted to define the semantics of $\mathcal{A}_T^\infty$ that captures an additional intuition of triggering: *an action is to occur as soon as possible after having been triggered.* The intuition shall be formulated based on a partial order of transition sequences.

Definition 3.8 (Ordering trajectories)**.** Let $\mathcal{D}$ be a domain description. Let σ and σ' be transition sequences of $\mathcal{D}$. Then $\sigma \sqsubset \sigma'$ iff there exists $0 \leq n$ such that $\sigma[0, n] = \sigma'[0, n]$ and $\mathsf{A}_n(\sigma) \subset \mathsf{A}_n(\sigma')$. □

It is straightforward to verify that $\sqsubset$ is a partial order.

Proposition 3.1. *Let σ, σ' and σ'' be transition sequences of a domain description $\mathcal{D}$ such that $\sigma \sqsubset \sigma'$ and $\sigma' \sqsubset \sigma''$. Then $\sigma \sqsubset \sigma''$.*

The order $\sqsubset$ determines a preference criterion to select models among the set of interpretations of an action theory: a model is a maximally preferred interpretation. Intuitively, if the occurrence of an action is "unnecessarily" delayed in an interpretation, then the interpretation cannot be a model.

Example 3.6. Let σ and σ' be interpretations of an action theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$. Assume that $\sigma \sqsubset \sigma'$. Then there exists n such that $\sigma[0, n] = \sigma'[0, n]$ and $\mathsf{A}_n(\sigma) \subset \mathsf{A}_n(\sigma')$. Let a be an action of the non-empty set $\mathsf{A}_n(\sigma') \setminus \mathsf{A}_n(\sigma)$. Since σ' is a trajectory of $\mathcal{D}$ and $a \in \mathsf{A}_n(\sigma')$, a is triggered at some state $\mathsf{s}_i(\sigma')$ where $i \leq n$. Besides, $\Phi(\mathsf{A}_n(\sigma'), \mathsf{s}_n(\sigma'))$ is defined. Then $\Phi(\mathsf{A}_n(\sigma) \cup \{a\}, \mathsf{s}_n(\sigma))$ is defined, because $\mathsf{A}_n(\sigma) \cup \{a\} \subseteq \mathsf{A}_n(\sigma')$ and $\mathsf{s}_n(\sigma') = \mathsf{s}_n(\sigma)$. Moreover, $\mathsf{s}_i(\sigma') = \mathsf{s}_i(\sigma)$, so a is triggered by $\mathsf{s}_i(\sigma)$. Thus a could have occurred at state $\mathsf{s}_n(\sigma)$ in the trajectory σ, but its occurrence has been delayed to some later time. □

Definition 3.9 (Model)**.** A model of $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is an interpretation σ of $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ such that there exists no interpretation σ' where $\sigma \sqsubset \sigma'$. A theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is called *consistent* if it has at least one model. The set of all models of $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ is denoted $mod(\mathcal{D}, \mathcal{E}, \mathcal{I})$. □

Example 3.7. We continue with Example 3.4. Let us determine the models of the action theory $(D, \emptyset, I)$. All the interpretations of $(D, \emptyset, I)$ are σ and σ_i, for all i. We have that $\mathsf{s}_0(\sigma) = \mathsf{s}_0(\sigma_i)$ for all i. Besides, $\mathsf{A}_0(\sigma) \subset \mathsf{A}_0(\sigma_0)$ and $\mathsf{A}_0(\sigma_i) \subset \mathsf{A}_0(\sigma_0)$ for all $i > 0$. By Definition 3.8, it follows that $\sigma \sqsubset \sigma_0$ and $\sigma_i \sqsubset \sigma_0$ for all $i > 0$. By Definition 3.9, the interpretation σ_0 is the unique model of $(D, \emptyset, I)$. □

3.2.3 Query entailment

As in the case of $\mathcal{A}_T^0$, entailments in $\mathcal{A}_T^\infty$ are defined only for consistent action theories. We define two kinds of entailments: *strong* and *weak* entailment. First, we define the entailment of a query Q by a model of an action theory.

Definition 3.10 (Entailment of query)**.** Let $\mathcal{T} = (\mathcal{D}, \mathcal{E}, \mathcal{I})$ be an action theory. Let σ be a model of $\mathcal{T}$, which is of the form:

$$\sigma = \langle s_0, A_0, s_1, A_1, \ldots, s_n, A_n \ldots \rangle$$

Let I_σ be the sequence of the states s_i's:

$$I_\sigma = \langle s_0, s_1, \ldots, s_n, \ldots \rangle$$

Then σ entails a query Q iff I_σ entails Q with respect to the standard linear temporal logic (LTL) semantics. □

Definition 3.11. An action theory $\mathcal{T} = (\mathcal{D}, \mathcal{E}, \mathcal{I})$ *strongly entails* a query Q, denoted $\mathcal{T} \models_s Q$, iff $\mathcal{T}$ is consistent and all the models σ of $\mathcal{T}$ entails Q. An action theory $\mathcal{T} = (\mathcal{D}, \mathcal{E}, \mathcal{I})$ *weakly* entails a query Q, denoted $\mathcal{T} \models_w Q$, iff there exists a model σ of $\mathcal{T}$ that entailers Q. □

Recall that $\mathcal{A}_T^\infty$ is motivated by the paradox of non-monotonic refinement (see Introduction section). We shall show that the paradox can be resolved in the framework of $\mathcal{A}_T^\infty$.

Example 3.8. Let $\mathcal{D}^+_{afg}$ be the updated version of the domain $\mathcal{D}_{afg}$ in Section 2.3 (based on the rules (7)-(9)).

$$\begin{aligned}\mathcal{D}^+_{afg} = \{\ \neg f, \neg f' \textbf{ triggers } a'\,; \qquad & a' \textbf{ causes } f'\,;\\ f', \neg g \textbf{ triggers } a\,; \qquad & a \textbf{ causes } f\,;\\ \neg g \textbf{ triggers } b\,; \qquad & b \textbf{ causes } g\ \}\end{aligned}$$

Let $I^+ = \{\textbf{initially } \neg f, \neg f', \neg g\}$. It can be verified that the action theory $\mathcal{T}_1 = (\mathcal{D}^+_{afg}, \emptyset, I^+)$ has the unique trajectory models

$$\sigma_1 = \langle\{\neg f, \neg f', \neg g\}, \{a', b\}, \{\neg f, f', g\}\rangle$$

Since $\sigma_1 \models \Box\, \neg f$, it follows that $\mathcal{T}_1 \models_s \Box\, \neg f$. Now let E be the event ordering:

$$E = \{\neg f, \neg f', \neg g\} \textbf{ restricts } a' \prec a$$

The event ordering E states that a must occur if $\neg f$, $\neg g$ hold and a' must occur before a. Intuitively, the event ordering captures information missing from the refinement of the trigger rule "$\neg f, \neg g$ **triggers** a ". The missing of this information leads to the paradox of non-monotonic refinement. Incorporating E to the action theory $\mathcal{T}_1$ provides a solution for the paradox. Indeed, let $\mathcal{T}_2 = (\mathcal{D}^+_{afg}, \{E\}, I)$. Then $\mathcal{T}_2$ has the unique model σ_2:

$$\sigma_2 = \langle\{\neg f, \neg f', \neg g\}, \{a'\}, \{\neg f, f', \neg g\}, \{a, b\}, \{f, f', g\}\rangle$$

Since $\sigma_2 \models \Diamond\, f$, it follows that $\mathcal{T}_2 \models_s \Diamond\, f$. □

The weak entailment is motivated by the non-deterministic nature of cellular signaling networks. The non-determinism manifests itself in a cellular signaling phenomenon called *specificity* [36]. In general, a signaling network may respond to the same output in different ways. The actual output depends not only on the structure of the signaling network but specific tissue or cell-line or physical conditions. In light of our action formalisms, this biological phenomenon is due to incompleteness information about an initial state or about a domain description. Consequently, there are multiple models of the action theory, each of which corresponds to a specific biological situation.

3.3 Reasoning with Non-immediate Triggers

It is important to be able to reason about effects of external agent actions on triggered evolutions of a system; for example, we want to know how to intervene to alter the cell behavior in desirable ways. Since $\mathcal{A}^\infty_T$ does not represent explicit time, we cannot specify an execution of an exogenous action by explicitly setting a time for it (e.g., exogenous actions with time steps, or implicitly setting the execution time using ordering (e.g., a plan as a sequence of actions). However, we can combine the modeled system (e.g., an interaction network) together with the agent control (e.g., a scientist) into one system then reason about the exogenous actions of the agent as triggers in the combined system.
Formally, we consider external interventions as conditional actions of the form:

$$\alpha = \textbf{ do } a \textbf{ if } f_1, f_2, \ldots, f_n,$$

where a is an action, and $f_1, f_2, \ldots, f_n$ are fluent literals. Given a conditional action α of the above form, we denote $trig(\alpha) = f_1, f_2, \ldots, f_n$ **triggers** a and $act(\alpha) = a$. Intuitively, $trig(\alpha)$ is the translation of α into a trigger rule. We use the same notations for sets of conditional actions. Given a set C of conditional actions, we denote that $trig(C) = \{trig(\alpha) \mid \alpha \in C\}$ and $act(C) = \{act(\alpha) \mid \alpha \in C\}$.

An *agent control* can be represented as a pair (C, L), where C is a set of conditional actions and L is a set of dynamic causal laws describing effects of the actions in $act(C)$. Let $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ be a theory in $\mathcal{A}_T^\infty$ and (C, L) be an agent control. We can assume that the actions of $act(C)$ are not in $\mathcal{D}$. *Planning* for a goal Q is to find a subset $P \subseteq C$ such that:

$$(\mathcal{D} \cup L \cup trig(P), \mathcal{E}, \mathcal{I}) \models_s Q \ .$$

4 Monotonic Refinement

4.1 Refinement of Trigger Rules

In the first section, we have provided some intuition about refinement of triggers. We presented several examples of refinements in the p53 signaling networks. In this section, we formalize a notion of refinement of triggers and study some of its properties. Particularly, we are interested in cases where the refinement is monotonic.

Definition 4.1 (Refinement of trigger rule)**.** Let r be the trigger rule $f_1, \ldots, f_n$ **triggers** a. Let D_r be a domain description that does not contain inhibitor rules and that does not contain causal rules for the action a. Let I_r be a set of observations in D_r. Let f_a be an additional fluent symbol that is different from the fluents of D_r. Let us denote $D'_r = D_r \cup \{\ a$ **causes** $f_a\ ;\ f_a$ **inhibits** $a\ \}$ and $I'_r = I_r \cup \{$ **initially** $f_1, \ldots, f_n\ \}$. Finally, let T_r be the action theory $(D'_r, \emptyset, I'_r)$. The pair $\langle D_r, I_r \rangle$ is a refinement of r iff:

(i) the theory T_r is consistent and all the models of T_r are finite trajectories;

(ii) any model σ_r of T_r is of the form

$$\sigma_r = \langle s_0, A_0, \ldots, A_{i-1}, s_i, \{a\}, s_{i+1} \rangle$$

where $a \notin A_j$ for all $0 \leq j \leq i-1$; and

(iii) no action of D_r affects any fluent in $\{f_1, \ldots, f_n\}$. □

Intuitively, D_r and I_r give more details about how a is triggered from the precondition $f_1, \ldots, f_n$. The domain D_r and the observation set I_r can be regarded as missing knowledge which had not been available when we formulated the trigger rule r. The condition *(ii)* implicates that a is "indirectly" triggered from $f_1, \ldots, f_n$. The condition *(iii)* captures the intuition that $f_1, \ldots, f_n$ are the essential conditions to trigger a. The fluent f_a and the related causal and inhibitor rule are to eliminate the infinite interpretations in which the action a is triggered infinitely. The following example helps to clarify Definition 4.1.

Example 4.1. We describe the refinement of the trigger rule r in the motivating example (Section 2.3). Recall that:

$$r = \neg f, \neg g \textbf{ triggers } a$$

Let D_r be the domain description:

$$D_r = \begin{cases} \neg f, \neg g \textbf{ triggers } a' \\ a' \textbf{ causes } f' \\ f', \neg g \textbf{ triggers } a \end{cases}$$

Let $I_r = \{\textbf{initially } \neg f'\}$. Then $\langle D_r, I_r\rangle$ is a refinement of r. Indeed, as in Definition 4.1, let $D'_r = D_r \cup \{a \textbf{ causes } f_a ; f_a \textbf{ inhibits } a\}$ and $I'_r = I_r \cup \{\textbf{initially } \neg f, \neg g\} = \{\textbf{initially } \neg f, \neg g, \neg f'\}$. Given a fluent literal l_a of f_a (i.e., $l_a \in \{f_a, \neg f_a\}$), let $s_0 = \{\neg f, \neg g, \neg f', l_a\}$, $s_1 = \{\neg f, \neg g, f', l_a\}$ and $s_2 = \{\neg f, \neg g, f', f_a\}$. Then s_0, s_1, s_2 are states in D'_r. We have that: s_0 triggers a', s_1 trigger a and $\Phi_{D'_r}(a', s_0) = s_1$, $\Phi_{D'_r}(a, s_1) = s_2$. There is no action triggered in s_2. It can be verified that the trajectories $\sigma_{l_a} = \langle s_0, \{a'\}, s_1, \{a\}, s_2\rangle$ (for $l_a \in \{f_a, \neg f_a\}$) are all the models of the theory $T_r = (D_r, \emptyset, I_r \cup \{\textbf{initially } \neg f, \neg g\})$. Since $\sigma_{l_a} \models \Diamond a$ and $\sigma_{l_a} \Box (\neg f \wedge \neg g)$, we have that $T_r \models \Diamond a$ and $T_r \models \Box (\neg f \wedge \neg g)$. Thus all the condition (i)-(iii) of Definition 4.1 are satisfied $\langle D_r, I_r\rangle$ $\Box$

The following proposition deals with a generalization of the refinement in Example 4.1.

Proposition 4.1. *Let r be the trigger rule F, G **triggers** a, where F and G are sets of fluent literals, which are not necessarily disjoint. Let D_r be the domain description:*

$$D_r = \begin{cases} F, H_0 \textbf{ triggers } a' \\ a' \textbf{ causes } H_1 \\ H_2, G \textbf{ triggers } a \end{cases}$$

where H_is are sets of fluent literals such that

$$\begin{aligned} &\exists f : f \in H_0 \wedge \neg f \in H_1 \\ &H_1 \subseteq H_2 \\ &(F \cup G) \cap (H_0 \cup H_2) = \emptyset \end{aligned}$$

Let I_r be the set of observations

$$I_r = \{\textbf{initially } f \mid f \in H_0 \cup (H_2 \setminus H_1)\}$$

Then $\langle D_r, I_r\rangle$ is a refinement of r.

Intuitively, a refinement of trigger rule introduces new interactions between actions and their effects. Due to the new interactions, the modified action theory may have models that are different from those of the original action theory. This difference results in contradictory entailments such as those discussed in the motivating example (Section 2.3). Event ordering information indirectly constraints the possible interactions. The following proposition shows that the models of the original action theory can always be found in among the models of the modified action theory, given proper event ordering information.

Proposition 4.2. *Let $\mathcal{T}$ be an $\mathcal{A}^{\infty}_{T}$ action theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$. Let r be a trigger rule $f_1, \ldots f_n$ **triggers** a in the domain $\mathcal{D}$. Assume that a is triggered at most once in any trajectory model of $\mathcal{T}$. Let $\langle D_r, I_r\rangle$ be a refinement of r such that the common alphabet symbols of D_r and $\mathcal{D}$ are the fluent symbols $f_1, \ldots, f_n$ and the action symbol a. Let $\mathcal{D}^+$ denote the domain description:*

$$\mathcal{D}^+ = \mathcal{D} \setminus \{r\} \cup D_r \ .$$

If a query Q is strongly entailed by $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ then Q is weakly entailed by the theory $(\mathcal{D}^+, \mathcal{E} \cup E_1 \cup E_2, \mathcal{I} \cup I_r)$, with E_i being the event orderings:

$$\begin{aligned} E_1 &= \{\ \{f_1, \ldots, f_n\} \textbf{ restricts } b_1 \prec b_2 \mid b_1 \in act(D_r) \setminus \{a\}, b_2 \in act(\mathcal{D})\ \} \\ E_2 &= \{\ lit(I_r) \textbf{ restricts } \{f_1, \ldots, f_n\} \preceq b \mid b \in act(D_r)\ \} \end{aligned}$$

where $act(D_r)$ and $act(\mathcal{D})$ denote the sets of the actions in D_r and $\mathcal{D}$ respectively, and $lit(I_r)$ denotes the set of the fluent literals in I_r.

Example 4.2. Let us apply Proposition 4.2 to the refinement of trigger in the motivating example (Section 2.3). Here we have that $\mathcal{D} = \mathcal{D}_{afg}$, $\mathcal{E} = \emptyset$ and $\mathcal{I} = \{$**initially** $\neg f,$ **initially** $\neg g\}$. The trigger rule r is:

$$r = \neg f, \neg g \textbf{ triggers } a \ .$$

Let $Q = \ \diamond f$. Then $(\mathcal{D}_{afg}, \mathcal{E}, \mathcal{I}) \models_s Q$. The refinement $\langle D_r, I_r \rangle$ of r has been described in Example 4.1. We have that: $act(D_r) = \{a, a'\}$ and $act(\mathcal{D}) = \{a, b\}$. Let E_1 and E_2 be the event orderings in Proposition 4.2, then:

$$E_1 = \{\ \{\neg f, \neg g\} \textbf{ restricts } a' \prec a;\ \{\neg f, \neg g\} \textbf{ restricts } a' \prec b\ \}$$

and

$$E_2 = \{\ \{\neg f'\} \textbf{ restricts } \{\neg f, \neg g\} \preceq a;\ \{\neg f'\} \textbf{ restricts } \{\neg f, \neg g\} \preceq a'\ \} \ .$$

By Proposition 4.2, we have that:

$$(\mathcal{D}_{afg} \setminus \{r\} \cup D_r, E_1 \cup E_2, \mathcal{I} \cup I_r) \models_w \ \diamond f \ .$$

4.2 Refinement of Causal Rules

A general causal rule r has the form "a **causes** f **if** F", where F is a set of fluent literals. A refinement of r gives more details of how a affects f. It is usually the case that a has some effects (which may or may not include f) that triggers an immediate action b. This immediate action b in turn causes f. Thus we may also say that a causes f indirectly through the triggering of b. The following definition captures this intuition.

Definition 4.2 (Refinement of causal rule)**.** Let $r = a$ **causes** f **if** F be a causal rule of a domain description $\mathcal{D}$. Let G, G_1, G_2 be non-empty disjoint sets of fluent literals that are not in $\mathcal{D}$. Let b be an action that is not in $\mathcal{D}$. A refinement of r with respect to $\mathcal{D}$ is a pair (D_r, I_r) where D_r is the following set of rules:

$$D_r = \begin{cases} a \textbf{ causes } G \textbf{ if } F, G_1 \\ G, G_2 \textbf{ triggers } b \\ b \textbf{ causes } f \end{cases}$$

and I_r is the set of initial observations:

$$I_r = \{\textbf{initially } \neg g \mid g \in G\} \cup \{\textbf{initially } g \mid g \in G_1\} \cup \{\textbf{initially } g \mid g \in G_2\}$$

Given a refinement (D_r, I_r) of the rule r defined above, r can be refined further by means of refinements of the trigger rule G, G_2 **triggers** b. Proposition 4.2 helps us in maintaining the monotonicity of trigger refinement. Thus to maintain the monotonicity of causal rule refinement, it is sufficient that we handle the monotonicity of the refinement given in Definition 4.2. For that purpose, we have the following result.

Proposition 4.3. *Let $\mathcal{T} = (\mathcal{D}, \mathcal{E}, \mathcal{I})$ be an $\mathcal{A}^{\infty}_{\mathcal{T}}$ theory. Let $r \in \mathcal{D}$ be a causal rule of the form "a* **causes** *f* **if** *F", where F is a set of fluent literals. Let (D_r, I_r) be the refinement of r given in Definition 4.2. If a query Q is strongly entailed by the theory $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ then Q is weakly entailed by the theory $(\mathcal{D} \setminus \{r\} \cup D_r, \mathcal{E} \cup E, \mathcal{I} \cup I_r)$, with E being the ordering:*

$$E = \{\{a\} \cup F \textbf{ restricts } b \prec act(\mathcal{D})\}$$

where $act(\mathcal{D})$ is the set of the actions in $\mathcal{D}$.

An example of the refinement of causal rules can be found later in Section 4.3.3.

4.3 Refinements in p53 Signaling Network

Before considering the biological example, let us go over the following short glossary:

protein: the major macromolecular constituent of cells.

tumor suppressor: a protein that works to prevent cells from turning into tumors.

carcinogen: substance that directly causes or facilitates cancer development.

gene expression: the production of proteins from their genes.

upregulation: the processing of increasing gene expression.

4.3.1 Overview of p53 Network

The p53 protein is a tumor suppressor that plays a key role in the regulation of the cell growth and cell death. It is estimated that about one half of human cancers contains mutant p53. It is also hypothesized that the p53 network is affected in the majority of the remaining tumors. Normally, a wild-type (i.e., not mutant) p53 functions to prevent cancer as follows (the solid lines in Figure 2). Stimuli such as UV, ionizing radiation or chemical carcinogens can induce DNA damage. DNA damage will lead to genomic instability, which in turn triggers uncontrolled cell growth (i.e., tumor formation). However, the stimuli also upregulate the gene expression of wild-type p53. The upregulated gene expression produces high levels of p53 concentration, which suppresses abnormal cell growth thus preventing cancer [15].

4.3.2 The initial knowledge base

The p53 network can be represented in $\mathcal{A}_T^0$ as follows.

$$\begin{aligned}\mathcal{D}_{p53} = \{\ & high(UV) \textbf{ triggers } damage(DNA)\\ & damage(DNA) \textbf{ causes } instable(cells)\\ & instable(cells), \neg high(p53) \textbf{ triggers } proliferate(cells)\\ & proliferate(cells) \textbf{ causes } tumorous\\ & high(UV), \neg mutant(p53) \textbf{ triggers } upregulate(p53)\\ & upregulate(p53) \textbf{ causes } high(p53)\}\end{aligned}$$

Given that only $high(UV)$ is true at time 0, actions $damage(DNA)$ and $upregulate(p53)$ will occur at time 0. Then $instable(cells)$ becomes true at time 1. Because $high(p53)$ is false, the action $proliferate(cells)$ is not triggered. As expected, the domain $\mathcal{D}_{p53}$ predicts the p53 prevention of cancer (Fig. 2). Formally, let $\mathcal{O}$ be the set of initial observations:

$$\mathcal{O} = \{\textbf{initially } high(UV), \textbf{initially } \neg instable(cells), \textbf{initially } \neg tumorous, \textbf{initially } \neg high(p53)\} \quad (14)$$

then we have that:

$$(\mathcal{D}_{p53}, \mathcal{O}) \models \neg tumorous \quad (15)$$

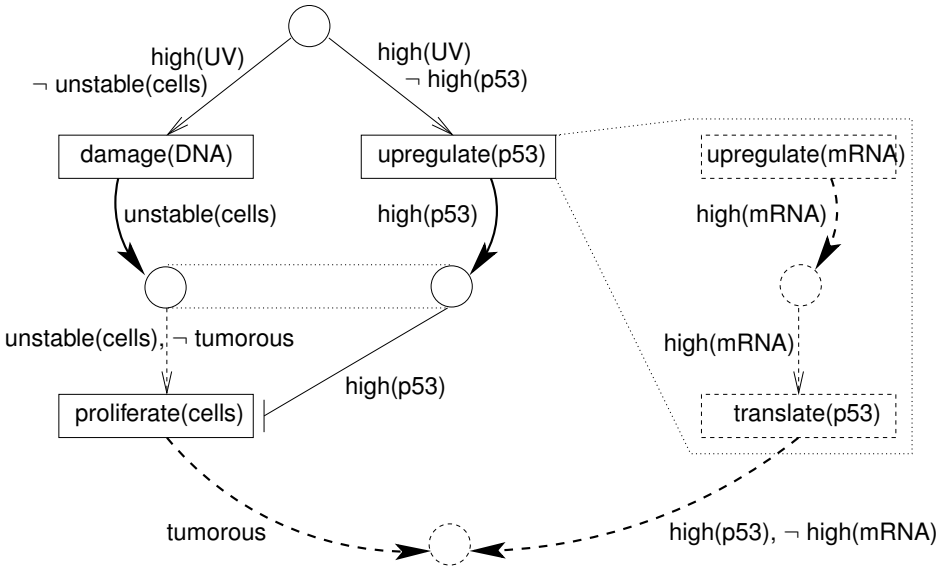

Figure 2: Detailed knowledge of p53 upregulation predicts cancer.

4.3.3 Refinement of more detailed knowledge

Each of the actions $damage(DNA)$, $proliferate(cells)$ and $upregulate(p53)$ represents a complex biological process. For example, let us update the network with the following refinement of the p53 upregulation [15]. The process first starts with the upregulation of the p53 mRNA, which causes a high level of mRNA. The high mRNA level induces translation of p53, which increases the quantity of p53 protein. Thus, to specify at a finer level, we replace the causal rule

$$upregulate(p53) \textbf{ causes } high(p53) \tag{16}$$

with the set of rules:

$$\begin{cases} upregulate(p53) \textbf{ causes } high(mRNA) \\ high(mRNA) \textbf{ triggers } translate(p53) \\ translate(p53) \textbf{ causes } high(p53) \end{cases}$$

(The refinement is drawn in dashed lines in Figure 2).

But unlike $\mathcal{D}_{p53}$, the updated domain description, which we refer to as $\mathcal{D}^{+}_{p53}$, does not predict that p53 prevents cancer. Assuming $high(UV)$ is true at 0, the actions occurring at 0 are $damage(DNA)$ and $upregulate(mRNA)$. Then $high(p53)$ remains false while $instable(cells)$ becomes true at time 1. Thus *proliferate*$(cells)$ is triggered at time 1. When $high(p53)$ becomes true at time 2, it is "too late" to block the occurrence of *proliferate*$(cells)$. The action *proliferate*$(cells)$ causes $tumorous$ to be true. It follows that:

$$(\mathcal{D}^{+}_{p53}, \mathcal{O}) \models tumorous \tag{17}$$

where $\mathcal{O}$ is the observation set in (14). This biological example illustrates that $\mathcal{A}^{0}_{T}$ does not gracefully deal with refinement of a causal rule.

4.3.4 Reasoning in the presence of refinements

Let $\mathcal{D}^{+}_{p53}$ be the updated version of $\mathcal{D}_{p53}$ in Let $\mathcal{I}$ be an initial state observation that only $high(UV)$ is true.

First, we have that $(\mathcal{D}_{p53}, \emptyset, \mathcal{I}) \models_s \ \Box \neg tumorous$. This entailment "corresponds" to the entailment in (15). Next, let $\mathcal{H}_0 = (\mathcal{D}^{+}_{p53}, \emptyset, \mathcal{I})$. We have that

$$(\mathcal{D}^{+}_{p53}, \emptyset, \mathcal{I}) \models_w \ \Box \neg tumorous \ ,$$

and that

$$(\mathcal{D}^{+}_{p53}, \emptyset, \mathcal{I}) \models_w \ \Box \ tumorous \ .$$

Thus the theory $\mathcal{H}_0$ can predict both contradictory conclusions derived from (15) and (17).

Now let $\mathcal{E}$ be the event ordering

$$high(UV), \neg mutant(p53) \textbf{ restricts } high(p53) \preceq instable(cells) \ .$$

This event ordering knowledge has an known biological counterpart. It is known that in a healthy cellular environment, the upregulation of p53 is triggered rapidly in the presence of high level UV and before the cell becomes instable.

Now let $\mathcal{H}_1 = (\mathcal{D}^{+}_{p53}, \mathcal{E}, \mathcal{I})$, then $\mathcal{H}_1 \models_s \ \Box \neg tumorous$. Thus one can maintain the consistency of the knowledge based (with respect to refinements) by incorporating event orderings.

5 Complexity Analysis

Example 3.3 gives an example of action theory that has an infinite number of trajectories, all of which are finite trajectories. There also exists action theories that has an infinite number of infinite trajectories.

Example 5.1. Let $D_{0,1}$ be the domain:

$$D_{0,1} = \begin{cases} f \textbf{ triggers } a_0 \\ f \textbf{ triggers } a_1 \\ a_0 \textbf{ causes } g \\ a_1 \textbf{ causes } \neg g \end{cases}$$

Let $I = \{f, g\}$ and $E = \emptyset$. Then any infinite sequence of actions a_0 and a_1 correspond to a model of (D, E, I). It is interesting to note that such an action sequence also correspond to the binary representation of an irrational number in $[0, 1]$, where a_0 represents the digit 0 and a_1 represents the digit 1. □

We shall be interested in analyzing the complexity of action theories whose models have lengths bounded by a polynomial of the size of their respective action theory.

Definition 5.1 (Size of domain description)**.** Let $\mathcal{D}$ be a domain description in $\mathcal{A}^{\infty}_{T}$. The size of $\mathcal{D}$, denoted $\|\mathcal{D}\|$ is the total of the number of the symbols in the alphabet of $\mathcal{D}$ and the number of the rules in $\mathcal{D}$. □

Definition 5.2 (Polynomial-time bounded action theory)**.** Let $\mathcal{T} = (\mathcal{D}, \mathcal{E}, \mathcal{I})$ be an action theory in $\mathcal{A}^{\infty}_{T}$. Then $\mathcal{T}$ is a polynomial-time bounded action theory iff there exists a polynomial p such that $|\mathcal{E}| < p(\|\mathcal{D}\|)$ and all trajectories of $\mathcal{D}$ have length less than $p(\|\mathcal{D}\|)$. We also say that $\mathcal{T}$ is polynomial-time bounded by p. □

Theorem 5.1. *To decide if* $(\mathcal{D}, \mathcal{E}, \mathcal{I}) \models_w Q$ *is* Σ_2^P*-complete, where* $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ *is a polynomial-time bounded action theory.*

Theorem 5.2. *To decide if* $(\mathcal{D}, \mathcal{E}, \mathcal{I}) \models_s Q$ *is in* Π_2^P*-complete, where* $(\mathcal{D}, \mathcal{E}, \mathcal{I})$ *is a polynomial-time bounded action theory.*

6 Conclusion

We have proposed a solution to the problem of non-deterministic triggering of interactions in biological networks. As done in [38, 4] as well as in many other related works, biological interactions can be modelled to some extent as triggered actions. These approaches implicitly assumed that the triggering of actions is immediate, which can not account for the inherent incompleteness of our knowledge about biological networks. For example, if in a biological network there are the trigger "f **triggers** a" and the trigger "f **triggers** b", it is not always the case that in a state s where f is true both a and b immediately occur. Yet we do not have the luxury of specifying exactly how long it takes before a trigger fires, as that information is usually not available and it may not even be a deterministic duration. In many cases, there is insufficient knowledge for one to say with certainty which of a and b occurs before the other. Thus a representation and reasoning mechanism should model non-immediate triggering as well as event ordering information.

The proposed action language $\mathcal{A}_T^\infty$ has three sub-languages for representing domain description, event ordering, and observations. Given an $\mathcal{A}_T^\infty$ action theory, the semantics of $\mathcal{A}_T^\infty$ selects minimal models out of possibly infinite number of interpretations. The semantics captures our intuition about non-immediate triggered actions. We have provided evidences that $\mathcal{A}_T^\infty$ can deal gracefully with the issue of knowledge refinement. Although $\mathcal{A}_T^\infty$ has the same simple syntax like $\mathcal{A}_T^0$, due to its semantics of non-immediate triggering, the complexity of computing entailments in $\mathcal{A}_T^\infty$ is very high. Approximations of reasoning in $\mathcal{A}_T^\infty$ have been studied and implemented using logic programming [39].

Building a knowledge base for biological networks is undeniably a challenging and important problem. There exist a vast array of approaches to this problem with various advantages and limitations. We advocates one approach based on logic-based methodologies for knowledge representation. Our approach as well as many other related works are very preliminary attempts toward the ultimate solutions. A final solution would probably involve features taken from multiple approaches. For example, differential equation based simulation systems can be used to generate data as well as verify predictions. Experimental and/or simulated data can be represented and processed at different levels of abstraction. Our knowledge base approach is geared towards representation of and reasoning about very high-level knowledge, thus it would need to take advantage of works modeling lower-level knowledge.

A deep underlying issue related to non-immediate triggering is how to formulate causality of events in continuous-timed distributed systems. The language $\mathcal{A}_T^\infty$ can be seen as an attempt for an approximate formulation based on discrete time. As discussed in [32], a well-established formalism for reasoning about causality in distributed system is a holy grail yet to be found.

References

[1] A. R. Meyer and L. J. Stockmeyer. The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space. In. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[2] M. Balduccini, M. Gelfond, M. Nogueira, R. Watson, and M. Barry. An A-Prolog decision support system for the space shuttle. In *AAAI Spring 2001 Symposium*, 2001.

[3] Marcello Balduccini, Michael Gelfond, Monica Nogueira, and Richard Watson. Planning with the USA-Advisor. In *3rd International NASA Workshop on Planning and Scheduling for Space*, 2002.

[4] Chitta Baral, Karen Chancellor, Nam Tran, Nhan Tran, and Michael Berens. A knowledge based approach for representing and reasoning about signaling networks. *Bioinformatics 20 (Suppl 1)*, pages i15–i22, 2004.

[5] C. Baral & et. al. From theory to practice: the utep robot in aaai 96 and 97 robot contests. In *Proceedings of the second international conference on automated agents (Agents 98)*, pages 32–38, 1998.

[6] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1976.

[7] D. Calvanese, G. de Giacomo, and M. Vardi. Reasoning about actions and planning in LTL action theories. In *Proc. KR*, pages 593–602, 2002.

[8] Nathalie Chabrier, Marc Chiaverini, Vincent Danos, F. Fages, and Vincent Schachter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.

[9] P. Doherty, G. Granlund, K. Kuchcinski, E. Sandewall, K. Nordberg, E. Skarman, and J. Wiklund. The WITAS unmanned aerial vehicle project. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*, pages 747–755, 2000.

[10] E. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B*, pages 997–1072, 1990.

[11] K Fukuda and T Takagi. Knowledge representation of signal transduction pathways. *Bioinformatics*, 17(9):829–37, 2001.

[12] Michael Gelfond and Vladimir Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17(2/3&4):301–321, 1993.

[13] L. Giordano, M. Martelli, and C. Schwind. Reasoning about actions in dynamic linear time temporal logic. *Journal of the IGPL*, 9(2):289–303, 2001.

[14] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artif. Intell.*, 153(1-2):49–104, 2004.

[15] Bengt Glimelius. Holland-Frei Cancer Medicine 6, vols 1 & 2. *JAMA*, 293(3):372–373, 2005.

[16] Henrik Grosskreutz and Gerhard Lakemeyer. cc-golog: Towards more realistic logic-based robot controllers. In *AAAI/IAAI*, pages 476–482, 2000.

[17] Christoph S. Herrmann and Michael Thielscher. Reasoning about continuous processes. In *Proceedings of AAAI*, pages 639–644, 1996.

[18] Antonis Kakas and Rob Miller. Reasoning about actions, narratives and ramifications. *ETAI*, 1:39–72, 1998.

[19] S. Khan, K. Decker, W. Gillis, and C. Schmidt. A multi-agent system-driven AI planning approach to biological pathway discovery. In *Proc. ICAPS*, 2003.

[20] John F. Kolen and Feng Zhao. A computational analysis of the reachability problem for a class of hybrid dynamical systems. In *Hybrid Systems*, pages 215–227, 1996.

[21] A. Lapouchnian and Y. Lesperance. Interfacing IndiGolog and OAA - a toolkit for advanced multiagent applications. *Applied Artificial Intelligence*, 16(9-10):813–829, 2002.

[22] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence 4*, pages 463–502, 1969.

[23] M. Nogueira. *Building knowledge systems in A-Prolog*. PhD thesis. University of Texas at El Paso, May 2003.

[24] M. Peleg, I. Yeh, and R. B. Altman. Modelling biological processes using workflow and petri net models. *Bioinformatics*, 18(6):825–837, 2002.

[25] J. A. Pinto. Temporal reasoning in the situation calculus. *PhD thesis, University of Toronto, Department of Computer Science*, 1994.

[26] R.A.Kowalski and M.J.Sergot. A logic-based calculus of events. *New Generation Computing*, 4, 1986.

[27] V. N. Reddy, M. N. Liebman, and M. L. Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, 26:9–24, 1996.

[28] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using π-calculus process algebra. In *Proc. PSB*, pages 459–470, 2001.

[29] R. Reiter. *Knowledge in action : logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.

[30] Raymond Reiter. Natural actions, concurrency and continuous time in the situation calculus. In *Proc. KR*, pages 2–13, 1996.

[31] Erik Sandewall. Combining logic and differential equations for describing real-world systems. In *Proceedings of KR*, pages 412–420, 1989.

[32] Reinhard Schwarz and Friedemann Mattern. Detecting causal relationships in distributed computations: In search of the holy grail. *Distributed Computing*, 7(3):149–174, 1994.

[33] M. P. Shanahan. A logical account of the common sense informatic situation for a mobile robot. *Electronic Transactions on Artificial Intelligence*, 2:69–104, 1998.

[34] Murray Shanahan. Representing continuous change in the event calculus. In *European Conference on Artificial Intelligence*, pages 598–603, 1990.

[35] C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In *Proc. PSB*, pages 568–580, 2004.

[36] Patrick B. Tan and Stuart K. Kim. Signaling specificity: the RTK/RAS/MAP kinase pathway in metazoans. *Trends in Genetics*, 15(4):145–149, 1999.

[37] M. Thielscher. Representing the knowledge of a robot. In *Proc. KR*, pages 109–120, 2000.

[38] N. Tran and C. Baral. Reasoning about triggered actions in AnsProlog and its application to molecular interactions in cells. In *Proc. KR*, pages 554–563, 2004.

[39] Nam Tran. Reasoning and Hypothesizing about Signaling Networks. *PhD thesis, Arizona State University*, 2006.

Technical Background

Complexity Theory

The classes Σ_k^P, Π_k^P and Δ_k^P of the Polynomial-time Hierarchy (PH) [1] are defined as follows:

$$\Sigma_0^P = \Pi_0^P = \Delta_0^P$$

and for all $k \geq 0$:

$$\Sigma_{k+1}^P = \text{NP}^{\Sigma_k^P}$$
$$\Pi_{k+1}^P = \text{coNP}^{\Pi_k^P}$$
$$\Delta_{k+1}^P = P^{\Delta_k^P}$$

Here $\text{P}^{\mathcal{C}}$ and $\text{NP}^{\mathcal{C}}$ denote the classes of problems that are solvable in polynomial time on a deterministic (resp. nondeterministic) Turing machine with an oracle for any problem $\mathcal{P}$ in the complexity class $\mathcal{C}$.

The complexity classes Σ_k^P and Π_k^P can be characterized as follows [6]. Let Γ be an alphabet. A relation R of dimension k over Γ^* is a set of k-tuples $\langle z_1, \ldots, z_k \rangle$ such that $z_i \in \Gamma*$ for $1 \leq i \leq k$. We say that R is recognizable in polynomial time if there is a polynomial time DTM that recognizes the language consisting of exactly the k-tuples of R.

Theorem 6.1. *[6] Let $L \subseteq \Gamma$ be a language, with $|\Gamma| \geq 2$. For any $k \leq 1$, $L \in \Sigma_k^P$ iff there exists polynomials $p_1, \ldots, p_k$ and a polynomial time recognizable relation R of dimension $k+1$ over Γ^* such that for all x in Γ^*:*

$$\begin{aligned} x \in L \iff & (\exists\, y_1 \in \Gamma^* \textit{ with } |y_1| \leq p_1(|x|)) \\ & (\forall\, y_2 \in \Gamma^* \textit{ with } |y_2| \leq p_2(|x|)) \\ & \vdots \\ & (Q_k\, y_k \in \Gamma^* \textit{ with } |y_k| \leq p_k(|x|)) \\ & \quad [\, \langle x, y_1, ..., y_k \rangle \in R \,] \end{aligned}$$

where the quantifiers alternate; and the quantifier Q_k on y_k is $\exists$ if k is odd and $\forall$ if k is even.

The complexity classes Π_k^P can be characterized analogously by interchanging $\exists$ and $\forall$ in Theorem 6.1.

Theorem 6.2. *[6] Let $L \subseteq \Gamma$ be a language, with $|\Gamma| \geq 2$. For any $k \leq 1$, $L \in \Pi_k^P$ iff there exists polynomials $p_1, \ldots, p_k$ and a polynomial time recognizable relation R of dimension $k+1$ over Γ^* such that for all x in Γ^*:*

$$\begin{aligned} x \in L \iff & (\forall\, y_1 \in \Gamma^* \textit{ with } |y_1| \leq p_1(|x|)) \\ & (\exists\, y_2 \in \Gamma^* \textit{ with } |y_2| \leq p_2(|x|)) \\ & \vdots \\ & (Q_k\, y_k \in \Gamma^* \textit{ with } |y_k| \leq p_k(|x|)) \\ & [\, \langle x, y_1, ..., y_k \rangle \in R \,] \end{aligned}$$

where the quantifiers alternate; and the quantifier Q_k on y_k is $\forall$ if k is odd and $\exists$ if k is even.

The lower-bound complexities for Σ_k^P and Π_k^P completeness can be proved by transformation from quantified Boolean formulas (QBFs). A QBF is an expression of the form

$$\Phi = Q_1 X_1 Q_2 X_2 \ldots Q_n X_k : E$$

where E is a Boolean expression whose atoms are from pair-wise disjoint nonempty sets of variables $X_1, X_2, \ldots, X_k$, $k \geq 1$, and Q_i's ($1 \leq i \leq k$) are alternating quantifiers from $\{\forall, \exists\}$. Let $\mathscr{P}_{k,\exists}$ be the decision problem of deciding if Φ is valid given $k \leq 1$ fixed and $Q_1 = \exists$. Let $\mathscr{P}_{k,\forall}$ be the decision problem of deciding if Φ is valid given $k \leq 1$ fixed and $Q_1 = \forall$. The following result is proven in [1] and [6].

Theorem 6.3. *For all $k \leq 1$, $\mathscr{P}_{k,\exists}$ is Σ_k^P-complete and $\mathscr{P}_{k,\forall}$ is Π_k^P-complete.*

Linear Temporal Logic

We shall use a propositional version of linear temporal logic (LTL) [10]. The language is built upon the standard propositional logic, using future temporal modalities $\bigcirc$, $\Box$, $\Diamond$ and $\cup$. The intuitive interpretations of the temporal modalities are:

- The modality $\bigcirc$ is read as "next". The expression $\bigcirc\ f$ means that f holds at the next time point.
- The modality $\Box$ is read as "always". The expression $\Box\ f$ means that f holds now and at all future time points.
- The modality $\Diamond$ is read as "eventually". The expression $\Diamond\ f$ means that f holds now or at some future time point.
- The modality $\cup$ is read as "until". The expression $f \cup g$ means that f holds from now until g becomes true.

The syntax of LTL formulas is constructively defined as follows.

- A propositional formula f is a LTL formula.
- If f is a LTL formula, then $\bigcirc\ f$, $\Box\ f$ and $\Diamond\ f$ are LTL formulas.
- If f and g are LTL formulas, then $f \cup g$ is an LTL formula.

A model of a set of LTL formulas is a sequence of the form $\mathcal{S} = \langle M_0, M_1, \ldots, M_i, \ldots \rangle$, where M_is are models of the propositional sub-language. First, we define the interpretations of LTL formulas with respect to the pairs $(\mathcal{S}, M_i)$. In the following, let p be a propositional formula and f, g be LTL formulas.

- $(\mathcal{S}, M_i) \models p$ iff p is true in the propositional model M_i .
- $(\mathcal{S}, M_i) \models \neg f$ iff $(\mathcal{S}, M_i) \not\models f$.
- $(\mathcal{S}, M_i) \models f \wedge g$ iff $(\mathcal{S}, M_i) \models f$ and $(\mathcal{S}, M_i) \models g$.
- $(\mathcal{S}, M_i) \models f \vee g$ iff $(\mathcal{S}, M_i) \models f$ or $(\mathcal{S}, M_i) \models g$.
- $(\mathcal{S}, M_i) \models \bigcirc f$ iff $(\mathcal{S}, M_{i+1}) \models f$.
- $(\mathcal{S}, M_i) \models \Box f$ iff $(\mathcal{S}, M_j) \models f$ for all $j \geq i$.
- $(\mathcal{S}, M_i) \models \Diamond f$ iff $(\mathcal{S}, M_j) \models f$ for some $j \geq i$.
- $(\mathcal{S}, M_i) \models f \cup g$ iff there exists $j \geq i$ such that $(\mathcal{S}, M_k) \models f$ for all $i \leq k < j$ and $(\mathcal{S}, M_j) \models g$.

Finally, the model $\mathcal{S}$ entails an LTL formula f, written as $\mathcal{S} \models f$, iff $(\mathcal{S}, M_0) \models f$.