

Reasoning about Triggered Actions in AnsProlog and Its Application to Molecular Interactions in Cells

Nam Tran & Chitta Baral

Department of Computer Science and Engineering
Arizona State University
Tempe, Arizona 85287
{namtran,chitta}@asu.edu

Abstract

Reasoning about molecular interactions and signaling pathways is important from various perspectives such as predicting side effects of drugs, explaining unusual cellular behavior and drug and therapy design. Because of the vast size of these interactions a typical biologist can only focus on a very small part of the network. Thus there is a great need to develop knowledge representation and reasoning formalisms and their implementations for modelling and reasoning about molecular interactions in cells of organisms. An important component of these interactions is the *action* of one molecule interacting with or binding to another, or one molecule separating into multiple other molecules. Thus, action theories and action languages are good candidates to model these interactions. One major lacking of most existing action languages is the notion of *triggered actions*, which is a common phenomena in the cellular domain. In this paper, we introduce a language for representing and reasoning about triggered actions, and show how to model reasoning about side effects, explaining observations, and designing drugs in our language through implementations using AnsProlog.

Keywords: action languages, reasoning about actions

Introduction and motivation

In the past action languages have been developed and applied to domains such as robots (Grosskreutz & Lake-meyer 2000; Reiter 2001; Thielscher 2000; Shanahan 1998; Baral *et al.* 1998), agents (Lapouchnian & Lesperance 2002), helicopters (Doherty *et al.* 2000), and space shuttles (Balduccini *et al.* 2001; 2002; Nogueira 2003). In this paper our motivation is the domain of molecular interactions in cells of organisms (AFCS ; STKE). An example of the kind of behavior we would like to model, is the happenings that follow when a particular ligand (often a protein molecule) comes in contact with a receptor molecule in the membrane of a cell. The immediate effect is that the ligand binds with the receptor; but such a binding in the presence of certain other molecules inside the cell may trigger an action (or another binding) which in turn may trigger other actions. Sometimes the presence of particular molecules inhibits certain actions which would have been otherwise triggered.

Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

While modelling these behaviors of the cell, we are interested in formalizing and implementing several reasoning abilities which 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.

With these somewhat lofty goals in mind, in this paper we take a first step towards developing a formalism that will allow us to represent various interactions (including gene-gene, gene-protein, protein-protein, and other molecular interactions), and perform the three kinds of reasoning mentioned above.

The above mentioned three kinds of reasoning are not new to action theory research. The contribution of this paper in that regard is that most of the action theories used in the past do not account for triggers and inhibitions of triggered and non-triggered actions. Researchers have tried to use other formalisms such as Petri nets (Reddy, Liebman, & Mavrovouniotis 1996; Peleg, Yeh, & Altman 2002), and π -calculus (A. Regev & Shapiro 2001) for biological modelling. But these approaches are less focussed on elaboration tolerant representation and reasoning and more focussed on modelling and simulation, and hence are not as adequate for the kind of representation and reasoning tasks we consider here.

Triggered interactions form the majority of interactions in a cellular regulatory network. However, there has been little research on action theories that can represent such interactions, except in (Baral & Tran 2003) and in some active database related papers. Our goal in this paper is to define

an action language \mathcal{A}_T^0 , which is inspired by \mathcal{A} (Gelfond & Lifschitz 1998). Our language has minimal new features but is able to handle some basic reasonings about triggered actions. We implement the language in AnsProlog (Baral 2003), and have built a prototype (Baral *et al.* 2004) with JAVA GUI for molecular biologists (and others) to test the system and provide feedback.

The rest of the paper is organized as follows. First we introduce the syntax and semantics of \mathcal{A}_T^0 . Then we formulate several basic reasoning problems in \mathcal{A}_T^0 . It is followed by the translation of \mathcal{A}_T^0 and of the reasoning problems into AnsProlog. We illustrate the usefulness of our formalism with a molecular interactions domain. Finally, we conclude with a comparison with the language in (Baral & Tran 2003) and with glimpses of our future plans.

Language \mathcal{A}_T^0

The alphabet of the language \mathcal{A}_T^0 consists of three nonempty disjoint sets of symbols: a set \mathbf{A}_{trig} of triggered actions, a set \mathbf{A}_{exo} of non-triggered actions and a set \mathbf{F} of fluents. A fluent expresses some property of the world. A *fluent literal* is a fluent or a fluent preceded by \neg . A state is a set of fluents that satisfies some conditions (to be mentioned later when discussing semantics).

The action language \mathcal{A}_T^0 is composed of three sub-languages: a domain description language, an observation language and a query language. We now present these components in details.

The domain description language

Syntax A domain description in \mathcal{A}_T^0 consists of propositions of the following form:

$$a \text{ causes } f \text{ if } f_1, \dots, f_n \quad (1)$$

$$g_1, \dots, g_m \text{ n_triggers } b \quad (2)$$

$$h_1, \dots, h_l \text{ inhibits } c \quad (3)$$

where f_i, g_j, h_k and f and g are fluent literals; a is an action and b, c are *triggered* actions. (1) represents a *dynamic causal law*, which states that f is guaranteed to be true after the execution of a if f_1, \dots, f_n are true when a occurs. (2) is a *triggering rule*, which states that *normally* action b will happen (unless inhibited) whenever g_1, \dots, g_m are all true. (3) is an *inhibition rule*, which states that a triggered action c can not occur whenever h_1, \dots, h_l are all true.

Given a domain description \mathcal{D} , we write $\mathcal{D}(\mathbf{A}_{trig}, \mathbf{A}_{exo}, \mathbf{F})$ to denote that \mathbf{A}_{trig} , \mathbf{A}_{exo} and \mathbf{F} are respectively the set of triggered actions, the set of non-triggered actions and the set of fluents in \mathcal{D} .

Intuitively, a domain description determines the possible trajectories along which the world evolves. A trajectory is a sequence of states and action occurrences. Propositions of the form (1) define how the world changes from one state to another state due to an action occurrence. Propositions of the forms (2)-(3) define what (triggered) actions will be triggered or inhibited in a state. We formalize the intuition in the following.

Semantics - trajectory The propositions of the form (1) define a transition function (Φ) from pairs of a set of actions and a state to the set of states: given a set A of actions and a state s , the transition function Φ defines the state $\Phi(A, s)$ that may be reached after executing the set A of actions in state s .

Let \mathcal{D} be a domain description in \mathcal{A}_T^0 . 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 . For a set X of fluent literals, let us write $X^+ = \{f | f \in X\}$ and $X^- = \{f | \neg f \in X\}$. Then X is consistent iff $X^+ \cap X^- = \emptyset$.

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 in \mathcal{D} is an interpretation of the fluents in \mathcal{F} .

The direct effect of an action a in a state s is the set $E(a, s) = \{f \mid a \text{ causes } f \text{ if } f_1, \dots, f_n \in \mathcal{D} \text{ and } s \models f_1 \wedge \dots \wedge f_n\}$.

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

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

For a domain description \mathcal{D} , the state $\Phi(A, s)$ that may be reached by executing a in s is defined as follows.

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

$$\Phi(A, s) = X \cup \{\neg f \mid f \in \mathbf{F} \setminus X\},$$

where X is the set $(s^+ \cup E^+(A, s)) \setminus E^-(A, s)$;

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

The intuition behind the above formulation is as follows. The direct effect of a set A of actions in a state s is determined by the dynamic causal laws and is given by $E(A, s)$. If $E(A, s)$ is inconsistent, then the resulting state $\Phi(A, s)$ is undefined. Otherwise, $\Phi(A, s)$ is computed from s by removing the fluent literals changed by A then adding the direct effect $E(A, s)$.

Example 1. Let us consider a simple domain description \mathcal{D} as follow. Let a, b_1 , and b_2 are actions; f, g, u and v are fluents. The domain consists of the propositions:

$$\begin{array}{ll} a \text{ causes } f & f, u \text{ n_triggers } b_1 \\ b_1 \text{ causes } \{g, \neg f\} & f, v \text{ n_triggers } b_2 \\ b_2 \text{ causes } \{\neg g, \neg f\} & g, \neg v \text{ inhibits } b_1 \end{array}$$

Let s_0, s_1, s_2 and s_3 be states:

$$\begin{array}{ll} s_0 = \{\neg f, \neg g, u, \neg v\} & s_2 = \{\neg f, g, u, \neg v\} \\ s_1 = \{f, \neg g, u, \neg v\} & s_3 = \{f, g, u, \neg v\} \end{array}$$

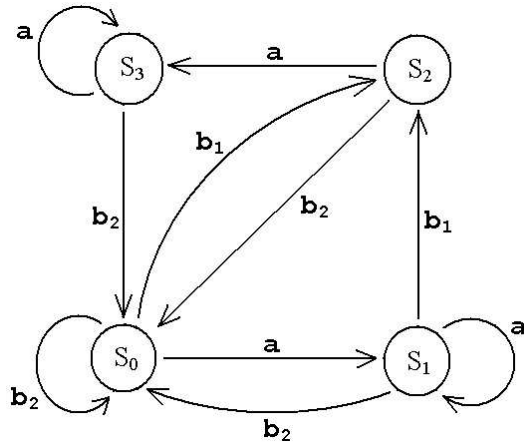


Figure 1: A partial transition function

Then by definition, we have that (Figure 1)

$$\begin{aligned}\Phi(\{a\}, s_0) &= s_1 \\ \Phi(\{b_1\}, s_0) &= s_2 \\ \Phi(\{b_2\}, s_0) &= s_0\end{aligned}$$

Besides, we also note that $\Phi(\{a, b_1\}, s_0)$, $\Phi(\{a, b_2\}, s_0)$, and $\Phi(\{b_1, b_2\}, s_0)$ are all undefined.

Definition 1. Let \mathcal{D} be a domain description. Let s be some state in \mathcal{D} . A triggering rule $r = f_1, \dots, f_m$ **n-triggers** a of \mathcal{D} is said to be *active* in s , if all the fluents f_1, \dots, f_m hold in s ; otherwise, r is said to be *passive* in s .

A inhibition rule $r' = f'_1, \dots, f'_k$ **inhibits** a' of \mathcal{D} is said to be *active* in s , if all the fluents f'_1, \dots, f'_k hold in s ; otherwise r' is said to be *passive* in s . \square .

Definition 2 (Trajectory). Let \mathcal{D} be a domain description. Let $s_0, s_1, \dots, s_n, \dots$ be states and A_1, \dots, A_n, \dots be sets of actions in \mathcal{D} . The sequence $\tau = s_0, A_1, s_1, A_2, \dots, A_n, s_n, \dots$ is called a trajectory in \mathcal{D} if the following conditions are all satisfied.

- State s_n is reached by executing A_n in state s_{n-1} ; that is, $s_n = \Phi(A_n, s_{n-1})$.
- If a triggering rule f_1, \dots, f_m **n-triggers** a is active in a state s_i , and all the inhibition rules with a as the inhibited action are passive in s_i , then $a \in A_{i+1}$.
- If a triggered action $a \in A_{i+1}$, then there must exist an active triggering rule in s_i with a as the triggered action; and all the inhibition rules with a as the inhibited action must be passive in s_i .
- If an inhibition rule f'_1, \dots, f'_k **inhibits** a' is active in a state s_i , then $a' \notin A_{i+1}$.
- If $A_j = \emptyset$, then $A_l = \emptyset$, for all $l \geq j$. \square

Example 2. Let us consider the domain description in Example 1. There are trajectories such as:

$$\begin{aligned}\tau_1 &= s_0, \emptyset, s_0, \emptyset, \dots \\ \tau_2 &= s_0, \{a\}, s_1, \{b_1\}, s_2, \emptyset, s_2, \emptyset, \dots \\ \tau_3 &= s_1, \{b_1\}, s_2, \{b_2\}, s_0, \{a\}, s_1, \{b_1\}, s_2, \{b_2\}, \dots\end{aligned}$$

The following sequences are non-trajectories:

$$\begin{aligned}\tau'_1 &= s_0, \{a, b_1\}, s_0, \emptyset, \dots \\ \tau'_2 &= s_0, \{a\}, s_1, \{b_2\}, s_0, \emptyset, s_0, \emptyset, \dots \\ \tau'_3 &= s_1, \{b_1\}, s_2, \{a\}, s_3, \{b_1\}, \dots \\ \tau'_4 &= s_1, \{b_1\}, s_2, \emptyset, s_2, \{a\}, \dots\end{aligned}$$

\square

A trajectory in a domain description \mathcal{D} is said to be *finite*, if there exists a time point at which no action occurs. The *length* of a finite trajectory is defined to be the first time point at which no action occurs.

Definition 3 (Trigger bounded domain). A domain description \mathcal{D} is called *trigger bounded*, if all trajectories in \mathcal{D} with only triggered actions are finite. \square

Let $\mathcal{D}\langle \mathbf{A}_{trig}, \mathbf{A}_{exo}, \mathbf{F} \rangle$ be a trigger bounded domain. By definition, there exists an upper bound on the lengths of trajectories in \mathcal{D} which contain only triggered actions. Denote the upper bound by $tbound(\mathcal{D})$. Then this upper bound is not more than the number of states in \mathcal{D} ; that is, $tbound(\mathcal{D}) \leq 2^{|\mathbf{F}|}$.

The observation language

Syntax Let f, f_1, \dots, f_n be fluent literals and A_1, \dots, A_n be sets of actions. Let t, t_1, \dots, t_n be time points, which are nonnegative integers. Propositions in the observation language take the following form:

- observation about a fluent at some time point:

$$f \text{ at } t. \quad (4)$$

- observation about a fluent at the initial time point:

$$f \text{ at } 0, \text{ or equivalently, initially } f. \quad (5)$$

- observation about occurrences of actions:

$$A_1 \text{ occurs_at } t_1, \dots, A_n \text{ occurs_at } t_n. \quad (6)$$

The intuitive meaning of the proposition (4) is that fluent literal f is true in the state at time point t . Similarly, (5) means that fluent literal f is initially true (in the initial state). The observation (6) says that the actions in A_1 occur at time point t_1 , ..., the actions in A_n occur at time point t_n .

Note that we do not require that t_1, \dots, t_n must be different in observation of the form (6). Thus the following two sets of observations are equivalent:

$$\begin{aligned}O_1 &= \{ A \text{ occurs_at } t \} \\ O_2 &= \{ a \text{ occurs_at } t \mid a \in A \}\end{aligned}$$

The set of observations \mathcal{O} is called *initial state complete*, if for every fluent f , \mathcal{O} contains either **initially** f or **initially** $\neg f$, but not both.

An initial state s_0 is *consistent* with a set of observations \mathcal{O} if for every fluent f :

- if **initially** $f \in \mathcal{O}$ then $f \in s_0$;
- if **initially** $\neg f \in \mathcal{O}$ then $\neg f \in s_0$.

We have the following simple result about the relation between a set of observations and a state.

Proposition 1. Let \mathcal{D} be a domain description and \mathcal{O} be an initial state complete set of observations in \mathcal{D} . Then there exists a unique initial state that is consistent with \mathcal{O} . \square

We use the domain description language and the observation language to represent our knowledge about the world, which are called action theories.

Definition 4 (Action theory). An action theory in \mathcal{A}_T^0 is a pair $(\mathcal{D}, \mathcal{O})$ where \mathcal{D} is a domain description and \mathcal{O} is a set of observations in \mathcal{A}_T^0 . An action theory $(\mathcal{D}, \mathcal{O})$ is called initial state complete if \mathcal{O} is initial state complete. \square

We follow (Baral & Tran 2003; Baral & Gelfond 2000) in defining models of an action theory $(\mathcal{D}, \mathcal{O})$.

Semantics - trajectory model Intuitively, a domain description \mathcal{D} specifies how the world can evolve. By incorporating observations, we can have a better understanding of the world; that is, we have more constraints on the evolution of the world.

Definition 5 (Trajectory interpretation). Let \mathcal{D} be a domain description and \mathcal{O} be a set of observations in \mathcal{D} . A trajectory $\tau = s_0, A_1, s_1, A_2 \dots A_n, s_n \dots$ is an interpretation of $(\mathcal{D}, \mathcal{O})$ if the following conditions are all satisfied:

- if f at $t \in \mathcal{O}$, then $f \in s_t$; and
- if B_1 occurs at t_1, B_2 occurs at t_2, \dots, B_j occurs at t_j belongs to \mathcal{O} , then $B_i \subseteq A_{t_i}$, for all $1 \leq i \leq j$. \square

We define an ordering on trajectories and select the “best” interpretations to be models. Intuitively, a model is such an interpretation that at every time point, the set of action occurrences is as minimal as possible. Thus if an action a is in \mathcal{A}_{exo} , there must be an observation about its occurrence; if a is in \mathcal{A}_{trig} then a must be triggered.

Definition 6 (Ordering of trajectories). Let τ and τ' be trajectory such that $\tau = s_0, A_1, s_1, \dots, A_n, s_n, \dots$ and $\tau' = s'_0, A'_1, s'_1, \dots, A'_m, s'_m, \dots$ be trajectories, such that $s_0 = s'_0$. We say that $\tau \leq \tau'$ if there exists a sequence $0 \leq i_1 < i_2 < \dots < i_n < \dots$ such that for every $1 \leq k$, $A_k \subseteq A'_{i_k}$. Moreover, $\tau = \tau'$ iff $A_k = A'_{i_k}$, for all $1 \leq k$. \square

Definition 7 (Trajectory model). Let \mathcal{D} be a domain description and \mathcal{O} be a set of observations in \mathcal{A}_T^0 . A trajectory $\tau = s_0, A_1, s_1, A_2, \dots, A_n, s_n \dots$ is said to be a trajectory model of $(\mathcal{D}, \mathcal{O})$ if

- τ is a trajectory interpretation of $(\mathcal{D}, \mathcal{O})$, and
- there does not exist a trajectory interpretation τ' of $(\mathcal{D}, \mathcal{O})$ such that $\tau' < \tau$. \square

Example 3. Consider the domain description from Example 1. Let \mathcal{O} be the following set of observations

$$\mathcal{O} = \{\text{initially } \neg g\} \cup \{b_1 \text{ at } 3t \mid t = 0, 1, 2, \dots\}$$

Then the following trajectories are interpretations of the theory $(\mathcal{D}, \mathcal{O})$:

$$\begin{aligned} & s_1, \{b_1\}, s_2, \{b_2\}, s_0, \{a\}, s_1, \{b_1\}, s_2, \{b_2\}, \dots \\ & s_0, \{b_1\}, s_2, \{a\}, s_3, \{b_2\}, s_0, \{b_1\}, s_2, \{a\}, \dots \\ & s_0, \{b_1\}, s_2, \{b_2\}, s_0, \{b_2\}, s_0, \{b_1\}, s_2, \{b_2\}, \dots \end{aligned}$$

These interpretations are also models of the theory $(\mathcal{D}, \mathcal{O})$. Let $\mathcal{O}' = \{\text{initially } \neg g, b_1 \text{ at } 0, b_1 \text{ at } 1\}$. Because $\mathcal{O}' \subset \mathcal{O}$, every interpretation of $(\mathcal{D}, \mathcal{O})$ is also an interpretation of $(\mathcal{D}, \mathcal{O}')$. However, because of the minimality property of models, interpretations of $(\mathcal{D}, \mathcal{O})$ cannot be models of $(\mathcal{D}, \mathcal{O}')$. Models of $(\mathcal{D}, \mathcal{O}')$ include:

$$\begin{aligned} & s_1, \{b_1\}, s_2, \{b_2\}, s_0, \{a\}, s_1, \{b_1\}, s_2, \emptyset, s_2, \emptyset, \dots \\ & s_0, \{b_1\}, s_2, \{a\}, s_3, \{b_2\}, s_0, \{b_1\}, s_2, \emptyset, s_2, \emptyset, \dots \\ & s_0, \{b_1\}, s_2, \{b_2\}, s_0, \{b_2\}, s_0, \{b_1\}, s_2, \emptyset, s_2, \emptyset, \dots \end{aligned} \quad \square$$

An action theory $(\mathcal{D}, \mathcal{O})$ is said to be *consistent* if it has a trajectory model. In this paper, we are concerned with a specific class of action theories, which have the following property.

Proposition 2. Let $(\mathcal{D}, \mathcal{O})$ be an action theory where \mathcal{O} is initial state complete. If $(\mathcal{D}, \mathcal{O})$ is consistent then it has a unique trajectory model. \square

Semantics - entailment of observations Let $(\mathcal{D}, \mathcal{O})$ be a consistent action theory. Let f be a fluent and t be a time point. $(\mathcal{D}, \mathcal{O})$ entails the observation $\{f \text{ at } t\}$ if for all trajectory model τ of $(\mathcal{D}, \mathcal{O})$, $\tau = s_0, A_1, s_1, A_2 \dots A_n, s_n \dots$, we have that $f \in s_t$. We then write that $(\mathcal{D}, \mathcal{O}) \models f \text{ at } t$.

Let A_1, \dots, A_n be sets of actions and t_1, \dots, t_n be time points. The theory $(\mathcal{D}, \mathcal{O})$ entails the observation $\{A_1 \text{ occurs at } t_1, \dots, A_n \text{ occurs at } t_n\}$, if for all trajectory model τ of $(\mathcal{D}, \mathcal{O})$, where $\tau = s_0, A'_1, s_1, A'_2 \dots A'_k, s_k \dots$, we have that $A_i \subseteq A'_{t_i}$ for $1 \leq i \leq n$. We then write that

$$(\mathcal{D}, \mathcal{O}) \models A_1 \text{ occurs at } t_1, \dots, A_n \text{ occurs at } t_n.$$

Given a set \mathcal{O}' of observations, we say $(\mathcal{D}, \mathcal{O})$ entails \mathcal{O}' , written as $(\mathcal{D}, \mathcal{O}) \models \mathcal{O}'$, if $(\mathcal{D}, \mathcal{O}) \models \omega$, for all $\omega \in \mathcal{O}'$.

The query language

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

$$f \text{ after } A_1 \text{ at } t_1, \dots, A_n \text{ at } t_n;$$

where f is a fluent, A_1, \dots, A_n are sets of actions and t_1, \dots, t_n are time points.

Let $(\mathcal{D}, \mathcal{O})$ be a consistent action theory. Let Q be a query $f \text{ after } A_1 \text{ at } t_1, \dots, A_n \text{ at } t_n$. Moreover, denote \mathcal{O}' be the set $\mathcal{O} \cup \{A_1 \text{ occurs at } t_1, \dots, A_n \text{ occurs at } t_n\}$. The pair $(\mathcal{D}, \mathcal{O})$ entails Q , written as $(\mathcal{D}, \mathcal{O}) \models Q$, if $(\mathcal{D}, \mathcal{O}')$ is consistent and for all trajectory model τ of $(\mathcal{D}, \mathcal{O}')$, where $\tau = s_0, A'_1, s_1, A'_2 \dots A'_m, s_m \dots$, there exists $N \geq t_n$ such that f is true in all the states $s_k, k > N$.

Example 4. Let \mathcal{D} be the domain description in Example 1. Let \mathcal{O} be a set of observations:

$$\mathcal{O} = \{\text{initially } \neg f, \text{initially } \neg g, \text{initially } u, \text{initially } \neg v\}.$$

Then trajectory $s_0, \emptyset, s_0, \emptyset, \dots$ is the unique trajectory model of the theory $(\mathcal{D}, \mathcal{O})$.

Let $Q = f \text{ after } \{a\} \text{ at } 0$. $\mathcal{O}' = \mathcal{O} \cup \{\{a\} \text{ at } 0\}$. Then trajectory $s_0, \{a\}, s_1, \{b_1\}, s_2, \emptyset, s_2, \emptyset, \dots$ is the unique trajectory model of $(\mathcal{D}, \mathcal{O}')$. We see that $f \in s_2$, hence the action theory $(\mathcal{D}, \mathcal{O})$ entails the query Q . \square

We now formulate several basic reasoning problems using \mathcal{A}_T^0 , including prediction, explanation and planning.

Reasoning in \mathcal{A}_T^0 framework

Prediction

In this kind of reasoning, we want to know what would be true in a state resulting from a course of actions. Let \mathcal{D} be a domain description and \mathcal{O} be an initial state complete set of observations, such that $(\mathcal{D}, \mathcal{O})$ is consistent. Let $Q = f$ **after** A_1 **at** t_1, \dots, A_n **at** t_n . We say that $(\mathcal{D}, \mathcal{O})$ *predicts* f after A_1 occurs at t_1, \dots, A_n occurs at t_n ; or $(\mathcal{D}, \mathcal{O})$ predicts Q for short, if $(\mathcal{D}, \mathcal{O}) \models Q$.

We denote by $pred(\mathcal{D}, \mathcal{O}, Q)$ the problem of checking if $(\mathcal{D}, \mathcal{O})$ predicts Q .

Explanation

Intuitively, given a set \mathcal{O} of observations about fluents and *triggered* actions, we want to find initial states and occurrences of *non-triggered* actions that can explain \mathcal{O} .

Definition 8. Let \mathcal{D} be a domain description and \mathcal{O}_{init} be a set of observations (not necessarily complete) about the initial state. Let \mathcal{O} be a set of observations about fluents and occurrences of *triggered* actions. Let \mathcal{O}_{exp} be a set of observations about the initial state and occurrences of *non-triggered* actions. Then \mathcal{O}_{exp} is an explanation¹ of \mathcal{O} with respect to $(\mathcal{D}, \mathcal{O}_{init})$ if $\mathcal{O}_{init} \cup \mathcal{O}_{exp}$ is initial state complete and $(\mathcal{D}, \mathcal{O}_{init} \cup \mathcal{O}_{exp}) \models \mathcal{O}$. \square

We also denote the problem of finding such an \mathcal{O}_{exp} by $exp(\mathcal{D}, \mathcal{O}_{init}, \mathcal{O})$ and \mathcal{O}_{exp} is then called a solution of $exp(\mathcal{D}, \mathcal{O}_{init}, \mathcal{O})$.

Example 5. Let \mathcal{D} be the domain description in Example 1. Let $\mathcal{O}_{init} = \{\text{initially } \neg f, \text{initially } \neg g, \text{initially } \neg v\}$. Let $\mathcal{O} = \{g \text{ at } 3\}$. Then $\{\text{initially } u\}$ is the explanation of \mathcal{O} w.r.t $(\mathcal{D}, \mathcal{O}_{init})$.

Indeed, there are only two possible explanations that contains only observations about the initial state, which are $\mathcal{O}_{exp}^- = \{\text{initially } \neg u\}$ and $\mathcal{O}_{exp}^+ = \{\text{initially } u\}$. It is easy to verify that $(\mathcal{D}, \mathcal{O}_{init} \cup \mathcal{O}_{exp}^-)$ is inconsistent, but $(\mathcal{D}, \mathcal{O}_{init} \cup \mathcal{O}_{exp}^+)$ has a unique trajectory model and $(\mathcal{D}, \mathcal{O}_{init} \cup \mathcal{O}_{exp}^+) \models g \text{ at } 3$. \square

Planning

Let $(\mathcal{D}, \mathcal{O})$ be a consistent action theory, where \mathcal{O} is initial state complete and contains only observations about the initial state. Let G be some fluent literal and $t_1 < t_2 < \dots < t_n$ be time points. Let A_1, A_2, \dots, A_n be sets of non-triggered actions. Then a sequence P such that $P = \langle A_1 \text{ at } t_1, A_2 \text{ at } t_2, \dots, A_n \text{ at } t_n \rangle$ is called a plan for the goal G if and only if:

$$(\mathcal{D}, \mathcal{O}) \models G \text{ after } A_1 \text{ at } t_1, A_2 \text{ at } t_2, \dots, A_n \text{ at } t_n$$

We denote the problem of finding such a P by $plan(\mathcal{D}, \mathcal{O}, G)$. Then P is called a solution of $plan(\mathcal{D}, \mathcal{O}, G)$.

¹In the literature (see for example (Baral, McIlraith, & Tran 2000; Balduccini & Gelfond 2003)) there are many different notions of explanation and diagnosis. Here we give one such notion. Other notions such as minimal explanation, preferred explanation can be exported to our formalism. We will discuss these in the full version of the paper.

\mathcal{A}_T^0 -based reasoning in AnsProlog framework

In this section, we present the general AnsProlog translation of the language \mathcal{A}_T^0 and of the reasoning problems. We will show the usefulness of the translation by a biological example in the next section.

When translating an action theory $(\mathcal{D}, \mathcal{O})$ to AnsProlog, we have to set an upper bound t_{max} of time steps. The number t_{max} depends on the reasoning problem to be solved and can be determined as we will see in the theorems coming later. Nevertheless, we can also make an educated guess of t_{max} , based on background knowledge about the represented system and its evolutions.

Translation of the domain description language

Recall that a domain description consists of propositions of the form (1), (2) and (3). The AnsProlog translation $\pi(\mathcal{D})$ of a domain \mathcal{D} includes *inertial rules*, *interpretation constraints* and the translations of all the propositions of \mathcal{D} .

For conciseness, we introduce some notations as follows. Given a fluent literal g and some fluent f , let us denote $\pi(g, t) \equiv holds(f, t)$ if $g \equiv f$; and let $\pi(g, t) \equiv holds(neg(f), t)$ if $g \equiv \neg f$. Given an action a , let $\pi(a, t) \equiv holds(occurs(a), t)$.

First, we describe the set of inertial rules. To capture the semantics of transition functions, we need the following inertial rules, for each fluent f and for all time points t in $[0, t_{max}]$:

$$\begin{aligned} \pi(f, t+1) &\leftarrow \pi(f, t), \text{ not } \pi(\neg f, t+1). \\ \pi(\neg f, t+1) &\leftarrow \pi(\neg f, t), \text{ not } \pi(f, t+1). \end{aligned}$$

Intuitively, the inertial rules mean that the truth values of fluent literals remain constant unless being affected by actions. For each fluent f , there are interpretation constraints of the following form, for all time point t in $[0, t_{max}]$.

$$\perp \leftarrow holds(f, t), holds(neg(f), t).$$

Intuitively, the interpretation constraints guarantee that both fluent literal f and $\neg f$ cannot hold at the same time.

Finally, we show how the propositions of \mathcal{D} are translated into AnsProlog:

- A proposition of the form (1) is translated into the following rules, $\forall t \in [0, t_{max}]$

$$\pi(f, t+1) \leftarrow \pi(a, t), \pi(f_1, t), \dots, \pi(f_n, t).$$

- A proposition of the form (2) is translated into the following rules, $\forall t \in [0, t_{max}]$

$$\begin{aligned} \pi(b, t) &\leftarrow \pi(g_1, t), \dots, \pi(g_m, t), \\ &\text{ not } holds(ab(occurs(b)), t). \end{aligned}$$

- A propositions of the form (3) is translated to the following rules, $\forall t \in [0, t_{max}]$

$$holds(ab(occurs(c)), t) \leftarrow \pi(h_1, t), \dots, \pi(h_l, t).$$

Note that logical atoms of the form $holds(ab(occurs(a), t))$ carry a special meaning. Intuitively, if $holds(ab(occurs(a), t))$ is true then a is prevented from occurring; that is, some abnormality happens to the occurrence of a .

Example 6. We illustrate how the domain \mathcal{D} in Example 1 can be translated into AnsProlog. Let us choose $t_{max} = 10$.

Because u, v, f and g are fluents, for all time points t in the interval $[0, 10]$, we add the constraints

$$\begin{aligned} \perp &\leftarrow holds(u, t), holds(neg(u), t). \\ \perp &\leftarrow holds(v, t), holds(neg(v), t). \\ &\dots\dots\dots \end{aligned}$$

The causal rules b_1 **causes** $\{g, \neg f\}$ is translated into the following rules, for all time points t in the interval $[0, 10]$.

$$\begin{aligned} holds(g, t+1) &\leftarrow holds(occurs(b_1), t). \\ holds(neg(f), t+1) &\leftarrow holds(occurs(b_1), t). \end{aligned}$$

Finally, for all $t \in [0, 10]$, the triggering f, u **n-triggers** b_1 is translated into

$$\begin{aligned} holds(occurs(b_1), t) &\leftarrow holds(f, t), holds(u, t), \\ &\quad not\ holds(ab(occurs(b_1)), t). \end{aligned}$$

and the inhibition $g, \neg v$ **n-triggers** b_1 is translated into

$$holds(ab(occurs(b_1), t) \leftarrow holds(g, t), holds(neg(v), t).$$

Translation of the observation language

Let \mathcal{O} be a set of observations in a domain description \mathcal{D} . The AnsProlog translation $\pi(\mathcal{O})$ of \mathcal{O} consists of the translations of all the observations of \mathcal{O} . The observations in \mathcal{O} are translated as follows.

- An observation of the form **initially** f is translated into the fact $holds(f, t) \leftarrow$.
- If $t > 0$, an observation of the form f **at** t is translated into the constraint $\perp \leftarrow not\ holds(f, t)$.
- If a is a triggered action and t is a time point, then the observation a **occurs.at** t is translated into the constraint $\perp \leftarrow not\ holds(occurs(a), t)$.
- If a is a non-triggered action, and t is a time point then the observation a **occurs.at** t is translated into the fact $holds(occurs(a), t) \leftarrow$.
- The translation of an observation of the form

$$A_1 \text{ occurs.at } t_1, \dots, A_n \text{ occurs.at } t_n$$

consists of all the translations of the observations a **occurs.at** t_i , where $a \in A_i, 1 \leq i \leq n$.

Correctness of the translation

Given a domain description \mathcal{D} , we write

$$\pi(\mathcal{D}, \mathcal{O}) = \pi(\mathcal{D}) \cup \pi(\mathcal{O}).$$

Corresponding to Proposition 2, we have the following result.

Proposition 3. Let $(\mathcal{D}, \mathcal{O})$ be a consistent action theory, where \mathcal{O} is initial state complete. Then the program $\pi(\mathcal{D}, \mathcal{O})$ has a unique answer set. \square

Given an answer set S of some translated AnsProlog program in a domain $\mathcal{D}(\mathbf{A}_{trig}, \mathbf{A}_{exo}, \mathbf{F})$, we write that

$$\begin{aligned} \Omega_{exo}(S) &= \{ a \text{ occurs.at } t \mid a \in \mathbf{A}_{exo}, holds(a, t) \in S \}; \\ \Omega_{init}(S) &= \{ \text{initially } f \mid f \in \mathbf{F}, holds(f, 0) \in S \} \cup \\ &\quad \{ \text{initially } \neg f \mid f \in \mathbf{F}, holds(neg(f), 0) \in S \}. \end{aligned}$$

We now present results showing the correctness of the AnsProlog translation, with respect to doing prediction, explanation and planning.

Prediction in AnsProlog Let \mathcal{D} be a trigger bounded domain description. Let \mathcal{O} be an initial state complete set of observations. Let $Q = f$ **after** A_1 **at** t_1, \dots, A_n **at** t_n where $t_1 < t_2 \dots < t_n$. Because \mathcal{D} is trigger bounded, it is sufficient to model evolutions of the world up to time $t_n + tbound(\mathcal{D})$. Let $t_{max} = t_n + tbound(\mathcal{D})$. In order to verify the entailment $(\mathcal{D}, \mathcal{O}) \models Q$, we compute the unique answer set S of the AnsProlog program

$$\pi(\mathcal{D}, \mathcal{O}) \cup \pi(\{ A_1 \text{ occurs.at } t_1, \dots, A_n \text{ occurs.at } t_n \}).$$

The prediction is true if the answer set exists, and there exists $N \geq t_n$ such that $hold(f, k) \in S$ for all $k \in (N, t_{max}]$.

Proposition 4. Let $pred(\mathcal{D}, \mathcal{O}, Q)$ be a prediction problem, where the domain \mathcal{D} is trigger bounded and Q is the query f **after** A_1 **at** t_1, \dots, A_n **at** t_n , where $t_1 < t_2 \dots < t_n$. Let $t_{max} = t_n + tbound(\mathcal{D})$; and let π_{pred} be the AnsProlog translation

$$\pi(\mathcal{D}, \mathcal{O}) \cup \pi(\{ A_1 \text{ occurs.at } t_1, \dots, A_n \text{ occurs.at } t_n \}).$$

Then the theory $(\mathcal{D}, \mathcal{O})$ predicts Q if and only if the program π_{pred} has at least one answer set; and for all the answer set S of the program π_{pred} , there exists $N \geq t_n$ such that $holds(f, k) \in S$ for every time point $k \in (N, t_{max}]$. \square

Explanation in AnsProlog Let \mathcal{D} be a domain description. For each fluent f , define $enum(f)$ be the AnsProlog program

$$\begin{aligned} holds(f, 0) &\leftarrow not\ holds(neg(f), 0). \\ holds(neg(f), 0) &\leftarrow not\ holds(f, 0). \end{aligned}$$

For each non-triggered action a and time point t , define $enum(a, t)$ be the following AnsProlog program.

$$\begin{aligned} holds(occurs(a), t) &\leftarrow not\ holds(ab(occurs(a)), t), \\ &\quad not\ holds(neg(occurs(a)), t). \\ holds(neg(occurs(a)), t) &\leftarrow not\ holds(occurs(a), t). \end{aligned}$$

For each non-triggered action a , we also define $enum(a)$ as follows.

$$enum(a) = \bigcup_{t=0}^{t_{max}-1} enum(a, t).$$

- $bound(cycD, cdk4)$ means that cyclin D is bound to cdk4. Similarly, $bound(cycE, cdk2)$ means that cyclin E is bound to cdk2.
- $bound(pRb, complex(e2f1, dp1))$ means that pRb protein is bound to the complex [E2F1:DP1] of E2F1 and DP1 protein.
- $phrylated(pRb, P^{(D)})$ and $phrylated(pRb, P^{(E)})$ mean pRb is respectively phosphorylated at subsets $P^{(D)}$ and $P^{(E)}$ of sites.
- $is_present(cdk2DN)$ represents the effect of the cdk2DN introduction.

The actions in the domain description can be divided into three groups:

1. actions related to the functions of the cyclin, including $bind(cycD, cdk4)$, $bind(cycE, cdk2)$, $phrylates(cycD, pRb)$, $phrylates(cycE, pRb)$; and
2. an action related to the function of pRb, which is $dissociates(pRb, complex(e2f1, dp1))$; and
3. an action related to performing experiments, namely $introduce(cdk2DN)$.

The meanings and effects of the actions are described as follows.

- $bind(cycD, cdk4)$ encode the action that cyclin D binds to cdk4, resulting in the complex [cyclin D: cdk4] of cyclin D and cdk4, in which cyclin D is bound to cdk4.

$bind(cycD, cdk4)$ **causes** $bound(cycD, cdk4)$.

- We encode the effect of $bind(cycE, cdk2)$ similarly to the case of action $bind(cycD, cdk4)$.

$bind(cycE, cdk2)$ **causes** $bound(cycE, cdk2)$.

- $phrylates(cycD, pRb)$ states that cyclin D phosphorylates pRb protein, which causes pRb being phosphorylated at the subset $P^{(D)}$ of sites.

$phrylates(cycD, pRb)$ **causes** $phrylated(pRb, P^{(D)})$.

If pRb is phosphorylated at sites $P^{(D)}$, it cannot be phosphorylated at the same sites. Hence, we have the following inhibition rule:

$phrylated(pRb, P^{(D)})$ **inhibits** $phrylates(cycD, pRb)$.

- $phrylates(cycE, pRb)$ denotes the action of cyclin E phosphorylating pRb protein. This action causes pRb being phosphorylated at the subset $P^{(E)}$, provided pRb has already been phosphorylated at $P^{(D)}$.

$phrylates(cycE, pRb)$ **causes** $phrylated(pRb, P^{(E)})$
if $phrylated(pRb, P^{(D)})$.

Similarly to the case of $phrylates(cycD, pRb)$, there is the inhibitory rule

$phrylated(pRb, P^{(E)})$ **inhibits** $phrylates(cycE, pRb)$.

- $dissociates(pRb, complex(e2f1, dp1))$ means that pRb dissociates from the [E2F1:DP1] complex.

$dissociates(pRb, complex(e2f1, dp1))$

causes $\neg bound(pRb, complex(e2f1, dp1))$.

The dissociation is not possible if pRb is not bound to the complex [E2F1:DP1], thus we have the rule

$\neg bound(pRb, complex(e2f1, dp1))$

inhibits $dissociates(pRb, complex(e2f1, dp1))$.

- Finally, $introduce(cdk2DN)$ corresponds to an experiment in which a foreign gene of cdk2DN is introduced into the cell, which causes cdk2DN protein to be present in the cell.

$introduce(cdk2DN)$ **causes** $is_present(cdk2DN)$.

In the presence of the dominant negative form cdk2DN, the [cyclin E: cdk2] complex will not be able to phosphorylate pRb; that is

$is_present(cdk2DN)$ **inhibits** $phrylates(cycE, pRb)$.

The triggering rules represent how the interactions of cyclins with their associated kinases regulate the function of pRb. The pRb protein is regulated by its states of phosphorylation. Once a complex is formed by a cyclin and its dependent kinase, it triggers a corresponding pRb phosphorylation:

$bound(cycD, cdk4)$ **n_triggers** $phrylates(cycD, pRb)$.

$bound(cycE, cdk2)$ **n_triggers** $phrylates(cycE, pRb)$.

pRb is said being "hyperphosphorylated", if it is phosphorylated at both sites $P^{(D)}$ and $P^{(E)}$. The hyperphosphorylation of pRb triggers the dissociation of pRb from the [E2F1:DP1] complex:

$phrylated(pRb, P^{(D)})$, $phrylated(pRb, P^{(E)})$

n_triggers $dissociates(pRb, complex(e2f1, dp1))$.

We have presented the domain description D_{pRb} of the regulatory network. Next, we show how to formulate and compute queries about this network.

Let s_0 be the initial state in which only fluent $bound(pRb, complex(e2f1, dp1))$ is true; that is, there is a unique protein complex which is formed by pRb being bound to [E2F2:DP1]. Let O_{init} be the initial state complete set of observations that is consistent with s_0 . Choosing $t_{max} = 5$ and applying the AnsProlog translation, we obtain the following results.

• Prediction

Phosphorylation of pRb by cyclin D is not sufficient to release pRb from [E2F1:DP1]:

$(D_{pRb}, O_{init}) \not\models bound(pRb, complex(e2f1, dp1))$
after $bind(cycD, cdk4)$ **at** 0.

Similarly, phosphorylation of pRb by cyclin E is not sufficient to release pRb from [E2F1:DP1]:

$(D_{pRb}, O_{init}) \not\models bound(pRb, complex(e2f1, dp1))$
after $bind(cycE, cdk2)$ **at** 0.

In order to have pRb dissociated from [E2F1:DP1], both the phosphorylation induced by cyclin D and E must happen in a sequence:

$(D_{pRb}, O_{init}) \models \neg bound(pRb, complex(e2f1, dp1))$
after $bind(cycD, cdk4)$ **at** 0, $bind(cycE, cdk2)$ **at** 1.
 $(D_{pRb}, O_{init}) \models \neg bound(pRb, complex(e2f1, dp1))$
after $bind(cycD, cdk4)$ **at** 0, $bind(cycE, cdk2)$ **at** 2.

• Explanation

We have seen that once the interactions between the cyclins and their dependent kinases are observed, pRb is expected to dissociate from [E2F1:DP1]. However, in the cdk2DN introduction experiment, after the binding of cyclin D to cdk4 and of cyclin E to cdk2, the binding of pRb to [E2F1:DP1] is observed staying intact. What can be an explanation for this “abnormal” behavior of pRb?

Let O be the observation

$O = \{ bound(cycD, cdk4) \text{ at } 0, bound(cycE, cdk2) \text{ at } 1, \\ bound(pRb, complex(e2f1, dp1)) \text{ at } 5 \}$

Then we have following explanations of O w.r.t (D_{pRb}, O_{init}) :

$E_1 = \{ introduce(cdk2DN) \text{ at } 0 \}.$
 $E_2 = \{ introduce(cdk2DN) \text{ at } 1 \}.$

Intuitively, the main mechanism would be that the introduction of cdk2DN has affected the cyclin E induced phosphorylation, which prevents pRb from being hyperphosphorylated. Consequently, binding of pRb to [E2F1:DP1] stays intact.

• Planning

Let $G = \neg bound(pRb, complex(e2f1, dp1))$. Then we find several plans, including:

$P_1 = \{ bind(cycD, cdk4) \text{ at } 0, bind(cycE, cdk2) \text{ at } 0 \}.$
 $P_2 = \{ bind(cycD, cdk4) \text{ at } 0, bind(cycE, cdk2) \text{ at } 2 \}.$
 $P_3 = \{ bind(cycE, cdk2) \text{ at } 0, bind(cycD, cdk4) \text{ at } 1 \}.$

Conclusions and future works

In this paper we introduced the action language \mathcal{A}_T^0 as a starting formalism for representation and reasoning about biological knowledge in molecular interactions domain. Taking a progressive approach toward the ultimate solution, we designed \mathcal{A}_T^0 with minimal features that afford sound theoretical and practical analysis of reasoning about triggered actions. We then showed how to do prediction, explanation and planning with respect to \mathcal{A}_T^0 using AnsProlog and discussed the applicability of our approach with respect to a biological example.

In regards to related works, (Baral & Tran 2003) gives a more general formalism for specifying evolution trajectories as part of an action language. The triggers in this paper, and the observation language in this paper is much more simpler with a focus on being able to implement in AnsProlog. Such an implementation is not a focus in (Baral & Tran 2003),

where the other focus is incorporating probabilities to the evolution trajectories.

In regards to the future we plan to expand \mathcal{A}_T^0 so as to better model molecular interactions. This involves issues such as incomplete information, non-determinism, feed-back loops, sensitization of molecules, and dealing with quantity of molecules.

Acknowledgments

The authors acknowledge the support of NSF through grant number 0070463 and NASA through grant number NCC2-1232, and the support of TGen and the college of Engineering of ASU for supporting the second author during the summer of 2003.

References

- A. Regev, W. S., and Shapiro, E. 2001. Representation and simulation of biochemical processes using π -calculus process algebra. In *Proceedings of the Pacific Symposium in BioComputing 2001 (PSB'01)*, 459–470.
- AFCS. Alliance for Cellular Signaling. <http://www.afcs.org>.
- Balduccini, M., and Gelfond, M. 2003. Diagnostic reasoning with A-Prolog. *Theory and Practice of Logic Programming* 3(4-5):425–461.
- Balduccini, M.; Gelfond, M.; Nogueira, M.; Watson, R.; and Barry, M. 2001. An A-Prolog decision support system for the space shuttle. In *AAAI Spring 2001 Symposium*.
- Balduccini, M.; Gelfond, M.; Nogueira, M.; and Watson, R. 2002. Planning with the USA-Advisor. In *3rd International NASA Workshop on Planning and Scheduling for Space*.
- Baral, C., and Gelfond, M. 2000. Reasoning agents in dynamic domains. In Minker, J., ed., *Logic-based artificial intelligence*. Kluwer Academic Publishers.
- Baral, C., and Tran, N. 2003. Representation and reasoning about evolution of the world in the context of reasoning about actions. In *Nonmonotonic Reasoning, Action, and Change (NRAC'03)*.
- Baral, C.; Floriano, L.; Hardesty, A.; Morales, D.; Nogueira, M.; and Tran, S. 1998. From theory to practice: the UTEP robot in AAAI 96 and AAAI 97 robot contests. In *Proceedings of the Second International Conference on Automated Agents (Agents 98)*, 32–38.
- Baral, C.; Chancellor, K.; Tran, N.; Tran, N.; and Berens, M. 2004. A knowledge based approach for representing and reasoning about signaling networks. In *Intelligent Systems for Molecular Biology/European Conference on Computational Biology 2004 (ISMB/ECCB 2004)*.
- Baral, C.; McIlraith, S. A.; and Tran, S. 2000. Formulating diagnostic problem solving using an action language with narratives and sensing. In *Principles of Knowledge Representation and Reasoning*, 311–322.
- Baral, C. 2003. *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press.

- Doherty, P.; Granlund, G.; Kuchcinski, K.; Sandewall, E.; Nordberg, K.; Skarman, E.; and Wiklund, J. 2000. The WITAS unmanned aerial vehicle project. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*, 747–755.
- Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions on Artificial Intelligence* 3.
- Grosskreutz, H., and Lakemeyer, G. 2000. cc-golog: Towards more realistic logic-based robot controllers. In *AAAI/IAAI*, 476–482.
- Lapouchnian, A., and Lesperance, Y. 2002. Interfacing IndiGolog and OAA - a toolkit for advanced multiagent applications. *Applied Artificial Intelligence* 16(9-10):813–829.
- Lundberg, A., and Weinberg, R. 1998. Functional inactivation of the retinoblastoma protein requires sequential modification by at least two distinct cyclin-cdk complexes. *Molecular and Cellular Biology* 18:753–761.
- Nogueira, M. 2003. *Building knowledge systems in A-Prolog*. PhD thesis.
- Peleg, M.; Yeh, I.; and Altman, R. B. 2002. Modelling biological processes using workflow and petri net models. *Bioinformatics* 18(6):825–837.
- Reddy, V. N.; Liebman, M. N.; and Mavrovouniotis, M. L. 1996. Qualitative analysis of biochemical reaction systems. *Computational Biology Medicine* 26:9–24.
- Reiter, R. 2001. *Knowledge in action : logical foundations for specifying and implementing dynamical systems*. MIT Press.
- Shanahan, M. P. 1998. A logical account of the common sense informatic situation for a mobile robot. *Electronic Transactions on Artificial Intelligence* 2:69–104.
- STKE. Signal Transduction Knowledge Environment. <http://www.stke.org>.
- Thielscher, M. 2000. Representing the knowledge of a robot. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 109–120.
- Zarkowska, R., and Mitnacht, S. 1997. Differential phosphorylation of the retinoblastoma protein by G1/S cyclin-dependent kinases. *Journal of Biological Chemistry* 272:12738–12746.