Some Results on the Completeness of Approximation Based Reasoning

Tran Cao Son, Enrico Pontelli, To Thanh Son
Department of Computer Science
New Mexico State University
P.O. Box 30001, MSC CS
Las Cruces, NM 88003, USA

Abstract

We present two results that relate the completeness conditions for the 0-approximation for two formalisms: the action description language $\mathcal{A}$ and the situation calculus. The first result indicates that the completeness condition for the situation calculus formalism implies the corresponding condition for the action language formalism. The second result indicates that an action theory in $\mathcal{A}$ can sometimes be simplified to an equivalent action theory whose completeness condition is weaker than the original theory for certain queries.

Introduction

Intelligent agents need to be able to reason about the effects of their actions (*RAC*) and to make decisions based on this reasoning. This has been the main thesis of the agent architecture proposed in (Baral & Gelfond 2000) and further developed in (Balduccini & Gelfond 2003). One of the most important problems in RAC is to determine whether a fluent formula φ is true after the execution of an action sequence α from an initial state δ—denoted by the query φ **after** α (a.k.a. *hypothetical reasoning*). Most formalisms for RAC solve this problem by defining an entailment relationship, denoted by $\models$, between action theories and queries (see, e.g., (Gelfond & Lifschitz 1993)). Let $\mathcal{D}$ denote an action domain and δ an initial state. We write $(\mathcal{D}, \delta) \models \varphi$ **after** α to denote that the action theory $(\mathcal{D}, \delta)$ entails φ **after** α—i.e., φ is true after the execution of α from the initial state. The majority of the original approaches to RAC define $\models$ assuming that δ is a *complete description* of the initial state.

The possible world semantics can be employed to reason about effects of actions in presence of incomplete information (Moore 1985). In this approach, a fluent formula φ is true after the execution of an action sequence α iff it is true after the execution of α in *every* possible initial state of the world. In simple words, an entailment relationship $\models^P$ is defined by adapting $\models$ to deal with incomplete information. It states that $(\mathcal{D}, \delta) \models^P \varphi$ **after** α iff for every possible completion δ' of δ, $(\mathcal{D}, \delta') \models \varphi$ **after** α. δ' is a completion of δ if it contains δ and it is a complete description of the initial state.

One main disadvantage of the possible world semantics is its high complexity. For example, (Baral, Kreinovich, & Trejo 2000) showed that, even for deterministic action theories, determining whether a fluent is true or false after the execution of a single action is co-NP complete. Moreover, the presence of incomplete information makes the *planning problem*—another important problem in RAC—computationally harder (see, e.g., (Baral, Kreinovich, & Trejo 2000)).

An alternative to the possible world semantics is the reasoning based on approximations (Son & Baral 2001). Instead of considering all possible states, approximations define what will *definitely* be true or false after the execution of an action. This approach reduces the complexity of the hypothetical reasoning but is in general incomplete. This stipulates the research in (Liu & Levesque 2005; Son & Tu 2006; Tu 2007) to determine conditions under which the approximation is complete. While the approach in (Liu & Levesque 2005) addresses the question *"when does the reasoning based on approximation coincide with the possible world semantics?"*, the approach in (Son & Tu 2006) focuses on answering the question *"when does the reasoning based on approximation for a particular fluent formula φ coincide with the possible world semantics?"*

In this paper, we investigate the relationship between these two completeness conditions. We will review the basic definitions associated to the completeness conditions of (Liu & Levesque 2005; Son & Tu 2006) in the next section. We discuss the relationships between them, and develop a transformation for simplification of action theories. Finally, we present a result that directly relates the two conditions.

Approximation Based Reasoning

We review the situation calculus language, the action language $\mathcal{A}$, and the 0-approximation in these two formalisms.

Situation Calculus

Situation calculus was introduced by McCarthy (1959) and further developed by McCarthy and Hayes (1969). It is probably the oldest formalism for representing and reasoning about actions. In situation calculus, actions and their effects are encoded directly into a first order theory. The basic components of the situation calculus language in the notation of Reiter (1991) include a special constant S_0, denoting the initial situation, a binary function symbol Do, where $Do(a, s)$ denotes the successor situation to s resulting from executing the action a, $fluent$ relations of the form $F(s)$ (or $F(\vec{x}, s)$), denoting that the fluent F (resp. $F(\vec{x})$) is true in the situation s, and a special predicate $Poss(a, s)$ (resp. $Poss(a(\vec{x}), s)$), denoting that action a (resp. $a(\vec{x})$) is executable in situation s.[1]

A dynamic domain can be represented by a theory $\mathcal{D}$ composed of *(i)* axioms describing the initial situation S_0; *(ii)* action precondition axioms (one for each primitive action A, characterizing $Poss(A, s)$); *(iii)* successor state axioms (SSA) (one for each fluent F, stating under what condition $F(Do(a, s))$ holds, as a function of what holds in s); *(iv)* unique name axioms for the primitive actions; and some foundational, domain independent axioms. In particular, each domain $\mathcal{D}$ is given by a set of axioms

$$\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una}$$

where $\mathcal{D}_0$, $\mathcal{D}_{ap}$, $\mathcal{D}_{ss}$, and $\mathcal{D}_{una}$ encode the axioms about initial situation, the action preconditions, the successor state axioms, and the unique name axioms, respectively. Each axiom in $\mathcal{D}_{ap}$ is in the form $Poss(a, s) \equiv \Pi_a[s]$ and each axiom in $\mathcal{D}_{ss}$ is of the form

$$F(Do(a, s)) \equiv \gamma_F^+(a, s) \vee (F(s) \wedge \neg\gamma_F^-(a, s)).$$

Example 1. Let us consider the *bomb in the toilet* example from (McDermott 1987), assuming that we do not have any knowledge about the initial situation. In this domain, we have two actions $Dunk$ and $Flush$ and two fluents $Clogged$ and $Armed$. Dunking a packet into the toilet disarms the bomb but causes the toilet to be clogged. Flushing the toilet makes it unclogged.

The basic action theory for this domain is given next[2]

$$\mathcal{D}^b = \mathcal{D}_0^b \cup \mathcal{D}_{ap}^b \cup \mathcal{D}_{ss}^b \cup \mathcal{D}_{una}^b$$

- The action precondition axioms ($\mathcal{D}_{ap}^b$) are
  - $Poss(Flush, s) \equiv \top$ (i.e., $\Pi_{Flush} = \top$).
  - $Poss(Dunk, s) \equiv \neg Clogged(s)$ ($\Pi_{Dunk} = \neg Clogged$).

  Note that Π_A is a formula in the language $\mathcal{L}$, whose propositions are the fluents in $\mathcal{D}$, where A is either $Flush$ or $Dunk$.
- The successor state axioms ($\mathcal{D}_{ss}^b$) for the fluents:
  - $Clogged(Do(a, s)) \equiv (a = Dunk) \vee (Clogged(s) \wedge \neg(a = Flush))$

    Here, $\gamma_{Clogged}^+(a) = (a = Dunk)$ and $\gamma_{Clogged}^-(a) = (a = Flush)$.
  - $Armed(Do(a, s)) \equiv \bot \vee (Armed(s) \wedge \neg(a = Dunk))$

    ($\gamma_{Armed}^+(a) = \bot$ and $\gamma_{Armed}^-(a) = (a = Dunk)$).
- $\mathcal{D}_{una}^b$ contains the following axiom: $Dunk \neq Flush$. This is because we only have 0-ary actions.
- $\mathcal{D}_0^b$ is empty. □

In the situation calculus, to determine whether φ is true after the execution of the action sequence α, we determine whether $\mathcal{D} \models \varphi(Do(\alpha, S_0)) \wedge Poss(\alpha, S_0)$ where, for an action a and action sequence α, $Do([a, \alpha], s)$ stands for $Do(\alpha, Do(a, s)))$, $Poss([a, \alpha], s)$ is the shorthand for $Poss(a, s) \wedge Poss(\alpha, Do(a, s)))$, and $\models$ is the logical entailment relationship in first order logic. [3] In this way, the possible world semantics is naturally employed as S_0 can be incomplete. In fact, we can easily check that

$$\begin{aligned} \mathcal{D}^b \quad \models \quad & Poss([Flush, Dunk], S_0) \wedge \\ & \neg Armed(Do([Flush, Dunk], S_0)). \end{aligned} \tag{1}$$

As we have mentioned, the progression problem becomes computationally harder (assuming that $NP \neq P$) in the presence of incomplete information about the initial situation. This has motivated Liu and Levesque (2005) to explore the use of approximation for the progression task and develop conditions under which the reasoning based on the approximation is complete. Their formulation is inspired by the reasoning algorithm developed for *proper knowledge bases* (Levesque 1998) and is restricted to local effect theories. Formally, these notions are defined as follows.[4]

- Situations are described by *proper* knowledge bases (proper KBs), i.e., theories Σ which are consistent (w.r.t. the axioms of equality) and where all formulae can be expressed in the form $\forall(e \supset \ell)$, where e is a quantifier-free formula containing only equalities and ℓ is a literal. We denote with ξ_ℓ the maximal disjunction of ground instances of the literal ℓ such that $\Sigma \models \xi_\ell$.
- A proper KB Σ is *complete* w.r.t. a fluent F if either $\Sigma \models F$ or $\Sigma \models \neg F$. Σ is *context-complete* w.r.t. the theory $\mathcal{D}$ if it is complete w.r.t. each F appearing in any γ_G^+ or γ_G^-.
- The theory $\mathcal{D}$ is assumed to be *local effect*, i.e., γ_F^+ and γ_F^- are finite disjunctions of formulae of the form $(a = A \wedge \varphi)$, where A is a ground action and the various φ are ground formulae. Given a ground action A and a ground fluent F, we will denote with F_A^+ (resp. F_A^-) the formula $\bigvee\{\varphi \mid (a = A \wedge \varphi) \textit{ appears in } \gamma_F^+\}$ (resp. $\bigvee\{\varphi \mid (a = A \wedge \varphi) \textit{ appears in } \gamma_F^-\}$).

Evaluation of formulae φ w.r.t. a proper KB Σ is based on a 3-value interpretation function $V(\Sigma, \varphi)$, which is sound, and can be proved complete when the formula meets certain criteria ($\mathcal{NF}$ normal form).

Liu and Levesque provide a definition of progression which preserves the *proper* property of the encoding of situations. In presence of a finite domain of constants, progression of a proper KB Σ w.r.t. a ground action A ($\mathcal{P}_A(\Sigma)$) is defined as the sentences: for each ground fluent F

$$\begin{aligned} def\_true_F \vee (\xi_F \wedge \neg poss\_false_F) \quad & \supset \quad F \\ def\_false_F \vee (\xi_{\neg F} \wedge \neg poss\_true_F) \quad & \supset \quad \neg F \end{aligned}$$

[1] For simplicity, we omit the parameters of actions and fluents.

[2] We use $\top$ and $\bot$ to denote true and false respectively.

[3] Strictly speaking, we need to add foundational axioms to $\mathcal{D}$.

[4] For simplicity, we assume a propositional language.

where

$$
\begin{aligned}
def\_true_F &= \begin{cases} \top & V(\Sigma, F_A^+) = 1 \\ \bot & \textit{otherwise} \end{cases} \\
poss\_true_F &= \begin{cases} \top & V(\Sigma, F_A^+) \neq 0 \\ \bot & \textit{otherwise} \end{cases} \\
def\_false_F &= \begin{cases} \top & V(\Sigma, F_A^-) = 1 \textit{ and } \\ & \quad V(\Sigma, F_A^+) = 0 \\ \bot & \textit{otherwise} \end{cases} \\
poss\_false_F &= \begin{cases} \top & V(\Sigma, F_A^-) \neq 0 \\ \bot & \textit{otherwise} \end{cases}
\end{aligned}
$$

The notion can be generalized to sequences of actions $\alpha = [A_1, \ldots, A_n]$, as $\mathcal{P}_\alpha = \mathcal{P}_{A_n} \circ \ldots \circ \mathcal{P}_{A_1}$. If Σ_0 is context complete and $\mathcal{D}$ is local effect then $\mathcal{P}_A(\Sigma_0)$ can be shown to be a classical progression.

Example 2. For the theory $\mathcal{D}^b$, the language of the knowledge base Σ_0 consists of two predicates, $Armed$ and $Clogged$, and there is no constant or function symbol in this language. The formulae expressing progression for the theory $\mathcal{D}^b$ are given below:

- for the action $Dunk$ we have

$$
\begin{array}{lcl}
\bot \vee (\xi_{Armed} \wedge \neg\top) & \supset & Armed \\
\top \vee (\xi_{\neg Armed} \wedge \neg\bot) & \supset & \neg Armed \\
\top \vee (\xi_{Clogged} \wedge \neg\bot) & \supset & Clogged \\
\bot \vee (\xi_{\neg Clogged} \wedge \neg\top) & \supset & \neg Clogged
\end{array}
$$

  This simplifies to $\top \supset \neg Armed$ and $\top \supset Clogged$.

- for the action $Flush$ we have

$$
\begin{array}{lcl}
\bot \vee (\xi_{Armed} \wedge \neg\bot) & \supset & Armed \\
\bot \vee (\xi_{\neg Armed} \wedge \neg\bot) & \supset & \neg Armed \\
\bot \vee (\xi_{Clogged} \wedge \neg\top) & \supset & Clogged \\
\top \vee (\xi_{\neg Clogged} \wedge \neg\bot) & \supset & \neg Clogged
\end{array}
$$

  So, we have $\xi_{Armed} \supset \neg Armed$ and $\top \supset \neg Clogged$.

This computation allows us to conclude that the approximation based reasoning in $\mathcal{D}^b$ will yield the same conclusion as in Eq. 1. Observe that this can also be inferred from the fact that $\mathcal{D}^b$ is *context-complete*. □

Language $\mathcal{A}$ and the 0-approximation Semantics

Gelfond and Lifschitz (1993) defined the language $\mathcal{A}$ for representing actions and reasoning about their effects. Dynamic domains are represented by action descriptions, whose semantics is defined by a transition function which maps pairs of actions and states into states.

The alphabet of a domain consists of a set $\mathbf{A}$ of action names and a set $\mathbf{F}$ of fluent names. A (fluent) literal l is either a fluent $f \in \mathbf{F}$ or its negation $\neg f$. Fluent literals of the forms f and $\neg f$ are said to be complementary to each other. By $\mathbf{L}$ we denote the set of all fluent literals, i.e., $\mathbf{L} = \{f, \neg f \mid f \in \mathbf{F}\}$. A *fluent formula* is a formula constructed from fluent literals using connectives $\wedge$, $\vee$, and $\neg$. A *domain description* $\mathcal{D}$ is a set of statements of the following forms:

$$a \textbf{ causes } l \textbf{ if } \psi \tag{2}$$

$$\textbf{executable } a \textbf{ if } \psi \tag{3}$$

where $a \in \mathbf{A}$ is an action, l is a fluent literal, and ψ is a set of fluent literals. (2) is called a *dynamic law*, describing the effect of action a. It says that, if a is performed in a situation where ψ holds, then l will hold in the successor state. (3) is an *executability condition* on a, stating that a is executable in any situation in which ψ holds.

Given a domain description $\mathcal{D}$, for a fluent literal l, we denote by $\neg l$ its complementary literal. For a set of fluent literals σ, we denote by $\neg\sigma$ the set $\{\neg l \mid l \in \sigma\}$. A set of fluent literals σ is consistent if for every fluent f, either f or $\neg f$ does not belong to σ. We will use the two terms *consistent set of fluent literals* and *partial state* interchangeably. A set of fluent literals σ is *complete* if, for every fluent f, either f or $\neg f$ belongs to σ. When σ is consistent and complete, it is called a *state*. A state s containing a partial state δ is called a *completion* of δ. For a partial state δ, we denote with $ext(\delta)$ the set of all completions of δ. For a set of partial states Δ, we denote with $ext(\Delta)$ the set of states $\cup_{\delta \in \Delta} ext(\delta)$.

A fluent literal l (resp. set of fluent literals γ) *holds* in a consistent set of fluent literals σ if $l \in \sigma$ (resp. $\gamma \subseteq \sigma$). l (resp. γ) *possibly holds* in σ if $\neg l \notin \sigma$ (resp. $\neg\gamma \cap \sigma = \emptyset$). The value of a formula φ in σ may be either true, false, or unknown and is defined as usual. It is easy to see that if σ is a state then for every formula φ, the value of φ is known (either true or false) in σ. From now on, to avoid confusion, we will use letters (possibly indexed) σ, δ, and s to denote a set of fluent literals, a partial state, and a state respectively.

An $\mathcal{A}$ action theory is a pair $(\mathcal{D}, \Delta)$ where $\mathcal{D}$ is a domain description and Δ is a set of partial states.

Example 3. The bomb in the toilet example is represented by $\Delta_1 = \emptyset$ and the action theory

$$
\mathcal{D}_1 = \begin{cases}
Dunk \textbf{ causes } \neg Armed \\
Dunk \textbf{ causes } Clogged \\
Flush \textbf{ causes } \neg Clogged \\
\textbf{executable } Dunk \textbf{ if } \neg Clogged \\
\textbf{executable } Flush \textbf{ if } \top
\end{cases}
$$

□

The 0-approximation semantics is introduced in (Son & Baral 2001) and is reviewed next. Let $\mathcal{D}$ be a domain description. An action a is *executable* in a partial state δ if there exists an executability condition

$$\textbf{executable } a \textbf{ if } \psi$$

in $\mathcal{D}$ such that ψ holds in δ.

For an action a and a partial state δ s.t. a is executable in δ, the set of *effects* of a in δ, denoted by $e(a, \delta)$, and the set of *possible effects* of a in δ, denoted by $pe(a, \delta)$, are:

$$
\begin{aligned}
e(a, \delta) &= \{l \mid \text{there exists } [a \textbf{ causes } l \textbf{ if } \psi] \text{ in } \mathcal{D} \\ &\qquad \text{s.t. } \psi \text{ holds in } \delta\} \\
pe(a, \delta) &= \{l \mid \text{there exists } [a \textbf{ causes } l \textbf{ if } \psi] \text{ in } \mathcal{D} \\ &\qquad \text{s.t. } \psi \text{ possibly holds in } \delta\}
\end{aligned}
$$

Intuitively, $e(a, \delta)$ and $pe(a, \delta)$ are the sets of literals that *certainly hold* and *may hold*, respectively, in the successor state of every state $s \in ext(\delta)$. The *successor partial state* of a state s after the execution of an action a, denoted by $\Phi^0(a, \delta)$, is defined as follows.

Definition 1. *For any action a and partial state δ,*

1. *if a is not executable in δ then $\Phi^0(a, \delta) = \bot$;*
2. *otherwise, $\Phi^0(a, \delta) = e(a, \delta) \cup (\delta \setminus \neg pe(a, \delta))$.*

The *final partial state* of a partial state δ after the execution of a sequence of actions α, denoted by $\widehat{\Phi}^0(\alpha, \delta)$, is defined as follows.

Definition 2. *For any sequence of actions $\alpha = [a, \beta]$ and partial state δ,*

1. *$\widehat{\Phi}^0([], \delta) = \delta$ and if $\beta = []$ then $\widehat{\Phi}^0(\alpha, \delta) = \Phi^0(a, \delta)$;*
2. *otherwise, $\widehat{\Phi}^0(\alpha, \delta) = \widehat{\Phi}^0(\beta, \Phi^0(a, \delta))$.*

Given the extended transition function $\widehat{\Phi}^0$, the entailment relationship between an action theory and a query with respect to the 0-approximation semantics, denoted by $\models^0$, is defined as follows (recall that Δ is a set partial states).

Definition 3. *An action theory $(\mathcal{D}, \Delta)$ entails a query $[\varphi \textbf{ after } \alpha]$ with respect to the 0-approximation semantics, denoted by $(\mathcal{D}, \Delta) \models^0 \varphi \textbf{ after } \alpha$, if for every $\delta \in \Delta$, $\widehat{\Phi}^0(\alpha, \delta) \neq \bot$ and φ is true in $\widehat{\Phi}^0(\alpha, \delta)$.*

It should be noted that the transition function $\Phi_0(a, \delta)$ coincides with the transition function defined for complete action theories in (Gelfond & Lifschitz 1993) if δ is complete. As such, the possible world semantics for an action theory $(\mathcal{D}, \Delta)$ can be characterized by $\widehat{\Phi}^0$ of the theory $(\mathcal{D}, ext(\Delta))$. For convenience of our discussion, we say $(\mathcal{D}, \Delta) \models^P \varphi$ **after** α iff $(\mathcal{D}, ext(\Delta)) \models^0 \varphi$ **after** α. The following example demonstrates the use of the 0-entailment in reasoning about the effects of actions.

Example 4. Consider the action theory $(\mathcal{D}_1, \Delta_1)$ from Example 3. We have $e(Flush, \emptyset) = pe(Flush, \emptyset) = \{\neg Clogged\}$. Hence, $\Phi^0(Flush, \emptyset) = e(Flush, \emptyset) \cup (\emptyset \setminus \neg pe(Flush, \emptyset)) = \{\neg Clogged\} = \delta_1$. Furthermore, we have $e(Dunk, \delta_1) = \{Clogged, \neg Armed\}$ and $pe(Dunk, \delta_1) = \{Clogged, \neg Armed\}$. Thus,

$$\begin{aligned} \Phi^0(Dunk, \delta_1) &= e(Dunk, \delta_1) \cup (\delta_1 \setminus \neg pe(Dunk, \delta_1)) \\ &= \{Clogged, \neg Armed\} \end{aligned}$$

This also implies that

$$(\mathcal{D}_1, \Delta_1) \models^0 \neg Armed \textbf{ after } [Flush; Dunk] \qquad (4)$$

□

The 0-approximation is sound (Son & Baral 2001), but it is, in general, incomplete.

Example 5. Let $\mathcal{D}_2$ be the domain obtained from $\mathcal{D}_1$ by replacing the axiom $[Dunk \textbf{ causes } \neg Armed]$ by the axiom

$$Dunk \textbf{ causes } \neg Armed \textbf{ if } Armed$$

We can easily check that $\neg Armed$ is true after the execution of $[Flush; Dunk]$ if the possible world semantics is used. However, $(\mathcal{D}_2, \Delta_1) \not\models^0 \neg Armed$ **after** $[Flush; Dunk]$. This shows that the 0-approximation is incomplete. □

The incompleteness of the 0-approximation motivated Son and Tu (2006) to find conditions for its completeness. In particular, given a fluent formula φ and $(\mathcal{D}, \Delta)$, they investigate the conditions under which $(\mathcal{D}, \Delta) \models^0 \varphi$ **after** α iff $(\mathcal{D}, \Delta) \models^P \varphi$ **after** α holds. Their conditions are based on the notion of dependency between fluent literals and the reducibility of a set of states to a partial state.

Definition 4. *Let $\mathcal{D}$ be a domain description. A fluent literal l depends on a fluent literal g, written as $l \lhd g$, iff one of the following conditions holds.*

1. *$l = g$.*
2. *$\mathcal{D}$ contains a law $[a \textbf{ causes } l \textbf{ if } \psi]$ such that $g \in \psi$.*
3. *There exists a fluent literal h such that $l \lhd h$ and $h \lhd g$.*
4. *The complement of l depends on the complement of g, i.e., $\neg l \lhd \neg g$.*

Note that the dependency relationship between fluent literals is reflexive, transitive but not symmetric. We next define the dependency between actions and fluent literals.

Definition 5. *Let $\mathcal{D}$ be a domain description. An action a depends on a fluent literal l, written as $a \lhd l$, iff one of the following conditions is satisfied.*

1. *$\mathcal{D}$ contains the statement $[\textbf{executable } a \textbf{ if } \psi]$ such that $l \in \psi$.*
2. *There exists a fluent literal g such that $a \lhd g$ and $g \lhd l$.*

For a fluent literal l (resp. action a), we will denote by $\Omega(l)$ (resp. $\Omega(a)$) the set of fluent literals that l (resp. a) depends on. A fluent literal l (resp. an action a) depends on a set of fluent literals σ, denoted by $l \lhd \sigma$ (resp. $a \lhd \sigma$), iff $l \lhd g$ (resp. $a \lhd g$) for some $g \in \sigma$.

A *disjunction* of fluent literals $\gamma = l_1 \vee \ldots \vee l_k$ depends on a set of fluent literals σ, denoted by $\gamma \lhd \sigma$ if there exists $1 \leq j \leq k$ such that $l_j \lhd \sigma$; otherwise, γ does not depend on σ, denoted by $\gamma \not\lhd \sigma$.

Example 6. For the domain description $\mathcal{D}_2$, we have

$\Omega(Clogged) = \{Clogged\}$
$\Omega(\neg Clogged) = \{\neg Clogged\}$
$\Omega(Armed) = \{Armed, \neg Armed\}$
$\Omega(\neg Armed) = \{Armed, \neg Armed\}$
$\Omega(Dunk) = \{\neg Clogged\}$ $\Omega(Flush) = \emptyset$ □

Definition 6. *Let $\mathcal{D}$ be a domain description. Let S be a belief state (i.e., a set of states), δ be a partial state, and $\varphi = \gamma_1 \wedge \ldots \wedge \gamma_n$ be a fluent formula where each γ_i is a disjunction of fluent literals. We say that S is* reducible *to δ with respect to φ, denoted by $S \gg_\varphi \delta$ if*

1. *δ is a subset of every state s in S*
2. *For every $1 \leq i \leq n$, there exists a state $s \in S$ such that $\gamma_i \not\lhd (s \setminus \delta)$*
3. *For any action a, there exists a state $s \in S$ such that $a \not\lhd (s \setminus \delta)$.*

Intuitively, the above definition specifies a condition under which reasoning using belief states (represented by S) can be done using the 0-approximation using the approximation state δ. For a set of partial states Δ, we say that $ext(\Delta) \gg_\varphi \Delta$ if for every $\delta \in \Delta$, $ext(\delta) \gg_\varphi \delta$. Indeed, it has been shown in (Tu 2007) that

Theorem 1. *Let $(\mathcal{D}, \Delta)$ be an action theory and $ext(\Delta) \gg_\varphi \Delta$. For every sequence of actions α, $(\mathcal{D}, \Delta) \models^P \varphi$ **after** α iff $(\mathcal{D}, \Delta) \models^0 \varphi$ **after** α.*

Example 7. Consider $\mathcal{D}_2$ (Example 5). Let $\delta_1 = \emptyset$. Then, $S_1 = ext(\delta_1)$ is not reducible to δ_1 with respect to $\varphi = \neg Armed$, i.e., $ext(\emptyset) \not\gg_{\neg Armed} \emptyset$, because for each $s \in S_1$, either fluent literal $Armed$ or $\neg Armed$ belongs to $s \setminus \delta_1$ and the fluent literal $\neg Armed$ depends on both $Armed$ and $\neg Armed$ (Condition 2 in Def. 6 is not satisfied). Similarly,

$$ext(\{Clogged\}) \not\gg_{\neg Armed} \{Clogged\}$$
$$ext(\{\neg Clogged\}) \not\gg_{\neg Armed} \{\neg Clogged\}$$

Let $\delta_2 = \{Armed\}$. Then we have $ext(\delta_2) = \{s_1, s_2\}$, where $s_1 = \{Armed, Clogged\}$ and $s_2 = \{Armed, \neg Clogged\}$. We can easily check that $ext(\delta_2) \gg_{\neg Armed} \delta_2$. Hence, we have $ext(\{Armed\}) \gg_{\neg Armed} \{Armed\}$. Similarly, we can check that $ext(\{\neg Armed\}) \gg_{\neg Armed} \{\neg Armed\}$. □

Relating the Two Completeness Conditions Some Results

The equivalence between situation calculus and the action language $\mathcal{A}$ has been proved in (Kartha 1993) for the case where the initial situation is completely described. A theory in situation calculus $\mathcal{D}$ is complete if $\mathcal{D}_{S_0} \models F(S_0)$ or $\mathcal{D}_{S_0} \models \neg F(S_0)$ for each fluent F in $\mathcal{D}$. We wish to explore the relationship between the two completeness conditions described above. Let us first discuss some differences between the two formulations through some examples.

Example 8. Consider the theory $\mathcal{D}$ with one action A and three fluents F, P, and Q. We assume $\gamma_F^+(a) = (a = A \wedge P) \vee (a = A \wedge Q)$, $\gamma_F^-(a) = \bot$, and, for $X \in \{P, Q\}$, $\gamma_X^+(a) = \gamma_X^-(a) = \bot$. Furthermore, let $\mathcal{D}_{S_0} = \{P(S_0)\}$. For $\mathcal{D}$ to be context-complete, we have that P and Q should be known in S_0. That is, S_0 should also contain Q or $\neg Q$ to guarantee that $V(F, \Sigma_0)$ is either 0 or 1.

A reasonable description of $\mathcal{D}$ in the language $\mathcal{A}$, say $\mathcal{D}_\mathcal{A}$, consists of two dynamic laws A **causes** F **if** P and A **causes** F **if** Q and $\Delta = \{\delta_0\}$, where $\delta_0 = \{P\}$. It is easy to check that $ext(\delta_0) \gg_F \delta_0$ and

$$(\mathcal{D}_\mathcal{A}, \Delta) \models^0 F \textbf{ after } A.$$

Thus, the knowledge about Q is not necessary in determining whether $V(F, \Sigma_0)$ is 1 or 0. □

This example shows that, to answer certain queries, the context-complete requirement of (Liu & Levesque 2005) might be too strong. The next example shows that, on the other hand, the 0-approximation is somewhat more sensitive to the specification of effects of actions in $\mathcal{A}$.

Example 9. Consider the action theories $(\mathcal{D}_1, \Delta_1)$ and $(\mathcal{D}_2, \Delta_1)$. Recall that the difference between $\mathcal{D}_1$ and $\mathcal{D}_2$ lies in that $[Dunk$ **causes** $\neg Armed]$ in $\mathcal{D}_1$ is replaced by

$$Dunk \textbf{ causes } \neg Armed \textbf{ if } Armed$$

in $\mathcal{D}_2$. Intuitively, these two representations are equivalent, in the sense that, for each complete state of the world s, the execution of $Dunk$ in s results in the same state in which the bomb is disarmed, as it is either a direct effect of $Dunk$ or it is true by inertia. On the other hand, Examples 4-7 show that $\models^0$ is complete for the fluent formula $\neg Armed$ w.r.t. $\mathcal{D}_1$ but not w.r.t. $\mathcal{D}_2$. □

The second example shows a weakness of the 0-approximation in that it is more sensitive to the domain specification than the situation calculus formalism. This is reflected by the completeness condition in (Son & Tu 2006): we have that $ext(\Delta_1) \gg_{\neg Armed} \Delta_1$ w.r.t. $\mathcal{D}_1$ but $ext(\Delta_1) \not\gg_{\neg Armed} \Delta_1$ w.r.t. $\mathcal{D}_2$. Why does the situation calculus formulation not suffer from this problem? The main reason is in the encoding of the successor state axioms. For example, the successor state axiom for $Armed$ is

$$Armed(Do(a,s)) \equiv \bot \vee$$
$$(Armed(s) \wedge \neg(a{=}Dunk \wedge Armed(s)))$$

Here, the precondition $Armed$ of the conditional effect $\neg Armed$ of $Dunk$ (in the dynamic law $Dunk$ **causes** $\neg Armed$ **if** $Armed$) is encoded as part of the formula $\gamma^-_{Armed}(Dunk)$. However, this can be simplified to

$$Armed(Do(a,s)) \equiv \bot \vee (Armed(s) \wedge \neg(a = Dunk))$$

which effectively makes $\neg Armed$ an effect of $Dunk$. This argument can be seen as a justification for the simplification of $\mathcal{D}_2$ to $\mathcal{D}_1$ in the action language $\mathcal{A}$.

At this point, one might be tempted to conclude that, if the complement of a fluent literal L appears in φ of a dynamic law A **causes** L **if** φ, then we can remove it from φ and the resulting action theory remains faithful to its original representation. This is, however, not the case. Consider a domain with one action $Flip$, that toggles a light bulb switch. Its effect is to makes the switch On and $\neg On$ if it was $\neg On$ and On, respectively. This is expressed by the two laws

$$Flip \textbf{ causes } On \textbf{ if } \neg On \quad \text{and} \quad Flip \textbf{ causes } \neg On \textbf{ if } On$$

Removing $\neg On$ or On in the **if** part from the first or second law respectively would create an inconsistent domain. Interestingly, the successor state axiom for On is

$$On(Do(a,S)) \equiv \gamma^+_{On}(a)[s] \vee (On(s) \wedge \neg\gamma^-_{On}(a)[s])$$

where $\gamma^+_{On}(a) \equiv a = Flip \wedge \neg On$ and $\gamma^-_{On}(a) \equiv a = Flip \wedge On$. The SSA for On could be simplified to

$$On(Do(a,S)) \equiv \gamma^+_{On}(a)[s] \vee (On(s) \wedge \neg(a = Flip))$$

Observe that the second representation would yield the same set of models for the theory. Nevertheless, setting $\gamma^-_{On}(a) \equiv a = Flip$ would violate the consistency condition by Reiter (1991) which says that $\not\exists s.[Poss(a,s) \supset \gamma^+_F(a,s) \wedge \gamma^- F(a,s)]$. This means that γ^-_{On} cannot be simplified in this case. On the other hand, the simplification for $\gamma^-_{Armed}(a)$ to $a = Dunk$ is acceptable since the consistency condition is satisfied by the new formula. This discussion suggests a simplification of $\mathcal{A}$ action theories.

Definition 7. *For a fluent literal L, a dynamic law A* **causes** *L* **if** *φ is* (A, L)-cyclic *if $\neg L \in \varphi$. L and $\neg L$ are* A-relevant *in $\mathcal{D}$ if $\mathcal{D}$ contains at least one (A, L)-cyclic and one $(A, \neg L)$-cyclic dynamic law.*

L and $\neg L$ are A-*irrelevant* if they are not A-relevant. Let $\mathcal{D}$ be a domain description and $\mathcal{D}_R$ be the domain description obtained from $\mathcal{D}$ by replacing each (A, L)-cyclic dynamic law A **causes** L **if** φ with A **causes** L **if** $\varphi \setminus \{\neg L\}$ if L and $\neg L$ are A-irrelevant.

We can see that $Armed$ and $\neg Armed$ are $Dunk$-irrelevant in both theories $\mathcal{D}_1$ and $\mathcal{D}_2$. On the other hand, Examples 4-7 show that $\mathcal{D}_1$ is preferable to $\mathcal{D}_2$ for dealing with incomplete information. We can show that

Theorem 2. *For every set of partial states* Δ*,* $(\mathcal{D}, \Delta)$ *is equivalent to* $(\mathcal{D}_R, \Delta)$ *w.r.t. the possible world semantics.*

From Situation Calculus Theory to Action Theory

We will now show a result relating the context-completeness condition in (Liu & Levesque 2005) and the reducibility condition in (Son & Tu 2006). We begin with a translation of local effect situation calculus theories into $\mathcal{A}$ action theories and conclude with a theorem relating these two conditions.

Our translation from a situation calculus theory $\mathcal{D}$ into $(\mathcal{D}_\mathcal{A}, \Delta_\mathcal{A})$ is inspired by the translation in (Kartha 1993) (we will only deal with propositional theories). Assume that

$$\mathcal{D} = \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una} \cup \mathcal{D}_0$$

Let **F** and **A** denote the set of fluents and actions of $(\mathcal{D}_\mathcal{A}, \Delta_\mathcal{A})$ respectively. The translation is done as follows.

- For each action precondition axiom

  $$Poss(A, s) \equiv \Pi_A[s] \text{ in } \mathcal{D}_{ap}$$

  A belongs to **A** and $\mathcal{D}_\mathcal{A}$ contains **executable** A **if** Π_A.
- For each successor state axiom

  $$F(do(a, s)) \equiv \gamma^+_F(a)[s] \vee (F(s) \wedge \neg\gamma^-_F(a)[s]) \text{ in } \mathcal{D}_{ss},$$

  F belongs to **F** and
  - For each disjunct $[a = A \wedge \phi]$ in $\gamma^+_F(a)$, $\mathcal{D}_\mathcal{A}$ contains

    A **causes** F **if** ϕ.
  - for each disjunct $[a = A \wedge \phi]$ in $\gamma^-_F(a)$, $\mathcal{D}_\mathcal{A}$ contains

    A **causes** $\neg F$ **if** ϕ
- Let $\Delta_\mathcal{A} = \{\delta_0 \mid \delta_0$ is a minimal set of fluent literals satisfying $\mathcal{D}_{S_0}\}$.

It is easy to check that the action theory (D_1, Δ_1) (Example 3) is the result of the above translation from the theory $\mathcal{D}^b$ in Example 1. Similarly to (Kartha 1993), we can prove:

Theorem 3. *For every fluent formula* φ *and action sequence* $\alpha = [a_1, \dots, a_n]$ *in* $\mathcal{D}$*,* $\mathcal{D} \models Poss(\alpha, S_0) \wedge \varphi(Do(\alpha, S_0))$ *iff* $(\mathcal{D}_\mathcal{A}, \Delta_\mathcal{A}) \models^P \varphi$ **after** α.

The next theorem relates the context-complete condition on $\mathcal{D}$ and the reducibility condition on $(\mathcal{D}_\mathcal{A}, \Delta_\mathcal{A})$.

Theorem 4. *Let* $\mathcal{D}$ *be a context-complete situation calculus theory and* $(\mathcal{D}_\mathcal{A}, \Delta_\mathcal{A})$ *be its translation to* $\mathcal{A}$*. Then* $ext(\Delta_\mathcal{A}) \gg_\varphi \Delta_\mathcal{A}$ *holds for each fluent formula* φ*.*

Observe that, in order to obtain the same form of action theory described earlier, it is necessary to convert the formulae in the conditions of the dynamic causal laws to disjunctive normal form and distribute the disjuncts in separate laws. We are working on extending the 0-approximation to allow the use of arbitrary propositional formulae in the dynamic causal laws.

Conclusions

In this paper, we studied the relationship between two completeness conditions for the 0-approximation. The insights gained through the study allowed us to develop a simplification procedure of an $\mathcal{A}$ action theory to an equivalent theory whose completeness condition can be weakened for certain queries. We also showed that the context-complete condition on local effect action theories proposed in (Liu & Levesque 2005) implies the reducibility condition for action theories in the language $\mathcal{A}$ developed in (Son & Tu 2006).

References

Balduccini, M., and Gelfond M. 2003. Diagnostic Reasoning with A-Prolog. *TPLP*, 3(4,5):425–461.

Baral, C., and Gelfond M. 2000. Reasoning agents in dynamic domains. *LBAI*:257–279.

Baral, C.; et al. 2000. Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence* 122:241–267.

Gelfond, M., and Lifschitz, V. 1993. Representing actions and change by logic programs. *JLP* 17(2,3,4):301–323.

Kartha, G. 1993. Soundness and completeness theorems for three formalizations of action. In *IJCAI*, 724–729.

Levesque, H. 1998. A completeness result for reasoning with incomplete first-order knowledge bases. In *KR*.

Liu, Y., and Levesque, H. 2005. Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In IJCAI.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence*, volume 4. Edinburgh University Press. 463–502.

McCarthy, J. 1959. Programs with common sense. In *Teddington Conference on the Mechanization of Thought Processes*, 75–91. Her Majesty's Stationery Office.

McDermott, D. 1987. A critique of pure reason. *Computational Intelligence* 3:151–160.

Moore, R. 1985. A formal theory of knowledge and action. In Hobbs, J., and Moore, R., eds., *Formal theories of the commonsense world*. Ablex, Norwood, NJ.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In *AI and Mathematical Theory of Computation*. Academic Press. 359–380.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press.

Son, T., and Baral, C. 2001. Formalizing sensing actions - a transition function based approach. *AIJ* 125(1-2):19–91.

Son, T. C., and Tu, P. H. 2006. On the Completeness of Approximation Based Reasoning and Planning in Action Theories with Incomplete Information. In *KRR*, 481–491.

Tu, P. H. 2007. *Reasoning and Planning with Incomplete Information in the Presence of Static Causal Laws*. Dissertation, New Mexico State University.