*Appears in* Proceedings of 8th Int'l Conference on Principles of Knowledge Representation and Reasoning (KR '2002).

# Projection in Decomposed Situation Calculus

**Eyal Amir**
Computer Science Division
University of California at Berkeley
Soda Hall 387, Berkeley, CA 94720-1776, USA
eyal@cs.berkeley.edu

## Abstract

We investigate the impact of decomposition on projection in the situation calculus. We show that performing projection with situation calculus theories can benefit from their decomposition into parts associated with sub-domains. Particularly, we provide message-passing algorithms that take advantage of the particular structure of situation calculus theories to perform the task of projection. These algorithms are shown to be sound and complete for this task for different scenarios, including actions with non-deterministic effects, partially specified initial situation and observations in situations later than the first one. They can be used for distributed reasoning about situation calculus theories or to speed up computation, in those cases where they are efficient. We characterize the kind of messages that must be sent between partitions for each of our algorithms and scenarios. This allows us to provide computational complexity results for the proposed algorithms under some assumptions. Our results are important for analyzing and devising planning, diagnosis and control algorithms for large domains that are made of interacting parts.

## 1 Introduction

Situation calculus (McCarthy and Hayes, 1969) is one of the leading logical formalizations for the representation of actions and change. It is used to specify high-level programs for robots (e.g., (Levesque et al., 1997)) and to do projection (e.g., (Reiter, 1992)), planning (e.g., (Green, 1969; Finzi et al., 2000)) and diagnosis (e.g., (McIlraith, 1997)) in dynamic systems. It is particularly useful for these tasks because its language is highly expressive and many extensions can be represented within it with relative ease (see (Reiter, 2001)). It also allows us to formally examine many other algorithms that reason about dynamic systems, analyze them and generalize them (e.g., (Lin and Reiter, 1995; Santibaez, 1999)). In general, reasoning about dynamic systems is computationally expensive (e.g., (Bylander, 1994; Baral et al., 2000)), whether done using situation calculus or otherwise. However, in the last 15 years some approaches that advocate decomposition of problems have been developed with some success (e.g., (Pearl, 1988; Dechter and Pearl, 1989; Darwiche, 1998; Amir and McIlraith, 2000; Pfeffer, 2001)).

In this paper we investigate the applicability of decomposition to projection in the situation calculus, projection being the prototypical reasoning problem for dynamic systems. We propose reasoning procedures for situation calculus theories that are composed of interacting sub-domains (we refer to each situation calculus theory that is associated with these sub-domains as a *partition*). Our algorithms use local computation for each partition and send messages of restricted form and length between the partitions. We prove the soundness and completeness of our algorithms, and provide computational analysis of these algorithms under different assumptions. Particularly, we provide a theorem that relates the structure of a given sequence of actions with the computational cost of projection of this sequence with our algorithms.

Our characterization of the messages that must be sent between partitions is important because it allows us to develop specialized reasoning procedures that look for these formulae directly, resulting in an efficient way to perform projection and planning in problems that are composed of related parts. Our theorems are applicable to a wide range of situation calculus theories, such as theories with non-deterministic actions, observations, knowledge-producing actions and other extensions developed for simple situation calculus theories (see (Reiter, 2001)). Finally, the results here are immediately applicable to the object-oriented situation calculus theories proposed in (Amir, 2000).

(Amir and McIlraith, 2000) provided message-passing algorithms for reasoning with logical theories that are made of interacting subtheories in the style of (Pearl, 1988). These algorithms are applicable in our setup, but for general First-Order Logic (FOL) theories they may end up producing arbitrarily many messages that may be very large. Some of our algorithms here can be seen as a restriction on the messages that these algorithms can be send between partitions.

Some proofs are omitted here for lack of space. They appear in (Amir, 2002b).

## 2 Background: Decomposed Situation Calculus

### 2.1 Situation Calculus

We review situation calculus briefly, and the reader is referred to (Reiter, 2001) for further background and details.

*Situation calculus* (McCarthy and Hayes, 1969) is a logical formalism for representing temporal information. The language consists of four sorts: *situations*, for situations in the world; *actions*, for events and actions; *fluents*, for situation-dependent properties; and *objects*, for simple other FOL objects.

Unless otherwise mentioned, $s$, $a$, $f$ (or subscripted versions thereof) are variables for situations, actions and fluents, respectively. All other variables are of sort "object". The predicate $Holds(f, s)$ asserts that a fluent $f$ holds in the situation $s$ ($f(s)$ is used as a shorthand for $Holds(f, s)$ when such a shorthand causes no confusion). The function $result(a, s)$ returns the situation that results from performing $a$ in $s$ (we use $res(a, s)$ as a shorthand). $S0$ is a situation constant. Figure 1 displays $A_{bw}$, a sample situation calculus theory for the blocks-world domain. Capitalized symbols are constants. Free variables in axioms are universally quantified with maximum scope.

$$
\begin{array}{l}
on(A, B, S0) \wedge on(B, Table, S0) \wedge on(C, Table, S0) \\
\\
on(x, y, s) \Rightarrow above(x, y, s) \\
above(x, y, s) \wedge above(y, z, s) \Rightarrow above(x, z, s) \\
clear(b, s) \iff \forall b' \neg on(b', b, s) \\
handEmpty(s) \iff \forall b \neg inHand(b, s) \\
on(x, y, s) \Rightarrow \neg inHand(x, s) \wedge \neg inHand(y, s) \\
inHand(x, s) \Rightarrow \neg on(x, y, s) \wedge \neg on(y, x, s) \\
on(x, y, s) \wedge y \neq z \Rightarrow \neg on(x, z, s) \\
\\
clear(b, s) \wedge handEmpty(s) \Rightarrow inHand(b, res(pickUp(b), s))) \\
(y = Table \vee clear(y, s)) \wedge inHand(x, s) \Rightarrow \\
\qquad\qquad on(x, y, res(putOn(x, y), s))
\end{array}
$$

Figure 1: Blocks-world in the situation calculus: $A_{bw}$.

In situation calculus theories, *effect axioms* are used to derive the consequences of actions. In Figure 1, effect axioms are given for the actions $putOn(x, y)$, $pickUp(b)$. *state constraints* are sentences that mention no action term. This ontology received a suitable set of foundational axioms, regarding its structure of time in (Lin and Reiter, 1994; Pinto, 1994) and others.

The *frame problem* concerns the conclusion of non-effects of actions from the known effects in a concise, correct, expressive and elaboration-tolerant manner (see (Shanahan, 1997)). One of the solutions generates *explanation closure* axioms (e.g., (Haas, 1987; Pednault, 1989; Schubert, 1990)) from the effect axioms and the domain constraints. When it is possible, we mechanically (outside the logic) add axioms saying that if something has changed, then one of the enumerated actions occurred and their proper preconditions held. For example, one axiom that is generated for explanation closure for $inHand$ in $A_{bw}$ is $\neg inHand(x, s) \wedge inHand(x, res(a, s)) \Rightarrow (clear(x, s) \wedge handEmpty(s) \wedge a = pickUp(x))$.

(Reiter, 1991) summarized the effort and showed how to generate such axioms automatically if there are no state constraints, (Lin and Reiter, 1994) extended this process for the presence of state constraints, using deduction, and (McIlraith, 2000) gave a closed-form solution in the presence of some restricted state constraints. These solutions also add other axiom sets, including unique names axioms (UNA) for sort "actions", preconditions for executing actions (summarized by the predicate $Poss(a, s)$) and foundational axioms for situations.

All of the results in the rest of this paper are stated for situation calculus theories in which the frame problem is solved using Reiter's solution (see (Reiter, 2001)) prior to reasoning. We assume that there are no state constraints or that whatever state constraints there are were compiled into that solution by the methods of (Lin and Reiter, 1994; Pinto, 1999; McIlraith, 2000).

### 2.2 Combining Action Theories

The way situation calculus can be joined using object-oriented design tools was examined in (Amir, 2000). It showed that domain theories that are represented using situation calculus can be joined without the need for significant recomputation of the solution to the frame problem. We use the results mentioned in (Amir, 2000) while avoiding the use of object-oriented notation, thus sidestepping unnecessary definitions.

Consider a domain theory regarding buying and selling items. In this theory, buying an item decreases the amount of money a robot has, but also places the purchased item in the robot's hand. Selling a block has the opposite effect.

Figure 2 presents the new axioms. The blocks-world theory and the buy-sell theory share only $S0, res, Holds$ and $inHand$. Figure 3 is a diagram of the situation.

$money(S0) = 10$

$money(s) \geq 0$
$inHand(b, s) \Rightarrow HasItem(b, s)$

$inHand(b, s) \wedge value(b, s) = v \wedge money(s) = m \Rightarrow$
$\neg hasItem(b, res(sell(b), s)) \wedge money(res(sell(b), s)) = m+v$
$value(b, s) = v \wedge money(s) = m \wedge m \geq v \Rightarrow$
$money(res(buy(b), s)) = m-v \wedge inHand(b, res(buy(b), s))$

Figure 2: The domain of buying and selling items, $\mathcal{A}_{bs}$.

Following (Amir and McIlraith, 2000), we say that $\{\mathcal{A}_i\}_{i \leq n}$ is a *partitioning* of a logical theory $\mathcal{A}$ if $\mathcal{A} = \bigcup_i \mathcal{A}_i$. Each individual $\mathcal{A}_i$ is a set of axioms called a *partition*, $L(\mathcal{A}_i)$ is its signature (the set of non-logical symbols), and $\mathcal{L}(\mathcal{A}_i)$ is its language (the set of formulae built with $L(\mathcal{A}_i)$). The partitions may share literals and axioms.

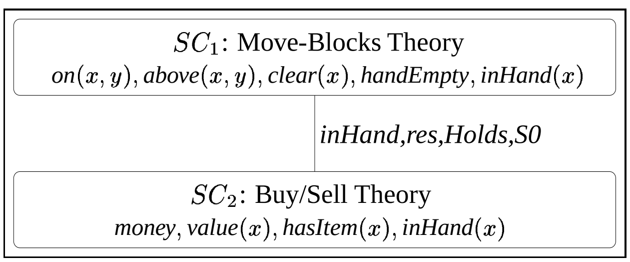


Figure 3: Combining blocks world with purchase and sale.

We call the graph in Figure 3 the *intersection graph* of the partitioned theory $\mathcal{A}_{bw} \cup \mathcal{A}_{bs}$. More generally, every partitioning of a theory induces a graphical representation, $G = (V, E, l)$, which we call the partitioning's *intersection graph*. Each node of the intersection graph, $i$, represents an individual partition, $\mathcal{A}_i$, ($V = \{1, ..., n\}$), two nodes $i, j$ are linked by an edge if $\mathcal{L}(\mathcal{A}_i)$ and $\mathcal{L}(\mathcal{A}_j)$ have a non-logical symbol in common ($E = \{(i, j) \mid L(\mathcal{A}_i) \cap L(\mathcal{A}_j) \neq \emptyset\}$), and the edges are labeled with the set of symbols that the associated partitions share ($l(i, j) = L(\mathcal{A}_i) \cap L(\mathcal{A}_j)$). We refer to $l(i, j)$ as the *communication language* between partitions $\mathcal{A}_i$ and $\mathcal{A}_j$. We ensure that the intersection graph is connected by adding a minimal number of edges to $E$ with empty labels, $l(i, j) = \emptyset$.

(Amir, 2000) showed that adding the explanation-closure solution to the frame problem can be done in a way that ensures that the graph does not become more connected. Situation calculus theories that are made of such connected subtheories (two subtheories or more) are called *oo-sitcalc theories*. They can be built from component theories or be the result of manual or automatic decomposition (Amir and McIlraith, 2000; Amir, 2001).

### 2.3 Message-Passing

Figure 4 displays MESSAGE-PASSING (MP), a message-passing algorithm proposed in (Amir and McIlraith, 2000) for partition-based logical reasoning. It takes as input a partitioned theory, $\mathcal{A}$, an associated graph structure $G = (V, E, l)$, and a query formula $Q$ in $\mathcal{L}(\mathcal{A}_k)$, and returns YES if the query was entailed by $\mathcal{A}$. The algorithm uses procedures that generate consequences (consequence finders) as the local reasoning mechanism within each partition or graphical node. It passes a concluded formula to an adjacent node if the formula's signature is in the communication language $l$ of the adjacent node, and that node is on the path to the node containing the query.

The messages in this algorithm are sent in a single direction, the direction of the *goal partition*. To determine the direction in which messages should be sent in the graph $G$, step 1 in MP computes a strict partial order over nodes in the graph using the partitioning together with a query, $Q$.

**Definition 2.1 ($\prec$)** *Given partitioned theory $\mathcal{A} = \bigcup_{i \leq n} \mathcal{A}_i$, associated graph $G = (V, E, l)$ and query $Q \in \mathcal{L}(\mathcal{A}_k)$, let $dist(i, j)$ ($i, j \in V$) be the length of the shortest path between nodes $i, j$ in $G$. Then $i \prec j$ iff $dist(i, k) < dist(j, k)$.*

PROCEDURE MESSAGE-PASSING (MP)($\{\mathcal{A}_i\}_{i \leq n}, G, Q$)
$\{\mathcal{A}_i\}_{i \leq n}$ a partitioning of the theory $\mathcal{A}$, $G = (V, E, l)$ a graph describing the connections between the partitions, $Q$ a query in $\mathcal{L}(\mathcal{A}_k)$ ($k \leq n$).

1. Determine $\prec$ as in Definition 2.1.
2. Concurrently,
   (a) Perform consequence finding for each of the partitions $\mathcal{A}_i$, $i \leq n$.
   (b) For every $(i, j) \in E$ such that $i \prec j$, for every consequence $\varphi$ of $\mathcal{A}_j$ found (or $\varphi$ in $\mathcal{A}_j$), if $\varphi \in \mathcal{L}(l(i, j))$, then add $\varphi$ to the set of axioms of $\mathcal{A}_i$.
   (c) If $Q$ is proven[a] in $\mathcal{A}_k$, return YES.

[a]Derive a subsuming formula or initially add $\neg Q$ to $\mathcal{A}_k$ and derive inconsistency.

Figure 4: A forward message-passing algorithm (McIlraith and Amir, 2001).

It was shown in (Amir and McIlraith, 2000; McIlraith and Amir, 2001) that this kind of algorithm is complete and sound for partitioned theories, and that it sometimes improves running time compared to standard deduction methods.

Completeness of the algorithm assumes that the graph $G$ used in the algorithm is *properly labeled* (in other fields this is sometimes called *satisfies the running intersection property*). The condition of proper labeling for a graph $G(V, E, l)$ was defined in (McIlraith and Amir, 2001) to say that $G$ is a tree of partitions such that for all $(i, j) \in E$ and $\mathcal{B}_1, \mathcal{B}_2$, the two subtheories of $\mathcal{A}$ on the two sides of the edge $(i, j)$ in $G$, it is true that $l(i, j) \supseteq L(\mathcal{B}_1) \cap L(\mathcal{B}_2)$. This condition always holds if the intersection graph of a partitioned theory is a tree, and there are algorithms for converting any graph into such a tree, e.g., BREAK-CYCLES($G$) in (Amir and McIlraith, 2000). The *width* of a properly-labeled tree is the size of the largest partition (size here includes the number of fluents in the partition and on the labels of the partition's edges). The *treewidth* of $SC$ is the lowest width among all properly-labeled trees for $SC$.

MP is immediately applicable to both projection and planning in situation calculus if we partition those theories as in the previous section. However, the number of messages sent and their lengths can be too large. The rest of the paper characterizes the messages that must be sent in similar algorithms and in bi-directional MP algorithms, showing that we can restrict the messages significantly for the task of projection.

## 3 Projection Using Bi-Directional Passing of Restricted Messages

In this section we propose two message-passing algorithms that perform projection in situation calculus theories by sending only messages of a restricted-form. Both algorithms perform projection by sending messages back-and-forth between partitions. They differ in the restrictions they put on the messages and in the range of theories for which they are applicable.

In the following we assume that each observation that we have (about $S0$ or otherwise, if allowed in our scenario) can be represented using the vocabulary that we associate with a single partition (or can be compiled (e.g., using the methods of (McIlraith and Amir, 2001)) into equivalent multiple observations that satisfy this condition).

Also, for simplicity we assume that fluents are propositional (i.e., there are no function fluents or predicate fluents, but rather fluents are named and no function generates new fluents from others).

### 3.1 Deterministic Situation Calculus Theories

Our first algorithm sends messages of the form $Holds(f, S_j)$ between partitions, for $f$ and $S_j$ being ground terms of sort fluent and situation, respectively. This algorithm is applicable to situation calculus theories that have a fully specified initial situation and deterministic actions. The procedure is present in Figure 5.

PROCEDURE MP-2dir-det-SitCalc (MP2d)($\{SC_i\}_{i \leq n}$, $G$, $Q$)

$\{SC_i\}_{i \leq n}$ an oo-sitcalc theory, $G = (V, E, l)$ a graph describing the connections between the partitions, $Q$ a query in $\mathcal{L}(SC_k)$ ($k \leq n$).

1. Concurrently,
   (a) Perform consequence finding for each of the partitions $SC_i$, $i \leq n$.
   (b) For every $(i, j) \in E$, for every consequence $Holds(\varphi, S)$ of $SC_j$ found (or $Holds(\varphi, S)$ in $SC_j$), if $\varphi \in \mathcal{L}(l(i, j))$ is a fluent term and $S$ a ground situation term, then add $Holds(\varphi, S)$ to the set of axioms of $SC_i$.
   (c) If $Q$ is proven[a] in $SC_k$, return YES.

[a]Derive a subsuming formula or initially add $\neg Q$ to $SC_k$ and derive inconsistency.

Figure 5: A bi-directional message-passing algorithm for deterministic scenarios.

**Theorem 3.1 (Soundness and Completeness of MP2d)** *Let $SC = \{SC_i\}_{i \leq n}$ be an oo-sitcalc theory, $a_1, \ldots, a_m$ actions in $L(SC)$ and for all $j \leq m$ $S_j = result(a_j, res(a_{j-1}, ..., res(a_1, S0)))$. Let $\varphi \in \mathcal{L}(SC_k)$ be a fluent term, for $k \leq n$. Then $SC \models Holds(\varphi, S_j)$ iff MP2d outputs YES for the query $Holds(\varphi, S_j)$ stated in partition $SC_k$.*

Most of the rest of this section is devoted to proving this theorem. The proof outline is as follows: We *unroll* the $result$ function for a partitioned situation calculus theory, $SC$, that has only two partitions, $SC_1$ and $SC_2$ (see Figure 6). We show that the resulting theory, $SC'$, is equivalent to the first one for a set of queries. We use this and the soundness and completeness of MP for arbitrary partitioned theories to find the messages that can be sent in MP running on $SC'$. These messages are then translated back to the original system, resulting in the desired characterization. Finally, the generalization of this characterization to the case of a tree of partitions (instead of only two partitions) is done by induction.

**Definition 3.2 (Unrolling *result*)** *Let $SC = \{SC_1, SC_2\}$ be an oo-sitcalc theory with two partitions (e.g., see Figure 3). We say that $SC'$* unrolls $SC$ for the action sequence *$a_1, \ldots, a_n$ if $SC' = \{SC^i\}_{1 \leq i \leq n}$ and for every $i \leq n$ $SC^i$ is a copy of $SC_1$ (if $a_i$ resides[1] in $SC_1$) or $SC_2$ (if $a_i$ resides in $SC_2$) that we change in three ways: First, we*

[1]We say that $a_i$ *resides* in $SC_j$ if the effect axioms of $a_i$ are in $SC_j$.

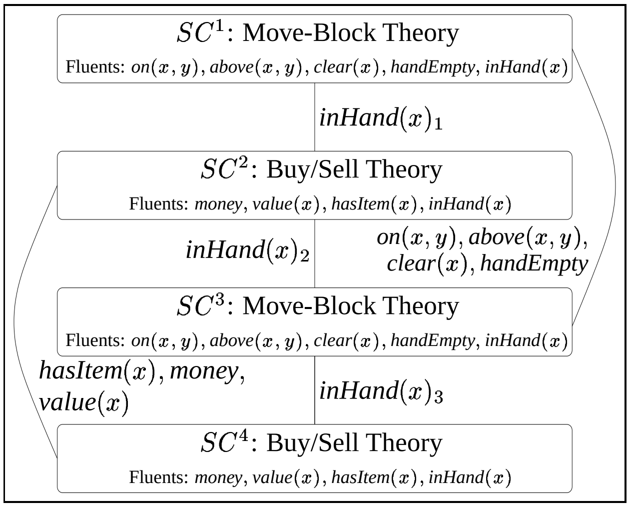


Figure 6: High-level diagrammatic view of unrolling *result*.

*replace every nonlogical symbol $X$ that is not a fluent name by a new symbol $SC^i.X$. Then, we add new propositional symbols $f_i$ for every fluent name $f \in L(SC^i)$. Finally, for every fluent name $f \in L(SC^{i-1}) \cap L(SC^i)$ we add the axioms*

$$f_{i-1} \iff SC^i.Holds(f, SC^{i-1}.res(SC^i.a_{i-1}, ..., res(SC^i.a_1, SC^i.S0)...))$$
$$f_i \iff SC^i.Holds(f, SC^i.res(SC^i.a_i, ..., res(SC^i.a_1, SC^i.S0)...))$$

*and for every $f \in L(SC_i) \setminus L(SC_{i-1})$ we add the axioms*

$$f_j \iff f_{j+1} \ \ ..., \ \ f_{i-2} \iff f_{i-1}$$

*if $j < i$ such that $SC^i, SC^j$ are copies of the same partition ($SC_1$ or $SC_2$) and $j$ is the largest such index.*

Figure 6 is an example unrolling of the theory $SC = \{\mathcal{A}_{bw}, \mathcal{A}_{bs}\}$ for four actions. The intuition behind this unrolling is that we try to keep the partitions $SC_1 = \mathcal{A}_{bw}, SC_2 = \mathcal{A}_{bs}$ as separate as possible. Every partition of axioms in Figure 6 corresponds to a situation, action and the action's result in that situation. It is important that every partition $SC^i$ does not include the entire state, nor does it need to reason about the absent fluents.

The definition roughly suggests that partition $SC^i$ receives information from the previous partition and one more partition, if its immediate predecessor is not a copy of the same original partition of $SC$. After applying the action $a_i$ in $SC^i$ we send some of the fluents values in the resulting situation to $SC^{i+1}$ and the rest of the fluents of $L(SC^i)$ to the next partition down the chain that needs them (a partition that is a copy of the same original partition of $SC$ as $SC^i$).

We show that this *unrolling* is equivalent to the original structure for queries of the form $Holds(\varphi, res(a_n, res(\ldots, res(a_1, S0))))$.

**Lemma 3.3 (Equivalence of Unrolling)** *Let $SC = \{SC_1, SC_2\}$ be an oo-sitcalc theory (as in Figure 3), and let $SC'$ be the unrolling of $SC$ for $a_1, \ldots, a_n$. Let $S_i = result(a_i, res(a_{i-1}, ..., res(a_1, S0)))$ for all $i \le n$, and let $\varphi \in \mathcal{L}(SC^n)$ a fluent term. Then,*

$$SC \models Holds(\varphi, S_n) \iff SC' \models SC^n.Holds(\varphi, res(SC^n.a_n, SC^n.S0)).$$

PROOF See Appendix A.1. ■

For the following lemma we assume full knowledge of the state $S0$ and actions that are deterministic.

**Lemma 3.4 (MP in Unrolled Theory)** *Let $SC = \{SC_1, SC_2\}$ be an oo-sitcalc theory, $a_1, \ldots, a_n$ actions in $L(SC)$ and $S_i = res(a_i, res(a_{i-1}, ..., res(a_1, S0)))$ for all $i \le n$. Let $\varphi \in \mathcal{L}(SC_j)$ be a fluent term, for $j \in \{1, 2\}$. Then $SC \models Holds(\varphi, S_n)$ iff MP outputs YES for the query[2] $SC^n.Holds(\varphi, SC^n.S_n)$ stated in partition $SC^n$.*

It is important to notice that we do not convert the graph $G$ into a tree before running MP. The fact that this lemma holds even without running BREAK-CYCLES on $G$ is a result of our assumption about deterministic actions and full knowledge of the fluents in $S0$.

Now we can show that the only messages that need to be sent in a back-and-forth Message-Passing are of the form of a single state observation or constraint. For the following theorem we assume full knowledge of the state $S0$ and actions that are deterministic.

**Theorem 3.5 (MP2d is Complete & Sound for 2 Parts.)** *Let $SC = \{SC_1, SC_2\}$ be an oo-sitcalc theory (as in Figure 3), $a_1, \ldots, a_n$ actions in $L(SC)$ and $S_i = result(a_i, res(a_{i-1}, ..., res(a_1, S0)))$ for all $i \le n$. Let $\varphi \in \mathcal{L}(SC_j)$ a fluent term, for $j \in \{1, 2\}$. Then $SC \models Holds(\varphi, S_n)$ iff MP2d outputs YES for the query $Holds(\varphi, S_n)$ stated in partition $SC_j$.*

PROOF SKETCH From Lemma 3.3 we know that $SC \models Holds(\varphi, S_G)$ iff $SC' \models SC^n.Holds(\varphi, SC^n.S_n))$. Furthermore, from Lemma 3.4 we know that the messages that need to be passed between partitions that are copies of $SC^1$ and those that are copies of $SC^2$ are of a form equivalent to $Holds(\psi, S_i)$. ■

The generalization of the last theorem to the tree case is done by induction, leading to the conclusion of the proof of Theorem 3.1. ■

[2]We abuse notations here and write $SC^n.S_n$ for $SC^n.res(SC^n.a_n, ..., SC^n.res(SC^n.a_1, SC^n.S0)...)$.

For this special case (deterministic actions and fully known initial state) we can show that MP2d is complete even if $G$ is not a tree. This follows from a similar argument that we do not bring here.

### 3.2 Non-Deterministic Situation Calculus Theories

It is important to notice that Theorem 3.1 does not hold if $S0$ is not fully specified or some actions have non-deterministic effects. To see this, assume that we know that $\mathit{Holds}(\mathit{on}(A,B) \vee \mathit{inHand}(A), S0)$. This implies that

$$\mathit{Holds}(\mathit{on}(A,B) \vee \mathit{inHand}(D), \mathit{res}(\mathit{buy}(D), \mathit{res}(\mathit{sell}(A), S0)))$$

is a valid consequence of $\mathcal{A}_{bw}$. However, we cannot prove this using messages of the form $\mathit{Holds}(\psi, S_i)$ when $\psi$ is a fluent term based on *inHand* because there is no conclusion that we can draw about $\mathit{inHand}(A)$ or $\mathit{inHand}(D)$ or any relationship between them in any single situation. We can conclude

$$\mathit{Holds}(\mathit{inHand}(A), S0) \Rightarrow \mathit{Holds}(\mathit{inHand}(D), \mathit{res}(\mathit{buy}(D), \mathit{res}(\mathit{sell}(A), S0))),$$

but this is a formula that includes two different situations.

Nonetheless, a generalization of Theorem 3.1 holds for the case of nondeterministic actions, partially-specified initial state and possible observations about states later than $S0$. By nondeterministic actions we refer to actions whose effect axioms specify results that are not a single conjunction of fluents. The specification of nondeterministic actions and the solution to the frame problem in such settings take different semantics in different works (Lin, 1996; Levesque et al., 1997). Here, we assume that the solution is given using some added first-order axioms, but we do not assume any particular information about them. Similar results can be achieved for the GOLOG model of nondeterministic selection of actions (Levesque et al., 1997).

Our algorithm performs projection by sending messages of the form $Holds(\psi_1, S_{i_1}) \wedge \ldots \wedge Holds(\psi_a, S_{i_a}) \Rightarrow Holds(\psi_{a+1}, S_{i_{a+1}})$ between partitions, for $\psi_j$ being ground terms of sort fluent and $S_{i_j}$ being ground terms of sort situation. It is more generally applicable than the first algorithm, allowing theories that include nondeterministic effects of actions, partially specified first situation and observations about later situations. The procedure is present in Figure 7.

**Theorem 3.6 (Soundness and Completeness of MP2n)** *Let $SC = \{SC_i\}_{i\leq n}$ be an oo-sitcalc theory, $a_1, \ldots, a_m$ actions in $\bar{L}(SC)$ and for all $j \leq m$ $S_j = result(a_j, res(a_{j-1}, ..., res(a_1, S0)))$. Let $\varphi \in \mathcal{L}(SC_k)$ be a fluent term, for $k \leq n$. Then $SC \models Holds(\varphi, S_j)$ iff MP2n outputs YES for the query $Holds(\varphi, S_j)$ stated in partition $SC_k$.*

PROCEDURE MP2-nondet-SitCalc (MP2n)($\{SC_i\}_{i\leq n}, G, Q$)

$\{SC_i\}_{i\leq n}$ an oo-sitcalc theory, $G = (V, E, l)$ a graph describing the connections between the partitions, $Q$ a query in $\mathcal{L}(SC_k)$ ($k \leq n$).

1. Concurrently,
   (a) Perform consequence finding for each of the partitions $SC_i$, $i \leq n$.
   (b) For every $(i,j) \in E$, for every consequence of the form $\Psi = Holds(\psi_1, S_{i_1}) \wedge \ldots \wedge Holds(\psi_a, S_{i_a}) \Rightarrow Holds(\psi_{a+1}, S_{i_{a+1}})$ of $SC_j$ found (or $\Psi \in SC_j$), if $\Psi \in \mathcal{L}(l(i,j))$, then add $\Psi$ to the set of axioms of $SC_i$.
   (c) If $Q$ is proven[a] in $SC_k$, return YES.

[a] Derive a subsuming formula or initially add $\neg Q$ to $SC_k$ and derive inconsistency.

Figure 7: A bi-directional message-passing algorithm for nondeterministic scenarios.

The proof of this theorem follows from Lemma 3.3 and the soundness and completeness of MP for properly labeled graphs (McIlraith and Amir, 2001). The proof follows if we notice that running MP on an oo-sitcalc theory as in Figure 6 after running BREAK-CYCLES (Amir and McIlraith, 2000) results in messages that can be converted to the proper form for our theorem.

## 4 Projection Using Messages-Passing in One Direction

The previous section characterized those messages that must be sent in a back-and-forth message passing proof over an oo-sitcalc theory. In this section we examine the case of sending messages in only one direction, the way MP does. We find that the messages that must be sent are of a similar form to that given for nondeterministic domains in the previous section: $Holds(\psi_1, S_{i_1}) \wedge \ldots \wedge Holds(\psi_a, S_{i_a}) \Rightarrow Holds(\psi_{a+1}, S_{i_{a+1}})$. Our algorithm is shown in Figure 8.

Surprisingly, we must send the same kind of messages even if we limit our attention to deterministic domains, i.e., formulae of the form $Holds(f, S)$ or $Holds(\psi, S) \Rightarrow Holds(\psi', S')$ are not enough. Roughly speaking, the reason for this is that when we send messages in a single direction each partition has to send enough messages to account for all the possible situations to which it might have been applied if we used bi-directional message-passing.

In the following theorem we allow partially specified initial situation, nondeterministic effects of actions and observations in situations later than $S0$.

PROCEDURE MP-nondet-SitCalc(MPn)($\{SC_i\}_{i\leq n}$, $G$, $Q$)
$\{SC_i\}_{i\leq n}$ an oo-sitcalc theory, $G = (V, E, l)$ a graph describing the connections between the partitions, $Q$ a query in $\mathcal{L}(SC_k)$ ($k \leq n$).

1. Determine $\prec$ as in Definition 2.1.
2. Concurrently,
   (a) Perform consequence finding for each of the partitions $SC_i$, $i \leq n$.
   (b) For every $(i, j) \in E$ such that $i \prec j$, for every consequence of the form $\Psi = Holds(\psi_1, S_{i_1}) \wedge \ldots \wedge Holds(\psi_a, S_{i_a}) \Rightarrow Holds(\psi_{a+1}, S_{i_{a+1}})$ of $SC_j$ found (or $\Psi \in SC_j$), if $\Psi \in \mathcal{L}(l(i, j))$, then add $\Psi$ to $SC_i$'s axioms set.
   (c) If $Q$ is proven[a] in $SC_k$, return YES.

[a]Derive a subsuming formula or initially add $\neg Q$ to $SC_k$ and derive inconsistency.

Figure 8: A forward message-passing algorithm for projection in deterministic and nondeterministic domains.

**Theorem 4.1 (Soundness and Completeness of MPn)** *Let $SC = \{SC_i\}_{i\leq n}$ be an oo-sitcalc theory, $a_1, \ldots, a_m$ actions in $L(SC)$ and for all $j \leq m$ $S_j = result(a_j, res(a_{j-1}, ..., res(a_1, S0)))$. Let $\varphi \in \mathcal{L}(SC_k)$ a fluent term, for $k \leq n$. Then $SC \models Holds(\varphi, S_j)$ iff MPn outputs YES for the query $Holds(\varphi, S_j)$ stated in partition $SC_k$.*

## 5 Structure and Computational Analysis

In situations where computation is distributed over several computers or agents, our procedures and their alike are the only mean for performing inference and projection. However, in cases where distribution is not necessary, there are simple methods that may outperform our procedures significantly, especially if we allow long messages to be sent. The computation offered by the procedures above can be very costly if the length of messages is not bounded.

In those cases, when is it better to use our procedures? In general, can we estimate the amount of computation involved given a sequence of actions and a set of partitions? In this section we characterize those cases and yield a few insights into the related problem of planning. Our analysis is based on the observation that the size of the messages can be kept small if there is only little back and forth transition between partitions in a given sequence of actions.

We make this intuition precise using the following definition.

**Definition 5.1** *Let $SC = \{SC_i\}_{i\leq n}$ be an oo-sitcalc theory, $a_1, \ldots, a_m$ actions in $L(SC)$ and $G(V, E, l)$ a graph describing the connections between partitions. For $(i, j) \in E$, we say that $\langle a_{l_1}, a_{l_2}, \ldots, a_{l_k}\rangle$ ($\forall x < k$ $1 \leq l_x < l_{x+1} \leq m$) is an* influencing sequence *for $SC_i$ from $SC_j$ if (A) there are $t_1, t_2 \leq m$ such that $l_k < t_1 \leq t_2$, $a_{t_1}, a_{t_2} \in \mathcal{L}(SC_i)$ and $a_{t_1}$ has preconditions in $\mathcal{L}(l(i, j))$ and $a_{t_2}$ has effects in $\mathcal{L}(l(i, j))$; and (B) for every $x \leq k$,*

1. *$a_{l_x} \in \mathcal{L}(SC_j)$,*
2. *$a_{l_x}$ has effects in $\mathcal{L}(l(i, j))$, and*
3. *if $x < k$, then there is $l' \leq m$ such that $l_x < l' < l_{x+1}$, $a_{l'} \in \mathcal{L}(SC_i)$ and $a_{l'}$ has preconditions in $\mathcal{L}(l(i, j))$.*

The intuition behind this definition is that if we have a sequence of actions such that some of them belong to $SC_i, SC_j$, then $SC_i$ needs to send $SC_j$ a message of the form

> if $S_1$ was to be the case after $a_{l_1}$, and $S_2$ was to be the case after $a_{l_2}$, and ... and $S_{l_k}$ was to be the case after $a_{l_k}$, then $S_{l_{k+1}}$ will be the case after $a_{l_{k+1}}$.

The length of the largest influencing sequence for $SC_i$ from $SC_j$ in a sequence of actions is exactly the maximal number of situations that must participate in a message that is sent from $SC_i$ to $SC_j$.

**Theorem 5.2** *Let $SC = \{SC_i\}_{i\leq n}$ be an oo-sitcalc theory, $a_1, \ldots, a_m$ actions in $L(SC)$ and $G(V, E, l)$ a tree describing the connections between partitions. Let $SC_i, SC_j$ be two partitions such that $(i, j) \in E$, and let $B_i, B_j$ be the two connected components of $G$ that result if we remove $(i, j)$ from $E$. In MP2n and in MPn, the only messages that we need to send from $SC_i$ to $SC_j$ are of the form*

$$Holds(\psi_1, S_{t_1}) \wedge \ldots \wedge Holds(\psi_l, S_{t_l}) \Rightarrow Holds(\psi_{l+1}, S_{t_{l+1}})$$

*such that $\langle a_{t_1}, ..., a_{t_l}\rangle$ is an influencing sequence for $B_i$ from $B_j$.*

Thus, if there is no influencing sequence in our sequence of actions (e.g., when the sequence of actions is such that the actions are grouped into the partitions, and the order between the action-groups follows $\prec$ from Definition 2.1), then the only messages that need to be sent are of the form $Holds(\varphi, S)$. If the only influencing sequences for $a_1, \ldots, a_m$ are of length 1, then the only messages that need to be sent are of the form $Holds(\varphi, S) \Rightarrow Holds(\varphi', S')$.

**Corollary 5.3** *Let $a_1, \ldots, a_m$ be a sequence of actions and let $SC = \bigcup_{i=1}^{n} SC_i$ be an oo-sitcalc theory with treewidth $k_1$ (treewidth is defined in Section 2.3). If the largest influencing sequence is of length $k_2$, then the projection with MPn takes time $O(2^{k_1 * k_2})$.*

This compares well with the coNP-completeness results for projection with partial information and nondeterministic actions (Baral et al., 2000; Amir, 2002a).

While doing projection we may sometimes take the given order of actions and change it to suit our purposes (not the actual execution, which is given, but the sequence of actions that we process in our algorithm). For example, we can try to group the actions so that they minimize $k_1 * k_2$ in Corollary 5.3 by looking at the dependencies between the actions, and rearranging the actions so that those dependencies are not altered or broken.

This result also suggests a new planning goal. When planning, try to find those plans that have the best *aggregation* of actions. Finding plans that have influencing sequences of length at most $k_2$, if such exist, takes time that is proportional to $2^{k_1 * k_2}$. This is subject of ongoing work.

## 6 Conclusions

This paper presented three novel algorithms for reasoning in partitioned domains using situation calculus. The algorithms were shown to be sound and complete for their respective classes of situation calculus theories. The first is a back-and-forth message-passing algorithm that needs to send only single-state formulae between partitions. The second is a back-and-forth message-passing algorithm that needs to send only effect-style formulae (two-state formulae) between partitions. The last algorithm is a single-direction message passing algorithm that needs to send $k$-state effect formulae between partitions. The first algorithm is applicable to situation calculus theories that have fully-specified initial states and deterministic effects of actions. The second and third are applicable to more general theories, including those that have only a partially specified initial state and nondeterministic effects of actions.

The results of this paper are important for several applications. First, in domains where reasoning is distributed among several machines or agents these algorithms allow us to perform projection. Also, algorithms for planning and diagnosis can be devised using our algorithms and theorems. Our results can serve as the basis for algorithms for reasoning about interacting agents and planning for collaborating agents. We can also analyze existing planning algorithms that are built around the idea of decomposition, such as (Lansky, 1988; Lansky and Getoor, 1995; Frank et al., 2000).

Also, our results can serve as a basis for reasoning algorithms for Markov Decision Processes (MDPs) and inference algorithms for some dynamic Bayes networks. They particularly shed some light on the applicability of algorithms for first-order MDPs (Boutilier et al., 2001). Finally, we are interested in building AI architectures that are based on networks of interacting interacting knowledge bases, and the algorithms and theorems offered in this paper are important for the understanding and scaling up of such architectures (see e.g., (Amir and Maynard-Reid II, 1999)).

## 7 Acknowledgments

I wish to thank Sheila McIlraith and Stuart Russell for discussions on subjects related to this paper. This research was supported by DARPA grant N66001-00-C-8018 (RKF program) and National Science Foundation grant ECS-9873474.

## A Proofs

### A.1 Lemma 3.3: Equivalence of Unrolling

**Backward Direction** The backward direction is seen by viewing the way MP works on $SC'$ (after applying BREAK-CYCLES to those *bypassing* edges in the graph). Let $S$ be a message-passing proof of $SC' \models SC^n.Holds(\varphi, SC^n.res(SC^n.a_n, SC^n.S0))$ represented as a sequence of formulae (this is the traditional Frege-Hilbert proof in which every formula is derived from previous ones in the sequence using a rule in the logic).

Every step in this proof is a possible in a regular proof in $SC$ because every deduction step in this proof can be made between the corresponding axioms in $SC$. The only axiom that we add in $SC'$ to the translated axioms of $SC$ are the equivalence axioms $f_{i-1} \iff f_j$ for fluents in far-apart partitions. However, this axiom is a translation of a valid consequence of $SC$: $Holds(f, S_{i-1}) \iff Holds(f, S_j)$.

To see that this is a valid consequence, assume, without loss of generality, that $SC^i$ is a translated copy of $SC_1$ (as in Definition 3.2). Then this formula follows from $SC$ because $f$ does not occur in $L(SC_2)$ and thus does not appear in $SC^{j+1}, ..., SC^{i-1}$. Because we do not have domain constraints (or they were compiled away) and we use explanation closure, this implies that the explanation closure axioms in $SC$ for the fluent $f$ show that $f$ does not change its value during the execution of actions $a_{j+1}, ..., a_{i-1}$. Thus, $SC \models Holds(f, S_{i-1}) \iff Holds(f, S_j)$. As a result, we can translate the proof in $SC'$ into a proof in $SC$, so

$$SC' \models SC^n.Holds(\varphi, SC^n.res(SC^n.a_n, SC^n.S0)) \\ \Rightarrow SC \models Holds(\varphi, S_G).$$

**Forward Direction** For the forward direction we show that for every model $\mathcal{M}'$ of $SC'$ there is a model $\mathcal{M}$ of $SC$ such that for every $i \leq n$, for every $\psi \in \mathcal{L}(SC^i)$ a fluent term, if $\mathcal{M}' \models$

$SC^i.Holds(\psi, SC^i.res(SC^i.a_i, ..., SC^i.res(SC^i.a_1, SC^i.S0)...))$, then $\mathcal{M} \models Holds(\psi, S_i)$.

**Only Effect Axioms** We show this for the case of situation calculus theories in which there are only effect axioms (no explanation closure) first. We then generalize this to the case including explanation closure, domain constraints and observations (in $S0$ and otherwise).

Assume otherwise. Let $i$ be the first index for which this assertion is not true. For $i = 0$ look at $SC^1$. It is isomorphic to $SC_j$ (for $j = 1$ or $j = 2$, whichever $SC^1$ is a copy of) under syntactic translation. Since there are only effect axioms in our theory, there are no domain constraints or observations regarding $S0$. Thus, there are no restrictions on $S0$ in $SC$ and there is a model $\mathcal{M}$ of $SC$ that has the fluent values specified by $\mathcal{M}'$ for fluents $L(SC^1)$. Thus, $\mathcal{M}' \models SC^i.Holds(\psi, SC^i.res(SC^i.a_i, ..., SC^i.res(SC^i.a_1, SC^i.S0)))$ implies that $\mathcal{M} \models Holds(\psi, S_i)$.

Thus, $i > 0$ (recall that $i$ is the first index for which our assertion is not true). Thus, the value of $Holds$ in $S_0, \ldots, S_i$ (in the respective partitions $SC^1, ..., SC^i$) according to $\mathcal{M}'$ is not consistent with the axioms of $SC$, but the value of $Holds$ in $S_0, \ldots, S_{i-1}$ is consistent with $SC$.

Let $\mathcal{A}'$ be the set of formulae of the form $SC^j.Holds(\psi, SC^j.res(SC^j.a_j, ..., SC^j.res(SC^j.a_1, SC^j.S0)))$ or $SC^j.Holds(\psi, SC^j.res(SC^j.a_{j-1}, ..., SC^j.res(SC^j.a_1, SC^j.S0)))$ that $\mathcal{M}'$ satisfies, for $j \leq i$. Let $\mathcal{A}$ be the translation of $\mathcal{A}'$ into $L(SC)$ (essentially removing $SC^j$ from all the axioms in $\mathcal{A}'$).

$\mathcal{A}$ is consistent because the only way it may not be consistent is if it includes both $SC^j.Holds(\psi, SC^j.res(SC^j.a_{j-1}, ..., SC^J.res(SC^j.a_1, SC^j.S0)))$ and $\neg SC^{j-1}.Holds(\psi, SC^{j-1}.res(SC^{j-1}.a_{j-1}, ..., SC^{j-1}.res(SC^{j-1}.a_1, SC^{j-1}.S0)))$ for some formula $\psi \in \mathcal{L}(SC^j) \cap \mathcal{L}(SC^{j-1})$, but this is inconsistent with the set of formulae $f_{j-1} \iff SC^j.Holds(\psi, SC^j.res(SC^j.a_{j-1}, ..., SC^j.res(SC^j.a_1, SC^j.S0)))$ and $f_{j-1} \Leftrightarrow SC^{j-1}.Holds(\psi, SC^{j-1}.res(SC^{j-1}.a_{j-1}, ..., SC^{j-1}.res(SC^{j-1}.a_1, SC^{j-1}.S0)))$.

Furthermore, let $\mathcal{A}_0$ be the set of formulae of $\mathcal{A}$ that do not mention $a_i$. Then, $\mathcal{A}_0$ is consistent with $SC$, but $\mathcal{A}$ is not. Let $\bar{SC}$ be the set of axioms of $SC$ with the additional precondition that the variable $s$ in those axioms that mention $a_i$ is different from $S_{i-1}$. $\bar{SC} \cup \mathcal{A}_0$ is consistent because $SC \cup \mathcal{A}_0$ is. Consequently, $\bar{SC} \cup \mathcal{A}$ is consistent because there are no axioms in $\bar{SC}$ that can interact with those of $\mathcal{A} \setminus \mathcal{A}_0$ (there is no effect axiom in $\bar{SC}$ that tells us anything about this particular situation).

Let $\mathcal{M}$ be a model of $\bar{SC} \cup \mathcal{A}$. We show that $\mathcal{M} \models SC$. Assume not. Then, there is an effect axiom in $SC$, $\Psi = \forall s\ Holds(\psi_1, s) \Rightarrow Holds(\psi_2, res(a_i, s))$, such that $\mathcal{M} \not\models \Psi$. Thus, $\mathcal{M} \models \neg\Psi$, because $\mathcal{M}$ is a structure and $\Psi$ a closed formula. Since $\mathcal{M} \models \bar{\Psi}$ (for $\bar{\Psi}$ being the matching axiom in $\bar{SC}$) we get that $\mathcal{M} \models \neg(Holds(\psi_1, S_{i-1}) \Rightarrow Holds(\psi_2, res(a_i, S_{i-1})))$. Rewriting this formula yields $\mathcal{M} \models Holds(\psi_1, S_{i-1}) \wedge \neg Holds(\psi_2, res(a_i, S_{i-1}))$.

Let $SC^i.\Psi = \forall s\ SC^i.Holds(\psi_1, s) \Rightarrow SC^i.Holds(\psi_2, SC^i.res(SC^i.a_i, s))$. By the way we defined $SC^i$, the axiom $SC^i.\Psi$ appears in $SC^i$. In particular $SC^i \models SC^i.Holds(\psi_1, SC^i.S_{i-1}) \Rightarrow SC^i.Holds(\psi_2, SC^i.res(SC^i.a_i, SC^i.S_{i-1}))$.

From the paragraph preceding the last one, $\mathcal{M} \models Holds(\psi_1, S_{i-1})$. Furthermore, $\psi_1 \in \mathcal{L}(SC^i)$ by the definition of $SC'$. Since $\mathcal{M} \models Holds(\psi_1, S_{i-1})$ it must be that $\mathcal{M}' \models SC^i.Holds(\psi_1, SC^i.S_{i-1})$ because otherwise $\mathcal{M}' \models SC^i.Holds(\neg\psi_1, SC^i.S_{i-1})$ which implies $\mathcal{M} \models Holds(\neg\psi_1, S_{i-1})$, which is a contradiction.

Thus, $\mathcal{M}' \models SC^i.Holds(\psi_1, SC^i.S_{i-1})$, and we get that $\mathcal{M}' \models SC^i.Holds(\psi_2, SC^i.res(SC^i.a_i, SC^i.S_{i-1}))$. This contradicts our previous consequence that $\mathcal{M} \models \neg Holds(\psi_2, res(a_i, S_{i-1}))$ (which implies $\mathcal{M}' \models \neg SC^i.Holds(\psi_2, SC^i.res(SC^i.a_i, SC^i.S_{i-1})))$.

Thus, $\mathcal{M} \models SC \cup \mathcal{A}$ which concludes the induction step.

**Adding Explanation Closure** First we examine the way that we add explanation closure axioms to $SC$ and to $SC'$. Without loss of generality, assume that our explanation closure axiom $\Psi$ is of the form

$$\Psi = \forall s, a\ Holds(f, res(a, s)) \Rightarrow (Holds(\psi_1, s) \wedge a = A_1) \vee ... \vee (Holds(\psi_k, s) \wedge a = A_k) \vee (Holds(f, s) \wedge a \neq A_1 ... A_k)$$

If $f \in L(SC_1) \setminus L(SC_2)$, then all the effect axioms that influence $f$ are in $SC_1$ and all of $A_1, ..., A_k$ are in $L(SC_1)$. Thus, $\Psi$ is added to $SC_1$ in $SC$ and $SC^i.\Psi$ is added to the copies $SC^i$ of $SC_1$ in $SC'$. The situation is the opposite if $f \in L(SC_2) \setminus L(SC_1)$.

If $f \in L(SC_1) \cap L(SC_2)$, then there are effect axioms for different actions in $SC_1, SC_2$ that influence $f$. Let $A_1, ..., A_l$ be the actions for whose effect axioms are in $SC_1$ and $A_{l+1}, ..., A_k$ be the actions for whose effect axioms are in $SC_2$. In $SC_1$ we add the effect axiom

$$\Psi_1 = \forall s, a\ Holds(f, res(a, s)) \Rightarrow (Holds(\psi_1, s) \wedge a = A_1) \vee ... \vee (Holds(\psi_l, s) \wedge a = A_l) \vee (a = A_{l+1} ... A_k) \vee (Holds(f, s) \wedge a \neq A_1 ... A_k)$$

and in $SC_2$ we add the effect axiom

$$\Psi_2 = \forall s, a\ Holds(f, res(a, s)) \Rightarrow (Holds(\psi_{l+1}, s) \wedge a = A_{l+1}) \vee ... \vee (Holds(\psi_k, s) \wedge a = A_k) \vee (a = A_1 ... A_l) \vee (Holds(f, s) \wedge a \neq A_1 ... A_k).$$

First, notice that $\Psi \Rightarrow \Psi_1 \wedge \Psi_2$ because each of $\Psi_1, \Psi_2$ is a weakened version of $\Psi$ (we weakened the consequent of $\Psi$ to get each of them). Then, notice that $\Psi_1 \wedge \Psi_2 \Rightarrow \Psi$. This is because we include a UNA for actions, which implies that $(a = A_{l+1} \vee ... \vee a = A_k) \Rightarrow \neg(a = A_1 \vee ... \vee a = A_l)$, and using the resolution rule (see (Genesereth and Nilsson, 1987)) we get $\Psi$. Thus,

$$\Psi_1 \wedge \Psi_2 \equiv \Psi.$$

For all $SC^i$ that is a copy of $SC_1$ we add $SC^i.\Psi_1$. Also, for all $SC^i$ that is a copy of $SC_2$ we add $SC^i.\Psi_2$.

Now, assume that $SC'$ already includes all the explanation closure axioms as detailed above and that $SC$ has no explanation closure. Let $[\![SC]\!]$ be the set of models of $SC$.

Let $\mathcal{M}' \in [\![SC']\!]$ a model of $SC'$ together with the explanation closure axioms. Let $\mathcal{M} \in [\![SC]\!]$ be a matching model, as found by our previous section of the proof (*Only Effect Axioms*).

(Lin and Reiter, 1994) provided a model-minimization policy that gives semantics[3] to explanation closure axioms. Roughly, it says that $\mathcal{M}$ is minimal if there is no model $\mathcal{M}'$ which agrees with $\mathcal{M}$ on all actions $a$, fluents $f$ and situations $s$ for which $\mathcal{M} \models Holds(f,s) \equiv Holds(f, res(a,s))$ but that has fluent $f$, situation $s$ and action $a$ for which $a$ is possible ($Poss(a,s)$) in $\mathcal{M}$ and $\mathcal{M} \models Holds(f,s) \equiv \neg Holds(f, res(a,s))$ but for which $\mathcal{M}' \models Holds(f,s) \equiv Holds(f, res(a,s))$. If there is such a model, $\mathcal{M}'$, we write $\mathcal{M}' \prec \mathcal{M}$. This minimization is for models that satisfy the effect axioms, the UNAs for actions and the foundational axioms for situation calculus (inclusion of other formulae is done only after the minimization is complete; the process, called *filtering*, was first discussed by Sandewall (Sandewall, 1989; Sandewall, 1994)).

We use this minimization policy to show that our $\mathcal{M}'$ has a corresponding model in the sense of our first case above (*Only Effect Axioms*). If $\mathcal{M}$ is not minimal, let $\widetilde{\mathcal{M}}$ such that $\widetilde{\mathcal{M}} \prec \mathcal{M}$. Assume that $\widetilde{\mathcal{M}}$ differs from $\mathcal{M}$ in $Holds(f, res(a_{i+1}, S_i))$ for a minimal $i \leq n$ (otherwise, $\widetilde{\mathcal{M}}$ still matches our $\mathcal{M}'$). Then $\widetilde{\mathcal{M}} \models Holds(f, S_i) \equiv Holds(f, res(a_{i+1}, S_i))$ while $\mathcal{M} \models Holds(f, S_i) \equiv \neg Holds(f, res(a_{i+1}, S_i))$.

If $f \in L(SC^{i+1})$, then $\mathcal{M}' \models SC^{i+1}.Holds(f, SC^{i+1}.S_i) \equiv \neg SC^{i+1}.Holds(f, SC^{i+1}.res(SC^{i+1}.a_{i+1}, SC^{i+1}.S_i))$ because of the similar property that holds in $\mathcal{M}$ and the fact that $\mathcal{M}', \mathcal{M}$ agree on $f$ in $S_i, S_{i+1}$. However, recall that $\mathcal{M}'$ satisfies all the explanation closure axioms that are stated in $SC'$. Since $\widetilde{\mathcal{M}} \models SC$ and is $\prec$-smaller than $\mathcal{M}$ we get another frame axiom (for the exact state of $S_i^{\mathcal{M}}$) that we can compile into an explanation closure axiom for $f$. This axiom then implies that

$$\forall a, s\ Holds(f,s) \wedge \neg Holds(f, res(a,s)) \Rightarrow (\neg Holds(state\_in\_S_i, s) \vee a \neq a_{i+1}).$$

Recall our observation about the way we can split an explanation closure axiom into the two partitions in a seemless way. By this observation we should have added the proper explanation closure axioms to the copies of $SC_1, SC_2$ in $SC'$. This would have prevented us from having $\mathcal{M}'$ as a model of $SC'$. Contradiction.

If $f \notin L(SC^{i+1})$, then $\mathcal{M}' \models f_i \equiv f_{i+1}$. This guarantees that if $\mathcal{M}'$ and $\widetilde{\mathcal{M}}$ agree on the value of $f$ in situation $S_i$ then they also agree in situation $S_{i+1}$ (where $\mathcal{M}, \mathcal{M}'$ did not agree). Let $\widetilde{\mathcal{M}}$ be such that $\widetilde{\mathcal{M}} \prec \mathcal{M}$ and the first $S_i$ in which $\widetilde{\mathcal{M}}, \mathcal{M}$ disagree has the smallest index $i$ among all models that are $\prec$-smaller than $\mathcal{M}$. We show that, for this $i$, $\mathcal{M}'$ and $\widetilde{\mathcal{M}}$ agree on the value of $f$ in situation $S_i$. This will show by induction that there is no situation $S_j$ in which $\mathcal{M}', \widetilde{\mathcal{M}}$ disagree (on the fluent-situation combinations that count, namely, those mentioned in the previous section of our proof).

Otherwise, $\mathcal{M}', \widetilde{\mathcal{M}}$ disagree on $f$ in $S_i$ and thus $\mathcal{M}', \mathcal{M}$ agree on $f$ in $S_i$. Take $j \leq i$ such that $j$ is the smallest index for which $\mathcal{M}', \mathcal{M}$ disagree on $f$ in $S_j$ and $SC^j, ..., SC^i$ are copies of the same partition of $SC$. Similarly, take $k \geq i$ such that $k$ is the largest index for which $\mathcal{M}', \mathcal{M}$ disagree on $f$ in $S_k$ and $SC^i, ..., SC^k$ are copies of the same partition of $SC$. If there is no such $j$, then $\widetilde{\mathcal{M}} \not\prec \mathcal{M}$ because $\mathcal{M}' \models f_i \iff f_{i+1}$ and similarly does $\mathcal{M}$ (because we did not find such $j$) while $\widetilde{\mathcal{M}}$ does have a change between $Holds(f, S_i)$ and $Holds(f, S_{i+1})$. Thus, there are such $j$ and $k$.

Define $\widehat{\mathcal{M}}$ to be identical to $\mathcal{M}$ but with the $Holds(f, S_l)^{\widehat{\mathcal{M}}} \equiv f_l^{\mathcal{M}'}$ for all $l$ such that $j \leq l \leq k$. $\widehat{\mathcal{M}} \prec \mathcal{M}$ because in $\widehat{\mathcal{M}}$ there is no change in the value of $f$ throughout the situations $S_j, ..., S_k$ whereas there is at least one such change for $S_i$ in $\mathcal{M}$. Furthermore, $\widehat{\mathcal{M}} \models SC$ because it satisfies all the effect axioms for all unchanged situations and fluents and there is no effect axiom that constrain $f$ in those situations (the actions we take in $S_j, ..., S_k$ are in the other partition of $SC$, so $f$ does not show in their effect axioms (recall that in this part of the proof $SC$ includes *only effects axioms*)).

This means that we found a model of $SC$ that contradicts our choice of $\widetilde{\mathcal{M}}$ (we chose it to have the first index $i$ such that $\widetilde{\mathcal{M}}, \mathcal{M}$ disagree on $S_i$). Thus, if $\mathcal{M}$ is not $\prec$-minimal then we can find a $\prec$-smaller model that

[3] Other semantics are sometimes used for nondeterministic actions (e.g., (Levesque et al., 1997)). The following proof proceeds similarly for those.

agrees with $\mathcal{M}'$. This, together with an assumption of well-foundedness of $\prec$ (also called smoothness (Kraus et al., 1990)), which is assumed by (Lin and Reiter, 1994), provides our result: For every model $\mathcal{M}'$ of $SC'$ there is a corresponding $\prec$-minimal model of $SC$. As a result, if $\widetilde{SC}$ is the set of effect axioms ($SC$) together with the explanation closure axioms, then for every $\mathcal{M}' \models SC'$ there is $\mathcal{M} \models \widetilde{SC}$ such that if $\mathcal{M}' \models SC^i.Holds(\psi, SC^i.res(SC^i.a_i, ..., SC^i.res(SC^i.a_1, SC^i.S0)...))$, for $\psi \in \mathcal{L}(SC^i)$, then $\mathcal{M} \models Holds(\psi, S_i)$.

**Adding Observations** The overall process that leads to the solution to the frame problem along the lines of successor-state axioms first computes those successor state axioms from effect axioms and state constraints (in the form of ramification constraints) and only then includes observations.

As a result, the pure deductive approach that we are taking here cannot treat state constraints. Any state constraints that we add are treated as observations.

For the case of observations (no uncompiled state constraints), assume that we have our set of explanation closure axioms already compiled into $SC$ and $SC'$. Adding the observations at the proper places is analogous to removing models from each one of them (those that contradict the observations). These observations do not need to be deterministic, but each sentence should be expressible in the language of a single situation and a single partition from $SC$. Then, we get the previous result about $\mathcal{M}'$ having a corresponding $\mathcal{M}$ immediately. ■

## References


Amir, E. (2000). (De)Composition of situation calculus theories. In *Proc. National Conference on Artificial Intelligence (AAAI '00)*, pages 456–463. AAAI Press/MIT Press.

Amir, E. (2001). Efficient approximation for triangulation of minimum treewidth. In *Proc. Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI '01)*, pages 7–15. Morgan Kaufmann.

Amir, E. (2002a). Projection, filtering and belief state representation in nondeterministic dynamic systems. Submitted for publication. Available on the author's website.

Amir, E. (2002b). Projection in decomposed situation calculus (with proofs). Available on the author's website.

Amir, E. and Maynard-Reid II, P. (1999). Logic-based subsumption architecture. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence (IJCAI '99)*, pages 147–152.

Amir, E. and McIlraith, S. (2000). Paritition-based logical reasoning. In *Principles of Knowledge Representation and Reasoning: Proc. Seventh Int'l Conference (KR '2000)*, pages 389–400. Morgan Kaufmann.

Baral, C., Kreinovich, V., and Trejo, R. (2000). Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence*, 122(1-2):241–267.

Boutilier, C., Reiter, R., and Price, B. (2001). Symbolic dynamic programming for first-order MDPs. In *Proc. Seventeenth International Joint Conference on Artificial Intelligence (IJCAI '01)*, pages 690–697. Morgan Kaufmann.

Bylander, T. (1994). The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1–2):165–204.

Darwiche, A. (1998). Model-based diagnosis using structured system descriptions. *Journal of Artificial Intelligence Research*, 8:165–222.

Dechter, R. and Pearl, J. (1989). Tree clustering for constraint networks. *Artificial Intelligence*, 38:353–366.

Finzi, A., Pirri, F., and Reiter, R. (2000). Open world planning in the situation calculus. In *Proc. National Conference on Artificial Intelligence (AAAI '00)*, pages 754–760. AAAI Press.

Frank, J., Johnsson, A. K., and Morris, P. (2000). On reforumlating planning as dynamic constraint satisfaction. In Choueiry, B. and Walsh, T., editors, *Abstraction, reformulation and approximation, Proceedings of the 4th international symposium (SARA'2000)*, volume 1864 of *LNAI*, pages 271–280. Springer-Verlag.

Genesereth, M. R. and Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

Green, C. (1969). Theorem-proving by resolution as a basis for question-answering systems. In Meltzer, B., Michie, D., and Swann, M., editors, *Machine Intelligence 4*, pages 183–205. Edinburgh University Press, Edinburgh, Scotland.

Haas, A. R. (1987). The case for domain-specific frame axioms. In *proceedings of the 1987 workshop on the frame problem*, pages 343–348.

Kraus, S., Lehmann, D., and Magidor, M. (1990). Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. *Artificial Intelligence*, 44(1):167–207.

Lansky, A. (1988). Localized event-based reasoning for multiagent domains. *Computational Intelligence Journal, Special Issue on Planning*, 4(4):319–340.

Lansky, A. L. and Getoor, L. C. (1995). Scope and abstraction: Two criteria for localized planning. In *Proc. Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 1612–1618, Montreal, Canada.

Levesque, H., Reiter, R., Lesprance, Y., Lin, F., and Scherl, R. (1997). Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31:59–84.

Lin, F. (1996). Embracing causality in specifying the indeterminate effects of actions. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.

Lin, F. and Reiter, R. (1994). State constraints revisited. *Journal of Logic and Computation, Special Issue on Actions and Processes*.

Lin, F. and Reiter, R. (1995). How to progress a database II: The STRIPS connection. In *Proc. Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 2001–2007, Montreal, Canada.

McCarthy, J. and Hayes, P. J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. In Meltzer, B. and Michie, D., editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press.

McIlraith, S. (1997). Representing action and state constraints in model-based diagnosis. In Senator, T. and Buchanan, B., editors, *Proc. National Conference on Artificial Intelligence (AAAI '97)*, pages 43–49, Menlo Park, California. American Association for Artificial Intelligence, AAAI Press.

McIlraith, S. (2000). Integrating actions and state constraints: a closed-form solution to the ramification problem (sometimes). *Artificial Intelligence*, 16(1–2):87–121.

McIlraith, S. and Amir, E. (2001). Theorem proving with structured theories. In *Proc. Seventeenth International Joint Conference on Artificial Intelligence (IJCAI '01)*, pages 624–631. Morgan Kaufmann.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann.

Pednault, E. P. D. (1989). ADL: exploring the middle ground between STRIPS and the situation calculus. In *Proc. First International Conference on Principles of Knowledge Representation and Reasoning (KR '89)*, pages 324–332.

Pfeffer, A. (2001). Sufficiency, separability and temporal probabilistic models. In *Proc. Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI '01)*, pages 421–428. Morgan Kaufmann.

Pinto, J. (1999). Compiling ramification constraints into effect axioms. *Computational Intelligence*, 15(3):280–307.

Pinto, J. A. (1994). *Temporal Reasoning in the Situation Calculus*. Ph.d. dissertation, Department of Computer Science, University of Toronto, Toronto. Also available as Technical Report Number KRR-TR-94-1.

Reiter, R. (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., editor, *Artificial Intelligence and Mathematical Theory of Computation (Papers in Honor of John McCarthy)*, pages 359–380. Academic Press.

Reiter, R. (1992). The projection problem in the situation calculus. In *Proceedings AIPS*, pages 198–203.

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

Sandewall, E. (1989). Filter preferential entailment for the logic of action in almost continuous worlds. In *IJCAI-89*.

Sandewall, E. (1994). *Features and Fluents*. Oxford University Press.

Santibaez, J. S. (1999). Declarative formalization of reasoning strategies: a case study on nonlinear planning. In Golumbic, M., Morgenstern, L., and Shimoni, E., editors, *Bar Ilan Symposium on the Foundations of Artificial Intelligence*. http://www-formal.stanford.edu/jsierra/aaai-draft-98-s.ps.

Schubert, L. K. (1990). Monotonic solution of the frame problem in the situation calculus. In Kryburg, H., Loui, R., and Carlson, G., editors, *knowledge representation and defeasible reasoning*, pages 23–67. Kluwer.

Shanahan, M. (1997). *Solving the Frame Problem, a mathematical investigation of the common sense law of inertia*. MIT press, Cambridge, MA.