Formalizing Narratives using Nested Circumscription

Chitta Baral*, Alfredo Gabaldon* and Alessandro Provetti‡

*Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968
U.S.A.
{chitta,alfredo}@cs.utep.edu

‡ C.I.R.F.I.D.
Università di Bologna
Via Galliera, 3. I-40121
Bologna, Italy
provetti@cirfid.unibo.it

Abstract

The representation of narratives of actions and observations is a current issue in Knowledge Representation, where traditional plan-oriented treatments of action seem to fall short. To address narratives, Pinto and Reiter have extended Situation Calculus axioms, Kowalski and Sergot have introduced the Event Calculus in Logic Programming, and Baral et al. have defined the specification language $\mathcal{L}$ which allows to express actual and hypothetical situations in a uniform setting. The $\mathcal{L}$ entailment relation can formalize several forms of reasoning about actions and change. In this paper we illustrate a translation of $\mathcal{L}$ theories into Nested Abnormality Theories, a novel form of circumscription. The proof of soundness and completeness of the translation is the main technical result of the paper, but attention is also devoted to the features of Nested Abnormality Theories to capture commonsense reasoning in general and to clarify which assumptions a logical formalization forces upon a domain. These results also help clarifying the relationship between $\mathcal{L}$ and other recent circumscriptive formalizations for narratives, such as Miller and Shanahan's.

Content Areas Temporal Reasoning, Nonmonotonic Reasoning, Knowledge Representation.

Introduction

The action description language $\mathcal{L}$ was introduced in (Baral, Gelfond & Provetti 1995) to incorporate narratives and actual situations (where actual situations are interpreted as actions that are part of the evolution) into the action description language $\mathcal{A}$ (Gelfond & Lifschitz 1992) that only allowed hypothetical situations. (Baral, Gelfond & Provetti 1995) contains the syntax and semantics of $\mathcal{L}$ along with a sound translation to logic programs.

Although narratives were earlier formalized in isolation using event calculus (Kowalski & Sergot 1986), only recently have several proposals been put forward that formalize narratives together with hypothetical situations (McCarhty 1995; Pinto & Reiter 1993; Miller & Shanahan 1994; Provetti 1996). All but the last have used situation calculus together with circumscription.

In this paper we present a translation of $\mathcal{L}$ to nested abnormality theories (NATs) (Lifschitz 1995), a new approach to the use of circumscription for representing knowledge. Although the nesting of blocks in NATs may at the first glance suggest loss of declarativeness and elaboration tolerance (McCarthy 1988), we believe it makes it easier to represent knowledge, particularly in the action domain (See (Lifschitz 1995; Kartha & Lifschitz 1994; Giunchiglia, Kartha & Lifschitz 1995) for more on this). This is because we can develop blocks that represent a meaningful structural unit and use the blocks in other units without worrying about undesired interactions[1].

We believe that the simplicity of our translation of $\mathcal{L}$ (as presented in this paper) demonstrates the usefulness of NATs for knowledge representation. Besides that, our result will make it easier to compare the entailment of $\mathcal{L}$ to that of the formalizations in (Pinto & Reiter 1993; Miller & Shanahan 1994; Shanahan 1995).

Overview of $\mathcal{L}$

The alphabet of $\mathcal{L}$ consists of three disjoint nonempty sets of symbols $\mathcal{F}$, $\mathcal{A}$ and $\mathcal{S}$, called *fluents*, *actions*, and *actual situations*. Elements of $\mathcal{A}$ and $\mathcal{S}$ will be denoted by (possibly indexed) letters A and S respectively. We will also assume that $\mathcal{S}$ contains two special situations S_0 and S_N called *initial* and *current* situation, respectively.

A *fluent literal* is a fluent possibly preceded by $\neg$. Fluent literals will be denoted by (possibly indexed) letters F and P (possibly preceded by $\neg$). $\neg\neg F$ will be equated to F. For a fluent F, $\overline{\neg F} = F$, and $\overline{F} = \neg F$.

*Support was provided by the National Science Foundation under grant Nr. IRI-9211662 and IRI-9501577.

[1] (Shanahan 1995) also has a discussion related to this. He is concerned with formalizations where minimization of change does not interfere with minimization of other knowledge.

There are two kinds of propositions in $\mathcal{L}$ called causal laws and facts. A *causal law* is an expression of the form:

$$A \textbf{ causes } F \textbf{ if } P_1, \ldots, P_n \qquad (1)$$

where A is an action, and $F, P_1, \ldots, P_n$ ($n \geq 0$) are fluent literals. $P_1, \ldots, P_n$ are called *preconditions* of (1). We will read this law as "F is guaranteed to be true after the execution of an action A in any state of the world in which $P_1 \ldots P_n$ are true". If $n = 0$, we write the causal law as A **causes** F.

An atomic *fluent fact* is an expression of the form

$$F \textbf{ at } S \qquad (2)$$

where F is a fluent literal and S is a situation. (Unless otherwise stated by situations we will mean actual situations.) The intuitive reading of (2) is "F is observed to be true in situation S".

An atomic *occurrence fact* is an expression of the form

$$\alpha \textbf{ occurs\_at } S \qquad (3)$$

where α is a sequence of actions, and S is a situation. It states that "the sequence α of actions was observed to have occurred in situation S". (We assume that actions in the sequence follow the next action in the sequence immediately).

An atomic *precedence fact* is an expression of the form

$$S_1 \textbf{ precedes } S_2 \qquad (4)$$

where S_1 and S_2 are situations. It states that the domain was in situation S_2 after being in situation S_1.

Propositions of the type (1) express general knowledge about effects of actions and hence are referred to as *laws*. Propositions (2), (3) and (4) are called *atomic facts* or *observations*. A *fact* is a propositional combination of atomic facts, while a collection of laws and facts is called a *domain description* of $\mathcal{L}$. The sets of laws and facts of a domain description D will be denoted by D_l and D_f respectively. We will only consider domain descriptions whose propositions do not contain the situation constant S_N.

Example 1 *(discovering occurrences)* Consider the following story: initially F is known to be false. At a later moment F is observed to be true. We know that there is only one action A, and that it causes F to become true. This story is described in $\mathcal{L}$ thus:

$$D_1 = \begin{cases} A \textbf{ causes } F \\ \neg F \textbf{ at } S_0 \\ F \textbf{ at } S_1 \\ S_0 \textbf{ precedes } S_1 \end{cases}$$

Intuitively, we would like to conclude that action A occurred in the initial situation causing F to become true.

Assumptions embodied in $\mathcal{L}$ domain descriptions

Domain descriptions in $\mathcal{L}$ are used in conjunction with the following informal assumptions which clarify their meaning:

a) Changes in the values of fluents can only be caused by execution of actions;
b) there are no actions except those from the language of the domain description;
c) there are no effects of actions except those specified by the causal laws;
d) No actions occur except those needed to explain the facts in the domain description;
e) Actions do not overlap or happen simultaneously.

These assumptions give an intuitive understanding of domain descriptions of $\mathcal{L}$. In the rest of the section we present the semantics of domain descriptions of $\mathcal{L}$ defined in (Baral, Gelfond & Provetti 1995), which precisely specifies the sets of acceptable conclusions which can be reached from such descriptions and assumptions (a)-(e).

Semantics of $\mathcal{L}$

Let us introduce the semantics of $\mathcal{L}$ domain descriptions. We start with defining causal models of D and the entailment relation ascribed to them.

Let a *state* be a set of fluent names. A *causal interpretation* is a partial function Ψ from sequences of actions to states such that: (i) the empty sequence [ ] belongs to the domain of Ψ; and (ii) Ψ is prefix-closed[2].

$\Psi([\,])$ is called the initial state of Ψ. The partial function Ψ serves as interpretation of the laws of D. If α belongs to the domain of Ψ we say that α is *possible* in the initial state of Ψ.

Given a fluent F and a state σ, we say that F *holds* in σ (F is *true* in σ) if $F \in \sigma$; $\neg F$ *holds* in σ (F is *false* in σ) if $F \notin \sigma$. The *truth* of a propositional formula w.r.t. σ is defined as usual.

To better understand the role Ψ plays in interpreting domain descriptions let us first use it to define *models* of descriptions consisting entirely of causal laws. To this goal we will attempt to carefully define effects of actions as determined by such a description D and our informal assumptions (a)-(e).

A fluent F is an *immediate effect* of (executing) A in σ if there is an effect law (1) in D whose preconditions hold in σ. Let us define the following sets:

$$E^+_A(\sigma) = \{F : F \text{ is an imm. effect of } A \text{ in } \sigma\}$$
$$E^-_A(\sigma) = \{F : \neg F \text{ is an imm. effect of } A \text{ in } \sigma\}$$
$$Res(A, \sigma) = \sigma \cup E^+_A(\sigma) \setminus E^-_A(\sigma)$$

[2]By "prefix closed" we mean that for any sequence of actions α and action A, if $\alpha \circ A$ is in the domain of Ψ then so is α, where $\alpha \circ A$ means the sequence of actions where A follows α.

The following definition captures the meaning of causal laws of D.

Definition 1 (Causal interpretation) *A causal interpretation Ψ satisfies causal laws of D if for any sequence $\alpha \circ A$ from the language of D*

$$\Psi(\alpha \circ A) = \begin{cases} Res(A, \Psi(\alpha)) & \\ \quad \text{if } E_A^+(\Psi(\alpha)) \cap E_A^-(\Psi(\alpha)) = \emptyset & \\ undefined & otherwise \end{cases}$$

We say that Ψ is a causal model of D if it satisfies all the causal laws of D. □

Let D be an arbitrary domain description and let a causal interpretation Ψ be a causal model of D. To interpret the observations of D we first need to define the meaning of situation constants $S_0, S_1, S_2, \ldots$ from $\mathcal{S}$. To do that we consider a mapping Σ from $\mathcal{S}$ to sequences of actions from the language of D.

Definition 2 *(Situation assignment) A mapping Σ is called a situation assignment of $\mathcal{S}$ if it satisfies the following properties:*

1. $\Sigma(S_0) = [\,]$;
2. $\forall s_i \in \mathcal{S}.\ \Sigma(S_i)$ *is a prefix of* $\Sigma(S_N)$.

□

Definition 3 *(Interpretation) An interpretation M of $\mathcal{L}$ is a pair (Ψ, Σ), where Ψ is a causal model of D, Σ is a situation assignment of $\mathcal{S}$ and $\Sigma(S_N)$ belongs to the domain of Ψ. $\Sigma(S_N)$ will be called the actual path of M and for simplicity will often be denoted by Σ_N.* □

Now we can define truth of facts of D w.r.t. an interpretation M. Facts which are not true in M will be called *false* in M.

Definition 4 (Entailment in $\mathcal{L}$) *For any interpretation $M = (\Psi, \Sigma)$.*

1. *(F **at** S) is true in M (or satisfied by M) if F is true in $\Psi(\Sigma(S))$;*
2. *(α **occurs_at** S) is true in M if $\Sigma(S) \circ \alpha$ is a prefix of the actual path of M;*
3. *(S_1 **precedes** S_2) is true in M if $\Sigma(S_1)$ is a proper prefix of $\Sigma(S_2)$*

Truth of non-atomic facts in M is defined as usual. Of course, a set of facts is true in interpretation M if all its members are true in M. □

To complete the definition of the model we need only to formalize assumption *d)* on domain descriptions: "no actions occur except those needed to explain the facts in the domain description". This is done by imposing a minimality condition on the situation assignments of $\mathcal{S}$ which leads to the following definition.

Definition 5 *An interpretation $M = (\Psi, \Sigma)$ will be called a model of a domain description D in $\mathcal{L}$ if the following conditions are satisfied:*

1. *Ψ is a causal model of D;*
2. *facts of D are true in M;*
3. *there is no other interpretation $N = (\Psi, \Sigma')$ such that N satisfies conditions 1) and 2) and $\Sigma'(S_N)$ is a subsequence[3] of $\Sigma(S_N)$.* □

The final definition is matter of course.

Definition 6 *A domain description D is said to be consistent if it has a model. A domain description D entails a fact ϕ —written $D \models_{\mathcal{L}} \phi$— iff ϕ is true in all models of D.* □

Example 2 Consider domain description D_1 from example (1). For every model $M = (\Psi, \Sigma)$ of D_1

- $F \notin \Psi(\Sigma(S_0))$, $F \in \Psi(\Sigma(S_1))$, $F \in \Psi(\Sigma(S_N))$
- $\Sigma(S_0) = [\,]$, $\Sigma(S_1) = \Sigma(S_N) = [A]$

Therefore, D_1 entails A **occurs_at** S_0.

From $\mathcal{L}$ to NAT

Nested Abnormality Theories (NATs) are a novel circumscription (McCarthy 1986; Lifschitz 1994) technique introduced in (Lifschitz 1995). NATs allow the use of several abnormality predicates to specify a body of common-sense knowledge without their circumscription conflicting. In this section we present a translation of domain descriptions in $\mathcal{L}$ into equivalent NATs.

Sort Definitions

$\mathcal{A} : A, A_1, \ldots$	actions;
$\mathcal{A}^* : \alpha, \alpha_1, \ldots$	sequences of actions (denote states);
$\mathcal{F} : F, F_1, \ldots$	fluents;
$\mathcal{S} : S, S_1, \ldots$	[actual] situations;

Sequences of actions are defined as follows. Assume the special constant symbol $\epsilon \in \mathcal{A}^*$ (representing the empty sequence) and function $\circ\ :\ \mathcal{A} \times \mathcal{A}^* \mapsto \mathcal{A}^*$. Then, sequences are denoted by nesting of $\circ$:

$$A_n \circ A_{n-1} \circ \ldots A_1 \circ \epsilon$$

where parentheses can be ignored without ambiguity.

Predicate and function definitions

$holds(F, \alpha)$: F is true after performing α from the init. situation;
$causes^+(A, F, \alpha)$: A performed in the state reached by α makes F true;
$causes^-(A, F, \alpha)$: A performed in the state reached by α makes F false;
$sit\_map : \mathcal{S} \mapsto \mathcal{A}^*$: maps a situation into its corresponding sequence of actions;
$prefix\_eq(\alpha_1, \alpha_2)$: the usual $\leq$ relation on strings;
$subsequence(\alpha_1, \alpha_2)$: α_1 is a subsequence of α_2;
$concatenate(\alpha_1, \alpha_2, \alpha_3)$: α_3 is obtained by concatenating α_1 and α_2.

[3]Recall that $\alpha = A_1, \ldots, A_m$ is a subsequence of $\beta = B_1, \ldots, B_n$ if there exists a strictly increasing sequence $i_1, \ldots, i_m$ of indices of β such that for all $j = 1, \ldots, m$, we have $A_j = B_{i_j}$.

Framework axioms To yield the expected results, the theory includes a set of extra axioms which represent the domain-closure assumption of $\mathcal{L}$ theories, and in particular assumption *b)*: "there are no actions except those from the language of the domain description". Also unique-name assumptions, and minimization of the 1-1 function constant "$\circ$" are captured by these axioms.

$$\forall x.\ action(x) \oplus sequence(x) \oplus fluent(x) \oplus situation(x)$$

$$\{\ min\ sequence\ :$$
$$sequence(\epsilon)$$
$$action(a)\ \wedge sequence(\alpha) \supset sequence(a \circ \alpha)$$

$$\{min\ action\ : action(A_1), \ldots, action(A_n), UNA[\mathcal{A}]\}$$

$$\{min\ fluent\ : fluent(F_1), \ldots, fluent(F_m), UNA[\mathcal{F}]\}$$

$$\{min\ situation\ :\ situation(S_1), \ldots, situation(S_l), UNA[\mathcal{S}]\}$$
$$\}$$

where by $UNA(\sigma)$ we intend a set of inequalities between each pair of constants from the set σ, e.g. $UNA[\mathcal{S}]$ stands for $S_0 \neq S_1$, $S_1 \neq S_2 \ldots$ etc. In the following the formulae above will be referred to as "framework axioms".

Translation of Facts The atomic facts of type (2), (3) and (4) are translated into facts of type $(\neg)at(F, S)$, $occurs(A_n \circ \ldots A_1 \circ \epsilon, S)$, and $precedes(S_1, S_2)$, respectively. Non-atomic Facts are translated in the obvious way. The resulting set of facts is denoted by $FACTS$.

Managing the Action-line

The action-line, as opposed to the time-line, is a sequence of actions to be performed starting from a given state of the domain. When a sequence has taken place in actuality, it is called actual action-line. Otherwise, action-lines represent simple plans.

In this section we discuss a set of auxiliary relations on action-lines needed for specifying domain descriptions. These are *prefix_eq*, *subsequence* and *concatenate*. These relations are defined by the NAT blocks B_{prefix_eq}, $B_{subsequence}$, and $B_{concatenate}$ which are presented in a later section.

It is important to ensure that all and only the intended instances of these relations are included in models of the theory described in the following sections. Let us start by defining the following relation $\preceq \subset \mathcal{A}^* \times \mathcal{A}^*$:

$$A_n \circ \ldots A_1 \circ \epsilon \preceq B_m \circ \ldots B_1 \circ \epsilon \ \Leftrightarrow\ \forall i, i \leq n.\ A_i = B_i$$

where above and in the rest of the section by "$=$" syntactic identity is meant. Relation $\preceq$ captures intuition on what it means for sequence to be a prefix-or-equal of another. This relation is defined by block B_{prefix_eq} in the theory.

The second relation we need to describe is subsequence: $\ll \subset \mathcal{A}^* \times \mathcal{A}^*$:

$$A_n \circ \ldots A_1 \circ \epsilon \ll B_m \circ \ldots B_1 \circ \epsilon \ \Leftrightarrow\ \exists \mu.\ \forall i. A_i = B_{\mu(i)}\ \wedge [i \leq j \Rightarrow \mu(i) \leq \mu(j)]$$

Relation $\alpha \ll \beta$ formalizes the notion of *subsequence:* β contains all the elements of α, in the same order, but it can contain more elements[4]. This relation is defined by block $B_{subsequence}$ in the theory.

The third predicate needed for dealing with action-lines is concatenation. We will concatenate sequences of actions in reverse, i.e. $\alpha \cdot \beta = \beta\alpha$:

$$(A_n \circ A_{n-1} \ldots A_1 \circ \epsilon) \cdot (B_m \circ B_{m-1} \ldots B_1 \circ \epsilon) = B_m \circ B_{m-1} \ldots B_1 \circ A_n \circ A_{n-1} \ldots A_1 \circ \epsilon$$

Block $B_{concatenate}$ defines the predicate *concatenate*.

The resulting NAT

The following theory captures the definition of $\mathcal{L}$-semantics (universal quantification is implicit on all variables):

$$T(D) =$$
$$\{\ sit\_map\ :$$
$$(*_1)\ sit\_map(S_0) = \epsilon$$
$$(*_2)\ prefix\_eq(sit\_map(s), sit\_map(S_N))$$
$$(\star)\ subsequence(\alpha, sit\_map(S_N)) \supset ab(\alpha)$$
$$(d)\ prefix(\alpha_1, \alpha_2) \equiv [prefix\_eq(\alpha_1, \alpha_2) \wedge \neg prefix\_eq(\alpha_2, \alpha_1)]$$
$$\tau(D_f)$$
$$SC(D_l)$$
$$B_{prefix\_eq}$$
$$B_{subsequence}$$
$$B_{concatenate}$$
Framework axioms
$$\}$$

where:

$$\tau(D_f) = \begin{cases} (e)\ at(f, s) \equiv holds(f, sit\_map(s)) \\ (f)\ occurs(\alpha_1, s) \equiv [concatenate(sit\_map(s), \alpha_1, \alpha_2) \wedge prefix\_eq(\alpha_2, sit\_map(S_N))] \\ (g)\ precedes(s_1, s_2) \equiv prefix(sit\_map(s_1), sit\_map(s_2)) \\ FACTS \end{cases}$$

$$SC(D_l) = \begin{cases} \neg causes^+(a, f, \alpha)\ \wedge \neg causes^-(a, f, \alpha) \supset [holds(f, \alpha) \equiv holds(f, a \circ \alpha)] \\ causes^+(a, f, \alpha) \supset holds(f, a \circ \alpha) \\ causes^-(a, f, \alpha) \supset \neg holds(f, a \circ \alpha) \\ \{\ min\ causes^+\ : \\ \quad h(P_1, \alpha)\ \wedge \ldots\ \wedge h(P_n, \alpha) \supset causes^+(A, F, \alpha) \\ \quad\quad \text{(for each } A \textbf{ causes } F \textbf{ if } P_1, \ldots, P_n \in D) \\ \} \\ \{\ min\ causes^-\ : \\ \quad h(P_1, \alpha)\ \wedge \ldots\ \wedge h(P_m, \alpha) \supset causes^-(A, F, \alpha) \\ \quad\quad \text{(for each } A \textbf{ causes } \neg F \textbf{ if } P_1, \ldots, P_m \in D) \\ \} \end{cases}$$

[4] Prefix-equal is a particular case of subsequence.

$B_{prefix\_eq} =$
$\{min\ prefix\_eq$:
$sequence(\alpha) \supset prefix\_eq(\alpha, \alpha)$
$prefix\_eq(\alpha, \alpha_1) \supset prefix\_eq(\alpha, a \circ \alpha_1)$
$\}$

Blocks $B_{subsequence}$ and $B_{concatenate}$ are defined in a similar way as B_{prefix_eq}. In the above translation, for a positive fluent literal F, $h(F, \alpha)$ denotes $holds(F, \alpha)$; while for a negative fluent literal $\neg G$, $h(\neg G, \alpha)$ denotes $\neg holds(G, \alpha)$.

Let us now compare the axioms of $T(D)$ with the definitions in the section on semantics. The axioms in $SC(D_l)$ encode Definition 1. The models of $SC(D_l)$ together with Framework axioms correspond to the causal models of D. The axioms $(*_1)$ and $(*_2)$ encode Definition 2. The axioms in $\tau(D_f)$ encode Definition 4. Finally, the minimality in condition 3 of Definition 5 is encoded by the axiom($\star$) plus circumscription of ab.
This elegant encoding is one of the main contributions of this paper.
By including axiom ($\star$) we exploit the fact that subsequence is a partial order on sequences, i.e. if α is a subsequence of β then all subsequences of the former are also subsequences of the latter. Now, suppose interpretation I of $T(D)$ maps $sit\_map(S_N)$ on α while interpretation I' maps it —other things being equal— onto β. As a result, the extent of ab under I is a proper subset of the extent of ab under I'. Therefore, I' is not a circumscriptive model of $T(D)$.

We believe that this strategy can easily be generalized to cater for minimization of constants and functions w.r.t. any partial ordering on terms.

Proposition 1 *(Minimization of functions and constants)*
Let theory T with Unique names assumptions define a partial ordering relation $\mathcal{R}$ and τ be a ground term. Then, for each model $\mathcal{M}$ of

$\{\tau$:
$\forall x.\ \mathcal{R}(x, \tau) \supset ab(x)$
T
$\}$

If $\mathcal{M} \models \tau = \nu$ then ν is minimal w.r.t. $\mathcal{R}$ and T. □

We will further explore the applications of Proposition 1 for knowledge representation in the follow-up of this work.

Examples

Example 3 Consider again domain description D_1. By translation:

$$FACTS = \begin{cases} \neg at(F, S_0) \\ at(F, S_1) \\ precedes(S_0, S_1) \end{cases}$$

and the sub-block of $SC(D_{l1})$ defining $causes^+$ will be

$\{min\ causes^+$:
$causes^+(A, F, \alpha)$
$\}$

The sub-block defining $causes^-$ will be empty, i.e. $causes^-(a, f, \alpha)$ is false for any a, f, α.

All models of the resulting theory $T(D_1)$ entail the following facts:

- $sit\_map(S_0) = \epsilon$;
- $sit\_map(S_1) = A \circ \epsilon$;
- $sit\_map(S_N) = A \circ \epsilon$;
- $\underline{occurs(A, S_0)}$;
- $causes^+(A, F, \alpha)$;
- $precedes(S_0, S_1)$;
- $\neg holds(F, \epsilon)$, $holds(F, A \circ \epsilon), \ldots$;
- $\neg at(F, S_0)$, $at(F, S_1) \ldots$

Example 4 *(abduction of fluent-values)* Consider the slightly more complex domain description:

$$D_2 = \begin{cases} A \textbf{ causes } F \textbf{ if } P \\ \neg F \textbf{ at } S_0 \\ F \textbf{ at } S_1 \\ S_0 \textbf{ precedes } S_1 \end{cases} \qquad FACTS = \begin{cases} \neg at(F, S_0) \\ at(F, S_1) \\ precedes(S_0, S_1) \end{cases}$$

Again, the sub-block defining $causes^-$ is empty, and the sub-block defining $causes^+$ contains the axiom
$holds(P, \alpha) \supset causes^+(A, F, \alpha)$.

All models of the resulting theory $T(D_2)$ entail the following facts:

- $sit\_map(S_0) = \epsilon$, $sit\_map(S_1) = A \circ \epsilon$, $sit\_map(S_N) = A \circ \epsilon$
- $\neg holds(F, \epsilon)$, $holds(F, A \circ \epsilon)$, $holds(P, \epsilon)$, $holds(P, A \circ \epsilon)$
- $\neg at(F, S_0)$, $at(F, S_1)$, $\underline{at(P, S_0)}$, $at(P, S_1), \ldots$
- $\underline{occurs(A, S_0)}$

plus facts in $\tau(D_2)$ and the extent of $prefix\_eq$ etc.

Correctness of the Translation

Let $\tau(\phi)$ stand for the translation of a domain proposition ϕ as described in the previous section. The equivalence of SC and Ψ is proved in the following lemma (we assume domain descriptions are such that for every model $M = (\Psi, \Sigma)$, Ψ is defined for every sequence of actions. This condition is satisfied if we do not allow contradictory causal laws in domain descriptions, where two causal laws of the form A **causes** F **if** $P_1, \ldots, P_n$ and A **causes** $\neg F$ **if** $Q_1, \ldots, \underline{Q_m}$, are said to be contradictory if $\{P_1, \ldots, P_n\} \cap \{\overline{Q_1, \ldots, Q_m}\} = \emptyset$).

Lemma 1 *(Causal Equivalence)*
Part 1: *For every causal model Ψ of D_l there exists a*

model $\mathcal{M}_{SC}$ of $SC(D_l) \cup$ Framework Axioms such that: for all fluents f and sequences of actions α [5]

$$f \in \Psi(\alpha) \Leftrightarrow \mathcal{M}_{SC} \models holds(f, \alpha). \quad (5)$$

Part2: *For every model $\mathcal{M}_{SC}$ of $SC(D_l)$ there exists a causal model Ψ of D_l such that (5) holds.* □

Lemma 2 (main) *(equivalence of models) For any domain description D, if interpretation (Ψ, Σ) is a model of D then there exists a model $\mathcal{M}$ of $T(D)$ such that for every fact ϕ in the language of D:*

$$(\Psi, \Sigma) \models_{\mathcal{L}} \phi \Leftrightarrow \mathcal{M} \models \tau(\phi) \quad (6)$$

and if $\mathcal{M}$ is a model of $T(D)$ then there exists a model (Ψ, Σ) of D such that (6) holds. □

Theorem 1 (main) *(Equivalence) For any domain description D and fact ϕ in the language of D:*

$$D \models_{\mathcal{L}} \phi \Leftrightarrow T(D) \models \tau(\phi)$$

□

A version of the paper with the proofs is available at http://cs.utep.edu/chitta/papers/L-NAT.ps. The original paper that describes the $\mathcal{L}$ language (Baral, Gelfond & Provetti 1995) is available at http://cs.utep.edu/chitta/papers/actual-actions.ps.

Conclusion

Amongst the other approaches (McCarhty 1995),(Pinto & Reiter 1993),
(Miller & Shanahan 1994), (Provetti 1996) that allow reasoning about both actual and hypothetical situations in a situation calculus framework, the approach in (Miller & Shanahan 1994) is closest to our work. Their function *state* that maps time points to situations (constructed using the function *Res*, the initial situation s_0 and action constants) is similar to our Σ which maps situation constants to sequences of actions. But they assume[6] that the domain description includes all occurrences of actions and they only allow fluent facts about the initial situation. Our approach is more general with respect to these restrictions and from the fact that we allow propositional combination of fluent facts, occurrence facts and precedence facts. Our semantics incorporates the abductive reasoning necessary to make conclusions regarding occurrences of actions and values of fluents in different situations, even when they are not explicitly stated in the domain description. (The section on examples partially illustrates this.) On the other hand (Miller & Shanahan 1994) contains discussions of allowing concurrent, divisible and overlapping actions, which we do not discuss in this paper.

[5] Even though we took the liberty of using the same symbol, α in the context of $\mathcal{L}$ stands for the term $[A_1, \ldots, A_m]$, and in the context of NATs for the term $A_m \circ \ldots \circ A_1 \circ \epsilon$.

[6] They do point out that these assumptions can be weakened using abduction.

References

Baral C., Gelfond M. and Provetti A. 1995. Representing Actions - I: Laws, Observations and Hypothesis. In the Working Notes of AAAI Spring Symposium *Extending Theories of Actions: Formal Theories and Practical Applications.* (Extended version to appear in Journal of Logic Programming.)

Gelfond M. and Lifschitz V. 1992. Representing actions in extended logic programs. In *Joint International Conference and Symposium on Logic Programming*, 559–573.

Giunchiglia E., Kartha N. and Lifschitz V. 1995. Actions with indirect effects (Extended Abstract). Working Notes of AAAI Spring Symposium *Extending Theories of Actions: Formal Theories and Practical Applications.* Stanford.

Kartha N. and Lifschitz V., 1994. Actions with indirect effects (preliminary report). In *KR 94*, 341–350.

Kowalski R. and Sergot M. 1986. A logic-based calculus of events. *New Generation Computing*, 4:67–95.

Lifschitz V. 1995. Nested Abnormality Theories. Artificial Intelligence, 74:351-365.

Lifschitz V. 1994. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming.* Oxford Un. Press.

McCarthy J. 1986. Applications of circumscription to formalizing common sense knowledge. Artificial Intelligence, 26(3):89–116.

McCarthy J 1988. Mathematical logic in Artificial Intelligence. *Dædalus*, pages 297-311.

McCarthy J. 1992. Overcoming an unexpected obstacle. manuscript.

McCarthy J. 1995. Situation calculus with concurrent events and narrative. manuscript.

Miller R. and Shanahan M. 1994. Narratives in the Situation Calculus. Journal of Logic and Computation, 4(5):513–530.

Provetti A. 1996. Hypothetical Reasoning: From Situation Calculus to Event Calculus. In *Computational Intelligence Journal*, 12(3). Forthcoming.

Pinto J. and Reiter R. 1993. Temporal reasoning in logic programming: A case for the situation calculus. In *Proc. of 10th International Conference in Logic Programming*, pages 203–221.

Shanahan M, 1995. A Circumscriptive Calculus of Events. Artificial Intelligence, 75(2).