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

September 25, 1995

Abstract

The specification language $\mathcal{L}$ has been defined for action specification in both narrative assimilation and planning. The syntax of $\mathcal{L}$ allow expressing actual and hypothetical situations, observations of the truth-value of fluents, and observations of actual occurrences of actions. The corresponding entailment relation can formalize various types of common-sense reasoning about actions and their effects not captured by previous formalisms. In this paper we present a translation from $\mathcal{L}$ theories to Nested Abnormality theories, a novel form of circumscription. The translation is shown correct and complete wrt $\mathcal{L}$ semantics. As a result, equivalences between $\mathcal{L}$ theories and other formalisms for actions can be studied in the common ground of classical logic. Moreover, the use of nested circumscription helps clarifying which assumptions are embodied in the definition of $\mathcal{L}$ and how they can be removed in places.

Keywords Reasoning about actions. Formal models of actions. Narrative assimilation. Nested circumscription.

1 Introduction

The action description language $\mathcal{L}$ was introduced in [BGP95b] to incorporate narratives and actual situations (where actual situations are interpreted as actions that are part of the evolution) into the action description language $\mathcal{A}$ [GelLif92] that only allowed hypothetical situations. [BGP95b] contained the syntax and semantics of $\mathcal{L}$ with a sound translation to logic programs.

Although, narratives were earlier formalized in isolation using event calculus [KowSer86], only recently several proposals have been put forward that formalize narratives together with hypothetical situation [McC95, PinRei93, MilSha94, Sha95, Pro96]. Most of them [McC95, PinRei93, MilSha94, Sha95] have used situation calculus together with circumscription.

In this paper we present a translation of $\mathcal{L}$ to nested abnormality theories (NAT's) [Lif95], a new approach to the use of circumscription for representing knowledge. Although, the nesting of blocks in NAT's may in the first glance suggest loss of declarativeness and elaboration tolerance [McC88], we believe that it makes it easier to represent knowledge, particularly in the action domain (See [Lif95, KarLif94, GKL95] 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 NAT's for knowledge representation. Besides that our result will make it easier to compare the entailment of $\mathcal{L}$ to that of the formalizations in [PinRei93, MilSha94, Sha95].

2 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 with F.

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

$$A \textbf{ causes } F \textbf{ if } P_1, \ldots, P_n \tag{1}$$

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 effect law as A **causes** F.

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

[1] [Sha95] has a discussion related to this. He is concerned with formalizations where minimization of change do not interfere with minimization of other knowledge.

$$F \ \textbf{at} \ S \tag{2}$$

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

$$\alpha \ \textbf{occurs\_at} \ S \tag{3}$$

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

$$S_1 \ \textbf{precedes} \ S_2 \tag{4}$$

where S_1 and S_2 are situations. It states that situation S_2 occurred after 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.

2.1 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 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 a 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 [BGP95b], which precisely specifies the sets of acceptable conclusions which can be reached from such descriptions and assumptions (a)-(e).

2.2 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[2].

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[3].

$\Psi([\,])$ 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 wrt 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 effect 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:

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

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

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

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

We say that Ψ is a causal model of D if it satisfies all the effect 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 is called a situation assignment of $\mathcal{S}$ if it satisfies the following properties:*

[2] Please Refer to the papers introducing $\mathcal{L}$, i.e. [BGP95b], for motivations and discussion of properties of $\mathcal{L}$ semantics.

[3] 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.

1. $\Sigma(S_0) = [\,]$;

2. $\forall S_i \in \mathcal{S}.\ \Sigma(S_i) < \Sigma(S_N)$.

where $<$ *stands for the usual prefix relation on strings.* □

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. $(\alpha$ **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*[4] *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*.* □

[4] 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 $A_j = B_{i_j}$.

3 Overview of Nested Circumscription

Nested Abnormality Theories (NATs) are a novel circumscription ([McC86, Lif94]) technique introduced in [Lif95]. NATs allow the use of several abnormality predicates to specify a body of common-sense knowledge without their circumscription conflicting. Lifschitz introduces the notion of Block: a set of axioms $A_1, \ldots, A_n$—possibly containing the abnormality predicate Ab— "describes" a set of predicates/constants $C_1, \ldots, C_m$, which are allowed to vary during circumscription of Ab. The notation for such a theory is

$$C_1, \ldots, C_m \;:\; A_1, \ldots, A_n. \tag{5}$$

where each A_i may itself be a block of form (5). A block may be viewed as a set of axioms describing a collection of predicates/constants, possibly depending on other descriptions in embedded blocks. Block (5) can be expressed in terms of the circumscription operator as follows:

$$\text{CIRC}[A_1 \;\wedge \ldots \;\wedge A_n; Ab; C_1, \ldots, C_m]. \tag{6}$$

In addition to NATs, Lifschitz ([Lif95]) introduces the idea of replacing the predicate *"Ab"* by an existentially quantified variable. Noticing that Ab plays an auxiliary role and that the interesting consequences of the theory are those which *do not* contain Ab, Ab is replaced by an existentially quantified variable. To put it differently, if $F(Ab)$ denotes (6), and *"ab"* is a variable of the same arity as Ab, we are interested in the consequences of the sentence $\exists ab F(ab)$. The effect of this modification is that abnormality predicates become local to the block where each of them is used.

3.1 Syntax and semantics of nested abnormality theories

The following definitions are from [Lif95]. Let L be a second order language which does not include Ab. For every natural number k, let L_k be the language obtained by adding the k-ary predicate constant Ab to L. $\{C_1, ..., C_m \;:\; A_1, ..., A_n\}$ is a *block* if each $C_1, \ldots, C_m$ is a predicate or a function constant of L, and each $A_1, \ldots, A_n$ is a formula of L_k or a block.

A *Nested Abnormality Theory* is a set of blocks. The semantics of NATs is characterized by a mapping φ from blocks into sentences of L. If A is a formula of language L_k, φA stands for the universal closure of A, otherwise

$\varphi\{C_1, ..., C_m \;:\; A_1, ..., A_n\} \;=\; \exists ab F(ab)$
where

$F(Ab) = \text{CIRC}[\varphi A_1 \wedge ... \wedge \varphi A_n; Ab; C_1, ..., C_m]$
For any NAT T, φT stands for $\{\varphi A | A \in T\}$. A *model* of T is a model of φT in the sense of classical logic. A *consequence* of T is a sentence ϕ of language L that is true in all models of T.

4 From $\mathcal{L}$ to NAT

4.0.1 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.

4.0.2 Predicate definitions

$holds(F, \alpha)$	fluent F is true after performing α from the init. situation;
$causes^+(A, F, \alpha)$	action A performed in the state reached by α makes F true;
$causes^-(A, F, \alpha)$	action A performed in the state reached by α makes F false;
$sit\_map : \mathcal{S} \mapsto \mathcal{A}^*$	maps a sit. 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 the sequence obtained by concatenating α_1 and α_2.

4.0.3 Framework assumptions

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.

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

$$
\begin{array}{l}
\{\ min\ sequence\ :\\
\quad sequence(\epsilon)\\
\quad action(a)\ \wedge sequence(\alpha) \supset sequence(a \circ \alpha)\\
\\
\quad \{min\ action\ : action(A_1), \ldots, action(A_n), UNA[\mathcal{A}]\}\\
\\
\quad \{min\ fluent\ : fluent(F_1), \ldots, fluent(F_m), UNA[\mathcal{F}]\}\\
\\
\quad \{min\ situation\ :\ situation(S_1), \ldots, situation(S_l), UNA[\mathcal{S}]\}\\
\}
\end{array}
$$

where, of course, 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.

4.0.4 Translation of Facts

The atomic facts (2), (3) and (4) are translated to $at(F, S)$, $occurs(A_n \circ \ldots A_1 \circ \epsilon, S)$, and $precedes(S_1, S_2)$, respectively.

Facts obtained by propositionally combining atomic facts are translated in the obvious way and are denoted by $FACTS$.

4.1 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 Section 4.2.

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 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. Prefix-equal is a particular case of subsequence. 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. $\alpha \cdot \beta = \beta\alpha$:

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

The block $B_{concatenate}$ defines the predicate *concatenate.*

4.2 The resulting NAT

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

$$T(D) =$$

$$\{ \ sit\_map \ :$$

$$
\begin{array}{rl}
(*_1) & sit\_map(S_0) = \epsilon \\
(*_2) & prefix\_eq(sit\_map(s), sit\_map(S_N)) \\
(\star) & \underline{subsequence(\alpha, sit\_map(S_N)) \supset ab(\alpha)} \\
& prefix(\alpha_1, \alpha_2) \equiv [prefix\_eq(\alpha_1, \alpha_2) \ \wedge \neg prefix\_eq(\alpha_2, \alpha_1)]
\end{array}
$$

$$
\left.
\begin{array}{l}
at(f, s) \equiv h(f, sit\_map(s)) \\
occurs(\alpha_1, s) \equiv concatenate(sit\_map(s), \alpha_1, \alpha_2) \ \wedge prefix\_eq(\alpha_2, sit\_map(S_N)) \\
preceeds(s_1, s_2) \equiv prefix(sit\_map(s_1), sit\_map(s_2)) \\
FACTS
\end{array}
\right\} = \tau(D_f)
$$

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

$$
\left.
\begin{array}{l}
\{ min \ prefix\_eq \ : \\
\qquad sequence(\alpha) \supset prefix\_eq(\alpha, \alpha) \\
\qquad prefix\_eq(\alpha, \alpha_1) \supset prefix\_eq(\alpha, a \circ \alpha_1) \\
\}
\end{array}
\right\} = B_{prefix\_eq}
$$

$$
\left.
\begin{array}{l}
\{ min \ subsequence \ : \\
\qquad subsequence(\epsilon, \epsilon) \\
\qquad subsequence(\alpha, \alpha_1) \supset subsequence(\alpha, a \circ \alpha_1) \\
\qquad subsequence(\alpha, \alpha_1) \supset subsequence(a \circ \alpha, a \circ \alpha_1) \\
\}
\end{array}
\right\} = B_{subsequence}
$$

$$
\left.
\begin{array}{l}
\{ min \ concatenate \ : \\
\qquad sequence(\alpha) \supset concatenate(\alpha, \epsilon, \alpha) \\
\qquad concatenate(\alpha, \beta, \gamma) \supset concatenate(\alpha, a \circ \beta, a \circ \gamma) \\
\}
\end{array}
\right\} = B_{concatenate}
$$

Framework axioms

$$\}$$

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 Section 2.2. The axioms in $SC(D_l)$ encode Definition 1. The models of $SC(D_l)$ together with Framework axioms corresponds 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 $(\star)$. *This elegant encoding is one of the main contribution of this paper.* It takes advantage of the fact that if $\Sigma(S_N)$ is a subsequence of $\Sigma(S'_N)$ then the set of subsequences of $\Sigma(S_N)$ is a subset of the set of subsequences of $\Sigma(S'_N)$. It suggests a general way of minimizing w.r.t. any arbitrary ordering $\leq$ between objects: "To minimize w.r.t. $\leq$ between objects of set S so that the selected model has only a minimal object, we need to find a mapping F from S such that for any $o_1, o_2 \in S$, $o_1 \leq o_2$ iff $F(o_1) \subseteq F(o_2)$." By encoding this mapping F and defining ab as $ab(X)$ iff $X \in F(o_i)$; we can ensure that by circumscribing ab we will be selecting the model that contains a minimal (w.r.t. $\leq$) object only.

5 Examples

Example 1 (discovering action occurrences) *Consider the following domain description and its translation:*

$$D_1 = \begin{cases} A \textbf{ causes } F \\ \neg F \textbf{ at } S_0 \\ F \textbf{ at } S_1 \\ S_0 \textbf{ precedes } S_1 \end{cases} \qquad FACTS = \begin{cases} \neg at(F, S_0) \\ at(F, S_1) \\ precedes(S_0, S_1) \end{cases}$$

The block defining $causes^+$ in the subtheory $SC(D_{1l})$ will contain only the fact $causes^+(A, F, \alpha)$, and the subblock 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:

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

Example 2 (implying new 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 subblock defining $causes^-$ *is empty, and the subblock defining* $causes^+$ *contains the fact*

$h(P, \alpha) \supset causes^+(A, F, \alpha)$.

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

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

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

6 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.

Lemma 1 *(Causal Equivalence) For every* Ψ *causal model of* D_l *there exists a* $\mathcal{M}_{SC}$ *model of* $SC(D_l) \cup$ *Framework Axioms such that: for all fluents* f *and sequences of actions* α[5]

$$f \in \Psi(\alpha) \ \Leftrightarrow\ \mathcal{M}_{SC} \models h(f, \alpha). \tag{7}$$

For every $\mathcal{M}_{SC}$ *model of* $SC(D_l)$ *there exists a causal model* Ψ *of* D_l *such that (7) holds.* □

Lemma 2 *(Equivalence of the time line) For any domain description D, an interpretation* $\langle \Psi, \Sigma \rangle$ *is a model of D iff there exists a model* $\mathcal{M}$ *of* $T(D)$ *such that: for all situations s and sequences of actions* α

$$\Sigma(s) = \alpha \ \Leftrightarrow\ \mathcal{M} \models sit\_map(s) = \alpha \tag{8}$$

□

Lemma 3 (main) *(equivalence of models) For any domain description D, if interpretation* $\langle \Psi, \Sigma \rangle$ *is a model of D then there exists a model* $\mathcal{M}$ *of* $T(D)$ *such that for every atomic fact* ϕ *in the language of D:*

$$\langle \Psi, \Sigma \rangle \models_{\mathcal{L}} \phi \ \Leftrightarrow\ \mathcal{M} \models \tau(\phi) \tag{9}$$

and if $\mathcal{M}$ *is a model of* $T(D)$ *the there exists* $\langle \Psi, \Sigma \rangle$ *model of D such that (9) holds.* □

[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 it stands for the term $A_m \circ \ldots \circ A_1 \circ \epsilon$.

Theorem 1 (main) *(Equivalence) For any domain description D and fact ϕ in the language of D:*
$D \models_{\mathcal{L}} \phi \;\Leftrightarrow\; T(D) \models \tau(\phi)$ □

References

[BGP95b] C. Baral, M. Gelfond and A. Provetti. Representing Actions - I: Laws, Observations and Hypothesis. Submitted for publication to the *The Journal of Logic Programming*, special issue on logic programming for actions. Also appears in the Working Notes of AAAI Spring Symposium *Extending Theories of Actions: Formal Theories and Practical Applications.* Stanford, March 1995.

[GelLif92] M. Gelfond and V. Lifschitz. Representing actions in extended logic programs. In *Joint International Conference and Symposium on Logic Programming*, pages 559–573, 1992.

[GKL95] E. Giunchiglia, G. Kartha and V. Lifschitz. Actions with indirect effects (Extended Abstract). Working Notes of AAAI Spring Simposium *Extending Theories of Actionss: Formal Theories and Practical Applications.* Stanford, March 1995.

[KarLif94] G. Kartha and V. Lifschitz. Actions with indirect effects (preliminary report). In *KR 94*, pages 341–350, 1994.

[KowSer86] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[Lif95] V. Lifschitz. Nested Abnormality Theories. Artificial Intelligence, 1995.

[Lif94] V. Lifschitz. Circumscription. Handbook of Logic in Artificial Intelligence and Logic Programming, 1994.

[McC86] J. McCarthy. Applications of circumscription to formalizing common sense knowledge. Atificial Intelligence, 26(3):89–116, 1986.

[McC88] J. McCarthy. Mathematical logic in Artificial Intelligence. *D≠dalus*, pages 297-311, 1988.

[McC92] J. McCarthy. Overcoming an unexpected obstacle. manuscript, 1992.

[McC95] J. McCarthy. Situation calculus with concurrent events and narrative. manuscript, 1992.

[MilSha94] R. Miller and M. Shanahan. Narratives in the Situation Calculus. Journal of Logic and Computation, 4(5):513–530, 1994.

[Pro96] A. Provetti. Hypothetical Reasoning: From Situation Calculus to Event Calculus. To appear on *Computational Intelligence Journal*, 12(3), 1996.

[PinRei93] J. Pinto and R. Reiter. Temporal reasoning in logic programming: A case for the situation calculus. In *Proceedings of 10th International Conference in Logic Programming, Hungary*, pages 203–221, 1993.

[Sha95] M. Shanahan. A Circumscriptive Calculus of Events. Artificial Intelligence, 75(2), 1995.