Formalizing Narratives using Nested Circumscription and Value Minimization

Chitta Baral, Alfredo Gabaldon [1]

Dept. of Computer Science, University of Texas at El Paso, El Paso, Texas 79968

Alessandro Provetti [2]

C.I.R.F.I.D., Università di Bologna, Via Galliera, 3. I-40121 Bologna, Italy

Abstract

Representing and reasoning about narratives together with the ability to do hypothetical reasoning is important for agents in a dynamic world. These agents need to record their observations and action executions as a narrative and at the same time, to account for exogenous changes in the world, need to make plans (or re-plan) from the *current* situation. The early formalisms that reasoned about actions and events did one or the other. For example, while the original situation calculus was meant for hypothetical reasoning and planning, the event calculus was more appropriate for narratives. Recently, there have been some attempts at developing formalisms that do both. Independently, there has also been a lot of recent research in reasoning about actions using circumscription. Of particular interest to us is the research on using high-level languages and their logical representation using nested abnormality theories (NATs) – a form of circumscription with blocks that make knowledge representation modular. *In this paper we formalize narratives together with hypothetical reasoning using NATs. We initially use the high level language $\mathcal{L}$, and then extend it to allow concurrent actions. In the process we study several knowledge representation issues such as filtering, and restricted monotonicity with respect to NATs. Besides, our formalization using NATs also involves the notion of minimizing functions; the first non-trivial use of this concept. Finally, we compare our formalization with other approaches, and discuss how our use of NATs makes it easier to incorporate other features of action theories, such as constraints, to our formalization.*

[1] {chitta,alfredo}@cs.utep.edu
[2] provetti@cirfid.unibo.it

Preprint submitted to Elsevier Science *24 June 1997*

Contents

1 Introduction

A narrative is a possibly incomplete set of observations about the world in terms of what actions/events occured and the value of fluents at different instants in the past. Initial formulations of narratives, such as Kowalski and Sergot's event calculus [KowSer86] and Allen's temporal logic [All84], were concerned about inferring values of fluents at time instants other than what is explicitly given in the narrative and also possibly abduce occurrences of actions not explicitly mentioned in the narrative. These formulations were not concerned[3] with hypothetical and/or counterfactual reasoning about values of fluents in a possible future world reached by executing a sequence of actions. Such hypothetical reasoning is important from the point of view of planning,

[3] Later it was shown that some of these formalisms can use abduction in the meta-level to do hypothetical reasoning and planning. But, in [Rei96] Reiter argues why it is better to do hypothetical reasoning at the object level. Besides, the action descriptions used in these formalisms were restrictive in the sense that they did not allow features such as constraints.

where an agent needs to construct a plan – normally a sequence of actions, to achieve a particular goal.

The formalism of Situation calculus [MH69] has been normally used for hypothetical reasoning about actions and forms the basis of classical planning. But in its original form it does not allow narratives; the only actual observations that can be expressed in it are about values of fluents in the initial situation.

Recently, researchers [MilSha94,KakMil96,PinRei93,McC95,BarGabPro96a,Pro96] have realized the importance of having a formalism that allows both hypothetical reasoning about effects of actions and narratives. Such a formalism is necessary to formulate planning and execution of actions of an agent in a dynamic environment, where exogenous actions may occur. The agent has to record observations about occurrences of actions (both its own action executions, and the exogenous actions), and sometimes infer them from observations about values of fluents, which form a narrative. But, the agent also has to make plans and more importantly may have to dynamically revise its plans or construct new plans when faced with exogenous events. These plans are not from the initial situation as in Situation calculus, but from the *current* situation. Although the terms 'dynamic planning', 'planning with execution', and 'reactive planning' have been used in the planning community, a formalism that allows both narratives and hypothetical reasoning is necessary to form the backbone of such planners – the role situation calculus plays in classical planners.

Miller and Shanahan [MilSha94] and Pinto and Reiter [PinRei93] were perhaps the first ones to develop formalisms that allow both narratives and hypothetical reasoning. Both formalisms made many important contributions. For example, the first one showed how situation calculus can be extended to incorporate narratives, and the second one introduced the concept of *actual situations* – the situations that are not hypothetical. But, both formalisms have several drawbacks. For example, Miller and Shanahan only allow fluent facts about the initial state, and require that all action occurrences be explicitly stated. On the other hand, Pinto and Reiter [PinRei93] do 'premature minimization'[4] of occurrences, which prevents them from making plans. As a result, both approaches lack the re-planning ability of an agent who after making a plan of packing his suitcase and driving the car to the airport, and doing the packing, observes his car with a demobilizing dent. The agent did not see the exogenous action that made the dent so it cannot put a corre-

[4] This term was coined by Reiter in [Rei96] where he discussed this drawback. The later papers of Reiter and Pinto [Rei96,Pin96] do not have this drawback but to prevent it they abandon minimization of action occurrences. We do a detailed comparison of our approach with theirs in Section 11.5.

sponding action occurrence to its narrative, but it should be able to infer such an occurrence from its observation that the car has a dent.

In this paper we focus on a more general formalism that allows both narratives and hypothetical reasoning and overcomes many of the limitations of the earlier proposals, such as allowing fluent facts about non-initial states and inferences of action occurrences not explicitly in the narrative. Our formulation is done using a novel form of circumscription called *Nested Abnormality Theories* (NATs) [Lif95].

One important aspect of our approach is that unlike most of the earlier formalizations of narratives [MilSha94,PinRei93] that were done directly in a logical language, our formalization is grounded to the high level language $\mathcal{L}$ developed by Baral, Gelfond and Provetti [BarGelPro95b] (Baral and Provetti are co-authors of this paper). In other words, *in the first half of the paper we give a translation of descriptions in $\mathcal{L}$ to nested abnormality theories and show their equivalence. Later, we extend $\mathcal{L}$ to allow concurrent actions and give a translation of descriptions in this extended language to nested abnormality theories.* Recently, many works [GL92a,San92,KakMil96] formalizing actions and effects have used formalizations in a high level language. Such an approach is useful for several reasons, including that: they allow a better understanding of the underlying ontological principles of reasoning about actions; they serve as a means to write formal specifications of the domain and the assumptions of the knowledge representer thus making the issue of soundness become a precise mathematical question; and they facilitate the comparison of different theories of action. (An example of the use of high level description languages for comparing different formalisms is [Kar93], where Kartha uses translations from $\mathcal{A}$ into three different formalisms to prove their equivalence.) In following the approach of having descriptions in a high level language with an independent semantics and a corresponding theory using circumscription we agree with Hoare's view [Hoa87] on software engineering when he says:

> "For specifications of the highest quality and importance I would recommend complete formalization of requirements in two entirely different styles, together with a proof that they are consistent or even equivalent to each other. ... A language for which two consistent and complementary definitions are provided may be confidently taken as a secure basis for software engineering."

One major advantage of our formalization of narratives and hypothetical reasoning using nested abnormality theories is that it allows easy incorporation of additional features about action theories, such as constraints, which have been formalized – in the absence of narratives, using nested abnormality theories by Giunchiglia, Kartha, and Lifschitz [KarLif94,GiuKarLif95]. Also, although the nesting of blocks in NATs may at first glance suggest loss of declarativeness and elaboration tolerance [McC88], we believe it makes it easier to represent knowledge, particularly in the action domain (See [Lif95,KarLif94,GiuKarLif95] for more on this). This is because we can develop blocks that represent meaningful structural units and use the blocks in other units without worrying about undesired interactions.Besides, given a NAT representing a particular theory, a change in a particular aspect of the theory can be achieved by only changing the corresponding block in the NAT without any non-local surgery of the rest of the theory. (In Section 9 we show how we can add constraints to our theory by only changing the block that encodes the transition function.) To some extent, using "flat" circumscription to formalize a large theory is like writing a large program without using subroutines. We believe that the simplicity of our translation of $\mathcal{L}$ (as presented in this paper) demonstrates the usefulness of NATs for knowledge representation.

An additional and important contribution of this work is that *the formalization presented here constitutes a full, and non-trivial example of the use of "value minimization"* [BarGabPro96b], a technique developed by Baral et. al. for minimizing the value of a function in a circumscriptive theory. This technique is used in our formalization to capture the assumption that the only action occurrences which can be derived from a set of observations are those which are necessary to explain those observations.

In reasoning with narratives, since we allow facts about non-initial states and the possibility that actions not specified in the narrative may have occurred, we need some form of abduction to entail occurrence of such actions and values of fluents at time points that are not explicitly stated in the narrative. Unlike [KakMil96], where a particular kind of abductive reasoning (through explanations to restore consistency) is done at the meta level, we use filtering [San89] at the object level.

Our formalization in this paper is thus of independent interest from the point of view of knowledge representation and nonmonotonic logics, since it illustrates the usefulness of NATs in terms of ease of representation, the use of value minimization of functions in knowledge representation, the formalization of filtering in NATs, and the restricted monotonicity property of NATs.

1.1 Organization of the rest of the paper

We start with an overview of the high level language $\mathcal{L}$ from [BarGelPro95b,BarGelPro97] in Section 2 and an overview of NATS in Section 3. We then give a translation of domain descriptions in $\mathcal{L}$ to NATs in Section 4 and illustrate the NAT characterization with respect to some examples in Section 6. In Section 5 we discuss value minimization of functions and its role in our translation described in Section 4. In Section 7 we formally relate the entailment in $\mathcal{L}$ to the entailment in our translation. In Section 8 we extend $\mathcal{L}$ to allow concurrent actions, give a translation of domain descriptions in this extended language to NATs, and relate the entailment relations in both. In Section 9 we show how our formulation in terms of NATs can be easily extended to take into account constraints, and in Section 10 we discuss the restricted monotonicity property of our formulation in NATs and show the relation between abductive reasoning, filtering and restricted monotonicity. In Section 11 we compare our work with earlier work on combining narratives and hypothetical reasoning – particularly by Miller, Shanahan, Pinto, Reiter, and McCarthy. Finally, we put all the proofs in the Appendix.

2 Overview of $\mathcal{L}$

The high level language $\mathcal{L}$ was developed by Baral, Gelfond and Provetti [BarGelPro95b,BarGelPro97] to allow representation of and reasoning with narratives together with hypothetical reasoning. The authors give the syntax and semantics of this language and give a translation to logic programs and Prolog for a subclass of this language. Here, we start with a translation of domain descriptions in $\mathcal{L}$ to NATs, and then extend it to allow concurrent actions. *We would like to make it clear that the contribution of this paper is not the development of $\mathcal{L}$, but the formalization of $\mathcal{L}$ in NATs, which allows easy integration of additional features such as constraints; and the study of our NAT formalization in terms of its impact to knowledge representation in general.*

We now give a short overview of the syntax and semantics of $\mathcal{L}$. We do present some motivations for the choice of constructs in this language, but additional detailed motivations can be found in [BarGelPro97].

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 usually denoted by letters F, P and Q with indexes. $\neg\neg F$ will be equated to F. For a fluent F, $\overline{\neg F} = F$, and $\overline{F} = \neg F$.

There are two kinds of propositions in $\mathcal{L}$ called causal laws and facts. Causal laws in $\mathcal{L}$ are the same as effect propositions in $\mathcal{A}$ [GL92a], the high level language introduced by Gelfond and Lifschitz for hypothetical reasoning. A *causal 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 where $P_1, \ldots, P_n$ are true." If $n = 0$, we write the causal law as A **causes** F.

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

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

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 was observed to be true in situation S." An agent in a dynamic world can update its knowledge about the world by adding such fluent facts (based on its observations) to its database.

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 a sequence follow each other immediately). An agent in a dynamic world can update its knowledge about action occurrences – both exogenous actions, and actions executed by the agent itself, by adding such occurrence facts to its database.

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 the domain was in situation S_2 after being in situation S_1. Precedence facts saying a situation precedes the initial situation, e.g., S_1 **precedes** S_0, are of course forbidden. The agent can use such precedence facts to record the temporal ordering between its observations about fluent facts and occurrence facts.

Since propositions of the type (1) express general knowledge about effects of actions, they 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. A collection of laws and facts is called *domain description*. The sets of laws and facts of a domain description D will be denoted by D_l and D_f respectively. Domain descriptions are required to satisfy the following properties:

– its propositions do not contain the situation constant S_N, and
– for every situation there is at most one occurrence fact.

The first property is very important and allows nonmonotonic interpretation of the situation constant S_N, which represents the situation NOW. The purpose of the second property is to prohibit concurrent actions in $\mathcal{L}$.

Since our agent is not God, it may miss certain happenings – in terms of both action occurrences and fluent facts, in the world. But since our agent is smart – at least that is what we would like our agent to be, it can reason with incomplete narratives and can make revisable and nonmonotonic conclusions about observations it might have missed. (Unlike in [MilSha94,KakMil96] we would like our agent to make this conclusion in the object language itself, not through meta level reasoning.) The following example further illustrates our point.

Example 1 *(discovering occurrences)* Consider the following version of the stolen-car example: we know that initially we have a car (in the garage). However, at a later instance of time we observe that we no longer have the car. We also know that after our car gets stolen then we will no longer have the car. The following is a description of this story in $\mathcal{L}$:

$$D_1 = \begin{cases} steal(car) \textbf{ causes } \neg has(car) \\ has(car) \textbf{ at } S_0 \\ \neg has(car) \textbf{ at } S_1 \\ S_0 \textbf{ precedes } S_1 \end{cases}$$

Intuitively, from this description we as a smart agent would like to conclude that action $steal(car)$ must have occured in situation S_0 thus explaining the fact that in that situation we had a car but in situation S_1 we no longer have it. In other words, we "abduce" that action $steal(car)$ occurred in the initial situation causing $has(car)$ to become false. It is clear that in reaching this conclusion we are making some assumptions about the domain. We are, for example, assuming that action $steal(car)$ is the only action that may cause $has(car)$ to become false. Otherwise the conclusion would be wrong since the fact could be explained by the occurrence of another action. All the assumptions embodied in domain descriptions in $\mathcal{L}$ are discussed in the subsection below. □

Note that fluent facts about situations different from the initial situation are allowed and, as it is shown in this example, a formalization of the semantics of $\mathcal{L}$ must be able to capture some form of *abductive reasoning* to be able to abduce the explanations to these observations.

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

Domain descriptions in $\mathcal{L}$ are used in conjunction with the following informal assumptions which restrict the language so that the agent can do intelligent (perhaps hasty, and thus revisable) reasoning.

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 in $\mathcal{L}$. We now present the semantics of domain descriptions in $\mathcal{L}$ as defined in [BarGelPro97], which precisely specifies the sets of acceptable conclusions which can be reached from such descriptions and assumptions (a)-(e).

2.2 Semantics of $\mathcal{L}$

In $\mathcal{L}$, states of the world are represented by sets of the fluents which are true. Actions executed in a particular state may add and remove such fluents according to causal laws. Thus, the set of all possible evolutions of the world described by the causal laws can be represented by a transition diagram with states corresponding to states of the world, and transitions labeled by actions.

Satisfying the facts in a domain description intuitively consists of selecting a state as the initial one, and a path in the diagram describing the actual evolution of the domain.

A *state* is 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. [5]

$\Psi([\,])$ is called the initial state of Ψ. A partial function Ψ serves as an interpretation [6] of the laws of D. If a sequence 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 with respect to σ is defined as usual.

To better understand the role Ψ plays in interpreting domain descriptions let us discuss its role in defining *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)-(c), and (e).

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

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

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

Definition 1 (Causal interpretation) *A causal interpretation Ψ satisfies*

[5] 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.

[6] In $\mathcal{A}$ [GL92a], laws are interpreted using a transition function from states and actions to states, and this function together with the interpretation of the initial state form the interpretation of the domain description. Because of our restricted syntax, we could use this formulation, but decided to stick to the original semantics of $\mathcal{L}$, which at that time was chosen to accommodate future extensions involving triggers and actions with non-deterministic effects.

causal laws of D if for any sequence $\alpha \circ A$ from the language of D,

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

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

Causal models are uniquely determined by their initial values, i.e., for any two causal models Ψ_1 and Ψ_2 of a domain description D, if $\Psi_1([\,]) = \Psi_2([\,])$ then $\Psi_1 = \Psi_2$.

We are now ready to discuss how observations are interpreted in $\mathcal{L}$. 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:*

(i) $\Sigma(S_0) = [\,]$;
(ii) $\forall s_i \in \mathcal{S}.\ \Sigma(s_i)$ *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.* □

Now we can define the truth of facts of D with respect to an interpretation M. Facts which are not true in M will be called *false* in M.

Definition 4 *For any interpretation $M = (\Psi, \Sigma)$,*

(i) $(F$ **at** $S)$ *is true in M (or satisfied by M) if F is true in $\Psi(\Sigma(S))$;*
(ii) $(\alpha$ **occurs_at** $S)$ *is true in M if $\Sigma(S) \circ \alpha$ is a prefix of the actual sequence of M;*
(iii) $(S_1$ **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 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." (A similar assumption is used by Pinto and Reiter in [PinRei93,Pin94].) This is done by imposing a minimality condition on situation assignments of $\mathcal{S}$, which leads to the following definition.

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

(i) *Ψ is a causal model of D;*
(ii) *facts of D are true in M;*
(iii) *there is no other interpretation $N = (\Psi, \Sigma')$ such that N satisfies condition (ii) and $\Sigma'(S_N)$ is a subsequence[7] of $\Sigma(S_N)$.* □

It is important to note that we are only minimizing the actual action occurrences between the initial situation and the current situation. No such minimization is done about the future. This is instrumental in our formalization not falling into the trap of 'premature minimization' as discussed in [Rei96].

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 $D \models \phi$) iff ϕ is true in all models of D.* □

We now formally show how we capture the intuition discussed in Example 1 in our semantics.

Example 2 Consider domain description D_1 from Example 1. By fact

$$has(car) \ \mathbf{at} \ S_0$$

we have that $has(car) \in \Psi(\Sigma(S_0))$. By the causal law

$$steal(car) \ \mathbf{causes} \ \neg has(car)$$

we have $has(car) \notin \Psi([steal(car), \ldots, steal(car)])$ for any sequence of one or more actions $steal(car)$. It is easy to see that $\Sigma(S_N) = \Sigma(S_1) = [steal(car)]$ is the minimal sequence satisfying the facts in the domain description. Thus, $\Sigma(S_0) \circ steal(car)$ is a prefix of $\Sigma(S_N)$. Therefore, D_1 entails $steal(car)$ **occurs_at** S_0.

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

2.3 Planning from the current situation

Planning from the current situation is necessary for an agent in a dynamic environment where exogenous events may occur, and our agent may need to revise its plan and construct a new plan from the current situation. To be able to do such plans, $\mathcal{L}$ has a construct called *hypotheses* which is of the following form:

$$(\neg)F \textbf{ after } [A_1, \ldots, A_n] \tag{5}$$

Intuitively, the above hypothesis means that F holds (does not hold) after actions $A_1, \ldots, A_n$ are executed in S_N.

Hypotheses are not part of a domain description, rather they are part of the query language.

We say a hypothesis of the form (5) is true in a model (Ψ, Σ) of a domain description D if $(\neg)F$ holds in

$$\Psi(\Sigma(S_N) \circ A_1 \circ \ldots \circ A_n)$$

and, D entails a hypotheses if it is true in all models of D.

We can now define a notion of a plan from the current situation.

Definition 7 ([BarGelPro97]) *Let D be a domain description and G be a set of fluent literals. A sequence α of actions is a plan for achieving a goal G if $D \models f$ **after** α for every fluent literal $f \in G$.*

In the following example we use $\mathcal{L}$ to show the planning and re-planning done by an agent in a dynamic world.

Example 3 Suppose our agent has the following initial knowledge, in terms of the various actions and their effects and the values of fluents in the initial situation.

$$\left.\begin{array}{l}
load\ \textbf{causes}\ \ loaded \\
unload\ \textbf{causes}\ \ \neg loaded \\
set_target\ \textbf{causes}\ \ target_set \\
shoot\ \textbf{causes}\ \ target_hit\ \textbf{if}\ \ loaded, target_set \\
loaded\ \textbf{at}\ \ S_0 \\
\neg target_set\ \textbf{at}\ \ S_0
\end{array}\right\} = D_{\mathrm{hyp0}}$$

Given the agent's goal to have the *target_hit*, it develops the plan $[set\_target, shoot]$ by checking that indeed

$$D_{\mathrm{hyp0}} \models target\_hit\ \textbf{after}\ \ [set\_target, shoot]$$

holds. Then the agent proceeds to execute the first action: *set_target*, and adds *set_target* **occurs_at** S_1 and S_0 **precedes** S_1 to the domain description. (Let us call the updated domain description D_{hyp1}.) At this point, the agent's domain description entails:

$$loaded\ \textbf{at}\ \ S_N$$

$$target\_set\ \textbf{at}\ \ S_N$$

Before executing the next action, however, it observes that the gun is no longer loaded, and updates its domain description by adding $\neg loaded$ **at** S_2 and S_1 **precedes** S_2. (Let us call the updated domain description D_{hyp2}.) It concludes that:

$$D_{\mathrm{hyp2}} \models target\_set\ \textbf{at}\ \ S_N$$

$$D_{\mathrm{hyp2}} \models \neg loaded\ \textbf{at}\ \ S_N$$

Recall that S_N always denotes the current situation. Additionally, it also concludes that:

$$D_{\mathrm{hyp2}} \models [set\_target, unload]\ \textbf{occurs\_at}\ \ S_0 \vee [unload, set\_target]\ \textbf{occurs\_at}\ \ S_0$$

The agent then notices that it can no longer continue with its original plan. This is because

$$D_{\mathrm{hyp2}} \not\models target\_hit\ \textbf{after}\ \ shoot$$

Hence, the agent proceeds to re-plan. This re-planning by the agent is carried out from the current situation, S_N, and not from the initial situation, S_0. Given its goal, the plan to achieve it, starting from the initial situation, is still $[set\_target, shoot]$. The new plan it develops is $[load, shoot]$ since

$$D_{\text{hyp2}} \models target\_hit \ \textbf{after} \ \ [load, shoot]$$

Assuming, no untoward incident takes place the agent can execute its current plan to reach its goal. □

3 Overview of Nested Circumscription

Nested Abnormality Theories (NATs) is a novel circumscription [McC86,Lif94] technique introduced by Lifschitz [Lif95]. With NATs it is possible to circumscribe several predicates each with respect to only parts of the theory of interest, as opposed to previous techniques such as parallelized and circumscription theories where the circumscription must be done with respect to all of the axioms in the underlying theory. Furthermore, all the complications arising from the interaction of multiple circumscription axioms in a theory are avoided in NATs with the introduction of blocks. A *block* is characterized by a set of axioms $A_1, \ldots, A_n$—possibly containing the abnormality predicate Ab — which "describe" a set of predicate/function constants $C_1, \ldots, C_m$. The notation for such a theory is

$$\{C_1, \ldots, C_m \ : \ A_1, \ldots, A_n\} \tag{6}$$

where each A_i may itself be a block of form (6). The "description" of $C_1, \ldots, C_m$ by a block may depend on other descriptions in embedded blocks.

Interference between circumscription in different blocks is prevented by replacing a predicate Ab with an existentially quantified variable. Lifschitz's idea is to make Ab "local" to the block where it is used, since abnormality predicates play only an auxiliary role, i.e. the interesting consequences of the theory are those which do not contain Ab. The next section contains the formal definitions of these concepts.

3.1 Syntax and semantics of NATs

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

$$F(Ab) = \mathrm{CIRC}[\varphi A_1 \wedge ... \wedge \varphi A_n; Ab; C_1, ..., C_m]$$

(Recall that $\mathrm{CIRC}[T; P; Q]$, means circumscription of the theory T, by minimizing the predicates in P, and varying the objects in Q.)

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. In this paper, as suggested in [Lif95], we use the abbreviation

$$\{C_1, \ldots, C_m, \ \min P : A_1, \ldots, A_n\}$$

to denote blocks of the form

$$\{C_1, \ldots, C_m, P : P(x) \supset Ab(x), A_1, \ldots, A_n\}$$

As the notation suggests, this type of block is used when it is necessary to circumscribe a particular predicate P in a block. In [Lif95] it is shown that

$$\varphi\{C_1, \ldots, C_m, \ \min P : A_1, \ldots, A_n\}$$

is equivalent to the formula

$$\mathrm{CIRC}[A_1 \wedge \ldots \wedge A_n; P; C_1, \ldots, C_m]$$

when each A_i is a sentence.

4 From domain descriptions in $\mathcal{L}$ to NATs

We are now ready to present our translation from $\mathcal{L}$ to NATs. We will start by formally describing some rather typical relations on sequences which we will use later in our proofs.

4.1 Relations on sequences

The relations we are interested in are prefix, subsequence and concatenate. These relations are captured by the NAT blocks $B_{\text{prefix\_eq}}$, $B_{\text{subsequence}}$, and $B_{\text{concatenate}}$, respectively, which are presented in a later section. Given the formal description of these relations, 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 prefix relation which will be denoted by $\preceq$:

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

where above and in the rest of this section by "=" we mean syntactic identity. Relation $\preceq$ captures the intuition of what it means for a sequence to be a *prefix* of another. This relation is defined by block $B_{\text{prefix\_eq}}$ in the theory.

The second relation we need to describe is subsequence: $\ll$:

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

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

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

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

Block $B_{\text{concatenate}}$ defines this relation.

4.2 The language of translation T

The language of the theory is many-sorted. The sorts are: *actions*, *fluents*, *situations*, and *sequences*. The variables for the first three sorts will be denoted by possibly indexed letters a, f and s, respectively, unless stated otherwise. Variables for sequences will be denoted by possibly indexed Greek letters α, β and γ. The language includes the elements of $\mathcal{A}$ as action constants, of $\mathcal{F}$ as fluent constants, and $\mathcal{S}$ as situation constants. It also includes the sequence constant ϵ which stands for the empty sequence and the constants in the sort *sequences* are only sequences of actions.

Sequences of actions are defined by means of a function $\circ$ from sequences and actions to sequences. Sequences of actions are then constructed by multiple application of the function $\circ$. We will use in-fix notation for $\circ$ and ignore parenthesis since doing so does not cause ambiguity. The following is an example of a sequence:

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

We have the following predicate constants:

- $At(f, s)$: with arguments of sort fluent and situation and intuitively meaning that f is true in the actual situation s;
- $Occurs(\alpha, s)$: with sorts sequence and situation, intuitively meaning that α was observed to occur in the situation s;
- $Precedes(s_1, s_2)$: both arguments of sort situation, meaning that the domain of interest has been in s_1 before being in s_2, or s_1 is temporally earlier than s_2.
- Sit_map: a function that maps situations into sequences;
- $Prefix\_eq(\alpha, \beta)$: both arguments of sort situation, this predicate captures relation $\leq$ on sequences, i.e. $\alpha \leq \beta$;
- $Prefix(\alpha, \beta)$: the strict version of $\leq$;
- $Subsequence(\alpha, \beta)$: captures relation $\ll$, i.e. $\alpha \ll \beta$;
- $Holds(f, \alpha)$: with sorts fluent and sequence, meaning that f is true in the hypothetical state resulting from executing α in the initial situation;
- $Concatenate(\alpha, \beta, \gamma)$: captures relation $\cdot$ on sequences, i.e. $\gamma = \alpha \cdot \beta$;
- $Causes^{+(-)}(a, f, \alpha)$: with sorts action, fluent and sequence, meaning that executing a in the state resulting from executing α in the initial situation makes f true(false);

4.3 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 for actions, fluents and situations are among these axioms.

$$(\forall a).a = A_1 \vee \ldots \vee a = A_l$$

$$(\forall f).f = F_1 \vee \ldots \vee f = F_m$$

$$(\forall s).s = S_0 \vee \ldots \vee s = S_N$$

$$UNA[actions],\ UNA[fluents],\ UNA[situations]$$

where by $UNA[sort]$ we mean a set of inequalities between each distinct pair of constants from $sort$, e.g. $UNA[situations]$ stands for $S_0 \neq S_1$, $S_1 \neq S_2 \ldots$ etc. In the rest of the paper we will refer to the above formulae as *framework axioms*.

4.4 Translation of facts

The atomic facts of type (2), (3) and (4) are translated into formulas $(\neg)At(F, S)$, $Occurs(A_n \circ \ldots \circ A_1 \circ \epsilon, S)$ and $Precedes(S_1, S_2)$, respectively. Non-atomic facts are translated in the obvious way. For any fact ϕ, we will use $\tau(\phi)$ to denote its translation. For the collection of facts D_f in a domain description D, the set of formulas $\tau(\phi)$ for each $\phi \in D_f$ will be denoted by $FACTS$.

4.5 The resulting NAT

We now present our translation of domain descriptions in $\mathcal{L}$ into NATs (universal quantification with the highest scope is implicit on free variables):

$$
\begin{aligned}
&T(D) = \\
&\quad \{\ Sit\_map\ : \\
&(a)\ Sit\_map(S_0) = \epsilon \\
&(b)\ Prefix\_eq(Sit\_map(s), Sit\_map(S_N)) \\
&(c)\ Subsequence(\alpha, Sit\_map(S_N)) \supset Ab(\alpha) \\
&(d)\ Prefix(\alpha,\beta) \equiv [Prefix\_eq(\alpha,\beta)\ \wedge \neg Prefix\_eq(\beta,\alpha)] \\
&\qquad \tau(D_f),\ SC(D_l) \\
&\qquad B_{\text{prefix\_eq}},\ B_{\text{subsequence}},\ B_{\text{concatenate}} \\
&\qquad \text{Framework axioms} \\
&\quad \}
\end{aligned}
$$

where:

$$
\tau(D_f) =
\begin{cases}
(e)\ At(f,s) \equiv Holds(f, Sit\_map(s)) \\
(f)\ Occurs(\alpha,s) \equiv \exists\beta[Concatenate(Sit\_map(s),\alpha,\beta)\ \wedge Prefix\_eq(\beta, Sit\_map(S_N))] \\
(g)\ Precedes(s_1,s_2) \equiv Prefix(Sit\_map(s_1), Sit\_map(s_2)) \\
FACTS
\end{cases}
$$

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

(The NATs defining $Causes^{+(-)}$ were first used by Lifschitz in [Lif95].)

$B_{\text{prefix\_eq}} =$

$\{\min Prefix\_eq$:

$Prefix\_eq(\alpha, \alpha)$

$Prefix\_eq(\alpha, \beta) \supset Prefix\_eq(\alpha, a \circ \beta)$

$\}$

$B_{\text{subsequence}} =$

$\{\min Subsequence$:

$Subsequence(\alpha, \alpha)$

$Subsequence(\alpha, \beta) \supset Subsequence(\alpha, a \circ \beta)$

$Subsequence(\alpha, \beta) \supset Subsequence(a \circ \alpha, a \circ \beta)$

$\}$

$B_{\text{concatenate}} =$

$\{\min Concatenate$:

$Concatenate(\alpha, \epsilon, \alpha)$

$Concatenate(\alpha, \beta, \gamma) \supset Concatenate(\alpha, a \circ \beta, a \circ \gamma)$

$\}$

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 blocks $B_{\text{prefix\_eq}}$, $B_{\text{subsequence}}$ and $B_{\text{concatenate}}$ define the predicates $Prefix\_eq$, $Subsequence$ and $Conactenate$ respectively. The axioms inside each of these blocks are definite clauses and in the appendix we use results from [MS89,Lif94] to show that indeed they are a correct representation of their corresponding relations. The axiom (d) defines the predicate $Prefix$ in terms of $Prefix\_eq$.
- The axioms in $SC(D_l)$ capture the meaning of causal laws. Models of $SC(D_l)$ correspond to the causal models of D. (We show this in Section 7.) Inside $SC(D_l)$ we have two blocks that minimize the predicates $Causes^+$ and $Causes^-$. Intuitively, they encode the informal assumption (c) in Sec-

tion 2.1, which says that there are no effects of actions except those specified by the causal laws. The first axiom of $SC(D_l)$ encodes the frame axiom – the assumption (a) in Section 2.1 as a first order statement, and the second and the third axioms in $SC(D_l)$, encode the effect of actions on fluents.

– The axioms (a) and (b) encode Definition 2 in a straight forward manner.

– The axioms in $\tau(D_f)$ encode Definition 4 in a straight forward manner.

– Finally, the minimality in condition (iii) of Definition 5 (which corresponds to the assumption (d) of Section 2.1) is encoded by axiom (c) plus circumscription of Ab. Intuitively, here we would like to minimize the *value of the term* $Sit\_map(S_N)$. We accomplish this through axiom (c) by minimizing Ab while varying Sit_map. A discussion on this axiom and the technique it exemplifies follows in Section 5.

– But before we move on to that, let us quickly review how the various informal assumptions described in Section 2.1 are encoded in our NAT. We already discussed the assumptions (a), (c), and (d). The assumption (b) is captured through the 'Framework axioms' and the assumption (e) is captured by the axiom (b) in $T(D)$ and our definition of sorts, which requires that $Prefix\_eq$ be a relation between action sequences with no possibility of representing concurrency.

4.6 Translation of hypotheses

Since hypotheses are not part of the domain description we did not discuss their translation so far. But we need to translate them to check if they are entailed by the translation of a domain description.

For a hypothesis of the form (5), its translation, $\tau(\phi)$ is the following:

$$(\forall\beta)Concatenate(Sit\_map(S_N), A_n \circ \ldots \circ A_1 \circ \epsilon, \beta) \supset (\neg)Holds(F, \beta)$$

5 Value minimization of functions in T

In [BarGabPro96b], we introduce the concept of value minimizing a function. That is, forcing a function to map the elements of its domain onto minimal elements of its co-domain, where the minimality criterion is with respect to an arbitrary partial order defined on the co-domain. This is completely different from all earlier formulations of circumscription where the extent of predicates (or formulas) is minimized. In the previous formulations functions are allowed to vary though.

As we mentioned in the introduction, our formalization of narrative applies this technique. In order to capture condition (iii) of Definition 5, we minimize the value of function Sit_map on term S_N, with respect to the partial order on sequences captured by predicate $Subsequence$. So this is an instance of "term minimization." Intuitively, this forces Sit_map to map S_N onto the 'minimal' possible sequence such that the facts and all the axioms remain satisfied.

By using the syntactic definition of value minimization with respect to our application, we can express the minimization of $Sit\_map(S_N)$ as follows: Let T_{subseq} stand for all the axioms of $T(D)$ except axiom (c), and let $T_{\text{subseq}}(\alpha)$ denote $T_{\text{subseq}} \wedge Sit\_map(S_N) = \alpha$. The term $Sit\_map(S_N)$ is minimized with respect to $Subsequence$ by postulating:

$$\begin{aligned} &(\forall \alpha)\ T_{\text{subseq}}(\alpha) \supset \\ &\quad \forall \alpha'[(T_{\text{subseq}}(\alpha') \ \wedge Subsequence(\alpha', \alpha)) \supset Subsequence(\alpha, \alpha')] \end{aligned} \tag{7}$$

In [BarGabPro96b] we show that value minimization of a function ϕ in a theory T with respect to an ordering $\mathcal{R}$ (which itself is defined in a block inside T) while varying the predicate/function constants Z, can be achieved by the following NAT:

$$\begin{aligned} &\{\phi, Z\ : \\ &\qquad \forall x, y.\ \mathcal{R}(y, \phi(x)) \supset Ab(x, y) \\ &\qquad T \\ &\} \end{aligned}$$

Moreover, we show that to minimize the function at only one input σ, the corresponding NAT characterization would be:

$$\begin{aligned} &\{\phi, Z\ : \\ &\qquad \forall y.\ \mathcal{R}(y, \phi(\sigma)) \supset Ab(y) \\ &\qquad T \\ &\} \end{aligned}$$

Using this result, and by observing that in our case ϕ is the function Sit_map,

$\mathcal{R}$ is the predicate $Subsequence$, and σ is S_N, we obtain the following NAT characterization of the formula (7).

$$\{Sit\_map\ :$$
$$\forall y.\ Subsequence(y, Sit\_map(S_N)) \supset Ab(y)$$
$$T_{\text{subseq}}$$
$$\}$$

Clearly, this is exactly our translation $T(D)$.

6 NAT characterization of some simple domain descriptions

In this section we illustrate our NAT characterization of domain descriptions in $\mathcal{L}$ through some examples.

Example 4 Consider again the domain description D_1 from Example 1. In its NAT translation we have:

$$FACTS = \begin{cases} At(has(car), S_0) \\ \neg At(has(car), S_1) \\ Precedes(S_0, S_1) \end{cases}$$

and the sub-block of $SC(D_{1l})$ defining $Causes^-$ will be

$$\{\min Causes^-:$$
$$Causes^-(steal(car), has(car), \alpha)$$
$$\}$$

which is equivalent to

$$(\forall a, f, \alpha).Causes^-(a, f, \alpha) \equiv a = steal(car)\ \wedge f = has(car) \tag{8}$$

The sub-block defining $Causes^+$ is empty, i.e. $Causes^+(a, f, \alpha)$ is false for all a, f, α. From this, (8), the axiom $Causes^-(a, f, \alpha) \supset Holds(f, a \circ \alpha)$, and the fact that $steal(car)$ is the only action in our language; we have that for every sequence β,

$\neg Holds(has(car), \beta)$ iff $\beta = steal(car) \circ \alpha$, for some α.

By axiom (e) in our NAT, we will have $\neg Holds(has\_car, Sit\_map(S_1))$ which means $Sit\_map(S_1) = steal(car) \circ \alpha$ for some sequence α. By value minimization and axiom (b),

$$Sit\_map(S_N) = Sit\_map(S_1) = steal(car) \circ \epsilon$$

and from this and the blocks $B_{\text{concatenate}}$ and $B_{\text{prefix\_eq}}$ we get that

$$Occurs(steal(car), S_0)$$

$\square$

We now consider a slightly different example where using the observations we can abduce truth values of fluents whose truth values are not explicitly given in the domain description.

Example 5 *(abduction of fluent-values)* Consider another scenario where there is a gun that causes Fred to die if fired at him when loaded. Suppose that Fred had been observed to be alive at some moment of time and to be dead at a later moment. Description D_2 below captures this scenario:

$$D_2 = \begin{cases} shoot \ \textbf{causes} \ \neg alive \ \textbf{if} \ loaded \\ alive \ \textbf{at} \ S_0 \\ \neg alive \ \textbf{at} \ S_1 \\ S_0 \ \textbf{precedes} \ S_1 \end{cases}$$

Let us consider the corresponding NAT, $T(D_2)$, starting from the inner blocks. Since the sub-block defining $Causes^+$ is empty we have

$$(\forall a, f, \alpha).\neg Causes^+(a, f, \alpha) \tag{9}$$

The sub-block defining $Causes^-$ contains only the axiom

$$Holds(loaded, \alpha) \supset Causes^-(shoot, alive, \alpha)$$

By minimization of $Causes^-$ in this block we have that

$$(\forall a, f, \alpha).Causes^-(a, f, \alpha) \equiv (a = shoot \ \wedge f = alive \ \wedge Holds(loaded, \alpha)) \tag{10}$$

From the fluent facts and axiom (e) we conclude that $Holds(alive, Sit\_map(S_0))$ and $\neg Holds(alive, Sit\_map(S_1))$. Now let us reason about the value of the fluent *loaded* in the initial state. There are two cases:

a) *loaded* was initially false ($\neg Holds(loaded, \epsilon)$) and since there is no action that affects this fluent, i.e. for all a, α, $\neg Causes^+(a, loaded, \alpha)$ and $\neg Causes^-(a, loaded, \alpha)$, we have that

$$(\forall \alpha)\neg Holds(loaded, \alpha) \tag{11}$$

From this and (10) we have that

$$(\forall a, f, \alpha)\neg Causes^-(a, f, \alpha) \tag{12}$$

From the fact *alive* **at** S_0 we have $At(alive, Sit\_map(S_0))$ and from this and axioms (a) and (e) we have $Holds(alive, \epsilon)$. From this, (9), (12), and axiom

$$\neg Causes^+(a, f, \alpha) \ \wedge \neg Causes^-(a, f, \alpha) \supset [Holds(f, \alpha) \equiv Holds(f, a \circ \alpha)]$$

we conclude that $(\forall \alpha).Holds(alive, \alpha)$. Clearly, by the above and the fact $\neg alive$ **at** S_1 (which is translated into $\neg At(alive, Sit\_map(S_1))$) axiom (e) cannot be satisfied regardless how function Sit_map is interpreted.

Thus, this interpretation of the value of *loaded* in the initial situation leads to an evolution of the domain which does not fit the observations.

b) The other possibility is that *loaded* was initially true, i.e. $Holds(loaded, \epsilon)$ holds. From (10) this implies that $Causes^-(shoot, alive, \epsilon)$. By axiom

$$Causes^-(a, f, \alpha) \supset \neg Holds(f, a \circ \alpha)$$

we have $\neg Holds(alive, shoot \circ \epsilon)$. Clearly, an interpretation of Sit_map such that S_1 and S_N are mapped onto sequence $shoot \circ \epsilon$ satisfies the facts and minimization of $Sit\_map(S_N)$. Finally, by axiom (f), $Occurs(shoot, S_0)$ holds. $\Box$

7 Correctness of the NAT formalization

In this section we formulate the results about the correctness of our NAT formalization of domain descriptions in $\mathcal{L}$. In the process, for a domain description D, we show the equivalence of models of the axioms in $SC(D_l)$ with respect to the causal models of D. This illustrates the modularity of our NAT formulation, and later (in Section 9) we discuss how this allows us to easily extend the language with additional features, such as constraints, by only replacing $SC(D_l)$. The proofs of the lemmas and the theorem in this section are given in the appendix.

For readability, we will use α, β, γ to denote sequence variables and $\vec{A}, \vec{B}, \vec{G}$ to denote sequence constants, both in D and the models of $T(D)$. For instance, we may write $Holds(f, \vec{A}) \;\Leftrightarrow\; f \in \Psi(\vec{A})$ meaning $Holds(f, A_m \circ \ldots \circ A_1 \circ \epsilon) \;\Leftrightarrow\; f \in \Psi([A_1, \ldots, A_m])$.

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** $Q_1, \ldots, Q_m$, are said to be contradictory if $\{P_1, \ldots, P_n\} \cap \{\overline{Q_1}, \ldots, \overline{Q_m}\} = \emptyset$). (We remove this restriction in the next section when we allow concurrent actions.)

Lemma 1 *(Causal Equivalence)*
Part 1: *For every causal model Ψ of D_l there exists a model $\mathcal{M}_{\rm SC}$ of $SC(D_l) \cup$ Framework Axioms such that for all f,α:*

$$f \in \Psi(\alpha) \;\Leftrightarrow\; \mathcal{M}_{\rm SC} \models Holds(f, \alpha). \tag{13}$$

Part2: *For every model $\mathcal{M}_{\rm SC}$ of $SC(D_l)$ there exists a causal model Ψ of D_l such that (13) holds.* □

Lemma 2 (main) *(Equivalence of Models) For any domain description D, if the 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:*

$$(\Psi, \Sigma) \models \phi \;\Leftrightarrow\; \mathcal{M} \models \tau(\phi) \tag{14}$$

and if $\mathcal{M}$ is a model of $T(D)$ then there exists a model (Ψ, Σ) of D such that (14) holds. □

(In the above lemma we have abused notation and used $\models$ for both entailment in $\mathcal{L}$ and entailment in classical logic. In all places we use $\models$ it is clear from the context which entailment we mean.)

Theorem 1 (main) *(Equivalence) For any domain description D and fact*

ϕ in the language of D:

$$D \models \phi \;\Leftrightarrow\; T(D) \models \tau(\phi)$$

$\square$

The following proposition shows the correctness of our theory with respect to hypotheses.

Proposition 1 *For any domain description D and hypothesis ϕ in the language of D:*

$$D \models \phi \;\Leftrightarrow\; T(D) \models \tau(\phi)$$

Proof:

Let F **after** $\vec{A}$ be an arbitrary hypothesis in the language of D. We need to show that

$$D \models F \textbf{ after } \vec{A} \;\Leftrightarrow\; T(D) \models \tau(F \textbf{ after } \vec{A})$$

where $\tau(F \textbf{ after } \vec{A})$ stands for:

$$(\forall \beta) Concatenate(Sit\_map(S_N), \vec{A}, \beta) \supset H(F, \beta)$$

By Lemma 1 we have that for all α, F is true in $\Psi(\alpha)$ iff $H(F, \alpha)$ holds in $T(D)$. By this and correctness of $B_{\text{concatenate}}$ (Lemma 10), to prove this proposition it is sufficient to show that for all models (Ψ, Σ) of D and models $\mathcal{M}$ of $T(D)$ and for all α,

$$\Sigma(S_N) = \alpha \;\Leftrightarrow\; \mathcal{M} \models (Sit\_map(S_N) = \alpha) \tag{15}$$

This follows from the proof of Lemma 2 where we showed how to construct a model of $T(D)$ from a model M of D, and vice-versa, such that (15) holds. $\square$

8 Narratives with concurrent actions

In this section we consider reasoning about concurrent actions together with narratives. This means we weaken the informal assumption (e) of Section 2.1 and allow concurrent cotemporaneous actions. But we still do not allow non-cotemporaneous overlapping actions. We extend the semantics of $\mathcal{L}$ to allow concurrent actions. We will only consider concurrent actions consisting of the the simultaneous execution of a finite number of basic actions with non contradictory effects. For a more sophisticated treatment of concurrent actions in a non-narrative setting see e.g., [BarGel93] and [LinSho92]. We will refer to the language with the extended semantics as $\mathcal{L}_c$.

8.1 Extended semantics

Concurrent actions will be denoted by special symbols using set notation. Basic actions will be denoted as before. By the symbol $\{A_1, \ldots, A_n\}$, we denote the concurrent action consisting of basic actions $A_1, \ldots, A_n$ from $\mathcal{A}$. As in sets, the order of the basic actions in a concurrent action is unimportant. I.e., symbols formed by permutations of the same collection of basic actions will be assumed to be the same. All concurrent actions of the form $\{A\}$ (represented by a singleton set) will be equated to A. For notational convenience, we will use $A \in B$ and $B \subseteq C$, where A is a basic action and B and C are concurrent actions, to denote similar relations on concurrent actions as with sets.

We will now consider the changes that are required in the semantics to characterize this extended language. First, we need to redefine what the effect of an action is since we now have a more general notion of action. We say that a fluent F is an *effect* of an action A in a state σ if F is an immediate effect of A in σ, or there is an action $B \in A$ such that F is an immediate effect of B in σ. Furthermore, we replace the definitions of sets $E_A^+(\sigma)$ and $E_A^-(\sigma)$ with the following:

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

We will also need the following notions. Let $\alpha = [A_1, \ldots, A_k]$ and $\beta = [B_1, \ldots, B_k]$. We say β is *embedded* in α (written as $\beta \subseteq \alpha$) if $B_i \subseteq A_i$ for each i. Intuitively, $\beta \subseteq \alpha$ means that we can obtain β from α by removing some basic actions from each (concurrent) action in the action sequence α.

Furthermore, we say β is an *embedded-subsequence* of γ if there is a subsequence α of γ such that β is embedded in α.

The definitions of Causal Interpretation, Situation Assignment, Interpretation, and consistent domain description remain the same. We only need to modify the definitions 4 and 5.

Definition 8 *In Definition 4 we only need to modify item (ii) as follows:*

(ii) (α **occurs_at** S) *is true in* M *if there is* β *such that* $\Sigma(S) \circ \beta$ *is a prefix of the actual path of* M *and* $\alpha \subseteq \beta$.

The rest remains the same.

□

Definition 9 *In Definition 5 we only need to modify condition (iii) as follows:*

(iii) There is no other interpretation $N = (\Psi, \Sigma')$ *such that* N *satisfies condition (ii) and* $\Sigma'(S_N)$ *is an embedded-subsequence of* $\Sigma(S_N)$.

The rest remains the same. □

8.2 Translation into NATs

In this section we give a NAT characterization – a modification of our NAT translation from Section 4 – of the extended semantics described in the previous section.

Let us start with the changes to the Frame Axioms. Now, in addition to the object constants of sort action corresponding to basic actions, we have constants of the form $\{A_1, \ldots, A_k\}$, for all finite number k of basic actions, which denotes a concurrent action. The axioms in $UNA[actions]$ now include inequalities about these constants as well.

In addition, we add axioms defining a new predicate In with two action arguments. Intuitively, $In(a, b)$ means basic action a is one of the actions in concurrent action b. We define this predicate explicitly by including an axiom:

$$In(a, \{A_1, \ldots, A_k\}) \equiv \bigvee_{1 \leq i \leq k} a = A_i$$

for each concurrent action $\{A_1, \ldots, A_k\}$ and finite number k of basic actions.

Now, we need to change the sub-theory $SC(D_l)$ to capture the effects of concurrent actions. The kind of concurrent actions considered here are such that they "inherit" the effects of their constituent actions. Thus, we include two blocks, one defining predicate $Inherits^+(a, f, \alpha)$ and one defining predicate $Inherits^-(a, f, \alpha)$, both with arguments of the sorts, action, fluent and sequence resp.:

$$\{\min \ Inherits^+ \ :$$

$$In(b,a) \ \wedge Causes^+(b,f,\alpha) \supset Inherits^+(a,f,\alpha)$$

$$\}$$

The block for $Inherits^-$ is defined in the obvious similar way. The intuitive meaning of $Inherits^{+(-)}(a,f,\alpha)$ is that concurrent action a inherits the effect f $(\neg f)$, from its sub-actions, in the situation after α is executed.

We also add the following block defining the predicate $Undefined$ characterizing action sequences where the causal interpretation Ψ is undefined. (We did not have this before because we did not allow contradictory causal laws.)

$$\{\min \ Undefined \ :$$

$$Inherits^+(a,f,\alpha) \ \wedge Inherits^-(a,f,\alpha) \supset Undefined(a \circ \alpha)$$

$$Undefined(\alpha) \supset Undefined(a \circ \alpha)$$

$$\}$$

Next, we replace the first three axioms of $SC(D_l)$ with the following axioms:

$$\neg Inherits^+(a,f,\alpha) \ \wedge \neg Inherits^-(a,f,\alpha) \supset [Holds(f,\alpha) \equiv Holds(f,a \circ \alpha)]$$

$$\neg Undefined(a \circ \alpha) \ \wedge Inherits^+(a,f,\alpha) \supset Holds(f,a \circ \alpha)$$

$$\neg Undefined(a \circ \alpha) \ \wedge Inherits^-(a,f,\alpha) \supset \neg Holds(f,a \circ \alpha)$$

Next, we add a new block, B_{embedded}, at the same level of nesting as blocks defining $Prefix\_eq$, $Subsequence$, etc., which defines a predicate $Embedded$, with two arguments of the sort sequence:

$$B_{\text{embedded}} =$$

$$\{\min Embedded \ :$$

$$Embedded(\epsilon,\epsilon)$$

$$[In(a,b) \supset In(a,c)] \supset Embedded(b \circ \epsilon, c \circ \epsilon)$$

$$[Embedded(b \circ \epsilon, a \circ \epsilon) \ \wedge Embedded(\beta,\alpha)] \supset Embedded(b \circ \beta, a \circ \alpha)$$

$$\}$$

The new ordering, *embedded-subsequence*, is captured by the predicate $Embsubseq$ which is defined in terms of the relations $Embedded$ and $Subsequence$ in the following block:

$B_{\text{embsubseq}} =$

$\{\min Embsubseq\ :$

$$Embedded(\alpha, \beta)\ \wedge Subsequence(\beta, \gamma) \supset Embsubseq(\alpha, \gamma)$$

$\}$

Axiom (f) needs to be modified to capture occurrence of sequences which include concurrent actions. We substitute it with the following axiom:

$$(f')\ Occurs(\alpha, s) \equiv \begin{bmatrix} (\exists\beta, \gamma) Embedded(\alpha, \beta)\ \wedge Concatenate(Sit\_map(s), \beta, \gamma) \\ \wedge Prefix\_eq(\gamma, Sit\_map(S_N))) \end{bmatrix}$$

At the top level of nesting we add the following axiom

$$\neg Undefined(Sit\_map(S_N))$$

which restricts actual situations to be mapped onto defined states. (We did not need this in $T(D)$, because our restriction on D to not allow contradictory causal laws guarantee that the causal interpretation Ψ is defined for all action sequences.)

Finally, minimization of $Sit\_map(S_N)$ is now with respect to the partial order $Embsubseq$, thus we replace axiom (c) with axiom

$$Embsubseq(\alpha, Sit\_map(S_N)) \supset Ab(\alpha) \tag{16}$$

We will denote the new theory by T_c and the modified block SC by SC_c.

$T_c(D) =$

$\{ \; Sit\_map \; :$

$\neg Undefined(Sit\_map(S_N))$

$(a) \; Sit\_map(S_0) = \epsilon$

$(b) \; Prefix\_eq(Sit\_map(s), Sit\_map(S_N))$

$(c') \; Embsubseq(\alpha, Sit\_map(S_N)) \supset Ab(\alpha)$

$(d) \; Prefix(\alpha_1, \alpha_2) \equiv [Prefix\_eq(\alpha_1, \alpha_2) \; \wedge \neg Prefix\_eq(\alpha_2, \alpha_1)]$

$\tau(D_f), \; SC_c(D_l)$

$B_{\text{prefix\_eq}}, \; B_{\text{subsequence}}, \; B_{\text{concatenate}}, \; B_{\text{embedded}}, \; B_{\text{embsubseq}}$

Framework axioms

$\}$

where:

$$\tau(D_f) = \begin{cases} (e) \; At(f,s) \equiv Holds(f, Sit\_map(s)) \\ (f') \; Occurs(\alpha, s) \equiv \begin{bmatrix} (\exists \beta\gamma) Embedded(\alpha, \beta) \; \wedge Concatenate(Sit\_map(s), \beta, \gamma) \\ \wedge Prefix\_eq(\gamma, Sit\_map(S_N))) \end{bmatrix} \\ (g) \; Precedes(s_1, s_2) \equiv Prefix(Sit\_map(s_1), Sit\_map(s_2)) \\ FACTS \end{cases}$$

$SC_c(D_l) =$

$$\left\{\begin{array}{l}
\neg Inherits^+(a,f,\alpha) \ \wedge \neg Inherits^-(a,f,\alpha) \supset [Holds(f,\alpha) \equiv Holds(f,a\circ\alpha)] \\
\neg Undefined(a\circ\alpha) \ \wedge Inherits^+(a,f,\alpha) \supset Holds(f,a\circ\alpha) \\
\neg Undefined(a\circ\alpha) \ \wedge Inherits^-(a,f,\alpha) \supset \neg Holds(f,a\circ\alpha) \\
\{ \ \min \ Causes^{+(-)} \ : \\
\quad H(P_1,\alpha) \ \wedge \ldots \ \wedge H(P_n,\alpha) \supset Causes^{+(-)}(A,F,\alpha) \\
\qquad\qquad\qquad\qquad \text{(for each } A \textbf{ causes } (\neg)F \textbf{ if } P_1,\ldots,P_n \in D) \\
\} \\
\{\min \ \ Inherits^{+(-)} \ : \\
\qquad In(b,a) \ \wedge Causes^{+(-)}(b,f,\alpha) \supset Inherits^{+(-)}(a,f,\alpha) \\
\} \\
\{\min \ \ Undefined \ : \\
\qquad Inherits^+(a,f,\alpha) \ \wedge Inherits^-(a,f,\alpha) \supset Undefined(a\circ\alpha) \\
\qquad Undefined(\alpha) \supset Undefined(a\circ\alpha) \\
\}
\end{array}\right.$$

We now state the correctness of our NAT formulation of $\mathcal{L}_c$ with lemmas and a theorem similar to those in Section 7. The proofs are given in the Appendix.

Lemma 3 *(Causal Equivalence)*
Part 1: *For every causal model Ψ of D_l there exists a model $\mathcal{M}_{\rm SC}$ of $SC_c(D_l)$ such that for all f, α:*

$$\begin{array}{l}
f \in \Psi(\alpha) \ \Leftrightarrow \ \mathcal{M}_{\rm SC} \models Holds(f,\alpha), \text{ and} \\
\Psi(\alpha) \textit{ is undefined } \Leftrightarrow \ \mathcal{M}_{\rm SC} \models Undefined(\alpha)
\end{array} \tag{17}$$

Part2: *For every model $\mathcal{M}_{\rm SC}$ of $SC_c(D_l)$ there exists a causal model Ψ of D_l such that (17) holds.* □

Lemma 4 *For any domain description D, if interpretation (Ψ,Σ) is a model of D then there exists a model $\mathcal{M}$ of $T_c(D)$ such that for every fact ϕ in the language of D:*

$$(\Psi,\Sigma) \models \phi \ \Leftrightarrow \ \mathcal{M} \models \tau(\phi) \tag{18}$$

and if $\mathcal{M}$ is a model of $T_c(D)$ then there exists a model (Ψ,Σ) of D such that

(18) holds. □

Theorem 2 *For any domain description D and fact ϕ in the language of D:*

$$D \models \phi \;\Leftrightarrow\; T_c(D) \models \tau(\phi)$$

□

Example 6 Consider a scenario where there are two guns which cause Fred to become dead if fired at him when loaded. Suppose all we know is that initially gun 1 or gun 2 is loaded and that both guns were fired at him in the initial situation. The following domain description captures this story:

$$\left.\begin{array}{l} shoot(1) \ \textbf{causes} \ \neg alive \ \textbf{if} \ loaded(1) \\ shoot(2) \ \textbf{causes} \ \neg alive \ \textbf{if} \ loaded(2) \\ alive \ \textbf{at} \ S_0 \\ (loaded(1) \ \textbf{at} \ S_0) \vee (loaded(2) \ \textbf{at} \ S_0) \\ \{shoot(1), shoot(2)\} \ \textbf{occurs\_at} \ S_0 \end{array}\right\} D_{\text{guns}}$$

It is easy to see that D_{guns} entails $\neg alive$ **at** S_N. □

9 Further extension of $\mathcal{L}$ – allowing actions with indirect effects

In this section we discuss how the NAT formalization of $\mathcal{L}$ can be easily extended to allow *ramification constraints, and axioms about executability of actions.*

Giunchiglia, Kartha and Lifschitz [KarLif94,GiuKarLif95] introduced the high level language $\mathcal{AR}$ with the above mentioned features (and some additional features, such as actions with non-deterministic effects) with non-propositional fluents, but in the absence of narratives. They then give a formulation of temporal projection problems in $\mathcal{AR}$ using NATs. Essentially, if we restrict their formulation to boolean fluents, then we can say that their NAT characterization corresponds to a causal model Ψ for an extended language that allows ramification constraints, executability conditions and non-deterministic effects.

Because we also use NATs in our paper and NATs are nestings of blocks, we can replace the $SC(D_l)$ part of the NAT $T(D)$ of Section 4.5 – that characterizes the causal model of our original but restricted language, by a new theory that captures the causal model of the extended language. Moreover, the resultant NAT gives a characterization of an extension of $\mathcal{L}$ where constraints and executability conditions are allowed.

We now present the syntax of (ramification) constraints, and executability conditions and briefly touch upon how their addition to $\mathcal{L}$ necessitates some changes in the semantics.

A constraint is of the form

$$\textbf{always} \;\; C \tag{19}$$

where C is a propositional fluent formula. The semantics of $\mathcal{L}$ extended with constraints is defined by requiring $Res(A, \sigma)$ to be the set of valid states closest to σ which contain $E_A^+(\sigma)$ and do not contain any element of $E_A^-(\sigma)$) – where valid states are states that satisfy all the constraints; and requiring $\Psi(\alpha \circ A)$ to be an element of $Res(A, \Psi(\alpha))$. This guarantees that for all $alpha$, either C holds in $\Psi(\alpha)$ or $\Psi(\alpha)$ is undefined and captures the indirect effect of actions due to constraints.

Following is an example (from [GiuKarLif97]) of a constraint:

$$\textbf{always} \;\; (in\_lake \supset wet)$$

which means that if some one is in the lake then he/she is wet.

An executability condition is of the form

$$\textbf{impossible} \;\; A \; \textbf{if} \;\; Q_1, \ldots, Q_n \tag{20}$$

where $Q_1, \ldots, Q_n$ are fluent literals. From the semantics point of view, a proposition of this type is satisfied by a causal interpretation Ψ if $\Psi(\alpha \circ A)$ is undefined when $Q_1, \ldots, Q_n$ are satisfied by $\Psi(\alpha)$.

An example of an executability condition is

$$\textbf{impossible} \;\; get\_out\_of\_the\_lake \; \textbf{if} \;\; \neg in\_lake$$

which means that it is impossible to execute the action $get\_out\_of\_the\_lake$ in a state where in_lake is false.

We are now ready to discuss the theory, T^+, which replaces SC in the NAT T (of Section 4.5) by a theory SC^+ which encodes the transition function Ψ in presence of constraints and executability conditions. We now present the theory SC^+.

$SC^+ =$

$(h)\ \gamma_\alpha C$ (for each constraint (19))

$(i)\ Holds^*\ (f, a \circ \alpha) \equiv Holds(f, a \circ \alpha)$

$\{Holds^*$:

$(j)\ \neg Undef(a \circ \alpha)\ \wedge \neg Ab(f, a, \alpha) \supset$

$[Holds(f, \alpha) \equiv Holds^*(f, a \circ \alpha)]$

$\{Holds^*$, min $Undef$:

$(k)\ H(P_1, \alpha)\ \wedge \ldots\ \wedge H(P_m, \alpha) \wedge$

$\neg Undef(A \circ \alpha) \supset H^*(F, A \circ \alpha)$

(for each causal law (1))

$(l)\ H(Q_1, \alpha)\ \wedge \ldots\ \wedge H(Q_n, \alpha) \supset Undef(A \circ \alpha)$

(for each proposition (20))

$(m)\ Undef(\alpha) \supset Undef(a \circ \alpha)$

$(n)\ \gamma^*_{a \circ \alpha} C$ (for each constraint (19))

}

}

where, $\gamma_\alpha C$ stands for the universally closed formula obtained from a constraint (19) by replacing each fluent literal F by $H(f, \alpha)$. Some observations about SC^+ are as follows:

- SC^+ is a slightly modified and simplified version of the main part of the NAT in [GiuKarLif95,Kar94]. The main differences are: our use of boolean propositions instead of non-boolean propositions as in [Kar94] and, our use of conjunction of fluent literals in the **if** part of (1) and (20) and in the **causes** part of (1) instead of fluent formulas, as in [Kar94]. We make these restrictions for simplicity, and to stay as close to $\mathcal{L}$ as possible, while still being able to make our point about easy incorporation of constraints and executability conditions to $\mathcal{L}$.
- The outer block of SC^+ encodes $Res(a, \sigma)$, for all actions a, and for all states σ that are mapped to by some action sequence. While σ is encoded

using $Holds$, $Res(a, \sigma)$ is encoded using $Holds^*$. The inner block defines which actions are undefined (or not executable) in which state.

The reason behind having two different predicates $Holds$ and $Holds^*$ and only varying the latter when minimizing Ab, is to individually minimize the difference between σ and $Res(a, \sigma)$; not to globally minimize all such differences as a whole. The latter precludes many models and may result in intuitive models being eliminated. For example, suppose we have a **causes** f **if** p and **always** $\neg f$ in our domain description. Suppose we replace $Holds^*$ in SC^+ by $Holds$. Then minimizing $Undef$ while varying $Holds$ in the inner block will result in $\neg Holds(p, \alpha)$ being true in all models of the inner block. This is prevented by using separate $Holds$ and $Holds^*$.

- Although at first glance it seems that only (l) and (m) are necessary in the inner block, the necessity of (k) and (n) are to specify undefined action sequences that may arise because of possible interaction between effect axioms and constraints.

  Note that our constraints are meant for ramification; i.e. if we have **always** $\neg(f \wedge g)$ and a **causes** f, then application of a in a state where g is true (directly) makes f true in the resulting situation and (indirectly, because of ramifications) makes g false in the resulting state.

  But if we have
  always $\neg(f \wedge g)$, a **causes** f and a **causes** g **if** p,
  then a is not executable in any situation where p is true. To capture this we need (k) and (n) in the inner block.

- The axiom (j) encodes the law of inertia.
- The axiom (h) restricts all states to be valid, making sure they satisfy all the constraints.
- The axiom (i) relates $Holds$ and $Holds^*$. Once the minimizations are done appropriately in the inner and outer blocks, axiom (i) selects only those models where $Holds$ and $Holds^*$ agree.
- Another difference between SC^+ and the corresponding theory in [KarLif94,Kar94] is that in our case the second argument of $Holds$ is a sequence of actions while in the other ones it is a state. This is because all situation constants in our language are forced to map to a sequence of actions. There is no such requirement in [KarLif94,Kar94].
- Finally, although we did not discuss allowing indeterminate propositions [Kar94], they can be easily incorporated. An indeterminate proposition of the form A *possibly_changes* F **if** $P_1, \ldots, P_k$ is incorporated into SC^+ by adding

$$H(P_1, \alpha) \wedge \ldots \wedge H(P_k, \alpha) \supset Ab(F, A, \alpha)$$

just before (j) inside the outer block of SC^+.

Although we do not formally state and prove the equivalence between the causal interpretation Ψ of a domain description D with constraints and executability conditions and the corresponding SC^+, the equivalence directly follows from [Kar94]. Moreover, the correspondence between domain descriptions in the extended $\mathcal{L}$ (with constraints and executability conditions) and the corresponding NATs, with SC replaced by SC^+, follows (with slight modifications) from [Kar94] and the proof of Lemma 2.

We would like to stress that the ability of NATs to allow us to treat blocks in a similar way as subroutines in structured programming makes NATs a strong tool for knowledge representation. This allowed us to take a block from the formalization of actions of Giunchiglia et. al. and "insert" it into our theory with practically no complications.

10 T and T^+ as knowledge representation testbeds

To the best of our knowledge T and T^+ are among the largest circumscriptive theories (T^+ is possibly the largest theory), in the literature, in terms of the number of levels of minimization and filtering [San89]. They are thus good testbed examples to analyze the expressibility of NATs (and circumscription in general) in terms of the various KR features they incorporate. In Section 5 we already discussed how 'value minimization' [BarGabPro96b] of functions was expressed using NATs in the theory T. In this section we focus on three additional KR aspects – abductive reasoning, filtering and restricted monotonicity, discuss some general relation between them, study them with respect to T and T^+ and point out future directions of research.

10.1 Abductive reasoning

Abduction is often considered as a method of reasoning which, given a theory T and an observation Q, finds possible explanations of Q in terms of a particular set of predicates (literals) referred to as abducible predicates (literals). T is usually in a nonmonotonic language, and we refer to its entailment relation by $\models$. Q is usually a set of sentences in first-order logic. We now formally define an explanation:

Definition 10 (Explanation) *A set of abducible literals E is said to be an explanation of Q with respect to a theory T if $T \cup E \models Q$ and $T \cup E$ is* *consistent.* □

In the above definition if we consider Q as an observation, then the explanation E explains the observation with respect to the theory T. Now what if we consider T and Q together. I.e., we are now given a theory T and also an observation Q. How do we define a new entailment relation with respect to the pair $\langle T, Q\rangle$ – which we refer to as an *abductive theory?*

Kakas et. al. [KakKowTon92] introduced the concept of generalized stable models of a pair $\langle T, Q\rangle$, where T is an abductive logic program, and Baral and Gelfond [BarGel94] used it to define an entailment relation for $\langle T, Q\rangle$. In the following we generalize the above concepts to define abductive models and abductive entailment with respect to pairs $\langle T, Q\rangle$, where T is a theory in any language. (I.e., T is not necessarily a logic program.)

Definition 11 *(***Abductive models and Abductive Entailment***)*

(i) M is an abductive model of $\langle T, Q\rangle$ iff there exists an explanation E of Q with respect to T such that M is a model of $T \cup E$.
(ii) $\langle T, Q\rangle \models_{\text{abd}} f$ iff f is true in all (abductive) models of $\langle T, Q\rangle$.

□

Proposition 2 *Abductive theories are monotonic with respect to addition of observations.* □

Proof: Suppose we have $Q_1 \subset Q_2$. Then any explanation of Q_2 with respect to T is an explanation of Q_1 with respect to T. Thus models of $\langle T, Q_2\rangle$ are models of $\langle T, Q_1\rangle$ and hence, $\models_{\text{abd}}$ is monotonic with respect to Q. □

A closely related notion to abduction is called filtering, which was first introduced by Sandewall in [San89]. We define it as follows:

Definition 12 *Let T be a (possibly nonmonotonic) theory and Q (an observation) be a set of sentences in first-order logic. By $Filter(T, Q)$, we refer to the theory whose models are the models (in the language of T) of T that are first-order models of Q.* □

Proposition 3 *$Filter(T, Q)$ is monotonic with respect to Q.* □

Proof: Follows directly from the definition of $Filter(T, Q)$.

10.2 Restricted monotonicity in abductive reasoning and filtering

The concept of 'restricted monotonicity' was introduced by Lifschitz [Lif93]. In this subsection we first recall some of the definitions from [Lif93] and then show how in general an abductive theory has the restricted monotonicity property about its observations, and how restricted monotonicity is captured in NATs.

Definition 13 *[Lif93] A* declarative formalism *is defined by a set S of symbolic expressions called* sentences*, a set P of symbolic expressions called* postulates*, and a map Cn from sets of postulates to sets of sentences.*

A set of postulates is referred to as a theory*; and a sentence A is a consequence of a theory T is $A \in Cn(T)$.* □

Definition 14 *[Lif93] Let $\langle S, P, Cn \rangle$ be a declarative formalism. Let a subset S_0 of S be designated as the set of assertions, and a set P_0 of P as the set of parameters. A theory T is said to satisfy the restricted monotonicity condition with respect to S_0 and P_0 if, for any sets $p, q \subset P_0$,*

$$p \subset q \Rightarrow Cn(T \cup p) \cap S_0 \subset Cn(T \cup q) \cap S_0$$ □

In [Lif93], Lifschitz gives several examples of restricted monotonicity in particular theories and in [KarLif94,GiuKarLif97] Kartha, Giunchiglia and Lifschitz point out restricted monotonicity property of $\mathcal{AR}$ and $\mathcal{AR}_0$ with respect to particular additions to the theory. *Here we generalize some of the above examples, and discuss the restricted monotonicity associated with abductive reasoning and filtering in general.*

In case of abductive reasoning we can say that an abductive theory has the restricted monotonicity property with respect to observations. This follows directly by considering parameters as statements in the language of the observations, defining $Cn(T \cup p)$ to be the set $\{f \mid \langle T, p \rangle \models_{\text{abd}} f\}$ and considering S_0 to be the set of all sentences.

In case of filtering we can say that the theory $Filter(T, Q)$ has the restricted monotonicity property with respect to Q. This follows directly by considering parameters as statements in the language of the observations, defining $Cn(T \cup p)$ to be the set $\{f \mid f$ is true in all models of $Filter(T, Q)\}$ and considering S_0 to be the set of all sentences.

Filtering is easily achieved in NATs by having a NAT with two blocks one representing Q and the other representing T; thus making it easy [8] to express restricted monotonicity in NATs. This is exactly what happens in T and T^+. We analyze various blocks of T and T^+ in the next subsection.

But before that let us say a few more words about NATs as a knowledge representation language. Besides the issue of recent advances of automated circumscriptive reasoning [Doherty et al. 95] that impacts NATs, *NATs are good for representing knowledge because they allow us to encode nonmonotonicity about certain aspects, and at the same time allow the encoding of restricted monotonicity with respect to other aspects.* Moreover, because of the nesting we can have several levels of filtering. Note that, the fact that $Filter(Filter(T,Q),Q')$ is equivalent to $Filter(T, Q \cup Q')$ may suggest that we do not need several levels of filtering. But often (as in [San89]), we need to compute $Filter(Min(Filter(T,Q), \leq), Q')$, where we minimize $Filter(T,Q)$ with respect to an ordering $\leq$ before filtering the result by Q'. Assuming that $\leq$ can be encoded by varying $P_1, \ldots, P_k$ and minimizing $P_{k+1}, \ldots, P_m$, and T itself is a NAT, we can represent $Filter(Min(Filter(T,Q), \leq), Q')$ by the following NAT:

$$
\begin{array}{l}
\{ \\
\quad Q' \\
\quad \{ \;\; P_1, \ldots, P_k, \;\; min P_{k+1}, \ldots, P_n \\
\qquad Q \\
\qquad T \\
\quad \} \\
\}
\end{array}
$$

10.3 Analyzing blocks and sub-blocks of T and T^+

- Consider the inner block, B_{inner}, of SC^+. Intuitively, this block (among other things) encodes a function from literals about the predicate $Holds$, to literals about the predicate $Undef$. The axioms $(k), (l), (m)$, and (n) partially define this function, which is further refined by the minimization of $Undef$ while varying $Holds^*$. With respect to this block, observations could

[8] Among other knowledge representation languages, filtering is easily done in logic programming by representing observations as integrity constraints. But, normally logic programs do not allow a hierarchy of integrity constraints. Thus we cannot represent several levels of filtering in logic programs.

be facts about $Undef$. Suppose we were able to execute a sequence of actions β. Then we know that $\neg Undef(\beta)$ is true, and we can treat this as an observation. We can incorporate this observation (with the restricted monotonicity property) by adding a new block B_1 consisting only of $\neg Undef(\beta)$. Our resultant theory B_2 would be the NAT consisting of B_{inner} and B_1. This would change SC^+ to the theory where $\neg Undef(\beta)$ is added just after the axiom (j), and before B_{inner}.

In the NAT B_2 consisting of B_{inner} and B_1 we are basically filtering B_{inner} by B_1, and the resultant models are the models of B_{inner} that are also models of B_1. Thus B_2 is monotonic with respect to addition of observations to B_1. But B_2 is nonmonotonic with respect to addition of causal laws, constraints, or executability conditions, which change B_{inner}.

We can also think of the set of $Holds$ literals as abducibles, and when we have an observation $\neg Undef(\beta)$ then the models of B_2 are also abductive models of $\langle B_{\text{inner}}, B_1 \rangle$, where each abductive model contains an explanation of the observation B_1.

- The outer block, B_{outer} of SC^+, intuitively encodes a function from literals about the predicate $Holds$, to literals about the predicate $Undef$ and $Holds^*$. Observations about the predicate $Holds^*$ are put outside of B_{outer}, as axioms (h) and (i) in SC^+, and axioms $FACTS$ and (e) in T^+. Here also, the observations filter the NAT B_{outer} and the resultant theory has the restricted monotonicity property about the observations.
- The block $T(D)$, represents a function from literals of predicates At, $Occurs$, $Precceds$, $Holds$, $Subsequence$, $Prefix\_eq$, $Concatenate$ to mappings of the function Sit_map. Observations about the value of the function Sit_map can be incorporated by putting these observations outside the NAT consisting of the block $T(D)$. These observations will then filter the models of $T(D)$ and keep only those models that agree with the observation.
- In [BarGelPro97] hypothesis of the form F **after** α **at** S were treated as observations on top of the domain description in $\mathcal{L}$. We can incorporate them to our NAT consisting of $T(D)$, by adding $\forall \beta Concatenate(Sit\_map(S), \alpha, \beta) \Rightarrow Holds(F, \beta)$ to the NAT. (I.e., as a block in addition to the original block $T(D)$. We do not put it inside $T(D)$.) These observations will filter $T(D)$, and select only those models of $T(D)$ that agree with the observations. The theory $Filter(T(D), Obs)$ will thus be monotonic with respect to the hypotheses, while it is nonmonotonic with respect to facts, which are inorporated by changing $T(D)$.

10.4 Future direction

In the previous subsections we discussed the relation between filtering and restricted monotonicity in general and instances of filtering and restricted monotonicity in T and T^+. Although we also discussed the relation between abductive reasoning and restricted monotonicity is general, we did not discuss all instances of abductive reasoning in T and T^+.

We believe that each filtering step in T and T^+ also encodes abductive reasoning. In fact we believe that abductive reasoning is done in T and T^+ through filtering. *We need to further study this issue*[9] *and identify all instances of abduction in T and T^+, and discuss under what circumstances abduction can be done through filtering.*

11 Related work

Until recently there were two distinct directions in reasoning about actions; the first based on situation calculus which did hypothetical reasoning and planning and normally considered simple actions (i.e., no continuous actions, no actions with durations, etc.), and the second (in particular, Allen's temporal logic [All84], Kowalski and Sergot's event calculus [KowSer86]) focused on reasoning with narratives and often allowed actions with durations, and continuous actions, but did not consider hypothetical reasoning. In the late eighties and early nineties several influential works [GelLifRab91,GL92a,San92,Rei91] have had a big impact in this field, and triggered a flurry of new research. Some of these appear in the special journal issues [Lif97,Geo94], in the workshops AAAI 96 Workshop on Reasoning about actions, planning and robot control: Bridging the gap; and AAAI 95 Spring symposium on Extending Theories of Action: Formal Theory and Practical Applications, and in recent AAAI, IJCAI and KR conferences.

Our work in this paper was influenced by the approach of using high level action description languages [GL92a,GiuKarLif97,BarGel93,BarGelPro97], and their formalization (particularly of $\mathcal{AR}$ [GiuKarLif97]) using nested circumscription [Lif95]. In the previous sections we argued why we believe nested circumscription is an excellent KR language, and listed some of its features. But, we must acknowledge Sandewall's idea of filtering [San89] which has a lot of similarity with NATs.

[9] Currently, we have some preliminary results on this.

In this section we first compare NATs with Sandewall's filter preferential entailment and then give a detailed comparison of our work with that of other proposals that do both narrative reasoning and hypothetical reasoning; particularly, those by McCarthy [McC95], Kakas and Miller [KakMil96], Miller and Shanahan [MilSha94], and Pinto and Reiter [PinRei93,Pin96].

11.1 *Filter preferential entailment of Sandewall*

The notion of filtering was first introduced by Sandewall [San89]. He introduced it to be able to formally *obtain explanations in terms of action occurrences* given action descriptions (in terms of conditions and effects), physical laws, and observations about values of fluent at specific time instants. He argued that his approach can also be used to obtain a plan for a specific goal, by considering the goal as an observation which needs to be explained, and considering an explanation as a plan that achieves the goal.

His notion of filtering is a general notion. The NATs formalism that we use in this paper is a particular approach to filtering, where circumscription is used for minimizations, with local abnormal predicates.

He motivates the intuition behind using filtering of the models of a theory containing action descriptions, and natural laws (No observations) by observations instead of the models of the theory containing action descriptions, natural laws and observations by appealing to an example involving a moving object. *We give a different justification in Section 10; we show that by using filtering the resultant theory has the restricted monotonicity property with respect to the observations.*

Although Sandewall's formalism [San89] considers one of the important aspects of narratives (i.e., explaining values of fluents at different time instants by action occurrences), it does not cover the whole issue of hypothetical reasoning in presence of narratives and does not have a notion of planning from the current situation, which we do in this paper. On the other hand Sandewall considers actions with durations, which is not considered in this paper.

11.2 *McCarthy's formulation of situation calculus with concurrent events and narratives*

In this section we relate our work with McCarthy's draft titled 'Situation Calculus with concurrent events and Narrative' [McC95] available through his home page. In our comparison we make several quotes from his draft.

- McCarthy says:

  > Situations in a narrative are partially ordered in time. The real situations are totally ordered, but the narrative does not include full information about this ordering. ...

  We agree with the above statement. But, in our formalization of a narrative description with partially ordered situations, instead of having models encoding this partial ordering, we will have several models each encoding a possible total ordering of the situations.

- He also says:

  > In a narrative, it is not necessary that what is said to hold in a situation be a logical consequence (even nonmonotonically) of what was said to hold about a previous situation and known common sense facts about the effects of events. ... Nevertheless, some narratives are anomalous. ... We want to introduce a concept of *proper narrative*, but it isn't clear exactly what it should be. The fluents holding in a new situation should be reasonable outcomes of the events that have been reported, except for those fluents that are newly asserted, e.g. ...

  In our formulation, if something is said to hold in a situation, it may not logically follow from the rest of the description by itself; but we incorporate the assumption that it must have an explanation. Since this assumption is part of our formulation, if something is said to hold in a situation, it is a logical consequence of action occurrences leading up to this situation. But, all of these action occurrences need not be explicitly stated in the original description, some may be abduced from the observations that are made. Of course, a description may have several models each suggesting a different action occurrence.

  We believe that our formulation suggests a definition for *proper narrative*, which is not defined in [McC95], and in our formulation we require that models of the description encode proper narratives, and lack of information leading to not having a unique proper narrative results in multiple models of the description.

- In the beginning we rule out actions, such as tossing a coin, whose effects are non-deterministic. But later (in Section 9) we say how such actions can be incorporated into our formalism.

- McCarthy also says:

  > Perhaps narrative seems easy, since it is not yet clear what facts must be included in a narrative and what assertions should be inferable from a narrative.

We hope that in the previous sections we have given a satisfactory answer to this.

- McCarthy talks about elaborations of actions in a narrative by finer actions and sub-narratives. We do not consider them here. McCarthy also talks about the role of contexts in formalizing narratives. We do not consider them here.
- McCarthy informally discusses an interesting example involving narratives. The example has two versions, one consisting of two narratives occuring concurrently and independently, and another where the same two narratives have a few points of interaction. In the following example, we show how the second version of the example can be represented in $\mathcal{L}$. (The representation of the first version is a subset of the representation of the second version.)

Example 7 *(Glasgow, London, Moscow and New York [McC95])* The story is the following: there is a person called Daddy in New York who is stacking blocks. At the same time, there is a person called Junior who is in Glasgow and has tickets for and is traveling from Glasgow to Moscow via London. Arriving into London, Junior loses his ticket and sends a telegram to Daddy asking for money. Daddy then sells one of his blocks to get the money and sends it to Junior. Junior then gets the money, buys a London-Moscow ticket and finishes his trip.

Part of a domain description in $\mathcal{L}$ which captures this story would be the following:

Causal law schemas:

$Flies(Prsn, X, Y)$ **causes** $At(Prsn, Y)$ **if** $At(prsn, X), Has(Prsn, tkt(X, Y))$

$Flies(Prsn, X, Y)$ **causes** $\neg At(Prsn, X)$ **if** $At(prsn, X), Has(Prsn, tkt(X, Y))$

$Lose(Prsn, X)$ **causes** $\neg Has(Prsn, X)$

$Buys(Prsn, X)$ **causes** $Has(Prsn, X)$ **if** $Has(Prsn, money)$

$Sells(Prsn, blk1)$ **causes** $Has(Prsn, money)$

$Sells(Prsn, blk2)$ **causes** $Has(Prsn, money)$

$Sends(Prsn1, Prsn2, money)$ **causes** $Sent(Prsn1, Prsn2, money)$ **if** $Has(Prsn1, money)$

$Receives(Prsn1, Prsn2, money)$ **causes** $Has(Prsn1, money)$ **if** $Sent(Prsn2, Prsn1, money)$

$Sends(Prsn1, Prsn2, tlgrm)$ **causes** $Sent(Prsn1, Prsn2, tlgrm)$ **if** $Has(Prsn1, money)$

$Receives(Prsn1, Prsn2, tlgrm)$ **causes** $Rcvd(Prsn1, tlgrm)$ **if** $Sent(Prsn2, Prsn1, tlgrm)$

$Stack(Prsn, X, Y)$ **causes** $On(X, Y)$ **if** $Clear(X), Clear(Y)$

Fluent facts about the initial situation:

Junior:

$At(jr, gw)$ **at** S_0

$Has(jr, gw\_ldn\_tkt)$ **at** S_0

$Has(jr, ldn\_mscw\_tkt)$ **at** S_0

Daddy:

$At(ddy, ny)$ **at** S_0'

$\neg Has(ddy, money)$ **at** S_0'

$Has(ddy, blk1)$ **at** S_0'

$Has(ddy, blk2)$ **at** S_0'

$On(blk1, tbl)$ **at** S_0'

$On(blk2, tbl)$ **at** S_0'

Occurrences of actions:

Junior:

$Flies(jr, gw, ldn)$ **occurs_at** S_0

$Lose(jr, tkt(ldn, mscw))$ **occurs_at** S_1

$Sends(jr, ddy, tlgrm)$ **occurs_at** S_2

Daddy:

$Stacks(ddy, blk2, blk1)$ **occurs_at** S'_0

$Receives(ddy, jr, tlgrm)$ **occurs_at** S'_1

$Sells(ddy, blk1)$ **occurs_at** S'_2

$Sends(ddy, jr, money)$ **occurs_at** S'_3

Junior:

$Receives(jr, ddy, money)$ **occurs_at** S_3

$Buys(jr, ldn\_mscw\_tkt)$ **occurs_at** S_4

$Flies(jr, ldn, mscw)$ **occurs_at** S_5

Ordering on situations:

S_0 **precedes** S_1

S'_0 **precedes** S'_1

S_1 **precedes** S_2

S_2 **precedes** S'_1

S'_1 **precedes** S'_2

S_2 **precedes** S_3

S'_2 **precedes** S'_3

S'_3 **precedes** S_3

S_3 **precedes** S_4

S_4 **precedes** S_5

The interesting point in this example is that parts of the sub-narratives occur simultaneously and thus the exact order of all events is not known. In our formalization, there will be multiple different models each with one of the possible total orders. □

11.3 Miller and Shanahan's circumscriptive approach

In [MilSha94] Miller & Shanahan introduced a formalization of narratives using the situation calculus and circumscription. Their formalization of narrative has many similarities to our work. Their function *State* which maps time points (real numbers) to situations (constructed using the function Res, an initial situation S_0 and action constants) is similar to our Σ which maps situation constants to sequences of actions.

However, their formalization requires [10] that the domain description include all occurrences of actions and fluent facts are restricted to be about the initial situation only.

Our approach is then more general with respect to the above restrictions. And further, propositional combinations of fluent facts, occurrence facts and precedence facts are also allowed in our formalization. Our semantics incorporates the abductive reasoning necessary to make conclusions regarding occurrences of actions and values of fluents in different situations, even if they are not explicitly stated in the domain description. On the other hand, [MilSha94] contains discussions on allowing divisible and overlapping actions, which we do not discuss in this paper.

11.4 Kakas and Miller's $\mathcal{E}$

Kakas and Miller [KakMil96] have introduced a high level language $\mathcal{E}$ based on event-calculus. A domain description in their language consists of c-propositions, h-propositions and i-propositions, which are similar to causal laws, atomic occurrence facts, and atomic fluent facts in our language. The time points in their corresponds to situation constants in our language.

The major differences between their work and that of ours are as follows:

– In our language the ordering between situation constants is part of the domain description. But in their language it is not part of the domain description, but rather part of the domain language.

– Hypothetical reasoning in $\mathcal{E}$ is simulated by defining Δ-sequences (which are hypothetical time points) corresponding to each hypothetical situations, and for a time point between two consecutive hypothetical situations; and then defining the ordering between these Δ-sequences.

  For example, the Δ-sequence $\langle\langle |load|, load, |shoot|, shoot\rangle\rangle$ corresponds to the hypothetical situation $Res(Shoot, Res(Load, S_0))$, and the Δ-sequence $\langle\langle |load|, load, |shoot|\rangle\rangle$ corresponds to the time point between the hypothetical situations $Res(Load, S_0))$ and $Res(Shoot, Res(Load, S_0))$. The ordering between Δ sequences is then done in the obvious manner. For example, they

[10] They do point out that these restrictions can be weakened using abduction.

have: $\langle\langle|load|\rangle\rangle$ precedes $\langle\langle|load|, load\rangle\rangle$ precedes $\langle\langle|load|, load, |shoot|\rangle\rangle$ precedes $\langle\langle|load|, load, |shoot|, shoot\rangle\rangle$.

Notice that the number of delta sequences will be infinite, thus making the domain language infinite. (They can be finitely expressed in a logical language such as logic programming though.)

We believe that our approach of incorporating narratives and hypothetical reasoning in $\mathcal{L}$ is more elegant than the simulation of hypothetical reasoning in the event calculus based $\mathcal{E}$.

- In $\mathcal{E}$, information about action occurrences (their h-propositions) is assumed to be complete. We do not have such an assumption. Although their assumption seems to be very restrictive, they later simulate incompleteness by introducing a notion of explanation that restores inconsistency in a theory where the completeness assumption makes it (the theory) inconsistent. They first start with explanation consisting of action occurrences (h-explanation), and then incorporate explanation consisting of c-propositions (our causal laws). *In our formalization we assume our set of causal laws to be complete.*

  They then consider projection domain descriptions where t-propositions (our atomic fluent facts) are only allowed about the initial situation (such t-propositions are referred to as i-propositions), and discuss how to incorporate observations about fluent values non-initial time points (such t-propositions are referred to as o-propositions). They incorporate observations by first explaining in terms of additional i-propositions (called i-explanations) and then explain in terms of both i-propositions and h-propositions (called ih-explanations), where the h-propositions are used to restore consistency.

  In contrast, our domain descriptions allow incomplete information about action occurrences, values of fluents at both initial and non-initial situations, and a partial specification of the ordering between the situation constants. *Our formulation is such that each model of the domain description fills in the missing action* occurrences, and has a total ordering between the situation constants. (The ordering between time-points in $\mathcal{E}$ cannot be a total ordering as it also orders the hypothetical time-points.)

  Thus we incorporate the concept of ih-explanation, i-explanation and h-explanation in the object language itself without resorting to restoring consistency or meta-level explanation. But, we do not allow explanations through c-propositions/causal laws. We consider them to be fixed.

- Finally, we believe our notions of *current situation* and *planning from the current situation* to be very important. These notions are not considered in [KakMil96].

11.5 Pinto and Reiter's actual and legal situations

Pinto and Reiter in [PinRei93] were among the first ones to introduce a time line to situation calculus, allowing not only the hypothetical reasoning inherent to situation calculus but also the expression of event occurrences and values of fluents at different time instants. *Their approach, based on Reiter's [Rei91] solution to the frame problem, was restricted to non-concurrent actions, and appealed to circumscription to minimize occurrences.* In their formalization they introduced a predicate *actual* on situations that characterized situations that lie on the path that describes the world's real evolution. Unlike the formalism in $\mathcal{L}$ [BarGelPro97], they do not have the notion of a *current* situation and the notion of *planning from the current situation.* Their notion of planning [11] then boils down to finding actual situations that satisfy a given goal. But, because of the minimization of *occurs* in the actual line, the possible existence of new situations – ones not dictated by the given axiomatization, is eliminated; resulting in not being able to find new situations where the goal is satisfied. Reiter refers to this as the 'premature minimization problem' [Rei96] and points out that Miller and Shanahan's formalism also suffers from the same problem. Even though we also minimize action occurrences, *our formalism does not suffer from this problem*, because our minimization is only used to define the current situation and we are free to do planning (by doing hypothetical reasoning) from the current situation.

Pinto in his thesis [Pin94] builds up on the work in [PinRei93], and discusses many new issues including concurrent actions, continuous actions and natural events; but the core of the approach in his thesis – the use of Reiter's solution to the frame problem, definition of actual, and minimization of occurs, remains unchanged from [PinRei93], and *thus suffers from the same problem of 'premature minimization' and inability to make executable plans.*

In their later work [PinRei95,Rei96,Pin96], Pinto and Reiter *abandon minimization of occurrences*, and in [Rei96] and [Pin96] they allow the possibility of several 'hypothetical actual branches' – whose situations are now referred to as 'legal' instead of 'actual', that *restores the capability of making plans.* But even here, there approach to planning is based on finding a legal situation where the goal is satisfied. *It is not clear to us if this notion of planning is useful in the scenario of an agent in a dynamic world. The agent does not need a plan from the initial situation – as Pinto and Reiter's approach would give us; rather it needs a plan from the <u>current</u> situation.*

[11] The issue of planning is not directly mentioned in [PinRei93], but based on the discussion in [Rei96] on 'premature minimization' and the notion of planning there, it seems that this is what Pinto and Reiter had in mind.

We will now try to list the major differences between our approach and that of Pinto and Reiter's.

- Our approach is based on a high level language, $\mathcal{L}$, which is restrictive compared to that of Pinto and Reiter's. For example, our domain description does not allow a statement that has both holds and occurs in them. Moreover, we do not allow triggers, and natural events, which are considered in [Pin94,Rei96,Pin96].
- In our formalism the only restriction on the legality of actions in a branch is due to the executability conditions. Thus our branches consists of 'legal' situations and are 'hypothetical actual branches' in their sense. In [Rei96] natural actions play a role in defining the legality of a situation.
- In [Rei96,Pin96] 'legal situations' replace the notion of 'actual situation'. Such is not the case in our formalism. While all situations in our branches are legal, only a finite set of situations in a finite line – called *the actual line*, starting from the initial situation and ending at the 'current' situation are considered to be 'actual'. *We minimize the occurrences of actions in this actual line.* In contrast, while in [PinRei93,Pin94] Pinto and Reiter face the problem of 'premature minimization' while minimizing occurs, in later works [Pin96,Rei96] they abandon[12] minimization of occurrences. *Also, it seems*[13] *Pinto in [Pin96] does not allow statements about values of fluents at non-initial situations. Allowing such statements and abducing additional information from them in the object language itself is an important aspect of our work.*

  We believe and in [PinRei93,Pin94] Pinto and Reiter agree with us on the intuitiveness of minimizing action occurrences on the description of the evolution of the world up until the current situation. We avoid the problem of 'premature minimization' because of carefully minimizing only the action occurrences in the 'actual line', while they minimize action occurrences without restraint. Our notion of the 'current situation' comes in handy.
- Pinto and Reiter stay within first order logic as much as possible. In [PinRei93,Pin94] all their axiomatization except the induction axiom, and the minimization of occurs at the end, are in first order logic; and later [Pin96,Rei96] all their minimization except the induction axiom is in first

[12] In [Pin96], Pinto shows how to characterize within the first order logical language preferences between action sequences based on some minimality criteria. Although, as he shows, such a characterization can be used to define particular kind of minimal plans, it is not clear how such minimizations can be used to minimize action occurrences in the actual line.

[13] This is based on the remark Pinto makes in the 'Conclusion' section of [Pin96], where he says he would like to extend his work to study the effect of adding statements regarding the value of fluents at different points in time.

order logic.

Our formalization here is based on circumscription, and seemingly makes it easier to allow constraints and even actions with uncertain effects together with effect axioms. Pinto concurs and in [Pin96] says:

> An important advantage of the circumscriptive approach is that the solution to the frame problem is more general and can be applied to theories that include state constraints as well as effect axioms. However, the correctness of solutions based on circumscription is hard to assess.

In our case since we show our circumscriptive formulation to be correct with respect to the semantics of the high level language $\mathcal{L}$; it makes it easier to asses the correctness of solutions based on our approach.

Finally, based on the above comparison between our work and the work by Pinto and Reiter we would like to do the following in the future:

- Carefully extend our language to allow additional constructs such as natural events, continuous actions, and various kinds of action occurrences beyond the simple ones that we have – such as non preventable occurrences, conditional occurrences, eventual occurrences, and triggers occurrences. One of us [BarLob96] has already participated in extending $\mathcal{L}$ to allow triggers so as to formulate active databases; but this high level formulation has not been axiomatized using circumscription.
- Introduce the notion of 'current situation' and 'planning from the current situation' to Pinto and Reiter's formalism; and have separate notions of 'legal' and 'actual' situations, and allow minimization of occurrences of actions in the actual line. With these additional notions we would then like to compare their axiomatization and our circumscriptive axiomatization on domain descriptions in $\mathcal{L}$, similar to the comparison in [Kar93].

12 Appendix: Proof of correctness of the translation T

In this section we will use the following notation:

$$\mathcal{M}[\![\pi]\!]$$

will stand for the set of tuples which belong to the extent of predicate π in interpretation $\mathcal{M}$. With functions we use

$$\mathcal{M}[\![\varphi]\!](\tau)$$

to denote the object which function φ maps τ into in interpretation $\mathcal{M}$.

In lemmas and proofs, by a model of a single block or a sub-theory we will mean a model of a block or sub-theory plus the *Framework Axioms*.

Readers who are familiar with logic programming might have noticed that the blocks defining $Prefix\_eq$ and the other relations on sequences are very similar to the typical representation of these relations in logic programming. The following lemma, and those which are similar, are analogous to a lemma due to Marek and Subrahmanian [MS89] which is very useful in logic programming. Our lemmas are not general results, however.

Lemma 5 Let $\mathcal{M}$ be a model of $B_{\text{prefix\_eq}}$. Then, for all $a, \alpha, \beta,$ [14]

$$\langle \alpha, \epsilon \rangle \in \mathcal{M}[\![Prefix\_eq]\!] \Rightarrow \alpha = \epsilon \tag{21}$$

$$\begin{array}{c} [\ \alpha \neq a \circ \beta\ \wedge \langle \alpha, a \circ \beta \rangle \in \mathcal{M}[\![Prefix\_eq]\!]\ ] \\ \Rightarrow \\ [\ \langle \alpha, \beta \rangle \in \mathcal{M}[\![Prefix\_eq]\!]] \end{array} \tag{22}$$

Proof:

Note that the axioms in $B_{\text{prefix\_eq}}$ are definite clauses, so their conjunction, which we will denote by $A(Prefix\_eq)$, is a formula definite [Lif94, Sec. 3.5] in $Prefix\_eq$. By Corollary 3.5.3 from [Lif94],

$$B_{\text{prefix\_eq}} \equiv CIRC[A(Prefix\_eq); Prefix\_eq] \tag{23}$$

is equivalent to

$$\forall \alpha, \beta.[Prefix\_eq(\alpha, \beta) \equiv \forall p(A(p) \supset p(\alpha, \beta))]. \tag{24}$$

Let us show that (21) holds. Let $\mathcal{M}$ be a model of (23) such that $\mathcal{M} \models Prefix\_eq(\alpha, \epsilon)$ for some α. Suppose that $\alpha \neq \epsilon$. It is easy to see that the model $\mathcal{M}'$ differing from $\mathcal{M}$ only in that $\langle \alpha, \epsilon \rangle \notin \mathcal{M}[\![Prefix\_eq]\!]$ satisfies $A(Prefix\_eq)$ and is therefore preferable to $\mathcal{M}$. But this contradicts our assumption that $\mathcal{M}$ is a model of (23). Therefore, $\alpha = \epsilon$.

Let us show that (22) holds. Let $\mathcal{M}$ be a model of (23) such that $\mathcal{M} \models Prefix\_eq(\alpha, a \circ \beta)$ for some sequences $\alpha \neq a \circ \beta$. Suppose that $\mathcal{M} \models$

[14] Recall that constant ϵ denotes the empty sequence.

$\neg Prefix\_eq(\alpha, \beta)$. Consider a predicate P with extent $\mathcal{M}[\![Prefix\_eq]\!] \setminus \langle \alpha, a \circ \beta \rangle$. Clearly, $A(P)$ is satisfied while $A(P) \supset P(\alpha, a \circ \beta)$ is not. Hence our assumption is wrong and $\mathcal{M} \models Prefix\_eq(\alpha, \beta)$. Therefore, (22) is satisfied by all models of (23). $\square$

Lemma 6 Let $\mathcal{M}$ be a model of $B_{\text{prefix\_eq}}$ and α, β be sequences of actions. Then $\mathcal{M} \models Prefix\_eq(\alpha, \beta)$ iff $\alpha \preceq \beta$.

Proof:

$(\Rightarrow)$

Let $\mathcal{M}$ be a model of $B_{\text{prefix\_eq}}$. We will show that for all α, β, if $\mathcal{M} \models Prefix\_eq(\alpha, \beta)$ then $\alpha \preceq \beta$ by induction on the length of β. Let $n \geq 0$ be the length of α and $m \geq 0$ be the length of β.

Base case: $m = 0$. Then, $\beta = \epsilon$. Suppose that $n > m$. Then, $\alpha \neq \epsilon$ and by (21) we get a contradiction. Thus, $n \leq m$. Since $m = 0$ and $n \geq 0$, we have that $n = 0$, i.e. $\alpha = \beta = \epsilon$. Therefore, $\alpha \preceq \beta$.

Inductive hypothesis: for all β of length $m \leq k$ and for all α, if $\mathcal{M} \models Prefix\_eq(\alpha, \beta)$ then $\alpha \preceq \beta$.

Induction: let $a \circ \beta$ be an arbitrary sequence of length $k+1$. Let us show that if $\mathcal{M} \models Prefix\_eq(\alpha, a \circ \beta)$ then $\alpha \preceq a \circ \beta$. From $\mathcal{M} \models Prefix\_eq(\alpha, a \circ \beta)$ and (22) we have that $\mathcal{M} \models Prefix\_eq(\alpha, \beta)$. By the inductive hypothesis, this implies that $\alpha \preceq \beta$. By definition of $\preceq$, $\alpha \preceq a \circ \beta$ follows from $\alpha \preceq \beta$.

$(\Leftarrow)$

Let $\mathcal{M}$ be a model of $B_{\text{prefix\_eq}}$. For all α, β, if $\alpha \preceq \beta$ then, by definition of $\preceq$, $\beta = B_n \circ \ldots \circ B_1 \circ \alpha$ for some $B_1, \ldots, B_n$ and $n \geq 0$.

Now, let us show by induction on n that $\mathcal{M} \models Prefix\_eq(\alpha, B_n \circ \ldots \circ B_1 \circ \alpha)$ for all α.

Base case: $n = 0$. We need to show $\mathcal{M} \models Prefix\_eq(\alpha, \alpha)$. But this is an axiom of $B_{\text{prefix\_eq}}$.

Induction hypothesis: for any $n \leq k$ and α, $\mathcal{M} \models Prefix\_eq(\alpha, B_n \circ \ldots \circ B_1 \circ \alpha)$.

Induction step: we need to show that $\mathcal{M} \models Prefix\_eq(\alpha, B_{k+1} \circ B_k \circ \ldots \circ B_1 \circ \alpha)$. By the induction hypothesis we have that $\mathcal{M} \models Prefix\_eq(\alpha, B_k \circ \ldots \circ B_1 \circ \alpha)$. Then, by axiom

$$(\forall \alpha, \beta, a). Prefix\_eq(\alpha, \beta) \supset Prefix\_eq(\alpha, a \circ \beta)$$

of $B_{\text{prefix\_eq}}$, we have that $\mathcal{M} \models Prefix\_eq(\alpha, B_{k+1} \circ B_k \circ \ldots \circ B_1 \circ \alpha)$. □

Lemma 7 Let $\mathcal{M}$ be a model of $B_{\text{subsequence}}$. Then for all α, β,

$$\begin{array}{c} a \neq b \ \wedge \langle a \circ \alpha, b \circ \beta \rangle \in \mathcal{M}[\![Subsequence]\!] \\ \Rightarrow \\ \langle a \circ \alpha, \beta \rangle \in \mathcal{M}[\![Subsequence]\!] \end{array} \tag{25}$$

$$\begin{array}{c} \langle a \circ \alpha, a \circ \beta \rangle \in \mathcal{M}[\![Subsequence]\!] \\ \Rightarrow \\ \langle \alpha, \beta \rangle \in \mathcal{M}[\![Subsequence]\!] \end{array} \tag{26}$$

Proof:

Similar to the proof of Lemma 5. □

Lemma 8 Let $\mathcal{M}$ be a model of $B_{\text{subsequence}}$. Then, for all α, β, $\mathcal{M} \models Subsequence(\alpha, \beta)$ iff $\alpha \ll \beta$.

Proof:

This can be proven by induction on the lengths of α and β in a manner similar to the proof of Lemma 6. □

Lemma 9 Let $\mathcal{M}$ be a model of $B_{\text{concatenate}}$. Then, for all α, β, γ,

$$[\langle \alpha, \epsilon, \gamma \rangle \in \mathcal{M}[\![Concatenate]\!]\ ] \Rightarrow \alpha = \gamma \tag{27}$$

$$\begin{array}{c} [\langle \alpha, a \circ \beta, b \circ \gamma \rangle \in \mathcal{M}[\![Concatenate]\!]\ ] \\ \Rightarrow \\ [\langle \alpha, \beta, \gamma \rangle \in \mathcal{M}[\![Concatenate]\!] \ \wedge a = b] \end{array} \tag{28}$$

Proof:

Similar to proof of Lemma 5.

□

Lemma 10 Let $\mathcal{M}$ be a model of $B_{\text{concatenate}}$. Then, for all α,β,γ, $\mathcal{M} \models Concatenate(\alpha, \beta, \gamma)$ iff $\gamma = \alpha \cdot \beta$.

Proof:

This can be proven by induction on the length of the second argument in a manner similar to the proof of Lemma 6. □

In the following lemma we will use the following notation: let A be an action and F be a fluent, formula $H^{+(-)}_{A,F}(\alpha)$ will stand for the disjunction of all the premises of axioms of the form:

$$H(p_1, \alpha) \ \wedge \ldots \ \wedge H(p_n, \alpha) \supset Causes^{+(-)}(a, f, \alpha)$$

such that $a = A$ and $f = F$. For instance, if axioms $Holds(P, \alpha) \supset Causes^{+}(A, F, \alpha)$ and $Holds(Q, \alpha) \supset Causes^{+}(A, F, \alpha)$ are all the axioms with consequent $Causes^{+}(A, F, \alpha)$, then $H^{+}_{A,F}(\alpha)$ stands for $Holds(P, \alpha) \vee Holds(Q, \alpha)$.

Lemma 11 Let D be a domain description. Block

$$\begin{array}{l} \{\min \ Causes^{+(-)} \ : \\ \qquad H(P_1, \alpha) \ \wedge \ldots \ \wedge H(P_n, \alpha) \supset Causes^{+(-)}(A, F, \alpha) \\ \qquad\qquad\qquad\qquad (\text{for each } A \ \mathbf{causes} \ \ F \ \mathbf{if} \ \ P_1, \ldots, P_n \in D) \\ \} \end{array} \tag{29}$$

is equivalent to the conjunction of formulas of the form:

$$\forall \alpha . [H^{+(-)}_{A,F}(\alpha) \equiv Causes^{+(-)}(A, F, \alpha)] \tag{30}$$

one for each pair of action-fluent constants.

Proof:

It follows from Proposition 3.1.1. of [Lif94] which shows under what conditions a circumscription formula can be simplified by *predicate completion*.

□

Lemma 1

Part 1: For every causal model Ψ of D_l there exists a model $\mathcal{M}_{\mathrm{SC}}$ of $SC(D_l)$ such that for all f, α:

$$f \in \Psi(\alpha) \;\Leftrightarrow\; \mathcal{M}_{\mathrm{SC}} \models Holds(f, \alpha) \tag{31}$$

Part2: For every model $\mathcal{M}_{\mathrm{SC}}$ of $SC(D_l)$ there exists a causal model Ψ of D_l such that (31) holds.

Proof:

Part 1: Let Ψ be a causal model of D_l and let $\mathcal{M}_{\mathrm{SC}}$ be an interpretation of $SC(D_l)$ such that for all a, f, α

$$\langle f, \alpha\rangle \in \mathcal{M}_{\mathrm{SC}}[\![Holds]\!] \text{ iff } f \in \Psi(\alpha)$$

and

$$\langle a, f, \alpha\rangle \in \mathcal{M}_{\mathrm{SC}}[\![Causes^{+(-)}]\!] \text{ iff} f \in E_a^{+(-)}(\Psi(\alpha)).$$

We will show that $\mathcal{M}_{\mathrm{SC}}$ is a model of $SC(D_l)$.

We will show first that the blocks defining $Causes^{+(-)}$ are satisfied. By Lemma 11 it is sufficient to show that for any action a, fluent f and sequence α,

$$\forall\alpha.[H_{a,f}^{+(-)}(\alpha) \equiv Causes^{+(-)}(a, f, \alpha)]$$

Let $A_0, F, \vec{A}$ be an action, fluent and sequence respectively. Suppose $H_{A_0,F}^{+(-)}(\vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$. Then, by construction of $\mathcal{M}_{\mathrm{SC}}$, there is a causal law A_0 **causes** F **if** $P_1, \ldots, P_n \in D$ such that $H(P_i, \vec{A})$ holds for $i = 1, \ldots, n$. Again by construction of $\mathcal{M}_{\mathrm{SC}}$, this implies that $P_i \in \Psi(\vec{A})$ for $i = 1, \ldots, n$, which in turn implies that $F \in E_{A_0}^{+(-)}(\Psi(\vec{A}))$. By construction of $\mathcal{M}_{\mathrm{SC}}$, $F \in E_{A_0}^{+(-)}(\Psi(\vec{A}))$ implies that $Causes^{+(-)}(A_0, F, \vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$.

Suppose now that $Causes^{+(-)}(A_0, F, \vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$ for an arbitrary action A_0, fluent F and sequence $\vec{A}$. Then, by construction of $\mathcal{M}_{\mathrm{SC}}$, $F \in E_{A_0}^{+(-)}(\Psi(\vec{A}))$. This implies that there is a causal law A_0 **causes** F **if** $P_1, \ldots, P_n \in D$ such that $P_i \in \Psi(\vec{A})$ for $i = 1, \ldots, n$. Again by construction of $\mathcal{M}_{\mathrm{SC}}$, it follows that $H(P_i, \vec{A})$ holds for $i = 1, \ldots, n$. Therefore, $H^{+(-)}_{A_0,F}(\vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$.

Now, let us show that the axiom

$$\neg Causes^{+}(a, f, \alpha) \ \wedge \neg Causes^{-}(a, f, \alpha) \supset [Holds(f, \alpha) \equiv Holds(f, a \circ \alpha)]$$

is satisfied by $\mathcal{M}_{\mathrm{SC}}$. Let $A_0, F, \vec{A}$ be an arbitrary action, fluent and sequence, respectively, such that LHS of the axiom holds in $\mathcal{M}_{\mathrm{SC}}$. By construction of $\mathcal{M}_{\mathrm{SC}}$, we have that $F \notin E^{+}_{A_0}(\Psi(\vec{A})) \cup E^{-}_{A_0}(\Psi(\vec{A}))$. By definition of causal model, this implies that $F \in \Psi(\vec{A})$ iff $F \in \Psi(\vec{A} \circ A_0)$. Therefore, by construction of $\mathcal{M}_{\mathrm{SC}}$, $Holds(F, \vec{A}) \equiv Holds(F, A_0 \circ \vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$.

Now consider the axioms

$$Causes^{+}(a, f, \alpha) \supset Holds(f, a \circ \alpha)$$
$$Causes^{-}(a, f, \alpha) \supset \neg Holds(f, a \circ \alpha)$$

Let $A_0, F, \vec{A}$ be an arbitrary action, fluent and sequence, respectively, such that $Causes^{+}(A_0, F, \vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$. Then, by construction of $\mathcal{M}_{\mathrm{SC}}$, $F \in E^{+}_{A_0}(\Psi(\vec{A}))$. Given our assumption that Ψ is defined on all sequences (Section 7), we have that $F \notin E^{+}_{A_0}(\Psi(\vec{A})) \cap E^{-}_{A_0}(\Psi(\vec{A}))$, thus by definition of Ψ, $F \in \Psi(\vec{A} \circ A_0)$. Therefore, by construction of $\mathcal{M}_{\mathrm{SC}}$, $Holds(F, A_0 \circ \vec{A})$ holds.

A similar argument holds for the second axiom.

Part 2: Let $\mathcal{M}_{\mathrm{SC}}$ be a model of $SC(D_l)$. Let Ψ be a transition function such that (31) holds. We show Ψ to be a causal model of D_l, i.e., for any a, α

$$\Psi(\alpha \circ a) = \Psi(\alpha) \cup E^{+}_{a}(\Psi(\alpha)) \setminus E^{-}_{a}(\Psi(\alpha))$$

We show this is indeed the case by induction on the length of the sequence.

Base Case: The length of α is 0. We need to show that for any a,

$$\Psi([a]) = \Psi([\,]) \cup E^{+}_{a}(\Psi([\,])) \setminus E^{-}_{a}(\Psi([\,]))$$

where $f \in \Psi([\,])$ iff $Holds(f, \epsilon)$ by construction of Ψ.

a) $\Psi([a]) \supseteq \Psi([\,]) \cup E^{+}_{a}(\Psi([\,])) \setminus E^{-}_{a}(\Psi([\,]))$.

Let F and A be an arbitrary fluent and action resp., such that $F \in \Psi([\,]) \cup E_A^+(\Psi([\,])) \setminus E_A^-(\Psi([\,]))$. Let us show that $F \in \Psi([A])$. Clearly, $F \notin E_A^-(\Psi([\,]))$, hence there is no causal law A **causes** $\neg F$ **if** $Q_1, \ldots, Q_m \in D_l$ such that the preconditions hold in $\Psi([\,])$. By construction of Ψ this means that $\mathcal{M}_{\mathrm{SC}} \models H(Q_i, \epsilon)$ does not hold for every $i = 1, \ldots, m$, for all such causal laws. By Lemma 11, this implies that

$$\mathcal{M}_{\mathrm{SC}} \models \neg Causes^-(A, F, \epsilon). \tag{32}$$

There are two cases of interest

i) $F \in E_A^+(\Psi([\,]))$
ii) $F \in \Psi([\,])$ and $F \notin E_A^+(\Psi([\,]))$

If (i) is the case, then there must be a causal law A **causes** F **if** $P_1, \ldots, P_n \in D_l$ such that the preconditions hold in $\Psi([\,])$. By construction of Ψ, this implies that $\mathcal{M}_{\mathrm{SC}} \models H(P_i, \epsilon)$ for each $i = 1, \ldots, n$. Therefore, $\mathcal{M}_{\mathrm{SC}} \models Causes^+(A, F, \epsilon)$ and by axiom $Causes^+(a, f, \alpha) \supset Holds(f, a \circ \alpha)$ we have that $\mathcal{M}_{\mathrm{SC}} \models Holds(F, A \circ \epsilon)$. Thus, by construction of Ψ, $F \in \Psi([A])$.

If (ii) is the case, then by construction of Ψ we must have $\mathcal{M}_{\mathrm{SC}} \models Holds(F, \epsilon)$. Furthermore, $F \notin E_A^+(\Psi([\,]))$ implies there is no causal law A **causes** F **if** $P_1, \ldots, P_n \in D_l$ such that the preconditions hold in $\Psi([\,])$. Hence, $\mathcal{M}_{\mathrm{SC}} \models H(P_i, \epsilon)$ does not hold for each $i = 1, \ldots, n$, for all causal laws of this form; and $\mathcal{M}_{\mathrm{SC}} \models \neg Causes^+(A, F, \epsilon)$. From this, (32), and axiom

$$\neg Causes^+(a, f, \alpha) \ \wedge \neg Causes^-(a, f, \alpha) \supset [Holds(f, \alpha) \equiv Holds(f, a \circ \alpha)]$$

we conclude that $\mathcal{M}_{\mathrm{SC}} \models Holds(F, A \circ \epsilon)$, and by construction of Ψ that $F \in \Psi([A])$.

b)$\Psi([a]) \subseteq \Psi([\,]) \cup E_a^+(\Psi([\,])) \setminus E_a^-(\Psi([\,]))$.

Let F and A be an arbitrary fluent and action resp., such that $F \in \Psi([A])$. By construction of Ψ, $\mathcal{M}_{\mathrm{SC}} \models Holds(F, A \circ \epsilon)$. Thus, $\mathcal{M}_{\mathrm{SC}} \models \neg Causes^-(A, F, \epsilon)$, and by Lemma 11, for every causal law A **causes** $\neg F$ **if** $Q_1, \ldots, Q_m \in D_l$ we have $\mathcal{M}_{\mathrm{SC}} \models H(Q_i, \epsilon)$ does not hold for all $i = 1, \ldots, m$. By construction of Ψ, for every causal law its preconditions do not hold in $\Psi([\,])$ either. Thus, $F \notin E_A^-(\Psi([\,]))$. We again have two cases

i) $\mathcal{M}_{\mathrm{SC}} \models Causes^+(A, F, \epsilon)$
ii) $\mathcal{M}_{\mathrm{SC}} \models \neg Causes^+(A, F, \epsilon) \ \wedge Holds(F, \epsilon)$

If (i) is the case, then by Lemma 11 there is a causal law A **causes** F **if** $P_1, \ldots, P_n \in D_l$ such that $\mathcal{M}_{\mathrm{SC}} \models H(P_i, \epsilon)$ for each $i = 1, \ldots, n$. By construction of Ψ, all the preconditions of such a causal law hold in $\Psi([\,])$. Therefore, $F \in E_A^+(\Psi([\,]))$, and since we assume that Ψ is defined on every sequence (Section 7), $F \notin E_A^-(\Psi([\,]))$. Thus, $F \in \Psi([\,]) \cup E_A^+(\Psi([\,])) \setminus E_A^-(\Psi([\,]))$.

If (ii) is the case then by construction of Ψ, $F \in \Psi([\,])$. We showed above that $F \notin E_A^-(\Psi([\,]))$. Therefore, $F \in \Psi([\,]) \cup E_A^+(\Psi([\,])) \setminus E_A^-(\Psi([\,]))$.

Induction Hypothesis: For all a, α

$$\Psi(\alpha \circ a) = \Psi(\alpha) \cup E_a^+(\Psi(\alpha)) \setminus E_a^-(\Psi(\alpha))$$

Inductive case: This case can be proven by following the same steps as in the proof of the base case. □

Lemma 2

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:

$$(\Psi, \Sigma) \models \phi \;\Leftrightarrow\; \mathcal{M} \models \tau(\phi) \tag{33}$$

and if $\mathcal{M}$ is a model of $T(D)$ then there exists a model (Ψ, Σ) of D such that (33) holds.

Proof:

$(\Rightarrow)$

Let $M = (\Psi, \Sigma)$ be a model of D. Let $\mathcal{M}_M$ be an interpretation of $T(D)$ such that

i) for f, a, α,

$$\mathcal{M}_M[\![Holds]\!] = \{\langle f, \alpha\rangle : f \in \Psi(\alpha)\}$$

$$\mathcal{M}_M[\![Causes^{+(-)}]\!] = \{\langle a, f, \alpha\rangle : f \in E_a^{+(-)}(\alpha)\}$$

ii) for all f, s, s_1, α,

$$\mathcal{M}_M[\![At]\!] = \{\langle f, s\rangle : f \in \Psi(\Sigma(s))\},$$

$$\mathcal{M}_M[\![Occurs]\!] = \{\langle \alpha, s\rangle : \alpha \circ \Sigma(s) \preceq \Sigma(S_N)\},$$

$$\mathcal{M}_M[\![Precedes]\!] = \{\langle s, s_1\rangle : \Sigma(s) \prec \Sigma(s_1)\}$$

iii) for all s, α, $\mathcal{M}_M[\![Sit\_map]\!](s) = \Sigma(s)$

iv) the framework axioms, FACTS, (d), $B_{\text{prefix\_eq}}$, $B_{\text{subsequence}}$, and $B_{\text{concatenate}}$ are satisfied.

We will show that $\mathcal{M}_M$ is a model of $T(D)$.

Axioms (a) and (b) are satisfied by definition of Σ, conditions (iii) and (iv), and Lemma 6.

Note that condition (i) on $Holds$ and $Causes^{+(-)}$ are the same conditions used to show that this interpretation of these predicates is a model of $SC(D_l)$ (See the proof of Lemma 1), thus axioms $SC(D_l)$ are satisfied.

It remains to be shown that $\mathcal{M}_M$ satisfies axioms (e), (f), (g), and that $\mathcal{M}_M$ is a minimal model with respect to the circumscription policy.

Axiom (e): By condition (ii), for all f, s, $\mathcal{M}_M \models At(f, s)$ iff $f \in \Psi(\Sigma(s))$. By cond. (iii) on Sit_map and cond. (i), $f \in \Psi(\Sigma(s))$ implies that $\mathcal{M}_M \models Holds(f, \alpha)$, where α is equal to $\Sigma(s)$.

Axiom (f): By cond. (ii), for all α, s, $\mathcal{M}_M \models Occurs(\alpha, s)$ iff $\alpha \circ \Sigma(s) \preceq \Sigma(S_N)$. By cond. (iii) on Sit_map, $Sit\_map(s)$ and $Sit\_map(S_N)$ are the same sequences as $\Sigma(s)$ and $\Sigma(S_N)$. By correctness of $B_{\text{concatenate}}$ and $B_{\text{prefix\_eq}}$ axiom (f) is satisfied.

Axiom (g): Again, by cond.(ii), for all s_1, s_2, $\mathcal{M}_M \models Precedes(s_1, s_2)$ iff $\Sigma(s_1) \prec \Sigma(s_2)$. Thus, by correctness of $B_{\text{prefix\_eq}}$ and axiom (d), axiom (g) is satisfied.

Finally, the value minimization axiom (c) captures exactly the minimality condition (See Section 5 for a detailed discussion), with respect to ordering *subsequence*, imposed on sequence $\Sigma(S_N)$ by the definition of models of domain descriptions in $\mathcal{L}$.

$(\Leftarrow)$

Let $\mathcal{M}$ be a model of $T(D)$. By Lemma 1 there is a causal model Ψ such that for all f, α,

$$f \in \Psi(\alpha) \text{ iff } \mathcal{M} \models Holds(f, \alpha) \tag{34}$$

Let Ψ be such a causal model and let Σ be a situation assignment such that for all s,

$$\Sigma(s) = \mathcal{M}[\![Sit\_map]\!](s) \tag{35}$$

We show $M = (\Psi, \Sigma)$ is a model of D.

First of all, by axiom

$$Sit\_map(S_0) = \epsilon$$

we have $\Sigma(S_0) = [\,]$, and by axiom

$$Prefix\_eq(Sit\_map(s), Sit\_map(S_N))$$

we have $\Sigma(s)$ is a prefix of $\Sigma(S_N)$ for all situations s in the language of D. Thus, Σ is a situation assignment. Since we assume that Ψ is defined on all sequences, $\Psi(\Sigma(S_N))$ is defined and therefore Ψ and Σ form an interpretation of D.

It only remains to be shown that facts in D are true in M and that there is no $N' = (\Psi, \Sigma')$ such that $\Sigma'(S_N) \ll \Sigma(S_N)$ and N' is a model of D.

a) Suppose $(F \textbf{ at } S) \in D$. Then, $\mathcal{M} \models At(F, S)$ and by axiom (e) $\mathcal{M} \models Holds(F, Sit\_map(S))$. By (35), $Sit\_map(S)$ in $\mathcal{M}$ is the same sequence as $\Sigma(S)$, thus, by (34), $F \in \Psi(\Sigma(S))$. Therefore $(F \textbf{ at } S)$ is true in M. The same argument holds for negated fluent facts.

b) Suppose $(\vec{A} \textbf{ occurs\_at } S) \in D$. Then $\mathcal{M} \models Occurs(\vec{A}, S)$ and by axiom (f) there exists β such that $\mathcal{M} \models (Concatenate(Sit\_map(S), \vec{A}, \beta) \wedge Prefix\_eq(\beta, Sit\_map(S_N))$. By (35) and correctness of $B_{\text{concatenate}}$ (Lemma 10), $\beta = Sit\_map(S) \cdot \vec{A}$. By correctness of $B_{\text{prefix\_eq}}$ (Lemma 6), β is a prefix of sequence $Sit\_map(S_N)$ in $\mathcal{M}$. By condition (35) on Σ, $\beta = \Sigma(S) \circ \vec{A}$ and β is a prefix of $\Sigma(S_N)$. Therefore, $(\vec{A} \textbf{ occurs\_at } S)$ is true in M.

c) Suppose (S_1 **precedes** S_2) $\in D$. Then, $\mathcal{M} \models Prefix(Sit\_map(S_1), Sit\_map(S_2))$. By correctness of $B_{\text{prefix\_eq}}$, axiom (d), and (35), $\Sigma(S_1)$ is a perfect prefix of $\Sigma(S_2)$. Therefore, (S_1 **precedes** S_2) is true in M.

Finally, we need to show there is no interpretation $N' = (\Psi, \Sigma')$ of D such that $\Sigma'(S_N)$ is a subsequence (not equal) of $\Sigma(S_N)$ and N' is a model of D. Suppose there is such a model N'. Then, by part ($\Rightarrow$) of this lemma, there is a model $\mathcal{M}'$ of $T(D)$ which differs from $\mathcal{M}$ only on the interpretation of function Sit_map and such that sequence $\mathcal{M}'[\![Sit\_map]\!](S_N)$ is subsequence (not equal) of $\mathcal{M}[\![Sit\_map]\!](S_N)$. This contradicts our assumption that $\mathcal{M}$ is a model of $T(D)$ since $T(D)$ includes value minimization of $Sit\_map(S_N)$ with respect to ordering subsequence (See Section 5). $\Box$

13 Appendix: Proof of correctness of the translation T_c

Lemma 12 Let $\mathcal{M}$ be a model of B_{embedded}. Then, for all a_1, a_2, α, β,

$$\begin{array}{c} \langle a_1 \circ \epsilon, a_2 \circ \epsilon \rangle \in \mathcal{M}[\![Embedded]\!] \\ \Rightarrow \\ [\forall a. \langle a, a_1 \rangle \in \mathcal{M}[\![In]\!] \Rightarrow \langle a, a_2 \rangle \in \mathcal{M}[\![In]\!]] \end{array} \tag{36}$$

$$\begin{array}{c} \langle a_1 \circ \beta, a_2 \circ \alpha \rangle \in \mathcal{M}[\![Embedded]\!] \\ \Rightarrow \\ [\langle \beta, \alpha \rangle \in \mathcal{M}[\![Embedded]\!] \wedge \langle a_1 \circ \epsilon, a_2 \circ \epsilon \rangle \in \mathcal{M}[\![Embedded]\!]] \end{array} \tag{37}$$

Proof:

Similar to the proof of Lemma 5. $\Box$

Lemma 13 Let $\mathcal{M}$ be a model of B_{embedded}. Then, for all α, β, $\mathcal{M} \models Embedded(\alpha, \beta)$ iff $\alpha \subseteq \beta$.

Proof:

This can be proven by induction on the length of the sequences and Lemma 12. $\Box$

Lemma 14 Let $\mathcal{M}$ be a model of $B_{\text{embsubseq}}$. Then, for all α, β,

$$\begin{array}{c} \langle \alpha, \beta \rangle \in \mathcal{M}[\![Embsubseq]\!] \\ \Rightarrow \\ \exists \gamma.[\langle \alpha, \gamma \rangle \in \mathcal{M}[\![Embedded]\!] \ \wedge \langle \gamma, \beta \rangle \in \mathcal{M}[\![Subsequence]\!]] \end{array} \tag{38}$$

Proof:

Similar to the proof of Lemma 5. □

Lemma 15 Let $\mathcal{M}$ be a model of $B_{\text{embsubseq}}$. Then, for all α, β, $\mathcal{M} \models Embsubseq(\alpha, \beta)$ iff α is an embedded-subsequence in β.

Proof:

($\Rightarrow$)

Let $\mathcal{M}$ be a model of $B_{\text{embsubseq}}$ and let $\vec{A}, \vec{B}$ be sequences such that $\mathcal{M} \models Embsubseq(\vec{A}, \vec{B})$. Then, by (38) we have $\mathcal{M} \models Embedded(\vec{A}, \vec{G}) \ \wedge$ $Subsequence(\vec{G}, \vec{B})$ for some sequence $\vec{G}$. By correctness of B_{embedded} (Lemma 13) we have that $\vec{A}$ is embedded in $\vec{G}$, and by correctness of $B_{\text{subsequence}}$ (Lemma 8) that $\vec{G}$ is a subsequence of $\vec{B}$. Therefore, $\vec{A}$ is an embedded-subsequence of $\vec{B}$.

($\Leftarrow$)

Let $\mathcal{M}$ be a model of $B_{\text{embsubseq}}$ and let $\vec{A}$, $\vec{B}$ be sequences such that $\vec{A}$ is an embedded-subsequence of $\vec{B}$. Then, by definition of embedded-subsequence, there is a sequence $\vec{G}$ such that $\vec{G}$ is a subsequence of $\vec{B}$ and $\vec{A}$ is embedded in $\vec{G}$. By correctness of $B_{\text{subsequence}}$ (Lemma 8) we have that $\mathcal{M} \models Subsequence(\vec{G}, \vec{B})$ and by correctness of B_{embedded} (Lemma 13) that $\mathcal{M} \models Embedded(\vec{A}, \vec{G})$. Therefore, by axiom

$$Embedded(\alpha, \beta) \ \wedge Subsequence(\beta, \gamma) \supset Embsubseq(\alpha, \gamma)$$

of $B_{\text{embsubseq}}$, $\mathcal{M} \models Embsubseq(\vec{A}, \vec{B})$. □

Lemma 16 The block which defines $Inherits^{+(-)}$:

$$
\begin{aligned}
&\{\min \ Inherits^{+(-)} \ : \\
&\qquad In(b,a) \ \wedge Causes^{+(-)}(b,f,\alpha) \supset Inherits^{+(-)}(a,f,\alpha) \\
&\}
\end{aligned}
$$

is equivalent to the following formula

$$
\forall a,f,\alpha.(\exists b.In(b,a) \ \wedge Causes^{+(-)}(b,f,\alpha)) \equiv Inherits^{+(-)}(a,f,\alpha) \quad (39)
$$

Proof:

It follows from Proposition 3.1.1. of [Lif94] which shows under what conditions a circumscription formula can be simplified by *predicate completion.* □

Lemma 17 Let $\mathcal{M}$ be a model of

$$
\begin{aligned}
&\{\min \ Undefined \ : \\
&\qquad Inherits^{+}(a,f,\alpha) \ \wedge Inherits^{-}(a,f,\alpha) \supset Undefined(a \circ \alpha) \\
&\qquad Undefined(\alpha) \supset Undefined(a \circ \alpha) \\
&\}
\end{aligned} \quad (40)
$$

Then, for all a, α,

$$
\begin{gathered}
\langle a \circ \alpha \rangle \in \mathcal{M}[\![Undefined]\!] \\
\Rightarrow \\
[\ ((\exists f).\langle a,f,\alpha\rangle \in \mathcal{M}[\![Inherits^{+}]\!] \ \wedge \langle a,f,\alpha\rangle \in \mathcal{M}[\![Inherits^{-}]\!]) \vee \\
\langle \alpha \rangle \in \mathcal{M}[\![Undefined]\!]\ ]
\end{gathered} \quad (41)
$$

Proof:

Similar to proof of Lemma 5.

□

Lemma 3

Part 1: For every causal model Ψ of D_l there exists a model $\mathcal{M}_{\mathrm{SC}}$ of $SC_c(D_l)$ such that: for all f, α,

$$\begin{aligned} &f \in \Psi(\alpha) \ \Leftrightarrow\ \mathcal{M}_{\mathrm{SC}} \models Holds(f, \alpha), \text{ and} \\ &\Psi(\alpha) \text{ is undefined } \Leftrightarrow\ \mathcal{M}_{\mathrm{SC}} \models Undefined(\alpha) \end{aligned} \tag{42}$$

Part2: For every model $\mathcal{M}_{\mathrm{SC}}$ of $SC_c(D_l)$ there exists a causal model Ψ of D_l such that (42) holds.

Proof:

Part 1: Let Ψ be a causal model of D_l. Let $\mathcal{M}_{\mathrm{SC}}$ be an interpretation of $SC_c(D_l)$ such that

i) $\alpha \in \mathcal{M}_{\mathrm{SC}}[\![Undefined]\!]$ iff $\Psi(\alpha)$ is undefined,
ii) $\langle a, f, \alpha\rangle \in \mathcal{M}_{\mathrm{SC}}[\![Causes^{+(-)}]\!]$ iff f is an immediate effect of a in $\Psi(\alpha)$,
iii) $\langle a, f, \alpha\rangle \in \mathcal{M}_{\mathrm{SC}}[\![Inherits^{+(-)}]\!]$ iff $f \in E_a^{+(-)}(\Psi(\alpha))$
iv) $\langle f, \alpha\rangle \in \mathcal{M}_{\mathrm{SC}}[\![Holds]\!]$ iff $f \in \Psi(\alpha)$

Note that (i) and (iv) correspond to condition (42). We will show $\mathcal{M}_{\mathrm{SC}}$ is a model of $SC_c(D_l)$.

We showed in Lemma 1 that a model built as above satisfies the blocks which define predicates $Causes^+$ and $Causes^-$.

Given the definition of sets $E_a^{+(-)}$, it is clear that condition (iii) corresponds exactly to formula (39), therefore the blocks defining predicates $Inherits^+$ and $Inherits^-$ are satisfied.

Let us show the block defining $Undefined$ is satisfied. Suppose there are $A_0, F, \vec{A}$ such that $Inherits^+(A_0, F, \vec{A}) \ \wedge Inherits^-(A_0, F, \vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$. Then by cond. (iii) we have that $F \in E^+_{A_0}(\vec{A}) \cap E^-_{A_0}(\vec{A})$ and thus this intersection is not empty. This implies that $\Psi(\vec{A})$ is undefined. By this cond. (i) we have that $\mathcal{M} \models Undefined(A_0 \circ \vec{A})$. Therefore the first axiom in the block is satisfied by $\mathcal{M}$. Now, suppose $\Psi(\vec{A})$ is undefined for some $\vec{A}$. Then, since Ψ is prefix closed, for any action a, $\Psi(a \circ \vec{A})$ is undefined. Therefore the second axiom is also satisfied. It is easy to see that the extent of $Undefined$ is not minimal only if there exist a, α such that $Undefined(a \circ \alpha)$ holds while there is no f such that $Inherits^+(a, f, \alpha) \ \wedge Inherits^-(a, f, \alpha)$ holds, nor $Undefined(\alpha)$ holds in $\mathcal{M}$. This however contradicts Lemma 17, thus the extent of $Undefined$ must be minimal and therefore this block is satisfied by

$\mathcal{M}$.

It is easy to see that the axioms

$$\neg Inherits^{+}(a,f,\alpha) \ \wedge \neg Inherits^{-}(a,f,\alpha) \supset [Holds(f,\alpha) \equiv Holds(f,a\circ\alpha)]$$
$$\neg Undefined(a\circ\alpha) \ \wedge Inherits^{+}(a,f,\alpha) \supset Holds(f,a\circ\alpha)$$
$$\neg Undefined(a\circ\alpha) \ \wedge Inherits^{-}(a,f,\alpha) \supset \neg Holds(f,a\circ\alpha)$$

of $SC_c(D_l)$ are satisfied by $\mathcal{M}_{\mathrm{SC}}$.

Part 2: Let $\mathcal{M}_{\mathrm{SC}}$ be a model of $SC_c(D_l)$. Let Ψ be a transition function such that (42) holds. We show Ψ to be a causal model of D_l.

It is clear from minimization of $Undefined$ and the axioms in the block which defines it, that $Undefined(\epsilon)$ does not hold in $\mathcal{M}_{\mathrm{SC}}$. Thus, $\Psi([\,])$ is defined.

Let us show that Ψ is prefix closed. Suppose it is not. Let $\vec{A}$ and $\vec{B}$ be sequences such that $\vec{A}$ is a prefix of $\vec{B}$ and $\Psi(\vec{B})$ is defined while $\Psi(\vec{A})$ is undefined. By construction of Ψ, it follows that $Undefined(\vec{A})$ and $\neg Undefined(\vec{B})$ hold in $\mathcal{M}_{\mathrm{SC}}$. However, from axiom

$$\forall \alpha, a. Undefined(\alpha) \supset Undefined(a\circ\alpha)$$

and the fact that $\vec{A}$ is a prefix of $\vec{B}$, we conclude that $Undefined(\vec{B})$ holds in $\mathcal{M}_{\mathrm{SC}}$. Thus our assumption was wrong and therefore Ψ is prefix closed.

Now, let us show that for all a, α, if $E_a^{+}(\alpha) \cap E_a^{-}(\alpha) \neq \emptyset$ then Ψ is undefined on sequence $a \circ \alpha$.

Suppose $F \in E_{A_0}^{+}(\vec{A})$ and $F \in E_{A_0}^{-}(\vec{A})$ for fluent F, action A_0 and sequence $\vec{A}$. This means F and $\neg F$ are effects of A_0 in $\Psi(\vec{A})$. Then, there exist causal laws

$$\begin{array}{l} B \textbf{ causes } F \textbf{ if } P_1,\ldots,P_k \\ B' \textbf{ causes } \neg F \textbf{ if } Q_1,\ldots,Q_l \end{array} \tag{43}$$

in D_l, where B and B' are sub actions of A_0 and such that P_i $(i=1,\ldots,k)$ and Q_i $(i=1,\ldots,l)$ are true in $\Psi(\vec{A})$. From this and construction of Ψ it follows that $\mathcal{M}_{\mathrm{SC}} \models H(P_i,\vec{A})$ $(i=1,\ldots,k)$ and $\mathcal{M}_{\mathrm{SC}} \models H(Q_i,\vec{A})$ $(i=1,\ldots,l)$, for any pair of causal laws of the form (43). This implies that $Causes^{+}(B, F, \vec{A})$ and $Causes^{-}(B', F, \vec{A})$ hold in $\mathcal{M}_{\mathrm{SC}}$. Predicate In is defined explicitly, so we also have $In(B, A_0)$ and $In(B', A_0)$. Therefore, by axiom

$$In(b, a) \ \wedge Causes^{+(-)}(b, f, \alpha) \supset Inherits^{+(-)}(a, f, \alpha)$$

we have that $Inherits^{+}(A_0, F, \vec{A})$ and $Inherits^{-}(A_0, F, \vec{A})$ hold in $\mathcal{M}_{\mathrm{SC}}$, and by axiom

$$Inherits^{+}(a, f, \alpha) \ \wedge Inherits^{-}(a, f, \alpha) \supset Undefined(a \circ \alpha)$$

we have that $Undefined(A_0 \circ \vec{A})$ holds in $\mathcal{M}_{\mathrm{SC}}$. Therefore, by construction of Ψ, $\Psi(\vec{A} \circ A_0)$ is undefined.

Now, let us show that for all a, α, if $E_a^{+}(\alpha) \cap E_a^{-}(\alpha) = \emptyset$, then $\Psi(\alpha \circ a) = \Psi(\alpha) \cup E_a^{+}(\alpha) \setminus E_a^{-}(\alpha)$. This proof is very similar to proof of Lemma 1. We will proof the base case only.

Base Case: Length of α is 0. We will show

$$\Psi([a]) = \Psi([\,]) \cup E_a^{+}(\Psi([\,])) \setminus E_a^{-}(\Psi([\,]))$$

a) $\Psi([a]) \supseteq \Psi([\,]) \cup E_a^{+}(\Psi([\,])) \setminus E_a^{-}(\Psi([\,]))$.

Assume that $F \in \Psi([\,]) \cup E_A^{+}(\Psi([\,])) \setminus E_A^{-}(\Psi([\,]))$ for some A and F. Let us show $F \in \Psi([A])$. Clearly, $F \notin E_A^{-}(\Psi([\,]))$. In the case when A is atomic, we showed in the proof of Lemma 1 this implies that $\mathcal{M}_{\mathrm{SC}} \models \neg Causes^{-}(A, F, \epsilon)$. From this and

$$In(b, a) \ \wedge Causes^{+(-)}(b, f, \alpha) \supset Inherits^{+(-)}(a, f, \alpha) \tag{44}$$

it follows that $\mathcal{M}_{\mathrm{SC}} \models \neg Inherits^{-}(A, F, \epsilon)$.[15] Now if A is not atomic then there must not exist an atomic action B in A such that $\neg F$ is an immediate effect of B. Again, this implies that $\mathcal{M}_{\mathrm{SC}} \models \neg Causes^{-}(B, F, \epsilon)$. And since this holds for all b such that $In(b, A)$ holds in $\mathcal{M}_{\mathrm{SC}}$, then $\mathcal{M}_{\mathrm{SC}} \models \neg Inherits^{-}(A, F, \epsilon)$.

Now, there are two cases of interest:

i) $F \in E_A^{+}(\Psi([\,]))$

[15] Recall that $In(a, a)$ is included for all basic actions.

ii) $F \in \Psi([\,])$ and $F \notin E^+_A(\Psi([\,]))$

Consider case (i). For the case when A is atomic, we showed in the proof of Lemma 1 that we get $\mathcal{M}_{\mathrm{SC}} \models Causes^+(A, F, \epsilon)$, and that by (44) we get $\mathcal{M}_{\mathrm{SC}} \models Inherits^+(A, F, \epsilon)$. If A is not atomic then there must be an action B in A such that

$$B \ \textbf{causes} \ \ F \ \textbf{if} \ \ P_1, \ldots, P_n \in D_l$$

such that the preconditions hold in $\Psi([\,])$. As shown in the first part of this proof, this implies that $\mathcal{M}_{\mathrm{SC}} \models Causes^+(B, F, \epsilon)$ and by (44) that $\mathcal{M}_{\mathrm{SC}} \models Inherits^+(A, F, \epsilon)$. Since we are considering the case when $\neg Undefined(A \circ \alpha)$ holds in $\mathcal{M}_{\mathrm{SC}}$ for all α, by axiom

$$\neg Undefined(a \circ \alpha) \ \wedge Inherits^+(a, f, \alpha) \supset Holds(f, a \circ \alpha)$$

we get that $\mathcal{M}_{\mathrm{SC}} \models Holds(F, A \circ \epsilon)$, which by construction of Ψ implies that $F \in \Psi([A])$.

Now consider case (ii). By construction of Ψ we have that $\mathcal{M}_{\mathrm{SC}} \models Holds(F, \epsilon)$. We showed above that from $F \notin E^-_A(\Psi([\,]))$ it follows that $\mathcal{M}_{\mathrm{SC}} \models \neg Inherits^-(A, F, \epsilon)$. It is possible to show the same for positive effects, i.e., that from $F \notin E^+_A(\Psi([\,]))$ it follows that $\mathcal{M}_{\mathrm{SC}} \models \neg Inherits^+(A, F, \epsilon)$.

Thus, we have that $Holds(F, \epsilon)$, $\neg Inherits^+(A, F, \epsilon)$, and $\neg Inherits^-(A, F, \epsilon)$ hold in $\mathcal{M}_{\mathrm{SC}}$. Then by the axiom

$$\neg Inherits^+(a, f, \alpha) \ \wedge \neg Inherits^-(a, f, \alpha) \supset [Holds(f, \alpha) \equiv Holds(f, a \circ \alpha)]$$

it follows that $\mathcal{M}_{\mathrm{SC}} \models Holds(F, A \circ \epsilon)$, and by construction of Ψ, that $F \in \Psi([a])$.

b) $\Psi([a]) \subseteq \Psi([\,]) \cup E^+_a(\Psi([\,])) \setminus E^-_a(\Psi([\,]))$.

Let $F \in \Psi([A])$ for an arbitrary A and F. Let us show that $F \in \Psi([\,]) \cup E^+_A(\Psi([\,])) \setminus E^-_A(\Psi([\,]))$. By construction of Ψ, $\mathcal{M}_{\mathrm{SC}} \models Holds(F, A \circ \epsilon)$. Since we are considering the case when $\mathcal{M}_{\mathrm{SC}} \models \neg Undefined(A \circ \epsilon)$, $\mathcal{M}_{\mathrm{SC}} \models \neg Inherits^-(A, F, \epsilon)$ would imply, by the axiom

$$\neg Undefined(a \circ \alpha) \ \wedge Inherits^-(a, f, \alpha) \supset \neg Holds(f, a \circ \alpha)$$

that $\mathcal{M}_{\text{SC}} \models \neg Holds(F, A \circ \epsilon)$. Therefore, $\mathcal{M}_{\text{SC}} \models \neg Inherits^{-}(A, F, \alpha)$. Then, by Lemma 16, we have that for every action b such that $\mathcal{M}_{\text{SC}} \models In(b, a)$, $\mathcal{M}_{\text{SC}} \models \neg Causes^{-}(b, f, \epsilon)$. Consequently, for every b in A such that b **causes** $\neg F$ **if** $q_1, \ldots, q_m \in D_l$, the preconditions do not hold in $\Psi([\,])$. From this it follows that $F \notin E_A^{-}(\Psi([\,]))$. Now, there are two cases of interest:

i) $\mathcal{M}_{\text{SC}} \models Inherits^{+}(A, F, \epsilon)$
ii) $\mathcal{M}_{\text{SC}} \models \neg Inherits^{+}(A, F, \epsilon) \ \wedge Holds(F, \epsilon)$

Consider case (i). By Lemma 16 we have that there exists an action B such that $\mathcal{M}_{\text{SC}} \models In(B, A) \ \wedge Causes^{+}(B, F, \epsilon)$ and by Lemma 11 that there is a causal law B **causes** F **if** $P_1, \ldots, P_n \in D_l$ such that $\mathcal{M}_{\text{SC}} \models H(P_i, \epsilon)$ for $i = 1, \ldots n$. By construction of Ψ, all P_i must hold in $\Psi([\,])$. Therefore, $F \in E_A^{+}(\Psi([\,]))$. From this and $F \notin E_A^{-}(\Psi([\,]))$, shown above, we conclude that $F \in \Psi([\,]) \cup E_A^{+}(\Psi([\,])) \setminus E_A^{-}(\Psi([\,]))$.

Now consider case (ii). By construction of Ψ, $F \in \Psi([\,])$. Since $F \notin E_A^{-}(\Psi([\,]))$, shown above, $F \in \Psi([\,]) \cup E_A^{+}(\Psi([\,])) \setminus E_A^{-}(\Psi([\,]))$. $\square$

Lemma 4

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

$$(\Psi, \Sigma) \models \phi \ \Leftrightarrow \ \mathcal{M} \models \tau(\phi) \tag{45}$$

and if $\mathcal{M}$ is a model of $T_c(D)$ then there exists a model (Ψ, Σ) of D such that (45) holds.

Proof:

$(\Rightarrow)$

Let $M = (\Psi, \Sigma)$ be a model of D. Let $\mathcal{M}_M$ be an interpretation of $T_c(D)$ such that:

i) $\alpha \in \mathcal{M}_M[\![Undefined]\!]$ iff $\Psi(\alpha)$ is undefined,
ii) $\langle a, f, \alpha \rangle \in \mathcal{M}_M[\![Causes^{+(-)}]\!]$ iff f is an immediate effect of a in $\Psi(\alpha)$,
iii) $\langle a, f, \alpha \rangle \in \mathcal{M}_M[\![Inherits^{+(-)}]\!]$ iff $f \in E_a^{+(-)}(\Psi(\alpha))$
iv) $\langle f, \alpha \rangle \in \mathcal{M}_M[\![Holds]\!]$ iff $f \in \Psi(\alpha)$

$\mathcal{M}_M[\![At]\!] = \{\langle f, s_1\rangle : f \in \Psi(\Sigma(s_1))\}$,

v) $\mathcal{M}_M[\![Occurs]\!] = \{\langle \alpha_1, s_1\rangle : \exists\beta.\beta \subseteq \alpha \ \wedge \beta \preceq \Sigma(S_N) \ \wedge \alpha = \Sigma(s)1 \cdot \alpha_1\}$,

$\mathcal{M}_M[\![Precedes]\!] = \{\langle s_1, s_2\rangle : \Sigma(s_1) \prec \Sigma(s_2)\}$

vi) $\mathcal{M}_M[\![Sit\_map]\!](s) = \alpha$ iff $\Sigma(s) = \alpha$.

vii) framework axioms, $FACTS$, B_{embedded}, $B_{\text{embsubseq}}$, $B_{\text{prefix\_eq}}$, $B_{\text{subsequence}}$, $B_{\text{concatenate}}$, and axiom (d) are satisfied.

We will show $\mathcal{M}_M$ is a model of $T_c(D)$.

By definition of models of D, Ψ is defined on $\Sigma(S_N)$, thus by conditions (i) and (vi) on $\mathcal{M}_M$, axiom $\neg Undefined(Sit\_map(S_N))$ is satisfied.

By definition of situation assignment, $\Sigma(S_0) = [\,]$ and for all s, $\Sigma(s_i)$ is a prefix of $\Sigma(S_N)$. Thus, by condition (vi) on $\mathcal{M}_M$, axioms

$$Sit\_map(S_0) = \epsilon$$

$$Prefix\_eq(Sit\_map(s), Sit\_map(S_N))$$

are satisfied.

Note that conditions (i), (ii), (iii), and (iv) on predicates $Undefined$, $Causes^{+(-)}$, $Inherits^{+(-)}$ and $Holds$, are the same as those used to show that such an interpretation of these predicates is a model of $SC_c(D_l)$ (see the proof of Lemma 3), therefore the set of axioms $SC_c(D_l)$ is satisfied.

Axiom (e) and (g) are the same as in T and can be shown to be satisfied as in the proof of Lemma 2.

Let us consider axiom (f'):

$$(f')\ Occurs(\alpha, s) \equiv \begin{bmatrix} (\exists\beta,\gamma)Embedded(\alpha,\beta) \ \wedge Concatenate(Sit\_map(s),\beta,\gamma) \\ \wedge Prefix\_eq(\gamma, Sit\_map(S_N))) \end{bmatrix}$$

For any α, β, if α is embedded in β then, by correctness of B_{embedded} (Lemma 13), $Embedded(\alpha,\beta)$ holds in $\mathcal{M}_M$. Also, by correctness of $B_{\text{prefix\_eq}}$ (Lemma 6), $\gamma \preceq \Sigma(S_N)$ implies that $Prefix\_eq(\gamma, \Sigma(S_N))$ holds in $\mathcal{M}_M$. Finally, if $\gamma = \Sigma(s) \cdot \beta$ then, by correctness of $B_{\text{concatenate}}$ (Lemma 10),

$Concatenate(\Sigma(s), \beta, \gamma)$ holds in $\mathcal{M}_M$. Therefore axiom (f') is satisfied by $\mathcal{M}_M$.

Finally, value minimization axiom (c) captures exactly the minimality condition, with respect to ordering *embedded-subsequence*, imposed on sequence $\Sigma(S_N)$ by the definition of models of domain descriptions in $\mathcal{L}_c$.

$(\Leftarrow)$

Let $\mathcal{M}$ be a model of $T_c(D)$. By Lemma 3 there exists a causal model Ψ such that for all f, α,

$$\begin{aligned} & f \in \Psi(\alpha) \text{ iff } \mathcal{M} \models Holds(f, \alpha), \text{ and} \\ & \Psi(\alpha) \text{ is undefined } \Leftrightarrow \mathcal{M}_{\mathrm{SC}} \models Undefined(\alpha) \end{aligned} \tag{46}$$

Let Ψ be such a causal model and Σ be a situation assignment such that for all s,

$$\Sigma(s) = \mathcal{M}[\![Sit\_map]\!](s). \tag{47}$$

We will show $M = (\Psi, \Sigma)$ is a model of D.

First, by (47) and axiom

$$Sit\_map(S_0) = \epsilon$$

we have $\Sigma(S_0) = [\,]$, and by axiom

$$Prefix\_eq(Sit\_map(s), Sit\_map(S_N))$$

we have $\Sigma(s)$ is a prefix of $\Sigma(S_N)$ for any situation symbol s in the language of D. Thus, Σ is a situation assignment. By (46) and axiom

$$\neg Undefined(Sit\_map(S_N))$$

of $T_c(D)$ we have that $\Sigma(S_N)$ belongs to the domain of Ψ. Therefore, Σ and Ψ form an interpretation of D.

It only remains to be shown that facts in D are true in M and that there is no $N' = (\Psi, \Sigma')$ such that $\Sigma'(S_N)$ is an embedded-subsequence of $\Sigma(S_N)$ and

N' is a model of D.

Axioms (e) and (g) are the same in T and T_c and it can be shown as in the proof of Lemma 2 that M satisfies facts of the forms F **at** S and S_1 **precedes** S_2.

We need to show M satisfies facts of the form

α **occurs_at** S

Suppose ($\vec{A}$ **occurs_at** S) $\in D$. Then $\mathcal{M} \models Occurs(\vec{A}, S)$ and there exist β, γ such that $Embedded(\vec{A}, \beta)$, $Concatenate(Sit\_map(S), \beta, \gamma)$, and $Prefix\_eq(\gamma, Sit\_map(S_N))$ hold in $\mathcal{M}$. Let $oB, \vec{G}$ be two sequences such that this holds. Then, by correctness of B_{embedded} (Lemma 13), we have that $\vec{A}$ is embedded in $\vec{B}$, by correctness of $B_{\text{concatenate}}$ (Lemma 10), $\vec{G} = Sit\_map(S) \cdot \vec{B}$, and by correctness of $B_{\text{prefix\_eq}}$ (Lemma 6), $\vec{G}$ is a prefix of $Sit\_map(S_N)$. By (47), $\vec{G} = \Sigma(S) \circ \vec{B}$ and $\vec{G}$ is a prefix of $\Sigma(S_N)$. Therefore M satisfies ($\vec{A}$ **occurs_at** S).

Finally, we need to show there is no interpretation $N' = (\Psi, \Sigma')$ of D such that $\Sigma'(S_N)$ is an embedded-subsequence (not equal) of $\Sigma(S_N)$ and N' is a model of D. Suppose there is such a model N'. Then, by part ($\Rightarrow$) of this lemma, there is a model $\mathcal{M}'$ of $T_c(D)$ which differs from $\mathcal{M}$ only on the interpretation of function Sit_map and such that sequence $\mathcal{M}'[\![Sit\_map]\!](S_N)$ is an embedded-subsequence (not equal) of $\mathcal{M}[\![Sit\_map]\!](S_N)$. This contradicts our assumption that $\mathcal{M}$ is a model of $T_c(D)$, since $T_c(D)$ includes value minimization of $Sit\_map(S_N)$ with respect to ordering *embedded-subsequence* (See Section 5). □

References

[All84] Allen, J.F. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23, pp. 123–154.

[Bar95] Baral C. 1995. Reasoning about actions: Non-deterministic effects, Constraints and Qualification. In *Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI95)*, Montreal, Canada, pp. 2017–2023.

[BarGel93] Baral C. and Gelfond M. 1993. Representing concurrent actions in extended logic programming. In *Proc. of 13th International Joint Conference on Artificial Intelligence (IJCAI93)*, Chambery, France, 866–871. Extended version appears in *Journal of Logic Programming*, 31 (1-3), pp. 85–117.

[BarGel94] Baral C. and Gelfond M. 1994. Logic programming and knowledge representation. *Journal of Logic Programming*, 19,20:73–148.

[BarGabPro96a] Baral C., Gabaldon A. and Provetti A. 1996. Formalizing Narratives using Nested Circumscription. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI96)*, pp. 652–657, Portland, Oregon.

[BarGabPro96b] Baral C., Gabaldon A. and Provetti A. 1996. Value Minimization in Circumscription. In L.C. Aiello, J. Doyle and S. Shapiro, *Proceedings of Knowledge Representation and Reasoning (KR96)*, pp. 474–481, Cambridge, Massachusetts.

[BarGelPro95b] 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.*

[BarGelPro97] Baral C., Gelfond M. and Provetti A. 1997. Representing Actions : Laws, Observations and Hypothesis. In *Journal of Logic Programming*, 31(1-3), pp. 201-243.

[BarLob96] Baral, C. and Lobo, J. 1996. Formal Characterization of Active Databases. In *Proc. of International Workshop on Logic in Databases.*

[Doherty et al. 95] Doherty, P. and Lukaszewicz, W. and Szalas, A. 1995. Computing circumscription revisited: a reduction algorithm. To appear in *Journal of Automated Reasoning.*

[GL92a] Gelfond M. and Lifschitz V. 1992. Representing actions in extended logic programs. In *Joint International Conference and Symposium on Logic Programming*, 559–573.

[GelLifRab91] Gelfond M., Lifschitz V., and Rabinov A. 1991. What are the limitations of the situation calculus? In Boyer R., editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe.* Kluwer Academic, Dordrecht.

[Geo94] Georgeff, M. 1994. Editor, Special issue of Journal of Logic and Computation on *Action and Processes*, 4(5), Oxford University Press.

[GiuKarLif95] 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.

[GiuKarLif97] Giunchiglia E., Kartha N. and Lifschitz V. 1997. Representing Actions: Indeterminacy and Ramifications. Manuscript.

[Hoa87] Hoare C. 1987. An overview of some formal methods for program design. *IEEE Computer*, 85-91, Sept 87.

[KakKowTon92] Kakas A., Kowalski, R. and Toni F. 1992. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770.

[KakMil96] Kakas A. and Miller, R. 1996. A simple declarative language for describing narratives with actions. *Journal of Logic Programming*, pp. .

[Kar93] Kartha, G.N. 1993 Soundness and completeness theorems for three formalizations of action. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI93*, pp 712–718.

[Kar94] Kartha, G.N. 1995 A mathematical investigation of reasoning about actions. Ph.D dissertation. University of Texas at Austin.

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

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

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

[Lif94] Lifschitz V. 1994. Circumscription. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *The Handbook of Logic in AI and Logic Programming,* volume 3, pages 298-352. Oxford University Press.

[Lif93] V. Lifschitz. Restricted Monotonicity. In *Proc. of AAAI-93*, pp. 432–437, 1993.

[Lif97] V. Lifschitz. Editor, Special issue of Journal of Logic Programming on *Reasoning about Actions and Change*, 31(1-3).

[LinSho92] Lin F., and Shoham Y., 1992. Concurrent actions in the situation calculus. In *Proc. AAAI-92.* pages 590-595.

[MS89] W. Marek and V.S. Subrahmanian. The Relationship Between Stable, Supported, Default and Auto-Epistemic Semantics for General Logic Programs. In G. Levi and M. Martelli, editors, *Proceedings of Sixth International Conference in Logic Programming*, pages 600–620, 1989.

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

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

[McC92] McCarthy J. 1992. Overcoming an unexpected obstacle. Manuscript.

[McC95] McCarthy J. 1995. Situation calculus with concurrent events and narrative. Preliminary manuscript. (Available only at http://www-formal.stanford.edu/jmc/narrative.dvi)

[MH69] McCarthy J. and Hayes P. 1986. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer B. and Michie D., editors, *Machine Intelligence*, 4:463–502. Edinburgh University Press.

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

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

[Pin94] Pinto, J. 1994. *Temporal Reasoning in the Situation Calculus*, PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.

[Pin96] Pinto, J. 1996. Occurrences and narratives as constraints in the branching structure of the situation calculus. Manuscript (http://lyrcc.ing.puc.cl/jpinto/).

[PinRei93] 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.

[PinRei95] Pinto J. and Reiter R. 1995. Reasoning about time in the situation calculus. In *Annals of Math and AI*, 14, 2-4, pp. 251-268.

[Rei91] Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In *Artificial Intelligence and Mathematical Theory of Computation*, ed. V. Lifschitz, 359-380.

[Rei96] Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In *KR 96*, 2-13.

[LevReiLesLinSch97] Levesque, H., Reiter, R., Lesperance, Y., Lin, F. and Scherl, R. 1997. Golog: A Logic programming language for dynamic domains. In *Journal of Logic Programming*, 31, 1-3.

[San89] Sandewall E. 1989. Filter preferential entailment for the logic of action in almost continuous worlds. In *Proc. of 11th International Joint Conference on Artificial Intelligence*, 894–899.

[San92] Sandewall E. 1992. Features and fluents: A systematic approach to the representation of knowledge about dynamical systems. Technical report, Institutionen for datavetenskap, Universitetet och Tekniska hogskolan i Linkoping, Sweden.

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