Action Languages, Temporal Action Logics and the Situation Calculus

Enrico Giunchiglia
DIST - Università di Genova
Viale Causa 13, 16145 Genova, Italy
Email: enrico@dist.unige.it

Vladimir Lifschitz
Department of Computer Sciences
University of Texas, Austin, TX 78712, USA
Email: vl@cs.utexas.edu

Abstract

In the first part of the paper, we investigate the relationship between the action description language $\mathcal{C}$ and the situation calculus. Action description languages are based on the model of time that includes only two time instants: 0 for the beginning of the execution of an action and 1 for the end. The ontology of time in the situation calculus is much more elaborate. We show how to bridge the gap between these two ontologies by defining a translation from $\mathcal{C}$ into the situation calculus. The second part of the paper relates $\mathcal{C}$ to the formalism of temporal action logics (TAL) by showing how a fragment of $\mathcal{C}$ can be embedded into TAL.

1 Introduction

On June 30, 1998 the *Electronic Newsletter on Reasoning about Actions and Change*, edited by Erik Sandewall, invited its subscribers to write reference articles summarizing the existing "standard approaches" to formalizing actions. These articles were expected to "serve as a natural basis for next-level articles that *compare* alternative approaches wrt expressiveness, range of applicability, etc." Further issues of the *Newsletter* announced the submission of several articles written in response to this appeal: on the fluent calculus, on temporal action logics, on action languages, on cognitive robotics logic and on the situation calculus.

This paper contributes to the second part of Sandewall's program by investigating the relationship between the most recent arrival in the family of action description languages—language $\mathcal{C}$ from [Giunchiglia and Lifschitz, 1998]—on the one hand, and the situation calculus [McCarthy and Hayes, 1969] and temporal action logics, as described in [Gustaffson and Doherty, 1996], on the other. This paper is thus similar to [Kartha, 1993; Thielscher, 1994] since it establishes a precise relationship between existing formalisms for reasoning about action and change.

Language $\mathcal{C}$, as every action description language, serves for describing transition diagrams—directed graphs whose vertices represent states of the world and whose edges represent the execution of actions. The execution of several actions in a row corresponds to a path in the transition diagram. To talk about such paths, one needs an "action query language"; action description languages, such as $\mathcal{C}$, do not provide syntactic means for doing that. Action description languages are based, essentially, on the model of time that includes only two time instants: 0 for the beginning of a transition and 1 for its end.

The situation calculus, on the contrary, employs an elaborate ontology of time. Its result function, denoted in this paper by Do, allows us to distinguish between "alternative futures"

$$Do(A_1, s) \text{ and } Do(A_2, s)$$

of a situation s. By iterating this function, we can form expressions like

$$Do(A_2, Do(A_1, s))$$

that denote the result of the execution of several actions consecutively.

In this paper we show how to bridge the gap between these two ontologies of time by defining a translation sit from $\mathcal{C}$ into the situation calculus. For any action description D in language $\mathcal{C}$, the corresponding theory $sit(D)$ is, intuitively, a situation calculus theory of the actions that are described by D. Theorem 1 below makes this claim precise.

The formalism of temporal action logics (TAL), inspired by the ideas of [Sandewall, 1995], describes narratives using high-level "surface languages" that are somewhat similar to action languages. The meaning of a domain description in a surface language is characterized by a circumscription that translates it into classical logic. As far as the ontology of time is concerned, TAL is rather flexible. The set of time instants is usually assumed to be infinite and to have a linear structure, like the set of natural numbers. Another possibility, however, is to represent time by the set $\{0, 1\}$, as in action description languages.

In this paper we relate the work on TAL to action language $\mathcal{C}$ by showing how a fragment of $\mathcal{C}$ can be embedded into TAL. Theorem 2 explains in what sense this embedding preserves the meaning of the original action description.

2 Review of $\mathcal{C}$

This review closely follows [Giunchiglia and Lifschitz, 1998].

2.1 Syntax and Semantics

Consider a propositional signature (set of atoms) partitioned into the set of *fluent symbols* and the set of *action symbols*. A *formula* is a propositional combination of atoms. An *action* is an interpretation of the action symbols. Intuitively, to execute an action α means to execute concurrently the "elementary actions" represented by the action symbols satisfying α.

An *action description* is a set of

- *static laws*, of the form:
$$\textbf{caused } F \textbf{ if } G, \tag{1}$$
and
- *dynamic laws*, of the form:
$$\textbf{caused } F \textbf{ if } G \textbf{ after } H, \tag{2}$$

where F, G, H are formulas such that F and G do not contain action symbols. Both in (1) and in (2), formula F is called the *head* of the law. An action description is *finite* if its signature is finite and it consists of finitely many laws.

Consider an action description D. A *state* is an interpretation of the fluent symbols that satisfies $G \supset F$ for every static law (1) in D. A *transition* is any triple $\langle \sigma, \alpha, \sigma' \rangle$ where σ, σ' are states and α is an action; σ is the *initial* state of the transition, and σ' is its *resulting* state. A formula F is *caused* in a transition $\langle \sigma, \alpha, \sigma' \rangle$ if it is

- the head of a static law (1) from D such that σ' satisfies G, or
- the head of a dynamic law (2) from D such that σ' satisfies G and $\sigma \cup \alpha$ satisfies H.

A transition $\langle \sigma, \alpha, \sigma' \rangle$ is *causally explained* in D if its resulting state σ' is the only interpretation of the fluent symbols that satisfies all formulas caused in this transition.

The *transition diagram* represented by an action description D is the directed graph which has the states of D as vertices, and which includes an edge from σ to σ' labeled α for every transition $\langle \sigma, \alpha, \sigma' \rangle$ that is causally explained in D.

The following abbreviations are useful. A dynamic law of the form
$$\textbf{caused } F \textbf{ if } \mathit{True} \textbf{ after } A \wedge H$$
where A is an action symbol and H does not contain action symbols, can be written as
$$A \textbf{ causes } F \textbf{ if } H. \tag{3}$$
(This law expresses that F is the effect of A when precondition H is satisfied.) For
$$\textbf{caused } F \textbf{ if } \mathit{True} \textbf{ after } A$$
we write
$$A \textbf{ causes } F. \tag{4}$$
A dynamic law of the form
$$\textbf{caused } F \textbf{ if } F \textbf{ after } F$$
can be written as
$$\textbf{inertial } F. \tag{5}$$
(This law says that if F was true and remains true then there is a cause for this; such laws are used in $\mathcal{C}$ to solve the frame problem.) Finally, we combine a group of propositions
$$\textbf{inertial } F_1, \ \textbf{inertial } F_2, \ldots$$
into
$$\textbf{inertial } F_1, F_2 \ldots.$$

2.2 Example

If an object is submerged in water then there is a cause for it to be wet. The action of putting a puppy in water causes it to be in water; consequently, this action has an indirect effect: it will make the puppy wet. The action of drying a puppy with a towel causes it to be dry; consequently, this action has an implicit precondition: it is impossible to dry a puppy with a towel when it is in water.

This domain can be described in $\mathcal{C}$ using fluent symbols *InWater* and *Wet* and action symbols *PutInWater* and *DryWithTowel*, as follows:
$$\begin{array}{l} \mathit{PutInWater} \textbf{ causes } \mathit{InWater}, \\ \mathit{DryWithTowel} \textbf{ causes } \neg \mathit{Wet}, \\ \textbf{caused } \mathit{Wet} \textbf{ if } \mathit{InWater}, \\ \textbf{inertial } \mathit{InWater}, \neg \mathit{InWater}, \mathit{Wet}, \neg \mathit{Wet}. \end{array} \tag{6}$$

The states of this action description are the 3 interpretations of the fluent signature $\{\mathit{InWater}, \mathit{Wet}\}$ that satisfy
$$\mathit{InWater} \supset \mathit{Wet}.$$

The causally explained transitions for (6) can be found using the method from [Giunchiglia and Lifschitz, 1998]. The method consists in transforming a given finite description D into a propositional formula $ct_1(D)$ whose models correspond to the transitions causally explained in D. In this propositional theory, every fluent symbol is given one of two subscripts 0, 1, to represent the value of the fluent at the corresponding time instant.

In the case of (6), $ct_1(D)$ can be rewritten as the conjunction of the formulas
$$\begin{array}{l} \mathit{InWater}_1 \equiv \mathit{InWater}_0 \vee \mathit{PutInWater}, \\ \mathit{Wet}_1 \equiv (\mathit{Wet}_0 \wedge \neg \mathit{DryWithTowel}) \vee \mathit{PutInWater}, \\ \mathit{InWater}_0 \supset \mathit{Wet}_0, \\ \mathit{DryWithTowel} \supset (\neg \mathit{InWater}_0 \wedge \neg \mathit{PutInWater}). \end{array} \tag{7}$$

The first two of these formulas characterize the values of the fluents at time 1. They show that the domain under consideration is deterministic, that is to say, the effect of every action in this domain is completely determined by the state in which it is executed. The second disjunctive term in the second line represents the indirect effect of *PutInWater* on *Wet*. The third line says that the initial state of a transition is indeed a state. The last line shows that the action *DryWithTowel* can be executed only when *InWater* is false—this is the implicit precondition mentioned above—and also that this action cannot be executed concurrently with *PutInWater*.

3 From $\mathcal{C}$ to the Situation Calculus

3.1 Causal Theories

The situation calculus comes in many flavors, because its basic syntactic feature—the result function—can be used either in the framework of classical logic or in combination with a nonmonotonic formalism. Lifschitz [1998] showed how to combine the situation calculus with the logic of causal explanation defined in [McCain and Turner, 1997] and generalized in [Lifschitz, 1997]. This "causal situation calculus" is particularly easy to relate to $\mathcal{C}$.

Recall that, according to [Lifschitz, 1997], a *causal rule* is an expression of the form

$$F \leftarrow G \tag{8}$$

where F and G are formulas of a fixed language of classical logic. Intuitively, (8) expresses that G provides a "causal explanation" for F. A formula F is identified with the rule

$$False \leftarrow \neg F. \tag{9}$$

A *causal theory* is defined by

- a finite set of nonlogical constants of the underlying language, called the *explainable symbols* of the theory, and
- a finite set of causal rules.

The semantics of causal theories is defined by a syntactic transformation that is somewhat similar to circumscription; a causal theory is viewed as shorthand for the sentence obtained as the result of this transformation [Lifschitz, 1997].

3.2 Translation

Let D be a finite action description in language $\mathcal{C}$. The language of the corresponding causal theory $sit(D)$ has variables of two sorts: for actions $a, a', \ldots$ and for situations $s, s', \ldots$. In this language, every fluent symbol of D becomes a unary predicate that requires a situation term as an argument, and every action symbol of D becomes an object constant. For instance, in the situation calculus counterpart of (6),

$$Wet(Do(PutInWater, s))$$

is an atomic formula.

The fluent symbols of D are the explainable symbols of $sit(D)$.

For any formula H of the language of D, $H(a, s)$ will stand for the situation calculus formula obtained from H by replacing

- each fluent symbol P with $P(s)$, and
- each action symbol A with $a = A$.

For instance, if H is

$$InWater \vee PutInWater$$

then $H(a, s)$ stands for

$$InWater(s) \vee a = PutInWater.$$

If H does not contain action symbols, we will write $H(s)$ instead of $H(a, s)$.

The set of causal rules of $sit(D)$ consists of three parts. Its first part is obtained by translating each of the laws of D into the situation calculus, one causal rule per law. The second part is a single formula —that is, a causal rule of form (9)— that characterizes the executability of actions (represented in the situation calculus by the atomic formula $Poss(a, s)$, "it is possible to execute action a in situation s"). The third part consists of several "domain independent" rules, determined by the underlying signature.

The first group includes the rule

$$F(s) \leftarrow G(s)$$

for every static law (1) in D, and the rule

$$F(s') \leftarrow G(s') \wedge H(a, s) \wedge s' = Do(a, s) \wedge Poss(a, s)$$

for every dynamic law (2) in D.

For instance, the causal rules corresponding to laws (6) are

$$\begin{array}{l}
InWater(s') \leftarrow a = PutInWater \wedge s' = Do(a, s) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge Poss(a, s), \\
\neg Wet(s') \leftarrow a = DryWithTowel \wedge s' = Do(a, s) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge Poss(a, s), \\
Wet(s) \leftarrow InWater(s), \\
InWater(s') \leftarrow InWater(s') \wedge InWater(s) \\
\qquad\qquad\qquad \wedge s' = Do(a, s) \wedge Poss(a, s), \\
\neg InWater(s') \leftarrow \neg InWater(s') \wedge \neg InWater(s) \\
\qquad\qquad\qquad \wedge s' = Do(a, s) \wedge Poss(a, s), \\
Wet(s') \leftarrow Wet(s') \wedge Wet(s) \\
\qquad\qquad\qquad \wedge s' = Do(a, s) \wedge Poss(a, s), \\
\neg Wet(s') \leftarrow \neg Wet(s') \wedge \neg Wet(s) \\
\qquad\qquad\qquad \wedge s' = Do(a, s) \wedge Poss(a, s).
\end{array}$$

The second part of $sit(D)$ is the formula

$$Poss(a, s) \equiv \exists p^1 \ldots p^n P^D(a, s, p^1, \ldots, p^n) \tag{10}$$

where $p^1, \ldots, p^n$ are distinct propositional variables, one per each of the fluent symbols $P^1, \ldots, P^n$ of D, and

$$P^D(a, s, p^1, \ldots, p^n)$$

stands for the formula obtained from $ct_1(D)$ by substituting

- $a = A$ for every action symbol A,
- $P^i(s)$ for every atom of the form P^i_0,
- p^i for every atom of the form P^i_1.

For instance, in the case of (6), $ct_1(D)$ is equivalent to (7). Consequently, (10) can be written as

$$\begin{array}{l} Poss(a,s) \equiv \\ \exists p^1 p^2 [p^1 \equiv (InWater(s) \vee a = PutInWater_0) \\ \quad \wedge p^2 \equiv ((Wet(s) \wedge a \neq DryWithTowel) \vee \\ \qquad\qquad a = PutInWater) \\ \quad \wedge (InWater(s) \supset Wet(s)) \\ \quad \wedge (a = DryWithTowel \\ \qquad\qquad \supset (\neg InWater(s) \wedge a \neq PutInWater))] \end{array}$$

which is logically equivalent to

$$\begin{array}{l} Poss(a,s) \equiv \\ \quad (InWater(s) \supset Wet(s)) \\ \quad \wedge (a = DryWithTowel \supset \\ \qquad\qquad (\neg InWater(s) \wedge a \neq PutInWater)). \end{array}$$

In the presence of the other rules of $sit(D)$ (including the unique names assumption below), this can be further simplified:

$$Poss(a,s) \equiv \neg(a = DryWithTowel \wedge InWater(s)). \quad (11)$$

Finally, the third group of rules of $sit(D)$ consists of the formulas

$$Do(a,s) = Do(a',s') \supset (a = a' \wedge s = s') \quad (12)$$

and

$$A_1 \neq A_2 \quad (13)$$

for every pair of distinct action constants A_1, A_2, and the rules

$$\begin{array}{l} P(s) \leftarrow P(s) \wedge \neg\exists a s'(Poss(a,s') \wedge s = Do(a,s')), \\ \neg P(s) \leftarrow \neg P(s) \wedge \neg\exists a s'(Poss(a,s') \wedge s = Do(a,s')) \end{array} \quad (14)$$

for every fluent symbol P.

Formulas (12) and (13) express the unique name assumption for situations and actions. We will denote the conjunction of these formulas by *UNA*. In each of the rules (14), the second conjunctive term says that s is an "initial" situation—not the result of executing any action. Including (14) in $sit(D)$ implies that in any initial situation a fluent can take any value compatible with the static laws of D.

3.3 Theorem 1

The theorem below relates $sit(D)$ to the transition diagram represented by D. To state the theorem, we need the following conventions.

An action α is identified with the set of the action symbols satisfied by α. The metavariable P ranges over the fluent symbols of D, A ranges over the action symbols of D, and σ ranges over the states of D.

For every σ, $s \sim \sigma$ stands for the formula

$$\bigwedge_{P:\sigma \models P} P(s) \wedge \bigwedge_{P:\sigma \not\models P} \neg P(s).$$

By $State^D(s)$ we denote the disjunction

$$\bigvee_\sigma (s \sim \sigma).$$

By $Res^D(s,a,s')$ we denote the disjunction of the formulas

$$s \sim \sigma \wedge a = A \wedge s' \sim \sigma'$$

over all transitions of the form $\langle \sigma, \{A\}, \sigma' \rangle$ that are causally explained in D, and of the formulas

$$s \sim \sigma \wedge \bigwedge_A (a \neq A) \wedge s' \sim \sigma'$$

over all transitions of the form $\langle \sigma, \emptyset, \sigma' \rangle$ that are causally explained in D.

Finally, by $Exec^D(a,s)$ we denote the disjunction of the formulas

$$s \sim \sigma \wedge a = A$$

over all the pairs $\langle \sigma, \{A\} \rangle$ such that for some state σ', the transition $\langle \sigma, \{A\}, \sigma' \rangle$ is causally explained in D, and of the formulas

$$s \sim \sigma \wedge \bigwedge_A (a \neq A)$$

over all the states σ such that for some state σ', the transition $\langle \sigma, \emptyset, \sigma' \rangle$ is causally explained in D.

Theorem 1. *For any finite action description D in $\mathcal{C}$, $sit(D)$ is equivalent to the universal closure of the conjunction of*

(i) $State^D(s)$,

(ii) $Poss(a,s) \wedge s' = Do(a,s) \supset Res^D(s,a,s')$,

(iii) $Poss(a,s) \equiv Exec^D(a,s)$,

(iv) *UNA*.

3.4 Discussion

Part (i) of the statement of Theorem 1 shows that, in any model of $sit(D)$, every situation corresponds to a state of D. Generally, a model does not satisfy the uniqueness condition

$$s \sim \sigma \wedge s' \sim \sigma \supset s = s',$$

so that several situations may correspond to the same state. A model may not satisfy the existence of situation conditions

$$\exists s (s \sim \sigma)$$

either, so that some states of D, generally, are not represented by situations.

Part (ii) shows that, in any model of $sit(D)$, applying *Do* to a pair satisfying *Poss* follows an edge of the transition diagram represented by D. If the action argument of *Do* corresponds to an action symbol A then applying *Do* follows an edge labeled $\{A\}$; otherwise, it follows an edge labeled $\emptyset$. Thus in the language of $sit(D)$ we can talk about executing any one of the "elementary actions" represented by the action symbols of D and also about doing nothing ("waiting") but not about the concurrent execution of several "elementary actions."

To make it easier to talk about waiting, we can extend $sit(D)$ by adding the object constant *Wait* and the axioms

$$Wait \neq A$$

for all action symbols A of D. In case of (6), and in many other action domains, nothing happens while you wait: $sit(D)$ entails

$$P(Do(Wait, s)) \equiv P(s).$$

Part (iii) shows that, in any model of $sit(D)$, the extent of *Poss* corresponds to the executability of actions in D.

4 From $\mathcal{C}$ to TAL

4.1 Conditions (A)–(C)

In the rest of the paper, we consider action descriptions D satisfying the following conditions:

(A) for every fluent symbol P, D contains the dynamic laws **inertial** P and **inertial** $\neg P$,

(B) every dynamic law in D not included in (A) has the form (3), where the head F is a literal or *False*,

(C) the head F of every static law (1) in D is a literal or *False*.

Note that any dynamic law of form (4) can be equivalently replaced by

$$A \textbf{ causes } F \textbf{ if } True.$$

Modulo this modification, example (6) satisfies conditions (A)–(C).

A finite action description satisfying conditions (A)–(C) can be viewed as a scenario description in the sense of [Gustaffson and Doherty, 1996], with the time points 0 and 1. In the syntax of TAL, (3) would be written as

$$A \leadsto ([0]H \supset [1]F),$$

and (1) would be written as

$$G \gg F.$$

Laws (A) are not shown in this syntax explicitly, as their effect is incorporated in the circumscriptive interpretation described below.

For instance, action description (6), written in the syntax of TAL, might look like this:

$$\begin{array}{l} PutInWater \leadsto [1]InWater, \\ DryWithTowel \leadsto [1]\neg Wet, \\ InWater \gg Wet. \end{array}$$

4.2 Translation

Let D be a finite action description satisfying conditions (A)–(C). In application to D, the circumscriptive semantics of TAL from [Gustaffson and Doherty, 1996] can be described as follows.

Consider the two-sorted language with variables for fluents and actions that has

- the fluent symbols and action symbols of D as object constants,
- $Holds_0$, $Holds_1$ and *Occlude* as the unary predicate constants whose argument is a fluent term, and
- *Occurs* as the unary predicate constant whose argument is an action term.

For any formula F of the language of D that does not contain action symbols, and for any $t \in \{0, 1\}$, $Holds_t(F)$ stands for the formula of this language obtained from F by replacing every atom P with $Holds_t(P)$. For instance, $Holds_1(\neg Wet)$ stands for $\neg Holds_1(Wet)$. For any fluent symbol P, $Occlude(\neg P)$ stands for $Occlude(P)$; $Occlude(False)$ stands for *False*. By *UNA* we denote the conjunction of the unique name assumption formulas for the fluent symbols and for the action symbols (which is different from the use of this symbol in the discussion of the situation calculus above).

$\Gamma(D)$ stands for the universal closure of the conjunction of the formulas

$$\begin{array}{l} Occurs(A) \wedge Holds_0(H) \\ \qquad \supset Holds_1(F) \wedge Occlude(F) \end{array}$$

for every dynamic law of form (3) in D,[1] and the formulas

$$Holds_t(G) \supset Holds_t(F) \qquad (t = 0, 1)$$

and

$$\neg Holds_0(G) \wedge Holds_1(G) \supset Occlude(F)$$

for every static law (1) in D.

The semantics of TAL interprets D as the circumscription of *Occlude* relative to $\Gamma(D)$ conjoined with a "persistence axiom" and the unique name assumption:

$$\begin{array}{l} \text{CIRC}[\Gamma(D); Occlude] \\ \wedge \forall f[\neg(Holds_0(f) \equiv Holds_1(f)) \supset Occlude(f)] \\ \wedge UNA. \end{array} \quad (15)$$

4.3 Theorem 2

The theorem below relates (15) to the transition diagram represented by D. To state the theorem, we need the following notation. As in the statement of Theorem 1, the metavariable P ranges over the fluent symbols of D and A ranges over the action symbols of D.

For every state σ, $Holds_t(\sigma)$ stands for the formula

$$\bigwedge_{P:\sigma \models P} Holds_t(P) \wedge \bigwedge_{P:\sigma \not\models P} \neg Holds_t(P)$$

$(t = 0, 1)$.

For every action α, $Occurs(\alpha)$ stands for the formula

$$\bigwedge_{A:\alpha \models A} Occurs(A) \wedge \bigwedge_{A:\alpha \not\models A} \neg Occurs(A).$$

By *Trans* we denote the disjunction of the formulas

$$Holds_0(\sigma) \wedge Occurs(\alpha) \wedge Holds_1(\sigma')$$

[1] Here we deviate from the translation presented in [Gustaffson and Doherty, 1996], which corresponds to Case 1 of Section 7.5.1 of [Doherty *et al.*, 1998] and is only suitable for reasoning about narratives. Instead, we adopt the translation presented in Case 2, which is suitable for reasoning about action descriptions also.

over all transitions $\langle \sigma, \alpha, \sigma' \rangle$ that are causally explained in D.

By *Nochange* we denote the formula

$$\forall f \left[\bigwedge_P (f \neq P) \supset (Holds_0(f) \equiv Holds_1(f)) \right].$$

Theorem 2. *For any finite action description D in $\mathcal{C}$ satisfying conditions (A)–(C), formula (15) is equivalent to the conjunction of*

$$Trans \wedge Nochange \wedge UNA$$

with an explicit definition of Occlude in terms of the other symbols of the language.

It follows that (15) is a conservative extension of the theory with the axioms *Trans*, *Nochange* and *UNA*.

4.4 Discussion

How restrictive are conditions (A)–(C)?

The section on noninertial fluents in [Giunchiglia and Lifschitz, 1998] mentions several reasons why condition (A) can be violated for an action description. First, we would not assume inertia for a fluent that is explicitly defined in terms of other fluents. Second, there are fluents whose values tend to change in a specific way rather than remain unchanged, such as the position of a swinging pendulum. Third, a fluent may have a "default value" that it takes unless an action causes it to take a different value; for instance, a spring-loaded door is closed unless someone has just opened it. The descriptions of such domains in $\mathcal{C}$ would not satisfy condition (A).

Condition (B) prohibits the dynamic laws (2) in which formula G is different from *True*, except those of form (5). Such laws are used in [Giunchiglia and Lifschitz, 1998] in the definition of the abbreviation

possibly caused

that is used in two examples there. Also, it eliminates dynamic laws describing the effect of the concurrent execution of several "elementary actions," such as

caused F **if** *True* **after** $A_1 \wedge A_2$,

where A_1 and A_2 are action symbols.

The restriction of the heads of rules to literals in conditions (B) and (C) is violated in the static laws that express explicit definitions: the head of such a law is an equivalence.

On the other hand, it should be remembered that the action descriptions satisfying conditions (A)–(C) represent just a small fragment of TAL. Recent extensions, surveyed in [Doherty *et al.*, 1998], define, for instance, the distinction between "persistent" and "durational" fluents; we have not investigated a possible correspondence between that distinction and the expressive possibilities of action languages.

5 Conclusion

In much of recent work on representing actions, similar intuitions about indirect effects and implicit preconditions have led researchers to a number of formalisms that seem to be closely related to each other, although that relationship may be not easy to describe in precise terms. In this paper we stated mathematical results that describe the relationship between three formalisms of this kind: action language $\mathcal{C}$, the causal situation calculus and temporal action logics. Our hope is that such results will help researchers better understand the work done by their colleagues, and maybe even collaborate on future projects.

Acknowledgements

We are grateful to Patrick Doherty, Norman McCain and Hudson Turner for useful discussions related to the subject of this paper. The first author was supported by the Italian Space Agency under the project "Un sistema Intelligente per la supervisione di robot autonomi nello spazio." The second author was partially supported by National Science Foundation under grant IIS-9732744.

References

[Doherty *et al.*, 1998] Patrick Doherty, Joakim Gustafsson,
Lars Karlsson, and Jonas Kvarnström. TAL: Temporal action logics language specification and tutorial. *Linköping Electronic Articles in Computer and Information Science ISSN 1401-9841*, 3(015), 1998. Available at **http://www.ep.liu.se/ea/cis/1998/015/**.

[Giunchiglia and Lifschitz, 1998] Enrico Giunchiglia and Vladimir Lifschitz. An action language based on causal explanation: Preliminary report. In *Proc. AAAI-98*, pages 623–630, 1998.

[Gustaffson and Doherty, 1996] J. Gustaffson and P. Doherty. Embracing occlusion in specifying the indirect effects of actions. In *Principles of Knowledge Representation and Reasoning: Proc. Fifth Int'l Conference*, 1996.

[Kartha, 1993] Neelakantan Kartha. Soundness and completeness theorems for three formalizations of action. In *Proc. IJCAI-93*, pages 724–729, 1993.

[Lifschitz, 1997] Vladimir Lifschitz. On the logic of causal explanation. *Artificial Intelligence*, 96:451–465, 1997.

[Lifschitz, 1998] Vladimir Lifschitz. Situation calculus and causal logic. In Anthony Cohn, Lenhart Schubert, and Stuart Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proc. Sixth Int'l Conf.*, pages 536–546, 1998.

[McCain and Turner, 1997] Norman McCain and Hudson Turner. Causal theories of action and change. In *Proc. AAAI-97*, pages 460–465, 1997.

[McCarthy and Hayes, 1969]
John McCarthy and Patrick Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, Edinburgh, 1969. Reproduced in [McCarthy, 1990].

[McCarthy, 1990] John McCarthy. *Formalizing Common Sense: Papers by John McCarthy.* Ablex, Norwood, NJ, 1990.

[Sandewall, 1995] Erik Sandewall. *Features and Fluents*, volume 1. Oxford University Press, 1995.

[Thielscher, 1994] Michael Thielscher. An analysis of systematic approaches to reasoning about actions and change. In *Proc. AIMSA*, 1994.