Available online at www.sciencedirect.com
SCIENCE DIRECT®
Artificial Intelligence 170 (2006) 643–652
Artificial Intelligence
www.elsevier.com/locate/artint

Existential assertions and quantum levels on the tree of the situation calculus

Francesco Savelli

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, I-00198 Roma, Italy

Received 14 February 2005; received in revised form 7 December 2005; accepted 5 January 2006

Available online 8 February 2006

Abstract

In a seminal paper, Reiter introduced a variant of the situation calculus along with a set of its properties. To the best of our knowledge, one of these properties has remained unproved and ignored despite its relevance to the planning problem and the expressivity of the theories of actions. We state this property in a more general form and provide its proof. Intuitively, whenever a theory of actions entails that there exists a situation satisfying a first order formula (e.g., a goal), at least one such situation must be found within a predetermined distance from the initial situation. This distance is *finite* and the *same* in all the models of the theory, since it depends only on the theory and the formula at hand.

Keywords: Knowledge representation; Reasoning about actions and change; Situation calculus

1. Introduction

In reasoning about actions and change with partial knowledge, computing a plan of actions that achieves a specified goal can be impossible even if the existence of the plan is formally guaranteed. Indeed, if a logical theory of actions does not include a complete axiomatic representation of the system's state, no one plan (or finite disjunction of plans) might exist that simultaneously meets the goal in all the models of the theory, though each model might still contain its own valid plan. When this is the case, no finite plan guaranteed to succeed can be built and executed in the real world, even though its existence is actually entailed by the theory.

This problem has a theoretical solution in the variant of the situation calculus [1] introduced by Reiter [2], which has represented one of the most influential approaches to laying formal foundations for planning, diagnosis, and agent specification in artificial intelligence, besides having contributed new insights into database theory. For nearly fifteen years, a diverse body of research has stemmed from Reiter's seminal paper [2], and the original ideas and notions contained therein are currently at the core of much work in cognitive robotics. All but one of the properties of the formalism were formally proved in a later paper [3] after cleaner and more systematic axiomatization, and have been extensively employed in later research.

E-mail address: Francesco.Savelli@dis.uniroma1.it (F. Savelli).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2006.01.002

To the best of our knowledge, the theorem in [2] that provides the solution to the problem introduced above has never been mentioned again, and its formal justification has not appeared in literature. The theorem formally guarantees that if a theory of actions entails the existence of a plan that achieves a given goal, a finite disjunctive instance can be built.

This result is still theoretically very relevant as the research community begins to investigate more complex systems, and consequently the problem of plan computability is being recasted in more sophisticated terms. One noteworthy example is the problem of *"knowing how"* to execute a program augmented with non-deterministic choices and sensing actions [4,5] in cognitive robotics.

In this paper we provide the missing proof of Reiter's theorem. We state the result in more general terms and show its implications for the expressivity of the formalism. Intuitively, whenever an action theory entails that there exists a situation satisfying a first order formula (e.g., a goal), at least one such situation must be found inside a special region of the tree of situations that characterizes the models of the theory. This region takes the form of a *finite* subset of all the levels of the tree, and it is the *same* in all the models of the theory, since it depends only on the theory and the formula at hand. We dub the situation-tree levels belonging to this subset *"quantum levels"*.

The rest of this paper is organized as follows. In Section 2 we recall the basic formal notions needed to make this paper self-contained, and introduce some preliminary definitions and results that are straightforward consequences of previous work. In Section 3 we state and prove some properties of the situation calculus, including the theorem mentioned above (Theorem 2 of [2]). In Section 4 we discuss the implications of these results for the expressivity of the situation calculus and for current research in reasoning about actions and change.

2. The situation calculus

We assume the reader is familiar with the basic concepts of classical logic [6]. We recall some notations.

In a formula, by τ_1/τ_2 we denote that the occurrences of τ_1 are replaced by τ_2. By $(\forall)$ or $(\exists)$ at the beginning of a formula we intend that all the free variables occurring in the formula are universally or existentially closed. Given a function or constant symbol ω of a language $\mathcal{L}$, $\omega^{\mathbb{M}}$ denotes the interpretation of ω in the structure $\mathbb{M}$ of $\mathcal{L}$. We extend the classical notation of entailment to set of sentences: $\mathbb{M} \models \{\phi_i\}_{i\in\mathbb{N}}$ means that the structure $\mathbb{M}$ entails every sentence $\phi_1, \phi_2, \ldots$ of the set simultaneously. The meaning of $T \models \{\phi_i\}_{i\in\mathbb{N}}$, where T is a theory, is likewise defined.

We use the notations, definitions, and results given in [3], which provides systematic and formally complete foundations for the situation calculus as formulated in [2]. These notions are recalled in Sections 2.1, 2.2, and 2.3. Sections 2.4, 2.5, and 2.6 introduce notions that simply follow from previous results.

2.1. The language $\mathcal{L}_{sitcalc}$

The language $\mathcal{L}_{sitcalc}$ is a three-sorted second order language with equality. The three sorts are *action* for actions, *situation* for situations, and a catch-all sort *object* for everything else depending on the domain of application.

We adopt the following notations, with subscripts and superscripts: α and a for terms and variables of sort *action*, σ and s for terms and variables of sort *situation*; t and x for terms and variables of any sort.

S_0 is the constant denoting the initial situation, and

$$do : action \times situation \to situation$$

is a special function that enables the inductive construction of successive situations. The term $do(\alpha, \sigma)$ interprets the situation resulting from performing the action α in the situation σ. Thus the term

$$do(\alpha_n, do(\ldots, do(\alpha_2, do(\alpha_1, S_0))))$$

—abbreviated as $do([\alpha_1, \ldots, \alpha_n], S_0)$—interprets the situation obtained after the sequence of actions $\alpha_1, \ldots, \alpha_n$ has taken place. (If $n = 0$, $do([\alpha_1, \ldots, \alpha_n], \sigma)$ is σ.)

Hence, situations represent sequences of actions, which should be viewed as histories of the dynamical domain, rather than as its states. This is a fundamental distinguishing feature of Reiter's variant of the situation calculus.

The actions are usually completely represented by a finite set of *action function symbols* with signature $(action \cup object)^n \to action$. As an example, consider the action function symbol $moveOn(x, y)$ denoting the action of moving

the object x on the object y. The term $do(moveOn(book, table), S_0)$ will interpret the situation in which this action has been applied to the two objects interpreted by the constant symbols *book* and *table*.

Two special predicates are $\sqsubset : situation \times situation$, which defines an ordering relation on situations, and $Poss : action \times situation$, which defines which actions can be performed in a given situation.

The dynamical properties of the domain are represented by function and predicate symbols with signatures $(action \cup object)^n \times situation \to (action \cup object)$ and $(action \cup object)^n \times situation$. These functions and predicates are respectively called *functional* and *relational fluents*. Following the example above, the relational fluents $On(x, y, s)$ and $Holding(z, s)$ may mean that in the situation s the object x is on the object y, and that the object z is being held by the agent. Non-fluent functions or predicates can be used to represent properties and facts independent of any specific situation, like $Flat(table)$ or $Larger(table, book)$.

The use of terms of sort *situation* in $\mathcal{L}_{sitcalc}$ is limited. The only constant and function symbols of this sort are S_0 and do. The only predicates allowed to take an argument of sort *situation* are *Poss*, $\sqsubset$, the equality, and the relational fluents. The only functions allowed to take an argument of sort *situation* are do and the functional fluents.

In a structure $\mathbb{M}$ of $\mathcal{L}_{sitcalc}$ the domain is partitioned into the subsets *Obj*, *Act*, and *Sit*, reflecting the three sorts of the language. Terms of sorts *object*, *action*, and *situation* range respectively over these different domains. We use ν for an assignment to free variables of any sort.

2.2. Basic action theories

A formula of $\mathcal{L}_{sitcalc}$ is *uniform* in σ iff it is first order, it does not mention the predicates *Poss* or $\sqsubset$, it does not mention equality on situations, it does not quantify over variables of sort *situation*, and σ is the only term of sort *situation* that is mentioned in the fluents occurring in the formula. The set of these formulas can be syntactically defined by induction [3]. Intuitively, the key property of a formula uniform in σ is that it "concerns only σ".

A *basic action theory* $\mathcal{D}$ is composed of the foundational axioms Σ, the set of action precondition axioms $\mathcal{D}_{ap}$, the set of successor state axioms $\mathcal{D}_{ss}$, the set of unique names axioms for the action function symbols $\mathcal{D}_{una}$, and the initial database $\mathcal{D}_{S_0}$. The structure of these sets and of their axioms are as follows.[1]

- Σ contains four *foundational, domain-independent axioms*.

  $$(\forall)\ do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2 \wedge s_1 = s_2 \tag{1}$$

  is a unique name axiom for situations.

  $$(\forall P)\ \{P(S_0) \wedge (\forall a, s)[P(s) \supset P(do(a, s))]\} \supset (\forall s) P(s) \tag{2}$$

  is a second order induction axiom on situations, whose purpose is very similar to that of the analogous axiom in Peano's arithmetic [6]. Axiom (2) has the important effect of limiting the domain *Sit* to the smallest set that (i) contains the interpretation of S_0, and (ii) is closed under the application of the function do to the elements of *Act* and recursively of *Sit* itself. This inductive definition constructs an infinite tree of situations that is rooted in S_0 and contains all the elements of *Sit* as its nodes. Every node branches on all the elements of *Act* [7].
  The last two axioms define the ordering relation on situations:

  $$(\forall s)\ \neg s \sqsubset S_0 \tag{3}$$

  $$(\forall s, s', a)\ s \sqsubset do(a, s') \equiv s \sqsubset s' \vee s = s' \tag{4}$$

  Intuitively, $s \sqsubset s'$ means that s' can be obtained from s by performing an appropriate sequence of actions.
- $\mathcal{D}_{ap}$ contains one *action precondition axiom* in the form

  $$(\forall x_1, \ldots, x_n, s)\ Poss(A(x_1, \ldots, x_n), s) \equiv \Pi_A(x_1, \ldots, x_n, s) \tag{5}$$

  for each action function symbol A of $\mathcal{L}_{sitcalc}$, where $\Pi_A(x_1, \ldots, x_n, s)$ is a formula uniform in s that does not mention any free variable other than $x_1, \ldots, x_n, s$.

[1] For ease of exposition, in this section we will not treat the functional fluents. The complete introduction requires a straightforward technical extension. The interested reader is referred to [3,7]. The theorems provided in this paper and their proofs are formally complete with respect to the full framework including functional fluents.

An action precondition axiom defines the preconditions to the executability of an action in a given situation, in terms of properties holding in that situation alone. Following the simple example above, we may define:

$$(\forall x, s)\ Poss(pickUp(x), s) \equiv \neg(\exists y) Holding(y, s)$$
$$(\forall x, s)\ Poss(putDown(x), s) \equiv Holding(x, s)$$
$$(\forall x, y, s)\ Poss(moveOn(x, y), s) \equiv Holding(x, s) \wedge Flat(y) \wedge Larger(y, x)$$

- $\mathcal{D}_{ss}$ contains one *successor state axiom* in the form

$$(\forall x_1, \ldots, x_n, a, s)\ F(x_1, \ldots, x_n, do(a, s)) \equiv \Phi_F(x_1, \ldots, x_n, a, s) \tag{6}$$

for each relational fluent F of $\mathcal{L}_{sitcalc}$, where $\Phi_F(x_1, \ldots, x_n, a, s)$ is a formula uniform in s that does not mention any free variable other than $x_1, \ldots, x_n, a, s$.
A successor state axiom defines how a fluent's truth value changes from situation to situation, based only on the action taken and on the properties holding in the current situation. For example:

$$(\forall x, y, a, s)\ On(x, y, do(a, s)) \equiv a = moveOn(x, y) \vee [On(x, y, s) \wedge a \neq pickUp(x)]$$
$$(\forall x, a, s)\ Holding(x, do(a, s)) \equiv a = pickUp(x) \vee [Holding(x, s) \wedge a \neq putDown(x)]$$

- $\mathcal{D}_{una}$ contains *unique name axioms for the action function symbols*. For every two distinct action function symbols A and B of $\mathcal{L}_{sitcalc}$:

$$(\forall)\ A(x_1, \ldots, x_n) \neq B(y_1, \ldots, y_m)$$

Besides, for each action function symbol A:

$$(\forall)\ A(x_1, \ldots, x_n) = A(y_1, \ldots, y_n) \supset x_1 = y_1 \wedge \cdots \wedge x_n = y_n$$

- $\mathcal{D}_{S_0}$, often called the *initial database*, is a set of first order sentences that are uniform in S_0. Their function is to describe the world as it is before any action has been executed. Some of these sentences may mention no situation, to the purpose of representing situation-independent properties. Unique name axioms for individuals are a typical case of such sentences.
An initial database for our simple example might be:

$$\neg(\exists x)\ Holding(x, S_0)$$
$$table \neq book$$
$$Flat(table)$$
$$Larger(table, book)$$

Note that a basic action theory thus defined is first order except for the induction axiom (2).

2.3. Relative satisfiability and regression

Relative satisfiability is a key property of basic action theories that will play an essential role in this paper.

Theorem 1 *(Relative satisfiability). A basic action theory $\mathcal{D}$ is satisfiable iff $\mathcal{D}_{S_0} \cup \mathcal{D}_{una}$ is.*

Regression is a form of automated reasoning widely employed in computational procedures for planning and diagnosis. There is a precise and natural way to characterize regression formally in the situation calculus. We first need to define the class of the regressable formulas of $\mathcal{L}_{sitcalc}$.

Definition 2 *(The regressable formulas).* A formula W of $\mathcal{L}_{sitcalc}$ is *regressable* iff

(1) W is first order.

(2) Every term of sort *situation* mentioned by W has the form $do([\alpha_1, \ldots, \alpha_n], S_0)$ for some[2] $n \geqslant 0$ and terms $\alpha_1, \ldots, \alpha_n$ of sort *action*.
(3) For every term of the form $Poss(\alpha, \sigma)$ mentioned by W, α has the form $A(t_1, \ldots, t_n)$ for some action function symbol A of $\mathcal{L}_{sitcalc}$, $n \geqslant 0$, and terms $t_1, \ldots, t_n$ of appropriate sorts.

Observe that any formula uniform in σ is trivially regressable if σ is not a variable of sort *situation*.

Given a basic action theory $\mathcal{D}$ and a regressable formula W, the *regression operator* $\mathcal{R}$ applied to W generates a formula $\mathcal{R}[W]$ uniform in S_0 that is logically equivalent to W in the models of $\mathcal{D}$. This operator is defined by induction both on the syntactic structure of W and on the number of actions in the terms of sort *situation* occurring in W. This definition as well as a complete and detailed account of all the syntactic transformations involved by $\mathcal{R}$ can be found elsewhere [3,7]. We emphasize here that such transformations are chiefly based on employing the logical definitions (equivalences) embodied by the successor state axioms as rewriting rules. In this way, the number of actions in the terms of sort *situation* can be reduced. Since the rewriting rules are axioms of $\mathcal{D}$, logical equivalence is preserved. After a finite number of iterations, this process ends with the formula $\mathcal{R}[W]$ uniform in S_0.

A formal account of the properties of $\mathcal{R}$ is given by two theorems.

Theorem 3. *Let W be a regressable formula of $\mathcal{L}_{sitcalc}$ and $\mathcal{D}$ a basic action theory. Then $\mathcal{R}[W]$ is uniform in S_0, and*

$$\mathcal{D} \models (\forall)\ (W \equiv \mathcal{R}[W])$$

By combining Theorems 1 and 3 the following important result is obtained [3] as well:

Theorem 4 *(Soundness and completeness of regression). Let W be a regressable formula of $\mathcal{L}_{sitcalc}$ and $\mathcal{D}$ a basic action theory. Then*

$$\mathcal{D} \models W \quad \textit{iff} \quad \mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$$

2.4. Executable situations

A logical definition of the executable situations is provided [7] by the recursive macro:

$$ex(s) \stackrel{def}{\equiv} s = S_0 \vee (\exists a, s')\ s = do(a, s') \wedge Poss(a, s') \wedge ex(s')$$

For a term $do([\alpha_1, \ldots, \alpha_n], S_0)$ of sort *situation*, in the models of Σ it follows that

$$ex(do([\alpha_1, \ldots, \alpha_n], S_0)) \equiv Poss(\alpha_1, S_0) \wedge Poss(\alpha_2, do(\alpha_1, S_0)) \wedge \cdots \wedge Poss(\alpha_n, do([\alpha_1, \ldots, \alpha_{n-1}], S_0)) \tag{7}$$

2.5. Closure on actions

In Theorem 10 we shall consider a basic action theory that includes the following action closure property as an axiom:

$$(\forall a) \quad \bigvee_{i=1\ldots q} (\exists x_1, \ldots, x_{n_i})\ a = A_i(x_1, \ldots, x_{n_i}) \tag{8}$$

where $A_1, \ldots, A_q$ are all the action function symbols for which action precondition axioms are provided in $\mathcal{D}_{ap}$.

Note that for such a basic action theory, given a formula $\theta(a)$ in which the variable a of sort action occurs free, the following holds

$$\mathcal{D} \models [(\forall a)\ \theta(a)] \equiv \bigwedge_{i=1\ldots q} (\forall x_1, \ldots, x_{n_i})\ \theta(a/A_i(x_1, \ldots, x_{n_i})) \tag{9}$$

[2] Different situation terms are typically mentioned by W, that is, n and $\alpha_1, \ldots, \alpha_n$ vary across the terms $do([\alpha_1, \ldots, \alpha_n], S_0)$ occurring in W. If this is not the case, W can be shown equivalent to a formula uniform in $do([\alpha_1, \ldots, \alpha_n], S_0)$.

2.6. Compactness of basic action theories

Compactness is a property of first order logic theories that in general does not extend to higher order languages. Basic action theories are a fortunate case in which compactness does hold, although they include the second order induction axiom (2) in Σ. This can be stated as a corollary of the relative satisfiability (Theorem 1):

Corollary 5 *(Compactness of basic action theories). Let $\mathcal{D}$ be an infinite basic action theory. $\mathcal{D}$ is unsatisfiable iff there exists a first order finite unsatisfiable subset of $\mathcal{D}$.*

Proof. The *if* part is straightforward. Let us address the *only if*.

$\mathcal{D}$ is unsatisfiable. Let $\mathcal{D}\_$ be the subtheory obtained when the second order induction axiom is removed from $\mathcal{D}$. If $\mathcal{D}\_$ were satisfiable, its subtheory $\mathcal{D}_{S_0} \cup \mathcal{D}_{una}$ would be satisfiable too, and so would $\mathcal{D}$ be by relative satisfiability, which contradicts the hypothesis. Then $\mathcal{D}\_$ must be unsatisfiable and, since it is a first order theory, it is also compact. Therefore there must exist an unsatisfiable finite subset of $\mathcal{D}\_$. □

3. Quantum levels and Reiter's theorem

As recalled in Section 2.2, the domain *Sit* in the models of Σ has the form of a tree rooted in S_0 whose nodes and arcs are respectively the elements of the domains *Sit* and *Act* [7]. There are countably infinitely many levels for this tree. This does not imply that *Sit* is countable, as the branching factor at every node is equal to the cardinality of *Act*; since no constraint for this is commonly given, in the general case *Sit* will take any infinite cardinality across all the models, due to the Lowenheim–Skolem–Tarski theorem [6]. However, if we split a quantification over situations into a double quantification, the first over the levels and the second over the situations belonging to the current level, the former can be syntactically expanded thanks to the countability of the tree's levels. This is formally stated and justified in Lemma 6 below.

First, note that for a formula $\varphi(s)$, in which s is a variable of sort *situation* occurring free, the universal and existential quantifications over the situations belonging to the level l of the situation tree have the form

$$(\forall a_1, \ldots, a_l)\ \varphi(s/do([a_1, \ldots, a_l], S_0))$$

and

$$(\exists a_1, \ldots, a_l)\ \varphi(s/do([a_1, \ldots, a_l], S_0))$$

where $a_1, \ldots, a_l$ are fresh variable symbols of sort *action*.

Lemma 6. *Let $\varphi(s)$ be a formula of $\mathcal{L}_{sitcalc}$, where s is a variable of sort situation occurring free. Let $\mathbb{M}$ be a model of Σ and v an assignment to the free variables occurring in $\varphi(s)$ except s, then*

$$\mathbb{M}, v \models (\forall s)\ \varphi(s) \quad \textit{iff} \quad \mathbb{M}, v \models \{(\forall a_1, \ldots, a_l)\ \varphi(s/do([a_1, \ldots, a_l], S_0))\}_{l \in \mathbb{N}}$$

where $a_1, \ldots, a_l$ are fresh variable symbols of sort action.

Proof. The *only if* part is straightforward. Let us address the *if*.

$$\mathbb{M}, v \models \Sigma \cup \{(\forall a_1, \ldots, a_l)\ \varphi(s/do([a_1, \ldots, a_l], S_0))\}_{l \in \mathbb{N}}. \tag{10}$$

Assume

$$\mathbb{M}, v \not\models (\forall s)\ \varphi(s)$$

that is

$$\mathbb{M}, v \models (\exists s)\ \neg\varphi(s) \tag{11}$$

Now pick a witness in $\mathbb{M}$ for (11), i.e., an element of the domain *Sit* of $\mathbb{M}$, say $\dot{s}$, such that $\varphi^{\mathbb{M},v}(\dot{s})$ is false. Since $\mathbb{M} \models \Sigma$ we know that every situation corresponds to a unique path on the situation tree, which is the sequence of

elements of *Act* connecting the situation to the root of the tree S_0. Formally, it is $\dot{s} = do^{\mathbb{M}}([\dot{a}_1, \ldots, \dot{a}_p], S_0^{\mathbb{M}})$ for some $p \in \mathbb{N}$ and some $\dot{a}_1, \ldots, \dot{a}_p$ elements of the domain *Act* of $\mathbb{M}$. Therefore

$$\mathbb{M}, \nu \not\models (\forall a_1, \ldots, a_p)\ \varphi(s/do([a_1, \ldots, a_p], S_0))$$

which contradicts the initial assumption (10). □

We can now provide the main result of this paper.

Theorem 7. *Let $\mathcal{D}$ be a satisfiable basic action theory and $\varphi(s)$ a first order formula of $\mathcal{L}_{sitcalc}$ such that, for any $n \in \mathbb{N}$ and $\alpha_1, \ldots, \alpha_n$ terms of sort action, the formula $\varphi(s/do([\alpha_1, \ldots, \alpha_n], S_0))$ is regressable.*[3] *Then*

$$\mathcal{D} \models (\forall)\ (\exists s)\ \varphi(s)$$

iff there exists a finite subset Λ of $\mathbb{N}$ such that

$$\mathcal{D} \models \bigvee_{l \in \Lambda} (\forall)\ (\exists a_1, \ldots, a_l)\ \varphi(s/do([a_1, \ldots, a_l], S_0))$$

where $a_1, \ldots, a_l$ are fresh variable symbols of sort action.

Proof. The *if* part is straightforward. Let us address the *only if*.

$\mathcal{D} \models (\forall)\ (\exists s)\ \varphi(s)$. This can also be stated as

$$\mathcal{D} \cup \{(\exists)\ (\forall s)\ \neg\varphi(s)\} \text{ is unsatisfiable.}$$

By Lemma 6 the theory

$$\mathcal{D} \cup \{(\exists)\ (\forall a_1, \ldots, a_l)\ \neg\varphi(s/do([a_1, \ldots, a_l], S_0))\}_{l \in \mathbb{N}}$$

must be unsatisfiable too. Every sentence

$$(\exists)\ (\forall a_1, \ldots, a_k)\ \neg\varphi(s/do([a_1, \ldots, a_k], S_0))$$

can be regressed (Theorem 3) into a sentence uniform in S_0, say ψ_k, such that

$$\mathcal{D} \models [(\exists)\ (\forall a_1, \ldots, a_k)\ \neg\varphi(s/do([a_1, \ldots, a_k], S_0))] \equiv \psi_k \tag{12}$$

Therefore, the set

$$\mathcal{D}' = \mathcal{D} \cup \{\psi_l\}_{l \in \mathbb{N}}$$

is an unsatisfiable theory. In particular, by letting

$$\mathcal{D}'_{S_0} = \mathcal{D}_{S_0} \cup \{\psi_l\}_{l \in \mathbb{N}}$$

it is easy to observe that $\mathcal{D}'$ formally amounts to a new infinite basic action theory

$$\Sigma \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una} \cup \mathcal{D}'_{S_0}$$

Then, by compactness (Corollary 5), there must exist an unsatisfiable finite subset $\mathcal{D}'_f$ of $\mathcal{D}'$. Since $\mathcal{D}$ is satisfiable by hypothesis, $\mathcal{D}'_f$ must contain some (finitely many) elements of $\{\psi_l\}_{l \in \mathbb{N}}$: let Λ be the finite subset of $\mathbb{N}$ containing their l indexes. The finite superset of $\mathcal{D}'_f$

$$\mathcal{D} \cup \{\psi_l\}_{l \in \Lambda}$$

must be unsatisfiable too.

Now, reasoning backward, by (12)

$$\mathcal{D} \cup \{(\exists)\ (\forall a_1, \ldots, a_l)\ \neg\varphi(s/do([a_1, \ldots, a_l], S_0))\}_{l \in \Lambda}$$

[3] Hence, s is the only variable of sort situation in $\varphi(s)$ and occurs free, and no variable of sort situation is mentioned by any term α_i.

is unsatisfiable, that is

$$\mathcal{D} \models \bigvee_{l \in \Lambda} (\forall)\ (\exists a_1, \ldots, a_l)\ \varphi(s/do([a_1, \ldots, a_l], S_0)). \qquad \square$$

This result points out how the expressive power of the situation calculus is affected by the compactness of the basic action theories, which in turn is due to the compactness of the first order theory $\mathcal{D}_{S_0}$ (see Corollary 5). For example, a particular property cannot come true in only one situation whose corresponding sequence of actions has a different length in every model of the theory. More generally, if a basic action theory entails the existence of a situation with a given property, this situation cannot range over the levels of the situation tree in full freedom. Indeed the basic action theory coupled with the property determines what we call the "quantum levels" of the situation tree: given $\mathcal{D}$ and $\varphi(s)$ such that $\mathcal{D} \models (\exists s)\ \varphi(s)$, there must exist a finite subset of levels of the situation tree, *depending only on* $\mathcal{D}$ *and* φ, on which at least one witness for $(\exists s)\ \varphi(s)$ is found. Note that this set is the same in *every model of* $\mathcal{D}$.

A proof of Reiter's Theorem 2 in [2]—rephrased below according to the notations and axiomatization in [3] followed here—can be easily obtained as a variant of the proof of Theorem 7 above. We first introduce some convenient definitions:

Definition 8. Let $\mathcal{D}$ be a basic action theory that additionally includes the action closure axiom (8). Let $k \in \mathbb{N}$, and $G(do([a_1, \ldots, a_k], S_0))$ be a first order formula of $\mathcal{L}_{sitcalc}$ uniform in $do([a_1, \ldots, a_k], S_0)$. Ψ_k is the formula obtained after all the occurrences of the variables $a_1, \ldots, a_k$ and their quantifications in

$$(\forall a_1, \ldots, a_k)\ \neg\left[G(do([a_1, \ldots, a_k], S_0)) \wedge \bigwedge_{i=1\ldots k} Poss(a_i, do([a_1, \ldots, a_{i-1}], S_0))\right] \tag{13}$$

have been replaced by applying the equivalence (9) k times.

After this transformation the predicate $Poss(\ldots)$ occurs consistently with Definition 2 of regressable formulas. The remaining subformula expanded from $G(\ldots)$ inherits regressability from $G(\ldots)$, which is regressable because it is uniform in $do([a_1, \ldots, a_k], S_0)$. Ψ_k is hence regressable, thus let

$$\Gamma_k = \neg\mathcal{R}[\Psi_k]$$

The purpose of the previous definition is to provide a regressable formula equivalent to (13). Indeed $Poss(a_i, do([a_1, \ldots, a_{i-1}], S_0))$ is not in a regressable form (Definition 2). We can then rely on a chain of equivalences:

Proposition 9. *For Definition* 8 *we have*:

$$\mathcal{D} \models (\forall a_1, \ldots, a_k)\ \neg[G(do([a_1, \ldots, a_k], S_0)) \wedge ex(do([a_1, \ldots, a_k], S_0))] \equiv$$

(*by* (7))

$$(\forall a_1, \ldots, a_k)\ \neg\left[G(do([a_1, \ldots, a_k], S_0)) \wedge \bigwedge_{i=1\ldots k} Poss(a_i, do([a_1, \ldots, a_{i-1}], S_0))\right] \equiv$$

(*by Definition* 8)

$$\Psi_k \equiv$$

(*by Definition* 8 *and Theorem* 3)

$$\neg\Gamma_k$$

Theorem 10. *Let* $\mathcal{D}$ *be a satisfiable basic action theory that additionally includes the action closure axiom* (8)*. Let* $G(s)$ *be a first order formula of* $\mathcal{L}_{sitcalc}$ *uniform in the variable of sort situations, which is also the only variable occurring free. Let* Γ_i*,* $i \in \mathbb{N}$*, be the sentences produced from* $G(s/do([a_1, \ldots, a_i], S_0))$ *as in Definition* 8*. Then*

$$\mathcal{D} \models (\exists s)\ [G(s) \wedge ex(s)]$$

iff for some $n \in \mathbb{N}$

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \bigvee_{i=1\ldots n} \Gamma_i$$

Proof. We address the *if* part, which is rather straightforward by Definition 8 and Proposition 9.

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \bigvee_{i=1\ldots n} \Gamma_i$$

By Proposition 9,

$$\mathcal{D} \models \bigvee_{i=1\ldots n} \neg\Psi_i$$

and, again by Proposition 9,

$$\mathcal{D} \models \bigvee_{i=1\ldots n} (\exists a_1, \ldots, a_i)\ [G(do([a_1, \ldots, a_i], S_0)) \wedge ex(do([a_1, \ldots, a_i], S_0))]$$

from which it follows

$$\mathcal{D} \models (\exists s)\ [G(s) \wedge ex(s)]$$

For the *only if* part the proof proceeds as for Theorem 7, provided some additional care for "$ex(s)$" as follows.

Let $\varphi(s) = G(s) \wedge ex(s)$; after the first step of the proof of Theorem 7 we had that the theory

$$\mathcal{D} \cup \{(\forall a_1, \ldots, a_l)\ \neg\varphi(s/do([a_1, \ldots, a_l], S_0))\}_{l\in\mathbb{N}}$$

is unsatisfiable.

For the next step the formula in curly brackets should be in regressable form; to this purpose we can rely on Proposition 9

$$\mathcal{D} \cup \{\Psi_l\}_{l\in\mathbb{N}} \text{ is unsatisfiable}$$

Proceeding as for Theorem 7, it can be shown that there exists a finite subset Λ of $\mathbb{N}$ such that

$$\mathcal{D} \models \bigvee_{l\in\Lambda} \neg\Psi_l$$

By Theorem 4 and Proposition 9, we have

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \bigvee_{l\in\Lambda} \Gamma_l$$

By choosing $n = \max \Lambda$, we can conclude

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \bigvee_{i=1\ldots n} \Gamma_i \qquad \square$$

4. Conclusion

This paper contributes the proof of a key property of Reiter's variant of the situation calculus. The property was first stated as a theorem in a seminal paper [2], but its proof was not provided. We have rephrased and proved the theorem in a more general form. To this end, we have pointed out that this result is a consequence of the compactness of the situation calculus' basic action theories, which significantly affects the expressivity of the formalism.

The theorem constrains how the situations satisfying an existential assertion can take place over the levels of the situation tree. If one or more situations exist in which a given property (e.g., a goal) is satisfied, at least one of these situations must be found on certain levels of the tree, within a finite distance from the initial situation. The set of such levels and the distance is the same in every model. This is a key formal guarantee for generating a plan as a finite syntactic term of the language when distinct models contain different plan solutions, as may be the case when knowledge is incomplete.

This important property of the situation calculus may not hold if a basic action theory does not comply with some of the constraints assumed in [2], and later required by definition in [3].

For instance, if the initial database ($\mathcal{D}_{S_0}$) is not first order (Section 2.2), Theorems 7 and 10 may not hold, since their proofs rest on the compactness of the basic action theories, which in turn is guaranteed by the fact that the initial

database is first order (Corollary 5). This case arises in the formalization of the chop problem [4,5], which is a simple action domain showing that an agent might not know how to execute a program with non-deterministic choices and sensing actions, even though its executability is formally guaranteed by the underlying basic action theory known to the agent. This basic action theory is purposely designed so that the first situation in which the goal is reached is at an arbitrary distance from the initial situation in every model. It follows that there can be no such quantum levels as predicted by Theorem 7, nor can there be such a finite number n as predicted by Theorem 10. This is only possible because the hypotheses of the two theorems do not hold: the initial database is second order. Indeed, since quantifying over the natural numbers is essential to the correctness of the chop-problem formalization, the exact standard model of $\mathbb{N}$ must be selected. The second order induction axiom of Peano's axiomatization of arithmetic is therefore required in the initial database, in order to rule out the non-standard models containing Z-chains [6].

The chop problem generalizes the problem of building a plan that is known to exist according to a given basic action theory—which was originally addressed by Reiter's theorem. The insight gained here is that this generalization, too, falls within the scope of the theorem; the chop problem, or one with a similar structure, cannot arise in basic action theories with a first order initial database.

Acknowledgements

I am grateful to Fiora Pirri for her stimulating discussion and for her comments on an earlier draft. I would like to thank Giuseppe De Giacomo for acquainting me with the chop problem. Two anonymous referees have provided valuable suggestions for improving this paper.

References

[1] J. McCarthy, Situations, actions and causal laws, in: M. Minsky (Ed.), Semantic Information Processing, MIT Press, Cambridge, MA, 1968, pp. 410–417.
[2] R. Reiter, The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression, in: V. Lifschitz (Ed.), Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, Academic Press, San Diego, CA, 1991, pp. 359–380.
[3] F. Pirri, R. Reiter, Some contributions to the metatheory of the situation calculus, J. ACM 46 (3) (1999) 261–325.
[4] Y. Lespérance, H. Levesque, F. Lin, R. Scherl, Ability and knowing how in the situation calculus, Studia Logica 66 (1) (2000) 165–186.
[5] S. Sardina, G. De Giacomo, Y. Lespérance, H. Levesque, On ability to autonomously execute agent programs with sensing, in: Proceedings of the 4th International Workshop on Cognitive Robotics (CoRobo-04), Valencia, Spain, 2004, pp. 88–93.
[6] H.B. Enderton, A Mathematical Introduction to Logic, Academic Press, San Diego, CA, 1972.
[7] R. Reiter, Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems, MIT Press, Cambridge, MA, 2001.