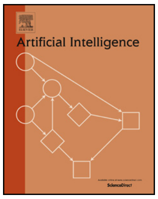
Artificial Intelligence 237 (2016) 172–203

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Bounded situation calculus action theories

Giuseppe De Giacomo [a,*], Yves Lespérance [b,*], Fabio Patrizi [c,**]

[a] *Dipartimento di Ingegneria informatica, automatica e gestionale, Sapienza Università di Roma, Italy*
[b] *Department of Electrical Engineering and Computer Science, York University, Toronto, ON, Canada*
[c] *KRDB Research Centre – Faculty of Computer Science, Free University of Bozen–Bolzano, Italy*

ARTICLE INFO

Article history:
Received 7 June 2015
Received in revised form 7 April 2016
Accepted 20 April 2016
Available online 3 May 2016

Keywords:
Knowledge representation
Reasoning about action
Situation calculus
Verification

ABSTRACT

In this paper,[1] we investigate bounded action theories in the situation calculus. A bounded action theory is one which entails that, in every situation, the number of object tuples in the extension of fluents is bounded by a given constant, although such extensions are in general different across the infinitely many situations. We argue that such theories are common in applications, either because facts do not persist indefinitely or because the agent eventually forgets some facts, as new ones are learned. We discuss various classes of bounded action theories. Then we show that verification of a powerful first-order variant of the μ-calculus is decidable for such theories. Notably, this variant supports a controlled form of quantification across situations. We also show that through verification, we can actually check whether an arbitrary action theory maintains boundedness.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

The situation calculus [65,75] is a well-known first-order formalism with certain second-order features for representing dynamically changing worlds. It has proved to be an invaluable formal tool for understanding the subtle issues involved in reasoning about action. Its comprehensiveness allows us to place all aspects of dynamic systems in perspective. Basic action theories let us capture change as a result of actions in the system [73], while high-level languages such as Golog [58] and ConGolog [26] support the representation of processes over the dynamic system. Aspects such as time [74], knowledge and sensing [79], probabilities and utilities [14], and preferences [11], have all been addressed. The price of such a generality is that decidability results for reasoning in the situation calculus are rare, e.g., [86] for an argument-less fluents fragment, and [49] for a description logic-like two-variable fragment. Obviously, we have the major feature of being able to rely on regression to reduce reasoning about a given future situation to reasoning about the initial situation [75]. Generalizations of this basic result such as just-in-time histories [33] can also be exploited. However, when we move to temporal properties, virtually all approaches are based on assuming a finite domain and a finite number of states, and often rely on propositional modal logics and model checking techniques [6,63]. There are only few exceptions such as [21,32,82], which develop incomplete fixpoint approximation-based methods.

---

* Corresponding authors.
** Principal corresponding author.
E-mail addresses: degiacomo@dis.uniroma1.it (G. De Giacomo), lesperan@cse.yorku.ca (Y. Lespérance), patrizi@dis.uniroma1.it (F. Patrizi).

[1] A preliminary version of this paper appeared as [27].

http://dx.doi.org/10.1016/j.artint.2016.04.006
0004-3702/© 2016 Elsevier B.V. All rights reserved.

In this paper, we present an important new result on decidability of the situation calculus, showing that *verification of bounded action theories is decidable*. Bounded action theories are basic action theories [75], where it is entailed that in all situations, the number of object tuples that belong to the extension of any fluent is bounded. In such theories, the object domain remains nonetheless infinite and an infinite run may involve an infinite number of objects, though at every single situation the number of objects we predicate on is finite and, in fact, bounded.

But why should we believe that practical domains conform to this boundedness assumption? While it is often assumed that the law of inertia applies and that fluent atoms persist indefinitely in the absence of actions that affect them, we all know that pretty much everything eventually decays and changes. We may not even know how the change may happen, but nevertheless know that it will. Another line of argument for boundedness is epistemic. Agents remember facts that they use and periodically try to confirm them, often by sensing. A fact that never gets used is eventually forgotten. If a fact can never be confirmed, it may be given up as too uncertain. Given this, it seems plausible that in several contexts an agent's knowledge, in every single moment, can be assumed to be bounded. While these philosophical arguments are interesting and relate to some deep questions about knowledge representation, one may take a more pragmatic stance, and this is what we do here. We identify some interesting classes of bounded action theories and show how they can model typical example domains. We also show how we can transform arbitrary basic action theories into bounded action theories, either by blocking actions that would exceed the bound, or by having persistence (frame axioms) apply only for a finite number of steps. Moreover we show that we can effectively check whether any arbitrary theory with a bounded initial situation description remains bounded in all executable situations (to do so we need to use verification).

The main result of the paper is that verification of an expressive class of first-order μ-calculus temporal properties in bounded action theories is decidable and, precisely, EXPTIME-complete. This means that we can check whether a system or process specified over such a theory satisfies some specification even if we have an infinite domain and an infinite set of situations or states. In a nutshell, we prove our results by focussing on the *active domain* of situations, i.e., the set of objects for which some atomic fluent holds; we know that the set of such active objects is bounded. We show that essentially we can abstract situations whose active domains are *isomorphic* into a single state, and thus, by suitably abstracting also actions, we can obtain an *abstract finite transition system* that *satisfies exactly the same formulas* of our variant of the μ-calculus.

This work is of interest not only for AI, but also for other areas of computer science. In particular it is of great interest for the work on data-aware business processes and services [53,45,38]. Indeed while there are well-established results and tools to analyze business processes and services, without considering the data manipulated, when data are taken into account results are scarce. The present work complements that in, e.g., [37,4,9,5,10], and hints at an even more profound relevance of the situation calculus in those areas [64]. More generally, our results can be recast in other formalisms for reasoning about action, both in AI and in CS.

The rest of the paper is organized as follows. In Section 2, we briefly review the situation calculus and basic action theories. Then in Section 3, we define bounded action theories. Then, in Section 4, we discuss various ways of obtaining bounded action theories, while showing that many practical domains can be handled. In Section 5, we introduce the $\mu\mathcal{L}_p$ language that we use to express first-order temporal properties. After that, we show that verification of $\mu\mathcal{L}_p$ properties over bounded action theories is decidable, first in the case where we have complete information about the initial situation (Section 6), and then in the general incomplete information case (Section 7). In Section 8, we characterize the worst-case computational complexity of the problem as EXPTIME-complete. In Section 9, we give a technique based on our verification results, to check whether an arbitrary basic action theory maintains boundedness. In Section 10, we review the related literature. Finally, in Section 11, we conclude the paper by discussing future work topics.

2. Preliminaries

The *situation calculus* [65,75] is a sorted predicate logic language for representing and reasoning about dynamically changing worlds. All changes to the world are the result of *actions*, which are terms in the language. We denote action variables by lower case letters a, action types by capital letters A, and action terms by α, possibly with subscripts. A possible world history is represented by a term called a *situation*. The constant S_0 is used to denote the initial situation where no actions have yet been performed. Sequences of actions are built using the function symbol *do*, where $do(a, s)$ denotes the successor situation resulting from performing action a in situation s. Besides actions and situations, there is also the sort of *objects* for all other entities. Predicates and functions whose value varies from situation to situation are called *fluents*, and are denoted by symbols taking a situation term as their last argument (e.g., $Holding(x, s)$, meaning that the robot is holding object x in situation s). For simplicity, and without loss of generality, we assume that there are no functions other than constants and no non-fluent predicates. We denote fluents by F and the finite set of primitive fluents by $\mathcal{F}$. The arguments of fluents (apart from the last argument which is of sort situation) are assumed to be of sort object.

Within this language, one can formulate action theories that describe how the world changes as the result of the available actions. Here, we concentrate on *basic action theories* as proposed in [67,75]. We also assume that there is a *finite number of action types*. Moreover, we assume that there is a countably infinite set of object constants $\mathcal{N}$ for which the unique name

Download English Version:

https://daneshyari.com/en/article/6853105

Download Persian Version:

https://daneshyari.com/article/6853105

Daneshyari.com