Artificial Intelligence 175 (2011) 25–48

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Non-Markovian control in the Situation Calculus ☆

Alfredo Gabaldon

Center for Artificial Intelligence, New University of Lisbon, Lisbon, Portugal

ARTICLE INFO

Article history:
Available online 3 April 2010

Keywords:
Reasoning about actions
Situation Calculus

ABSTRACT

In reasoning about actions, it is commonly assumed that the dynamics of domains satisfies the *Markov Property*: the executability conditions and the effects of all actions are fully determined by the present state of the system. This is true in particular in Reiter's Basic Action Theories in the Situation Calculus. In this paper, we generalize Basic Action Theories by removing the Markov property restriction, making it possible to directly axiomatize actions whose effects and executability conditions may depend on past and even alternative, hypothetical situations. We then generalize Reiter's regression operator, which is the main computational mechanism used for reasoning with Basic Action Theories, so that it can be used with non-Markovian theories.

Since the 1960's when John McCarthy's papers (in particular the 1969 paper with Pat Hayes) appeared introducing the Situation Calculus, researchers have been studying and working on this language for reasoning about dynamic domains. The Situation Calculus, one of John's many great inventions, is the topic of this paper and I am delighted to have this opportunity to make a contribution to a special issue in John's honor.

1. Introduction

An assumption commonly made in formalisms for reasoning about the effects of actions is the so called *Markov property*: the executability of an action and its effects are entirely determined by the current state or situation. In particular, Reiter's Basic Action Theories [2], a Situation Calculus [3,4] based axiomatization, define the value of a fluent after the execution of an action in terms of a formula that can only talk about the situation in which the action would be executed. The preconditions of an action are specified by formulas with the same restriction. In this paper we generalize Basic Action Theories by removing this restriction. The generalized theories will allow the executability conditions and the effects of an action to depend not only on what holds when the action is to occur, but also on whether certain conditions were satisfied at different points in the past and even alternative hypothetical evolutions of the system.

As an example, imagine a robot that works in a biological research facility with different safety-level areas. The dynamics is such that a material will be considered contaminated after the robot touches it if the robot has been to a low safety area or has directly been in contact with a hazardous material, and has not been to the disinfection station since then. So the effect of touching the material depends on the history of robot activities. We could also imagine that the robot cannot execute the action $open(Entrance, Lab1)$ if $temp(Lab1) > 30$ was ever true since the last time $closed(Entrance, Lab1)$ occurred. The latter is an example of an action with non-Markovian preconditions.

In simple scenarios, it is not difficult to extend a theory to preserve the necessary history by means of new state variables, especially when the domain is finite. But in complex domains it may not be obvious how to do it, and the

☆ A preliminary abstract of this paper appeared in Proc. of AAAI'02 (A. Gabaldon (2002) [1]).
E-mail address: ag@di.fct.unl.pt.

0004-3702/$ – see front matter
doi:10.1016/j.artint.2010.04.012

resulting theory may be substantially more complex, with a larger number of state variables and corresponding axioms describing their dynamics.

Our goal in this paper is to generalize Reiter's Basic Action Theories [5,2] by removing the Markov property requirement and generalize the main reasoning mechanism used with these theories, namely the regression operator $\mathcal{R}$, so that it can be used with non-Markovian theories, and applied to Situation Calculus formulas that refer to the past, to alternative evolutions, and to definite future situations.

This generalized regression operator is not only useful in cases where the action theory is non-Markovian. Even if the background theory is Markovian, the operator is useful for answering queries that refer to past situations through quantification and the subsequence relation $\sqsubset$. This is not possible with Reiter's original regression operator. In [2, Section 4.8], Reiter presents a few specialized procedures for evaluating certain *historical queries* with respect to a database log (a sequence of ground action terms) and a Markovian action theory. Those queries are a very small subset of the class of queries that the generalized regression operator we present here can handle.

Our work is relevant to a variety of research problems that involve the formalization of dynamic properties:

(1) Some work in database theory has been concerned with the semantics of dynamic integrity constraints [6,7]. These constraints are typically expressed in Past Linear Temporal Logic, a logic with temporal connectives *Previous*, *Sometime* in the past, *Always* in the past, and *Since*. In a formalization of a database system in the Situation Calculus, such temporal connectives amount to references to past situations, and the constraints to restrictions on when a sequence of actions can be considered a "legal" database system evolution. These past temporal logic connectives have an encoding as formulas in the non-Markovian Situation Calculus and hence the latter can be used as a logical framework for the study, specification and modeling of databases with dynamic integrity constraints. The advantage of carrying out such work in this framework is that all the different aspects of the problem, i.e. database dynamics, transactions and constraints, can be captured within the same Situation Calculus framework.

(2) Also in the area of databases, more specifically in work on database transaction systems, the *rollback* operation, which reverts a database back to its original state after a long transaction fails or is canceled, clearly has a non-Markovian flavor: its effects depend not on what is true in the state it is executed, but on the state right before the transaction being reversed started. Indeed, Kiringa [8] and Kiringa and Gabaldon [9,10] present logical specifications of database transactions in the non-Markovian Situation Calculus.

(3) In planning, domain dependent knowledge for search control has been used with great success [11,12]. Bacchus and Kabanza's forward-chaining planning system, TLPlan, uses search control knowledge in the form of temporal logic formulas. The same approach has been applied in the Situation Calculus with some simple planners written in Golog [2]. The latter planners perform a forward search, eliminating partial plans if they lead to "bad situations." Search control knowledge is encoded through a predicate $badSituation(s)$ whose definition is restricted to properties of the current situation s. The generalization of the action theories and the regression operator we shall develop here allows the definition of this predicate to refer to any situation that precedes s and bounded future situations. As we mention above, past temporal logic expressions can be encoded as Situation Calculus formulas suitable for regression with our generalized operator and be used in the definition of $badSituation(s)$. In other words, the generalized regression operator allows one to use temporal search control knowledge of a similar form and expressive power as used in TLPlan directly in Golog planners with the $badSituation(s)$ predicate. Search control knowledge in this context is further explored in our recent work [13,14].

(4) Another area where non-Markovian features arise naturally is in specifying reward functions in decision theoretic planning. There, agents are often rewarded based on their long-term behavior rather than just on the current state of affairs. Bacchus, Boutilier and Grove [15,16] have developed techniques for solving such non-Markovian Decision Processes. More recent work on non-Markovian rewards appears in [17,18].

(5) Finally, some time ago John McCarthy [19] described a programming language called "Elephant 2000" which, among other features, "does not forget." This is a language that would allow one to write programs that explicitly and directly refer to past states of the programming environment. The generalized regression operator we present here could form the foundation for a non-forgetting Golog [20]. Such a dialect of Golog would allow test conditions that refer to the past, for instance, as in the statement **if** $(P$ **since** $Q)$ **then** δ.

This paper is organized as follows: we start in Section 2 with an overview of the Situation Calculus and Reiter's Basic Action Theories. In Section 3 we introduce a class of Situation Calculus formulas that can refer to past and finite future situations and can be regressed, and based on this, our generalization of action theories for non-Markovian control. In Section 4 we present a regression operator that works for those formulas and theories and prove its correctness, followed by a Prolog implementation in Section 5 and our concluding remarks in Section 6.

2. Overview of the Situation Calculus and Basic Action Theories

The Situation Calculus [3,4] is a dialect of classical logic for representing and reasoning about dynamically changing worlds. A theory in this language consists of a collection of axioms describing how the world changes when actions occur. Accordingly, the ontology of the Situation Calculus includes three main ingredients: actions, situations, and fluents. Situ-