Available online at www.sciencedirect.com
ScienceDirect
Data & Knowledge Engineering 59 (2006) 397–434
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

The ramification problem in temporal databases: Changing beliefs about the past

Nikos Papadakis *, Grigoris Antoniou, Dimitris Plexousakis

Department of Computer Science, University of Crete, Institute of Computer Science, FORTH, GR-711 10 Heraklion, Crete, Greece

Received 21 February 2005; accepted 21 September 2005
Available online 18 October 2005

Abstract

In this paper we study the ramification problem in the setting of temporal databases. Standard solutions from the literature on reasoning about action are inadequate because they rely on the assumption that fluents persist, and because actions have effects on the next situation only. In this paper we provide a solution to the ramification problem based on an extension of the situation calculus and the work of McCain and Turner. More specifically, we study the case where the effects of an action refer to the past, a particularly complex problem.

Keywords: Ramification problem; Temporal databases; Common sense reasoning; Knowledge representation and reasoning; Belief revision

1. Introduction

The ramification and qualification problems [9] are hard problems that arise in robotics, software engineering, in databases, and generally speaking all systems exhibiting a dynamic behavior. In this paper we consider the case of temporal databases.

Let us illustrate the problems. Suppose we are interested in maintaining a database that describes a simple circuit, which has two switches and one lamp (Fig. 1A).

The circuit's behavior is described by the following integrity constraints. First, when the two switches are up, the lamp must be lit. Second, if one switch is down then the lamp must not be lit. The integrity constraints are expressed as the following formulas, employing predicates *up* and *light*:

$$up(s_1) \wedge up(s_2) \equiv light$$
$$\neg up(s_1) \supset \neg light$$
$$\neg up(s_2) \supset \neg light$$

* Corresponding author.
E-mail addresses: npapadak@csd.uch.gr (N. Papadakis), ga@csd.uch.gr (G. Antoniou), dp@csd.uch.gr (D. Plexousakis).

0169-023X/$ - see front matter
doi:10.1016/j.datak.2005.09.003

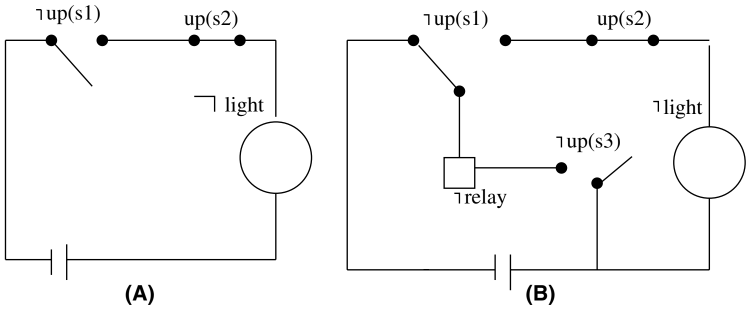

Fig. 1. The circuit.

Action *toggle_switch* changes the situation of a switch as follows:

$$toggle\_switch(s) \supset up(s) \quad \text{if } \neg up(s)$$
$$toggle\_switch(s) \supset \neg up(s) \quad \text{if } up(s)$$

The above propositions describe the direct effects of the action *toggle_switch*. A situation is called consistent when it satisfies all integrity constraints. Assume that the circuit is in situation $S = \{up(s_1), \neg up(s_2), \neg light\}$. The situation S is consistent, because it satisfies all integrity constraints. Now assume that we execute the action $toggle\_switch(s_2)$. This action has a sdirect effect to change the state of switch s_2 from $\neg up(s_2)$ to $up(s_2)$. Now the situation of the circuit is $S_1 = \{up(s_1), up(s_2), \neg light\}$. This situation is inconsistent, because it violates the first integrity constraint. The reasonable conclusion is that the lamp must be lit. So the final situation is $S_2 = \{up(s_1), up(s_2), light\}$. The change of the condition of the lamp is the indirect effect of the action $toggle\_switch(s_2)$. Notice that indirect effects exist because of the presence of integrity constraints. *The ramification problem refers to the concise description of the indirect effects of an action in the presence of constraints.*

Several ways for addressing the ramification problem have been suggested in the literature. The majority of them are based on the situation calculus [9]. The situation calculus is a second-order language that represents the changes which occur in a domain, as results of actions. One possible evolution of the world is a sequence of actions and is represented by a first-order term. The situation at which no action has occurred yet, is called the initial situation (S_0). There is a binary function $do(a, S)$ which yields the new situation resulting from the execution of action a in the situation S. Predicates, called fluents, may change truth value from one situation to another. Similarly, one can represent functions whose values are dependent on the situations on which they are evaluated (functional fluents).

The simplest of the technique suggested in the literature is the *minimal-change* approach [21]. Thiss suggests that, when an action occurs in a situation S, we need to find a consistent situation S' which has fewer changes from the situation S (S' is closer to S than to any other situation).

Another solution is the *categorization* of fluents [5–7]. Fluents are categorized as primary and secondary. A primary fluent may change only as a direct effect of an action, while a secondary fluent may change only as an indirect effect of an action. After an action takes place, we choose the situation with the fewest changes in primary fluents. The categorization of fluents solves the ramification problem only if all fluents can be categorized. If some fluents are primary for some actions and secondary for some others this solution is not satisfactory.

As we can observe from the above examples, the change of fluent f's truth value potentially affects the truth value of some fluents, while it does not affect that of the others. We define a binary relation I between fluents as follows: if $(f,f') \in I$, then a change in fluent f's value may affect the value of f'. In the above example, $(up(s_1), light) \in I$, whereas $(up(s_1), up(s_2)) \notin I$. A fluent could change or remain unchanged after an action. This depends on the context in which an action takes place.

Causal relationships [8,19,20] capture this dependence between an action and an indirect effect. A causal relationship has the form