Temporal Reasoning Involving Counterfactnais and Disjunctions

Thomas Dean
Yale University

ABSTRACT

This paper describes a mechanism for nonmonotonic temporal reasoning involving counterfactuals and disjunctions. The mechanism supports a method for exploring alternatives well suited to automatic planning. The application of these techniques to robot problem solving is discussed with an emphasis on reasoning about exclusive choices and monitoring the continued warrant and effectiveness of prevention tasks.

I. Introduction

A critical part of planning involves selecting a plan for achieving a particular task. A reduction type planner (e.g. NOAH [7]), faced with the problem of formulating a plan for the conjunction of two or more nonprimitive tasks, is often forced to make plan selection decisions on the basis of incomplete knowledge. The planner cannot anticipate interactions with tasks whose reduction (or detailed specification) it has yet to explore. Given that there exists no strong warrant for choosing one plan over another why should a planner make a choice at all? In certain situations it would seem that procrastination is appropriate.

The problem here is similar to one addressed by Sacerdoti in developing the NOAH planner. NOAH employed a strategy known as *least commitment* and a data structure known as a *procedural net* to avoid making ordering decisions until they were warranted by interactions discovered in the course of planning. This was a reaction to previous planners' use of what is called the *linear assumption:* the tasks in a conjunction of tasks to be solved are assumed to be independent and hence there is no harm in committing to some arbitrary order for the purposes of planning. The problem was that by committing early (and arbitrarily) the planner might have to explore a large number of alternative orderings before finding one that worked. Most planners [7] [9] [10] have made a similar assumption as regards plan selection; namely that in the absence of information to the contrary any plan for achieving a task will suffice.

In order to deal with the problems that arise as a result of making this assumption a number of strategies have been proposed. NOAH represented alternative plans for achieving a given task as a disjunction (implicitly exclusive) of plans. The planner expanded the current procedural net with each disjunct noting the type and number of interactions: this information served as the basis for choosing among the alternatives. A more general technique employing a context mechanism was proposed by Wilkins [11]. The problem with both of these approaches is that the planner is responsible for proposing descriptions of the world which follow from making certain decisions (choosing among alternatives). If a planner wanted to make sure it hadn't missed some fortuitous combination of choices it would have to construct and evaluate all such worlds (combinations of choices). The planner had no way of anticipating good and bad interactions without exploring the alternatives. What was really needed was an efficient method of discovering sets of choices to avoid (those that lead to problems) and sets of choices to consider seriously (those that consolidate effort).

If you are entertaining a number of apparently independent alternatives it seems reasonable that you are able to notice interactions, both advantageous and detrimental, involving sets of these alternatives. Suppose that I have a task of mailing a package and I'm considering either driving to the local UPS depot or using the regular mail service and walking to the post office. If I've recently thought about buying some stamps then I might want to take advantage of the opportunity of being at a location selling stamps afforded by the latter. Of course some consolidations are not possible. A planner must have some means of ignoring deductions based on effects achieved by exclusive alternatives. If you are planning a weekend trip but undecided on whether to spend it in a rural or an urban setting it would be a bit silly to formulate a plan which closely couples a relaxing walk in the countryside with a Broadway show.

In addition to reasoning about alternatives and aspects of the world that the planner has control over, a planner quite often finds itself in the position of reasoning about ways the world might have been had events turned out otherwise (counterfactuals). To reason about methods for preventing some predicted unpleasantness we have to be careful to distinguish the circumstances that prompted us to action from those brought about by our attempts to avoid the predicted events. I might plan to deposit a sum of money in my checking account to avoid the penalty from an overdraft. Having determined to do so it is not likely I will be fined but this shouldn't somehow lessen my resolve to actually carry out the actions required to make the deposit. Silly as this may sound, it requires some sophistication to avoid just such confusions.

This initial discussion was meant to outline some problems which have not been handled adequately by previous planning systems. The examples were meant to illustrate a class of situations which require reasoning about alternative, possibly exclusive, descriptions of the world. In the remainder of this paper I will describe a temporal reasoning module designed as part of a planning system which supports this sort of hypothetical reasoning. This short paper cannot hope to satisfy the curiosity of all and I encourage the interested reader to refer to [2] for a more complete exposition.

II. Planning and temporal reasoning

This section outlines the module used for temporal reasoning in the FORBIN planner [6]. This module is referred to as a *time map manager* (TMM) [1] and is loosely based on the representation described in [4]. A *time map* is a data structure which captures what is known about events occurring over time (e.g. their duration and relative ordering) and the effects of processes and actions (e.g. the persistence or the period of time over which such effects can be said to endure). The time map itself consists of a network whose nodes are points corresponding to the begin and end of what are called *time tokens*. A time token denotes an interval associated with the occurrence of an event or an instance of a fact being believed to become true and persist for some period. The points in the network are connected by arcs indicating *constraints* on the distance separating points. A constraint is represented as an interval on the real number line denoting an upper and lower bound on the distance separating two points in the time map.

The TMM p>erforms a rather simple sort of temporal reason maintenance. The system uses data dependencies and constraint propagation to ensure that whenever two time tokens asserting contradictory facts (or mutually exclusive states) are ordered as to their beginning points the earlier will be constrained (or *clipped)* to end before the later.

Plans generally have prerequisites or facts which must be true in order that the plan be applicable in a given situation. Plan choice involves selecting a plan and an interval in which that plan's prerequisites are predicted to be true. Having made such a selection the planner creates tokens corresponding to the plan steps (subtasks) and their effects, adds them to the time map along with associated constraints, and asserts that the plan is *plausible.* This assertion is however defeasible. Other tasks might be introduced during subsequent planning that clip the persistence of a prerequisite fact thereby invalidating the plan. The TMM monitors the continued validity of plausibility assertions by setting up nonmonotonic data dependency justifications composed of assertions called called *protections* (after [8]). A fact Q is said to be protected throughout an interval, ptl to pt2, just in case there exists a time token asserting Q that begins before ptl and it's consistent to believe that it doesn't end before pt2.

If a protection fails then the TMM notifies the planner of any plans threatened by the failed protection. The failure is annotated in such a way as to facilitate corrective action. The TMM also anticipates possible protection failures and suggests ordering constraints to avoid undesirable interactions.

HI. Handling disjunctions

Still the machinery outlined above won't allow us to represent protections of the form "a task T for preventing an event E is warranted just in case E would occur if T didn't". Neither will it allow us to consider more than one reduction for a given task at a time. The latter involves reasoning about disjunctions.

The techniques supporting this sort of reasoning rely upon a method for computing the choices warranting assertions in the time map. Choices (e.g. alternative plans or ordering decisions) are represented as boolean variables. Each assertion is labeled with a boolean formula indicating under which choices the assertion is believed. Labels are computed using a method similar to that employed in McDermott's context mechanism [5]. McDermott's system was designed to reason about a large number of contexts sequentially. It demands that the user specify a particular context (set of choices) to consider. It is essential to our problem that we be able to reason about many sets of choices simultaneously. The mechanism must also be able to ignore certain sets of choices (e.g. those including exclusive alternatives). The TMM employs techniques similar to those used by deKleer [3] to reason about exclusive alternatives and disregard sets of choices no longer deemed worth considering. The algorithms are described in detail in [2].

To illustrate how the TMM works in reasoning about choices I'll provide an abbreviated example drawn from the machine shop domain. Suppose that the robot operator in a job shop is contracted to machine a special order flange from a blank casting. The task requires that a certain diameter hole be drilled and one side of the casting be faced (i.e. cut or milled flush). These two subtasks can be performed in any order. Both drilling and facing can be done on either the vertical mill or on the engine lathe. Suppose that the robot chooses to expand the drilling task using two exclusive plans, one using the mill and another using the lathe. I'll ignore all subtasks except for prerequisite tasks responsible for installing attachments. Initially the mill has the face milling attachment installed and the lathe has a collet chuck. Drilling on the mill will require a fly cutting attachment while a four jaw chuck is required for drilling on the lathe. The facing task is expanded next and the robot, aided by the TMM, chooses to explore two options: use the lathe, taking advantage of the fact that the four jaw chuck is already in place or use the mill (in this case requiring the collet chuck) and schedule the facing task before the task to install the fly cutter. At the time the facing task alternatives are submitted to the TMM it sets up the necessary protections to monitor the tasks' plausibility, and suggests a number of constraints to help avoid protection violations. Notice that while the two tasks, drilling and facing, are still independent, the plans chosen for them are now very much interlinked. To face using the lathe requires that the robot choose to drill using the lathe. If the robot chooses to both drill and face using the mill then it must do the facing task first in order to avoid a protection failure.

This just begins to demonstrate the utility of temporal reasoning about choices. The TMM handles dependent choices (nested decisions) and provides considerable assistance at plan selection time in pointing out options that give rise to advantageous or disadvantageous interactions. This same machinery is also used for reasoning about counterfactuals and we'll consider how in the next section.

IV. Complex protections

For the most part, in reasoning about hypothetical situations, what we want is a virtual copy of some existing situation with just a few changes. For instance suppose that I'm concerned about a situation in which I ignore a dozen or so parking tickets and the city tows my car. So I consider another situation in which I bribe a city official to indefinitely postpone action on my case. My warrant for paying the bribe depends upon the immediate threat of having my car towed. If that threat evaporates (e.g. the news reports a backlog in the handling of traffic violations which has forced the city to issue amnesty for all offenders) then I should definitely question my motivation for giving money to that official. On the other hand if I discover that my method for handling the situation is ineffectual (e.g. the official has been dismissed having been indicted for accepting bribes) then I had better consider alternative methods.

This sort of reasoning has often been modeled along the lines of a conditional proof where introducing and discharging assumptions is considered analogous to pushing and popping contexts. There are a number of problems with this, most of them stemming from the nonmonotonic character of reasoning about beliefs changing over time. From a computational point of view conditional proofs cache deductions in order to ease the computational burden incurred by continually rederiving the same formulae. Once one allows nonmonotonicity the process of caching is considerably complicated, as all deductions are potentially defeasible. A conditional proof in these circumstances must be represented as an ongoing computation. The TMM handles this in terms of reasoning about choices.

Let's consider the plight of a maintenance robot in a factory who is informed at 8:00 AM that OSHA (the U.S. government agency concerned with occupational health and safety) will make an inspection of his factory sometime after 12:00. The robot makes a critical assessment of his work space, mindful of the OSHA rules, notices that the main corridor is cluttered with scrap metal, and predicts that he will be fined for the safety hazard. The robot then determines to clean the aisles before noon in order to avoid the fine. In order to monitor the warrant for and effectiveness of the task to clean the aisles the TMM tags the token for the clean-up task as a choice. If the robot schedules work that will result in the corridor getting cluttered after the clean-up but before the inspection then the TMM will notice that the clean-up task is no longer effective. If on the other hand someone else cleans the corridor or the OSHA visit is called off then the clean-up task will no longer be warranted. The TMM monitors the continued warrant for prevention tasks essentially by considering a world in which the prevention task was never executed.

V. Conclusion

The TMM is applicable in planning domains in which for most tasks the planner has a number of alternative plans and the choice of alternative can make a significant difference in the efficiency of the composite plan.

References

[I] Dean, T.L. *Planning and Temporal Reasoning under Uncertainty.* In Proc. IEEE Workshop on Principles of Knowledge-Based Systems, Denver, Colorado, IEEE Dec 1984.

[2] Dean, T.L. *Temporal Imagery: An Approach to Reasoning about Time for Planning and Problem Solving.* Technical Report (forthcoming), Yale University 1984.

[3]deKleer, J. *Choices Without Backtracking.* In Proc. AAAI 84, Austin, Texas, AAAI, Aug 1984.

[4] McDermott, D.V. *A Temporal Logic for Reasoning about Processes and Plans.* Cognitive Science, 6:101-155, 1982.

[5] McDermott, D.V. *Contexts and Data Dependencies: a Synthesis.* In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 3, pages 237-246, May 1983.

[6] Miller D.P., Firby, R.J., Dean T.L. *Deadlines, Travel Time, and Robot Problem Solving.* In Proc IJCAI-85, Los Angeles, CA, IJCAI, Aug 1985.

[7] Sacerdoti, E. *A Structure for Plans and Behavior.* American Elsevier Publishing Company, Inc., 1977.

[8] Sussman, Gerald J. *A Computer Model of Skill Acquisition.* American Elsevier Publishing Company Inc., 1975.

[9] Tate, A. *Generating Project Networks.* In Proc. IJCAI-5., IJCAI, 1977.

[10] Vere, S.A. *Planning in Time: Windows and Durations for Activities and Goals.* In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 3, pages 246-267, May 1983.

[11] Wilkins, D. *Domain Independent Planning: Representation and Plan Generation.* Artificial Intelligence, 1984.