Monitors as Responses to Questions: Determining Competence

Eric Mays

Department of Computer and Information Science
Moore School of Electrical Engineering/D2
University of Pennsylvania
Philadelphia, Pa. 19104

ABSTRACT

This paper discusses the application of a propositional temporal logic to determining the competence of a monitor offer as an extended response by a question-answering system. Determining monitor competence involves reasoning about the possibility of some future state given a description of the current state and possible transitions.

I INTRODUCTION

The offer of a monitor as a response becomes possible when the system views the knowledge base (KB) as a dynamic rather than a static entity. That is, in addition to answering a user's question on the basis of the information the system currently contains, a system with a dynamic view can offer to monitor for additional relevant information which it will provide to the user if and when it learns of it. Such additional information could be about some possible future event or some previous event about which the system's knowledge is currently incomplete. In the following question-answer pairs, Q-A1 illustrates a monitor for some possible future event. The pair Q-A2 is an example of a monitor for some additional information about a previous event. Responses such as Q-A2 require reasoning that some event, of which knowledge regarding its outcome would enable us to answer the question, has taken place. At some point in the future we will learn of its outcome, when we will then answer.

Q1: Has John registered for CSE110?
A1: No, shall I let you know if he does?

Q2: Did John pass CSE110?
A2: I don't know yet. The semester has ended, but Prof. Tardy has not turned in his grades. Shall I let you know when I find out?

In order to offer monitors as extended responses the system must be competent to offer to monitor for only those events which might possibly occur or, if the system has incomplete knowledge of some event that has occured, only that additional information it may learn of.* This requires some notion of what events are possible or what additional information may be acquired given the current state of the knowledge base. For example, ignorance of the stages through which undergraduates proceed in the university would leave a system attempting to offer monitors unable to discriminate between the following two cases.

Q1: Is John a sophomore?
A1: No, he's a freshman. Shall I let you know when he becomes a sophomore?

Q2: Is Mary a sophomore?
A2: No, she's a junior. Shall I let you know when she becomes a sophomore?

The remainder of this paper is concerned with determining monitor competence with regard to possible future events. We leave open for now the question of competence for those monitors that require reasoning about incomplete knowledge of some previous event.

II REPRESENTATION

Temporal logic [3] is a modal logic for reasoning about the relative possibility of some state to some other state, where the relative possibility is with respect to time or sequences of events. (In contrast to, for example, relative possibility with respect to obligation or belief.) Although one might develop a suitable first order theory to deal with the problems discussed here, it seems worthwhile to study this problem within the framework of temporal logic for reasons of conceptual clarity. Restriction to the propositional setting enables us to concentrate on those issues involved with reasoning about possible change.

We model the evolution of a KB in a propositional temporal logic. The future fragment is a unified branching temporal logic [1] which

----

* This work is partially supported by a grant from the National Science foundation, NSF-MCS 81-07290.

----

* In either case it must be able to identify those future conditions which are relevant to the user's intentions. The discussion here will be limited to determination of monitor competence. See [3] for a brief discussion on relevance.

makes it possible to describe properties on some or all futures. By merging the existential operators with the universal operators, a linear temporal logic is formed for the past fragment (i.e. AXp <-> EXp).

A. Syntax

Formulas are composed from the symbols,

- A set P of atomic propositions.
- Boolean connectives: v, -.
- Temporal operators: AX (every next), EX (some next), AG (every always), EG (some always), AF (every eventually), EF (some eventually), L (immediately past), P (sometime past), H (always past).

using the rules,

- If p e P, then p is a formula
- If p and q are formulas, then (-p), (p v q) are formulas.
- If m is a temporal operator and p is a formula, then (m)p is a formula.

Parenthesis will occasionally be omitted, and &, ->, <-> used as abbreviations.

B. Semantics

A structure T is a triple (S, TT, R) where,

- S is a set of states.
- TT:(S -> 2^P) is an assignment of atomic propositions to states.
- R C (S x S) is an accessibility relation on S. Each state is required to have at least one successor and exactly one predecessor, As (Et (sRt) & E!t (tRs)).

Define an s-branch
b = (..., s(-1), s=s(0), s(1), ...) such that s(i)Rs(i+1).
The satisfaction of a formula p at a node s in a structure T, <T, s> |= p, is defined as follows:

(Note: "e" denotes set inclusion.)

```
<T, s> |= p       iff  peTT(s),
                       for p a proposition
<T, s> |= -p      iff  not <T,s>|=p
<T, s> |= p v q   iff  <T,s>|=p  or  <T,s>|=q
<T, s> |= AGp     iff  Ab At ((teb & t>s) ->
                             <T,t>|=p)
(p is true at every time of every future)
<T, s> |= AFp     iff  Ab Et (teb & t>s &
                             <T,t>|=p)
(p is true at some time of every future)
<T, s> |= AXp     iff  At (sRt -> <T,t>|=p)
(p is true at every immediate future)
<T, s> |= EGp     iff  Eb At ((teb & t>s) ->
                             <T,t>|=p)
(p is true at every time of some future)
<T, s> |= EFp     iff  Eb Et (teb & t>s &
                             <T,t>|=p)
(p is true at some time of some future)
<T, s> |= EXp     iff  Et (sRt & <T,t>|=p)
(p is true at some immediate future)
<T, s> |= Hp      iff  Ab At ((teb & t<s) ->
                             <T,t>|=p)
(p is true at every time of the past)
<T, s> |= Pp      iff  Ab Et (teb & t<s &
                             <T,t>|=p)
(p is true at some time of the past)
<T, s> |= Lp      iff  Et (tRs & <T,t>|=p)
(p is true at the immediate past)
```

A formula p is valid if for every structure T and every node s in T, <T, s> |= p.

C. Axioms

In the following axioms D1-3, A1-4, E1-4 and rules R1-3 form a complete deductive system for the future fragment [1]. Similarly, D4-5, P1-4, R1, R2, R4 are complete for the past fragment. The idea of U1 and U2 is that the relationship between the past and the future may be described locally. Using the induction axioms, we can derive the following theorems, which are a more conventional form (as in [3]):

```
EF(Hp) -> Hp
P(AGp) -> AGp
```

See [1] for a list of many other useful theorems for the future fragment.

```
D1)  AFp <-> -EG-p
D2)  EFp <-> -AG-p
D3)  AXp <-> -EX-p
D4)  Pp <-> -H-p
D5)  Lp <-> -L-p

A1)  AG(p -> q) -> (AGp -> AGq)
A2)  AX(p -> q) -> (AXp -> AXq)
A3)  AGp -> p & AXp & AX(AGp)
A4)  AG(p -> AXp) -> (p -> AGp)

E1)  AG(p -> q) -> (EGp -> EGq)
E2)  EGp -> p & EXp & EX(EGp)
E3)  AGp -> EGp
E4)  AG(p -> EXp) -> (p -> EGp)

P1)  H(p -> q) -> (Hp -> Hq)
P2)  L(p -> q) -> (Lp -> Lq)
P3)  Hp -> p & Lp & L(Hp)
P4)  H(p -> Lp) -> (p -> Hp)

U1)  L(AXp) -> p
U2)  p -> AX(Lp)

R1)  If p is a tautology, then |- p.
R2)  If |- p and |- (p -> q), then |- q.
R3)  If |- p, then |- AGp.
R4)  If |- p, then |- Hp
```

III EXAMPLE

Consider as an example representing that portion of a university KB dealing with students passing and registering for courses. Let the propositional variables Q and R mean "student has passed course" and "student is registered for course", respectively. One might have the following non-logical axioms:

1) (AG)(Q -> (AX)Q) - once a student has passed a course it remains so

2) (AG)((-Q & -R) -> (EX)R) - if a student has not passed a course and is not registered then it is next possible that s/he is registered
3) (AG)(R -> (EX)Q) - if a student is registered for a course then it is next possible that s/he has passed
4) (AG)(R -> (EX)(-Q & -R)) - if a student is registered for a course then it is next possible that s/he has not passed and is not registered
5) (AG)(Q -> -R) - if a student has passed a course s/he is not registered for it
6) (AG)(R -> -Q) - if a student is registered for a course s/he has not passed it (equivalent to 5)

Given the following question,

Is John registered for CSE110?,

there are three possibilities depending on the present state of the KB:

1) John is not registered (-R), but he has passed (Q). If we consider John registering for CSE110 as a possible monitor, it would be ruled out on the basis that it is provable that John cannot register for CSE110. Specifically, from Q and axioms 1 and 5, it is provable that -(EF)R. It would therefore be incompetent to offer to monitor for that condition.

2) John is not registered (-R), but he has not passed (-Q). In this case we could offer to monitor for John registering for CSE110, since (EF)R is derivable from axiom 2.

3) John is registered for (R), hence he has not passed (-Q). One could competently offer to monitor for any of the following:
   a) John no longer registered for CSE110; (EF)-R
   b) John passed CSE110; (EF)Q
   c) John registered for CSE110 again; (EF)(-R & (EX)R)

This last case is interesting in that it can be viewed as a monitor for -R whose action is to set a monitor for R (whose aciton is to inform the user of R). Also, one may wish to include monitors that are responsible for determining whether or not some monitor that is set can still be satisfied. That is, just because something was possible once does not imply that it will always be possible. The user should probably be informed when a situation s/he may still be expecting (because a monitor was offered) can no longer occur. For example, if the system has offered to inform the user if John registers for CSE110, then s/he should be informed if John receives advance placement credit and can no longer register.

The following set of axioms illustrate the use of the past operators. Note that axiom 1 from the above set may be eliminated, due to the ability of the past operators to "look back".

1) (AG)((-(P)Q & -R) -> (EX)R)
2) (AG)(R -> (EX)Q)
3) (AG)(R -> (EX)(-Q & -R))
4) (AG)((P)Q -> -R)
5) (AG)(R -> -(P)Q)

A more important use of the past operators is the ability to express conditions that depend on sequences of events. Thus, expressing the condition that in order for a student to register for a course s/he must not have registered for it twice before (say, because s/he dropped out or failed), requires a formula of the following form: (AG)(-(P)(R & (L)(P)R) & -(P)Q & -R -> (EX)R)

IV CONCLUSION

A simple theorem prover based on the tableau method has been implemented for the propositional branching time temporal logic as described in [1]. Current investigations are aimed towards formulating a quantified temporal logic, as well as the complicated issues involved in increasing efficiency of making deductions. This effort is part of a larger, more general attempt to provide extended responses in question-answering systems [4].

A final comment as to general structure of this enterprise. One could conceivably develop a suitable first order theory to deal with the problems discussed here. It seems worthwhile, however, to study this problem within the framework of temporal logic for reasons of conceptual clarity. Restriction to the propositional setting enables us to deal strictly with those issues involved with reasoning about possible change. Also, we may be able to gain some insight into a reasonable decision procedure.

ACKNOWLEDGEMENT

Scott Weinstein, Aravind Joshi, Bonnie Webber, Sitaram Lanka, and Kathy McCoy have provided valuable discussion and/or comments.

REFERENCES

[1] M. Ben-Ari, Z. Manna, and A. Pneuli, "The Temporal Logic of Branching Time", Eighth Annual ACM Symposium on Principles of Programming Languages, Williamsburg, Va., January 1981.

[2] E. Mays, A. Joshi, and B. Webber, "Taking the Initiative in Natural Language Data Base Interactions: Monitoring as Response", Proceedings of ECAI 82, Orsay, France, July 1982.

[3] N. Rescher and A. Urquhart, Temporal Logic, Springer-Verlag, New York, 1971.

[4] B. Webber, A. Joshi, E. Mays, and K. McKeown, "Extended Natural Language Data Base Interactions", to appear.