A Decision Procedure for A Theory of Actions and Plans

Erik Sandewall

Abstract

The paper describes a decision procedure for a preferential (and therefore non-monotonic), temporal logic for reasoning about actions and plans. The procedure can be used for temporal prediction, temporal postdiction, and also for plan generation. The basic method for the procedure is to operate on a working set of partial interpretations, and to gradually strengthen these partial interpretations until they are minimal models for the given axioms.

This research was supported by the Swedish Board of Technical Development and the IT4 research program.

Appeared in Z. Ras et al. (eds), Methodologies for Intelligent Systems, *Springer Verlag, 1989.*

IDA Technical Report 1989
LiTH-IDA-R-89-39
ISSN-0281-4250

Department of Computer and Information Science,
Linköping University,
S-581 83 Linkping, Sweden

1 Introduction

While "temporal reasoning" and "planning" are important areas within Artificial Intelligence, it is remarkable how little connection there is between the results in those two areas. The logical theories which have been developed in A.I. for reasoning about time and action, are not being used or are not considered usable as the basis of algorithms for "planning" in the A.I. sense of the word. In a coherent theory for A.I. one would expect to have a logic for reasoning about actions and processes over time, a definition of the logical status of planning in the context of that logic (specifying for example whether goals entail plans, or vice versa), and a decision procedure which can perform both temporal prediction, temporal postdiction (= temporal explanation), planning, plan recognition, and other logical operations on action plans. Contemporary planning algorithms, such as Chapman's TWEAK algorithm[Cha87] would then presumably re-appear as special cases of some decision procedure for the logic in question.

The current state of the art in temporal reasoning and in planning does not at all match that ideal. For example, although the TWEAK algorithm is formulated in terms of elementary concepts from logic: data objects such as "propositions" and "preconditions", access operations such as the "modal truth criterion", etc., still the algorithm itself is not identified as implementing an established logical operation such as deduction or abduction in a precisely defined temporal logic.

Most other current planning systems follow the same pattern as STRIPS and TWEAK: they are algorithms whose elementary concepts rely on logic, but where the logical status of the algorithm as a whole remains unspecified. For example Wilkins recently wrote: *"Approaches to reasoning that use unrestricted logics, e.g. general frame axioms... and circumscription..., provide significantly greater expressiveness than SIPE. However, they suffer from inherent computational difficulties, the need to write many axioms with all the details right, and possibly other problems such as unintended models..."*[Wil88].

The GEM concurrency model and GEMPLAN system represent an exception to the general pattern. Using an event based domain model (where state conditions must always be derived from previous events) and temporal logic, the task of the GEMPLAN planner is to construct a plan, *"all of whose executions satisfy a given set of domain constraints and achieve some* *stated set of goals"*[Lan88]. GEM however is based on conventional temporal logic, and was in fact first developed for specification and analysis of concurrency. Therefore GEM, although logic based, does not provide any bridge between A.I. style "reasoning about time and action" and A.I. style planning algorithms, either. Also, due to the richness of its modelling mechanisms, and the built-in implementation considerations, the relationship between the GEMPLAN system and the underlying logic appear to be quite complex.

The present paper describes a decision procedure which in a very straightforward way implements a preferential (i.e. non-monotonic) temporal logic for reasoning about actions and plans, and which subsumes both temporal prediction, temporal postdiction, and planning. The account here must by necessity be brief. Departmental technical reports describe the temporal logic that is being used[San88a], the extensions to the logic for controlling concurrency conditions between simultaneous actions [San88b], and the decision procedure[San88c].

The decision procedure is the main topic for the present paper. Section 2 which follows next, outlines the logic as a background; subsequent sections describe the decision procedure in general form and through examples.

2 The Preferential Entailment Condition in ETL

The logic for reasoning about time and action, and in particular its notion of semantic entailment, provides the foundation for the decision procedure including also the planning algorithm. We use an *Explicit Temporal Logic, ETL*[1]. Section 5 below defines key aspects of formulas and interpretations in ETL, and the evaluation rule for a formula in an interpretation. Here it suffices to say that each interpretation characterizes a possible history of the world from two points of view: on one hand the history contains a sequence of "snapshots" or "states", so that each property has a truth-value at each point in time. On the other hand the history also contains a set of actions, each of which has a starting time, an ending time, and an action type. In this way an interpretation may contain several simultaneous actions. A straightforward syntax for formulas characterizing such interpretations is also introduced.

[1] i.e. a temporal logic where time-points are explicitly referred to in formulas

Logical consequence in ETL is defined using preferential entailment[McC80, Lif85, Sho88]. The following are the basic preferences:

1. Interpretations with a smaller set of property changes are preferred over interpretations with a larger such set;

2. Interpretations with a smaller set of actions are preferred over interpretations with a larger set of actions.

However the first condition is qualified by a distinction between *normal* and *abnormal* property changes. For example in the Yale turkey shot, the axiom characterizing the action type "fire" specifies that a fire action which is performed from time t to time u, *controls* the properties of the gun being loaded and the target being alive, during the time period from t to u if the gun is loaded at time t.

A property change (i.e. an instance of a property changing its truth-value from one time-point to the next) is considered normal if the property is controlled at that time, and abnormal otherwise. The "controls" predicate is minimized by preference, so that in preferred interpretations a property is only controlled at times when it is necessary for it to be so. The preference for a smaller set of property changes is more precisely a preference for a smaller set of *abnormal* property changes, whereas normal property changes do not matter for the preference. Also of course the preference w.r.t. abnormal property changes is applied when the "controls" predicate has already been minimized.

In principle there are three preferences (minimizing the "controls" predicate, the set of abnormal property changes, and the set of actions) but in the formal treatment one only needs two preference relations. If J and J' are two interpretations, we write $J \bullet\!\!< J'$ iff J is preferred over J' on the basis of a smaller set of abnormal changes, or (given the same set of abnormal changes) a smaller set of actions. Also we write $J \ll J'$ iff J is preferred over J' by having a smaller extent for the $Controls$ predicate. (The full definition of the $\ll$ relation is actually more complex because it also handles the concurrency controlling relations).

Preferential entailment is then defined so that if Γ is a set of axioms and α is a proposed conclusion, $\Gamma \models_{\ll,\, \bullet\!<} \alpha$ iff

$$Min(\ \bullet\!\!<\ , Min(\ll, Mod(\Gamma))) \subseteq Mod(\alpha)$$

where $Mod(\Gamma)$ is the set of models of Γ, and $Min(\bullet\!< , S)$ is the subset of S consisting of those members of S which are minimal with respect to the partial order $\bullet\!<$ on interpretations.

The ETL approach to reasoning about actions and plans satisfies the obvious minimal requirements, such as behaving correctly for logical ramification[2] and for the Yale turkey example. It works both for prediction, postdiction and planning, and it has been generalized so that it also handles simultaneous actions correctly. For planning, ETL uses a set of axioms characterizing the world, the initial situation and the goal. These axioms preferentially entail the disjunction of applicable plans.

The ability to reason non-monotonically even about plans with simultaneous actions is a distinguishing feature of ETL. Another distinguishing feature is that the preferential entailment condition is directly reflected in the decision procedure.

3 A decision procedure based on ascent among partial models

In order to implement a preferential entailment condition computationally, one possible technique is offered by circumscription, replacing the model minimization operation by additional axioms. However Hintikka in a recent paper[Hin88] has proposed a different approach namely to incorporate the model minimization into the proof procedure of semantic tableaux. Also Ginsberg[Gin88] has described a model based circumscription theorem prover using ATMS techniques.

For the two-layered preferential entailment that was outlined above, we use algorithms which compute a *complete set of partial models* for the given axioms. Partial interpretations may be defined in general as follows. If a logic with the conventional two truth values has been defined, partial interpretations are structures where each formula has either of the truth-values $\mathcal{T}$, $\mathcal{F}$, or $\mathcal{U}$ for unknown. Also partial interpretations must be generalizations of total ones, so that each total interpretation is a partial interpretation. Furthermore there must be a partial order $\sqsubset$ over the partial interpretations,

[2] e.g. if some axioms are expressed in terms of a property *alive* and others in terms of a property *dead*, with the obvious relationship between those two properties

where $J \sqsubset J'$ if J is less specific than J'. If we write also $\mathcal{U} \sqsubset \mathcal{T}$, $\mathcal{U} \sqsubset \mathcal{F}$, then evaluation of (ordinary) formulas must be monotone with respect to $\sqsubseteq$. Thus if J and J' are two partial interpretations and $J \sqsubseteq J'$, the value of a formula in J is $\sqsubseteq$ its value in J'.

Total interpretations are maximal with respect to $\sqsubseteq$. A *completion* of a partial interpretation is a total interpretation which is $\sqsupseteq$ to it. If a set S of total models is given, then a set $\{J_1, J_2, ...\}$ of partial models is a *covering* of S iff

1) every completion of each J_i, is a member of S;

2) every member of S is a completion of at least one of the J_i.

Let us return now to the decision procedure. Given that Γ is a set of wff, α is one more such formula, and $\ll$ and $\bullet\!<$ are preference relations on interpretations, the task is to determine whether

$$Mod(\Gamma) \subseteq Mod(\alpha)$$

and as a generalization whether

$$Min(\ll, Mod(\Gamma)) \subseteq Mod(\alpha)$$

and similarly for two preference relations. The strategy is now to compute coverings of $Mod(\Gamma)$, $Min(\ll, Mod(\Gamma))$, etc. Then to determine whether α is entailed, it is evaluated in all members of the covering. If it is $\mathcal{T}$ everywhere it is entailed; if it is $\mathcal{F}$ in some member it is not entailed, and if it is $\mathcal{U}$ in some p.i. in the covering then that p.i. must be replaced by several "strengthenings", according to a procedure to be defined below, and the process be repeated.

Both the Mod operation and the preferential minimizations are realized as operations that return coverings. However the two minimizations are realized in different ways. The Mod operation and the $\ll$ minimization are realized using a *partial model ascent* algorithm which will be described next. The $\bullet\!<$ minimization on the other hand is realized using a kind of generate-and-test procedure which will be further described in section 7 below.

The partial model ascent procedure will first be described for the propositional case, ignoring for a while the requirement that interpretations shall characterize processes over time. Assume therefore that partial interpretations are mappings from proposition symbols to $\{\mathcal{T}, \mathcal{F}, \mathcal{U}\}$. The information content relation $\sqsubset$ is defined in the obvious way for such partial interpretations. Evaluation of propositional formulas is also generalized using Kleene's strong definition of truth-tables for three-valued logic[Kle52]. The model set for a formula is the set of partial models where the formula is $\mathcal{T}$, not $\mathcal{U}$ or $\mathcal{F}$. It is easily verified, and well known, that evaluation of formulas is then monotone with respect to $\sqsubseteq$.

The partial model ascent algorithm uses a *working set* of partial interpretations, which is initially chosen as the set of the bottom partial interpretation, which maps all proposition symbols to $\mathcal{U}$. The algorithm then successively "strengthens" the members of the working set. In each step some member of the working set is selected, and is replaced with one or more partial interpretations which are $\sqsupset$ to it, or it is removed without replacement. The strengthening process continues until all the axioms are satisfied (have the value $\mathcal{T}$) in all the partial interpretations in the working set, at which point the working set is a covering.

In order to replace a selected partial interpretation in the working set by one or more "stronger" ones, the selected p.i. is related to one of the axioms in Γ. If the value of the axiom in the selected partial interpretation is $\mathcal{T}$, then the axiom need never more be considered for the forthcoming strengthenings of the same p.i., since it will remain $\mathcal{T}$ due to the monotonicity. If the value of the axiom is $\mathcal{F}$ then the p.i. must be removed from the working set for the similar reason.

Finally if the value is $\mathcal{U}$ it is necessary to replace the p.i. by two others, namely changing the value of a proposition symbol in the p.i. from $\mathcal{U}$ to both $\mathcal{T}$ and $\mathcal{F}$. In principle, both the resulting p.i:s shall be put back into the working set. However by choosing the combination of p.i. and axiom judiciously in each step, one can often arrange that one of the strengthenings is immediately discarded, which means the size of the working set does not change. For example if the selected p.i. maps a and b to $\mathcal{T}$, and c to $\mathcal{U}$, and one of the current axioms is

$$a \wedge b \rightarrow c$$

then the value of that axiom is $\mathcal{U}$ in the current p.i., and $\mathcal{T}$ and $\mathcal{F}$ in the two strengthenings obtained by changing c to $\mathcal{T}$ or to $\mathcal{F}$. Consequently the selected p.i. is replaced in the working set by the former strengthening.

The strenghtening process is continued until all axioms are $\mathcal{T}$ in all the p.i:s in the working set. One has then obtained a representation of the model set, $Mod(\Gamma)$, where one partial interpretation represents a number of total interpretations. The compactness of the representation of the model set i.e.

the cardinality of the covering, depends on a number of factors including also the strategies which have been used for the choices during the partial model ascent process.

Partial model ascent may be viewed as a kind of constraint propagation, and it is apparently quite similar to the process that is performed in simple rule-based expert systems. The view of a data base as a partial model (rather than as a set of propositions) is sometimes defended not only for expert systems but also for data bases.

If the result of the partial model ascent algorithm is used for determining whether a proposed conclusion α follows from the given axioms, and α has the value $\mathcal{U}$ in some of the resulting partial interpretations (and does not have the value $\mathcal{F}$ in any of them), it is necessary to further strengthen those p.i:s in order to determine whether α follows.

In order to use partial model ascent for temporal reasoning, one must find a suitable way of generalizing total, temporal interpretations to partial ones. The solution to this problem will be described in section 5 below.

4 Partial model ascent with a preference relation

Due to the special properties of the $\ll$ relation, it is straight-forward to compute a covering of $Min(\ll, Mod(\Gamma))$ from a covering of $Mod(\Gamma)$. The partial model ascent algorithm can therefore be extended so that it also handles the $\ll$ preference relation.

The method is basically to continue the process of successive strengthening of the members of the working set, using also the preference relation. First of all the partial order $\ll$ on total interpretations is re-considered as a mapping from pairs of interpretations, to truth-values $\mathcal{T}$ and $\mathcal{F}$. It is then generalized to a mapping from pairs of partial interpretations, to partial truth-values $\mathcal{T}$, $\mathcal{F}$ or $\mathcal{U}$, where the mapping must be monotone in both arguments. In other words if $J \ll J'$ is $\mathcal{T}$ then every completion of J must be $\ll$ every completion of J'.

The partial model ascent algorithm then proceeds as follows. The covering of $Mod(\Gamma)$ is taken as the initial working set. In each step, two members of the working set are selected and compared using the generalized $\ll$

relation. If they are related with truth-value $\mathcal{T}$ then the less preferred ($\ll$-larger) partial model is removed from the working set. If they are related with truth-value $\mathcal{U}$ then one of them is replaced in the working set by a pair of strengthenings. The algorithm terminates when all pairs of members of the working set are related with the truth-value $\mathcal{F}$.

Thus the generalized partial model ascent algorithm consists of two major phases, each of which contains a number of successive steps where the working set is strengthened by applying an axiom or by applying the preference relation. These two phases do not in general commute or interleave, i.e. one can not in general apply the preference relation before all the axioms have been satisfied. If the preference relation is applied before all axioms are satisfied, it may happen that the preferred partial interpretations (or all their strengthenings) are later deleted from the working set because they can not satisfy all the axioms. In such a case the previously dispreferred partial interpretation should have been retained because some of its completions are of relevance after all. In computational terms this means that if the application of the preference relation is interleaved with axiom-based strengthening of the members of the working set, *backtracking* must be available. This is of course also what one does in practical rule based systems. We shall see how the same situation appears for the case of temporal reasoning.

5 Partial temporal interpretations

In order to apply the method of partial model ascent to the case of temporal reasoning, the structure of both total and partial, temporal interpretations must be defined. The following definitions are used. An ordinary, total interpretation is a tuple

$$\langle \mathbf{T}, tp, R, P, M_T, M_A, X, Y, Z \rangle$$

where $\mathbf{T}$ is a nonempty, enumerable set of time-points, $tp \subset \mathbf{T} \times \mathbf{T}$ is a strict total order representing temporal precedence, $R \subseteq \mathbf{T} \times C$, $P \subseteq \mathbf{T} \times H \times \mathbf{T}$, $X \subseteq \mathbf{T} \times C$, and $Y, Z \subseteq P \times C$. Also M_T and M_A define the values of constant symbols for time-points and actions. Here C is the set of symbols for properties, such as "gun is loaded" or "fred is alive", R is a mapping which specifies whether at some point in time some property holds, and X is the relation which specifies the "controls" predicate i.e. whether the property "may change" from a given time-point to its successor.

Furthermore H is the set of symbols for action types, such as "firing the gun", and P is the plan, i.e. the set of tuples each of which specifies that an action of a certain type is performed from a certain starting time to a certain ending time. The relations Y and Z express that a certain action (member of P) influences or is dependent on a certain property, information which is used in axioms which do concurrency control. (The relations Y and Z are minimized along with X by the preference relation $\ll$).

It is straight-forward to define formulas and their evaluation over such interpretations. For example the rule which specifies the effect of firing the gun, in the Yale Turkey Shot, goes as follows (without concurrency control conditions):

$$
\begin{aligned}
&Holds(a, t, fire, u) \rightarrow \\
&\quad Notholds(t, gunloaded, u) \vee \\
&\quad [Holds(t, gunloaded) \wedge Notholds(t, fredalive, u) \wedge \\
&\quad\quad \neg Holds(u, gunloaded) \wedge Controls(a, gunloaded)] \vee \\
&\quad [Holds(t, gunloaded) \wedge Holds(t, fredalive) \wedge \\
&\quad\quad \neg Holds(u, gunloaded) \wedge \neg Holds(u, fredalive) \wedge \\
&\quad\quad Controls(a, gunloaded) \wedge Controls(a, fredalive)]
\end{aligned}
$$

Here the symbols *gunloaded* and $fredalive$ are properties and therefore members of C, and $fire$ is an action type and a member of H. $Holds(t, c)$ is interpreted in the obvious way using the relation R in the interpretation; the argument t is evaluated and the argument c is not. $Holds(t, c, u)$ is interpreted as $\mathcal{T}$ iff c is assigned the value $\mathcal{T}$ according to R at all times between (the value of) t and (the value of) u inclusive, and $Notholds(t, c, u)$ is interpreted as $\mathcal{T}$ iff c is assigned the value $\mathcal{F}$ at all times during the same interval. $Holds(a, t, fire, u)$ assumes a to evaluate to a member of P, and is interpreted as $\mathcal{T}$ iff a^* is a triple $\langle t^*, fire, u^* \rangle$ where t^* is the value of t, and similarly for u^* and a^*. The predicate $Controls$ is interpreted similarly to $Holds$ but using the X component instead of the R component of the interpretation.

Total interpretations are therefore defined using linear time rather than branching time, only because linear time is so ontologically obvious. However when it comes to partial interpretations, a more complex notion of time must be introduced, in order for the partial model ascent algorithm to work properly without an undue combinatorial explosion of possible time-points.

The definition of partial temporal interpretations therefore is as a tuple

$$\langle \mathbf{T}^{+}, tp, R, P, M_T, M_A, X, Y, Z \rangle$$

as before, but with the following changes. The time domain has been extended so that it contains both *time points* and *time tokens*. The time points are totally ordered like before, but time tokens are objects which have not been finally identified with a particular time-point. Time points and time tokens are together called *times*.

The total time domain consisting of both time points and time tokens is only partially ordered. The idea is that a partial interpretation may map a time-point constant in a formula to a time token, and thereby leave the exact value of the constant on the time axis open until additional information becomes available.

Also the ordering tp can not simply be a partial order over the extended time domain, since one must distinguish the case where two time-tokens (or a time-point and a time-token) are known to be in sequence and not equal, from the case where they are known to be in sequence but possibly equal. Therefore the component tp of the partial interpretation is defined as a mapping from $\mathbf{T}^{+} \times \mathbf{T}^{+}$ to a set of indicators for possible orderings, written as $\prec$, $\preceq$, $\succ$, $\simeq$, and so on. The information content ordering $\sqsubset$ for these indicators is defined as in figure 1, and the derived definition of $\sqsubseteq$ over the domain of mappings tp is immediately obtained.

A given "relation" tp divides the domain $\mathbf{T}^{+}$ into equivalence sets of times that are known to be $\simeq$, and also defines a relation $[\preceq]$ over such equivalence sets in the obvious way. That relation can therefore be understood as the set of all recognized time intervals.

In the partial interpretation, R is defined as a mapping

$$[\preceq] \times C \rightarrow \{\mathcal{T}, \mathcal{F}, \mathcal{U}\}$$

i.e. for every recognized time interval and every property in C, it is specified whether the property holds over that (whole) interval, or does not hold over any part of the interval, or something else is the case. The mapping X is defined similarly.

With these definitions it is straight-forward to define the strength relation $\sqsubset$ for partial interpretations, and to define the evaluation of formulas in partial interpretations so that the evaluation is monotone with respect to $\sqsubset$ like in the propositional case. Also the preference relation $\ll$ extends in straight-forward ways to partial interpretations. Therefore the partial model ascent algorithm, and its extension to the case of $\ll$-preference, can be immediately applied. The details are found in [San88c].

6 An Example

The method of ascent through partial temporal models will be illustrated using the standard example, i.e. the Yale turkey shooting[3]. We shall show how the method applies both for temporal prediction (given the "load" and "shoot" actions, predict that the turkey will be dead), and for temporal planning (given that the turkey is initially alive and later dead, identify the sequence of actions "load" - "shoot").

For the examples it is convenient to have a graphical representation for partial interpretations as illustrated in figure 2. The set $\mathbf{T}^+$ of generalized time points consists of t_1, t_2, t_3 and t_4 listed at the top of the figure. The temporal order tp between the time-points is indicated by broken arrows below. The set P of actions is indicated next below, with each action $\langle t, h, u \rangle$ as an arrow from the starting-time t to the ending-time u, and the action type h written over the arrow.

Finally there is a matrix, which for each property symbol c in C, and for each time in $\mathbf{T}^+$ specifies the truth-value $\mathcal{T}$, $\mathcal{F}$, or $\mathcal{U}$ which the partial interpretation assigns to c at the time. Over each interval between two successive time-points the truth-value for c is characterized by either of the following cases:

1. the property c is "controlled" by an action, i.e. the value of c at the beginning, the end, and during the interval have been chosen according

[3]It is true that this example suffers from over-use, but on the other hand it is convenient to use an example which everybody knows and which therefore does not need to be explained

to an axiom characterizing an action over the same interval. In figure 2 the action labelled "load" in the P section of the figure, corresponds to the transition of the property *gunloaded* from $\mathcal{F}$ to $\mathcal{T}$. This case is indicated with a solid arrow in the matrix.

2. the property c is unchanged by default. By preferential entailment the logic minimizes the extent of "control" of properties, so if no action implies that a property is controlled over any part of an interval, then those interpretations are preferred where the control relation X is false over the interval. Such intervals where $Controls$ has the value $\mathcal{F}$, meaning that the property should persist by default, are indicated with a broken arrow in the matrix.

Some additional cases arise due to the mechanisms for concurrency control of simultaneous actions, but they will not be needed for the examples here. With these preliminaries we are ready to consider the examples of reasoning by ascent through partial temporal models.

6.1 Example of temporal prediction

Let the following axioms be given for characterizing the scenario:

$$Holds(t_1, load, t_2)$$

$$t_2 \preceq t_3$$

$$Holds(t_3, fire, t_4)$$

$$\neg Holds(t_1, gunloaded)$$

$$Holds(t_1, fredalive)$$

In addition we need the axiom in section 5 above which characterizes the *load* action type, and a similar axiom for the $fire$ action type namely:

$$Holds(a, t, load, u) \;\Rightarrow\; Holds(t, gunloaded, u) \vee$$

$$\neg Holds(t, gunloaded) \wedge Holds(u, gunloaded) \wedge Controls(a, gunloaded)$$

Finally we need some fairly trivial general-purpose axioms such as

$$Holds(t, h, u) \Rightarrow t \prec u$$

The partial interpretation in figure 3a satisfies the scenario axioms but not the axioms for the two action types. Figures 3b, 3c, etc. differ from figure 3a only with respect to the matrix, i.e. the table characterizing the truth-values for pairs (t, c) and the $Controls$ relation. In order to also satisfy the axiom for *load*, the partial interpretation must be strengthened to the one shown in figured 3b. (The axiom has the form $\phi \Rightarrow \psi \vee \theta$ where ϕ is $\mathcal{T}$ in the partial interpretation 3a, ψ is $\mathcal{F}$, and θ is $\mathcal{U}$, so the only way of satisfying the axiom is to make θ become $\mathcal{T}$).

Next it is convenient and natural, from a common-sense computational point of view, to impose the defaults shown in figures 3c and 3d, to the effects that $fredalive$ will persist until time t_2, and both *gunloaded* and $fredalive$ will persist until time t_3. The partial interpretation 3d satisfies the same axioms as 3b, but still does not satisfy the axiom for $fire$. It can be strengthened in only one way, as in figure 3e, to a model for all the axioms.

The chain of successive strengthenings from 3a to 3e may be the natural way of reasoning about the problem, but it is guilty of the error (or at least shortcut) which was discussed in section 4 above i.e. interleaving of axiom application and preferencing. Without interleaving, we instead proceed from a working set consisting only of p.i. 3b, to a working set consisting of all partial models for all the axioms, consisting of the partial models in figures 4a, 4b and 4c. The next step is to minimize the extent of the $Controls$ predicate, as shown in figures 4d, 4e and 4f, and finally to minimize the set of property changes during intervals where a property is uncontrolled whereby 4f (with an empty set of such changes) is preferred over 4d and 4e. (Actually the remaining $\mathcal{U}$'s in 4d, 4e and 4f are also strengthened in the obvious way in order to minimize property changes).

Regardless of whether interleaving is used or not, the algorithm concludes with a single partial model which satisfies the formula

$$\neg Holds(t_4, fredalive)$$

which is therefore preferentially entailed by the given axioms.

It is clear from the example that interleaving of axiom application and preferencing is necessary in practice, in order to prune many useless branches of the search as early as possible. One can also see that in some practical situations interleaving is in fact safe and can not lead to backtracking. The full paper describes a systematic method for safe interleaving.

The algorithm is symmetric with respect to time, so exactly the same method works for temporal postdiction, also called temporal explanation.

6.2 Example of plan generation

Let us now revise the example so that the initial situation and the goal situation are known, but the actions are not known. The axioms are therefore

$$t_1 \preceq t_4$$

$$Holds(t_1, fredalive)$$

$$\neg Holds(t_4, fredalive)$$

$$\neg Holds(t_1, gunloaded)$$

together with the same axioms as before characterizing the available repertoire of action types. The reasoning process should produce a covering for the axioms where the following formulas are satisfied for suitable choices of the temporal constants t_2 and t_3:

$$Holds(t_1, load, t_2)$$

$$t_2 \preceq t_3$$

$$Holds(t_3, fire, t_4)$$

Actually there is also a more general partial, preferred model for the given axioms, namely one where there are waiting periods between t_1 and the beginning of the loading action, and similarly between the end of the firing action and t_4, but this is unimportant for the example.

The decision procedure works as follows in this case. Figure 5a shows the partial interpretation satisfying the scenario axioms with a minimal set P of actions, namely the set of actions in the interpretation is empty. Therefore the p.i. in 5a also satisfies the action axioms. Figure 5b shows the same p.i. after the relation $Controls$ has been minimized, i.e. the extent of the broken arrows has been maximized within the constraints imposed by the axioms. The property $fredalive$ changes its value abnormally in the p.i. 5b.

The key idea with the outer preference relation $\bullet\!\!<$ which was introduced in section 2, is now that it *both* minimizes the set P of actions, *and* the set of abnormal property changes. However if there is a trade-off so that a larger set of actions will "buy" a smaller set of abnormal property changes, the preference relation will prefer such a change. In the present case the algorithm will therefore extend P in search of a model with a smaller (viz. empty) set of abnormal property changes.

Figure 5c shows the effect of modifying 5a with one more action namely a $fire$ action, and then minimizing $Controls$. Two of the three alternatives are shown in the figure; alternative 1 is the only one of interest. The minimization of $Controls$ used in figure 5b becomes irrelevant after the set of actions has been changed). The abnormal change of $fredalive$ has been removed, but there is now an abnormal change of *gunloaded.* Finally figure 5d shows the effect of also adding a *load* action and again recomputing the minimal extent of the $Controls$ relation. Again there are several alternatives, but only one is shown in the figure. The partial model illustrated in 5d clearly has all preferred models of the given axioms as its extensions.

7 Minimization of $<$

We have now defined a partial model ascent algorithm, which for a given set Γ of axioms and a given preference relation $\ll$ will compute a set of partial models whose extensions together form

$$Min(\ll, Mod(\Gamma))$$

Also we are using an explicit temporal logic where preferential entailment is defined so that $\Gamma \models_{\ll,\,\bullet\!<} \alpha$ iff

$$Min(\bullet\!\!<\,, Min(\ll, Mod(\Gamma))) \subseteq Mod(\alpha)$$

We have seen already that the various temporal reasoning operations can be readily defined once an adequate set of preferred partial models have been computed. As the example has illustrated, temporal prediction and temporal postdiction can be seen as the problem of deciding whether a given set Γ of axioms preferentially entails a given α. Also temporal planning can be seen as the problem of finding a suitable plan α which is preferentially entailed from a given Γ; suitable in the sense of syntactic constraints: α should characterize a set of actions, temporal relations between them, and conditions on actions in the case of conditional plans.

It therefore only remains to extend the partial model ascent algorithm, which implements the Mod operation and the minimization with respect to $\ll$, so that one obtains a set of partial models whose extensions together are exactly the members of

$$Min(\bullet\!\!<\ , Min(\ll, Mod(\Gamma)))$$

The example of plan generation above has already suggested how the minimization with respect to $<$ is realized. It is basically a generate-and-test procedure, which in the simplest case searches the space of possible action sets P.

It works as follows. First assume that a model can be found which does not contain any abnormal property changes. Search the space of possible plans P (each plan being a set of actions) starting with the empty plan, and gradually extending with more and more actions. For each plan P use the partial model ascent method to identify, among all interpretations which contain the plan P, those which are preferred models (according to the preference relation $\ll$) for the given axioms.

If the set of $\ll$-preferred models for a chosen P is empty, then the search must continue by extending P. Similarly if the $\ll$-preferred models contain abnormal changes the search must continue by extending P. The search over the various ways of extending P is a tree search. If one is only interested in one plan, it is sufficient to search until one node in the tree yields a partial model without abnormal changes. On the other hand if one is interested in the set of all possible plans, or if one is interested in determining whether a proposed conclusion α follows from the given axioms, then the search must in principle proceed until the tree has been truncated in all directions by finding a node yielding a partial model without abnormal changes.

Finally it may happen that certain abnormal property changes can never be eliminated, for example if the scenario for a planning problem specifies several changes some of which can not be achieved by any of the available actions. In such a case one may still be interested in a plan which deals adequately with the other changes. The full algorithm specifies that one should then relax the set of "admitted abnormal property changes", which was initially chosen as the empty set, and search the tree of possible plans P for partial models which only have the admitted abnormal property change(s).

This is the algorithm in principle. In practice it is necessary to add heuristics for constraining which extensions of P are being considered. In the absence of any such heuristics, the algorithm will even search the infinite space of all possible cases of first adding some useless actions to P, and then adding the goal-relevant actions afterwards.

8 Discussion

The procedure described above combines several well established types of A.I. algorithms into a coherent decision procedure for the ETL logic and its semantic entailment condition. In particular, the search implementing the $\bullet\!<$ minimization is in essence a planning algorithm, and when search control strategies have been incorporated it should become similar to Chapman's TWEAK algorithm [Cha87].

Similarly the partial model ascent algorithm, which is used here both for identifying in concise form the models for the given axioms, and for minimizing with respect to the $\ll$ preference, can be seen as a constraint propagation algorithm with the axioms in Γ as the constraint rules.

Management of, and access to the temporal precedence relation tp in the partial interpretations is a non-trivial task. Current methods and algorithms for management of temporal data bases, as in [All84, DM87] are applicable for this purpose.

TMS systems[Doy79] were developed as computationally tractable algorithms for non-monotonic reasoning, and Ginsberg's circumscription theorem prover[Gin88] is based on an ATMS[dK86]. One may therefore ask how the partial model ascent algorithm being used here, compares to TMS or ATMS systems. The simple algorithm without interleaving has no need to keep track of justifications, and therefore has no obvious relation to TMS's or ATMS's. Since backtracking is needed for supporting "unsafe" interleaving of axiom application and preference application in the partial model ascent algorithm, it would be interesting to determine whether TMS or ATMS would be appropriate computational vehicles for partial model ascent with backtracking,

Work on an implementation of the decision procedure described here is presently in progress.

References

[All84] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[Cha87] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–377, 1987.

[dK86] Johan de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28:127–162, 1986.

[DM87] Thomas Dean and Drew V. McDermott. Temporal data base management. *Artificial Intelligence*, 32, 1987.

[Doy79] Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[Gin88] Matthew L. Ginsberg. A circumscriptive theorem prover. In Michael Reinfrank et al., editor, *Non-Monotonic Reasoning*, pages 100–114. Springer Verlag, 1988.

[Hin88] Jaakko Hintikka. Model minimization - an alternative to circumscription. *Journal of Automated Reasoning*, 4:1–13, 1988.

[Kle52] S. Kleene. *Introduction to Metamathematics.* Van Nostrand, 1952.

[Lan88] Amy L. Lansky. Localized event-based reasoning for multi-agent domains. *Computational Intelligence*, 4:319–340, 1988.

[Lif85] Vladimir Lifschitz. Computing circumscription. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 121–127. Morgan Kaufmann, 1985.

[McC80] John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[San88a] Erik Sandewall. Non-monotonic entailment for reasoning about time and action, part I: Sequential actions. Technical Report LiTH-IDA-R-88-27 and LAIC-IDA-88-TR13, IDA, 1988.

[San88b] Erik Sandewall. Non-monotonic entailment for reasoning about time and action, part II: Concurrent actions. Technical Report LiTH-IDA-R-88-28 and LAIC-IDA-88-TR14, IDA, 1988.

[San88c] Erik Sandewall. Non-monotonic entailment for reasoning about time and action, part III: decision procedure. Technical Report LiTH-IDA-R-88-29 and LAIC-IDA-88-TR15, IDA, 1988.

[Sho88] Yoav Shoham. *Reasoning about Change.* MIT Press, 1988.

[Wil88] David E. Wilkins. Causal reasoning in planning. *Computational Intelligence*, 4:373–380, 1988.

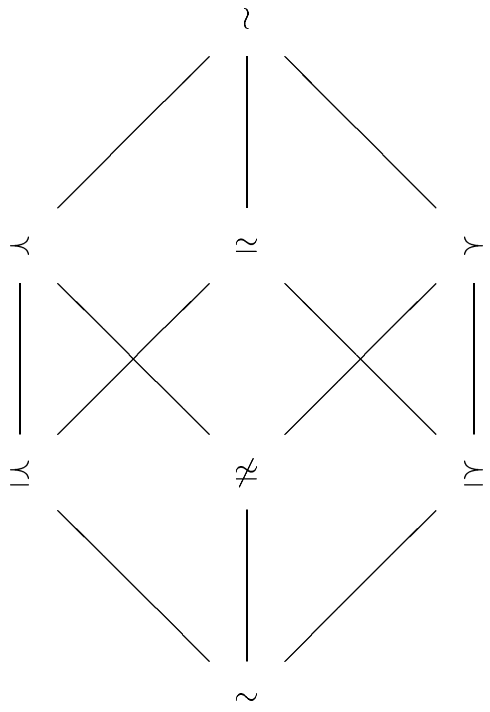
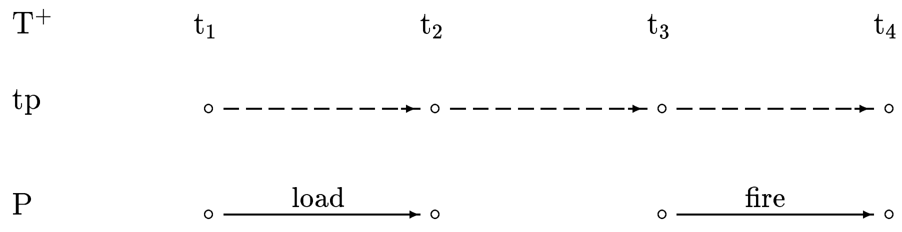


Figure 1



$$\begin{tikzcd}[row sep=small]
\text{R,X,Y,Z} \\
\text{gunloaded} & \mathrm{F} \arrow[r, "\text{load}"] & \mathrm{T} \arrow[r, dashed] & \mathrm{T} \arrow[r, "\text{fire}"] & \mathrm{F} \\
\text{fredalive} & \mathrm{T} \arrow[r, dashed] & \mathrm{T} \arrow[r, dashed] & \mathrm{T} \arrow[r, "\text{fire}"] & \mathrm{F}
\end{tikzcd}$$

Figure 2

T^+	t_1		t_2		t_3		t_4
tp	$\circ$	$\dashrightarrow$	$\circ$	$\dashrightarrow$	$\circ$	$\dashrightarrow$	$\circ$
P	$\circ$	$\xrightarrow{\text{load}}$	$\circ$		$\circ$	$\xrightarrow{\text{fire}}$	$\circ$
R,X,Y,Z (3a)							
gunloaded	F		U		U		U
fredalive	T		U		U		U
R,X,Y,Z (3b)							
gunloaded	F	$\xrightarrow{\text{load}}$	T		U		U
fredalive	T		U		U		U
R,X,Y,Z (3c)							
gunloaded	F	$\xrightarrow{\text{load}}$	T		U		U
fredalive	T	$\dashrightarrow$	T		U		U
R,X,Y,Z (3d)							
gunloaded	F	$\xrightarrow{\text{load}}$	T	$\dashrightarrow$	T		U
fredalive	T	$\dashrightarrow$	T	$\dashrightarrow$	T		U
R,X,Y,Z (3e)							
gunloaded	F	$\xrightarrow{\text{load}}$	T	$\dashrightarrow$	T	$\xrightarrow{\text{fire}}$	F
fredalive	T	$\dashrightarrow$	T	$\dashrightarrow$	T	$\xrightarrow{\text{fire}}$	F

Figure 3

T^+	t_1		t_2		t_3		t_4
tp	$\circ$	$\dashrightarrow$	$\circ$	$\dashrightarrow$	$\circ$	$\dashrightarrow$	$\circ$
P	$\circ$	$\xrightarrow{\text{load}}$	$\circ$		$\circ$	$\xrightarrow{\text{fire}}$	$\circ$
R,X,Y,Z (4a)							
gunloaded	F	$\xrightarrow{\text{load}}$	T		F	$\xrightarrow{\text{fire}}$	F
fredalive	T		U		U		U
R,X,Y,Z (4b)							
gunloaded	F	$\xrightarrow{\text{load}}$	T		T	$\xrightarrow{\text{fire}}$	F
fredalive	T		U		F		F
R,X,Y,Z (4c)							
gunloaded	F	$\xrightarrow{\text{load}}$	T		T	$\xrightarrow{\text{fire}}$	F
fredalive	T		U		T	$\xrightarrow{\text{fire}}$	F
R,X,Y,Z (4d)							
gunloaded	F	$\xrightarrow{\text{load}}$	T	$\dashrightarrow$	F	$\xrightarrow{\text{fire}}$	F
fredalive	T	$\dashrightarrow$	U	$\dashrightarrow$	U	$\dashrightarrow$	U
R,X,Y,Z (4e)							
gunloaded	F	$\xrightarrow{\text{load}}$	T	$\dashrightarrow$	T	$\xrightarrow{\text{fire}}$	F
fredalive	T	$\dashrightarrow$	U	$\dashrightarrow$	F	$\dashrightarrow$	F

R,X,Y,Z (4f)

gunloaded	$F \xrightarrow{\text{load}} T \dashrightarrow T \xrightarrow{\text{fire}} F$
fredalive	$T \dashrightarrow U \dashrightarrow T \xrightarrow{\text{fire}} F$

Figure 4

T^+	t_1		t_2		t_3		t_4
tp	$\circ$	$\dashrightarrow$	$\circ$	$\dashrightarrow$	$\circ$	$\dashrightarrow$	$\circ$
P (5a-b)	$\circ$		$\circ$		$\circ$		$\circ$
R,X,Y,Z (5a)							
gunloaded	F						U
fredalive	T						F
R,X,Y,Z (5b)							
gunloaded	F	$\dashrightarrow$					U
fredalive	T	$\dashrightarrow$					F
P (5c)	$\circ$		$\circ$		$\circ$	$\xrightarrow{\text{fire}}$	$\circ$
R,X,Y,Z (5c alt. 1)							
gunloaded	F	$\dashrightarrow$			T	$\xrightarrow{\text{fire}}$	F
fredalive	T	$\dashrightarrow$			T	$\xrightarrow{\text{fire}}$	F
R,X,Y,Z (5c alt. 2)							
gunloaded	F	$\dashrightarrow$			T	$\xrightarrow{\text{fire}}$	F
fredalive	T	$\dashrightarrow$			F	$\dashrightarrow$	F
P (5d)	$\circ$	$\xrightarrow{\text{load}}$	$\circ$		$\circ$	$\xrightarrow{\text{fire}}$	$\circ$

R,X,Y,Z (5d alt. 1)

$$
\begin{array}{ll}
\text{gunloaded} & F \xrightarrow{\quad\text{load}\quad} T \dashrightarrow T \xrightarrow{\quad\text{fire}\quad} F \\
\text{fredalive} & T \dashrightarrow T \dashrightarrow T \xrightarrow{\quad\text{fire}\quad} F
\end{array}
$$

Figure 5