LiTH-IDA-R-88-27

ISSN-0281-4250

INSTITUTIONEN FÖR DATAVETENSKAP
UNIVERSITETET OCH TEKNISKA HÖGSKOLAN
LINKÖPING

Non-Monotonic Entailment for Reasoning about Time and Action Part I: Sequential Actions

by

Erik Sandewall

RESEARCH REPORT
RKLLAB, September 1988

Postadress:
Institutionen för datavetenskap
Universitetet i Linköping och
Tekniska Högskolan
581 83 Linköping

Mailing address:
Department of Computer and
Information Science
Linköping University
S-581 83 Linköping, Sweden

The Department of Computer and Information Science
Linköping University

PhD theses:

* Available at: University of Microfilms Intl., 300 N. Zeeb Road, Ann Arbor, MI 48106, USA.

(*Linköping Studies in Science and Technology. Dissertations.*)

*No 77	**Östen Oskarsson:** Mechanisms of Modifiability in large Software Systems, 1982.
No 94	**Hans Lunell:** Code Generator Writing Systems, 1983.
No 97	**Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983.
*No 109	**Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984.
*No 111	**Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984.
*No 155	**Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons. 1987.
No 165	**James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning. 1987.
No 170	**Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems. 1987.

Licentiate of engineering Theses:

(*Linköping Studies in Science and Technology. Theses.*)

No 48	**Henrik Nordin:** On the Use of Typical Cases for Knowledge-Based Consultation and Teaching. 1985.
No 52	**Zebo Peng:** Steps Towards the Formalization of Designing VLSI Systems. 1985.
No 60	**Johan Fagerström:** Simulation and Evaluation of an Architecture based on Asynchronous Processes. 1985.
No 71	**Jalal Maleki:** ICONStraint, A Dependency Directed Constraint Maintenance System. 1987.
No 72	**Tony Larsson:** On the Specification and Verification of VLSI Systems. 1986.
No 73	**Ola Strömfors:** A Structure Editor for Documents and Programs. 1986.
No 74	**Christos Levcopoulos:** New Results about the Approximation Behaviour of the Greedy Triangulation. 1986.
No 104	**Shamsul I. Chowdhury:** Statistical Expert Systems - a Special Application Area for Knowledge-Based Computer Methodology. 1987.
No 108	**Rober Bilos:** Incremental Scanning and Token-based Editing. 1987
No 111	**Hans Block:** SPORT-SORT Sorting Algorithms and Sport Tournaments. 1987.
No 113	**Ralph Rönnquist:** Network and Lattice Based Approaches to the Representation of Knowledge. 1987.
No 118	**Mariam Kamkar, Nahid Shahmehri:** Affect-Chaining in Program Flow Analysis Applied to Queries of Programs. 1987
No 126	**Dan Strömberg:** Transfer and Distribution of Application Programs. 1987
No 127	**Kristian Sandahl:** Case Studies in Knowledge Acquisition, Migration and User Acceptance of Expert Systems. 1987
No 139	**Christer Bäckström:** Reasoning about Interdependent Actions. 1988.
No 140	**Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing. 1988.

LiTH-IDA-R-88-27 ISSN-0281-4250

Non-Monotonic Entailment for Reasoning about Time and Action Part I: Sequential Actions

by

Erik Sandewall
E-mail: ejs@ida.liu.se

Completed while visiting at the Computer Science Department,
Stanford University

Abstract. The paper difines a logic for reasoning about time and action, in terms of syntax, semantics, and preference relations on interpretations. Temporal logic with explicit representation of time-points is used, rather than situation calculus. The preference relations prefer (1) interpretations which maximize the expectations that fluents do not change, (2) interpreta- tions which minimize the number of instances where fluents change in spite of expectations to the opposite, and (3) interpretations which minimize the set of actions. The preferential entailment relation based on these preferen- ces is equally applicable to temporal projection, temporal explanation, and plan construction (as used for planning, story understanding, and diagnosis). A number of examples are described supporting the thesis that this semantics obtains the results dictated by common sense.

This research was supported by the Swedish Board for Technical Development, Rank Xerox AB, and the Rockwell Corporation.

Postadress:
Institutionen för datavetenskap
Universitetet i Linköping och
Tekniska Högskolan
581 83 Linköping

Mailing address:
Department of Computer and
Information Science
Linköping University
S-581 83 Linköping, Sweden

1. Introduction and Topic.

The present paper proposes an improved preference relation between interpretations, for use in defining semantic entailment for non-monotonic reasoning about time and action. The work is based on an explicit temporal logic rather than situation calculus.

By way of introduction, we first discuss the choice between explicit temporal logic and situation calculus, and then discuss the choice of the preference relation for reasoning about time and action.

EXPLICIT TEMPORAL LOGIC VS. SITUATION CALCULUS. By an *explicit temporal* logic we mean a multi-sorted, first-order logic where time-points is one of the sorts, and where one can make statements such as

$$Holds(t_1, loadgun, t_2).$$

meaning that during the time period from t_1 to t_2, the action of loading the gun takes place. We contrast such explicit(ly) temporal logic with situation calculus, where one typically uses terms such as

$$Result(wait, Result(loadgun, s))$$

for the situation which arises if one starts in situation s, loads the gun, and then waits. The crucial aspect of the situation calculus approach is of course not the use of situations as such, but the functional assumption that the state of the world after an action has been performed, is a function of the state of the world before the action.

We use the term "explicit" temporal logic to distinguish from modal temporal logics.

In the A.I. literature on reasoning about actions and their effects, explicit temporal approaches have been used by Shoham [Sho88], before him by Allen [All84] and D. McDermott [McD82], and also recently by e.g. Reichgelt [Rei87]. In addition there is a considerable literature on reasoning about time intervals as such, without addressing actions and the effects of actions.

The situation calculus approach was introduced by McCarthy [MH69], and has since been used by a large number of authors, including notably Lifschitz [Lif87] and Manna and Waldinger [MW87]. In particular, most published results on non-monotonic temporal reasoning, proof-theoretic or model-theoretic, have used the situation calculus.

There are several good reasons for preferring the explicit temporal approach. To begin with, it is better suited for reasoning about actions that occur in parallel. Although it is in principle possible to introduce parallel actions in the situation calculus as well, one would then have to use an operation *par* so that e.g.

$$Result(par(loadgun, sneeze), s)$$

is the result of loading the gun and sneezing at the same time. This representation is not entirely general, since there are action structures which can not

be expressed using only *par* together with the sequencing of actions which is obtained by nested use of *Result*. However to the best of my knowledge this approach has not been tried. In the explicit temporal approach, on the other hand, one can express statements about parallel actions without any additional constructs, for example

$$Holds(t_1, loadgun, t_2) \wedge Holds(t_1, sneeze, t_2).$$

Another advantage with the explicit temporal approach is obtained in plan construction types of reasoning, where static facts about the world at two or more time-points are given, and it remains to identify which actions must be performed, or have been performed, in order to obtain the given changes in the world state. Plan construction is used not only for planning (in the A.I. sense of the word), where the current state and desired goal state of the world are given, and one has to identify a sequence (or structure) of actions which can be expected to achieve the goals; it is also used for story understanding and for diagnostics of dynamic systems ("figuring out what has happened").

In the explicitly temporal approach, plan construction is performed by forward deduction from the given facts, to conclude what actions "must have" occurred, modulo some closed world assumption with respect to the set of available actions. In situation calculus one must instead phrase the planning problem as "which s' satisfies $Start(s) \rightarrow Goal(s')$?" where *Start* and *Goal* are predicates for the start and goal states. It is not clear how to extend this approach to other kinds of plan construction, such as story understanding, where a mix of static facts (statements about properties) and statements about actions are given, and one alternatingly infers actions from property changes, and property changes from actions.

The assumption of situation calculus that the state of the world at the end of an action is uniquely determined by the state of the world when the action started, is a very strong assumption and in may applications too strong and therefore misleading. Explicit temporal approaches of course do not have that problem.

For example, the functionality assumption has led to the "double wait" problem, which has been quoted as a "causal anomaly" [BG88]: suppose we are told that a loaded gun will become unloaded if we wait twice, and no information is otherwise known about the effect of the action "wait". What can we conclude about the state of the gun after the first wait action?

The obvious common-sense answer is that the problem is underdetermined, since it may be that "wait" actions sometimes unload the gun and sometimes not. In general, we should not expect a logic to be able to infer general rules about the effect of an action from concrete examples, and particularly not from a single example. The only way such an inference can be made is if we have very strong constraints on the actions' behaviors, e.g. if we assume that they are functions in a very constrained state-space, so that the single example narrows down the number of possible action behaviors

sufficiently. This is exactly what happens when the double wait problem is formulated in situation calculus with a very small number of actions and fluents.

PREFERENCE RELATIONS FOR REASONING ABOUT TIME AND ACTION. The choice of preference ordering is of key importance in a non-monotonic logic. Several preference orderings have been used in earlier A.I. work on reasoning about time and action. *Minimization of change* is the idea that was tried first, and was shown by Hanks and McDermott in their "Turkey shoot" paper to have a serious bug. *Chronological ignorance* has been proposed by Shoham [Sho88], *causal minimization* by Haugh [Hau87] and independently by Lifschitz [Lif87]. Each of these approaches has significant problems, as discussed in the paper by Baker and Ginsberg [BG88].

The approach used here retains some aspects of both change minimization and causal minimization. With respect to change minimization, we argue that there are two change-related minimizations that must done after each other: first, the system should maximize the *expectations for persistence,* in the presence of axioms which state that in certain situations persistence can not be expected. For example in the "Turkey shoot" problem, the property (propositional fluent) of whether then gun is loaded or not, should not be expected to persist during the period when the gun-loading action takes place. If the gun is loaded and Fred is alive, then neither of those properties shall be expected to persist during the gun-firing action. At all times when persistency expectation has not been explicitly disclaimed, it should be expected to hold.

When the persistency expectations have been established, there is a second round for minimization of the set of instances where a property *changes in spite of the expectations* not to change. The parasitical extension in the turkey shoot scenario (where the gun mysteriously becomes unloaded during the waiting period) is characterized exactly by such an unexpected property change. If now the two minimizations are done in the correct order, the parasitical extensions (preferred interpretations) no longer arise in scenarios of the turkey shoot type.

With such a two-step minimization of change, the system will only attempt to minimize those changes that occur in unexpected areas, and will not assign any preference to changes that occur in the course of an action that is known to change the property. Two step change minimization must therefore be combined with *minimization of the set of actions,* since there could otherwise be preferred interpretations with arbitrarily many extra actions besides those which are needed by the axioms.

In summary we have three preference criteria, two related to change and one for minimization of the set of actions. Formally it is possible to combine the last mentioned change criterion with the action-set minimization, so we actually use two preference orderings $\ll$ and $<$: $J \ll J'$ iff J has higher expectations about the persistence of properties ("propositional fluents") than

what J' does. Also $J < J'$ iff J breaks those expectations fewer times, or (if the sets of expectation breaks are equal) if the set of actions in J is a subset of that in J'. These two orderings are then used for preferential entailment, so that we take the set of models of the given axioms, cut it down to those models which are minimal with respect to $\ll$, and then cut down that set to those models which are minimal with respect to $<$.

It might seem that one should be able to combine all three criteria into one single ordering, and just do one minimization, but that in fact does not work well as we shall demonstrate.

Causal minimization methods have been criticized for not being able to deal correctly with logical ramification or "causal anomalies" [BG88]. We show a simple method, different from those proposed before, for dealing with such problems.

Another aspect of previous work using causal minimization was that it used an explicit relation *causes* which characterizes cause-effect properties of action types. This has the disadvantage of imposing notational constraints on what one can say about actions. The work reported here does not share that aspect of causal minimization approaches.

RESULTS. We make the necessary formal definitions for the approach which has now been outlined, and demonstrate through a number of examples that the chosen approach can deal equally well with temporal projection (reasoning forward in time about effects of known actions), temporal explanation (reasoning backward in time to determine the state of the world before a set of known actions), and plan construction (determining which actions must happen, or must have happened, in order to obtain specified changes in the world).

This paper is the first one in a trilogy, where the present paper defines and uses semantic entailment for total (i.e. conventional) interpretations and sequential actions. Part II deals with concurrency issues which are important when actions occur in parallel. Part III generalizes to partial interpretations, where propositions may have the truth-value T, F, or U (for "unknown"), and uses them as the basis for a semantic decision procedure using constraint propagation in partial models (without using inference rules).

2. Basic definitions.

The following definitions are similar to those used by Y. Shoham in "Reasoning About Change" [Sho88].

PREFERENCE SEMANTICS.

Preferential entailment is defined as follows. Let $\mathcal{L}$ be a logic of our choice, and assume that a reverse preference ordering $<$ is given over the interpretations used in $\mathcal{L}$. The ordering shall be reverse in the sense that we

prefer interpretations which are "smaller" w.r.t. $<$.

Now let α be a wff in $\mathcal{L}$, and let Γ be a set of such wff. We define $Mod(\alpha)$ to be the set of all interpretations that satisfy α, and $Mod(\Gamma)$ to be the set of all interpretations that satisfy all members of Γ. Furthermore if Δ is a set of interpretations, then $Min_<(\Delta)$ is defined as the set of all members of Δ which are minimal w.r.t. the reverse preference ordering $<$, i.e. no other member is smaller. We write

$$\Gamma \models_< \alpha$$

and say that Γ *preferentially entails* α iff

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

i.e. α is satisfied by each most preferred model for Γ. [1]

The minimization criterium that is used here must be applied in two steps in order to work correctly in all cases. We therefore extend the definition of preferential entailment to two orderings in the following obvious way: $\Gamma \models_{\ll,<} \alpha$ iff

$$Min_<(Min_{\ll}(Mod(\Gamma))) \subseteq Mod(\alpha)$$

A LOGIC OF ACTIONS IN TIME INTERVALS.

We now proceed to introduce a simple, "propositional" logic of actions in time intervals, similar to the one Shoham uses in section 2.2.2 in [Sho88]. However, we make the following changes, in order to deal with persistence. The set of primitive propositions (P, in Shoham's notation) is split into two parts, namely the homogeneous (or liquid) ones, and the others. The reason is that persistence can only be assumed for homogeneous propositions. We will use the term *properties* for homogenous proposition symbols, and *action types* for the others. - Furthermore, a relation $Persists$ is added to the syntax and the semantics.

With this preamble, the following definitions should be fairly obvious.

[1] This notation differs from the one used by Shoham in that when he would write the preference ordering between two interpretations as

$$M \sqsubset M'$$

we write it as

$$M' < M$$

The reason is that later in the trilogy, when we generalize to partial interpretations, the partiality ordering between interpretations defines a lattice, and it is therefore more appropriate to use $\sqsubset$ to denote that ordering. Preference orders are not in general required to be lattices, and neither $<$ nor $\ll$ (as redefined in part II) are.

A reason for turning $<$ around is that we usually think of non-monotonic reasoning as minimizing some entities, so the most preferred interpretations should be smallest in the ordering. Also, we want to think of preferred interpretations as those that make fewer assumptions or commitments. This intuition applies for the $\sqsubset$ ordering, where lesser interpretations map more propositions to the unknown truth-value, U. It therefore makes sense to apply the same intuition consistently to preference orderings as well.

SYNTAX

Let the following be given:
C, a set of properties (primitive homogenous propositions);
H, a set of action types (primitive non-homogeneous propositions);
TC, a set of time-point constant symbols;
TV, a set of variable symbols for time-points.

The set of logical formulas (i.e. well formed formulas) is defined inductively to consist of the following:

$$t = u$$

$$t \prec u$$

$$Holds(t, h, u)$$

$$Holds(t, c, u)$$

$$Persists(t, c, u)$$

$$\alpha \wedge \beta$$

$$\neg\alpha$$

$$\forall v \alpha$$

and, for notational convenience also the following formulas which always could have been expressed using the previous ones:

$$t \preceq u$$

$$Notholds(t, c, u)$$

$$Notpersists(t, c, u)$$

$$Holds(t, c)$$

and the usual reportoire of propositional connectives. The syntactic variables in the definitions are supposed to range as follows: $t, u \in TC \cup TV$; $h \in H$; $c \in C$; α, β are logical formulas; $v \in TV$.

SEMANTICS

An *interpretation* is a tuple $\langle \mathbf{T}, tp, R, P, M_T, X \rangle$, where

$\mathbf{T}$ is a nonempty, enumerable set of timepoints

$tp \subseteq \mathbf{T} \times \mathbf{T}$ is a strict total order

$R \subseteq \mathbf{T} \times C$

$P \subseteq \mathbf{T} \times H \times \mathbf{T}$

M_T is a mapping $TC \mapsto \mathbf{T}$

$X \subseteq \mathbf{T} \times C$

We assume that $\mathbf{T}$ and tp behave like the integers, so that for each time-point t there is a unique successor $succ(t)$. The identity of the members of $\mathbf{T}$ is immaterial, so if an interpretation is changed by replacing every time-point t with $f(t)$ where f is a permutation on $\mathbf{T}$, everywhere in all components of the interpretation, then we consider that we still have the "same" interpretation. When comparing two interpretations, for example for the purpose of preference ordering, we can therefore always assume that they have the same $\mathbf{T}$ and tp components.[2]

The members of P will be called *actions*. Thus an action is a triple whose second element is an action type.

A *variable assignment* VA is a mapping $TV \mapsto \mathbf{T}$.

The truth-value that a wff α has in an interpretation J with these components, under the variable assignment VA, is written

$$J \models \alpha_{[VA]}$$

It shall be T under the following conditions, and F otherwise:[3]

If α has the form $t = u$: iff $val(t) = val(u)$

If α has the form $t \prec u$: iff $\langle val(t), val(u)\rangle \in tp$

If α has the form $t \preceq u$: iff either of the previous two conditions holds.

If α has the form $Holds(t, h, u)$: iff $\langle val(t), h, val(u)\rangle \in P$

If α has the form $Holds(t, c)$: iff $\langle val(t), c\rangle \in R$

If α has the form $Holds(t, c, u)$: iff $t \preceq t' \preceq u \rightarrow \langle val(t'), c\rangle \in R$

If α has the form $Persists(t, c, u)$: iff $t \preceq t' \prec u \rightarrow \langle val(t'), c\rangle \in X$

If α has the form $Notholds(t, c, u)$: iff $t \preceq t' \preceq u \rightarrow \langle val(t'), c\rangle \notin R$

If α has the form $Notpersists(t, c, u)$: iff $t \preceq t' \prec u \rightarrow \langle val(t'), c\rangle \notin X$

where $val(t)$ refers to the value of a temporal term, determined in the obvious way using M_T and VA. The symbols $\prec$ and $\preceq$ are borrowed from the object language to the meta language in the last four cases, and the formulas formed with propositional connectives and quantifiers are defined as usual.

Informally, $Holds(t, c, u)$ means that the property c is true throughout the interval from time t to time u, including both endpoints, and $Notholds(t, c, u)$ means that it is false throughout the same interval. $Persists(t, c, u)$ means that the property c will be preferred (in the sense of the preference ordering on interpretations) to be unchanged over the same interval, so that for each t' in the interval, including the left endpoint t *but not* the right endpoint u, we prefer interpretations where

[2]The reason why we do not omit $\mathbf{T}$ and tp and use the integers themselves as time-points, is in order to facilitate the forthcoming generalization to partial interpretations.

[3]The reason that $\models$ is seen as a function whose value is T or F, is that later on we extend the definition so that the value can be either of T, F, or U.

$$\langle t', c\rangle \in R \leftrightarrow \langle succ(t'), c\rangle \in R$$

Notpersists means that we do not have any particular preference for c remaining constant in any point in the interval; however we do not have any preference for it changing either.

For a given logical formula α, $Mod(\alpha)$ is the set of interpretations J for which

$$J \models \alpha_{[VA]}$$

is T for all variable assignments VA.

AN EXAMPLE OF THE NOTATION.

We exemplify the notation with the perennial Yale turkey-shot-example [HM87]. The following axioms would be used to characterize the specific scene:

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

$$Holds(t_2, wait, t_3).$$

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

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

$$Holds(t_1, fredalive).$$

Here of course *load*, *wait*, and *fire* are in H, and *gunloaded* and *fredalive* are in C. The constants t_1 through t_4 are in TC. (Temporal variables will be written t, u, t', u',...) We would expect that these axioms, together with appropriate axioms specifying the effects of loading, waiting, and firing, and appropriate general axioms for temporal reasoning, should preferentially entail for example that

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

In fact, since (in the usual rendering of this example) no effect is associated with *wait* except the lapse of time, the example could be rendered just as well by replacing the second formula with

$$t_2 \prec t_3.$$

and the same conclusion should be entailed.

3. Axioms and preference order < for interpretations.

When we define the preference order on interpretations, we can always assume that the first two elements are equal between the interpretations, for the reasons stated above. Also, the component M_T will always be irrelevant when comparing the preference between interpretations. It is therefore the three components R, P, and X that determine preferences.

We use the following approach for preference criteria. Axioms for the

effects of specific action types are written as in the following example for "toggleload", an action type which will load the gun if it was unloaded, and unload it if it was loaded:

$$Holds(t, toggleload, u) \rightarrow Notpersists(t, gunloaded, u) \wedge$$
$$([Holds(t, gunloaded) \wedge \neg Holds(u, gunloaded)] \vee$$
$$[\neg Holds(t, gunloaded) \wedge Holds(u, gunloaded)]).$$

This axioms regulates and relates the preconditions and the postconditions in terms of values for properties, in this case *gunloaded*, and it also specifies that those properties which are affected by the action, shall not persist for the duration of the action.

The relation *Persists* could more precisely have been called *PreferablyPersists*, and expresses an expectation or preference that the property in the argument does not change. In the axiom, the negated relation *Negpersists* states that there shall not be any expectation that *gunloaded* does not change while the gun is being loaded. Comparing two interpretations with the same R and P components, the interpretation with a larger (in the sense of $\subset$) X component is always preferred. This means that we primarily maximize the relation *Persists*. (Remember that *Persists* is interpreted using the relation X in interpretations). Axioms specifying the effects of actions will normally imply nonpersistence for some properti(es) during a time interval; the present preference condition guarantees that *Persists* holds for all argument combinations where it has not been explicitly implied not to hold, by the current set of actions P.

It is easily seen that for axioms of the kind given above, there will for each combination of P and R be exactly one interpretation which is preferred by the primary preference condition.

Next, comparing actions with different P or R, we take two additional considerations. On one hand, we want to minimize the set P, i.e. to prefer fewer actions over larger sets of actions. This is because we do not want the reasoning system to assume unwarranted actions, and their resulting effects. On the other hand, we also prefer to minimize the number of points where $Holds(t, c)$ changes value, i.e. where

$$Holds(t, c) \leftrightarrow \neg Holds(succ(t), c)$$

but *considering only those* $\langle t, c\rangle$ *where c persists,* i.e. where according to the current axioms and the maximization of *Persists* we would otherwise expect that it should not change value.

More precisely, for a given interpretation J we define the abnormal change set $abch(J)$ as

$$\{\langle t, c\rangle \mid \langle t, c\rangle \in X \wedge [\langle t, c\rangle \in R \leftrightarrow \langle succ(t), c\rangle \notin R]\}$$

and prefer J over J' when $abch(J) \subset abch(J')$.

If we minimize separately over P and over $abch(J)$, it remains to deter-

mine in which order. If we minimize secondarily over *abch* and tertiarily over P, it means that if there is an interpretation with a smaller $abch(J)$ set, and ideally an empty such set, then it will be preferred regardless of how large its P set is. With the opposite order, we will minimize the number actions, which presumably means that we only get to include actions which are explicitly stated in the axioms, or which are explicitly implied by the axioms from other facts, but not actions which would be needed in order to account for otherwise mysterious state changes.

The former choice does most things right, but it does have the disadvantage that it handles *logical ramification* incorrectly in some cases. Suppose we have two properties *alive* and *dead* (referring only to the aliveness of a single individual turkey, for simplicity), and an axiom

$$Holds(t, alive) \leftrightarrow \neg Holds(t, dead).$$

If all action types which may kill or resurrect refer to the same one of these two properties, e.g. *alive*, then there will be no problems, since there will be no interpretation for which $abch(J)$ is the empty set. However, suppose the action type *fire* (referring to the gun) has been specified so that it changes the turkey's *alive* property from T to F, and the action type *suicide* has been specified so that it changes the turkey's *dead* property from F to T. If now the scene axioms state that the turkey is initially *alive*, and that a *fire* action takes place, every most preferred model must certainly let *alive* go from T to F, and let *dead* go from F to T at the same time. If there is no additional action in the interpretation, then *Persists* will only be negated for *alive*, and by its maximization *Persists* will apply for *dead* during the action interval. Therefore the *abch* set will not be empty. However one can construct another interpretation by adding a *suicide* action during the same time interval, whereby the abnormal change set becomes smaller and the action set becomes larger, which is supposed to be preferred. Thus the axioms that Fred was initially alive, and that he was shot dead, will preferentially entail that he both was killed and committed suicide.

One can think of several ways out of this problem. The easiest way is to complement each axiom which relates properties, such as

$$Holds(t, alive) \leftrightarrow \neg Holds(t, dead).$$

with axioms such as

$$Nonpersists(t, alive, u) \leftrightarrow Nonpersists(t, dead, u).$$

This admittedly gets to be a bit clumsy especially for more complex relationships between properties, but it can always be done. It is better than other solutions which have been discussed, such as to reify the notion of implication itself, or to introduce a classificiation into "primitive" and "derived" properties. Logical ramification like in this example is also related to the concurrency issues which will be treated in part II.

Besides logical ramification, there is also the problem of *physical ramifi-*

cation, meaning that an action has a number of additional effects that we do not wish to enumerate in the action specification axiom. Firing a gun has side-effects such as a noise, sooting the pipe, and so on. We shall not discuss that topic in the present paper, but feel that the likely best approach to physical ramification is to use causation axioms which say that one action (firing the gun) causes those other actions (sound happening, sooting,...) which then in turn may change properties. Our preference relation tends to minimize the set of actions among the available models, as one of its criteria, and will therefore admit the actions which are implied by causation axioms, but not arbitrary other actions. This approach also means that concurrent actions become involved, just like for logical ramification.

In summary, therefore, the precedence ordering between interpretations is defined to be such that we primarily maximize X, secondarily minimize $abch(J)$, and tertially minimize P. As was mentioned in section 1, we have to explicitly use two orderings,[4] defined as follows. Assuming that

$$J_1 = \langle \mathbf{T}, tp, R_1, P_1, M_1, X_1 \rangle$$

$$J_2 = \langle \mathbf{T}, tp, R_2, P_2, M_2, X_2 \rangle$$

we define (with $\supset$ the superset relation)

$$J_1 \ll J_2 \leftrightarrow R_1 = R_2 \wedge P_1 = P_2 \wedge X_1 \supset X_2$$

$$J_1 < J_2 \leftrightarrow [abch(J_1) \subset abch(J_2)] \vee$$
$$[abch(J_1) = abch(J_2) \wedge P_1 \subset P_2]$$

The definition of preferential semantic entailment in our logic is now complete. It uses a set of axioms Γ_s characterizing the scene, another set Γ_a of axioms specifying the action types used in the scene, and a set Γ_t of general axioms for temporal reasoning, consisting (for sequential actions, only) of all instances of the axiom schema

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

where a is any member of A. We recognize as a conclusion any logical formula α which satisfies

$$\Gamma_s \cup \Gamma_a \cup \Gamma_t \models_{\ll,<} \alpha$$

[4]The reason why we cannot combine these two orderings into one is the following. Suppose we have two interpretations J_1 and J_2 which have the same, non-empty abnormal change sets, and J_2 contains a larger set of actions, so $J_1 < J_2$. Consider J_2' which differs from J_2 by choosing X_2' as the empty set, which means that also $abch(J_2')$ is empty. By the definitions we have

$$J_2 \ll J_2' < J_1 < J_2$$

so if the two orderings $<$ and $\ll$ are combined we obtain contradictions with the strictness and transitivity axioms. On the other hand, the second and the third criterium can be combined into one preference relation $<$ since *abch* only depends on X and R, and is independent of P.

using the definition of $\models_{\ll,<}$ of chapter 2 and the definitions of $\ll$ and $<$ given here.

4. Properties of the entailment relation for some simple examples.

The claim is that preferential entailment $\models_{\ll,<}$ will give the right results for typical temporal reasoning problems, and in particular for the standard toy examples in the literature. Of course a key point with the definitions made here is that they allow parallel actions, so additional examples will be needed in order to demonstrate that the entailment works properly there also. That is the topic of part II of this paper. However it is appropriate to first verify that the system can handle the usual test batch.

TEMPORAL PROJECTION: THE YALE TURKEY SHOOT.

The scene description for this classical example [HM87] was given at the end of section 2. The defining axioms for *load* and *fire* are:

$$
\begin{aligned}
&Holds(t, load, u) \rightarrow \\
&\quad [\neg Holds(t, gunloaded) \wedge Holds(u, gunloaded) \wedge \\
&\qquad Notpersists(t, gunloaded, u)] \ \vee \ Holds(t, gunloaded, u).
\end{aligned}
$$

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

It is immediately seen that there will be a number of distinct interpretations which are formed by choosing differently the function M_T which defines the values of the temporal constants, and that these interpretations do not have any preference between them. It is also seen that all models for these axioms must have at least two different actions in the set P. Therefore if there is *some* model J where $abch(J)$ is empty and with only these two actions, then *all* preferred models will fall into that category, and it only remains to minimize $\ll$.

The definition of $\ll$ stated that for equal P and R one should maximize the relation X. Clearly the best one can do is to choose X so that it applies everywhere except when it is explicitly implied not to apply, which is within the duration of the actions.

Within the actions there may be one or more changes. Thus there will be some models where *gunloaded* goes once from false to true in the course of the *load* action, and other models where it goes back and forth several times. Also, even if the change only occurs once, there will be different models depending on exactly when the change takes place. Neither of these interpretations will be preferred over the others.

In summary, then, the number of preferred models is fairly large. Still it is clear that the expected conclusion,

$$\neg Holds(t_4, alive).$$

is satisfied in all of them.

TEMPORAL EXPLANATION: THE STANFORD MURDER MYSTERY.

The "Stanford Murder Mystery" [BG88] is a toy example of deriving information about earlier world-states, when something is known about a current state in the world, and the actions which led up to it. In this example we use the same action-type specifications as in the Yale Turkey Shoot, but the following scene description:

$$Holds(t_1, fire, t_2).$$

$$t_2 \prec t_3.$$

$$Holds(t_1, fredalive).$$

$$\neg Holds(t_3, fredalive).$$

What can then be concluded about

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

Again it is clear that one can always (regardless of how the temporal constants are chosen) find a model with one *fire* action, and a *Persist* relation which applies everywhere outside the action. In all such preferred models, the abnormal change set is empty, and only one of the alternatives in the specification for *fire* can be applied, so

$$Holds(t_1, gunloaded).$$

has the value T as it should. Interpretations where it is F have to either add more actions, or increase the abnormal change set, and are therefore dispreferred by $<$.

PLAN CONSTRUCTION: THE TURKEY CONSPIRACY.

In plan construction, the scene axioms give information about the state of the world at various times, and the problem is to identify the actions which must be taken, or which must have been taken, for these changes to occur. In explicit temporal logic, as we discussed in the introductory section, plan construction can be done by forward inference, i.e. the given facts entail the plan. A simple example, still in the context of the action specification axioms of the Yale turkey shooting, is as follows. Let the scene axioms be

$Holds(t_1, fredalive)$.

$\neg Holds(t_1, gunloaded)$.

$t_1 \prec t_3$.

$Holds(t_3, fire, t_4)$.

$\neg Holds(t_4, fredalive)$.

It is easily seen that all $\ll, <$-minimal models for these axioms are obtained by adding one *load* action, so the axioms entail[5]

$$\exists t_1' \exists t_2 [Holds(t_1', load, t_2) \wedge t_1 \preceq t_1' \wedge t_2 \preceq t_3].$$

At this point there is an obvious objection: what if the axioms do not specify whether the gun is loaded at time t_1 or not? Then the $\ll, <$-minimal models will be those where the gun is loaded already and no *load* action is necessary. Those models are preferred over models where the gun was initially unloaded, because they contain an action less.

Rather than trying to change the rules of entailment, we think it is better to leave the matter as it stands, and deal with it at the receiving end in a planning or understanding system. In general, a plan construction system should report not only a proposed plan, but also report what assumptions the plan makes. In this case, the proposition $Holds(t_1, gunloaded)$ is also T in all the minimal models, and is therefore entailed by the given axioms. It should be reported by the plan construction system as an assumption of the plan. If the overall planner does not like the assumption, it should come back and request a revised plan which works in the contrary case that the gun is unloaded.

One of the requirements on a decision procedure that is going to be used for plan construction, is therefore that is must be able to recognize the extra assumptions that are being made in plans. The procedure that we describe in part III of these papers does have that property.

The problem we have described is a variety of a conditional planning problem, where the set of actions required in one branch is a subset of the actions required in the other branch. Notice that for other conditional planning problems, where alternative choices of actions are needed, there is no problem at all. Suppose for example that we have the properties *thirsty, coke-available,* and *pepsi-available;* the scene axioms are[6]

$Holds(t_1, thirsty)$.

[5]Actually this is not quite true, since we have not straightened out the problems of concurrency yet. The axioms we have so far, admit some interpretations where *load* and *fire* are performed during overlapping time intervals. The additional requirements in part II will solve this problem.

[6]Switching to this other context is a peaceful alternative and a case of affirmative action. Otherwise we could have extended the turkey shoot scenario with alternative ways of killing Fred in case the gun was unloadable.

$\neg Holds(t_4, thirsty)$.

$t_1 \prec t_4$.

and furthermore there is an action *have-coke* which makes *thirsty* be false at the end of the action under the condition that *coke-available* is T, and similarly for *pepsi-available*. Then there will be some $\ll, <$-minimal models where the following formula is T:

$$\exists t_2 \exists t_3 (Holds(t_1, cokeavailable, t_4) \wedge Holds(t_2, havecoke, t_3) \wedge$$
$$t_1 \preceq t_2 \wedge t_3 \preceq t_4).$$

and some where the following one is T:

$$\exists t_2 \exists t_3 (Holds(t_1, pepsiavailable, t_4) \wedge Holds(t_2, havepepsi, t_3) \wedge$$
$$t_1 \preceq t_2 \wedge t_3 \preceq t_4).$$

There are no other minimal models[7], since in the case where neither pepsi nor coke is available, the system does not know of any action that solves the problem. The only next thing available is interpretations with a non-empty abnormal change set and where the thirst goes away by itself, but such models have been defined to have lower $<$ preference. Therefore, the disjunction of the two last mentioned formulas is T in all minimal models, and constitutes the conditional plan.

5. Discussion.

The strong points for the logic described here are:

(1) that it is equally applicable to temporal projection, temporal explanation, and plan construction;

(2) that it is readily able to characterize concurrent actions;

(3) that all the definitions involved are relatively simple.

Also of course it gives the right common-sense answers for current test-cases in the literature.

One may wonder whether it is not possible to simplify the logic even further. In particular, why is it necessary to go through the two steps of first generating expectations for no change, and then identify the violations of the expectations, and to minimize once for each step? The alternative would be to let the relation *Persists imply* no-change, using an extra axiom schema, and to just maximize *Persists* and minimize P. That approach however has the same problem as Hanks and McDermott's original formalizations of the Yale turkey shoot [HM87]. The axiom that specifies the effects of *fire* implies that *fredalive* does not persist, if the gun is loaded, but does not imply

[7] Again we have to assume that the solutions to concurrency issues in the paper's part II are available here

nonpersistence if the gun is unloaded. Therefore, the simplified preference ordering would give equal preference to the alternative interpretation where *fredalive* is T and persists during the firing, but *gunloaded* does not persist during the wait period. The use of two distinct minimization steps solves that problem.

We have not addressed the question of an axiomatization for the semantics described here, for the simple reason that we do not at this point see any major need for it. Traditionally, the logical axioms (the axioms for the logic itself, rather than those describing the application) and the inference rules are seen as prerequisites of an implementation. However in the third paper of the present trilogy, we describe a semantic decision procedure for $\ll, <$-preferential entailment, using constraint propagation in partial interpretations. The semantics that was defined in this paper is generalized in a straightforward way, and can then be used directly for the data structures of the reasoning engine. No axiomatization of the logic is needed.

Suggestions for future work are discussed at the end of the third paper in the sequence.

References

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

[BG88] Andrew B. Baker and Matthew L. Ginsberg. Some problems in temporal reasoning. To appear, 1988.

[Hau87] B. Haugh. Simple causal minimizations for temporal persistence and projection. In *Proc. Sixth AAAI*, pages 218–223, 1987.

[HM87] Steve Hanks and Drew McDermott. Nonmonotonic logics and temporal projection. *Artificial Intelligence*, 33(3):379–412, 1987.

[Lif87] Vladimir Lifschitz. Formal theories of action (preliminary report). In F. Brown, editor, *Proceedings of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, Lawrence, Kansas, 1987. Morgan Kaufmann.

[McD82] Drew V. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[MH69] John McCarthy and Patrick Hayes. Some philosophical problems from the viewpoint of artificial intelligence. In *Machine Intelligence, Vol. 4*, pages 463–502. Edinburgh University Press, 1969.

[MW87] Zohar Manna and Richard Waldinger. How to clear a block: A theory of plans. *Journal of Automated Reasoning*, 3:343–377, 1987.

[Rei87] Hans Reichgelt. Semantics of reified temporal logic. In J. Hallam and C. Mellish, editors, *Advances in Artificial Intelligence*, pages 49–61. Wiley and Sons, 1987.

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

Avdelning, institution, fakultet
Division, department, faculty

Department of Computer and
Information Science
Linköping University

ISBN:

ISSN: ISSN-0281-4250

Rapportnr:
Report no: LiTH-IDA-R-88-27

Upplagans storlek:
Number of copies:

Datum:
Date: September 1988

Projekt:
Project:

Titel:
Title: Non-Monotonic Entailment for Reasoning about Time and Action
Part I: Sequential Actions

Författare:
Author: Erik Sandewall

Uppdragsgivare:
Commissioned by:

Dnr.:
Call no:

Rapporttyp:
Kind of report:

- ☐ Examensarbete/Final project
- ☐ Delrapport/Progress report
- ☐ Reserapport/Travel report
- ☐ Slutrapport/Final report
- ☒ Övrig rapport/Other kind of report

Rapportspråk:
Language:

- ☐ Svenska/Swedish
- ☒ Engelska/English
- ☐ _____

Sammanfattning (högst 150 ord):
Abstract (150 words):

Abstract. The paper difines a logic for reasoning about time and action, in terms of syntax, semantics, and preference relations on interpretations. Temporal logic with explicit representation of time-points is used, rather than situation calculus. The preference relations prefer (1) interpretations which maximize the expectations that fluents do not change, (2) interpretations which minimize the number of instances where fluents change in spite of expectations to the opposite, and (3) interpretations which minimize the set of actions. The preferential entailment relation based on these preferences is equally applicable to temporal projection, temporal explanation, and plan construction (as used for planning, story understanding, and diagnosis). A number of examples are described supporting the thesis that this semantics obtains the results dictated by common sense.

Nyckelord (högst 8):
Keywords (8):

Bibliotekets anteckningar

A Selection of Previous Research Reports.

LiTH-IDA-R-88-18 **Nils Dahlbäck:** Mental Models and Text Understanding - a Commented Review.

LiTH-IDA-R-88-17 **Bengt Lennartsson:** Multi-Level Software Architectures - Industrial Experiences and Research Directions. This report is a revised version of a paper in *Advance Working Papers of The Second International Workshop on Computer-Aided Software Engineering.* Cambridge, Massachusetts. July 12-15, 1988.

LiTH-IDA-R-88-16 **Lin Padgham:** A Model and Representation for Type Information and Its Use in Reasoning with Defaults. Also in *Proc. of AAAI'88, American Association for Artificial Intelligence*, 1988.

LiTH-IDA-R-88-15 **Lin Padgham, Ralph Rönnquist:** LINCKS: An Imperative Object Oriented System. Also in *Proc. of the 20th Annual Hawaii Int. Conf. on System Sciences*, Hawaii, 1987.

LiTH-IDA-R-88-13 **Erik Tengvald:** Ett kartorienterat planeringssystem för autonoma farkoster, en design diskussion.

LiTH-IDA-R-88-12 **Erik Sandewall, Johan Hultman, Erik Tengvald:** Software Architecture and Programming Paradigms for Robotics Applications.

LiTH-IDA-R-88-11 **Mats Wirén:** An Incremental Chart Parser for PATR.

LiTH-IDA-R-88-10 **Mats Wirén:** A Control-Strategy-Independent Parser for PATR. Revised version of paper appearing in *Proc. of First Scandinavian Conf. on Artificial Intelligence*, Tromsö, Norway, March 9 - 11, 1988.

LiTH-IDA-R-88-09 **Peter Fritzson:** Incremental Symbol Processing.

LiTH-IDA-R-88-08 **Erik Sandewall:** Formal Semantics for Reasoning about Change with Ramified Causal Minimization.

LiTH-IDA-R-88-07 **Erik Sandewall et al:** IDA Annual Report 1987.

LiTH-IDA-R-88-06 **Christer Bäckström:** A Representation of Coordinated Actions Characterized by Interval Valued Conditions.

LiTH-IDA-R-88-05 **Christer Bäckström:** Keeping and Forcing: How to Represent Cooperating Actions.

LiTH-IDA-R-88-04 **Wlodek Drabent, Simin Nadjm-Tehrani, Jan Maluszynski:** Algorithmic Debugging with Assertions. Also in *Proc. of META 88 - Workshop on Meta Programming*, Bristol, June 21-24 1988.

LiTH-IDA-R-88-03 **Staffan Bonnier, Jan Maluszynski:** Towards a Clean Amalgamation of Logic Programs with External Procedures. Accepted to the *Joint Fifth Int. Logic Programming Conf. and Fifth Logic Programming Symp.*, Seattle Washington, August 15-19, 1988.

LiTH-IDA-R-88-02 **Ulf Nilsson:** Inferring Restricted AND-Parallelism in Logic Programs using Abstract Interpretation.

LiTH-IDA-R-88-01 **Jacek Leszczylowski, Staffan Bonnier, Jan Maluszynski:** Logic Programming with External Procedures: Introducing S-Unification. A shorter version is to appear in *Information Processing Letters*, North Holland.

LiTH-IDA-R-87-26 **Jonas Löwgren:** Applying a Rapid Prototyping System to Control Panel Dialgues.

LiTH-IDA-R-87-24 **Sven Moen:** Drawing Dynamic Trees.

LiTH-IDA-R-87-22 **Lars Ahrenberg, Arne Jönsson:** An Interactive System for Tagging Dialogues. Also in *Proc. of 14th Conference of the Association for Literary and Linguistic Computing*, Göteborg 1-5 June, 1987.

LiTH-IDA-R-87-21 **Harold W. Lawson, Jr.:** Challenges and Directions in Computers and Education.

LiTH-IDA-R-87-20 **Krzysztof Kuchcinski, Zebo Peng:** Parallelism Extraction from Sequential Programs for VLSI Applications. This paper is to appear in *Microprocessing and Microprogramming, the Euromicro Journal*, 1988.

The Department of Computer and Information Science
Linköping University

organizes undergraduate and graduate studies in *Computer Science*, *Telecommunication and Computer Systems*, and *Administrative Data Processing*. Research activities have an emphasis on advanced software technology and computer systems design and are organized in a number of research laboratories:

- *ACTLAB - Laboratory for Complexity of Algorithms*, which is concerned with the design and analysis of efficient sequential and parallel algorithms, and complexity theory, especially in the areas of computational geometry, data structures on bounded domains and graph algorithms.
- *ASLAB - Application Systems Laboratory,* which studies design of advanced support systems for interactive use of computers, including tools for automated construction of applications software.
- *CADLAB - Laboratory for Computer-Aided Design of Electronics*, which concentrates its research activities around tools for integrated development of hardware and software, graphics-based modelling and simulation techniques.
- *LIBLAB - Laboratory for Library and Information Science,* which studies methods for access to documents and the information contained in the documents, concentrating on catalogs and bibliographic representations, and on the human factors of library use.
- *LOGPRO - Laboratory for Logic Programming*, which concentrates its research activities around foundations of logic programming, relations to other programming paradigms and methodology.
- *NLPLAB - Natural Language Processing Laboratory*, which conducts research related to the development and use of natural language interfaces to computer software.
- *PELAB - Programming Environments Laboratory*, which works with design of tools for software development, specific functions in such tools and theoretical aspects of programs under construction.
- *RKLLAB - Laboratory for Representation of Knowledge in Logic*, which covers issues and techniques such as non-monotonic logic, probabilistic logic, modal logic and truth maintenance algorithms and systems.

Research Reports 1988

LiTH-IDA-R-88-27 **Erik Sandewall:** Non-Monotonic Entailment for Reasoning about Time and Action. Part I: Sequential Actions.

LiTH-IDA-R-88-26 **Lin Padgham:** NODE: A Database for Use by Intelligen Systems. Also in *Proc. of ISMIS '88,* Torino Italy, October 12-15, 1988.

LiTH-IDA-R-88-23 **Johan Fagerström, Yngve Larsson:** Two Contributions on Debugging Distributed Systems. Also in *Proc. of ACM SIGPLAN/SIGOPS, Workshop on Parallel and Distributed Debugging*, Madison, Wisconsin, May 5-6, 1988.

LiTH-IDA-R-88-22 **Johan Fagerström:** Design and Test of Distributed Applications. Also in *Proc. of the International Conference on Software Engineering, ICSE*, Singapore, April 1988.

LiTH-IDA-R-88-21 **Johan Fagerström, Lars Strömberg:** A Paradigm for Distributed System Design and Test. Also in *Proc. of COMPCON'SPRING 88*, San Fransicso, Febr-March 1988.

LiTH-IDA-R-88-19 **Lin Padgham, Ralph Rönnquist:** From a Technical to a Humane Environment: A Software System Supporting Cooperative Work. Also in *Proceedings of the Nordicana Conference on USER INTERFACES,* Rüscheln, 1986.