Computer Science
University of Manchester

Domain Constraint Maintenance

Ying Zhang, Howard Barringer, David Carlisle

Technical Report UMCS–94–8–1

Domain Constraint Maintenance *

Ying Zhang, Howard Barringer, David Carlisle

Department of Computer Science
University of Manchester
Oxford Rd., Manchester, U.K.
yzhang@cs.man.ac.uk, howard@cs.man.ac.uk, carlisle@cs.man.ac.uk

August 1994

Abstract

In this report, we propose a set of domain independent and human independent restrictions on actions and we show that the restrictions can prevent actions from being improperly defined. Then, based on the set of restrictions, we propose a new conflict detecting method for nonlinear plans and we show that this new method is sound and complete.

Recent technical reports issued by the Department of Computer Science, Manchester University, are available by anonymous ftp from `ftp.cs.man.ac.uk` in the directory `pub/TR`. The files are stored as PostScript, in compressed form, with the report number as filename. They can also be obtained on WWW via URL `http://www.cs.man.ac.uk/csonly/cstechrep/index.html`. Alternatively, all reports are available by post from The Computer Library, Department of Computer Science, The University, Oxford Road, Manchester M13 9PL, UK.

1 Introduction

Basically, planning can be seen as an activity of manipulating actions. In the planning literature, actions are defined in an ad-hoc fashion. There do not yet exist human independent rules to distinguish between "correctly defined" actions and "incorrectly defined" actions. We believe that for any properly defined action, all its properties (i.e. preconditions and postconditions) have some kind of inherent connection among them. None of these properties can be detached from the definition of the action and none of other irrelevant properties can be attached to the definition. For example, we can't expect that an action can be properly implimented with one of its preconditions unsatisfied. We find that the inherent connection among the properties of an action can be captured by domain constraints.

Domain constraints specify impossible states of a planner's world in a specific domain. For example, in the blocks world, $(On(x,y), Clear(y))$ is a domain constraint which states that it is impossible that block y is clear when block x is on the top of it and vice-versa. In planning, domain constraints are used to improve planning efficiency by pruning the search space to avoid branches which will lead to impossible states specified by the domain constraints (Warren, 1974; Drummond & Currie, 1987; Currie & Tate, 1989; Currie & Tate, 1990; Allen, 1991a). Kelleher et al. suggested that some of the domain constraints can be automatically constructed from actions (Kelleher & Cohn, 1992; Kelleher, 1990). This, from the other direction, confirms the mutually dependent relation between actions and domain constraints.

In this report, we propose a set of domain independent and human independent restrictions which gives a formal description of the mutually dependent relation between the actions and the domain constraints in a problem domain. We also show that these restrictions can prevent actions from being improperly defined.

Nonlinear planning is believed to be exponentially more efficient than linear planning (Chapman, 1987). Nonlinear planning avoids most of the unnecessary backtrackings experienced by linear planning. However, some other computational problems arise in nonlinear planning. One of the problems is that it is more difficult to verify nonlinear plans than to verify linear plans. This difficulty is caused by the fact a nonlinear plan is partially described. A nonlinear plan is partially specified in two ways, the orderings are partially specified and the variables in the plan are partially described. A nonlinear plan is conflict free only if all the possible completions of the partially specified plan are conflict free.

Based on the proposed restrictions on actions, in this report, we present a new approach to detect conflicts in nonlinear plans. Using this new method, instead of by considering all the possible completions, we detect conflicts in a nonlinear plan by checking that if the set of domain constraints in the concerned problem domain is satisfied by the nonlinear plan. We term this conflict detecting method *domain constraint maintenance*. we prove that, under the proposed restrictions, a nonlinear plan is conflict free if and only if the corresponding set of domain constraints is satisfied in its model; i.e. domain constraint maintenance is sound and complete.

2 Preliminaries

In this section, we give descriptions of domain constraints and models of nonlinear plans.

For deductive planners, planning is a reasoning process within some formal logic. In (Zhang, 1994), we presented a point-based, reified temporal interval logic for nonlinear planning. In the temporal logic, time is discrete and is in a partial order structure; time points are the only temporal primitives and, therefore, time intervals are referred to by their end-points; assertions of proposition types are interpreted over the biggest possible time intervals; a proposition type is associated with a time interval through a global predicate; there are two proposition types, property and action in the logic; to distinguish between these two types, we use two global predicates $Hold$ and $Exec$. The atomic formula $Hold(P, t_i, t_j)$ expresses that the property P holds true over the biggest possible time interval (t_i, t_j) and the atomic formula $Exec(A, t_i, t_j)$ expresses that the action A is executed over the biggest possible time interval (t_i, t_j). An interpretation of the temporal logic is a tuple $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{D}, \pi_F, \pi_P \rangle$, where $\mathcal{M}$ is a set of time points; $\mathcal{R}$ is a partial order relation among the time points in $\mathcal{W}$; $\mathcal{D}$ is the domain of non-temporal variables; π_F is an interpretation of the function symbols and π_P is an interpretation of the proposition types. In appendix A, we give the formal definition of the temporal logic.

In a state-based language, a domain constraint expresses that two properties which are defined to be mutually exclusive are not allowed to hold true at the same state or time point. In our nonlinear interval temporal logic, if two properties are defined to be mutually exclusive, then the temporal intervals over which they hold true respectively must not *possibly* and properly overlap. Then, a set of domain constraints Σ_D for a problem domain is a set of formulas of the form:

$$\forall t_1, t_2, t_3, t_4.(Hold(p, t_1, t_2) \land Hold(q, t_3, t_4) \Longrightarrow t_2 \preccurlyeq t_3 \lor t_4 \preccurlyeq t_1)$$

If Σ_D contains the above formula, the properties p and q are said to be *mutually exclusive* according to Σ_D. In the following discussion, for convenience, we often use the notation $p \leftrightsquigarrow q$ to represent the domain constraint of the above form. It is not difficult to prove that if a nonlinear temporal model satisfies a domain constraint of the above form, then the intervals of the two properties involved will not possibly and properly overlap.

In this temporal logic, a nonlinear plan can be expressed as a well formed temporal formula.

Example 1 *In the blocks world, let P be a nonlinear plan consisting of two actions $Exec(Puton(A, T, B), t_1, t_1)$, which moves block A from the table T to the top of block B at the time point t_1, and $Exec(Puton(B, T, C), t_2, t_2)$, which moves block B from the table T to the top of block C at the time point t_2. In P, both time points t_1 and t_2 are after the initial time point T_0 and there is no direct temporal relation between t_1 and t_2. This nonlinear plan can be expressed as a well formed formula as follows.*

$$\begin{aligned} P \;=\; & Exec(Puton(A, T, B), t_1, t_1) \land T_0 \prec t_1 \land \\ & Exec(Puton(B, T, C), t_2, t_2) \land T_0 \prec t_2 \end{aligned}$$

In planning, a given problem domain can be described by a set of actions Γ and a set of domain constraints Σ_D. In our temporal logic, each action in Γ can be specified by a set of formulas in Horn clause form ((Zhang, 1994)). However, in this report, we only concern ourselves with the relationship between the actions and the domain constraints. To make the results in this report widely applicable, we use the STRIPS representation of actions to explore this relation. In STRIPS representation, an action consists of a set of preconditions, a set of deleted properties and a set of asserted properties.

$\mathcal{W} = \{\ t_i$: t_i *is a time point in* $P\} \cup \{T_0, T_\infty\}$ *where* T_0 *represents the initial time point and* T_∞ *represents the universal ending point.*

$\mathcal{R} = \{t_i \prec t_j : t_i \prec t_j$ *is a temporal relation of* $P\} \cup$
$\{T_0 \prec t_i, t_i \prec T_\infty$: *for every time point* t_i *in* $P.\}$

For every action A *in* P
$\pi_P(A) = \{(t_i, t_i)$: A *is executed at* t_i *in* $P.\}$

For every property p *which is either a precondition or a postcondition of an action in* P
$\pi_P(p) = \{(t_i, t_j)$: *if one of the following conditions holds.*

1. p *is an asserted property of an action executed at* t_i *in* P *and*
   (a) p *is a deleted property of an action at* t_j *in* P *and* $t_i \prec t_j$ *holds in* P, *and there is no other action which deletes* p *and is executed at time points between* t_i *and* t_j *in* P.
   (b) *Otherwise* $t_j = T_\infty$;
2. p *is a precondition of an action executed at* t_k *and both the following conditions hold*
   (a) p *is an asserted property of an action executed at* t_i *and* $t_i \prec t_k$ *and there is no other action which asserts* p *and is executed at the time points between* t_i *and* t_k *in* P. *Otherwise,* $t_i = T_0$.
   (b) p *is an deleted property of an actions executed at* t_j *and* $t_k \preccurlyeq t_j$ *and there is no other action which deletes* p *and is executed at time points between* t_k *and* t_j *in* P. *Otherwise,* $t_j = T_\infty.\}$

Let P be the nonlinear plan given in Example 1, then under the *closed world assumption* and the definition of $Puton(x, y, z)$ given in Figure 1, we can construct a nonlinear temporal model of P as shown in Fig. 2. In this figure, T_∞ represents the universal ending point. A bi-directed line of a property indicates a time interval over which the property hold true. For example, the property $On(B, T)$ holds true over the interval that is started from T_0 and is ended at t_2. One of the completions of this nonlinear plan, P', is the linear plan in which the action $Puton(A, T, B)$ is executed before $Puton(B, T, C)$.

$$\begin{aligned} P' &= Exec(Puton(A, T, B), t_1, t_1) \wedge T_0 \prec t_1 \wedge \\ &\quad Exec(Puton(B, T, C), t_2, t_2) \wedge T_0 \prec t_2 \wedge t_1 \prec t_2 \end{aligned}$$

This completion contains conflict. The action $Puton(A, T, B)$ destroies the precondition $Clear(B)$ of the action $Puton(B, T, C)$. Therefore, P is not conflict free. The conflict in P can be detected by the fact that the domain constraint $On(A, B) \leftrightsquigarrow Clear(B)$, i.e.,

$$Hold(On(A, B), t_1, T_\infty) \wedge Hold(Clear(B), T_0, T_\infty) \Longrightarrow T_\infty \preccurlyeq t_1 \vee T_\infty \preccurlyeq T_0$$

is not satisfied in the temporal model of P.

The Temporal Model of Nonlinear Plan Definition connects a nonlinear plan with a temporal model. Our purpose is to use domain constraint consistency checking in the nonlinear temporal model of a nonlinear plan to detect conflicts in the nonlinear plan. To achieve this, we need to explore further the internal connection between the set of domain constraints Σ_D and the set of actions Γ.

tion restriction, consistency restriction, inclusion restriction and minimum restriction.

3.1 Complement Restriction

Given a property p, we term all the properties in the problem domain which are mutually exclusive with p according to a set of domain constraints Σ_D the *complements of* p *according to* Σ_D. The *complement restriction* requires that if a property is deleted by an action, then at least one of its complements must be asserted by the action. This requirement is based on the understanding that the real world is "complete". That is, in the real world, for any property, at any moment, either the property holds true or one of its complements holds true. For example, the property that the room is empty is mutually exclusive with the property that there are some people in the room, with the property that there is a set of furniture in the room, with the property that there is a dog in the room, et cetera. At any time, either the room is empty or there is something in it. For the same reason, if a property is asserted by an action, then one of its complements must be deleted by the action.

Definition 2 (Complement Restriction) *Let Σ_D be a set of domain constraints and Γ be a set of actions. Γ is said to be* complementary for Σ_D *if the following conditions hold. For any property p, if p is an asserted (resp. deleted) property of an action A in Γ, then there exists at least one deleted (resp. asserted) property q of A such that $p \leftrightsquigarrow q \in \Sigma_D$.*

Example 3 *In the blocks world, let Γ consist of the action Puton(x,y,z) which is described in Fig. 1. Let Σ_D be a set domain constraints defined below.*

$$\begin{aligned}\Sigma_D \;=\; \{ & On(x,y) \leftrightsquigarrow On(x,y')|_{y \neq y'}, \\ & On(x,y) \leftrightsquigarrow On(x',y)|_{x \neq x' \wedge y \neq Table}, \\ & On(x,y) \leftrightsquigarrow Clear(y)|_{y \neq Table}, \\ & On(x,y) \leftrightsquigarrow On(y,x)\}\end{aligned}$$

One can easily check that Γ is complementary for Σ_D. For example, the two asserted properties $On(x,z)$ and $Clear(y)$ are mutually exclusive with one of the deleted properties $On(x,y)$ according to Σ_D. The Γ and Σ_D described above will be referred to several times when we discuss other restrictions in this section.

Example 4 *The action paint-ceiling(ceiling) given in (Yang, 1989) can be expressed as in Fig. 3. The property $Have(paint)$ may or may not be deleted by the action. Even if it is a deleted property of the action, it is still not considered to be mutually exclusive with the asserted property $Painted(ceiling)$. Therefore, there is not a deleted property which is mutually exclusive with the asserted property $Painted(ceiling)$. This definition of action is not complementary. In reality, no one would purposelessly repeat painting the ceiling forever. If one does need to paint the ceiling, he must have a reason such as the ceiling is unpainted, the ceiling is in an unpleasant colour, the ceiling has not been painted for over five years, et cetera. Any of the above properties should be mutually exclusive with the property $Painted(ceiling)$ according to a set of reasonable domain constraints. To prevent the action from being purposelessly repeated forever, at least one of them should be a precondition of the action and this precondition should be deleted by the action.*

Figure 3: I

3.2 Connection Res

The connection restriction is

erly attached to an action as

associating previous action c

Figure 4: The

Fig. 6 shows another un

tively, such kinds of action

Figure

4 Domain Constraint Maintenance

In the following discussion, whenever we state that a set of domain constraints Σ_D and a set of actions Γ satisfy the restrictions, we mean that Γ is connective, consistent, inclusive and minimum for Σ_D.

Given a nonlinear plan, we can build a nonlinear temporal model according to the corresponding set of action definitions. We suggest that the possible conflicts in a nonlinear plan can be detected by checking whether the corresponding set of domain constraints is satisfied by the temporal model of the plan. We term this conflict detecting method for nonlinear plans *domain constraint maintenance*. In this section, we will prove that domain constraint maintenance is sound and complete.

4.1 Domain Constraints via Necessary-Biggest-Interval Axioms

In our temporal logic for nonlinear planning, properties are asserted over the necessary biggest possible time intervals. That is, if a property is asserted over an interval in an interpretation, then the interval should be the biggest possible for the property in all the completions of the interpretation[3]. This requirement can be captured by the following temporal formula.

$$\forall t_1, t_2, t_3, t_4.$$
$$(Hold(p, t_1, t_2) \wedge Hold(p, t_3, t_4) \Longrightarrow t_2 \prec t_3 \vee t_4 \prec t_1 \vee (t_1 = t_3 \wedge t_2 = t_4))$$

This formula states that, for any two intervals of the same property p, either they are equal or one of them is strictly before the other. We term such a formula *necessary-biggest-interval* axiom and we often use Σ_B to represent the set of necessary-biggest-interval axioms in a problem domain.

We will prove that if Σ_B is not satisfied by the temporal model of a nonlinear plan, then the nonlinear plan contains conflicts. This means that Σ_B is a necessary condition for conflict freeness of nonlinear plans. However, Σ_B is not a sufficient condition. For example, if two different actions that have one common deleted property are executed at exactly the same time point in a nonlinear plan, then Σ_B may still be satisfied in the temporal model of this plan. However, this nonlinear plan obviously contains a deleted condition conflict.

The set of domain constraints Σ_D is stronger than Σ_B. We will show that Σ_D is necessary and sufficient for conflict freeness of nonlinear plans and Σ_D implies Σ_B. The necessary-biggest-interval axiom schema in our temporal logic corresponds to the *main proposition constraint* in Allen's interval temporal logic (Allen & Koomen, 1983; Allen, 1991b). Allen has already used a set of *domain specific axioms* (i.e. domain constraints) to replace the main proposition constraint in his planning system. To quote Allen:

> In general, we will not use this constraint[4], but use domain-specific axioms for predicates that are mutually exclusive. For instance, in the blocks world, every block is always either clear, being held, or has another block on top of it (Allen, 1991b).

[3]If an interpretation $\mathcal{M}$ is the tuple $\langle \mathcal{W}, \mathcal{R}, \mathcal{D}, \pi_F, \pi_P \rangle$, then $\mathcal{M}' = \langle \mathcal{W}, \mathcal{C}(\mathcal{R}), \mathcal{D}, \pi_F, \pi_P \rangle$ is a *completion* of $\mathcal{M}$, where $\mathcal{C}(\mathcal{R})$ is a total order relation over the time points in $\mathcal{W}$ such that $\mathcal{R} \subseteq \mathcal{C}(\mathcal{R})$.

[4]Here, Allen refers to the main proposition constraint which is corresponding to Σ_B in our temporal logic.

However, Allen did not explain why this replacement can be made and he did not give the conditions under which the replacement can be made. We are going to deal with these problems in this section. Before we prove the main theorem for these problems, we give a useful proposition first. In fact, this proposition states the minimum restriction in a different way and it is straightforward from the minimum restriction.

Proposition 2 *Let Σ_D be a set of domain constraints and Γ be a set of actions such that Σ_D and Γ satisfy the restrictions. For any property p, if p is an asserted property of two different actions A_1 and A_2 in Γ or p is a deleted property of both the actions, then*

either *at least, one of the deleted properties of A_1 is mutually exclusive with one of the deleted properties of A_2 according to Σ_D*

or *at least, one of the asserted properties of A_1 is mutually exclusive with one of the asserted properties of A_2.*

The following theorem states that, under certain conditions, Σ_D implies Σ_B.

Theorem 1 *Let Σ_D be a set of domain constraints and Γ be a set of actions such that Σ_D and Γ satisfy the restrictions. Suppose that P is a nonlinear plan and $\mathcal{M}$ is the nonlinear temporal model of P. If $\mathcal{M}$ satisfies Σ_D, then $\mathcal{M}$ satisfies Σ_B.*

Proof Let p be any property. We are going to prove that if $\mathcal{M}$ satisfies Σ_D, then $\mathcal{M}$ satisfies the formula

$$Hold(p, t_1, t_2) \wedge Hold(p, t_3, t_4) \Longrightarrow t_2 \prec t_3 \vee t_4 \prec t_1 \vee (t_1 = t_3 \wedge t_2 = t_4)$$

We prove this by induction on the number of the intervals over which p holds in the model.

When the number of the intervals of p equals one, the theorem is trivially satisfied. Suppose that the theorem holds when the number equals $n-1$. Now, we prove that the theorem holds true when the number equals n.

Suppose that the nth interval of p is asserted by action A_1 at t_n and finished by action A_2 at t'_n[5]. By induction hypothesis, the other $n-1$ intervals of p, $(t_1, t'_1), \ldots, (t_{n-1}, t'_{n-1})$, must be in the linearity relation. By the complement restriction, there must be a deleted property q of A_1 and an asserted property r of A_2 such that $p \leftrightsquigarrow q \in \Sigma_D$ and $p \leftrightsquigarrow r \in \Sigma_D$. By the assumption that Σ_D is satisfied by the model, the intervals of q and r must be in the linearity relation with the other $n-1$ intervals of p. Without loss of generality, we can assume that the interval of q is between the ith and (i+1)th intervals of p, see Fig. 11.

Then there are only two possibilities about the position of the interval (t'_n, t_l) of r.

Case 1. The interval of r is before the $(i+1)th$ interval of p. That is the temporal relation $t_l \preccurlyeq t_{i+1}$ is satisfied in the model. In this case, the nth interval of p, (t_n, t'_n) is in the linearity relation with the other $n-1$ intervals of p.

[5] We can choose the nth interval (t_n, t'_n) such that if $t_n = T_0$, then $t'_n \neq T_\infty$ and if $t'_n = T_\infty$ then $t_n \neq T_0$. Otherwise, there is only one interval of p.

in the model. Therefore, the necessary biggest interval axiom for $Clear(B)$ is not satisfied. According to the above theorem, this implies that the set of domain constraints Σ_D is not satisfied in the model. In fact, it is the domain constraint $On(A, B) \leftrightsquigarrow Clear(B)$ that is not satisfied in the model.

In this section, we have proved that Σ_D implies Σ_B. However, the reverse relation does not exist.

4.2 Soundness of Domain Constraint Maintenance

In this section, we are going to prove the soundness of domain constraint maintenance. Domain constraint maintenance detects conflict in a nonlinear plan by checking whether the temporal model of the nonlinear plan satisfies the corresponding set of domain constraints. Let P be an arbitrary nonlinear plan, $\mathcal{M}$ be its temporal model and Σ_D be the corresponding set of domain constraints. Domain constraint maintenance is said to be sound if the fact that $\mathcal{M}$ does not satisfy Σ_D implies that P contains conflicts.

Theorem 2 (Soundness of Domain Constraint Maintenance) *Let Σ_D be a set of domain constraints and Γ be a set of actions such that Σ_D and Γ satisfy the restrictions. Let P be a nonlinear plan that consists of the actions in Γ and $\mathcal{M}$ be the nonlinear temporal model of P. If P is conflict free, then $\mathcal{M}$ satisfies Σ_D.*

The proof of this theorem is given in Appendix B.

We have mentioned that Σ_B is a necessary condition for conflict freeness for nonlinear plans. This can be easily proved from the above theorem.

Theorem 3 *Let Σ_D be a set of domain constraints and Γ be a set of actions such that Σ_D and Γ satisfy the restrictions. Let P be a nonlinear plan that consists of the actions in Γ, $\mathcal{M}$ be the nonlinear temporal model of P and Σ_B be the set of necessary-biggest-interval axioms. If P is conflict free, then $\mathcal{M}$ satisfies Σ_B.*

Proof By Theorem 2, $\mathcal{M}$ satisfies Σ_D. By Theorem 1, this theorem is proved. ■

4.3 Completeness of Domain Constraint Maintenance

In this section, we will prove the completeness of domain constraint maintenance. Let P be an arbitrary nonlinear plan, $\mathcal{M}$ be its temporal model and Σ_D be the corresponding set of domain constraints. Domain constraint maintenance is said to be complete if the fact that P contains conflicts implies that $\mathcal{M}$ does not satisfy Σ_D.

Theorem 4 (Completeness of Domain Constraint Maintenance) *Let Σ_D be a set of domain constraints and Γ be a set of actions such that Σ_D and Γ satisfy the restrictions. Let P be a nonlinear plan that consists of the actions in Γ and $\mathcal{M}$ be the nonlinear temporal model of P. If $\mathcal{M}$ satisfies Σ_D, then P is conflict free.*

5 Discussion

In this report, we propose a set of domain independent and human independent restrictions on actions. These restrictions are not hard. The domain constraints and the actions in most problem domains can satisfy these restrictions or can be reasonably modified to satisfy these restrictions. We show that these restrictions can prevent actions from being improperly defined. These restrictions also lay down the basis for domain constraint maintenance. We believe that the basic principles of these restrictions can also be applied in automated program synthesis to regulate program statements and in automated circuit design to regulate circuit units. It is desirable to automatically check if a given set of domain constraints and a set of actions satisfy the restrictions proposed in this report. For a limited set of domain constraints and a limited set of actions, the cost of the checking is not prohibitive and the checking is needed only once for a specific problem domain.

For a given finite nonlinear plan P, an algorithm of domain constraint maintenance always stops in a limited time. If the nonlinear plan contains conflicts, the algorithm will find and report the conflicts. Otherwise, the algorithm shows that the plan is conflict free.

Given a nonlinear plan P, an algorithm for domain constraint maintenance needs to check a domain constraint for each of the $|P|$ actions in the plan, for each of the k preconditions, p, of the action, for each of the l properties, q, that are mutually exclusive with p, for each of the m ($< |P|$) appearances of q in the model of P. Therefore, such an algorithm has the worst case time complexity of $O(|P|^2)$. Considering the $O(|P|^2)$ worst case time complexity needed for the construction of the temporal model of P, the domain constraint maintenance has the worst case time complexity of $O(|P|^4)$. Clearly, domain constraint maintenance is much more efficient than detecting conflicts of a nonlinear plan by checking all its possible completions. For checking every possible completions of a nonlinear plan P, the worst case time complexity is $O(|P|! \times |P|^3 \times n^m)$ where m is number of variables in P and n is the average number of possible values for each variable.

Although not explicitly pointed out, the main theoretical results in this report are achieved under the *closed world assumption*. This is because that, whenever we mention a temporal model of a nonlinear plan, from the corresponding definition, we assume that all the properties in the model are only affected by the actions in the plan.

In this report, we only concern ourselves with point actions. The corresponding results for interval actions are given in (Zhang, 1994).

6 Acknowledgements

We would like to thank Behnam Bani-Eqbal, Clare Dixon, Ian Pratt and Rajeev Goré who read an early version of this paper and gave valuable comments.

This paper is a modified version of part of Zhang's Ph.D. thesis (Zhang, 1994). The work was partially supported by the British Council and the State Education Commission of China.

A A Temporal Logic for Nonlinear Planning

A.1 Syntax

Given

DV: a set of data variables,

TV: a set of temporal variables which is disjoint from DV,

DF: a set of data function symbols,

TF: a set of temporal function symbols which is disjoint from DF,

$\mathcal{A}$: a set of primitive action symbols,

$\mathcal{P}$: the set of primitive property symbols which is disjoint from $\mathcal{A}$,

the set of data terms, T_D, is defined recursively as follows

1. if $t \in DV$, then $t \in T_D$,
2. if $f \in DF$, $arity(f) = n$ and $t_1, \ldots, t_n \in T_D$, then $f(t_1, \ldots, t_n) \in T_D$.

The definition of the set of *temporal terms*, T_T, is similar to that of data terms except that the arguments of temporal functions are not restricted to be temporal terms. The set of temporal terms, T_T, is defined as follows:

1. if $t \in TV$, then $t \in T_T$,
2. if $f \in TF$, $arity(f) = n$, and $t_1, \ldots, t_n \in T_T \bigcup T_D$, then $f(t_1, \ldots, t_n) \in T_T$.

The set of *well-formed formulas(wffs)* is defined inductively as follows:

1. If $dt_1, \ldots, dt_n \in T_D$, $tt_1, tt_2 \in T_T$, $P \in \mathcal{P}$, $arity(P) = n$, then $Hold(P(dt_1, \ldots, dt_n), tt_1, tt_2)$ is a wff,
2. if $dt_1, \ldots, dt_n \in T_D$, $tt_1, tt_2 \in T_T$, $A \in \mathcal{A}$, $arity(A) = n$, then $Exec(A(dt_1, \ldots, dt_n), tt_1, tt_2)$ is a wff,
3. if $tt_1, tt_2 \in T_T$, then $tt_1 \prec tt_2$ and $tt_1 \preccurlyeq tt_2$ are wffs,
4. if φ and ψ are wffs, then so are $\neg\varphi$, $\varphi \wedge \psi$,
5. if φ is a wff and $v \in DV \bigcup TV$ is a free variable of φ, then $\forall v.\varphi[v]$ is a wff.

We assume the usual definitions of $\vee$, $\Longrightarrow$, $\Longleftrightarrow$ and $\exists$.

A.2 Semantics

Definition 8 *A* partial order time structure *is a pair* $(\mathcal{W}, \mathcal{R})$ *where* $\mathcal{W}$ *is a non-empty universe of time points and* $\mathcal{R}$ *is a* partial order relation *between the time points in* $\mathcal{W}$.

Definition 9 *An* interpretation *of a temporal language is a tuple :* $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{D}, \pi_F, \pi_P \rangle$ *where*

$(\mathcal{W}, \mathcal{R})$ *is a partial order time structure,*

$\mathcal{D}$ *is the domain of* data *for all the time points in* $\mathcal{W}$.

π_F *is a interpretation of the function symbols. If* $f \in DF \bigcup TF$ *and* $arity(f) = n$, *then* $\pi_F(f) \in (\mathcal{D} \bigcup \mathcal{W})^n \longrightarrow \mathcal{D} \bigcup \mathcal{W}$.

π_P *is a interpretation of the proposition types (actions and properties). If* $P \in \mathcal{P} \bigcup \mathcal{A}$ *and* $arity(P) = n$, *then* $\pi_P(P) \in \mathcal{D}^n \longrightarrow 2^{\mathcal{W} \times \mathcal{W}}$. *We require that if* $(t_1, t_2) \in \pi_P(P)(dt_1, \ldots, dt_n)$, *then* $(t_1, t_2) \in \mathcal{R}$. *To characterize that assertions are interpreted over the necessarily biggest possible intervals,* π_P *has the property that, if* $(t_1, t_2), (t_3, t_4) \in \pi_P(P)(dt_1, \ldots, dt_n)$ *and* (t_1, t_2) *and* (t_3, t_4) *do not refer to the same time interval, then either* $(t_2, t_3) \in \mathcal{R}$ *or* $(t_4, t_1) \in \mathcal{R}$.

A *variable assignment* ν for a given interpretation $\mathcal{M}$, is a mapping from variable symbols to the elements of the domains, $\nu : (DV \bigcup TV) \longrightarrow (\mathcal{D} \bigcup \mathcal{W})$ which satisfies sort restriction, i.e. $\nu(x) \in \mathcal{D}$ if $x \in DV$ and $\nu(x) \in \mathcal{W}$ if $x \in TV$.

For a given interpretation $\mathcal{M}$ and variable assignment ν, a *term assignment* τ is a mapping, $\tau : (T_D \bigcup T_T) \longrightarrow (\mathcal{D} \bigcup \mathcal{W})$ which can be defined as follows.

1. If $t \in DV \bigcup TV$, then $\tau(t) = \nu(t)$,
2. if $f \in DF \bigcup TF$ and $arity(f) = n$, then $\tau(f(t_1, \ldots, t_n)) = \pi_F(f)(\tau(t_1), \ldots, \tau(t_n))$.

The semantics of the wffs is given with respect to an interpretation $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{D}, \pi_F, \pi_P, \rangle$ and a variable assignment ν. The *satisfaction relation* ($\models$)of the well formed formulas under such a pair is defined recursively as follows.

$$
\begin{array}{lcl}
\langle \mathcal{M}, \nu \rangle & \models & Hold(P(dt_1, \ldots, dt_n), tt_1, tt_2) \text{ iff} \\
 & & (\tau(tt_1), \tau(tt_2)) \in \pi_P(P)(\tau(t_1), \ldots \tau(t_n)) \\
\langle \mathcal{M}, \nu \rangle & \models & Exec(A(dt_1, \ldots, dt_n), tt_1, tt_2) \\
 & & \text{ iff } (\tau(tt_1), \tau(tt_2)) \in \pi_P(A)(\tau(t_1), \ldots \tau(t_n)) \\
\langle \mathcal{M}, \nu \rangle & \models & t_1 \prec t_2 \text{ iff } (\tau(t_1), \tau(t_2)) \in \mathcal{R} \\
\langle \mathcal{M}, \nu \rangle & \models & t_1 \preccurlyeq t_2 \text{ iff } (\tau(t_1), \tau(t_2)) \in \mathcal{R} \text{ or } \tau(t_1) = \tau(t_2) \\
\langle \mathcal{M}, \nu \rangle & \models & \neg\varphi \text{ iff } \langle \mathcal{M}, \nu \rangle \models \varphi \text{ is false} \\
\langle \mathcal{M}, \nu \rangle & \models & (\varphi \wedge \psi) \text{ iff } \langle \mathcal{M}, \nu \rangle \models \varphi \text{ and } \langle \mathcal{M}, \nu \rangle \models \psi \\
\langle \mathcal{M}, \nu \rangle & \models & \forall v.\varphi \text{ iff } \langle \mathcal{M}, \nu' \rangle \models \varphi \text{ for all } \nu' \text{ which} \\
 & & \text{agree with } \nu \text{ except possibly on } v.
\end{array}
$$

An interpretation $\mathcal{M}$ is a *model* for a wff φ (written as $\mathcal{M} \models \varphi$) if $\langle \mathcal{M}, \nu \rangle \models \varphi$ for all variable assignments ν. A wff is a *closed formula* if it contains no free variables. If a closed formula φ is satisfied by a interpretation $\mathcal{M}$ and a variable assignment, then $\mathcal{M}$ is a model of φ. A wff φ is said to be *satisfiable* if it has a model. A wff φ is said to be *valid* (written as $\models \varphi$) if its negation is not satisfiable.

B Soundness of Domain Constraint Maintenance

Theorem 5 (Soundness of Domain Constraint Maintenance) *Let Σ_D be a set of domain constraints and Γ be a set of actions such that Σ_D and Γ satisfy the restrictions. Let P be a nonlinear plan that consists of the actions in Γ and $\mathcal{M}$ be the nonlinear temporal model of P. If P is conflict free, then $\mathcal{M}$ satisfies Σ_D.*

Proof We prove that if Σ_D is not satisfied by the model, then there exist deleted condition conflicts in at least one of the completions of P.

Let (t_i, t'_i) over which p holds true and (t_j, t'_j) over which q holds be the *pair of intervals which is the nearest to the initial time point T_0 such that $p \leftrightsquigarrow q \in \Sigma_D$* and neither $t'_i \preccurlyeq t_j$ nor $t'_j \preccurlyeq t_i$ is satisfied in the model. By Definition 1, for any interval of any property in the model, either it is asserted by an action or it is a precondition of an action. Then there are three possibilities.

Case 1. Neither of the intervals of p and q are asserted by an action. That is $t_i = t_j = T_0$. In this case, by Definition 1 (in page 3) and by the consistency restriction, they must be preconditions of two different actions in P and there are no other actions to reassert p and q for these two actions. Then one of them should be "deleted" by the requirement that the initial conditions for any problem should be consistent. A deleted condition conflict in P occurs.

Case 2. Neither of the intervals of p and q is a precondition of an action in the model. By Definition 1, $t'_i = t'_j = T_\infty$. By the consistency restriction, the two intervals of p and q must be asserted by two different actions, say A_i at t_i and A_j at t_j respectively. According to the temporal relation between t_i and t_j, there are three possible cases.

Subcase 1. Neither $t_i \preccurlyeq t_j$ nor $t_j \preccurlyeq t_i$ holds in the model. See Fig. 14.

By the inclusion restriction, A_i must have a precondition r such that either $r \leftrightsquigarrow q \in \Sigma_D$ or $r = q$ and r is a deleted property of A_i. By the nearest to the initial point assumption, $r = q$ and r is a deleted property of A_i. Otherwise, the pair of the intervals for r and q is nearer to the initial point than that for p and q. For the same reason, A_j has a deleted property r' such that $r' = p$. Therefore $r \leftrightsquigarrow r' \in \Sigma_D$. This contradicts the nearest to the initial point assumption for the intervals (t_i, t'_i) of p and (t_j, t'_j) of q. So this case is not possible in any model.

Subcase 2. $t_i \prec t_j$ or $t_j \prec t_i$ holds true in the model. Suppose $t_i \prec t_j$, see Fig. 15.

For the same reason given in the subcase 1, $r' = p$. In this case, $t_i \prec t_j$ and the action A_j deletes p. By Definition 1, the interval (t_i, t'_i) of p should not exist in the model. Therefore, this is also an impossible case.

Figure 19: Part

Tenenberg, Josh D. 1991. Abstraction in Planning. *Pages 213–280 of:* Allen, J., Kautz, H., Pelavin, R., & Tenenberg, J. (eds), *Reasoning about Plans*. Morgan Kaufmann Publishers, Inc.

Warren, D. H. D. 1974. *WARPLAN: a System for Generating Plans*. Tech. rept. Department of Computational Logic Meno No. 76. University of Edinburgh, Edinburgh.

Yang, Q. 1989. *Improving the Efficiency of Planning*. Ph.D. thesis, Univ. of Maryland, Computer Science Dept.

Zhang, Ying. 1994 (May). *Constraint Logic Planning: A Formal Framework for Nonlinear Planning*. Ph.D. thesis, Manchester Univ., Computer Science Dept.