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:

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

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.*

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

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.

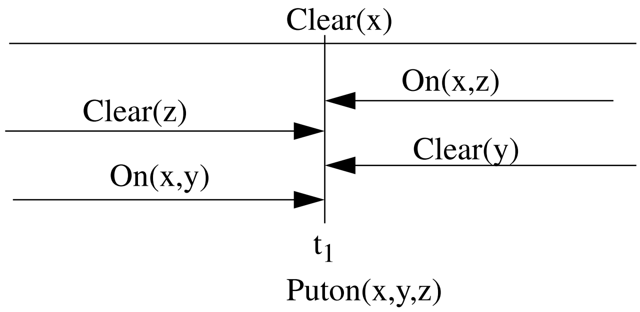
Puton(x,y,z):

Precondtions: {Clear(x), Clear(z), On(x,y)}

Deleted properties: {On(x,y), Clear(z)}

Asserted properties: {On(x,z), Clear(y)}

a. The action Puton(x,y,z)



b.The temporal structure of the point action Puton(x,y,z).

Figure 1: The point action Puton(x,y,z).

Example 2 *In the blocks world, the action of stacking blocks can be described in Fig. 1a*[1]*. The temporal relations for this action are shown in Fig. 1b.*

An asserted property of an action is also called a postcondition of the action and a precondition which is not a deleted property of the action is also termed a postcondition.

In our temporal logic, syntactically, a nonlinear plan can be expressed as a well formed temporal formula. Semantically, a nonlinear plan, together with the corresponding definitions of actions, describes a temporal model. A nonlinear plan consists of a set of time points at which some actions are executed and a set of temporal relations among the time points. Given a nonlinear plan and a set of actions Γ, based on the *closed world assumption*, a temporal model can be built such that the specification of actions in Γ are satisfied by the model.

Definition 1 (Temporal Model of Nonlinear Plan) *Given a nonlinear plan P which consists of a set of time points at which some actions in Γ are executed and a set of temporal relations among the time points. Then $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{D}, \pi_F, \pi_P \rangle$ is said to be* the temporal model of P according to Γ *if it is constructed as follows.*

[1]In this definition, x is not equal to the table. If y is equal to the table, then $Clear(y)$ is not an asserted property. If z is equal to the table, then $Clear(z)$ is not a deleted property.

$\mathcal{W}$ = { 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.

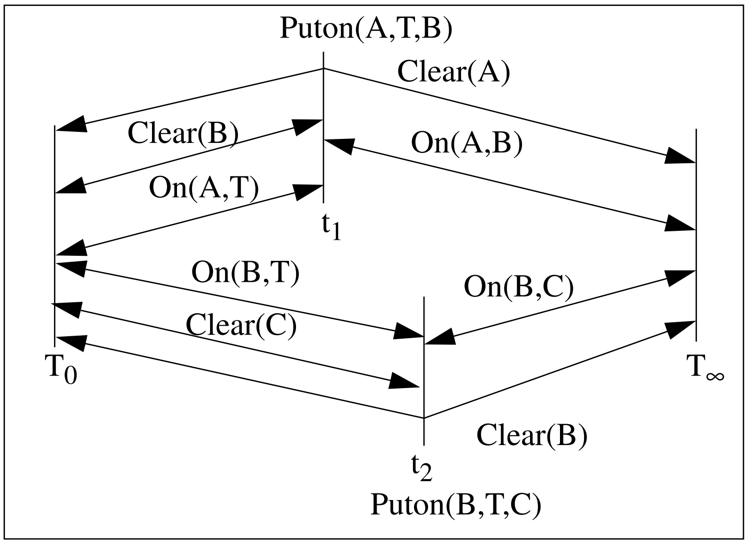


Figure 2: The temporal model of a nonlinear plan which contains conflict.

3 Restrictions on Actions

Generally, given a problem domain, the set of domain constraints is a set of property pairs which are forbidden to co-exist. This explanation does not prevent one from arbitrarily defining a set of property pairs in a given problem domain as the domain constraints. For example, in the blocks world, one can require that the properties $On(A, B)$ and $On(C, D)$, are mutually exclusive for some special reason. However, such kind of domain constraints are never considered in the planning literature. In planning, only "reasonable" domain constraints are considered.

Informally, a domain constraint is reasonable if the two properties of the constraint do not hold true together at any state that is obtainable from an execution of a conflict free plan. This requirement implies that the definitions of the set of reasonable domain constraints should mutually depend on the definitions of the actions in the problem domain. For example, in the blocks world, if the action $Puton(x, y, z)$ is defined such that $Clear(z)$ is one of its preconditions and one of its deleted properties, then the property pair $(On(x, z), On(x', z))$ should be a domain constraint. However, if the action is defined such that the $Clear(z)$ is not one of its preconditions and deleted properties, then $(On(x, z), On(x', z))$ should not be a domain constraint. This happens when the blocks considered are of different size or z is the table.

The set of reasonable domain constraints is mutually dependent upon the set of reasonable actions. To formally specify this mutually dependent relation, one may add restrictions on actions based on domain constraints or vice versa. In this section, we propose a set of restrictions on actions based on the set of domain constraints in the same problem domain. These restrictions formally specify the mutually dependent relation between reasonable domain constraints and actions. We show that the set of restrictions can prevent actions from being improperly defined. In (Zhang, 1994), we give a set of restrictions on domain constraints based on actions.

The restrictions that are presented in this section are complement restriction, connection 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.*

paint-ceiling(ceiling):
P: {Have(ladder), Have(paint),Climbable(ladder)}
D: {...}
A: {Painted(ceiling),...}

Figure 3: The definition of the action paint-ceiling.

3.2 Connection Restriction

The connection restriction is used to prevent irrelevant properties from being improperly attached to an action as part of its preconditions or postconditions. For example, associating previous action definitions, the property $Painted(ceiling)$ should not be an asserted property of the action $Puton(x, y, z)$ in any circumstances. For any action, all its deleted properties and all its asserted properties should have some internal connections amongst them. These connections can be captured by the mutually exclusive relations according to a set of reasonable domain constraints.

Definition 3 *Given a set of domain constraints Σ_D and a set of properties $\mathcal{S}$, two properties $p, q \in \mathcal{S}$ are said to be* connective under Σ_D within $\mathcal{S}$ *if $p \leftrightsquigarrow q \in \Sigma_D$ or there is a property $r \in \mathcal{S}$ such that $p \leftrightsquigarrow r \in \Sigma_D$ and r and q are connective under Σ_D within $\mathcal{S}$. If all the properties in the set $\mathcal{S}$ are connective under Σ_D within $\mathcal{S}$, then $\mathcal{S}$ is said to be connective under Σ_D.*

It should be noticed that the fact that two properties are connective under a set of domain constraints does not imply that the two properties are mutually exclusive. For example, let the Σ_D be the set of domain constraints defined in Example 3. Then the two domain constraints $On(A, B) \leftrightsquigarrow On(A, C)$ and $On(A, C) \leftrightsquigarrow Clear(C)$ are in Σ_D. According to the above definition, $On(A, B)$ and $Clear(C)$ are connective under Σ_D within a set of properties $\mathcal{S}$ that contains the three properties. However, the two properties are not mutually exclusive under Σ_D.

Definition 4 (Connection Restriction) *Given a set of domain constraints Σ_D, an action is said to be* connective under Σ_D *if the set of all the deleted properties and all the asserted properties of the action is connective under Σ_D. If all the actions in a set Γ are connective under Σ_D, then Γ is said to be connective under Σ_D.*

For example, let Γ be the set of actions and Σ_D be the set of domain constraints that are defined in Example 3. Then Γ is connective under Σ_D and the connective relation of the action is shown in Fig. 4.

Example 5 *Suppose A is an action defined in Fig. 5a. The connective relations under Σ_D, defined in Example 3, between the deleted properties and the asserted properties of the action are shown in Fig. 5b. In the figure, the deleted property $On(D, E)$ is not connective to the other deleted and asserted properties, so the action is not connective. This action is also not complementary for the Σ_D. Intuitively, $On(D, E)$ is irrelevant to the other properties of the action and should not be attached to this action.*

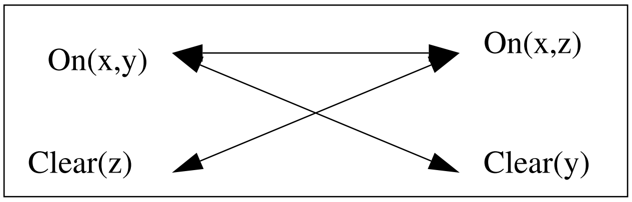


Figure 4: The connective relation of action Puton(x,y,z).

Fig. 6 shows another unconnected action, which is, however, complementary. Intuitively, such kinds of action should be replaced by two or more irrelevant sub-actions. The action in Fig. 6 should be replaced by two irrelevant, complementary and connective actions $Puton(A, B, C)$ and $Unstack(D, E)$[2]*.*

3.3 Consistency Restriction

The consistency restriction requires that for any action, all its preconditions and all its postconditions should be consistent respectively. The necessity of this restriction is obvious and intuitive and the actions in any problem domain should satisfy this requirement.

Definition 5 (Consistency Restriction) *Given a set of domain constraints Σ_D, an action A is said to be* consistent with Σ_D *if for any two preconditions p and q of A, $p \leftrightsquigarrow q \notin \Sigma_D$ and for any two postconditions r and s of A, $r \leftrightsquigarrow s \notin \Sigma_D$. If all the actions in a set Γ are consistent with Σ_D, then Γ is said to be consistent with Σ_D.*

With the introduction of the connection restriction and consistency restriction, the complement restriction becomes redundant. Though the complement restriction itself is not necessary, the discussion of its introduction is helpful for understanding the relationship between the set of actions and the set of domain constraints in the same problem domain.

Proposition 1 *Given a set of actions Γ and a set of domain constraints Σ_D, if Γ is connective and consistent according to Σ_D, Γ is complementary according to Σ_D.*

Proof: For any property p and any action $A \in \Gamma$, if p is an asserted (a deleted) property of A, by connective restriction, it must be mutually exclusive to a property q in the set of all the deleted and all the asserted properties of A. By the consistency restriction, q must be a deleted (an asserted) property of A. ■

[2]The action $Unstack(x, y)$ moves block x from the top of block y to the table. The action will be discussed later in this section.

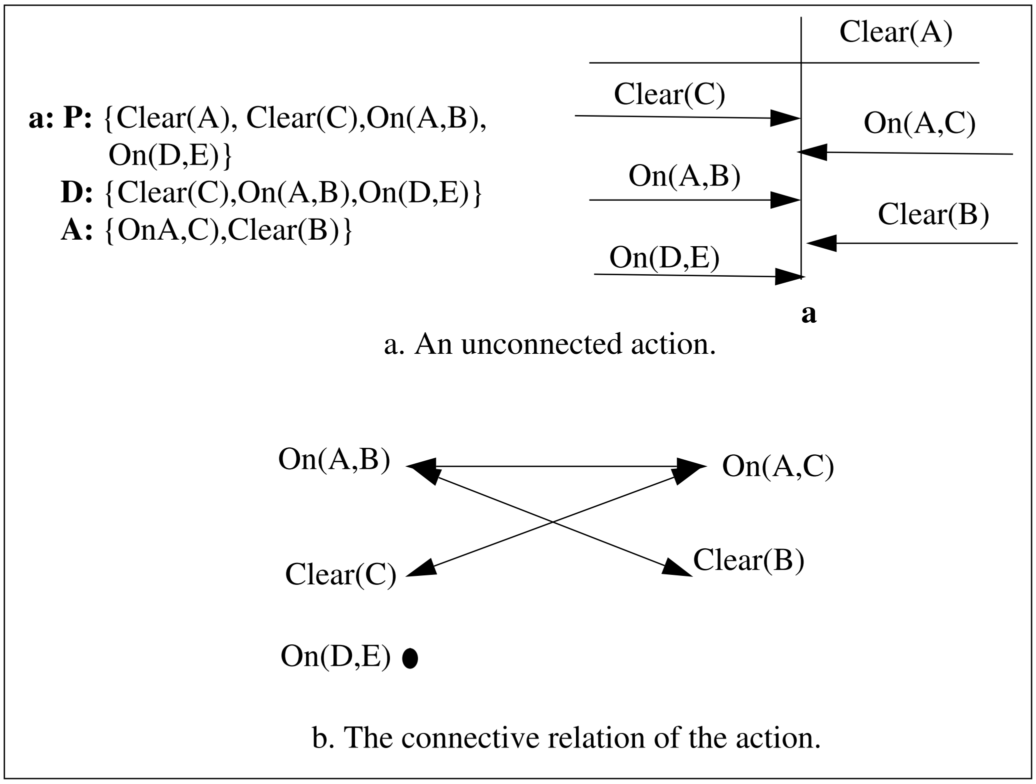


Figure 5: A unconnected and noncomplementary action.

3.4 Inclusion Restriction

The inclusion restriction can be regarded as the counterpart of the connection restriction. The connection restriction requires that if a deleted or an asserted property p of an action is not connective to the other asserted and deleted properties of the action under the domain constraints Σ_D, then p should not be attached to the action. In contrast, the inclusion restriction requires that all the related properties of an action, according to the set of domain constraints, should be attached to it either as deleted properties or as asserted properties as long as the consistency restriction and connective restriction are satisfied.

Definition 6 (Inclusion Restriction) *Given a set of domain constraints Σ_D, an action A is said to be* inclusive according to Σ_D *if the following condition hold. For every asserted (resp. deleted) property p of A, if q is a property such that $p \leftrightsquigarrow q \in \Sigma_D$, then either q is a deleted (resp. asserted) property of A or q is mutually exclusive with a precondition (resp. postcondition) of A according to Σ_D. If all the actions in an action set Γ are inclusive according to Σ_D, then Γ is said to be inclusive according to Σ_D.*

According to this restriction, the set of the deleted properties and the set of the asserted properties of an action should include as many properties as possible as long as the connection and consistency restrictions are satisfied.

Let the action $Puton(x, y, z)$ and the set of domain constraints be the ones that are defined in Example 3. It is easy to check that the action $Puton(x, y, z)$ is inclusive

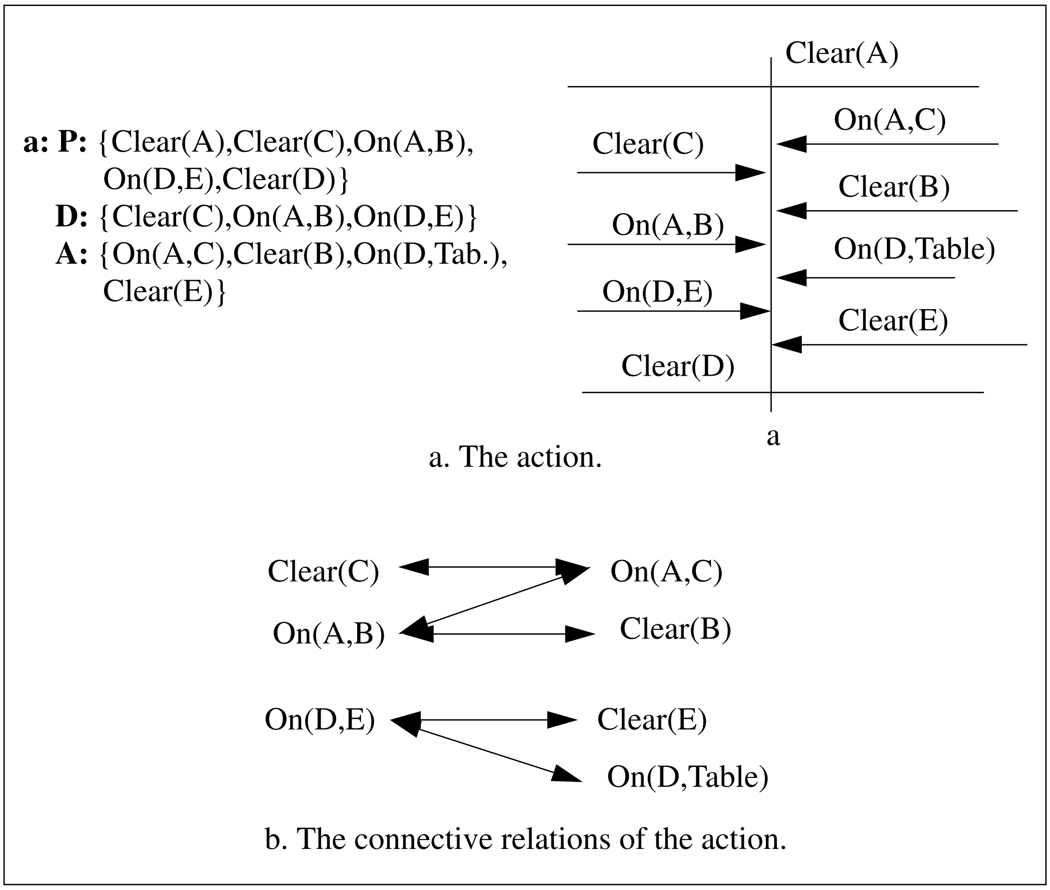


Figure 6: A complementary, unconnected action.

according to Σ_D. However, if we remove $Clear(y)$ from the asserted properties or remove $Clear(z)$ from the deleted properties, the action will not be inclusive according to Σ_D.

Example 6 *In the blocks world, we add a new action: $Unstack(x, y)$, which moves the block x from the top of block y down on to the Table. This action is described in Fig. 7. This action is also inclusive according to the Σ_D given in Example 3, though it has fewer properties than the action $Puton(x, y, z)$ does. For example, the property $Clear(Table)$ should not be added as a deleted property to $Unstack(x, y)$ for it is not mutually exclusive with either of the asserted properties. Although the property $On(y, x)$ is mutually exclusive with the deleted property $On(x, y)$, it cannot be added as an asserted property as it is also mutually exclusive with the postcondition $Clear(x)$.*

Example 7 *Suppose that $Swap(x, y)$ is the action defined in Fig. 8a. This action tries to put block x on block y directly from the state that y is on x. This definition is complementary, consistent and connective for the set of domain constraints Σ_D defined in Example 3. However, it is not inclusive. Not all the properties that are mutually exclusive with the asserted property $On(x, y)$ and are not mutually exclusive with a precondition of the action are attached to the action and neither are the properties that are mutually*

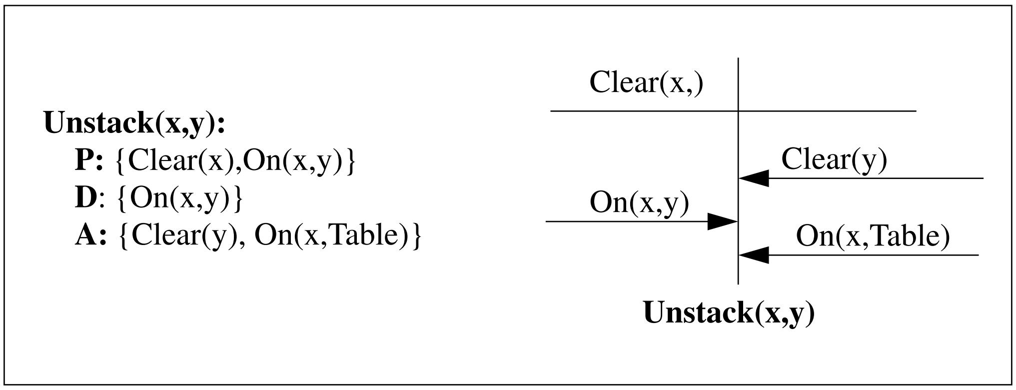


Figure 7: The action Unstack.

exclusive with the deleted property $On(y,x)$. *The problem is that more than one action schemata fit this specification. For example, after* $Swap(x,y)$, *the block* y *may be on exactly where* x *was or* y *may be on the top of some other block that was clear before the action. Also, the block* x *must be either on the table or on some other block before the action. Two possible situations suitable for the definition is shown in Fig. 8b. However, the definition is suitable for neither of the situations. For example, the definition fails to describe the change of the property* $On(A,C)$ *in both the situations. Therefore, this action is improperly defined. Fig. 9 shows one possible modification* $Swap(x,y,z)$, *which puts the block* y *exactly where* x *was. This modified action is inclusive according to the* Σ_D *and uniquely specifies one action schemata.*

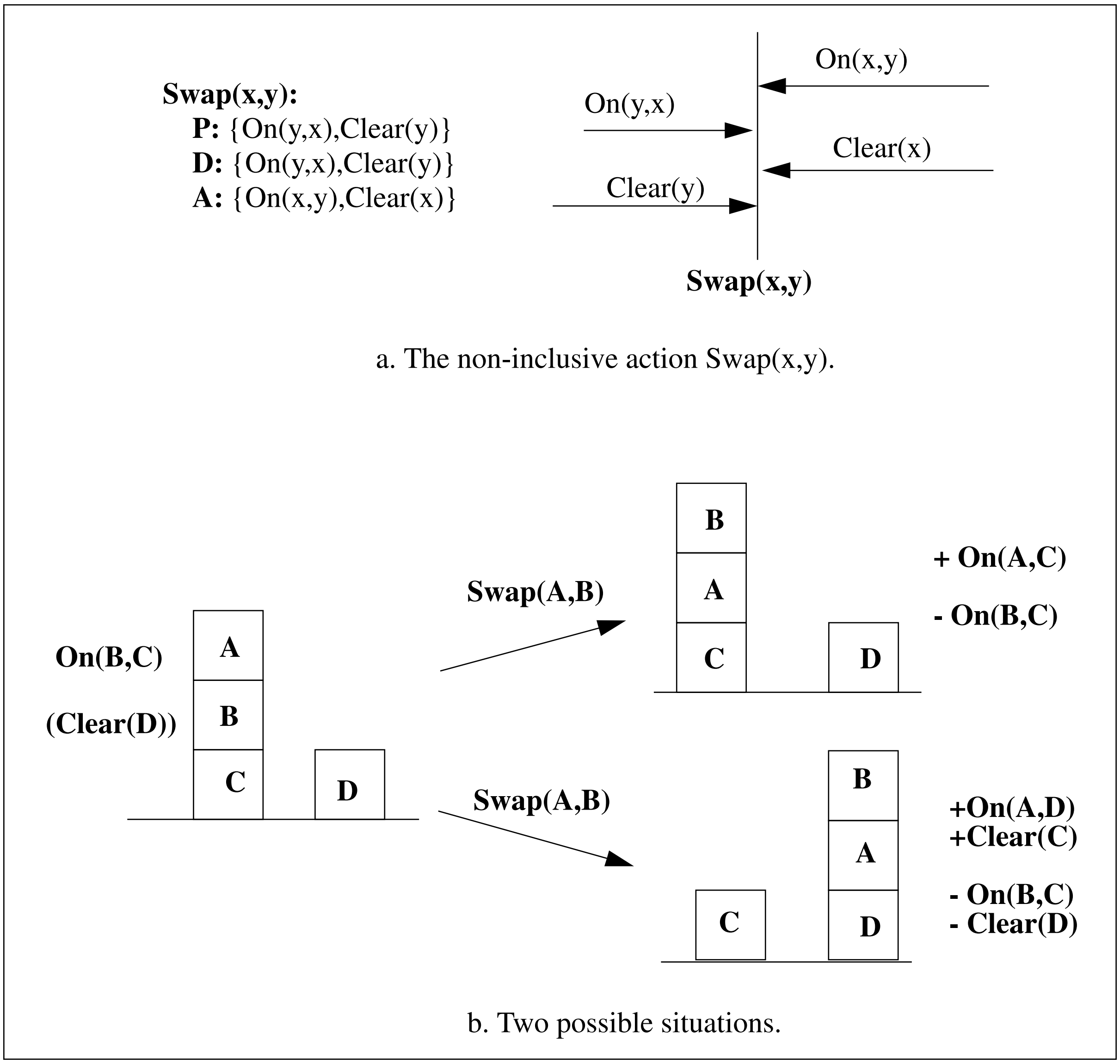


Figure 8: A non-inclusive action.

3.5 Minimum Restriction

The above four restrictions are defined on the individual actions. The minimum restriction is applied to the whole set of actions for a problem domain. This restriction requires that if two different actions in the set Γ have the same outside effects or the outside effects of one action are included in that of the other, then the two actions be replaced by one. In this way, the set of actions remains to be minimum.

Definition 7 (Minimum Restriction) *Given a set of domain constraints Σ_D, a set of actions Γ is said to be* minimum *if for any two different actions A_1 and A_2 in Γ,*

either *the actions A_1 and A_2 do not have a common deleted property and they do not have a common asserted property,*

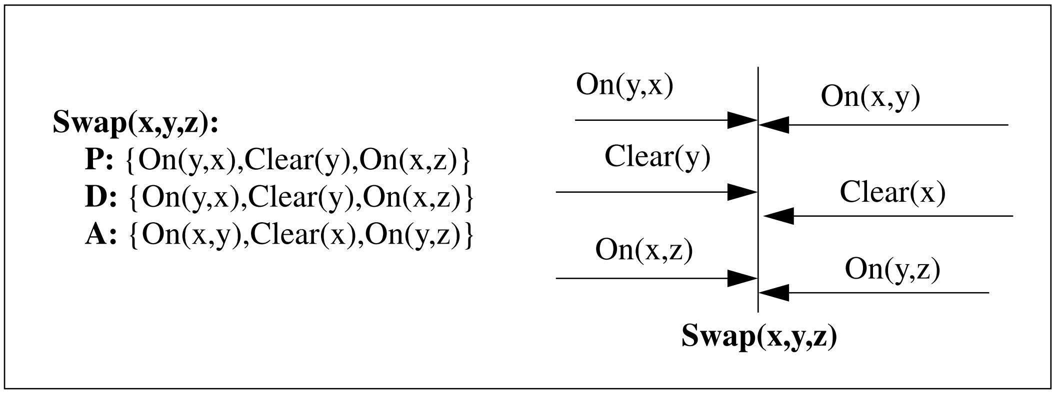


Figure 9: An inclusive action Swap(x,y,z).

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

Let Σ_D be the set of domain constraints and $Puton(x,y,z)$ be the action defined in Example 3 and Γ_1 be a set of actions that consists of the actions $Puton(x,y,z)$ and $Swap(x,y,z)$ (defined in Fig. 8b). Let Γ_2 be a set of actions that consists of the actions $Puton(x,y,z)$ and $Unstack(x,y)$ (defined in Fig. 7). It is easy to check that both Γ_1 and Γ_2 are minimum according to the Σ_D. In fact, Γ_1 and Γ_2 satisfy all the other restrictions according to the Σ_D. However, the set $\Gamma_1 \cup \Gamma_2$ is not minimum for the Σ_D. For example, all the deleted properties and all the asserted properties of the action $Unstack(A,B)$ are included in those of the action $Swap(B,A,Table)$. Therefore, the two actions should not co-exist according to the minimum restriction. The reason for this problem is that the two actions are not at the same (abstract) level. The action $Swap$ can be seen as a non-atomic action which may take the atomic actions $Unstack$ and $Puton$ as its sub-actions. The action $Swap$, therefore, should take more time than the other two actions and it should be considered as an interval action. When non-atomic actions (interval actions) are considered, new properties (internal properties) and new domain constraints should also be introduced. In this example, if the action $Swap$ is defined to be an interval action, then at least a new property $Holding(x)$ and new domain constraints $Holding(x) \leftrightsquigarrow On(y,x)$, $Holding(x) \leftrightsquigarrow On(x,y)$ and $Holding(x) \leftrightsquigarrow Clear(x)$ should be introduced. The minimum restriction can be satisfied by a set of actions that contains both atomic and non-atomic actions (for example, $\Gamma_1 \cup \Gamma_2$) and the set of domain constraints that involves internal properties. The above example shows that the minimum restriction can prevent actions at different abstraction levels, for example, the action of building a house and the action of fixing an electric bulb in the house, from being assigned the same execution time (i.e. from being treated both as point actions).

Actions are characterized by their effects on the outside world. An action's effects on the outside world are decided by its deleted and asserted properties. The minimum restriction requires that two different actions do not co-exist if they have the same outside effects or all the outside effects of one action are included in that of the other. In a set of actions, if two different actions have the same outside effects or the outside effects of one action are included in that of the other, then, because of the inclusion restriction, the two different actions must differ in the undeleted preconditions. In this case, to make the set

sliceShroomWslicer(x,y,z)
P: On(x,z),Slicer(y),
Mushroom(x),Whole(x)
D: Whole(x)
A: Sliced(x)

sliceShroomWchefKnife(x,y,z)
P: On(x,z),ChefKnife(y),
Mushroom(x),Whole(x)
D: Whole(x)
A: Sliced(x)

.
.
.

diceHamWCleaver(x,y,z)
P: On(x,z), Cleaver(y),
Ham(x), Whole(x)
D: Whole(x)
A: Diced(x)

a. a set of actions which is not minimum

makeInPieces(x)
P: HaveKnife, Food(x),
Available(x), Whole(x)
D: Whole(x)
A: InPieces(x)

b. the “equivalent” action which is minimum

Figure 10: Modifying a set of actions to make them minimum.

of actions minimum, the corresponding undeleted preconditions should be modified so that the two actions can be merged into one. Modification of undeleted preconditions of actions will not change the outside effects of the original actions. The following example in Kautz's cooking domain from (Tenenberg, 1991) shows how undeleted preconditions can be modified so that a set of actions that is not minimum can be replaced by one that is minimum.

Example 8 *Let Γ consists of the actions defined in Fig. 10a. Γ is not minimum, because any two different actions in the set delete and assert the same properties. Tenenberg uses a variant of the action in Fig. 10b to replace these actions at the abstract level in planning. Using this abstract action to replace all the actions given in Fig. 10a makes Γ minimum.*

Tenenberg uses the example to show how the *concrete-level actions* containing concrete level details which expand search space can be mapped to more efficient abstract level actions of which approximate solutions can be obtained. This replacement will not harm the original purpose of planning in the problem domain. For example, suppose that the planning process on the modified set of actions decides that the mushrooms must be cut into pieces before they are fried in the pan. Then the selection between the Slicer and ChefKnife to cut the mushroom will not affect this decision (the plan).

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.

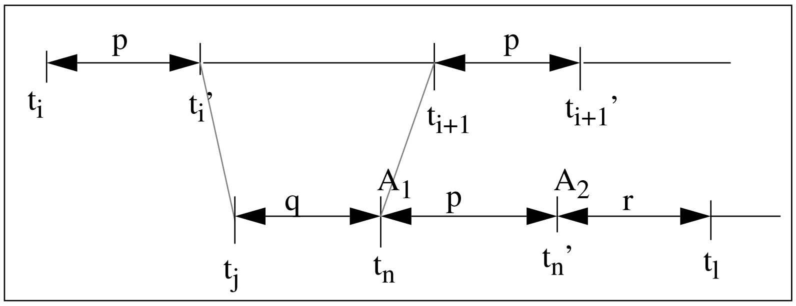


Figure 11: The situation when there are n intervals of p.

Case 2. The interval of r is after the $(i+1)$th interval of p, i.e. $t'_{i+1} \preccurlyeq t'_n$. In this case, the interval of q must also be finished by some action a_1 before the time point t_{i+1}. By Proposition 2 and by the assumption that Σ_D is satisfied, a_1 and A_1 must be the same action, see Fig. 12. Suppose that the interval (t_{i+1}, t'_{i+1}) of p is started by action a_2. By the complement restriction, there is at least one deleted property q' of a_2 such that $q' \leftrightsquigarrow p \in \Sigma_D$. According to the assumption, either the interval of q' is after or is before the interval (t_n, t'_n). Because the interval (t'_n, t_l) of r is after (t_{i+1}, t'_{i+1}) of p, the only possibility is $t_n = t_{i+1}$

The temporal relation $t'_n = t'_{i+1}$ can be proved in a similar way. Therefore, the nth interval of p is exactly the $(i+1)$th interval of p and is in the linearity relation with the other intervals of p by induction hypothesis.

■

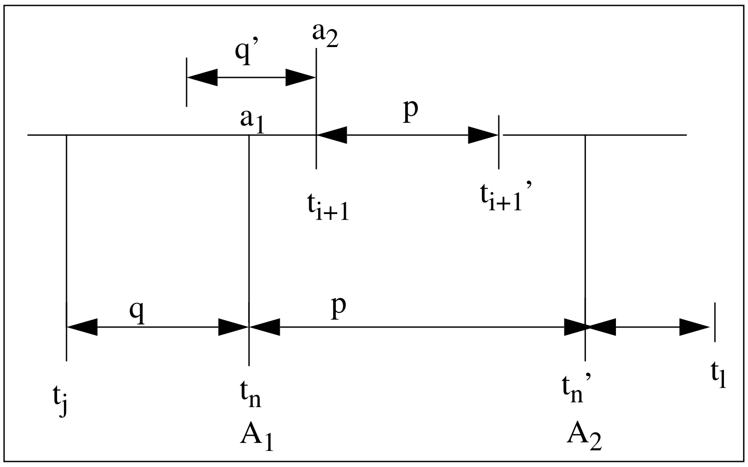


Figure 12: Part of the temporal model in Case 2.

In Example 1, we give a nonlinear plan P that contains conflicts. The temporal model of the nonlinear plan is shown in Fig. 2. In this model, the property $Clear(B)$ holds true over two intervals (T_0, T_∞) and (T_0, t_1) with the temporal relation $T_0 \prec t_1 \prec T_\infty$ satisfied 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.*

Proof By Contradiction. Suppose that in the nonlinear plan P, the action A_i executed at t_i may possibly delete one of the preconditions p of the action A_j executed at t_j. Therefore, in the temporal model of P, t_i is not necessarily after t_j. Then, there are only three possibilities about the relationship between the two time points in the model.

Case 1. $t_i = t_j$. In this case, A_i and A_j are not the same action, otherwise, there is no conflict at all. Again, there are two possible cases.

Subcase 1. p is also deleted by the action A_j. By Proposition 2, either there are two asserted properties r_1 and r_2 of the two actions respectively such that $r_1 \leftrightsquigarrow r_2 \in \Sigma_D$ or there are two deleted properties q_1 and q_2 of the two actions respectively such that $q_1 \leftrightsquigarrow q_2 \in \Sigma_D$. In both cases, Σ_D is not satisfied by the model.

Subcase 2. p is not deleted by action A_j. Because p is deleted by action A_i, by the complement restriction, there exists at least an asserted property q of A_i such that $q \leftrightsquigarrow p \in \Sigma_D$. In this case, q and p necessarily and properly overlap. Therefore, Σ_D is not satisfied .

Case 2. $t_i \prec t_j$. Because A_i possibly deletes the precondition p of A_j in P, then there does not exists another action which asserts p and is necessarily between the two actions in P. Therefore, we can construct a completion of the model of P, in which the actions asserting p for A_j precede A_i. Part of this completion is shown in Fig. 13. In this completion of the model, the interval of p finished at t_i properly overlaps the interval of p which is finished at t_j. This implies that the necessary-biggest-interval axiom for p is not satisfied in the model of P. By Theorem 1, Σ_D is not satisfied by the model.

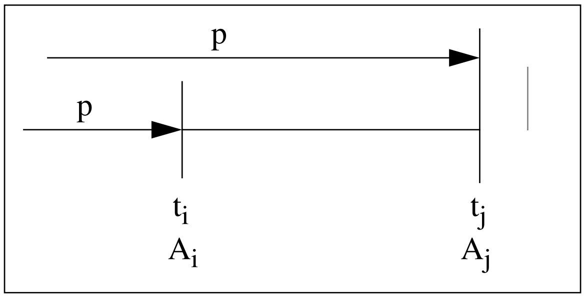


Figure 13: Part of the temporal model in Case 2.

Case 3. Neither $t_i \prec t_j$ nor $t_j \preccurlyeq t_i$ hold true in the model. In this case, because p is a deleted property of A_i and p is a precondition of A_j, the necessary-biggest-interval axiom for p is not satisfied in the model. By Theorem 1, Σ_D is not satisfied in the model.

■

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.

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

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.

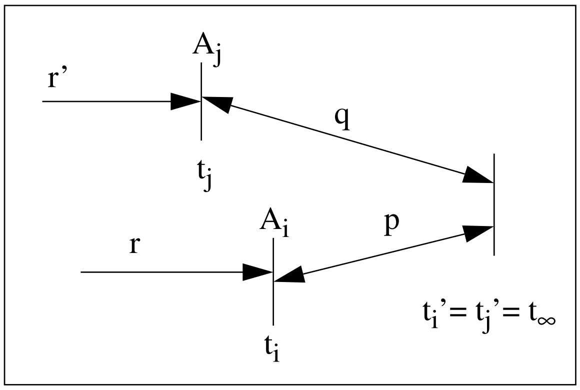


Figure 14: Part of the temporal model in Subcase 1 of Case 2.

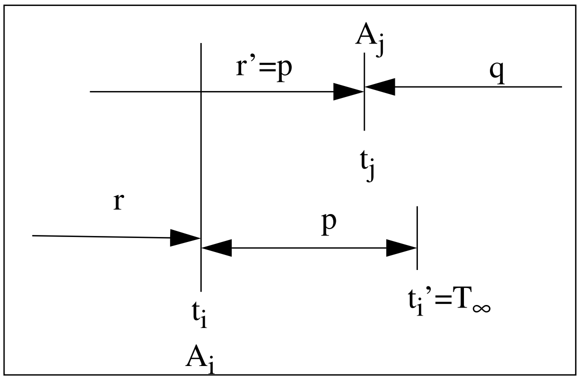


Figure 15: Part of the temporal model in Subcase 2 of Case 2.

Subcase 3. $t_i = t_j$. In this case, the only possibility which is different from that of subcase 1 and subcase 2 is that $r \leftrightsquigarrow q \in \Sigma_D$ and $r' \leftrightsquigarrow p \in \Sigma_D$ and r and r' are deleted properties of A_i and A_j respectively. Then by the inclusion restriction, r is either a deleted property of A_j or mutually exclusive to a precondition of A_j. If r is a deleted property of A_j, then, no matter how many actions assert r before t_i (t_j), a deleted condition conflict occurs among A_i and A_j for r. Otherwise, the case contradicts the nearest to the initial time point assumption.

Case 3. At least the interval of one property is asserted by an action in P and the interval of the other property is a precondition of some action in P. We assume that (t_i, t'_i) of p is asserted by action A_i at t_i and the interval (t_j, t'_j) of q is a precondition of action A_k at t_k such that $t_j \prec t_k \preccurlyeq t'_j$. By the assumption that Σ_D is not satisfied, neither $t'_i \preccurlyeq t_j$ nor $t'_j \preccurlyeq t_i$ is satisfied in the model. According to the relative position of point t_i in the model to the time points t_j and t_k, there are five possibilities.

Subcase 1. $t_i = t_j$. The proof of this case is the same as that of the subcase 3 of case 2.

Subcase 2 $t_j \prec t_i \preccurlyeq t_k$ holds true or $t_j \prec t_i$ and t_i has no relation with t_k in the model. We prove the case $t_j \prec t_i \preccurlyeq t_k$, see Fig. 16. The proof for the other case is similar. By inclusion restriction, A_i has a precondition r such that

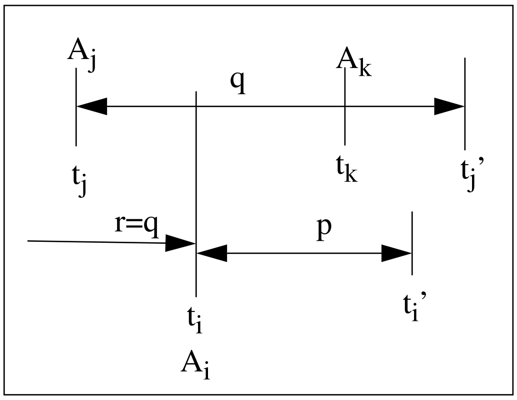


Figure 16: The first possible temporal model in Subcase 2 of Case 3.

either $r = q$ and r is a deleted property of A_i or $r \leftrightsquigarrow q \in \Sigma_D$. By the nearest to T_0 assumption, $r = q$. Therefore, A_i deletes the precondition q of A_k.

In this case, suppose that there is another action A_l at t_l which may reassert q for A_k. If t_l is necessarily between t_j of A_j and t_k of A_k in P, see Fig. 17, then by Def 1, q should not hold over the interval (t_j, t'_j) in the model. Otherwise, we can construct a completion of the model in which $t_l \prec t_i$ or $t_k \prec t_l$ holds true and A_i still deletes the precondition q for A_k in this completion.

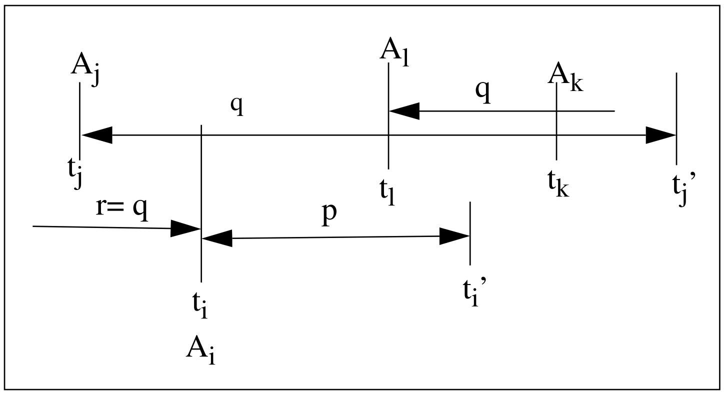


Figure 17: The second possible temporal model in Subcase 2 of Case 3.

Subcase 3. t_i has no relation with t_j and t_k or t_i has no relation with t_j and $t_i \preccurlyeq t_k$. If $t_j = T_0$, then by Definition. 1, there is no other action necessarily before t_k to assert q for A_k. Then we can construct a completion such that $t_j \prec t_i \preccurlyeq t_k$ and no action which asserts q is between t_i and t_k. Then A_i deletes the precondition q for A_k.

If $t_j \neq T_0$, then the interval (t_j, t'_j) of q is asserted by an action A_j at t_j. In this case, one of the preconditions r of A_i is either mutually exclusive with q or equal to q. The case $r = q$ implies that r is mutually exclusive to a deleted property of A_j at t_j. Both cases contradict the nearest to the initial time point assumption.

Subcase 4. $t_k \prec t_i \prec t'_j$ holds true or $t_i \prec t_k$ and t_i has no relation with t'_j in the model. We prove the case $t_k \prec t_i \prec t'_j$ first. For the same reason in subcase 2, A_i has a deleted property r such that $r = q$. In this case, $t_j \prec t_k \prec t_i$ holds true in the nonlinear plan P. Because A_i deletes q, by Definition 1, either $t'_j \preccurlyeq t_i$ or there exists another action A'_k at t'_k such that $t_k \prec t'_k \preccurlyeq t'_j$, $t'_k \not\preccurlyeq t_i$, q is a precondition of A'_k and there is not an action that asserts q is necessarily between t_i and t'_k. The first possibility, i.e. the case $t'_j \preccurlyeq t_i$, contradicts the assumption that $p \leftrightsquigarrow q$ is not satisfied by the model. For the second possibility, we can construct a completion of the model in which $t_i \prec t'_k$ holds and no actions which assert q are between t_i and t'_k. See Fig. 18. In this completion of the model, A_i deletes the precondition q for A'_k.

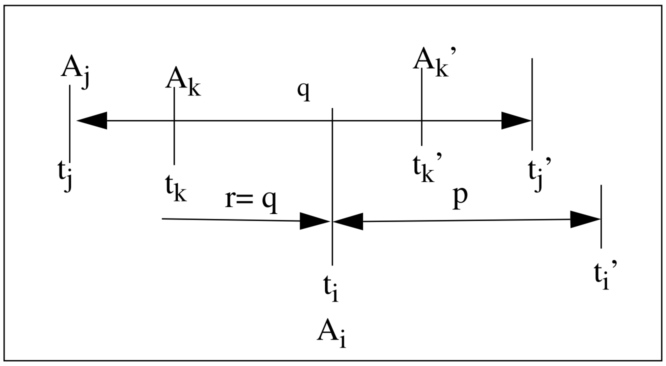


Figure 18: Part of the temporal model in Subcase 4 of Case 3.

For the case that there is no temporal relation between t_i and t'_j, there is not an action asserting q for A'_k necessarily between t_i and t'_k. Otherwise, $t_i \prec t'_j$. Then we can construct a completion of the model in which A_i deletes q for A'_k.

Subcase 5. $t_i \prec t_j$ holds true in the model. Then q must be an asserted property of action A_j at t_j. For the similar reason given in the subcase 2, A_j must have a deleted property r' such that $r' = p$. Because A_j is necessary after A_i to delete p, by Definition 1, either $t'_i = t_j$ which contradicts that $p \leftrightsquigarrow q$ is not satisfied or there exists another action A_l at t_l such that $t_i \prec t_l \preccurlyeq t'_i$ and p is a precondition of it. See Fig. 19. This is equivalent to the case that consists of the subcase 2 and the subcase 4.

■

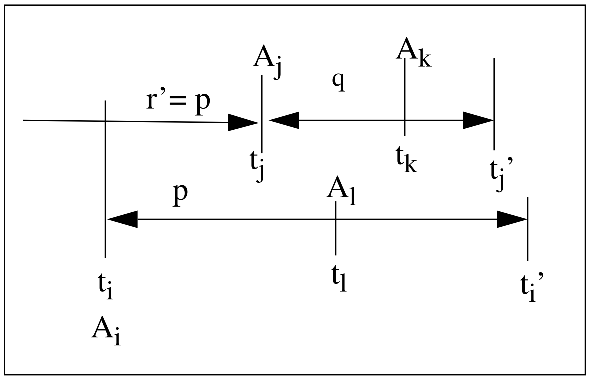


Figure 19: Part of the temporal model in Subcase 5 of Case 3.

References

Allen, James F. 1991a. Planning as Temporal Reasoning. *Pages 3–14 of: Proceedings of the 12th International Joint Conference on Artificial Intelligence.*

Allen, James F. 1991b. Temporal Reasoning and Planning. *In:* Allen, James F., Kautz, H., Pelavin, R., & Tenenberg, J. (eds), *Reasoning about Plans.* San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Allen, James F., & Koomen, J. A. 1983. Planning Using a Temporal World Model. *In: Proceedings of the 8th International Joint Conference on Artificial Intelligence.*

Chapman, David. 1987. Planning for Conjunctive Goals. *Artificial Intelligence*, **32**, 333–377.

Currie, Ken, & Tate, Austin. 1989 (May). *O-Plan: the Open Planning Architecture.* Tech. rept. AIAI-TR-67. Artificial Intelligence Applications Institute, University of Edinburgh, Edinburgh.

Currie, Ken, & Tate, Austin. 1990. *O-Plan:Task Formalism User Manual.* Tech. rept. AIAI-SR-12. Artificial Intelligence Applications Institute, University of Edinburgh, Edinburgh.

Drummond, Mark, & Currie, Ken. 1987. *Temporal Coherence in Non-linear Plan Construction.* Tech. rept. AIAI-TR-23. Artificial Intelligence Applications Institute, University of Edinburgh, Edinburgh.

Kelleher, G. 1990. *An Automatic Planner for the Support of Users of Information Processing Systems.* Ph.D. thesis, University of Leeds.

Kelleher, G., & Cohn, A. G. 1992. Automatically Synthesising Domain Constraints from Operator Descriptions. *Pages 653–655 of: Proceedings of the 10th European Conference on Artificial Intelligence.*

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.