The Logic of Reachability

David E. Smith
NASA Ames Research Center
Mail stop 269–2
Moffett Field, CA 94035
de2smith@email.arc.nasa.gov

Ari K. Jónsson
Research Institute for Advanced Computer Science
NASA Ames Research Center
Mail stop 269–2
Moffett Field, CA 94035
jonsson@email.arc.nasa.gov

Abstract

In recent years, Graphplan style reachability analysis and mutual exclusion reasoning have been used in many high performance planning systems. While numerous refinements and extensions have been developed, the basic plan graph structure and reasoning mechanisms used in these systems are tied to the very simple STRIPS model of action.

In 1999, Smith and Weld generalized the Graphplan methods for reachability and mutex reasoning to allow actions to have differing durations. However, the representation of actions still has some severe limitations that prevent the use of these techniques for many real-world planning systems.

In this paper, we 1) develop a logical notion of reachability that is independent of the particular representation and inference methods used in Graphplan, and 2) extend the notions of reachability and mutual exclusion to more general notions of time and action. As it turns out, the general rules for mutual exclusion reasoning take on a remarkably clean and simple form. However, practical instantiations of them turn out to be messy, and require that we make representation and reasoning choices.

Introduction

In 1995, Blum and Furst introduced a method for reachability analysis in planning [2, 3]. The method involves incremental construction of a plan graph to provide information about which propositions and actions are possible at each time step. Since then, plan graph analysis has been a key part of several high performance planning systems such as IPP [18], STAN [19], and Blackbox [16]. More recently, reachability analysis has been used for another purpose – to help compute more accurate heuristic distance estimates for guiding state-space search [4, 11, 24, 22] and guiding search in partial-order planners [23].

Reachability analysis and mutual exclusion reasoning have also been the subject of both efficiency improvements [19, 6], and extensions to deal with things like limited forms of uncertainty [26, 28], and resources [17]. Unfortunately, the basic plan graph structure and reasoning mechanisms are limited to the very simple STRIPS model of action. In STRIPS, one cannot talk about time – actions are considered to be instantaneous, or at least of unit duration, preconditions must hold at the beginning of actions, and effects are true in the subsequent state. Many real world planning problems require a much richer notion of time and action; actions can have differing durations, preconditions may need to hold over some or all of that duration, effects may take place at differing times, and exogenous events or conditions may occur.

In 1999, Smith and Weld [27] generalized the Graphplan methods for reachability and mutex reasoning to allow actions to have differing durations. However, the representation of actions used by Smith and Weld still made a number of simplifying assumptions:

1. All effects take place at the end of an action.
2. Preconditions that are unaffected by an action hold throughout the duration of the action.
3. Preconditions that are affected by an action are undefined throughout the duration of the action.
4. There are no exogenous events.

Unfortunately, these restrictions are not reasonable for many real-world domains [14, 25]. Many actions have resource consumption effects that occur at the beginning of the action. Others have effects that are transient. In addition, some action preconditions need only hold at the beginning of an action, or for a limited period. As an example that illustrates all of these, turning a spacecraft involves firing thrusters for periods at the beginning and end of the turn. As a result, there are transient needs for various resources (valves, controllers), transient effects like vibration and heat that occur near the beginning and end, and outright resource consumption (fuel) that occurs near the beginning, and near the end.

Finally, exogenous events are crucial in many domains. For example, in planning astronomical observations, celestial objects are only above the horizon during certain time windows, and they must not be occluded by other bright objects.

While Smith and Weld's Temporal Graphplan (TGP) planner performs reasonably well, the representation cannot be easily extended to remove the above restrictions. In particular, when exogenous events and/or transient effects are permitted, reachability and mutual exclusion relationships hold over intervals of time. For example, the action of observing a particular celestial object is only reachable during the intervals when the object is visible. A second problem with TGP is that the mutex rules are complex, and it has been difficult to verify that they are sound.

In this paper we extend the notions of reachability and mutual exclusion reasoning to deal with the deficiencies in TGP. In particular, we allow: 1) actions with general conditions and effects, and 2) exogenous conditions. Note that our objective here is not to develop a planning system that does this reasoning, but rather to lay down a formal set of rules for doing this reasoning. Given such a set of rules, there are choices concerning how much reachability reasoning one actually wants to do, which in turn leads to different possibilities data structures, implementations, and search strategies.

In the next section we introduce notation for time and actions. Using this notation, we then develop the laws for simple reachability without mutual exclusion. We then develop a very general but simple set of laws for mutual exclusion reasoning. Finally, we discuss practical issues of implementing these laws. In particular, we discuss some possible restrictions that one might want to impose on mutex reasoning and discuss how these laws can be implemented using a constraint network and generalized arc-consistency techniques.

The Basics

Propositions, Time and Intervals

To model many real world planning domains, we need to talk about propositions (fluents) holding at particular points in time, and over intervals of time. We will use the notation $p;t$ to indicate that fluent p holds at time t. We will use the notation $p;i$ to indicate that p holds over the interval i. Thus:

$$p;i \Leftrightarrow \forall(t \in i)p;t$$

We use the standard notation $[t_1, t_2]$, (t_1, t_2), $(t_1, t_2]$, $[t_1, t_2)$ to refer to closed, open, and partially open intervals respectively, and use i^- and i^+ to refer to the left and right endpoints of an interval. For any constant t, the notation $t+i$ refers to the translated interval with left and right endpoints $t+i^-$, and $t+i^+$ respectively.

For our purposes, we do not need a full set of interval relations, such as those defined by Allen [1]. However, we do need the simple relation *meet*. Two intervals *meet* if the right endpoint of the first is equal to the left endpoint of the second, and the common endpoint is contained in at least one of the two intervals (they can't be both open):[1]

$$\text{Meets}(i, j) \Leftrightarrow i^+ = j^- \wedge i^+ \in i \cup j$$

Finally, we use $i \| j$ to refer to the concatenation of two intervals that meet.

Actions

In many real world domains, actions take time. In order for an action to be successful, certain conditions may need to hold over part or all of the action. Furthermore, different effects of the action may not all occur at the same time. In fact, some of these effects may be *transient* – that is, they are only temporarily true during the action. For example, an action may use a resource (such as a piece of equipment) but release it at the end. In this case the resource becomes unavailable during the action, but becomes available again at the end of the action. To capture all of this, we model actions as having sets of conditions and effects.[2] Thus, an action is represented as:

$$\begin{array}{lll} a & \text{cond:} & p_1;\delta_1, \ldots, p_n;\delta_n \\ & \text{eff:} & e_1;\varepsilon_1, \ldots, e_n;\varepsilon_n \end{array}$$

where the times for conditions and effects are specified relative to the start of the action. More precisely, the semantics of this representation is that if action a is performed at time t, and each condition p_k holds over the interval $t+\delta_k$, then each effect e_k will hold over the interval $t+\varepsilon_k$. If the conditions do not hold, then the outcome of the action is unknown. We also require that:

1. the conjunction of the conditions and effects must be logically consistent (i.e. we cannot have inconsistent conditions, inconsistent effects, or an effect that negates a proposition at a time when it is required by the conditions.)
2. each effect must start at or after the beginning of the action, that is: $\varepsilon_k^- \geq 0$

Using this action representation, a simple STRIPS action with preconditions $p_1, \ldots, p_n$ and effects $e_1, \ldots, e_n$ would be modelled as:

$$\begin{array}{lll} a & \text{cond:} & p_1;0, \ldots, p_n;0 \\ & \text{eff:} & e_1;1, \ldots, e_n;1 \end{array}$$

As a more complex example, consider an action that requires that p hold throughout the action, and requires a resource r for two time units before releasing r and producing its final effect f. This would be modelled as:

$$\begin{array}{lll} a & \text{cond:} & p;[0, 2], r;0 \\ & \text{eff:} & \neg r;(0, 2), r;2 \wedge f;2 \end{array}$$

Note that there is a subtle difference between the effects: $\{e;[0, 2), \neg e;2\}$, $e;[0, 2)$, and $\{e;0, \neg e;2\}$. The first specifies that e holds over t+[0,2), and ceases to hold after that. The second says that e holds over t+[0,2) but may persist after that if nothing else interferes. The last specifies that e holds at t, does not hold at t+2, but leaves the status of e at intermediate times subject to persistence or interference by other actions. All three of these turn out to be useful, but the first is generally the most common.

For convenience we will use Cond($a;t$) and Eff($a;t$) to refer to the conditions and effects for action a performed at time t. Thus, if the action a is described by:

$$\begin{array}{lll} a & \text{cond:} & p_1;\delta_1, \ldots, p_n;\delta_n \\ & \text{eff:} & e_1;\varepsilon_1, \ldots, e_n;\varepsilon_n \end{array}$$

1. Unlike the definition of Allen [1], our definition of Meets is not symmetric. We also permit the endpoint to be in both intervals. Technically this would be considered overlap by Allen.

2. For simplicity, we have chosen not to include disjunctions in the condition, or conditional effects. Both of these can be handled, but complicate the axioms.

we get:

$$\text{Cond}(a;t) = \left\{p_1;t+\delta_1, \ldots, p_n;t+\delta_n\right\}$$

$$\text{Eff}(a;t) = \left\{e_1;t+\varepsilon_1, \ldots, e_n;t+\varepsilon_n\right\}$$

It is not particularly important how we define the duration of an action, but in keeping with the usual intuitions, we will define it as being the difference between the end of the last effect, and the start of the action. Thus:

$$D(a;t) \equiv \max_{\{j\,:\,e;j \in \text{Eff}(a;t)\}} j^+$$

Exogenous Conditions

In order to model more realistic planning problems, we need to model *exogenous conditions*. By an exogenous condition, we mean any condition dictated by actions or events not under the planner's control. For a STRIPS planning problem, the initial conditions are the only type of exogenous conditions permitted. More generally, exogenous conditions can include such things as the intervals during which certain celestial objects are visible, or the times at which resources become available. We can consider exogenous conditions as being the effects of unconditional exogenous actions. For convenience, we will lump all exogenous conditions together, and consider them as being the effects of a single unconditional action, X that occurs at time 0:

X cond:
eff: $xc_1;\delta_1, \ldots, xc_n;\delta_n$

where for initial conditions, the interval would be the time point 0. Thus, for a telescope observation problem, we might have something like:

X cond:
eff: Telescope-parked;0
Sunset;0023
Visible(C842);[0217, 0330]
…

For purposes of this paper, we have chosen to consider only unconditional exogenous events. More generally, we might want to consider *conditional* exogenous events – i.e., events that occur only if the specified conditions are met. As it turns out, this extension requires a few additional axioms, but is otherwise not particularly difficult. We will elaborate on this later.

Simple Reachability

We first consider a very simple notion of reachability; we regard a proposition as being *reachable* at time t if there is some action that can achieve it at time t, and each of the conditions for the action is reachable at/over the specified time or interval. This is a very optimistic notion of reachability because even though two conditions for an action might be possible, they might be mutually exclusive, and we are not yet considering this interaction. To formalize reachability, we will use two modal operators, $\Diamond(p;t)$, and $\Delta(p;t)$. $\Diamond(p;t)$, means that $p;t$ is logically *possible* – that is, $p;t$ is consistent with the exogenous conditions. $\Delta(p;t)$ means that $p;t$ is optimistically achievable or *reachable* – that is, there is some plan that could (optimistically) achieve $p;t$. According to these definitions, if $p;t$ is reachable, it is possible. However the converse is not true – $p;t$ can be logically possible, but not reachable, because the set of actions is not sufficiently rich to achieve $p;t$.

For convenience, we will allow $\Diamond$ and Δ to apply to intervals as well as single time points:

$$\Diamond(p;i) \equiv \forall(t \in i)\Diamond(p;t)$$

$$\Delta(p;i) \equiv \forall(t \in i)\,\Delta(p;t)$$

In general, modal logics tend to have nasty computational properties, but the logic we will develop here is particularly simple – we do not require any nesting of these modal operators, and we will not be allowing any quantification inside of a modal operator.

Exogenous Conditions

The first set of axioms we need are the exogenous conditions. Thus:

$$(\text{Eff}(X;0) \vdash p;t) \;\vdash p;t \qquad (1)$$

Of course, the exogenous conditions are also both possible and reachable:

$$p;i \Rightarrow \Diamond(p;i) \qquad (2)$$

$$p;i \Rightarrow \Delta(p;i) \qquad (3)$$

Likewise, the negation of any exogenous condition cannot be either possible or reachable:

$$p;i \Rightarrow \neg\Diamond(\neg p;i) \qquad (4)$$

$$p;i \Rightarrow \neg\Delta(\neg p;i) \qquad (5)$$

Finally, we need to be able to apply the closed world assumption to the exogenous conditions, inferring that anything that is not explicitly prohibited by the initial conditions is possible:

$$(\text{Eff}(X;0) \nvdash \neg p;t) \;\vdash\; \Diamond(p;t) \qquad (6)$$

Persistence

Next, we need a frame axiom for reachability – that is, an axiom that allows us to infer that if a proposition is reachable at a given time then it is reachable later on, just by allowing it to persist. However, we need to make sure that the proposition isn't forced to become false by an exogenous condition. To do this, we require that the proposition also be possible. A first version of this axiom is:

$$\Delta(p;i) \wedge \text{meets}(i, j) \wedge \Diamond(p;j) \Rightarrow \Delta(p;i \,\|\, j) \qquad (7)$$

Here, the intervals i and j can be either open or closed – all we require is that they meet. Most commonly, i will be a single time point t, and j an open interval (t,t'), where t' is either ∞, or the next time point at which the proposition p becomes false because of exogenous conditions.

Unfortunately, this axiom is a bit too optimistic – it allows us to persist transient effects of an action indefinitely into the future. Normally this is correct, but if an exogenous condition blocks a condition for that action at some time in the future, then the transient effect should not persist indefinitely. For example, suppose that we have a single action a having condition $p;0$, and requiring a resource r for two time units before releasing r and producing its final effect f. This would be modelled as:

$$\begin{array}{lll} a & \text{cond:} & p;0,\ r;0 \\ & \text{eff:} & \neg r;(0,2),\ r;2,\ f;2 \end{array} \quad (8)$$

Now suppose that the conditions p and r are initially true, but p becomes false at time 3. As a result, a is only reachable up until time 3. The effect f is first reachable at time 2, but can persist indefinitely. However, $\neg r$ can only occur during the action, and should therefore only be reachable in the interval (0,5). However, Axiom (7) would allow us to persist the reachability of $\neg r$ indefinitely into the future.

The way we fix this problem is to specialize axiom (7) to only allow action effects to persist if they are not later overridden by the action. Formally, we define $p;i$ to be a *persistent effect* for an action if there is no other effect $q;j$ such that q is inconsistent with p and j ends after i:[3]

$$\text{PersistEff}(a;t) = \left\{ p;i \in \text{Eff}(a;t) : \left(\neg\exists q;j \in \text{Eff}(a;t) : (q;j \Rightarrow \neg p;s) \wedge s > i^{+}\right) \right\}$$

Using this definition, we can restrict axiom (7) by requiring that $p;t$ be a persistent effect:

$$\exists a, t : \Delta(a;t) \wedge p;i \in \text{PersistEff}(a;t) \wedge \text{meets}(i, j) \wedge \Diamond(p;j) \Rightarrow \Delta(p; i \parallel j) \quad (9)$$

This allows us to persist the reachability of persistent effects, but not transient ones.

Actions

Finally, we need axioms that govern when actions are reachable, and what their effects will be. An action is reachable if its conditions are reachable and the effects are not prevented:

$$\Delta\text{Cond}(a;i) \wedge \Diamond(\text{Eff}(a;i)) \Rightarrow \Delta(a;i) \quad (10)$$

Conversely, if an action is reachable, both its conditions and its effects must be reachable:[4]

$$a;t \Rightarrow \text{Cond}(a;t) \wedge \text{Eff}(a;t) \quad (11)$$

$$\Delta(a;i) \wedge (a;t \Rightarrow p;t') \Rightarrow \Delta(p;t') \quad (12)$$

3. Since the effects of an action must be consistent, the intervals i and j will actually be disjoint.

4. Technically, equation (11) is not valid because it is possible to initiate an action $a;t$ even though some of its later conditions fail to hold. According to our semantics, the outcome of such an action is undefined. However, for our purposes, we will assume that no planner would include the action $a;t$ without guaranteeing Cond($a;t$). As a result, we can get away with this assumption.

Conjunctive Optimism

Although Axiom (10) is technically correct, it is difficult to satisfy. The trouble is the premise $\Delta\text{Cond}(a;t)$. Typically, the condition for an action will be a conjunction of propositions, so we need to be able to prove that this conjunction is reachable in order to be able to use the axiom. Unfortunately, we cannot usually do this, because our axioms only allow us to infer that individual effects are possible, (or at best, conjunctions of effects resulting from the same action). Deciding whether a conjunction of propositions is reachable is a planning problem, so there is little hope that we can do it efficiently. Instead, we will be extremely optimistic, and suppose that if the individual propositions are reachable, then the conjunction is reachable:

$$\Delta(p_1;i_1) \wedge \ldots \wedge \Delta(p_n;i_n) \Rightarrow \Delta(p_1;i_1 \wedge \ldots \wedge p_n;i_n) \quad (13)$$

In the next section we will revise this axiom to require that the propositions are not *mutually exclusive*.

An Example

To see how the axioms for simple reachability work, we return to our example action shown in equation (8). This action has a condition $p;0$, and requires a resource r for two time units before releasing it and producing the effect f. We suppose that the conditions p and r are initially true, but p becomes false at time 3. We therefore have the exogenous conditions:

$$\begin{array}{lll} X & \text{cond:} & \\ & \text{eff:} & p;0,\ r;0,\ \neg p;3 \end{array}$$

Using the axioms developed above, we can now derive reachability for the propositions p, r, $\neg r$, e, and the action a:

1.	$p;0$, $r;0$, $\neg p;3$	$X;0$, (1)
2.	$\Delta(p;0)$, $\Delta(r;0)$	1, (3)
3.	$\Diamond(p;(0,3))$, $\Diamond(r;(0,\infty))$	1, (6-CWA)
4.	$\Delta(p;[0,3))$, $\Delta(r;[0,\infty))$	2, 3, (9-Persist.)
5.	$\Delta(a;[0,3))$	4, (10)
6.	$\Delta f;[2,5)$, $\Delta(\neg r;(0,5))$	5, (12)
7.	$\Diamond(f;[5,\infty))$	1, (6-CWA)
8.	$\Delta(f;[2,\infty))$	6, 7, (9-Persist.)

In this proof the numbers at right refer to the previous lines of the proof, and the axioms that justify the step. A graphical depiction of the final reachability intervals is shown in Figure 1.

Thus, we can see that because the action a is only possible until time 3, $\neg r$ only persists until time 5, but f can persist indefinitely. Of course, if there were an exogenous effect that forced f to be false at some time in the future, then the persistence of f would also be curtailed by axioms (6) and (9). If p later became true again, we would be able to apply action a again, so the action a, and propositions f and $\neg r$ could become reachable during additional intervals.

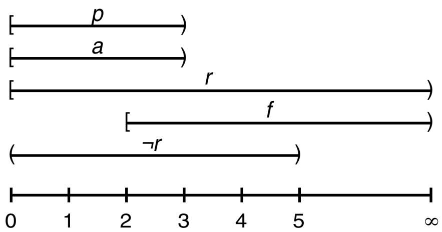

Figure 1: Reachability intervals for a simple example.

The style of reasoning that we have done here closely mimics what goes on in Graphplan – we started at time 0, and worked forward in time, adding new actions and propositions as they became reachable. However, we are not limited to a strict temporal progression – we can draw conclusions in any order, as long as they are sanctioned by the axioms.

Mutual Exclusion

Much of the power of Graphplan comes from the use of mutual exclusion reasoning, which rules out many combinations of incompatible actions and propositions. From the point of view of our logic, proving that two or more actions or propositions are mutually exclusive amounts to proving that the conjunction is not reachable. We will use an n-ary modal operator

$$M(p_1;t_1, \ldots, p_n;t_n)$$

to indicate that the propositions $p_1;t_1, \ldots, p_n;t_n$ are mutually exclusive. We note that the arguments to M are commutative and associative. As before we will extend the notation to work on intervals:

$$M(p_1;i_1, \ldots, p_n;i_n)$$
$$\Rightarrow \forall(t_1 \in i_1, \ldots, t_n \in i_n)\ M(p_1;t_1, \ldots, p_n;t_n)$$

Using mutual exclusion, we revise the conjunctive optimism axiom (13) to be:

$$\left(\Delta(p_1;i_1) \wedge \ldots \wedge \Delta(p_n;i_n)\right) \wedge \neg M(p_1;i_1, \ldots, p_n;i_n)$$
$$\Rightarrow \Delta(p_1;i_1 \wedge \ldots \wedge p_n;i_n) \qquad (14)$$

Our job then, is to write a set of axioms that allows us to infer when propositions are mutually exclusive. This will restrict what we can infer with axiom (14), and hence restrict our ability to infer when actions are reachable using axiom (10). We note that if any set of propositions and/or actions are mutually exclusive, then any superset is mutually exclusive:

$$M(s) \wedge s \subset s' \Rightarrow M(s')$$

As in the work on Temporal Graphplan [27], the fact that we are dealing with a much more general notion of time means that actions and propositions can overlap in arbitrary ways. As a result, it helps to define mutual exclusion between actions and propositions, as well as between pairs of actions and pairs of propositions. In addition, because of exogenous events, and transient action effects, mutual exclusion relationships can come and go repeatedly.[5] As it turns out, the general rules for mutual exclusion reasoning take on a remarkably clean and simple form. However, practical instantiations of them turn out to be more complex.

Logical Mutex

If a set of propositions and or actions are logically inconsistent then they are mutex. Formally:

$$\neg(\psi_1 \wedge \ldots \wedge \psi_k) \Rightarrow M(\psi_1, \ldots, \psi_k) \qquad (15)$$

where the ψ_i can be either propositions $p;t$, or actions $a;t$. This rule is the seed that allows us to infer a number of simple logical mutex relationships. For example, if $\psi_1 \equiv p;t$ and $\psi_2 \equiv \neg p;t$ we get the obvious mutex rule:

$$M(p;t, \neg p;t)$$

which forms the basis for Graphplan mutual exclusion reasoning. Similarly, if $\psi_1 \equiv p;t$, and $\psi_2 \equiv a;t'$, and $a;t'$ has a precondition or effect $\neg p;t$, then the action and proposition are mutex (since $(a;t' \Rightarrow \neg p;t)$):

$$(a;t' \Rightarrow \neg p;t) \Rightarrow M(p;t, a;t')$$

Going a step further, if we have two actions with logically inconsistent preconditions or effects this rule allows us to conclude that the actions are mutex:

$$(a_1;t_1 \Rightarrow p;t) \wedge (a_2;t_2 \Rightarrow \neg p;t) \Rightarrow M(a_1;t_1, a_2;t_2)$$

Although we will not illustrate it here, rule (15) also admits the possibility of inferring additional logical mutex from domain axioms that might be available (e.g. an object cannot be in two places at once). It can also be used to derive logical mutex between actions that have more general resource conflicts.

All of these logical mutex relationships are the seeds that serve to drive the remainder of the mutex reasoning. As we will see below, they allow us to infer additional mutex relationships between actions and propositions, pairs of actions, and ultimately pairs of propositions.

Implication Mutex

Our second mutex rule is also remarkably simple, but more subtle. Let $\Gamma = \Gamma_1 \cup \Gamma_2$ be a set of propositions/actions that are mutex. Suppose that a second set of propositions/actions Ψ implies Γ_1 . Then the set $\Psi \cup \Gamma_2$ is also mutex. Formally:

$$M(\Gamma_1 \cup \Gamma_2) \wedge (\Psi \Rightarrow \Gamma_1) \Rightarrow M(\Psi \cup \Gamma_2) \qquad (16)$$

Again, the set elements can be either propositions or actions. For binary mutex, this reduces to the formula:

$$M(\psi_1, \psi_2) \wedge (\psi_3 \Rightarrow \psi_1) \Rightarrow M(\psi_3, \psi_2)$$

As an example of the use of this formula, suppose that ψ_1 and ψ_2 are mutex propositions, and ψ_3 is an action that has ψ_1 as a precondition. Since the action implies its preconditions, this rule allows us to infer that the action is mutex with ψ_2 . Going one step further, if ψ_2 is an action, then this rule allows us to conclude that the actions ψ_3 and ψ_2 are mutex. Thus, this single rule allows us to move from proposition/

5. In Graphplan and even TGP, once a mutex relationship disappears, it cannot reappear at a later time.

proposition mutex to proposition/action mutex, to action/action mutex.

To see how this works, consider two simple STRIPS actions: *a*, having precondition *p* and effect *e*, and *b*, having precondition *q* and effect *f*. Suppose that both *p* and *q* are reachable at time 1, but that they are mutex as depicted graphically in Figure 2. We can therefore apply the above

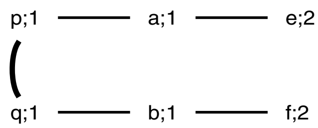

Figure 2: A simple STRIPS example with p and q mutex at time 1.

rule to conclude that *a;1* is mutex with *q;1* and *b;1* is mutex with *p;1*. Having done this, we can apply the rule again to conclude that *a;1* is mutex with *b;1* as shown in Figure 3.

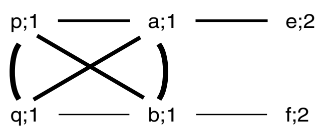

Figure 3: Mutex derived by the implication rule

While axiom (16) works fine for a discrete STRIPS model of time, more generally, we do not want to do the mutex reasoning for each individual time point. Instead, we would like to do it for large intervals of time. So suppose we start out with two propositions/actions φ_1 and φ_2 being mutex over the intervals i_1 and i_2, and $\varphi_3;t_3 \Rightarrow \varphi_1;t_1$. Then to find the time interval over which φ_3 will be mutex with $\varphi_2;i_2$, we need to gather up all the times t_3 where φ_3 implies φ_1 at some point in i_1. Formally:

$$M(\varphi_1;i_1, \varphi_2;i_2) \ \wedge \ i_3 = \left\{ t_3 : \varphi_3;t_3 \Rightarrow \left(\exists(t_1 \in i_1) : \varphi_1;t_1\right)\right\} \Rightarrow M(\varphi_3;i_3, \varphi_2;i_2) \quad (17)$$

To illustrate how this works, we extend our example from Figure 2 to continuous time, and imagine that *p* and *q* are produced by mutually exclusive actions of different duration. In particular, suppose that *p* over [1,3) is mutually exclusive with *q* over [2,3). Using (17) we could conclude that:

$$M(a;[1,3), q;[2,3))$$
$$M(b;[2,3), p;[1,3))$$
$$M(a;[1,3), b;[2,3))$$

as illustrated in Figure 4.

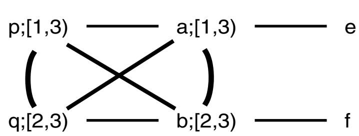

Figure 4: Implication mutex for intervals

Explanatory Mutex

Our final rule is somewhat subtle and tricky. As a result, we will only show the binary version here, although it too can be generalized to n-ary mutex. This rule is, in effect, the explanatory version of the previous rule. Basically, it says that if all ways of proving ψ_1 are mutex with ψ_2 then ψ_1 and ψ_2 are mutex:

$$\left(\forall\psi_3\left((\psi_3 \Rightarrow \psi_1) \Rightarrow M(\psi_3, \psi_2)\right)\right) \Rightarrow \ M(\psi_1, \psi_2) \quad (18)$$

The tricky part is the phrase "all ways of proving". For our purposes, we are interested in the case where ψ_1 is a proposition p;t and ψ_3 is a way of achieving *p;t*. We could achieve *p;t* by performing an action *a;t'* that has *p;t* as an effect, but we could also potentially perform the action *a* at some earlier time and allow *p* to persist. Thus, we need to account for all of these possibilities. Furthermore, if *p* is achieved earlier and allowed to persist, that "means of achieving" could be mutex with ψ_2 for one of two reasons: either *a;t'* is mutex with ψ_2, or the persistence of *p* is mutex with ψ_2.

To formalize this, we define the *support* of a proposition as being the union of the *direct support* and the *indirect support* for the proposition:

$$\text{Supp}(p;t) \ = \ \text{DirSupp}(p;t) \cup \text{IndSupp}(p;t)$$

The direct support is simply the set of actions that can directly achieve the proposition:

$$\text{DirSupp}(p;t) \ = \ \left\{ a;t' : \Delta(a;t') \wedge (\text{Eff}(a;t') \Rightarrow p;t)\right\}$$

The indirect support is a set of miniature plans for achieving the proposition, each consisting of an action *a;t'* that achieves the proposition before *t*, and the persistence of the proposition until *t*. As with persistence axiom (9), we need to be careful not to rely on the persistence of transient effects:

$$\text{IndSupp}(p;t) \ = \ \left\{ a;t' \wedge p;(t'', t\,] : \Delta(a;t') \wedge t'' < t \wedge (\text{PersistEff}(a;t') \Rightarrow p;t'') \wedge \Diamond(p;(t'', t\,])\right\}$$

Using this concept of support, we can restate our more specific version of (18) as:

$$\left(\forall\sigma \in \text{Supp}(p;t) : M(\sigma, \psi)\right) \ \Rightarrow \ M(p;t, \psi) \quad (19)$$

For the case of direct support, σ is just an action *a;t*, so we can directly evaluate $M(\sigma, \psi)$. However, for indirect effects, σ is a conjunction of an action *a;t* and a persistence *p;i*. If either of these is mutex with ψ, then the conjunction is mutex with ψ. More generally:

$$M(\sigma_1, \psi) \vee M(\sigma_2, \psi) \ \Rightarrow \ M(\sigma_1 \wedge \sigma_2, \psi)$$

As a result, we expand axiom (19) into the more useful form:

$$\left(\forall\sigma \in \text{DirSupp}(p;t) : M(\sigma, \psi)\right)$$

$$\wedge \left(\forall (\alpha \wedge \pi) \in \mathsf{IndSupp}(p;t) : M(\alpha, \psi) \vee M(\pi, \psi) \right)$$

$$\Rightarrow M(p;t, \psi)^{6} \quad (20)$$

To illustrate how this axiom works, we return to the simple example in Figure 3. From implication mutex we already know that *a;1* and *b;1* are mutex. Effect *e;2* has only the direct support *a;1*. As a result, we can use the above rule to conclude that *b;1* is mutex with *e;2*. Similarly, we can conclude that *a;1* is mutex with *f;2*. Finally, using these facts we can conclude that *e;2* is mutex with *f;2* as shown in Figure 5.

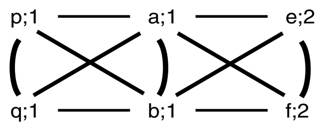

Figure 5: Mutex derived by the implication rule

As with Implication Mutex, we would like to be able to apply (19) and (20) to intervals rather than just single time points. If we generalize the notion of support to intervals, we can state the more general version as:

$$\left(\forall \sigma \in \mathsf{Supp}(p;i_1) : M(\sigma, \varphi;i_2) \right) \Rightarrow M(p;i_1, \varphi;i_2) \quad (21)$$

As we did with (19) we could expand out to the longer but more useful form containing direct and indirect support.

Practical Matters

Limiting Mutex Reasoning

The above mutex theory is very general, but it can produce huge numbers of mutex conclusions, many of which would not be very useful. In order to make the reasoning practical, we need to constrain the application of these axioms so that only the most useful mutex relationships are derived.

The first, and most obvious way of limiting the mutex rules is to only apply them to propositions and actions that are actually reachable. If something isn't reachable at a given time, it is mutex with everything else, so there is no point in trying to derive additional mutex relationships. As with ordinary Graphplan, it also seems likely that we only want to derive *binary* mutual exclusion relations, since the cost of checking for higher order mutex relationships rises dramatically with arity.

While these limitations certainly help, they are not enough. The trouble is that our laws allow us to conclude mutual exclusion relationships for propositions and actions at wildly different times. For example, we might be able to conclude that *p;2* is mutually exclusive with *q;238*. While this fact could conceivably be useful, it is extremely unlikely. To understand why, and what to do about it, we need to consider how mutex are used.

Fundamentally, we use mutex to decide whether or not the conditions for actions are reachable, and hence whether the actions themselves are reachable (axioms (14) and (10)). Thus, the mutex relations that ultimately matter are the proposition/proposition mutex between conditions for an action. With simple STRIPS actions, this means we are concerned with propositions being mutex at exactly the same time. Unfortunately, with more general conditions we can't do this – an action may require *p;t*, and *q;t*+3. Thus, we'd need to know whether $M(p;t, q;t+3)$ in order to decide whether the action was reachable. However, we do not care about $M(p;t, q;t+5)$. Suppose we define the *separation* for a pair of conditions in an action as the distance between the intervals over which the conditions are required to hold. For our example above, the condition separation was 3. We then take the maximum over all conditions for an action, and the maximum over all actions. This tells us the maximum range of times that we ultimately care about for proposition/proposition mutex relationships. In the extreme case where all preconditions of actions are required at the start of the action, we only need to consider whether propositions are mutex at the same time.

We can draw similar conclusions concerning action/action and action/proposition mutex, although in the latter case, the ranges are somewhat wider. This is because we are considering actions that support propositions, which means the actions start before their effects.

Limiting the application of the axioms to such time ranges drastically reduces the number of mutex conclusions. It is not yet clear (either theoretically or practically) whether such restrictions would result in the loss of any useful mutex relationships. This issue needs to be carefully investigated.

Constraint-Based Reachability Reasoning

We now turn our attention to the issue of finding an effective way to calculate reachability information. For this, we turn to constraint reasoning, which is an effective foundation for reasoning about temporal planning problems. The constraint-based reachability reasoning tracks variables that describe reachability, and enforces constraints that eliminate times where actions or propositions are not reachable.

The approach is motivated by the interval representation used for temporal reasoning in various planning systems. In simple temporal network propagation [7], event time domains are described as intervals, and the algorithm is used to infer distance relations between events in plans.

The basic idea appears similar to temporal networks; for each action and proposition, we have a variable representing when it is reachable, and constraints that relate action and proposition reachability. However, this reachability problem does not map to a classical temporal constraint satisfaction problem. This is because action reachability requires necessary conditions to extend over periods of time, so there is no notion of a satisfying assignment to those variables. We therefore turn to a more general class of constraint reasoning

6. In practice, if ψ is mutex with *p;t*. then we do not need to check actions that support *p* prior to *t* (since the persistence of *p* will be mutex with ψ). Thus we only need to consider support for *p* at times *t* after *p* is mutex with ψ. This involves moving the check for persistence mutex back into the definition of independent support.

problems, where the variables are linked by elimination procedures [12], that specify when intervals can be eliminated from the domains. The result is a network where reachability can be determined effectively by constraint propagation, but there is no notion of a solution to the network. Different constraint propagation methods, such as generalized arc consistency, can be applied to propagate the procedural constraints. A very simple propagation method is to apply the set of elimination procedures to quiescence.

Let T be the set of possible times, which may be continuous and infinite. Typically, T will be a sub-interval of the integers or the real numbers. For each action a, we define a variable v_a, and for each proposition p, we define a variable v_p. The initial domain of each variable is T, and the intended semantics are that the variables represent the times at which an action or proposition is reachable.

The simplest reachability procedure enforces that if a fluent is not possible, it is not reachable. This gives rise to the following intervals being eliminated for each variable v_p:

$$\{i : \neg\Diamond(p\,;\,i)\}$$

The action reachability axioms are relatively straightforward as well. Let a be an action with a condition or effect $p;\delta$. If p is not reachable within an interval $[x, y]$, then the action is not reachable at those times that require p to be true in $[x, y]$; specifically, a is not reachable in the interval:

$$[x - \delta^{+}, y - \delta^{-}]\ . \tag{22}$$

Enforcing the persistence axiom is more involved. The basic idea is that an interval $[x, y]$ where p is not reachable can be extended up to the point where an action can achieve p or an exogenous event establishes p. In other words, we can eliminate the interval:

$$\left[y, \min\left\{j^{-} > y : p;j \vee \left(t \in v_a \wedge p;j \in \mathrm{Eff}(a;t)\right)\right\}\right] \tag{23}$$

Note that in computing the upper bound on this interval, we do not need to look beyond the next time z (following y) at which p is already known to be not reachable. Thus, we can confine our search for actions that achieve p to the interval $[y, z]$

If all action effects were persistent, the above rule would be sufficient. However, with transient effects we can continue the elimination beyond the transient effect (if the transient effect ends before a persistent effect becomes possible). To do this, we compute the above interval considering only persistent action effects, and then subtract out the subintervals in which the transient effects are reachable:

$$\left[y, \min\left\{j^{-} > y : p;j \vee \left(t \in v_a \ \wedge\ p;j \in \mathrm{PersistEff}(a;t)\right)\right\}\right]$$

$$-\bigcup\left\{j : \left(i \subseteq v_a \ \wedge\ \mathrm{TransientEff}(a;i) \Rightarrow p;j\right)\right\} \tag{24}$$

The result is a set of intervals that can be eliminated from v_p. To make this elimination more efficient, we can confine our consideration of transient effects to those that are reachable in the interval we are subtracting from. But more importantly, we want to limit the application of this rule to those situations where we know that something relevant has changed. There are two circumstances where we actually need to apply this rule:

1. when a reachability interval has been eliminated from v_p (using one of the earlier elimination rules).
2. when a reachability interval has been eliminated for an action – in this case, the above rule must be applied to all effects of that action.

To see how the application of the elimination rule works, we again look at the earlier example. Initially, the following intervals have been eliminated:

$$\neg p;0\ ,\ p;3\ ,\ \neg r;0\ ,\ f;0$$

Based on this, the action condition reachability rules only allow us to eliminate $a;3$.

Applying the persistence rule to $p;3$, we get that no reachable action establishes p. Formula (23) therefore allows us to eliminate the interval $[3, \infty)$ for v_p. Applying the persistence rule to other eliminated intervals allows us to eliminate:

$$\neg p;[0, 3)$$
$$f;(0, 2)$$

Now that more intervals have been eliminated for p, the application of the action condition reachability rules allows $[3, \infty)$ to be eliminated from v_a.

Finally, the interval 0 has been eliminated from $v_{\neg r}$. Since the effect $\neg r;(0, 2)$ is not persistent, Formula (24) evaluates to $[0, \infty) - (0, 5)$. This allows us to eliminate $[5, \infty)$ from $v_{\neg r}$. Note that the final result is the same as applying the logical axioms to determine when actions and propositions may be reachable.

Constraint-Based Mutual Exclusion Reasoning

The above formulation does not include mutual exclusion reasoning. To extend the constraint reasoning to identify mutual exclusions, we add variables that correspond to pairs of propositions/actions, each of which represents the set of time pairs when the action/proposition in question are not proven to be mutually exclusive.

For any pair of propositions and/or actions, (ψ_1, ψ_2), we define a variable $\overline{M}(\psi_1, \psi_2)$ that takes its values from the set $T \times T$. The intended semantics is that the variable represents time pairs where the two elements are not mutually exclusive. In other words, each variable is the inverse of the set of timepoints where the two elements are mutually exclusive. Eliminating a pair of times (t_1, t_2) from a variable thus indicates that ψ_1 being true at time t_1 is mutually exclusive with ψ_2 being true at time t_2.

As in our approach to eliminating values to determine reachability, we use special-purpose elimination procedures to do the work. Each procedure implements a rule that determines a set of mutual exclusion relations and eliminates a set of time pairs. The rules are applied in combination by a suitable consistency achievement method.