Knowledge Preconditions for Plans

ERNEST DAVIS, *Courant Institute of Mathematical Science, New York University, New York, NY 10012, USA.*
E-mail: davise@cs.nyu.edu

Abstract

For an agent to be able to rely on a plan, he must know both that he is physically capable of carrying out the physical actions involved, and that he knows enough to carry out the plan. In this paper, we advance and discuss new definitions of 'knowing enough to carry out a plan', for the case of a single agent carrying out a sequence of primitive actions one at a time. We consider both determinate and indeterminate plans. We show how these definitions can be expressed in a formal logic, using a situation calculus model of time and a possible worlds model of knowledge. The definitions strictly subsume previous theories for the single-agent case without concurrent actions. We illustrate the power of the definition by showing that it supports results of the following kinds:

- Positive verification: showing that a plan is feasible.
- Negative verification: showing that a plan is infeasible.
- Monotonicity: the more an agent knows, the more plans are executable.
- Reduction for omniscient agent: for an omniscient agent, a plan is epistemically feasible if and only if it is physically feasible.
- Simple recursive rules that are sufficient conditions for the feasibility of a plan described as a sequence or a conditional combination of subplans.

Keywords: Preconditions, feasibility, semantics of plans, formal theory of knowledge.

1 Introduction

The question of whether an agent is able to carry out a plan can be divided into two parts. First, is the plan *physically feasible* for the agent; that is, is it physically possible to carry out the actions specified by the plan? Second, is the plan *epistemically feasible*; that is, does the agent know enough to perform the plan? A plan like 'Make money at the roulette wheel by betting on the numbers that win' is not a useful one; though the physical actions involved are feasible, there is no way to find out in time what they are.

The epistemic feasibility of a plan depends both on the knowledge the agent has at the beginning of execution, and on the knowledge he[1] gains during the course of execution. For example, suppose that my aunt Edith does not know my phone number, but she has a directory in which she knows that my name is listed. In that case, she is not immediately able to carry out the plan 'dial Ernie's number', but she is immediately able to carry out the plan 'sequence(look up Ernie's number; dial Ernie's number)'.

The problem addressed in this paper is to characterize the epistemic feasibility of a plan in terms of the physical content of the plan and of the evolving knowledge of the agent. We propose a characterization of epistemic feasibility that applies to any plan, determinate or indeterminate, carried out by a single agent performing one primitive action at a time. We show, for this class of plans, our definition strictly subsumes previous definitions of epistemic feasibility, and that it supports a wide range of natural and powerful conclusions.

[1] Anonymous agents are here denoted 'him' with no implication that they may not be 'her'.

Section 2 informally presents the problem and solution. Section 3 discusses the formal structure of the theory. Section 4 presents some general theorems. Section 5 discusses possible extensions to the theory.

2 Knowledge preconditions

Reasoning about the epistemic feasibility of plans requires a theory that integrates temporal reasoning with reasoning about knowledge. In order to determine whether an agent knows enough to perform a plan, we must be able to characterize what the agent knows at the beginning of the plan, and how the state of the world and the knowledge of the agent change as a result of the execution of the plan. Like the relation, 'At time T agent A knows fact ϕ', the relation, 'At time T, agent A knows enough to perform plan P' is referentially opaque (intensional) in its final argument. Aunt Edith does not at this moment have enough information to perform the action 'Dial Ernie's phone number', but she does have enough information to perform the action 'Dial 998-3123' which is extensionally the same action.

The problem of characterizing the epistemic feasibility of actions or plans, sometimes called the *knowledge preconditions* problem, was briefly addressed by McCarthy and Hayes [11]. The first in-depth study of the problem was that of Moore [13, 5], which we will describe below. More recently, the problem was considered by Morgenstern [16], who modified Moore's theory by using a syntactic theory of knowledge and extended it to apply to plans involving multiple agents and the communications between them. However, for the single agent case, the basic structure of the two theories is quite similar, and, though I will use Moore's theory as a referent below, the same analysis applies with minor changes to Morgenstern's.

Moore divides the problem of epistemic feasibility into two parts:

1. The knowledge preconditions (KP) problem for *actions*. Characterizing whether an agent knows enough to perform a single specified action in a given situation. For example, expressing the fact that my aunt does not now have enough information to execute 'Dial Ernie's office number'; however, if she finds out that 'Ernie's office number is 998-3123' then she will have enough information. (We are here viewing dialling a seven-digit number as a single atomic action, rather than as a series of separate finger movements.)
2. The KP problem for *plans*. For example, determining that the plan 'sequence(look up Ernie's number; dial Ernie's number)' is epistemically feasible for my aunt, given that she knows that, after looking up my number, she will know what my number is.

Clearly, a solution of the KP problem for plans must rest on a solution of the KP problem for actions.

We proceed as follows:

- In Section 2.1, we present Moore's solution to the KP problem for actions. Here, we have no changes to suggest, and we adopt this solution without modification.
- In Section 2.2, we consider the KP problem for determinate plans. We begin by presenting Moore's solution to the problem. We show that this solution is not quite adequate and requires an extension. Having made the extension, we can derive a simple criterion for the KP problem for determinate plans. We then throw away the scaffolding by showing that this simple criterion by itself subsumes Moore's theory.

- In Section 2.3 , we consider the KP problem for indeterminate plans. Moore did not consider indeterminate plans at all; Morgenstern considered a limited class of indeterminate plans.

2.1 Actions

Moore [5] proposes the following solution to the KP problem for actions: an action E is epistemically feasible for agent A at time T iff A knows at T a specific behaviour that constitutes that action in T. My aunt cannot execute dial Ernie's number because she does not know what behaviour would constitute dialling Ernie's number. If she finds out that dialling 998-3123 constitutes dialling Ernie's number, then she will know how to dial Ernie's number. In other words, in order to perform the action 'Dial Ernie's number', she must know what that action is.

Following Hintikka [8] the notion of 'knowing what Q is' is represented in formal theories by using an existential quantifier of larger scope than the epistemic operator. In a modal theory, the proposition 'A knows what Q is' may be expressed as

$$\exists_X \text{ know}(A, X = Q).$$

In a possible worlds theory, it is expressed by stating that Q has constant value in all accessible worlds [5].

$$\exists_X \forall_{W1} \text{ k\_acc}(A, W0, W1) \Rightarrow X{=}\text{value\_in}(W1, Q).$$

In a syntactic theory, it is expressed by stating that A knows a formula '$X{=}Q$' for some standard identifier X [6].

$$\exists_X \text{ stid}(X) \wedge \text{know}(A, \prec\downarrow X\downarrow = Q\succ).$$

(The possible worlds theory will be described in detail in Section 2.3. For discussions of the other representations, see [2, 7, 8, 5, 6].)

These representations are not without their difficulties and limitations, mostly deriving from the rather vague character of the relation 'knowing what Q is'. However, nothing better has been developed, and in this paper this family of solutions to the KP problem for actions is adopted without further discussion.

2.2 Determinate plans

In addressing the KP problem for plans, Moore posits a set of axioms that characterize knowledge preconditions for plans built up recursively from primitive actions using control structures such as 'sequence($P1 \ldots Pk$)', 'if($Q, P1, P2$)', and 'while($Q, P1, P2$)'. The following axioms, among others, are posited:

RULE 2.1
The plan 'sequence($P1, P2$)' is epistemically feasible for agent A at time T if A knows at T that

a. $P1$ is epistemically feasible for A at T; and
b. after $P1$ is executed, $P2$ will be epistemically feasible.

For example, the plan 'sequence(look up Ernie's number; dial Ernie's number)' is epistemically feasible because my aunt now knows how to look up my number, and she knows that, after she looks up my number, she will know how to dial my number.

RULE 2.2
The plan 'if Q then do $P1$ else do $P2$' is epistemically feasible for agent A at time T if either

a. A knows at T that Q holds at T and $P1$ is epistemically feasible for A at T; or
b. A knows at T that Q does not hold at T and $P2$ is epistemically feasible for A at T.

For example, the plan 'if it is raining out then put on a raincoat else put on a jacket' is feasible if either I know that it is raining out and I know how I can put on a raincoat or if I know that it is not raining out and I know how I can put on a jacket.

RULE 2.3
The plan 'while Q do PA' is epistemically feasible for agent A at time T if A knows at T that

a. at each iteration point (that is, at T and at the end of each iteration of PA), A will know whether Q holds;
b. at each iteration point, if Q holds, then PA is epistemically feasible;
c. eventually, the loop will terminate; that is, there will come an iteration point when Q does not hold.

(This is not Moore's statement of the rule; however, it is equivalent given Rules 2.1 and 2.2.)

However, there are gaps in these rules; these conditions are sufficient but not necessary.[2] Examples:

- (Failure of the converse of Rule 2.1.) There are cases where 'sequence(PA, PB)' is epistemically feasible even though PA is not. For example, suppose that there is a barrel containing an unknown number of apples; PA is 'take half the apples out of the barrel' and PB is 'take out the remaining half'. Then 'sequence(PA, PB)' can be carried out simply by emptying the barrel. However PA is not epistemically feasible, since you don't know when to stop.

  (Some readers may have the intuition that under these circumstance PA should be considered epistemically feasible; just empty the barrel and you know that you did carry out PA plus some extra. But under that criterion, Rule 2.1 would fail as a necessary condition; PA by itself would be epistemically feasible, but 'sequence(PA;announce success)' would not.

  Other readers may have the intuition that 'sequence(PA, PB)' should not be considered epistemically infeasible, since an ordinary plan interpreter will not be able to execute it. However, it is certainly possible to implement a plan interpreter that can execute it, by using multiple program counters when a condition cannot be resolved. Moreover, it is unwise to base the definition of fundamental concepts on the convenience of programming.)
- (Failure of the converse of Rule 2.2.) There are cases where an agent knows how to perform 'if Q then do $P1$ else do $P2$' in S without knowing whether Q holds in S; namely, if $P1$ is the same as $P2$, or if $P1$ and $P2$ begin the same, and the truth of Q in S will be found out before they start to diverge. For example, consider the parameterized plan p(X, Y) = 'if q holds then do f(X) else do f(Y)'. In the case where $X = Y$, p(X, Y) can be performed without knowing whether q holds, just by performing f(X).

In both of the above cases, though Moore's axioms do not imply that the plan in the given form is feasible, they do imply that a plan known to be equivalent is feasible. In the first case, the agent knows that the plan is equivalent to 'take all the apples out of the barrel'; in the second case, the agent knows that the plan is equivalent to 'do f(X)'. This observation suggests that we augment Moore's theory with the following rule:

[2] Moore [13] incorrectly states both rules as biconditionals; Morgenstern [6] correctly presents both rules as sufficient but not necessary conditions.

RULE 2.4
Plan P is epistemically feasible for agent A at time T if there exists a plan P' such that

a. P' is epistemically feasible for A at T; and
b. A knows at time T that P' is equivalent to P starting at T.

It is important in Rule 2.4 that the existence of P' has larger scope than A's knowledge of its equivalence to P. The statement 'A knows that there exists a P' …' would not be a sufficient condition. For example, Aunt Edith knows that there exists an epistemically feasible plan that is equivalent to 'Dial Ernie's number'; namely 'Dial(N)' for some value of N. The point is that she does not know *which* such P' is equivalent.

We now consider a particular plan transformation: For any plan P, let T1(P) be the plan

```
while ($P$ has not successfully finished)
      do (the next step[3] of $P$)
```

All we have done here is to push the whole structure of P away into the term 'the next step of P'. However, this manoeuvre has very important consequences for our definition, since 'the next step of P' is a term that denotes a single action, and, therefore, it comes under the solution of the KP problem for actions. Thus Rule 2.4 allows us to reduce the analysis of any plan to the case of a single while loop containing a single primitive action.

By Rule 2.4, P is epistemically feasible if T1(P) is. Applying Rule 2.3 to T1(P), and making use of our solution to the KP problem for actions from the previous section, we derive the following rule:

RULE 2.5
P is epistemically feasible for A at T if A knows at T that,

a. at each stage of the execution of P, A will know whether P has successfully finished;
b. at each stage of the execution of P, if P has not successfully finished, then A knows which specific action constitutes the next step of P; and
c. P will eventually finish successfully.

For example, the plan 'sequence(look up Ernie's number; dial Ernie's number)' is epistemically feasible for Aunt Edith because:

> Initially, Edith knows that the first step is to look up Ernie's number, which she knows how to do.
> After the first step, Edith knows that the next step is to dial Ernie's number, which she now knows how to do.
> After the first two steps, Edith knows that she has successfully finished the plan.

Note that Rule 2.5 works for the two cases mentioned above where Rules 2.1 and 2.2 fail. The plan 'sequence(take half the apples out of the barrel; take out the remaining half)' is epistemically feasible because:

> Until the barrel is empty, the agent knows that the next step is to take out an apple.
> When the barrel is empty, the agent knows that the plan has successfully finished.

[3] As we shall see in section 3.2 strictly speaking one should speak of 'the next step of P following an interval' or 'following a sequence of steps' rather than 'at an instant'. However, the latter is much less clumsy in English, so, for our informal discussion in this section we will be careless on this point.

The plan 'if Q then do f(X) else do f(X)' is epistemically feasible because the agent knows that the first step is f(X).

But now we can see that Rule 2.5 is all we need; Rule 2.5 entirely subsumes Rules 2.1 through 2.4. (We will prove this formally for Rules 2.1, 2.2, and 2.4 in Section 4; the proof of Rule 2.3 is similar.) The semantics of the planning language defines what is the next step of a sequence, a conditional, or a loop in every circumstance. Once that is defined for a plan, Rule 2.5 is sufficient to determine whether the plan is epistemically feasible.

Our final step is to turn Rule 2.5 into a definition by making it a biconditional:

DEFINITION 2.6
A determinate plan P is epistemically feasible for A at T if and only if A knows at T that,

a. at each stage of the execution of P, A will know whether P has successfully finished;

b. at each stage of the execution of P, if P has not successfully finished, then A knows which specific action constitutes the next step of P; and

c. P will eventually finish successfully.

There is also a weaker notion of epistemic feasibility that is sometimes relevant. Suppose that it is in fact the case that the directory lists my name and number, but Aunt Edith does not know this. Then an omniscient observer can see that if she attempts to execute the plan 'sequence(look up Ernie's number; dial Ernie's number)' she will succeed, but she herself does not know this. We will call such a plan *blindly epistemically feasible*; if the agent is presented with the plan and follows it in blind faith, he will get through it. The definition differs from Definition 2.6 only in dropping the outer knowledge condition:

DEFINITION 2.7
A determinate plan P is *blindly epistemically feasible* for A at T if and only if

a. at each stage of the execution of P, A will know whether P has successfully finished;

b. at each stage of the execution of P, if P has not successfully finished, then A will know which specific action constitutes the next step of P; and

c. P eventually terminates.

Definition 2.6 can now be reworded as follows: a determinate plan P is epistemically feasible for A at T if and only if A knows at T that P is blindly epistemically feasible.

This distinction becomes more important and richer in the context of indeterminate plans.

2.3 *Indeterminate plans*

An indeterminate plan is one that can be executed in more than one way. Indeterminate plans were introduced into the AI literature in NOAH [19], and have since been studied extensively [1, 21]. The indeterminacy may involve a partial ordering on the steps of the plan, or arguments to actions that are constrained but not fully bound, or simply options between two choices. In executing an indeterminate plan, there may be several options for the next step of the plan at a given moment. In some cases, some of the options at a given moment may be executable, while others may be epistemically infeasible, physically impossible, or logically impossible. (A logically impossible action is one that refers to a non-existent object, such as 'Send a letter to the King of France'.)

Defining epistemic feasibility is substantially trickier, both technically and conceptually, for indeterminate plans than for determinate plans. There is more than one concept to be considered.

We will begin by considering a version of epistemic feasibility that turns out to be the clearest conceptually, though not the most natural intuitively. The basic idea is this: Let us view an indeterminate plan P as a task that has been assigned, rather than as a guideline to follow. Some taskmaster has told the agent 'Carry out a series of actions that conforms to P' and the agent must figure out some way of doing so. If the agent can figure out a way of performing P, then we will say that P is epistemically feasible *as a task*. For example, suppose that Aunt Edith is assigned the plan, 'sequence(look up anyone you choose in the directory; dial Ernie's number)'. Edith can figure out that if she chooses to look up Ernie's number as the first step, then she will be able to carry out the plan. This plan is therefore epistemically feasible for Edith as a task; that is, Edith knows enough so that she can be sure that she will be able to perform it.

Other types of indeterminacy work the same way. Suppose that Edith is given either the plan

```
sequence(either(sing Yankee Doodle,
                look up Ernie's number);
         dial Ernie's number)
```

or the plan

```
in either order do both {dial Ernie's number;
                         look up Ernie's number }.
```

In either case, she can determine that if she first looks up Ernie's number and then dials it, she will be able to finish the plan successfully. Thus these plans are epistemically feasible as tasks.

In any of these cases, Edith knows that the determinate plan 'sequence(look up Ernie's number; dial Ernie's number)' is epistemically feasible and is a way to carry out the specified indeterminate plan. This suggests the following definition, a generalization of Rule 2.4 of the previous section:

DEFINITION 2.8
Plan P is *epistemically feasible as a task* for agent A at time T if there exists a determinate plan P' such that:

a. P' is epistemically feasible for A at T; and
b. A knows in T that an execution of P' starting in T will constitute an execution of P starting in T.

In general, if P' is a determinate plan such that an execution of P' starting in T constitutes an execution of P, we will say that P' is a *completion* of P. The form of P' may be more complex than the form of P. For example, if P is the plan 'for I := 1 to 100 do either A or B', then one completion of P is

```
for I := 1 to 100 do
    if the temperature in degrees Farenheit in the Ith largest city is
      a prime number
    then do A
    else do B.
```

There are two possible objections to Definition 2.8: first, that it is too narrow, and, second, that it is too broad. We will address them in turn.

It may seem that the definition requires too much. Why should the agent be obliged to generate from the beginning a plan P' that will take him all the way through P? Would it not be more in the spirit of Definition 2.6 to require only that he be able to see his way one step at a time?

The answer to this objection is that the definition presumes a very general sense of 'plan'. When we come to the formalization of this theory, we will define a plan extensionally as a set of intervals, and then we can use the comprehension and choice axioms of set theory to guarantee the existence of all sorts of plans. For the moment, we will illustrate the richness of the class of plans as follows. We begin by defining recursively the notion of being on the right track to finish a plan successfully.

DEFINITION 2.9
Agent A is *on the right track* for plan P at time T if either

a. A knows at T that P has finished successfully at T; or
b. there is an action E such that A knows that, if he performs E, then he will (recursively) be on the right track for P. Such an action will be called a *right move* of P for A at the time.

Furthermore, we posit the existence of a choice function, that, given a particular set of actions, returns one specific element of the the set. For example, if the set of all actions can be well-ordered, then one choice function would be to map a set of actions AA into the element lowest in the well-ordering.

We now define a transformation T2(P) that takes an arbitrary plan P into a particular completion of P. For any plan P, T2(P) is the plan

```
while ($P$ has not finished successfully)
    do (choice-function(the right moves of $P$)).
```

Clearly, under practically any circumstances where one can say that the agent knows enough to carry out P, this plan will satisfy the conditions for P' in Definition 2.8.[4]

As we shall elaborate below (Section 3.2.1), our notion of plans is rich enough to include T2(P) for any plan P.

The other objection to Definition 2.8 is that it does not capture the concept of an 'executable' plan, but rather addresses a different, much weaker notion. A plan should be something that the agent can simply execute, not something that requires the agent to think long and deeply about how to carry it out. In fact, as far as Definition 2.8 is concerned, 'achieve(G)' is a perfectly fine plan as long as the agent can figure out how to achieve G. What is wanted is the notion of a plan that the agent can execute one step at a time without thinking ahead.

This view of executability is supported not only by intuition but also by all previous work on indeterminate plans. The standard definition in the literature [1] is that an indeterminate plan is necessarily physically feasible if every completion is feasible. Definition 2.8 above admits plans that are not even considered necessarily physically feasible. The indeterminate plan 'Either wave a flag or turn water into wine' is not necessarily physically feasible, because the completion 'Turn water into wine' is impossible; however, it satisfies Definition 2.8 because the agent can choose to pick the completion 'Wave a flag'.

More or less, we are aiming for a definition roughly along the following lines:

> Plan P is executable for agent A at time T if, after each beginning of P starting in A, A knows how to perform all the possible next steps of P.

(One might suppose that the complementary notion to Definition 2.8 would be 'All completions of P are epistemically feasible'. However, this condition can only be achieved by omniscient

[4] Actually, there is a small class of anomalous plans P for which T2(P) does not necessarily terminate, but which do satisfy Definition 2.8. Thus, Definition 2.8 is slightly more general than one based around the idea of repeatedly executing a 'right move'.

agents. For example, let P be the plan 'do either A or B' and let P' be the plan, 'if Q then do A else do B'. Then P' is a completion of P; however, P' is only epistemically feasible if Q is known.)

Unfortunately, there is an inherent clash between the notion of executability that we are aiming toward here and the context of intelligent agents. After all, we are trying to define epistemic feasibility relative to an agent who knows something and can reason; in fact, we will idealize the agent as being able to do arbitrary deductive reasoning instantaneously. The notion of executability we are looking for, by contrast, assumes that the agent is not willing to look further ahead than his nose; he will blindly try to execute the next step of the plan whatever it is. Thus, we are in a certain sense trying to force the agent to be more stupid than he actually is. It is not clear that there is any particularly natural way to do this.

The result of this conceptual incoherence is that there are a variety of choices to be made in defining the above notion of executability and there is no clear model or intuition to justify one choice over another. For example,

1. Is the agent allowed to exclude next steps that are logically, physically or epistemically impossible, and to choose only among the next steps that are possible? That is, are we excluding the agent only from choosing steps on the basis of reasoning about the future, or are we also excluding from choosing steps on the basis of reasoning about the present?
2. Is the agent obliged to know about all the possible next moves? If so, is he obliged to know that the next moves he is aware of are, in fact, exhaustive? If not, is it necessary that the possible next moves he is unaware of be feasible? For example, suppose that Fred is one block away from the World Trade Center, that Fred knows that the World Trade Center is 1350 feet high, but that he does not know the heights of the Empire State Building or the Chrysler Building (1250 feet, not counting the TV tower, and 1046 feet, respectively.) Is the plan 'Take a taxi to a New York building taller than 1000 feet', to be considered inexecutable, because Fred does not know that it can be achieved by going to Empire State Building or the Chrysler Building? What about the plan 'Walk in five minutes to a New York building taller than 1000 feet', which is physically infeasible if the building is chosen to be either of the other two? What about the plan, 'Go to a New York building more than 1300 feet tall', for which Fred in fact knows all the possible next steps, but does not know that his list is exhaustive?
3. Our proposed definition says that, however the agent begins plan P, he will be able to continue it. But does this mean any beginnings whatever of P? or only if he knows that he has begun P? or only if he could have planned to begin P in the way that he has? For example, let P be the plan, 'Take the train from New Haven to a city of population greater than one million; then take a taxi to the Empire State Building'. Suppose that New York is the only city that the agent knows has a population greater than one million. Shall we say that the plan is executable, because, if he chooses to execute it, he will certainly plan to go to New York? Suppose we know that at the Philadelphia train station is a sign saying 'Welcome to Philadelphia; population > 4,000,000'. Shall we say that the plan is not executable, because he may get off at Philadelphia pursuing some other plan, realize that he has begun P, and attempt to finish P?
4. Consider the following case: Edith is sitting with Jack and Algernon. She does not know who is older. P is the plan, 'Either speak to the older of the two or speak to the younger of the two'. Edith knows, of course, that this plan is equivalent to 'Either speak to Jack or speak to Algernon'. Is P executable, because it is equivalent to an executable plan? Or is P inexecutable, because if she chooses either branch, she ends up with an inexecutable plan?

5. In the case of determinate plans, we made a distinction between plans that the agent knows are feasible, and those that are blindly feasible. How do we make that choice in this context?

Not wishing to drag the reader through a dozen different alternate definitions, and lacking a clearly defined model or intuition, we choose a single definition that is easily expressed in our formal language and that has 'nice' formal properties. Its definition in the formal language is, in face, almost exactly the same as the formal definition of 'blind epistemic feasibility', dropping the condition of determinacy and widening the notion of 'the next step of the plan' to 'the set of next steps of the plan'.

DEFINITION 2.10
A plan P is *executable*[5] for agent A at time T if and only if,

a. P terminates when executed starting in T; and
b. after any beginning of the execution of P starting in T,
b.i A will know whether P has successfully finished;
b.ii A will know of every action whether or not it is a next step of P; and
b.iii All the next steps of P are feasible.

Definition 2.10 is a comparatively narrow one, though not the narrowest possible. It gives the following solutions to the issues raised above:

1. All next steps of the plan must be logically, physically and epistemically feasible.
2. The agent must be aware of all the next steps, and aware that his knowledge is exhaustive.
3. The agent must be able to continue the plan after a beginning whether or not he could have deliberately executed it that way. We require that, whenever the agent begins the plan, he must be aware that he had begun it.
4. As long as agent can execute every action in the set of next steps, he need not be able to divide them up along the same lines as the plan.
5. We admit blind feasibility.

Some good properties enjoyed by this definition include:

- For an omniscient agent, executability reduces to necessary physical feasibility in the usual sense (Theorem 5, Section 6.5).
- Given suitable assumptions, the executability of plans is monotonic with respect to knowledge; the more the agent knows, the more plans are executable. (Theorem 3, Section 6.3).
- For determinate plans, executability is equivalent to blind epistemic feasibility.

3 Formalization

Formalizing Definitions 2.6–2.10 requires a theory with the following elements:

- A theory of time. (Section 3.1)
- A theory of plans, including the definitions of a 'plan', of the 'execution' and 'beginning' a plan, of the 'next step' of a plan, of a 'determinate' plan, and of a 'completion' of a plan. The general definitions are in Section 3.2.1. Section 3.2.2 gives a semantics supporting these concepts for two simple planning languages.

[5] This English language definition is not, of course, unambiguous relative to the issues discussed above, but it in fact corresponds closely to the formal definition (Section 3.4).

- A theory of knowledge (Section 3.3).

We use a sorted first-order logic with set theory. In our exposition below, we will state symbolically the axioms of the theory that can be expressed without set theory; those that require set theory will be expressed only in English.

3.1 Temporal theory

We use the situation calculus [11] as our temporal theory. In the situation calculus, time is construed as an directed graph whose nodes are situations (instantaneous states of the universe) and whose arcs correspond to events that transmute one situation into another. The predicate[6] 'result($S1, E, S2$)' will mean that event E changes situation $S1$ into situation $S2$. Figure 1 shows a small branching time structure for a blocks world scenario.

Branching time is used in the literature for a variety of different types of uncertainty or indeterminacy about the future. However, in this paper we will use branching only to represent the possible actions that an agent may choose between. That is, the outarcs from a given situation correspond to those actions that are feasible in the situation. Indeed, we will take this as the definition of feasibility of actions. (We take the time structure to be the given, and define other primitives in terms of it.)

DEFINITION 3.1
Action E is *feasible* in $S1$ if there is an $S2$ which is the result of performing E in $S1$.

We define a relation on situations '$S1 < S2$' (read '$S1$ precedes $S2$') to be the strict transitive closure of the result relation. We posit that the time structure is forward branching; that is, the situations preceding any situation S are totally ordered. Thus 'precedes' is a strict partial ordering. The other order relations ($\leq, >, \geq$) are defined in the usual way.

A *fluent* is a parameter whose value changes over time. A *Boolean fluent*, such as 'raining', has Boolean values; other fluents, such as 'president(usa)' take values in other spaces (in this case, the space of people). We use the predicate 'holds(S, Q)' to mean that boolean fluent Q has value TRUE in situation S. We use the function 'value_in(S, Q)' to denote the value of non-Boolean fluent Q in situation S. We will sometimes abbreviate these notations by attaching extra arguments to fluents; thus, we may write 'raining(S)' rather than 'holds(S,raining)' or 'president(usa,S)' rather than 'value_in(S,president(usa))'.

We shall be particularly interested in fluents that range over actions. For example, the term 'shake_hands(president(usa))' might denote the fluent whose value is the action of shaking hands with that person who is currently President. The value of this fluent for situations in 1986 is the action of shaking hands with Ronald Reagan, while its value for situations in 1993 is shaking hands with Bill Clinton.

In our informal discussion, we will frequently make use of *finite time intervals*. For $SA \leq SB$, the closed interval $[SA, SB]$ is the set of all points S such that $SA \leq S \leq SB$. This set of points is totally ordered; thus, slightly abusing notation, we will sometimes denote the interval $[SA, SB]$ as the tuple $\langle S_1 = SA, S_2 \ldots S_k = SB \rangle$ of the situations in order. Our symbolic axiomatization, however, does not use intervals, but just the two end-points.

We will say that action E is *executed* over interval $[S1, S2]$ iff result($S1, E, S2$).

Tables 1 and 2 give a formal axiomatization of the temporal theory.

[6] In the literature, 'result' is usually taken to be a function; however, since we will be much concerned with the case where this function is undefined, it will be easier to view it as a predicate.

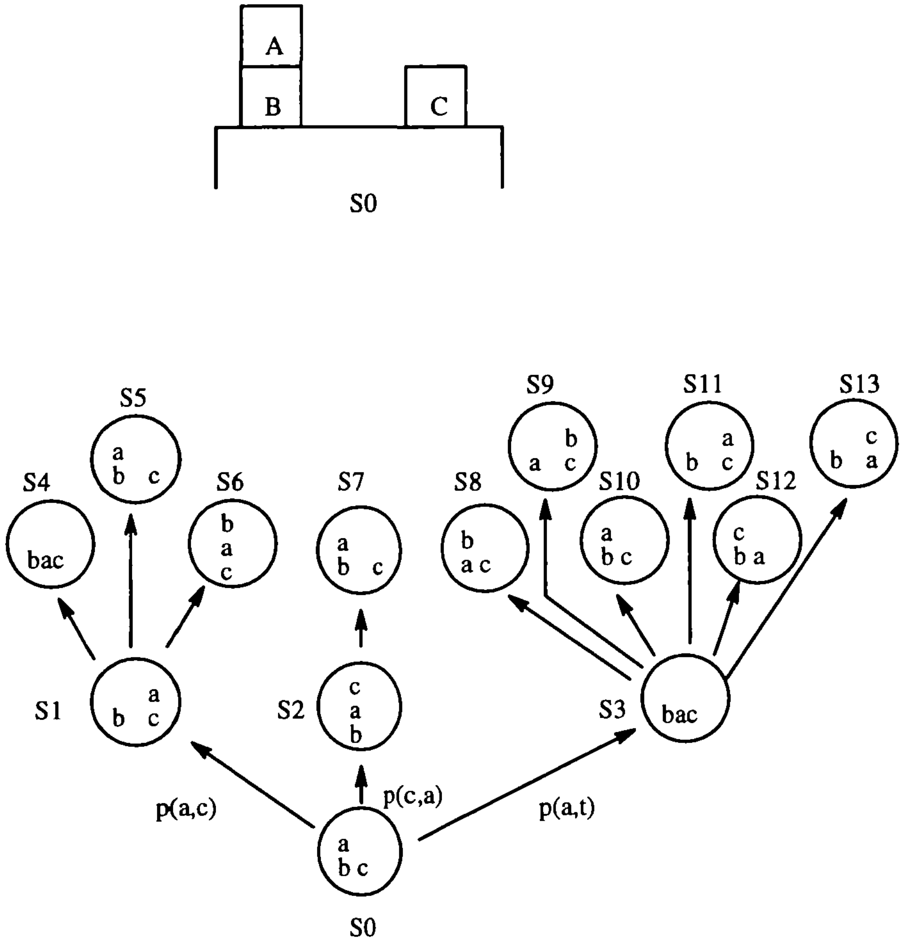


FIG. 1. Branching Time Structure. The picture at the top shows the starting situation. The state of the blocks in each situation is depicted inside the corresponding node. The arcs on the bottom level are labelled with the corresponding action; for example, the leftmost bottom arc is labelled 'p(a,c)' for 'puton(a,c)'. (Labels are omitted from the upper level arcs because of space limitations.) Each situation is labelled with an identifier for future reference.

3.2 Plans

3.2.1 General approach

The fundamental notion in our theory of planning will be the predicate 'Plan P executes over interval $[S1, S2]$'. The definition of this relation for a given plan will constitute the semantics of the plan. All other properties of plans are defined in terms of this one.

There are a few tricky issues to be dealt with. The first issue is the treatment of plans that require or permit the execution of an infeasible action. The approach that we will take here is analogous to standard practice in programming language semantics. If the plan specifies an

We use upper case italicized symbols for variables. Free variables are assumed to be universally quantified with the whole sentence as scope. The sorts of variable are indicated by their first letter.
Sorts in the temporal theory: situations (S), actions (E), fluents (Q).
Non-logical symbols:

$S1 < S2, S1 > S2, S1 \leq S2, S1 \geq S2$ — predicates. The order relations.
result($S1, E, S2$) — predicate. $S2$ results if E is performed in $S1$.
follows($S1, S2$) — predicate. $S2$ is a situation immediately following $S1$.
feasible(E, S) — predicate. Action E is feasible in situation S.
holds(S, Q) — predicate. Fluent Q holds in situation S.
value_in(S, Q) — function. The value of fluent Q in situation S.

TABLE 1. The formal language of time

infeasible action, the agent will attempt it, but will in fact execute the action 'fail'.[7]

'Fail' is an action that is always feasible. It has the effect of setting the Boolean fluent 'error' to be true. Its effects on other fluents is undefined.[8] We posit that there is no action that sets 'error' back to true. Fig. 2 shows the modified version of the branching time line in Fig. 1.

Thus, the ***execution*** of a plan may succeed or fail. If plan P executes successfully over interval $[S1, S2]$, we will say that P ***succeeds*** over $[S1, S2]$.

A second issue is the problem of infinite loops. In our theory, execution of a plan is defined only over finite intervals. Therefore, if, intuitively, a plan P goes into an infinite loop starting in $S1$, our formal statement will be that plan P does not execute any finite interval starting in $S1$. We will say that P is *vacuous* in $S1$. Thus the conditions in Definitions 2.6, 2.7, 2.8 and 2.10 that P terminate are expressed formally by requiring that P be non-vacuous. We do not distinguish here between plans that, intuitively, involve an infinite sequence of feasible actions, such as 'while (true) wave a flag' and those that, intuitively, involve an infinite sequence of impossible actions, such as, 'while (true) turn a gallon of water into wine'; both are consider equally vacuous.

A third issue is the fact that, in any given situation, there may be more than one active instance of a plan P. For example, let p0 be the plan 'For i := 1 to 50 do: take a stone out of the basket'. Let s0 be a situation in which there are 51 stones in the basket, and let the interval ⟨s0, s1 …s50⟩ be an interval in which stones are taken out of the basket one at a time. Consider now what is happening at time s50. There is an instance of p0 that started at time s0 that has just succeeded. There is also an instance of p0 that started at time s1 which has executed its first 49 steps, and which will succeed after one more step. There are also instances of p0 that started at times s2, s3 …s49 which have executed 48, 47 …1 steps. These will be able to execute one more step, but will not be able to succeed. Finally, there is an instance of p0 that is starting now at s50, which will execute one step and then get stuck. For this reason, the notions of 'executing', 'succeeding', 'beginning', and 'next step' are all defined relative to two situations: a starting situation and an ending or a reference situation.

[7] The reason to introduce this mythical action 'fail', rather than simply say that an infeasible action brings about the error state is to deal with logically impossible actions. One cannot say that the action of mailing a letter to the King of France brings about the error state, because there cannot, at the moment, be any such action. What this theory says, instead, is that, if a plan specifies 'Mail a letter to the King of France', what it really means is 'Execute fail'.

[8] That is sufficient for the purpose of this theory, since we are never interested in tracing plans past their first failure. A theory that actually dealt with agents that fail and what they do afterwards would need a more precise account, of course, since the effects of failed attempts at actions are not actually arbitrary.

Definitions:

T.1 follows$(S1, S2) \Leftrightarrow \exists_E$ result$(S1, E, S2)$.
(Definition of follows: $S2$ follows $S1$ if there is an action that turns $S1$ into $S2$.)

T.2 feasible$(E, S1) \Leftrightarrow \exists_{S2}$ result$(S1, E, S2)$.
(Definition of feasible: E is feasible in $S1$ if there is a situation $S2$ that results from performing E in $S1$.)

T.3 $S1 > S2 \Leftrightarrow S2 < S1$.
$S1 \leq S2 \Leftrightarrow S1 < S2 \vee S1 = S2$.
$S1 \geq S2 \Leftrightarrow S1 > S2 \vee S1 = S2$.
(Definition of the other order relations in terms of 'precedes'.)

Axioms:

T.4 $S1 < S2 \Rightarrow \neg(S2 < S1)$.
(Antisymmetry of 'precedes'.)

T.5 $[S1 < S2 \wedge S2 < S3] \Rightarrow S1 < S3$.
(Transitivity of 'precedes').

T.6 follows$(S1, S2) \Rightarrow S1 < S2$.
(If $S2$ follows $S1$ then $S1$ precedes $S2$.)

T.7 follows$(S1, S2) \Rightarrow \neg\exists_{SM}\ S1 < SM \wedge SM < S2$.
(Minimality of the 'follows' relation.)

T.8 The 'precedes' relation is the strict transitive closure of the 'follows' relation. That is, 'precedes' has the minimal extension consistent with T.6 and T.5.

T.9 a. $[S1 < S \wedge S2 < S] \Rightarrow [S1 \leq S2 \vee S2 \leq S1]$.
b. [follows$(S1, S)$ $\wedge$ follows$(S2, S)$] $\Rightarrow S1 = S2$.
(Forward branching of the time line. Axioms (a) and (b) are equivalent, given T.8.)

T.10 result$(S1, E, SA)$ $\wedge$ result$(S1, E, SB) \Rightarrow SA = SB$.
(An event has a unique result.)

T.11 result$(S1, EA, S2)$ $\wedge$ result$(S1, EB, S2) \Rightarrow EA = EB$.
(A unique event occurs over an atomic interval.)

TABLE 2. Axiomatization of time

Finally, it should be noted that an indeterminate plan may both succeed over $[S1, S2]$ and have a next step after $[S1, S2]$. For instance, consider the plan 'Say "Hello" two or three times'. After saying 'Hello' twice, the plan has succeeded; it also, at that point, has a next step of saying 'Hello' a third time.

We will illustrate how the predicate 'P executes over $[S1, S2]$' can be defined for two planning languages in Section 3.2.2. The definitions of the other properties of plans in terms of execution and of the error state are straightforward; they are stated in Definitions 3.2–3.10 below, and are expressed symbolically in Tables 3, 4 and 5.

DEFINITION 3.2
The extension of a plan P is the set of finite intervals over which P executes. Any set of

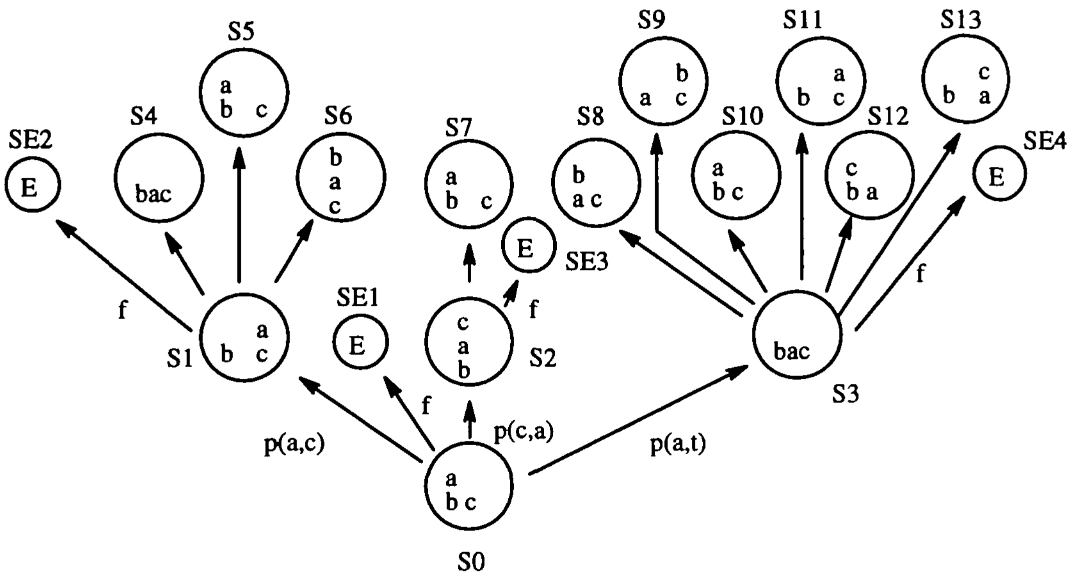


FIG. 2. Time structure with failure and error

finite intervals corresponds to a plan. Two plans that execute over the same set of intervals are considered identical.

DEFINITION 3.3
A plan P ***succeeds*** over $[S1, S2]$ if it executes over $[S1, S2]$ and 'error' does not hold in $S2$.

DEFINITION 3.4
A plan P ***begins*** over $[S1, S2]$ if

i. for some $S3 \geq S2$, P executes over $[S1, S3]$; and

ii. 'error' does not hold in $S2$.

DEFINITION 3.5
Action E is a ***possible next step*** of P after $[S1, S2]$ if

i. E is feasible in $S2$; and

ii. let $S3$ be the result of executing E in $S2$. Then the interval $[S1, S3]$ is an initial segment of an execution of P; that is, there is an $S4$ such that $S1 < S3 \leq S4$ and P executes over $[S1, S4]$.

DEFINITION 3.6
Plan P is ***determinate*** in $S1$ if there exists exactly one $S2$ such that P executes over $[S1, S2]$.

DEFINITION 3.7
Plan P is ***vacuous*** in $S1$ if there exists no $S2$ such that P executes over $[S1, S2]$. As discussed above, plans that intuitively go into infinite loops are formally considered vacuous.

DEFINITION 3.8
Plan P is ***necessarily physically feasible*** in $S1$ if P is non-vacuous in $S1$ and, for any $S2$, if P executes over $[S1, S2]$ then P succeeds over $[S1, S2]$. This is equivalent to the usual notion of

necessary feasibility of indeterminate plans [1], that every completion of the plan is feasible. Each completion corresponds to an execution; an infeasible completion corresponds to an execution that ends in failure.

DEFINITION 3.9
Plan P is *possibly physically feasible* in $S1$ if there exists an $S2$ such that P succeeds over $[S1, S2]$.

DEFINITION 3.10
Plan P' is a *specialization* of plan P in $S1$ if, for any $S2$, if P' executes over $[S1, S2]$, then P executes over $[S1, S2]$.

DEFINITION 3.11
Plan P' is a *completion* of plan P in $S1$ if P' is determinate and is a specialization of P.

As an illustration of the above definitions, consider the plan P0 =

```
        sequence(either(puton(a,c)
                        puton(a,table));
                 puton(c,b))
```

executing starting in situation S0 of Fig. 2. P0 executes over [S0,SE2] and [S0,S12]. It succeeds over [S0,S12]. It begins over [S0,S0], [S0,S1], [S0,S3], and [S0, S12]. The next steps of p0 after [S0,S0] are puton(a,c) and puton(a,table). The next step after [S0,S1] is 'fail'. The next step after [S0,S3] is puton(c,b). P0 is possibly but not necessarily physically feasible. P0 is non-vacuous and indeterminate. The plan 'sequence(puton(a,table); puton(c,b))' is a completion of P0, since its extension is just the set {[S0,S12]}, which is a subset of the extension of P0.

Finally, we define the concept of 'attempting' an action E; that is, executing E if possible, else failing. This will be useful in defining the semantics of plans with infeasible actions.

DEFINITION 3.12
An action E is *attempted* over interval $[S1, S2]$ if one of the following two conditions hold:

i. E is feasible in $S1$, and $S2$ is the result of executing E in $S1$; or

ii. E is not feasible in $S1$ and $S2$ is the result of executing 'fail' in $S1$.

3.2.2 Planning languages

What remains is to define the conditions under which plan P executes over interval $[S1, S2]$. We illustrate how this is done by defining the semantics of two simple planning languages.

In the first planning language, PLAN1, a plan is a partial ordering of steps. Each step has a *content*, which is an atomic action. This is similar to the representation of indeterminate plans in TWEAK [1], with the differences that (i) we exclude variables and (ii) we do not assume any particular representation for the preconditions and effects of actions or restrictions on them.

The semantics of PLAN1 are defined in the natural way:

DEFINITION 3.13
A plan P in PLAN1 is executed over interval $\langle S_0, S_1 \ldots S_k \rangle$ iff for some total ordering $\langle T_1 \ldots T_k \rangle$ of the steps of P, the content of T_i is attempted over $[S_{i-1}, S_i]$.

For example, consider the plan P1 consisting of two unordered steps, W1 = puton(c,a) and W2 = puton(a,table), to be executed in situation S0 of Fig. 1. The total ordering ⟨W1,W2⟩ is

Additional sorts: plans (P).
Non-logical primitives:

executes($P, S1, S2$) — predicate. Plan P executes over $[S1, S2]$.
succeeds($P, S1, S2$) — predicate. Plan P succeeds over $[S1, S2]$.
begins($P, S1, S2$) — predicate. Plan P begins over $[S1, S2]$.
next_step($E, P, S1, S2$) — predicate. Event E is a possible next step for P after $[S1, S2]$.
determinate(P, S) — predicate. Plan P is determinate in S.
vacuous(P, S) — predicate. Plan P is vacuous in S.
np-feasible(P, S) — predicate. P is necessarily physically feasible in S.
pp-feasible(P, S) — predicate. P is possibly physically feasible in S.
specialization(PR, P) — predicate. Plan PR is a specialization of P.
completion(PR, P) — predicate. Plan PR is a completion of plan P.
fail. — constant. The action of failing.
error. — constant. The fluent of being in error.

TABLE 3. The Language of plans

attempted unsuccessfully over interval ⟨S0, S2, SE3⟩. The total ordering ⟨W2,W1⟩ is attempted successfully (i.e. successfully executed) over interval ⟨S0,S3,S13⟩. Thus plan P1 executes over the two intervals [S0,SE3] and [S0,S13]; it is possibly physically feasible but not necessarily physically feasible.

The second planning language, PLAN2, is a simple ALGOL-like language with an indeterminacy operator. A plan is built up from primitive actions by applying recursive control structures to primitive statements. A primitive statement has either the form 'prim_action(E)', where E is an action, or 'prim_fluent(QE)', where QE is a fluent ranging over actions. For example, in the blocks world, 'puton(blocka,blockb)' denotes the action of putting block A onto block B, and 'prim_action(puton(blocka, blockb))' denotes the plan consisting of that single action. The term 'puton_q(top_of(stacka), top_of(stackb))' denotes the fluent whose value in any situation S is the action of putting the block that is at the top of stack A in situation S onto the block that is at the top of stack B. The term 'prim_fluent(puton_q(top_of(stacka), top_of(stackb)))' denotes the plan consisting of that single action. Here, we construe 'top_of(K)' as a function mapping a stack K to a fluent ranging over blocks, and we construe 'puton_q($Q1, Q2$)' as a function mapping two fluents ranging over blocks to a fluent ranging over actions.[9]

The control structures we will consider in PLAN2 are 'sequence($P_1 \ldots P_k$)', 'if(QB, P_1, P_2)', 'repeat(P_1, QB)', meaning 'repeat P_1 until QB', and 'indet($P_1 \ldots P_k$)', meaning 'indeterminately do either P_1 or P_2 or ...or P_k'. Here P_i is a subplan, and QB is a Boolean fluent such as 'on(blocka,blockb)' or 'raining'.

For example, the following plan moves all the blocks in stack A to stack C, and then all the blocks in stack B to stack C. (We assume that neither stack A nor stack B is initially empty.)

```
sequence(repeat(prim_fluent(puton_q(top_of(stacka),top_of(stackc)))
```

[9] It is common practice to extend automatically functions or predicates over atemporal objects to take fluents as arguments; for example, to extend the function 'puton' above to take as argument either a block or a fluent over blocks. However, since the definition of 'knowing what τ is' below depends critically on quantifying over situation-independent objects and not over fluents, we will be very rigid in distinguishing between a fluent and its value.

P.1 For any set $\mathcal{P}$ of finite intervals, there is a unique plan P such that
executes($P, S1, S2$) $\Leftrightarrow [S1, S2] \in \mathcal{P}$.
(Definition 3.2 of 'plan'.)

P.2 succeeds($P, S1, S2$) $\Leftrightarrow$ executes($P, S1, S2$) $\wedge$ $\neg$error($S2$).
(Definition 3.3 of 'succeeds'. Note the 'extra-argument' notation in 'error'.)

P.3 begins($P, S1, S2$) $\Leftrightarrow$
$\exists_{S3}\ S1 \leq S2 \leq S3 \wedge$ executes($P, S1, S3$) $\wedge$ $\neg$error($S2$).
(Definition 3.4 of 'begins'.)

P.4 next_step($E, P, S1, S2$) $\Leftrightarrow$
$\exists_{S3,S4}\ S1 \leq S2 < S3 \leq S4 \wedge$ result($S2, E, S3$) $\wedge$ executes($P, S1, S4$).
(Definition 3.5 of 'next_step'.)

P.5 determinate($P, S1$) $\Leftrightarrow \exists^1_{S2}$ executes($P, S1, S2$).
(Definition 3.6 of determinate.)

P.6 vacuous($P, S1$) $\Leftrightarrow \neg\exists_{S2}$ executes($P, S1, S2$).
(Definition 3.7 of vacuous.)

P.7 np-feasible($P, S1$) $\Leftrightarrow$
[$\neg$vacuous($P, S1$) $\wedge \forall_{S2}$ executes($P, S1, S2$) $\Rightarrow$ succeeds($P, S1, S2$)].
(Definition 3.8 of necessary physical feasibility.)

P.8 pp-feasible($P, S1$) $\Leftrightarrow \exists_{S2}$ succeeds($P, S1, S2$).
(Definition 3.9 of possible physical feasibility.)

P.9 specialization($PI, P, S1$) $\Leftrightarrow$
[$\neg$vacuous($PI, S1$) $\wedge \forall_{S2}$ executes($PI, S1, S2$) $\Rightarrow$ executes($P, S1, S2$).]
(Definition 3.10 of specialization.)

P.10 completion($PC, P, S1$) $\Leftrightarrow$ [specialization($PC, P, S1$) $\wedge$ determinate($PC, S1$)].
(Definition 3.11 of completion.)

P.11 attempt($E, S1, S2$) $\Leftrightarrow$
[result($S1, E, S2$) $\vee$ [$\neg$feasible($E, S1$) $\wedge$ result($S1$,fail,$S2$)]].
(Definition 3.12 of 'attempting' an action.)

TABLE 4. Definitions of the language of plans

```
         empty(stacka)),
     repeat(prim_fluent(puton_q(top_of(stackb),top_of(stackc)))
         empty(stackb))).
```

The semantics of PLAN2 can be defined by recurring down the form of the plan.

DEFINITION 3.14
(Semantics of PLAN2).

a. Let P=prim_action(E). P executes over $[S1, S2]$ iff E is attempted over $[S1, S2]$.

b. Let P=prim_fluent(QE). P executes over $[S1, S2]$ if value_in($S1, QE$) is attempted over $[S1, S2]$.

P.12 executes($P, S1, S2$) $\Rightarrow S1 \leq S2$.
(A constraint on language semantics: plans must execute from earlier to later times.)

P.13 $\forall_S$ feasible(fail,S).
(Failing can always occur.)

P.14 result($S1$,fail,$S2$) $\Rightarrow$ error($S2$).
(Failing results in error.)

P.15 error($S1$) $\wedge$ $S1 < S2 \Rightarrow$ error($S2$).
(Error is irrecoverable.)

P.16 [$\neg$error($S1$) $\wedge$ error($S2$) $\wedge$ $S1 < S2$] $\Rightarrow$
$\exists_{SA,SB}$ $S1 \leq SA < SB \leq S2$ $\wedge$ result(SA,fail,SB).
(Frame axiom for 'error': the error state is only entered as a result of executing 'fail'.)

TABLE 5. Axioms for the theory of plans

c. Let P=sequence($P_1 \ldots P_k$). P executes over $[S_0, S_k]$ iff there exist $S_1, S_2 \ldots S_{k-1}$ such that $S_{i-1} \leq S_i$ and such that P_i executes over $[S_{i-1}, S_i]$, for $i = 1 \ldots k$.

d. Let P=if(Q, PA, PB). P executes over $[S1, S2]$ iff one of the following holds:
- Q holds in $S1$ and PA executes over $[S1, S2]$; or
- Q does not hold in $S1$ and PB executes over $[S1, S2]$.

e. We define the semantics of the repeat loop by first giving a recursive definition of executing some of the iterations of a repeat loop, and then using that to define executing the entire loop.

e.i. Let P=repeat(PA, Q). P executes some iterations over $[S1, S2]$ iff one of the following holds:
* PA executes over $[S1, S2]$; or
* For some $S3$ such that $S1 < S3 \leq S2$, P executes some iterations over $[S1, S3]$; Q does not hold in $S3$; and PA executes over $[S3, S2]$.

e.ii. Let P=repeat(PA, Q). P executes over $[S1, S2]$ iff P executes some iterations over $[S1, S2]$ and Q holds in $S2$.

f. Let P=indet($P_1 \ldots P_k$). P executes over $[S1, S2]$ iff P_i executes over $[S1, S2]$ for some $i \in 1 \ldots k$, and none of the P_i are vacuous in $S1$.

Table 5 shows the symbolic expression of this definition.

The definition as a whole recurs down the structure of the plan. Item (e.i) also has an internal recursion through iterations of a loop down the time line.

Note that every execution of a plan takes at least one time unit. (The use of repeat rather than while loops is to insure this and thus avoid the problem of defining a semantics for a while loop with an instantaneous body.)

3.3 *Theory of knowledge*

We use a possible-worlds theory of knowledge [8]. Following Moore [13, 5], we identify a possible world with a situation; a possible world is one way the universe can be at an instant. To express facts about an agent's knowledge, we introduce the notion of a *knowledge accessibility*

Non-logical primitives:

prim_action(E) — function mapping action E to the plan of doing E.
prim_fluent(QE) — function mapping QE, a fluent over actions,
  to the plan of doing the current value of QE.
sequence($P_1 \dots P_k$) — function mapping plans $P_1 \dots P_k$
  to the plan of performing these in sequence.
if(Q, PA, PB) — function mapping Boolean fluent Q and plans PA, PB
  to the plan of doing PA if Q, else PB.
repeat(PA, Q) — function mapping plan PA and Boolean fluent Q
  to the plan of repeating PA until Q.
indet($P_1 \dots P_k$) — function mapping plans $P_1 \dots P_k$
  to the plan of doing one of them.

Axioms:

PL2.1 executes(prim_action(E),$S1, S2$) $\Leftrightarrow$ attempt($E, S1, S2$).

PL2.2 executes(prim_fluent(QE),$S1, S2$) $\Leftrightarrow$ attempt(value_in($S1, QE$),$S1, S2$).

PL2.3 executes(sequence(PA, PB),$S1, S2$) $\Leftrightarrow$
  $\exists_{SM}\ S1 \leq SM \leq S2 \wedge$ executes($PA, S1, SM$) $\wedge$ executes($PB, SM, S2$).

PL2.4 executes(if(Q, PA, PB),$S1, S2$) $\Leftrightarrow$
  [[holds($S1, Q$) $\wedge$ executes($PA, S1, S2$)] $\vee$
  [$\neg$holds($S1, Q$) $\wedge$ executes($PB, S1, S2$)]].

PL2.5 iterates(repeat(PA, Q),$S1, S2$) $\Leftrightarrow$
  [executes($PA, S1, S2$) $\vee$
  $\exists_{SM}$ iterates(repeat(PA, Q),$S1, SM$) $\wedge$ $\neg$holds(SM, Q) $\wedge$
    executes($PA, SM, S2$)].

PL2.6 executes(repeat(PA, Q), $S1, S2$) $\Leftrightarrow$
  iterates(repeat(PA, Q),$S1, S2$) $\wedge$ holds($S2, Q$).

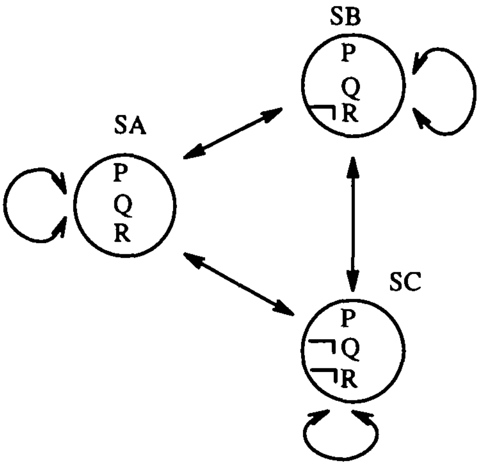
PL2.7 executes(indet(PA, PB),$S1, S2$) $\Leftrightarrow$
  [$\neg$vacuous($PA, S1$) $\wedge$ $\neg$vacuous($PB, S1$) $\wedge$
  [executes($PA, S1, S2$) $\vee$ executes($PB, S1, S2$)]].

TABLE 6. Semantics of PLAN2

relation between worlds. World s2 is knowledge accessible from world s1 if, as far as the agent[10] knows in s1, the world could just as well be in s2. The statement that A knows ϕ is thus expressed by stating that ϕ holds in every accessible world; that is, no world in which ϕ is false is consistent with what is known. Thus, the statement that the agent knows in s1 that it is currently raining is expressed by stating that in every world accessible from s1 it is raining (Fig. 3).

Following Hintikka [8], we represent, 'A knows in what τ is' using the large-scope quantification, 'There is an X such that A knows that $X = \tau$'. In a possible-worlds representation, this is expressed in the formula, 'τ has the same value in every accessible world' or, equivalently, 'There is an X such that, for every accessible world W, X is the value of τ in W.' For example,

[10] In this paper, we deal only with a single agent and hence only a single knowledge accessibility relation.



In SA:

P, Q, R are true.

P is known.

Q is not known.

R implies Q is known.

FIG. 3. System of possible worlds

the statement 'John knows what the capital of New York is, but does not know what the capital of California is', is expressed by saying that the capital of New York is the same thing (namely Albany) in all accessible worlds, but the capital of California is different things in different accessible worlds. For example, if John is uncertain whether the capital is Sacramento or Los Angeles, then there are some accessible worlds in which Sacramento is the capital and there are other accessible worlds in which Los Angeles is the capital.

We must thus extend the notion of fluent from 'something whose value can change over time' to 'something that could conceivably have another value'. For example, in a purely temporal theory, the gravitational constant would be a logical constant. However, if we wish to describe agents who do not know the value of the gravitational constant, then it must be a fluent whose value varies from one world to another.

Knowledge about past and future is expressed by combining knowledge accessibility with temporal relations. Thus, the statement, 'The agent knows in s1 that, if he picks up block A, then block B will be clear', is expressed in the formula

```
For every S2 and S3,
    if S2 is accessible from s1,
        and S3 is the result of picking up block A in S2,
    then block B is clear in S3.
```

$\forall_{S2,S3}$ k_acc(s1,$S2$) $\wedge$ result($S2$,pickup(blocka),$S3$) $\Rightarrow$ clear(blockb,$S3$).

K.1 $\forall_S$ k_acc(S, S).
(Reflexivity = veridicality.)

K.2 k_acc$(SA, SB) \land$ k_acc$(SB, SC) \Rightarrow$ k_acc(SA, SC)
(Transitivity = positive introspection.)

K.3 know_val$(Q, S) \Leftrightarrow$
$\exists_X \forall_{S2}$ k_acc$(S, S2) \Rightarrow X$=value_in(S, Q).
(Definition of knowing what Q is.)

K.4 [result$(S1A, E, S2A) \land$ k_acc$(S2A, S2B)] \Rightarrow$
$\exists_{S1B}$ result$(S1B, E, S2B) \land$ k_acc$(S1A, S1B)$.
(Axiom of memory).

K.5 k_acc$(SA, SB) \Rightarrow$ [error$(SA) \Leftrightarrow$ error(SB)].

TABLE 7. Axioms of knowledge

The overall structure of possible worlds consists of a number of parallel branching time-structures, connected by knowledge accessibility relations.

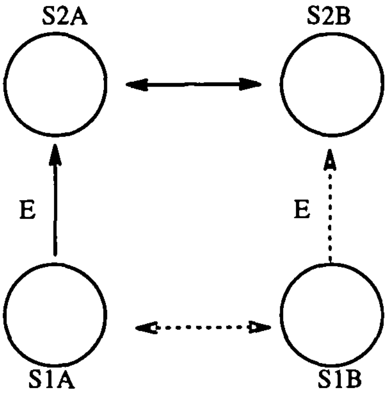
To axiomatize this theory of knowledge, we introduce the predicate 'k_acc$(S1, S2)$' (meaning situation $S2$ is knowledge accessible from situation $S1$), and posit the axioms enumerated in Table 7.

Axioms K.1 and K.2 are purely constraints on the knowledge relation. As is well known, they generate a theory of knowledge at an instant that corresponds to the modal theory S4 [7]. In particular, axiom K.1 corresponds to the axiom of veridicality, that if A knows ϕ then ϕ must be true, and axiom K.2 corresponds to the axiom of positive introspection, that if A knows ϕ then A knows that A knows ϕ. The other properties of an S4 logic are necessary consequences of the structure of a possible worlds semantics. In particular, the theory has the 'consequential closure' property, that an agent knows all necessary truths and all logical consequence of his knowledge.

Axiom K.3 expresses the definition of 'knowing what τ is' discussed above. The value of fluent Q is known in S if it has the same value in all worlds accessible from S.

Axiom K.4 relates knowledge to time. It states that an agent remembers anything he once knew and also remembers the events that have passed. (Since time structures are forward branching, there is a unique sequence of events prior to any given situation.) Specifically it states that, if $S2B$ is knowledge accessible from $S2A$, then the same event E must have preceded both situations, and the situation preceding $S2B$ was knowledge accessible from the situation preceding $S2A$ (Figure 4). Since the class of accessible situations has not increased in going from $S1A$ to $S2A$, the agent has not forgotten anything, and since the two interposing events are the same, the agent knows what it is. Using the axioms of time, one can show by induction that the agent knows all prior events and always knows as much in a later state as in previous one. (Of course, the agent may become increasingly ignorant about the *current* state. For example, an agent may know that a pair of dice is showing twelve; if he then rolls the dice but does not look, then he will not know what the dice are showing after rolling the dice. What the axiom does guarantee is that after rolling the dice he still remembers that the dice showed twelve before he rolled the dice.)

Axiom K.5 asserts that the agent always knows whether he is in the error state. This is primarily useful for deducing that he knows that he is not in the error state at the beginning of plan execution.



If $S1A$, $S2A$, and $S2B$ are connected as shown by the solid lines, then $S1B$ must exist and be connected as shown by the dotted lines.

FIG. 4. Axiom of memory

There is one further tricky issue in integrating our theory of plans with the possible worlds theory of knowledge. As discussed in Section 2.1.1, the planning concepts 'P succeeds', 'P has begun', and 'E is the next step of P'are all defined relative to an interval $[S1, S2]$. However, our language of knowledge does not allow us to express a statement like 'A knows in $S2$ that P succeeds over $[S1, S2]$' directly; it does not make any sense to say, 'In every situation accessible from $S2$ it is the case that P completes over $[S1, S2]$'.

The problem, then, is to identify a particular instance of a plan over different time structures related by knowledge accessibility relations. In general, the cross-world identification problem of events across chronicles is a difficult one [12]. Here, however, there are two very helpful restrictions on the problem:

- For the purposes of our definitions, we need only express the agent's knowledge about the state of the plan over an interval just completed. That is, we only need to say that the agent knows 'P has succeeded over $[S1, S2]$' in the situation $S2$ itself, and not in any other situation.
- By postulating (axiom T.9) that time is forward-branching, so that each situation has a unique history leading to it, and (axiom K.4) that an agent remembers all the events that have occurred, it follows that in any situation, the agent knows the entire history of the world up to this point.

In view of these two features, it is natural to posit that the agent can identify a particular instance of a plan in terms of the sequence of events that have passed since it started. We will say that interval $[S1B, S2B]$ corresponds to interval $[S1A, S2A]$ if $S2B$ is knowledge accessible from $S2A$ and the same sequence of events occurs between $S1B$ and $S2B$ as between $S1A$ and $S2A$. The axiomatic definition of 'corresponds($S1A, S2A, S1B, S2B$)' is given in axioms CO.1–CO.3, Table 8.

3.4 *Epistemic feasibility*

We can now state the formal definitions of epistemic feasibility. We reformulate the definitions of Section 2 below. Table 9 shows the symbolic axiomatization.

Non-logical primitive:

same_events($S1A, S2A, S1B, S2B$) — predicate. The same sequence of events occurs over interval $[S1A, S2A]$ as over $[S1B, S2B]$.

corresponds($S1A, S2A, S1B, S2B$) — predicate. Interval $[S1B, S2B]$ corresponds to interval $[S1A, S2A]$.

CO.1 same_events($S1A, S2A, S1B, S2B$) $\Leftrightarrow$
$[[S1A = S2A \wedge S1B = S2B] \vee$
$[\exists_{SMA,SMB,E}$ same_events($S1A, SMA, S2A, SMB$) $\wedge$ result($SMA, E, S2A$) $\wedge$
result($SMB, E, S2B$)]].
(Recursive definition of 'same_events'. The base case is the null sequence of events; the recursive case recurs down the last event in the sequence.)

CO.2 'same_events' is the minimal relation consistent with CO.1.

CO.3 corresponds($S1A, S2A, S1B, S2B$) $\Leftrightarrow$
[k_acc($S2A, S2B$) $\wedge$ same_events($S1A, S2A, S1B, S2B$)].
(Definition of 'corresponds'.)

TABLE 8. Definition of the predicate 'corresponds'

DEFINITION 3.15 (Reformulation of Definition 2.10)
A plan P is *executable* for agent A in situation $S1$ if and only if

a. P is non-vacuous in $S1$; and
b. if P begins over $[S1, S2]$ and $[S1A, S2A]$ corresponds to $[S1, S2]$ then
b.i P succeeds over $[S1A, S2A]$ if and only if it succeeds over $[S1, S2]$;
b.ii E is a next step of P after $[S1A, S2A]$ if and only if E is a next step of P after $[S1, S2]$; and
b.iii 'fail' is not a next step of P after $[S1, S2]$.

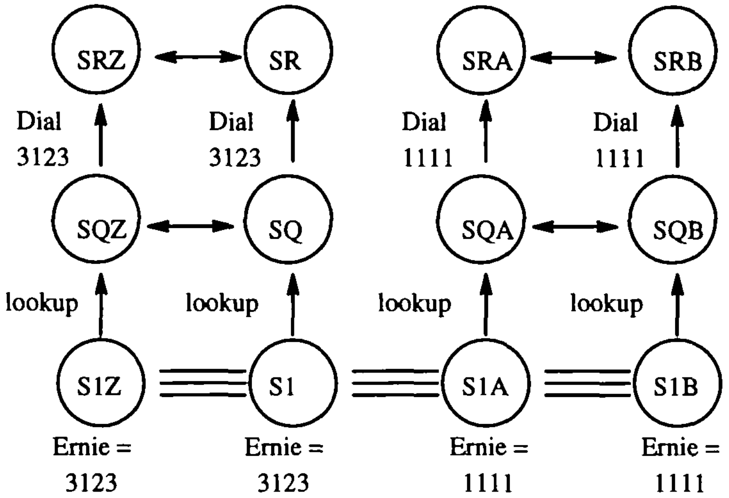
DEFINITION 3.16 (Reformulation of Definition 2.7)
A plan P is *blindly epistemically feasible* for A in $S1$ if it is determinate in $S1$ and executable for A in $S1$.

DEFINITION 3.17 (Reformulation of Definition 2.6)
Determinate plan P is epistemically feasible in $S1$ if and only if A knows in $S1$ that P is blindly epistemically feasible. Formally, if $S2A$ is knowledge accessible from $S1$, then P is blindly epistemically feasible from $S1$.

DEFINITION 3.18 (Reformulation of Definition 2.8)
Plan P is *epistemically feasible as a task* for agent A in $S1$ if there exists a plan P' such that A knows in $S1$ that:

a. P' is executable for A in $S1$; and
b. P' is a specialization of P starting in $S1$.

Formally, there exists P' such that, for any $S1A$, if $S1A$ is knowledge accessible from $S1$ then P' is executable in $S1A$ and P' is a specialization of P in $S1A$.



The figure shows part of the structure of possible worlds for the plan 'sequence(look up Ernie's number; dial Ernie's number)'. Initially, the situations S1, S1Z, S1A, S1B are all mutually knowledge accessible (indicated by the triple lines). In S1 and S1Z, Ernie's number is 3123; in S1A and S1B, it is 1111. The result of performing the action 'lookup' is that the agent learns what Ernie's number is. Thus, situations SQA and SQB are separated from SQ and SQZ. In each of these situations therefore, the next step of the plan, 'Dial Ernie's number' denotes the same action in all accessible worlds. In all worlds accessible from SQ or SQZ, it denotes the action 'Dial 3123'; in all worlds accessible from SQA or SQB, it denotes the action, 'Dial 1111'.

FIG. 5. Possible worlds structure for epistemic feasibility

It may seem odd to use 'executability' as the foundation of the definitions of the other notions of epistemic feasibility, given that, as we argued in Section 2.3, executability is a more arbitrary notion than the others. However, formally, this is the simplest direction to go.

Figure 5 illustrates the structure of temporal and possible-worlds relations involved in the example of looking up a number and dialling it.

4 Some formal proofs from our theory

In this section we will illustrate the power of the theory presented in Section 3 by proving a number of results:

I. Sample positive result: under suitable assumptions, the plan 'Look up Ernie's phone number; dial Ernie's phone number', is epistemically feasible.

II. Sample negative result: under assumptions similar to (I), the plan 'Look up Fred's phone number; dial Ernie's phone number', is not epistemically feasible.

III. Monotonicity with respect to knowledge: under suitable assumptions, the more an agent knows, the more plans are epistemically feasible.

IV. Reduction for determinate plans: if plan P is known in S to be determinate in S, P is epistemically feasible if and only if P is feasible as a task.

Non-logical primitives:

dep_feasible(P, S) — determinate plan P is epistemically feasible in S.
bep_feasible(P, S) — determinate plan P is blindly epistemically feasible in S.
task_ep_feasible(P, S) — plan P is epistemically feasible as task in S.
executable(P, S) — plan P is executable as task in S.

Definitions

EF.1. $\text{executable}(P, S1) \Leftrightarrow$
$\neg\text{vacuous}(P, S1) \wedge$
$\forall_{S2}\ \text{begins}(P, S1, S2) \Rightarrow$
$[\neg\text{next\_step}(\text{fail}, P, S1, S2) \wedge$
$\forall_{S1A,S2A}\ \text{corresponds}(S1, S2, S1A, S2A) \Rightarrow$
$[[\text{succeeds}(P, S1A, S2A) \Leftrightarrow \text{succeeds}(P, S1, S2)] \wedge$
$\forall_{E}\ \text{next\_step}(E, P, S1A, S2A) \Leftrightarrow \text{next\_step}(E, P, S1, S2)]]$.
(Definition 2.10 of executability.)

EF.3. $\text{bep\_feasible}(P, S1) \Leftrightarrow [\text{determinate}(P, S1) \wedge \text{executable}(P, S1)]$.
(Definition 2.7 of blind epistemic feasibility for determinate plans.)

EF.4. $\text{dep\_feasible}(P, S1) \Leftrightarrow$
$\forall_{S1A}\ \text{k\_acc}(S1, S1A) \Rightarrow \text{bep\_feasible}(P, S1A)$.
(Definition 2.6 of epistemic feasibility for determinate plans.)

EF.5 $\text{task\_ep\_feasible}(P, S1) \Leftrightarrow$
$\exists_{PC}\ \forall_{S1A}\ \text{k\_acc}(S1, S1A) \Rightarrow$
$[\text{specialization}(PC, P, S1A) \wedge \text{executable}(PC, S1A)]$.
(Definition 2.8 of epistemic feasibility as a task.)

TABLE 9. Definitions of epistemic feasibility

V. Reduction for omniscient agent: for an omniscient agent, plan P is executable if and only if it is necessarily feasible; and P is feasible as a task if and only if it is possibly feasible.

VI. Sufficiency of Moore's [5] rules for sequences (Rule 2.1, Section 2.2): if
A. PA is epistemically feasible in $S1$; and
B. it is known in $S1$ that, after PA completes, PB will be epistemically feasible,
then the plan 'sequence(PA, PB)' is epistemically feasible in $S1$.

VII. Sufficiency of Moore's [5] rules for conditionals (Rule 2.2, Section 2.2): If either
A. it is known that Q holds in S and PA is epistemically feasible in S; or
B. it is known that Q does not hold in S and PB is epistemically feasible in S
then the plan 'if(Q, PA, PB)' is epistemically feasible in S.

By way of comparison, in Moore's theory [5] (VI) and (VII) are axioms (for the determinate case); (I) is easily proven; (II) can be proven if Rule 2.1 is taken to be a necessary as well as

Sorts: people (X), phone numbers (N).
Domain primitives:

look_up_number(X) — function. The action of looking up the phone number of X.

numberq(X) — function. The fluent whose value in situation S is the phone number of X in S.

dial(N) — function. The action of dialling number N.

dialq(QN) — function. The fluent of dialling the current value of QN.

ernie — constant. A person.

px — constant. The plan, 'sequence(look up Ernie's number; dial Ernie's number)'.

pxa,pxb — constants. The first and second steps of px.

Axioms

A.1 $\forall_{S,X}$ feasible(look_up_number(X),S).
(Looking up a number is always physically feasible.)

A.2 $\forall_{S,N}$ feasible(dial(N),S).
(Dialing a number is always physically feasible.)

A.3 $\forall_{X,N}$ look_up_number(X) $\neq$ fail $\wedge$ dial(N) $\neq$ fail.
(Unique names assumption: neither looking up a number nor dialling is inherently failing.)

A.4 $\forall_{S,QN}$ value(dialq(QN),S) = dial(value_in(QN, S)).
(Definition of the function dialq as the natural extension of dial to fluents.)

A.5 $\forall_{S1,X,S2}$ result($S1$,look_up_number(X),$S2$) $\Rightarrow$ know_val(numberq(X),$S2$).
(Causal axiom: after you look up the number of X, you know the number of X.)

A.6 pxa = prim_action(look_up_number(ernie)) $\wedge$
pxb = prim_fluent(dialq(numberq(ernie))) $\wedge$
px = sequence(pxa,pxb).
(Definition of symbols denoting pieces of the plan.)

To prove: $\forall_S$ $\neg$error(S) $\Rightarrow$ dep_feasible(px,S).
(The plan is always epistemically feasible.)

TABLE 10. Positive example

sufficient condition; (III) and (V) can probably be proven for plans within Moore's planning language (PLAN2 without indeterminacy) by induction over the form of the plan; and (IV) is not meaningful since Moore's theory deals only with determinate plans.

4.1 *Positive result*

We wish to show that the plan 'Look up Ernie's phone number; dial Ernie's phone number', is epistemically feasible. A plausible axiomatization of the domain is given in Table 10.

Lemma 4.1

px executes by first executing the first step and then the second step.

executes(px,$S1, S2$) $\Leftrightarrow \exists_{SM}$ executes(pxa,$S1, SM$) $\wedge$ executes(pxb,$SM, S2$).

Proof. From the definition of px (A.6) and the definition of execution of a sequence (PL2.3), together with P.12. ■

Lemma 4.2

The first step pxa executes by executing the action of looking up Ernie's number.

executes(pxa,$S1, S2$) $\Leftrightarrow$ result($S1$,look_up_number(ernie),$S2$).

Proof. From the definitions of pxa (A.6), of executing a single action plan (PL2.1), of attempting an action (P.11), and the feasibility of looking up a number (A.1). ■

Lemma 4.3

The second step pxb executes by executing the action of dialling Ernie's number.

executes(pxb,$S1, S2$) $\Leftrightarrow$ result($S1$,dial(numberq(ernie,$S1$)),$S2$).

Proof. From the definitions of pxb (A.6), of executing a single fluent plan (PL2.2), of attempting an action (P.11), of the fluent function 'dialq' (A.4), and the feasibility of dialling a number (A.2). ■

Lemma 4.4

Plan px is executed by first looking up Ernie's number then dialling it.

executes(px,$S1, S2$) $\Leftrightarrow$
$\exists_{SM}$ result($S1$,look_up_number(ernie),SM) $\wedge$
result(SM,dial(numberq(ernie, SM)), $S2$).

Proof. From Lemmas 4.1, 4.2 and 4.3. ■

Lemma 4.5

If $S1$, $S2$, and $S3$ are consecutive situations, and SA comes between $S1$ and $S3$. then SA is either equal to $S1$, to $S2$, or to $S3$.

[follows($S1, S2$) $\wedge$ follows($S2, S3$) $\wedge$ $S1 \leq SA \leq S3$] $\Rightarrow$
$[SA = S1 \vee SA = S2 \vee SA = S3]$.

Proof. From the forward branching of time (T.9.a) it follows that SA must be ordered relative to $S2$. From the minimality of the 'follows' relation (T.7), it follows that SA cannot come between $S1$ and $S2$ nor between $S2$ and $S3$. The result then follows from the definition of $\leq$ (T.3). ■

Lemma 4.6

If $S1$, $S2$, and $S3$ are consecutive situations, and SA and SB are consecutive situations occurring between $S1$ and $S3$ then $[SA, SB]$ is either $[S1, S2]$ or $[S2, S3]$.

[follows($S1, S2$) $\wedge$ follows($S2, S3$) $\wedge$ $S1 \leq SA < SB \leq S3$ $\wedge$ follows(SA, SB)] $\Rightarrow$
$[[SA = S1 \wedge SB = S2] \vee [SA = S2 \wedge S3 = SB]]$.

Proof. From Lemma 4.5 and axioms T.6 and T.7. ■

Lemma 4.7

If $S3$ is the result from $S1$ of doing first $E1$ then $E2$, and EA occurs at some time between $S1$ and $S3$, then EA is either $E1$ or $E2$.

[result($S1, E1, S2$) $\wedge$ result($S2, E2, S3$) $\wedge$ $S1 \leq SA \leq SB \leq S3$ $\wedge$
result(SA, EA, SB)] $\Rightarrow$
$[EA = E1 \vee EA = E2]$.

PROOF. From Lemma 4.6 and the definition of follows (T.1), $[SA, SB]$ is equal to either $[S1, S2]$ or $[S2, S3]$. Since a unique event occurs during any time interval (T.11), EA must be equal either to $E1$ or to $E2$. ■

LEMMA 4.8
Executing px does not bring about the error state.

$$[\neg\text{error}(S1) \wedge \text{executes}(\text{px}, S1, S2)] \Rightarrow \neg\text{error}(S2).$$

PROOF. From Lemmas 4.4 and 4.7 and the fact (A.3) that neither looking up a number nor dialling it is the failure act, it follows that no failure occurs between $S1$ and $S2$. By the frame axiom for error (P.16), error cannot come about unless a failure occurs. ■

LEMMA 4.9
All executions of px are successful.

$$\text{succeeds}(\text{px}, S1, S2) \Leftrightarrow [\neg\text{error}(S1) \wedge \text{executes}(\text{px}, S1, S2)].$$

PROOF. From the definition of succeeds (P.2) and Lemma 4.8. ■

LEMMA 4.10
The only beginnings of the execution of px are the null beginning; the execution of the first step of px; and the execution of all of px.

$$\begin{aligned}&\text{begins}(\text{px}, S1, S2) \Leftrightarrow\\ &[\neg\text{error}(S1) \wedge\\ &\ [S1 = S2 \vee \text{result}(S1, \text{look\_up\_number}(\text{ernie}), S2) \vee \text{executes}(\text{px}, S1, S2)]].\end{aligned}$$

PROOF. From the definition of begins (P.3) together with Lemmas 4.4, 4.5 and 4.8. ■

LEMMA 4.11
The only next steps of px are looking up Ernie's number after the null beginning; and dialling Ernie's number after looking it up.

$$\begin{aligned}&\neg\text{error}(S1) \Rightarrow\\ &[\text{next\_step}(E, \text{px}, S1, S2) \Leftrightarrow\\ &\ [[S1 = S2 \wedge E{=}\text{look\_up\_number}(\text{ernie})] \vee\\ &\ \ [\text{result}(S1, \text{look\_up\_number}(\text{ernie}), S2) \wedge E{=}\text{dial}(\text{numberq}(\text{ernie}, S2))]]].\end{aligned}$$

PROOF. From the definition of 'next_step' (P.4), together with Lemmas 4.4 and 4.6. ■

LEMMA 4.12
Plan px is never vacuous.

$$\neg\text{vacuous}(\text{px}, S).$$

PROOF. From Lemma 4.4, the feasibility of looking up and dialling (A.1, A.2), and the definitions of feasibility of actions (T.2) and of vacuity of plans (P.6). ■

LEMMA 4.13
If $[S1A, S2A]$ corresponds to $[S1, S2]$, then the two events $E1, E2$ occur over $[S1, S2]$ just if the same events occur over $[S1A, S2A]$.

$$\begin{aligned}&\text{corresponds}(S1, S2, S1A, S2A) \Rightarrow\\ &\forall_{E1,E2}\ [\exists_{SM}\ \text{result}(S1, E1, SM) \wedge \text{result}(SM, E2, S2)] \Leftrightarrow\\ &\qquad [\exists_{SMA}\ \text{result}(S1A, E1, SMA) \wedge \text{result}(SMA, E2, S2A)]\end{aligned}$$

Proof. From the definition of corresponds (CO.1, CO.2, CO.3). Strictly speaking, this requires a second-order argument. ■

Lemma 4.14
If $S1$, $S2$, and $S3$ are consecutive situations, then $S3$ is not equal to $S1$ and $S3$ does not immediately follow $S1$.

$[\text{follows}(S1, S2) \wedge \text{follows}(S2, S3)] \Rightarrow [\, S1 \neq S3 \wedge \neg\, \text{follows}(S1, S3)]$.

Proof. From the temporal axioms T.4, T.5, T.6, and T.7. ■

Lemma 4.15
The agent knows that he has not executed px over a null interval or over an interval with only one step.

$[[S1 = S2 \vee \text{result}(S1, E, S2)] \wedge \text{corresponds}(S1, S2, S1A, S2A)] \Rightarrow$
$\neg\text{succeeds}(\text{px}, S1A, S2A)$.

Proof. By Lemmas 4.4 and 4.9, px can only succeed over an interval with two steps. By Lemma 4.13, an interval with two steps can only correspond to another interval with two steps, and by Lemma 4.14, an interval with two steps cannot also be a null interval or an interval with one step. ■

Lemma 4.16
If px is successfully executed from $S1$ to $S2$, then the agent knows that it has been successfully executed.

$[\text{succeeds}(\text{px}, S1, S2) \wedge \text{corresponds}(S1, S2, S1A, S2A)] \Rightarrow$
$\text{succeeds}(\text{px}, S1A, S2A)$.

Proof. By Lemma 4.4, if px executed from $S1$ to $S2$ then, for some middle situation SM, look_up_number(ernie) executed over $[S1, SM]$ and dial(n1) executed over $[SM, S2]$, where n1 = numberq(ernie,SM), Ernie's number in SM. By Lemma 4.13, if $[S1A, S2A]$ corresponds to $[S1, S2]$, then there is a middle situation SMA such look_up_number(ernie) executes over $[S1A, SMA]$ and dial(n1) executes over $[SMA, S2A]$. (Note that it is *not* necessarily the case, at this point in the argument that n1 is also Ernie's number in SMA; this has to be established.) By axioms T.1 and T.9.a, SMA is the unique situation preceding $S2A$. By axiom K.4, since $S2A$ is accessible from $S2$, SMA is accessible from SM. By the causal rule A.5, that one knows a number one has looked up, and the definition of know_val (K.3), the value of numberq(ernie) is the same in all worlds accessible from SM. In particular, it is the same in SMA as in SM. Thus, n1 = numberq(ernie,SMA) and dial(n1) = dialq(numberq(ernie),SMA) (A.4). Therefore, by Lemma 4.4 and 4.9, px succeeds over the interval $[S1A, S2A]$. ■

Lemma 4.17
After each beginning of px, the agent knows whether it has succeeded.

$[\text{begins}(\text{px}, S1, S2) \wedge \text{corresponds}(S1, S2, S1A, S2A)] \Rightarrow$
$[\text{succeeds}(\text{px}, S1, S2) \Leftrightarrow \text{succeeds}(\text{px}, S1A, S2A)$.

Proof. Combining Lemmas 4.10, 4.15 and 4.16. ■

Lemma 4.18
The first step(s) of px is the same in all possible worlds.

$[\neg\text{error}(SA) \wedge \neg\text{error}(SB)] \Rightarrow$
$[\text{next\_step}(E, \text{px}, SA, SA) \Leftrightarrow \text{next\_step}(E, \text{px}, SB, SB)]$.

PROOF. By Lemma 4.11, looking up Ernie's number is always the only first step of px. ∎

LEMMA 4.19

After looking up Ernie's number, the agent will know what the next step of plan px is.

$$[\neg\text{error}(S1A) \wedge \text{result}(S1A,\text{look\_up\_number(ernie)},S2A) \wedge \text{corresponds}(S1A, S2A, S1B, S2B)] \Rightarrow [\text{next\_step}(E,\text{px},S1A, S2A) \Leftrightarrow \text{next\_step}(E,\text{px},S1B, S2B)].$$

PROOF. By the definition of 'corresponds' (CO.1, CO.2, CO.3), look_up_number(ernie) executes over $[S1B, S2B]$ as well. By Lemma 4.11 the only next step of px after $[S1A, S2A]$ is dialling the value of Ernie's number in $S2A$, and the only next step of px after $[S1B, S2B]$ is dialling the value of Ernie's number in $S2B$. By causal rule A.5, the value of Ernie's number is the same in all worlds accessible from $S2A$; hence, it is the same in $S2A$ as in $S2B$ (A.1, CO.3). Thus, there is one action which is the unique next step of px both after $[S1A, S2A]$ and after $[S1B, S2B]$. ∎

LEMMA 4.20

There is no next step of px after it succeeds.

$$\text{succeeds}(\text{px},S1, S2) \Rightarrow \neg\exists_E\ \text{next\_step}(E,\text{px},S1, S2).$$

PROOF. Immediate from Lemma 4.11, together with Lemmas 4.4, 4.9 and 4.14. ∎

LEMMA 4.21

After each beginning stage of px, the agent knows what the next steps of px are.

$$[\text{begins}(\text{px},S1A, S2A) \wedge \text{corresponds}(S1A, S2A, S1B, S2B)] \Rightarrow \forall_E\ \text{next\_step}(E,\text{px},S1A, S2A) \Leftrightarrow \text{next\_step}(E,\text{px},S1B, S2B).$$

PROOF. Combining Lemma 4.10 with Lemmas 4.18, 4.19 and 4.20. ∎

LEMMA 4.22

'Fail' is never the next step of the plan.

$$\text{begins}(\text{px},S1, S2) \Rightarrow \neg\text{next\_step}(\text{fail},\text{px}, S1, S2).$$

PROOF. From Lemma 4.11 and axiom A.3. ∎

LEMMA 4.23

px is always executable.

$$\text{executable}(\text{px},S1).$$

PROOF. Combining Lemmas 4.17, 4.21 and 4.22 with the definition of executability EF.1. ∎

LEMMA 4.24

px is always determinate.

$$\text{determinate}(\text{px},S1).$$

PROOF. From Lemma 4.4, which characterizes the execution condition of px as the sequence of two determinate actions; axiom T.10, which posits that an action has a unique effect; and the definition of determinate, P.5. ∎

LEMMA 4.25

px is always blindly epistemically feasible.

$$\text{bep\_feasible}(\text{px},S).$$

PROOF. From Lemmas 4.23 and 4.24 with the definition of blind epistemic feasibility EF.3. ∎

THEOREM 4.26

Plan px is always epistemically feasible.

$$\text{dep\_feasible}(\text{px},S).$$

PROOF. From Lemma 4.25 and definition EF.4. ∎

Additional domain primitives:

fred — constant. A person.
py — constant. The plan, 'Look up Fred's number; dial Ernie's number'. pya,pyb — constants. The first and second steps of py.

Axioms, in addition to A.1 - A.6.

B.1 [¬know_val(numberq(X),$S1$) ∧ result($S1, E, S2$) ∧
know_val(numberq(X),$S2$)] ⇒
E=look_up_number(X).
(Frame axiom: the only way to find out a telephone number is to look it up. Frame axioms over knowledge are considered at length in [17] and [20]. The axiom here can be derived as special case of the 'Successor State Axiom for K' of [20].)

B.2 $X \neq Y \Rightarrow$ look_up_number(X) $\neq$ look_up_number(Y).
(Looking up X's number is a different action from looking up Y's number if $X \neq Y$.)

B.3 $N1 \neq N2 \Rightarrow$ dial($N1$) $\neq$ dial($N2$).
(Dialing $N1$ and dialling $N2$ are different actions if $N1 \neq N2$.)

B.4 ernie ≠ fred.

B.5 ¬know_val(numberq(ernie),s1).
(In situation s1, the agent does not know Ernie's number).

B.6 py=sequence(pya,pyb).
pya = prim_action(look_up_number(fred)).
pyb = prim_fluent(dialq(numberq(ernie))).
(Definition of symbols denoting pieces of the plan.)

B.7 ¬error(s1).

To prove: ¬dep_feasible(py,s1).
(The plan is not epistemically feasible.)

TABLE 11. Axioms for negative result

4.2 *Negative result*

We wish to show that the plan. 'Look up Fred's phone number; dial Ernie's phone number', is not epistemically feasible. The domain is the same as in the previous section. The additional symbols and axioms needed are shown in Table 11.

LEMMA 4.27
Plan py executes if the agent first looks up Fred's number and then dials Ernie's number.

executes(py,$S1, S2$) ⇔
$\exists_{SM}$ result($S1$,look_up_number(fred),SM) ∧
result(SM,dial(numberq(ernie,SM)),$S2$).

PROOF. Analogous to Lemma 4.4. ■

LEMMA 4.28
Executing py does not bring about the error state.

[¬error($S1$) ∧ executes(py,$S1, S2$)] ⇒ ¬error($S2$).

PROOF. Analogous to Lemma 4.8. ■

LEMMA 4.29
The only beginnings of the execution of py are the null beginning; the execution of the first step of py; and the execution of all of py.

$$\begin{aligned}&\text{begins}(\text{py}, S1, S2) \Leftrightarrow\\&[\neg\text{error}(S1) \land\\&\ [S1 = S2 \lor \text{result}(S1, \text{look\_up\_number}(\text{fred}), S2) \lor \text{executes}(\text{py}, S1, S2)]].\end{aligned}$$

PROOF. Analogous to Lemma 4.10. ■

LEMMA 4.30
The only next steps of py are looking up Fred's number after the null beginning; and dialling Ernie's number after looking up Fred's.

$$\begin{aligned}&\neg\text{error}(S1) \Rightarrow\\&[\text{next\_step}(E, \text{py}, S1, S2) \Leftrightarrow\\&\ [[S1 = S2 \land E = \text{look\_up\_number}(\text{fred})] \lor\\&\ \ [\text{result}(S1, \text{look\_up\_number}(\text{fred}), S2) \land E = \text{dial}(\text{numberq}(\text{ernie}, S2))]]].\end{aligned}$$

PROOF. Analogous to Lemma 4.11. ■

LEMMA 4.31
There is a situation that results from s1 after looking up Fred's number.

$\exists_{S2}\ \text{result}(\text{s1}, \text{look\_up\_number}(\text{fred}), S2)$.

PROOF. From the feasibility of looking up a number (A.1) and the definition of feasible (T.2). ■

We will call this situation 's2'.

LEMMA 4.32
Plan py begins over [s1,s2].

$\text{begins}(\text{py}, \text{s1}, \text{s2})$.

PROOF. From Lemmas 4.29 and 4.31. ■

LEMMA 4.33
If $[S1A, S2A]$ corresponds to [s1,s2], then the action 'Look up Fred's number' occurs over $[S1A, S2A]$.

$\text{corresponds}(\text{s1}, \text{s2}, S1A, S2A) \Rightarrow \text{result}(S1A, \text{look\_up\_number}(\text{fred}), S2A)$.

PROOF. From Lemma 4.31 and the definition (CO.1, CO.2, CO.3) of corresponds. ■

LEMMA 4.34
The next step of py after an interval $[S1A, S2A]$ corresponding to [s1,s2] is to dial Ernie's number.

$$\begin{aligned}&[\text{correspond}(\text{s1}, \text{s2}, S1A, S2A) \land \text{next\_step}(E, \text{py}, S1A, S2A)] \Rightarrow\\&E = \text{dial}(\text{numberq}(\text{ernie}, S2A)).\end{aligned}$$

PROOF. From Lemmas 4.30 and 4.33. ■

LEMMA 4.35
Ernie's number is not known in s2.

$\neg\ \text{know\_val}(\text{numberq}(\text{ernie}), \text{s2})$.

PROOF. From the frame axiom on not knowing phone numbers (B.1), the definition of s2 (Lemma 4.31), the fact that the agent does not originally know Ernie's number (B.5), and the fact that looking up Fred's number is not the same as looking up Ernie's number (B.2 and B.4). ∎

LEMMA 4.36
There is a possible world $S2A$ which is knowledge accessible from s2 such that dialling Ernie's number in $S2A$ is different from in s2. That is, the agent does not know in s2 what action is dialling Ernie's number.

$$\exists_{S2A}\ \mathrm{k\_acc}(\mathrm{s2},S2A) \wedge \mathrm{dial}(\mathrm{numberq}(\mathrm{ernie},S2A)) \neq \mathrm{dial}(\mathrm{numberq}(\mathrm{ernie},\mathrm{s2})).$$

PROOF. From the definition of 'know_val' (K.3), the fact that dialling two different numbers is two different actions (B.3), and the fact that the agent does not know Ernie's number in s2 (Lemma 4.35). ∎

LEMMA 4.37
There is an interval $[S1A, S2A]$ corresponding to [s1,s2] such that the next step of py after [s1,s2] is not a next step of py after $[S1A, S2A]$.

$$\exists_{S1A,S2A,E}\ \mathrm{corresponds}(\mathrm{s1},\mathrm{s2},S1A,S2A) \wedge \mathrm{next\_step}(E,\mathrm{py},\mathrm{s1},\mathrm{s2}) \wedge \neg\mathrm{next\_step}(E,\mathrm{py},S1A,S2A).$$

PROOF. From the axiom of memory (K.4), together with Lemmas 4.34 and 4.36. ∎

LEMMA 4.38
Plan py is not executable in s1.
¬executable(py,s1).

PROOF. From Lemmas 4.32 and 4.37 together with the definition of executability (EF.1). ∎

THEOREM 4.39
Plan py is not epistemically feasible in s1.

$$\neg\mathrm{dep\_feasible}(\mathrm{py},\mathrm{s1}).$$

PROOF. From Lemma 4.38 and definitions EF.2, EF.3. ∎

4.3 *Monotonicity with respect to knowledge*

We next prove that, under suitable assumptions, the more an agent knows the more plans are executable. (Throughout this section, we will use the phrase 'A is more than B' to mean 'A is a (possibly improper) superset of B'.)

To justify this conclusion, we need to impose three conditions. First, we must restrict attention to *knowledge-independent* plans; that is, plans whose execution conditions are independent of the knowledge of the agent.[11] A plan like, 'If you know who was the first President then raise him from the dead else wave a flag', is executable only if you do not know who was the first President; we must therefore exclude these.

Second, we must assume that what an agent knows does not affect physical laws. Specifically, we must assume that whether a given sequence of physical actions is feasible does not depend on what the agent knows. If it were the case that putting A onto C were feasible only if the agent did not know who was the first President, then the desired result would not hold.

Third, we must assume that knowing more initially cannot lead to knowing less later, other things being equal. If the world were such that looking up Ernie's number caused one to know

[11] David Etherington pointed this out to me.

Ernie's number only if one did not previously know who was the first President, then the plan 'sequence(look up Ernie's number; dial Ernie's number)' would be feasible only if the agent did not know who was the first President.

To state these conditions formally, we introduce the predicate 'same_phys($S1, S2$)', meaning that possible worlds $S1$ and $S2$ are the same in all physical respects. This property satisfies the conditions of axioms C.3–C.6 of Table 12. (For some purposes, it may be desirable or convenient to define two situations to be physically the same iff all fluents have the same value in the two situations.)

We define an agent as 'knowing more' in SA than in SB if the worlds accessible from SA are a subset of those accessible from SB. Since knowledge accessibility is dual to knowledge, this condition means that what is known in SB is a subset of what is known in SA. It should be noted that this is quite a strong notion. In particular, suppose that we posit the property of *negative introspection*, that if the agent does not know ϕ, then he knows that he does not know ϕ. In such a logic, it is impossible for an agent to know strictly more in one situation than in another, since, if he knows ϕ in SA but not in SB, then he knows that he does not know ϕ in SB but not in SA. Similarly, the strength of this notion means that the monotonicity property for epistemic feasibility as a task is trivial. To say that P is epistemically feasible as a task in SB means that the agent knows the fact 'Plan PC is a specialization of P and is executable' in SB. If he knows more in SA than in SB then he must know the same fact 'Plan PC is a specialization of P and is executable' in SA.

Table 12 gives the formal statement of the axioms we need and of the result to be proven.

LEMMA 4.40
If $S1A$ and $S1B$ are physically the same and P is knowledge independent, then the successful completions, beginnings, and next steps of P starting in $S1A$ match those starting in $S1B$.

$$\begin{aligned}
&[\text{same\_phys}(S1A, S1B) \wedge \text{k\_independent}(P)] \Rightarrow \\
&\forall_{S2A>S1A}\, \exists_{S2B}\ \text{same\_events}(S1A, S2A, S1B, S2B) \wedge \\
&\qquad [\text{succeeds}(P, S1A, S2A) \Leftrightarrow \text{succeeds}(P, S1B, S2B)] \wedge \\
&\qquad [\text{begins}(P, S1A, S2A) \Leftrightarrow \text{begins}(P, S1B, S2B)] \wedge \\
&\qquad [\text{next\_step}(E, P, S1A, S2A) \Leftrightarrow \text{next\_step}(E, P, S1B, S2B)].
\end{aligned}$$

PROOF. By C.5, the same sequences of events are feasible and lead to physically identical results, which, by C.4, includes the same value of the error fluent. By C.1, the execution intervals of P must therefore match. The result then follows from the definitions of succeeds, begins, and next_step (P.2, P.3, P.4). ∎

LEMMA 4.41
If the agent knows more in $S1A$ than in $S1B$ then after the passage of a fixed series of events, there are more intervals corresponding to $[S1B, S2B]$ than to $[S1A, S2A]$.

$$\begin{aligned}
&[\text{know\_more}(S1A, S1B) \wedge \text{same\_events}(S1A, S2A, S1B, S2B) \wedge \\
&\quad \text{corresponds}(S1A, S2A, S1C, S2C)] \Rightarrow \\
&\text{corresponds}(S1B, S2B, S1C, S2C).
\end{aligned}$$

PROOF. By C.6, every world accessible from $S1A$ is also accessible from $S2A$. The result then follows from the definition of 'corresponds' (CO.3) together with the fact that 'same_events' is an equivalence relation on intervals (CO.1, CO.2). ∎

THEOREM 4.42
In two physically identical situations, the more you know, the more knowledge-independent plans are executable.

Primitives

same_phys($S1, S2$) — predicate. $S1$ and $S2$ are physically the same.
k_independent(P) — predicate. Plan P is knowledge independent.
know_more($S1, S2$) — predicate. The agent knows more in situation $S1$ than in $S2$.

Definitions

C.1 k_independent(P) $\Leftrightarrow$
$\forall_{S1A,S1B,S2A,S2B}$
[same_phys($S1A, S1B$) $\wedge$ same_events($S1A, S2A, S1B, S2B$)] $\Rightarrow$
[executes($P, S1A, S2A$) $\Leftrightarrow$ executes($P, S1B, S2B$)].
(Definition of knowledge independence: P is knowledge independent if given two intervals with physically the same starting point and the same sequence of events, either both or neither are executions of P.)

C.2 know_more(SA, SB) $\Leftrightarrow$ [$\forall_{SC}$ k_acc(SA, SC) $\Rightarrow$ k_acc(SB, SC)].
(Definition of knowing more: the agent knows more in SA than in SB if the situations accessible in SA are a subset of those accessible in SB.)

Axioms

C.3 same_phys(S, S).
same_phys(SA, SB) $\Rightarrow$ same_phys(SB, SA).
same_phys(SA, SB) $\wedge$ same_phys(SB, SC) $\Rightarrow$ same_phys(SA, SC).
(Physical sameness is an equivalence relation.)

C.4 same_phys(SA, SB) $\Rightarrow$ [error(SA) $\Leftrightarrow$ error(SB)].
(Two situations physically the same have the same value of the error fluent.)

C.5 [same_phys($S1A, S1B$) $\wedge$ $S1A < S2A$] $\Rightarrow$
$\exists_{S2B}$ same_events($S1B, S2B$) $\wedge$ same_phys($S1B, S2B$).
(If $S1B$ is physically the same as $S1A$ then any sequence of events feasible in $S1A$ is also feasible in $S1B$ and the two resultant situations are physically the same.)

C.6 [same_phys($S1A, S1B$) $\wedge$ know_more($S1A, S1B$) $\wedge$
same_events($S1A, S2A, S1B, S2B$)] $\Rightarrow$
know_more($S2A, S2B$).
(If the agent knows more in $S1A$ than in $S1B$ but the two situations are physically the same, then, after the passage of the same sequence of events, the agent will still know more in the situation after $S1A$ than in the situation after $S1B$.)

To prove:
[same_phys(SA, SB) $\wedge$ k_independent(P) $\wedge$
know_more(SA, SB) $\wedge$ executable(P, SB)] $\Rightarrow$
executable(P, SA).

(In two physically identical situations, the more you know, the more knowledge-independent plans are executable.)

TABLE 12. Axioms of knowledge-independent physics and plans

$[\text{same\_phys}(SA, SB) \wedge \text{k\_independent}(P) \wedge$
$\text{know\_more}(SA, SB) \wedge \text{executable}(P, SB)] \Rightarrow$
$\text{executable}(P, SA)$.

PROOF. By Lemma 4.41, the intervals corresponding to a beginning from SA are a subset of those corresponding to a beginning from SB. Therefore, given the condition (part of the definition of executability EF.1), 'The intervals corresponding to a particular beginning of P from SB are either all or none successful executions of P and all have the same next steps', this same condition must hold for the intervals corresponding to the matching beginning of P from SA. Moreover, by Lemma 4.40, the condition that 'fail' is never a next step of P must hold for executions beginning from SA if it holds for executions beginning from SB. Thus, all the conditions of executability (EF.1) are met. ■

4.4 *Reduction for determinate plans*

In this section we show that, if plan P is known in S to be determinate in S, P is epistemically feasible if and only if P is feasible as a task.

LEMMA 4.43
If determinate plan P is epistemically feasible then it is epistemically feasible as a task.
$[\text{dep\_feasible}(P, S1) \Rightarrow \text{task\_ep\_feasible}(P, S1)]$.

PROOF. Choosing P as its own specialization PC, it is trivial to check that the conditions of definition EF.4 are satisfied. ■

LEMMA 4.44
If P is determinate and PS is a specialization of P starting in $S1$, then they occur over the same interval $[S1, S2]$.
$[\text{determinate}(P, S1) \wedge \text{specialization}(PS, P, S1)] \Rightarrow$
$[\text{executes}(P, S1, S2) \Leftrightarrow \text{executes}(PS, S1, S2)]$.

PROOF. From the definitions of determinate and specialization (P.5,P.9). ■

LEMMA 4.45
If it is known that PE executes from the current state if and only if PF does, then PE is executable if and only if PF is.
$[\forall_{S1A,S2A}\ \text{k\_acc}(S1, S1A) \Rightarrow$
$[\text{executes}(PE, S1A, S2A) \Leftrightarrow \text{executes}(PF, S1A, S2A)]] \Rightarrow$
$[\text{executable}(PE, S1) \Leftrightarrow \text{executable}(PF, S1)]$

PROOF. For any $S2 > S1$, let $[S1A, S2A]$ be any interval corresponding to $[S1, S2]$. By the axiom of memory K.4, $S1A$ is knowledge accessible from $S1$. Therefore it is easily checked that

- PF begins over $[S1A, S2A]$ iff PE does;
- E is a next step for PF after $[S1A, S2A]$ iff it is a next step for PE;
- PF succeeds over $[S1A, S2A]$ iff PE succeeds.

The result then follows from the definition of executability (EF.1). ■

Note that Lemma 4.45 is the formal statement of Rule 2.4 of Section 2.2 for executability.

LEMMA 4.46
If P is epistemically feasible as a task in $S1$ and P is known to be determinate in $S1$ then P is epistemically feasible in $S1$.

$$[\text{task\_ep\_feasible}(P, S1) \wedge [\forall_{S1A}\ \text{k\_acc}(S1, S1A) \Rightarrow \text{determinate}(P, S1A)]] \Rightarrow \text{dep\_feasible}(P, S1).$$

PROOF. By definition EF.4, there is a plan PC that is known to be a specialization of P in $S1$ and that is known to be executable in $S1$. By Lemma 4.44, PC and P execute over the same intervals starting in situations knowledge accessible from $S1$. By Lemma 4.45, since PC is executable in each such situation, so is P. Thus, by definitions EF.2 and EF.3, P is epistemically feasible. ∎

THEOREM 4.47
If plan P is known to be determinate, then it is epistemically feasible as a task if and only if it is epistemically feasible.

$$[\forall_{S1A}\ \text{k\_acc}(S1, S1A) \Rightarrow \text{determinate}(P, S1A)]] \Rightarrow [\text{task\_ep\_feasible}(P, S1) \Leftrightarrow \text{dep\_feasible}(P, S1).$$

PROOF. From Lemmas 4.43 and 4.46. ∎

4.5 Reduction for omniscient agent

In this section we show that for an omniscient agent, plan P is executable if and only if it is necessarily feasible; and P is feasible as a task if and only if it is possibly feasible.

We begin by defining an omniscient agent as one for whom the only situation accessible from S is S itself. Since we are not explicitly representing agents, 'omniscience' becomes a propositional atom.

DEFINITION 4.48
omniscient $\Leftrightarrow \forall_{SA,SB}\ \text{k\_acc}(SA, SB) \Rightarrow SA = SB$.

THEOREM 4.49
For an omniscient agent, plan P is executable if and only if it is necessarily feasible.

$$\text{omniscient} \Rightarrow [\text{executable}(P, S) \Leftrightarrow \text{np\_feasible}(P, S)].$$

PROOF. Given the omniscience condition, 'corresponds$(S1, S2, S1A, S2A)$' reduces to '$S1 = S1A \wedge S2 = S2A$' (CO.3), and the biconditionals in definition EF.1 thus become trivial. Definition EF.1 thus boils down to

$$\text{executable}(P, S1) \Leftrightarrow [\forall_{S2}\ \text{begins}(P, S1, S2) \Rightarrow \neg\text{next\_step}(\text{fail}, P, S1, S2)].$$

It is easily shown that this is equivalent to the definition of necessary physical feasibility, given the frame and causal axioms on the error condition (P.14, P.16), that state that error comes about if and only if a fail action is executed; and given the fact that execution takes place over finite intervals (P.12, T.8), so that, in any execution with a failure there must be a first failure. ∎

LEMMA 4.50
A plan is possibly physically feasible just if it has a specialization that is necessarily physically feasible.

$$\text{pp\_feasible}(P, S) \Leftrightarrow \exists_{PD}\ \text{specialization}(PD, P, S) \wedge \text{np\_feasible}(PD, S).$$

PROOF. Let $S_P = \{[S1, S2] \mid \text{succeeds}(P, S1, S2)\ \}$be the set of all intervals over which P succeeds. Since P is possibly feasible in S, S_P contains an interval starting in S (P.8). By axiom

P.1, there is a plan whose extension is S_P; that is, a plan PD such that PD executes over an interval just if P succeeds over that interval. Then PD satisfies the conditions of the lemma. ∎

THEOREM 4.51
For an omniscient agent, plan P is epistemically feasible as task if and only if it is possibly feasible.

omniscient ⇒ [task_ep_feasible(P, S) ⇔ pp_feasible(P, S)].

PROOF. Using the fact that the knowledge accessibility relation is just identity and that executability is just necessary feasibility, the condition for the epistemic feasibility of P as task reduces to the statement that there is a necessarily feasible specialization of P. By Lemma 4.50, this is equivalent to the condition that P be possibly physically feasible. ∎

4.6 *Sufficiency of Moore's rules for sequences*

In this section we will show the following two versions of Moore's [5] sufficient condition for the epistemic feasibility of sequences (Rule 2.1 of Section 2.2):

Rule 2.1 for executability: If

(a) PA is executable in $S1$; and

(b) after any completion of PA, PB will be executable;

then the plan 'sequence(PA, PB)' is executable in $S1$. (Note that Moore's condition that (b) be known in $S1$ is not needed for executability.)

Rule 2.1 for epistemic feasibility as task: If

(a) PA is epistemically feasible as a task in $S1$; and

(b) it is known in $S1$ that, after PA completes, PB will be epistemically feasible as a task;

then the plan 'sequence(PA, PB)' is epistemically feasible in $S1$.

LEMMA 4.52
sequence(PA, PB) succeeds if first PA succeeds and then PB succeeds.

succeeds(sequence(PA, PB),$S1, S2$) ⇔
$\exists_{SM}$ succeeds($PA, S1, SM$) ∧ succeeds($PB, SM, S2$).

PROOF. From the definition of the execution of a sequence (PL2.3), the definition of success (P.2), and the fact that error is irrevocable (P.15). ∎

LEMMA 4.53
sequence(PA, PB) begins over an interval if either PA begins over the interval or a successful completion of PA is followed by a beginning of PB. Note that these two disjuncts may not be mutually exclusive if PA is indeterminate.

begins(sequence(PA, PB),$S1, S2$) ⇔
[begins($PA, S1, S2$) ∨ [$\exists_{SM}$ succeeds($PA, S1, SM$) ∧ begins($PB, SM, S2$)]].

PROOF. From the definitions (PL2.3, P.2, P.3) and the fact that error is irrevocable (P.15.) ∎

LEMMA 4.54
The next step of sequence(PA, PB) after [$S1, S2$], assuming $S2$ is not an error state, is either a next step of PA after [$S1, S2$] or a next step of PB after [$SM, S2$], where SM is the end of a successful completion of PA. Again, if PA is indeterminate, then these two are not necessarily mutually exclusive.

$\neg$error($S2$) $\Rightarrow$
[next_step(E,sequence(PA, PB),$S1, S2$) $\Leftrightarrow$
[next_step($E, PA, S1, S2$) $\vee$
$\exists_{SM}$ succeeds($PA, S1, SM$) $\wedge$ next_step($E, PB, SM, S2$)]].

PROOF. From the definitions (PL2.3, P.2, P.4) and the fact that error is irrevocable (P.15.) ∎

LEMMA 4.55
If intervals $[S1, S2]$ and $[S1A, S2A]$ cover the same sequence of events, then each situation SM in $[S1, S2]$ has a parallel situation SMA in $[S1A, S2A]$.
[same_events($S1, S2, S1A, S2A$) $\wedge$ $S1 \leq SM \leq S2$] $\Rightarrow$
$\exists^1_{SMA}$ same_events($S1, SM, S1A, SMA$) $\wedge$ same_events($SM, S2, SMA, S2A$).

PROOF. Using the definition of same_events (CO.1, CO.2) and making an argument by induction on the length of the interval $[S1, S2]$. ∎

LEMMA 4.56
If interval $[S1A, S2A]$ corresponds to interval $[S1, S2]$ then $S1A$ is knowledge accessible from $S1$.
corresponds($S1, S2, S1A, S2A$) $\Rightarrow$ k_acc($S1, S1A$).

PROOF. By induction on the length of the interval $[S1, S2]$, using the definition of corresponds (CO.1 - CO.3) and the axiom of memory (K.4). ∎

LEMMA 4.57
If interval $[S1A, S2A]$ corresponds to interval $[S1, S2]$ then each situation SM in $[S1, S2]$ has a corresponding situation SMA in $[S1A, S2A]$.
[corresponds($S1, S2, S1A, S2A$) $\wedge$ $S1 \leq SM \leq S2$] $\Rightarrow$
$\exists_{SMA}$ corresponds($S1, SM, S1A, SMA$) $\wedge$ corresponds($SM, S2, SMA, S2A$).

PROOF. From Lemmas 4.55 and 4.56 using the definition of 'corresponding' intervals as knowledge accessible intervals covering the same events (CO.3). ∎

LEMMA 4.58
If interval $[S1A, S2A]$ corresponds to interval $[S1, S2]$, then each situation SMA in $[S1A, S2A]$ has a corresponding situation SM in $[S1, S2]$.
[corresponds($S1, S2, S1A, S2A$) $\wedge$ $S1A \leq SMA \leq S2A$] $\Rightarrow$
$\exists_{SM}$ corresponds($S1, SM, S1A, SMA$) $\wedge$ corresponds($SM, S2, SMA, S2A$).

PROOF. Using Lemma 4.55, let SM be the situation parallel to SMA. By Lemma 4.56, there is some situation SMA' in $[S1A, S2A]$ which corresponds to SM; by the uniqueness condition in Lemma 4.55, this can only be SMA. ∎

To simplify the statements of the remaining lemmas, let pa, pb, and p be plans and s1 be a situation such that:

p = sequence(pa,pb).
$\neg$error(s1).
executable(pa,s1).
$\forall_{S2}$ succeed(pa,s1,$S2$) $\Rightarrow$ executable(pb,$S2$).

LEMMA 4.59
Failing is never a next step of p.
$\neg$error($S2$) $\Rightarrow$ $\neg$next_step(fail,p,s1,$S2$).

PROOF. Let E be a next step of p after [s1,$S2$]. By Lemma 4.54, E is always either a next step of pa after [s1,$S2$] or a next step of pb after [$SM, S2$] where pa succeeds over [s1,SM]. In the former case, E cannot be fail since pa is executable in s1 (EF.1). In the latter case, E cannot be fail, since pb is executable following every successful completion of pa. ■

LEMMA 4.60

If p succeeds over [s1,$S2$], then the agent knows in $S2$ that it has succeeded.

[succeeds(p,s1,$S2$) ∧ corresponds(s1,$S2$,$S1A, S2A$)] ⇒ succeeds(p,$S1A, S2A$).

PROOF. If p has succeeded over [s1,$S2$] then, by Lemma 4.52, pa succeeds over [s1,SM] and pb succeeds over [$SM, S2$]. By the executability of pa, the agent knows in SM that pa has succeeded; that is, pa succeeds in every interval corresponding to [s1,SM]. By the executability of pb after the execution of pa, the agent knows in $S2$ that pb has succeeded; that is, pb succeeds in every interval corresponding to [$SM, S2$]. By Lemma 4.57, every interval corresponding to [s1,$S2$] consists of an interval corresponding to [s1,SM] followed by one corresponding to [$SM, S2$]. Therefore, every interval corresponding to [s1,$S2$] consists of a successful execution of pa followed by a successful execution of pb. ■

LEMMA 4.61

If p does not succeed over [s1,$S2$], then the agent knows in $S2$ that it has not succeeded.

[¬succeeds(p,s1,$S2$) ∧ corresponds(s1,$S2$,$S1A, S2A$)] ⇒ ¬succeeds(p,$S1A, S2A$).

PROOF. Let SM be any situation between s1 and $S2$. Let SMA be the situation between $S1A$ and $S2A$ corresponding to SM, as in Lemma 4.57. Since p does not succeed over [s1,$S2$], it must be the case that either (1) pa does not succeed over [s1,SM] or (2) pa does succeed over [s1,SM] but pb does not succeed over [$SM, S2$]. In the former case, since pa is executable in s1, the agent will know in SM that pa has not succeeded; thus, pa does not succeed over [s1a,SMA]. In the latter case, since pb is executable after any successful execution of pa, the agent will know in $S2$ that pb has not succeeded over [$SM, S2$]; thus, pb will not succeed over [$SMA, S2A$]. In either case, then p does not succeed over [$S1A, S2A$]. ■

LEMMA 4.62

If E is a next step of p after [s1,$S2$], then the agent knows in $S2$ that E is a next step.

[¬error($S2$) ∧ next_step(E,p,s1,$S2$) ∧ corresponds(s1,$S2, S1A, S2A$)] ⇒
next_step(E,p,$S1A, S2A$).

PROOF. By Lemma 4.54, there are two cases to be considered:

Case I: pa begins over [s1,$S2$] and E is a next step of pa after [s1,$S2$]. By the executability of pa in s1, the agent knows in $S2$ that E is a next step of pa after [s1,$S2$]; that is, E is a next step of pa after [$S1A, S2A$]. Therefore, by Lemma 4.54, E is a next step of p after [$S1A, S2A$]; that is, the agent knows in $S2$ that E is a next step of p.

Case II: pa succeeds over [s1,SM]; pb begins over [$SM, S2$]; and E is a next step of pb after [$SM, S2$]. Let SMA be the situation between $S1A$ and $S2A$ corresponding to SM (Lemma 4.57). By the executability of pa, the agent knows in SM that pa has succeeded; that is, pa succeeds over [$S1A, SMA$]. Since pb is executable after a successful execution of pa, the agent knows in $S2$ that E is a next step of pb after [$SM, S2$]; that is, E is a next step of pb after [$SMA, S2A$]. Thus, by Lemma 4.54, E is a next step of p after [$S1A, S2A$]. ■

LEMMA 4.63

The possibility that E is a next step of p after [s1,$S2$] is consistent with what the agent knows in $S2$ only if E is indeed a next step of p after [s1,$S2$].

[begins(p,s1,$S2$) $\wedge$ corresponds(s1,$S2$, $S1A$, $S2A$)] $\Rightarrow$
[next_step(E,p,$S1A$, $S2A$) $\Rightarrow$ next_step(E,p,s1,$S2$)].

PROOF. By Lemma 4.54, there are two cases to consider.

Case I: pa begins over [$S1A$, $S2A$] and E is a next step of pa after [$S1A$, $S2A$]. Since pa is executable in s1, and [$S1A$, $S2A$] corresponds to [s1,$S2$], it follows that E must be a next step of pa after [s1,$S2$].

Case II: pa succeeds over [$S1A$, SMA], pb begins over [SMA, $S2A$], and E is a next step of pb after [SMA, $S2A$]. Let SM be the situation in [s1,$S2$] corresponding to SMA, as in Lemma 4.58. Since pa is executable in s1, pa must succeed over [s1,SM]. Since pb is executable in SM, E must be a next step of pb after [SM, $S2$]

In either case, E is a next step of p after [s1,$S2$]. ■

LEMMA 4.64
Plan p is executable in s1.
executable(p,s1).

PROOF. Combining Lemmas 4.59 through 4.63 with the definition of executability (EF.1). ■

THEOREM 4.65
If PA is executable in $S1$ and PB is executable after every successful completion of PA, then sequence
(PA, PB) is executable in $S1$.
[executable(PA, $S1$) $\wedge$ [$\forall_{S2}$ succeeds(PA, $S1$, $S2$) $\Rightarrow$ executable(PB, $S2$)]] $\Rightarrow$
executable(sequence(PA, PB),$S1$).

PROOF. Universal abstraction from Lemma 4.64, using the assumptions enumerated before Lemma 4.59. ■

LEMMA 4.66
If PAS is a specialization of PA and PBS is a specialization of PB, then sequence(PAS, PBS) is a specialization of sequence(PA, PB).
[specialization(PAS, PA) $\wedge$ specialization(PBS, PB)] $\Rightarrow$
specialization(sequence(PAS, PBS), sequence(PA, PB)).

PROOF. From the definitions of sequence (PL2.3) and specialization (P.9). ■

THEOREM 4.67
If PE is epistemically feasible as task in $S1$ and the agent knows in $S1$ that PF will be epistemically feasible as task after every successful completion of PE, then sequence(PE, PF) is epistemically feasible as task in $S1$.
[task_ep_feasible(PE, $S1$) $\wedge$
[$\forall_{S1A,S2A}$ [k_acc($S1$, $S1A$) $\wedge$ succeeds(PE, $S1A$, $S2A$)] $\Rightarrow$
task_ep_feasible(PF, $S2$)]] $\Rightarrow$
task_ep_feasible((sequence(PE, PF),$S1$).

PROOF. From definition of epistemic feasibility as a task (EF.4) we can infer:

- There exists a plan PES such that in every world $S1A$ accessible from $S1$, PES is a specialization of PE and PES is executable.

- For every pair of worlds $S1A, S2A$ such that $S1A$ is accessible from $S1$ and PE succeeds over $[S1A, S2A]$, there exists a plan pfs(PF, $S2A$) such that in every world $S2B$ accessible from $S2A$, pfs(PF, $S2A$) is a specialization of PF in $S2B$ and is executable in $S2B$.

Let $\mathcal{I}$ be the set of all intervals $[S2A, S3A]$ satisfying the following conditions: there exists an $S1A$ that is knowledge accessible from $S1$; PES executes over $[S1A, S2A]$; and pfs(PF, $S2A$) executes over $[S2A, S3A]$. Let PFS be the plan corresponding to $\mathcal{I}$ (axiom P.1). Let PS = sequence(PES, PFS).

Now, by construction, PES is executable in every situation $S1A$ accessible from $S1$. By construction, for every such $S1A$, and for every $S2A$ such that PES succeeds over $[S1A, S2A]$, PFS is executable. Therefore, by theorem 4.67, PS is executable in every such $S1A$. By Lemma 4.66, PS is a specialization of sequence(PE, PF). Therefore, by definition EF.4, PS is epistemically feasible as a task. ∎

4.7 *Sufficiency of Moore's rules for conditionals*

In this section we will show the following two versions of Moore's sufficient condition for the epistemic feasibility of conditionals: (Rule 2.2 of Section 2.2).

Rule 2.2 for executability: If either
(a) it is known that Q holds in S and PA is executable in S; or
(b) it is known that Q does not hold in S and PB is executable in S
then the plan 'if(Q, PA, PB)' is executable in $S1$.

Rule 2.2 for epistemic feasibility as task: If either
(a) it is known that Q holds in S and PA is epistemically feasible as a task in S; or
(b) it is known that Q does not hold in S and PB is epistemically feasible as task in S
then the plan 'if(Q, PA, PB)' is epistemically feasible as task in $S1$.

Actually, we will just prove part (a) of each; the proof of (b) is exactly analogous.

LEMMA 4.68
If it is known that PE executes from the current state if and only if PF does, then PE is epistemically feasible as task if and only if PF is.

$$[\forall_{S1A,S2A}\ \text{k\_acc}(S1, S1A) \Rightarrow$$
$$[\text{executes}(PE, S1A, S2A) \Leftrightarrow \text{executes}(PF, S1A, S2A)]] \Rightarrow$$
$$[\text{task\_ep\_feasible}(PE, S1) \Leftrightarrow \text{task\_ep\_feasible}(PF, S1)].$$

PROOF. Follows directly from Lemma 4.45, together with the observation that any specialization of PF from a knowledge accessible situation $S1A$ is also a specialization of PE. Note that this corresponds to Rule 2.4 of Section 2.2 for epistemic feasibility as task. ∎

THEOREM 4.69
If Q is known in $S1$, then the conditional 'if(Q, PE, PG)' is executable just if PE is executable and is epistemically feasible as task just if PE is epistemically feasible as task.

$$[PF\text{=if}(Q, PE, PG) \wedge \forall_{S1A}\ \text{k\_acc}(S1, S1A) \Rightarrow \text{holds}(Q, S1A)] \Rightarrow$$
$$[\text{executable}(PF, S1) \Leftrightarrow \text{executable}(PE, S1)] \wedge$$
$$[\text{task\_ep\_feasible}(PF, S1) \Leftrightarrow \text{task\_ep\_feasible}(PE, S1)].$$

PROOF. By definition of the conditional (PL2.4), for any situation $S1A$ accessible from $S1$, PF executes over $[S1A, S2A]$ if and only if PE executes over $[S1A, S2A]$. The result then follows from Lemmas 4.45 and 4.68. ∎

5 Limitations, extensions, and future work

The theory above assumes a particular representation language and a restrictive ontology. It will be necessary to loosen these if the theory is to be applied to rich real-world domains. Certain extensions can be made easily; others appear to be very difficult:

Continuous time. In a forthcoming paper [3] I will describe the generalization of this theory to continuous action. We posit that for each agent there is a minimum time Δ between his perceiving a fact and his reacting to his perception. Under that assumption, we propose the following definitions:

DEFINITION 5.1
Plan P is executable for agent A with delay Δ in situation $S1$ if: (1) All executions of P starting in $S1$ complete successfully; and (2) After any beginning of P starting in $S1$, A will know whether P will complete within time Δ and A will know all the ways to continue P for time Δ.

DEFINITION 5.2
Plan P is epistemically feasible as task for agent A in situation $S1$ with delay Δ if there is a plan PC such that A knows in $S1$ that PC is a specialization of P and that PC is executable in $S1$ with delay Δ.

Epistemic theory. It is easy to adapt most of the theory to a more explicitly intensional representation for epistemic states, such as a modal or a syntactic theory. The largest difference is in the theory of plans. The view we have taken that any set of intervals constitutes a plan becomes unnatural in an intensional theory. However, without that assumption, some of the proofs in Section 3 fall through. It would be interesting to study how this gap could be filled in.

Concurrency. Synchronous concurrency can be handled by modifying the time line so that an arc between situations corresponds to a set of primitive actions being executed simultaneously [6, 10]. Asynchronous concurrency can be reduced to synchronous concurrency by representing an action A as the iteration of the primitive action 'continue(A)' for an indeterminate number of time quanta.

Actions with indeterminate effects. Our theory assumes that in each situation an action has a determinate result. In many cases, such as rolling a die, this assumption may not be appropriate. There are two ways of extending our theory to accommodate such actions. The simpler approach is to treat this simply as a case of ignorance. Before rolling a die, there are six accessible possible worlds: in the first, the die will come up one; in the second, it will come up two; and so on.

An alternative would be to modify the time line to admit two kinds of branching: one corresponding to the agent's choice of action and the other to indeterminacy of effect. This approach would require the theory to be reformulated to distinguish clearly between the sequence of actions performed, which is under the agent's control, and the interval traversed, which is not. (The current theory essentially uses an interval as a convenient way to denote a sequence of events.)

External events. The incorporation of external events into our theory does not raise any further difficulties besides those already addressed above. The fact that external events occur concurrently with the agent's actions can be handled in the same way as concurrent actions. The fact that they may not be entirely predictable can be handled in the same way as actions with indeterminate effects. The fact that they occur as a result of physical law rather than agent choice can be handled by asserting a theory of physical constraints among events and states of the usual kind. Some specific examples are analysed in [3].

Other agents. If there is more than one autonomous intelligent agent, then each agent has his own epistemic state and his own choice of actions. Giving agents separate epistemic states requires only making the agent an additional argument to the knowledge accessibility relation (or whatever other epistemic representation is used). Giving agents separate powers of choice requires restructuring the time line along the lines discussed above: we must admit more than one type of branching (one for each agent), and we must consider an agent as controlling only the sequence of his own actions but not the overall interval that will be traversed.

A further difficulty is that, in planning, we often wish to predict that, though an agent has a choice of actions, he will in fact do one particular thing. For example, in an environment of cooperative agents, we would like to state the default rule, 'If one agent makes a request of another, then the latter will carry it out, if he can'. To express this, we need to distinguish between what an agent *can* do and what he *will* do; that is, we need to be able to say that one particular branch of the time line corresponds to what actually will happen. The full treatment of this issue has not yet been worked out [12, 18].

Planning language. For practical applications, perhaps the most important problem is to develop a simple tractable language of knowledge effects and knowledge preconditions, analogous to the STRIPS [5] language of add lists, delete lists, and precondition lists. An interesting preliminary attempt at this is made in [4].

Hierarchy of actions. A fundamental premiss of our work is the assumption that there exists a particular level of 'primitive' actions. It is often useful to reason about actions at multiple levels of granularity [9], refining a coarse description using high-level actions into a more precise description using low-level actions. For such reasoning to be coherent, there should be some connection between the conclusions that are drawn at the various levels; if a plan P is found to be feasible when described at a high level, then (at a minimum) there should be a low-level refinement of P that is likewise feasible. It is not at all clear that this kind of consistency can be achieved for the theory we have presented here.

Better fundamental theory. Finally, I am not convinced that our theory has reached rock bottom in the analysis of epistemic feasibility. The definitions are complicated and must be mentally argued through; they do not evoke immediate assent. It is clear that different notions of feasibility are appropriate to different circumstances; our theory does not provide any indication when to use one definition rather than another. It seems likely that there is some deeper sense of an agent being able to carry out or to think through a plan, of which the theory developed here is a consequence or an approximation.

References

[1] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, **32**, 333–378, 1987.

[2] E. Davis. *Representations of Commonsense Knowledge*. Morgan Kaufmann, San Mateo, CA, 1990.

[3] E. Davis. Branching continuous time and the semantics of continuous action. In *Proceedings of the Second International Conference on AI Planning Systems*, 1994.

[4] O. Etzioni, S. Hanks, D. Weld, D. Draper, N. Lesh and M. Williamson. An approach to planning with incomplete information. In *Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992.

[5] R. E. Fikes and N. J. Nilsson. Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, **2**, 189–208, 1971.

[6] M. Gelfond, V. Lifschitz and A. Rabinov. What are the limitations of the situation calculus? In *Automated Reasoning: Essays in Honor of Woody Bledsoe*, R. Boyer, ed., pp. 167–179. Kluwer Academic, Dordrecht, 1991.

[7] J. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 480–490, 1985.

[8] J. Hintikka. Semantics for propositional attitudes. In *Reference and Modality*, L. Linsky, ed., pp. 145–167. Oxford University Press, 1969.

[9] J. Hobbs. Granularity. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 432–435, 1985.

[10] F. Lin and Y. Shoham. Concurrent actions in the situation calculus. In *Proceedings, Tenth National Conference on Artificial Intelligence*, pp. 590–595, 1992.

[11] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, B. Meltzer and D. Michie, eds, pp. 463–502. Edinburgh University Press, Edinburgh, 1969.

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

[13] R. Moore. Reasoning about knowledge and action. Note 191, SRI International, Menlo Park, CA, 1980.

[14] R. Moore. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*, J. Hobbs and R. Moore, eds, pp. 319–358. ABLEX Publishing, Norwood, NJ, 1985.

[15] L. Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp. 867–874, 1987.

[16] L. Morgenstern. *Foundations of a Logic of Knowledge, Action, and Communication*. PhD thesis, New York University, 1988.

[17] L. Morgenstern. Knowledge and the frame problem. In *Reasoning Agents in a Dynamic World: The Frame Problem*, K. Ford and P. Hayes, eds. JAI Press, 1991.

[18] J. Pinto and R. Reiter. Adding a time line to the situation calculus. In *Second Symposium on Logical Formalizations of Commonsense Reasoning*, 1993.

[19] E. Sacerdoti. *A Structure for Plans and Behavior*. Elsevier, New York, 1975.

[20] R. B. Scherl and H. J. Levesque. The frame problem and knowledge producing actions. In *Proceedings, Eleventh National Conference on Artificial Intelligence*, pp. 689–695, 1993.

[21] D. Weld. An introduction to least commitment planning. *AI Magazine*, 1994.

Received 2 June 1993