A Proper Ontology for Reasoning About Knowledge and Planning

Leora Morgenstern
IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, N.Y. 10598
leora@watson.ibm.com

Abstract:

Research on the knowledge preconditions problems for actions and plans has sought to answer the following questions:
(1) When does an agent know enough to perform an action?
(2) When can an agent execute a multi-agent plan?

It has been assumed that the choice of temporal ontology is not crucial. This paper shows that this assumption is wrong and that it is very difficult to develop within existing ontologies theories that can answer both questions (1) and (2). A theory of linear time does not support a solution to the knowledge preconditions problem for action sequences. A theory of branching time solves this problem, but does not support a solution to the knowledge preconditions problem for multi-agent plan sequences. Linear time supports prediction, but does not support hypothetical reasoning; branching time supports hypothetical reasoning, but does not support prediction. Since both prediction and hypothetical reasoning are essential components of the solution to the knowledge preconditions problems, no comprehensive solution has yet been proposed.

To solve this problem, we introduce a new temporal ontology, based on the concept of an occurrence that is real *relative* to a particular action. We show that this ontology supports both hypothetical reasoning and prediction. Using this ontology, we define the predicates needed for the proper axiomatization for both knowledge preconditions problems.

1 Introduction

Intelligent agents not only possess knowledge, but they reason about the knowledge that they possess. This sort of introspection is particularly crucial for planning. Agents are not capable of performing every action, so an agent who constructs a plan must reason about his ability to perform the actions in his plan. Since the ability to perform many actions rests directly upon an agent's knowledge, he must reason about whether he has that knowledge, or how he can get that knowledge. For example, an agent who plans to perform the sequence of actions: *(open up safe, remove money)* must know that he knows the combination of the safe in order to predict that his plan will be successful.

There has been a fair amount of research in the field of knowledge and planning in the last 15 years. Most of this work ([Moore, 1980], [Konolige, 1982]) has focussed on the *knowledge preconditions problem for actions:* what does an agent need to know in order to perform an action? This question is only part of the story, however: if an agent *does not* know enough to perform an action, he will presumably not just drop his goal. Instead, he will either plan to get the information, possibly by asking another agent, or by delegating the task to another more knowledgeable agent. In either case, he will have to construct a more complex multi-agent plan. This gives rise to the *knowledge preconditions problem for plans*: what does an agent have to know in order to successfully execute a plan? For example, if I don't know the combination of the safe, I may ask Bob to tell me the combination. To predict that this plan will work, I must know that Bob knows the combination, that he will tell it to me, and so on. Presumably, the knowledge preconditions for this sort of plan are weaker than for my plan to open the safe – but they are difficult to make explicit.

In [Morgenstern, 1988], we studied the knowledge preconditions problem for plans in detail, and furnished axioms giving sufficient knowledge preconditions for various sorts of plans, including sequences, conditionals, and loops. However, as noted there, these axioms are overly strong; they entail that an agent has sufficient knowledge to execute a plan even when intuition tells us otherwise. For example, what seems to be a straightforward or "natural" way of axiomatizing the knowledge preconditions for plan sequences entails that an agent could always do a sequence of actions as long as he *could* perform the first action, *but* (and this is the crucial point) *he never actually did.* This was true even if the second action was impossible to perform. The theory is still valid for forward reasoning planning; however, it is clearly undesirable to have a theory that legitimates degenerate plans. This paper addresses and solves this problem.

We had previously suggested that the problem was most probably due to the use of linear time, and claimed that using a more sophisticated temporal ontology such as branching time would solve the problem. As we will show in this paper, branching time is also not sufficient. We need to construct a new and richer underlying temporal ontology.

This paper is structured as follows: We briefly describe the logical language used, and give a natural language characterization of the solution to the knowledge preconditions problems. Next we show that formalizing these axioms in a linear theory of time will not work. The following section shows that the seemingly obvious solution – recasting these axioms using a branching theory of time – does not work either. Finally, we introduce a new temporal ontology, called *relativized branching time*, which takes elements of both branching and linear times and is based on the notion of the "most real" world, relative to a particular action. We show that this temporal ontology can be used to construct a correct theory of knowledge preconditions for actions and plans.

2 The Logical Language

We will be working in a logical language $\mathcal{L}$, an instance of the first order predicate calculus.
(What follows is terse and incomplete, due to space considerations. $\mathcal{L}$ is modeled on the logic used in [Morgenstern, 1988])
$\mathcal{L}$ is distinguished by the following features:
[1] $\mathcal{L}$ contains a 3-place predicate *Know*. *Know(a,p,s)* means that agent a knows the sentence represented by the term p in the situation ("time-point") s.

When we say that the term p represents a sentence, we are indicating that a quotation construct is present in $\mathcal{L}$. Thus:
[2] $\mathcal{L}$ allows quotation. We can use a term or a wff in $\mathcal{L}$ and talk *about* that term or wff in $\mathcal{L}$. We do this by associating with each term or wff of $\mathcal{L}$ the quoted form of that term or wff. In general, we will denote the term representing a wff or term as that wff or term surrounded by quotation marks.

Some notes on quotation: unrestricted use of quotation can lead to paradox [Montague, 1963]; some sort of resolution is necessary. Here we choose:
[3] $\mathcal{L}$ is interpreted by a three-valued logic, which is transparent to the user and ignored in the remainder of this paper.
[4] Quantification into quoted contexts is a somewhat messy enterprise, involving some sort of quasi-quotes. We use the notation of [Davis, 1990] : The delimiters $\wedge\wedge$ and $\downarrow\downarrow$ are used when the variables that are quantified into quoted contexts range over strings; @ is used for variables that range over objects other than strings. The partial function h maps a string onto the term it represents; it is abbreviated as the . (the period). Those unfamiliar with quasi-quotation should just ignore these symbols.

As we have indicated, and will be arguing at greater length, the choice of a temporal ontology will be crucial for our endeavor. Nevertheless, there are some elements that will be present in any choice. They are:
[5] The basic building block is the situation, or time point. (How these points are organized is the crux of the differences between approaches). Intervals of time are indicated by a pair of time-points, the starting time and the ending time. An action or event is a collection of intervals – intuitively those intervals in which the action takes place. An event is an action restricted to a particular agent (*the performing agent*). The function *Do* maps an agent and an action onto an event. Actions and events can be structured using standard programming language constructs. A plan is any structure of events, e.g., *Sequence(Do(Susan(ask(Bob,combination)), Do(Bob(tell(Susan, combination))))* A restricted subset of actions are *primitive:* - they cannot be further decomposed. Other actions are *complex* and are built up out of primitive actions using our programming language structures.

In all formulas of the theory and metatheory, all variables are assumed to be universally quantified unless otherwise indicated.

3 What We Want to Say

In English, the solution to the knowledge preconditions problem for actions can be stated as follows: it is assumed that all agents know how to perform the basic action types of primitive actions. In order to know enough to perform a primitive action, then, one must only know what the parameter of the action is. That is, one must know of a constant equivalent to the parameter. Thus, for example, suppose that *dial* is a primitive action. Then one knows how to dial the combination of a safe if one knows of a sequence of digits equivalent to the combination of the safe. The knowledge preconditions for complex actions are given recursively in terms of the knowledge preconditions for primitive actions.

If an action is complex, an agent must explicitly know its decomposition into primitive actions, and know how to perform the decomposition. Moreover, if one cannot perform an action, one generally constructs some multiple agent plan whose end result is the achievement of the original goal. The solution to the knowledge preconditions problem for plans can therefore be stated as follows: An agent knows how to execute a plan if he knows how to perform all of the actions of the plan for which he is the performing agent, and can predict that the other agents in the plan will perform their actions when their time allows. For example, Susan can execute the plan sequence *sequence(do(Susan, ask(Bob, combination)), do(Bob, tell(Susan, combination)))* if Susan can ask Bob for the combination, and she knows that as a result of her asking him for the combination, he will tell it to her. Note that in order for Susan to predict that Bob will tell her the combination, she must know that Bob in fact knows it, and that he is willing to share the information.

The above natural language description is a succinct summary of the observations of [Moore, 1980] (for primitive actions) and [Morgenstern, 1988] (for complex actions and multi-agent plans). The difficulty now is in formalizing this – correctly – within a formal logic. It is necessary to formalize *prediction* – knowing that an event will happen in the future – and the notion of *vicarious control* – controlling a plan even if you are not involved in it. The problem addressed in this paper arises in the characterization of the knowledge preconditions for complex plans in terms of primitive plans. We focus here on sequences of plans.

We would like to say that an agent knows how to perform a sequence of actions if he knows how to perform the first action, and as a result of performing the first action, he will be able to perform the second action. Similarly, an agent knows how to execute a sequence of plans if he can execute the first, and as a result of the first plan's occurrence, he can execute the second. We turn to the formalization of these principles in the next section.

4 Difficulties With Linear Time

One of the simplest ways to view time is as a straight line - i.e., the standard time line of school history books. There is a total ordering on time points or situations. We call this representation of time "linear time." An interval of time is a segment of the time line; as mentioned in Section 2, intervals are denoted by their start and end points. An action is a collection of intervals; *Occurs(act1,s1,s2)* is true iff *(s1,s2)* is an element of *act1*.

The knowledge preconditions for primitive actions are omitted here. They can be found in [Morgenstern, 1988]. The axiom for one simple case can be found in this paper's appendix.

We focus here on complex actions. Recall that we would like to say that an agent knows how to perform a sequence of *act1, act2* if he knows how to perform *act1* and knows that as a result of performing *act1*, he will know how to perform *act2*. A reasonable try at the knowledge preconditions axiom for action sequences might thus be:

Axiom 1: *(Knows-how-to-perform(a,act1,s1)* & *(Occurs(do(a,.act1),s1,s2)* $\Rightarrow$ *Knows-how-to-perform(a,act2,s)))* $\Rightarrow$ *Knows-how-to-perform(a,'sequence(*$\wedge$*act*$\wedge$*,* $\wedge$*act2*$\wedge$*)',s1)*

Despite this axiom's plausibility, it does not say what we want. It allows agents to know how to perform some very odd action sequences. In particular, it entails that an agent knows how to perform a sequence of two actions if (s)he knows how to perform the first act but does not perform this act – even if (s)he doesn't know how to perform the second act! For example, consider the agent Nancy Kerrigan, the action sequence *(ice_skate, build_atom_bomb)* , and the situation S1 representing January 7, 1994. It is clear that on January 7, Nancy Kerrigan knew how to ice skate. We know, however, that due to injuries, she did not skate on that day. Then the statement *Knows-how-to-perform(Kerrigan,'sequence(ice_skate, build_atom_bomb)', S1* is true, since the second conjunct of the left-hand side of the axiom is vacuously true.

The problem, when we examine this anomaly more closely, seems to be that material implication is being used to capture the notion of "*as a result of* performing action 1." The truth is that material implication is quite different from, and much stronger than, the notion of result. This is the reason it is so much more difficult to modify Axiom 1 than one might suppose. It is not merely that we have somehow missed something in the formalization. The problems inherent in material implication have appeared in many suggested modifications of this axiom as well, since material implication plays a central role in these axioms as well.

This problem strikes a familiar chord. In fact, there are many types of reasoning, such as counterfactual reasoning, and concepts in temporal reasoning, such as prevention and causality, that would seem to be straightforward to implement, but which fail due to the very strong nature of material implication. One approach to solving such problems has been to examine these concepts within the framework of a richer ontology. Often, the ontology chosen has been *branching time* [McDermott, 1982]. We examine the knowledge precondition problems in the context of branching time in the next section.

Figure 1: *McDermott's branching time. Real chronicle in bold*

5 Difficulties With Branching Time

In branching time, time points are ordered by a partial order as opposed to a total order. There is a unique least point, and one cannot have $s1 < s2$ and $s3 < s2$ unless either $s1 < s3$ or $s3 < s1$ (that is, every child has at most one parent). Thus, while one could visualize linear time as a straight line, the best way to visualize branching time is as a sideways tree (See Figure 1). Conceptually, the branch points correspond to action choice points; each branch represents a different action performed. Following [Mc-

Dermott 1982], any linearly ordered set of points (or path), beginning with the least point, and without gaps, is called a chronicle. There ia s one chronicle that is designated as the "real chronicle"; this corresponds to the way the world is. A time point is called *real* if it lies on the real chronicle. An interval is called real if it contains only real time points. We introduce the predicate *Real-occurs*:

Definition: *Real-occurs(act,s1,s2)* $\Leftrightarrow$ *Occurs(act1,s1,s2)* & *Real((s1, s2))*.

Since we used linear time in the last section, the *Occurs* predicate used there corresponds to the *Real-occurs* predicate of this section. Axiom 1 is now correct. The left-hand conjunct is not vacuously true in the Nancy Kerrigan example, above; the axiom now says: if Nancy Kerrigan knew how to ice skate on January 7, and in any possible world resulting from her skating on January 7, she knew how to build an atom bomb, then she knows how to perform the sequence of actions. In fact, it is safe that assume that in no possible world resulting from Nancy Kerrigan's skating did she know how to build an atom bomb; thus she does not know how to perform the sequence of actions. This is just what we would anticipate. Indeed, the fact that the axiom now works is to be expected; [Moore 1980] used branching time (his temporal ontology was a variation of the situation calculus) and was able to correctly formalize knowledge preconditions for action sequences.

The problem now is that branching time *cannot* be used for formalizing knowledge preconditions for *plans*. The reason, briefly, is that in order for an agent to reason that he can execute a multi-agent plan, the agent must be able to predict that other agents will perform certain actions. Predicting means knowing that an event will *actually* occur - i.e., that the occurrence will be part of the *real* chronicle. But suppose, now, that an agent, Susan, is reasoning about her ability to execute *sequence(pln1, pln2)*. E.g., assume that Susan is reasoning about her ability to execute the plan *sequence(Do(Susan,ask(Bob,combination)),Do(Bob,tell (Susan,combination)))*. We assume that *pln1* is a single action where Susan is the performing agent; *pln2* is a single action where Bob is the performing agent. Then Susan must know that she can perform *pln1* [1] and that as a result of performing *pln1*, Bob will perform *pln2*. That is, she must know that in any possible world resulting from the event *Do(Susan,ask(Bob,combination))*, Bob will perform *Do(Bob,tell(Susan,combination))*. But this is impossible by nature of the definitions: Bob can only *really* perform *pln2* in the one real chronicle, not in every branch in which Susan performs *pln1*. Moreover if Susan doesn't perform *pln1*, then Bob's performance of *pln2* will only occur in non-real chronicles! This situation is shown in Figure 2.

Figure 2: *Branching time doesn't support hypothetical reasoning: Bob doesn't "really" tell Susan the number when Susan asks for it (non-bold segments)*

Thus, we are now in a situation that is precisely the opposite of the situation that occurred in linear time. The theory based on linear time is too liberal; it entails that agents know how to perform sequences of two actions even if they do not know how to perform the second action. The theory based on branching time, on the other hand, is too restrictive. It is virtually impossible to prove, under reasonable assumptions, that an agent can execute a standard sequence of plans, such as asking a friend for a piece of information, and receiving that information.

More formally, consider the following axioms:

Axiom 2:
Can-execute-plan(a,'sequence($\wedge$*pln1*$\wedge$*,*$\wedge$*pln2*$\wedge$*)',s1)*
$\Leftrightarrow$
Know(a,'Vicarious-control(@a, $\downarrow$*pln1*$\downarrow$*,@s1)',s1)* &
Know(a,'Occurs($\wedge$*pln1*$\wedge$*,@s1,s2)*$\Rightarrow$ *Vicarious-control(@a,*$\downarrow$ *pln2* $\downarrow$*,s2)',s1)*

Axiom 3:

actors(pln) $=$ { *a* } & *Can-perform(a,'action(@a,* $\wedge$*pln* $\wedge$ *)',s)* $\Rightarrow$
Vicarious-control(a,pln,s)

Axiom 4:

actors(pln) $\neq$ { *a* } & $\exists$ *s2 Real-occurs(.pln,s,s2)* $\Rightarrow$ *Vicarious-control(a,pln,s)* .

Vicarious-control, in the axioms above, can be thought of meaning "one of the following: I can do it or it will happen." That is, one vicariously controls a plan if *one can count on it happening.* I can count on my fixing myself a scrambled egg in the morning because I know how to perform the action; thus, by Axiom 3, I vicariously control it. I can count on the sun rising this morning because I can predict that it will happen; thus, by Axiom 4, I vicariously control it.

Axiom 2 states that I can count on a sequence of

[1] In order to reason about plan execution, one must reason not only about knowledge preconditions, but also physical and social feasibility. When all three are satisfied, an agent *can-perform* an action. See Appendix.

plans if I can count on the first plan, and as a result of the first plan's occurrence, I can count on the second.

Now consider the plan sequence *sequence(do(Susan, ask(Bob, comb)), do(Bob, tell(Susan, comb)))*. It can easily be seen that under most normal sets of assumptions, *Can-execute-plan(the above plan)* cannot be proven using Axioms 1 through 4. This is just one anomalous case. Similar problems occur with conditional plans, and in cases where agents are not directly involved in any aspect of their plan – i.e., when the entire plan consists of actions that have been delegated. The problem arises whenever one must predict that an action will take place if a piece of a plan has occurred.

6 A Solution That Works: Branching Time With Relativized Real States

Thus far, we have demonstrated that linear time is difficult to use to formalize knowledge preconditions because it does not allow for generalized hypothetical reasoning; that branching time is likewise difficult because it emphasizes hypotheticals too strongly and does not allow for generalized prediction. What we want is a theory that supports both hypothetical reasoning and prediction. [2] That is, we would like to develop a theory in which we can say: given that *act1* has occurred, *act2* will surely occur. This "sureness" or "realness" is relative to the action that has occurred. We call this *relativized branching time.*

To capture this concept, we modify the ontology of branching time as follows. We introduce a collection of partial orders $<_r$ (to be read as "more real than") on branch segments of our tree. There is a partial order $<_{r_i}$ at each branching point i; $<_r$ is the collection of $<_{r_i}$ for all i. Where no confusion will result, we will simply write $<_r$ for $<_{r_i}$. $<_r$ has the following properties:

For each branch point i with n branch segments $b_1 \cdots b_n$, $\exists! b_j \ni$

1. $b_j <_r b_1, \cdots b_j <_r b_{j-1}, b_j <_r b_{j+1}, \cdots b_j <_r b_n$ *(existence and uniqueness of least element under* $<_{r_i}$ *)*

2. $\forall k, l \neq j, \neg b_k <_r b_l, \neg b_l <_r b_k$ *(Other than the least element, branches are incomparable.)*

This b_j is the "most real branch " at point i. Intuitively, it is the branch most likely to occur at time i. b_j is the unique minimal element in the partial order induced by $<_{r_i}$. [3] Note also that condition (2) may be dropped if we wish to model a world in which there are different levels of preferred occurrences relative to some action. For example, condition (2) would most likely be dropped in a theory that allowed for defeasible reasoning. If one originally inferred that some action would happen because it was on the most preferred branch, and then had to retract that conclusion, it would be helpful to know which of the remaining branches was most likely to occur, and make new predictions based on this information.

We can use the notion of a *most real branch* to define the concept of a *most real path* at a point s. Specifically, define a path in a tree as a sequence of branches $b_1, \cdots b_j$ where for each b_i, $i \in (1, j-1)$, the endpoint of b_i is the starting point of b_{i+1}.

Definition: $(b_1, \cdots b_j)$ is the most real path iff for all $i \in$ (1,j) b_i is the most real branch segment relative to b_i's starting point.

Thus, for example, in Figure 3, the path $(s_0, s_2, s_6, s_{11}, s_{13}, s_{14})$ is the most real path at the point x_0 because all the branch segments are the most real at their starting points. On the other hand, the path $(s_0, s_2, s_6, s_7, s_{10})$ is not most real at s_0 because (s_6, s_7) is not the most real branch segment at s_6.

Let s_0 be the root of a branching tree structure. Note that the *most real path* at s_0 corresponds precisely to McDermott's *real chronicle.* Our move to a richer temporal ontology has thus lost us nothing in expressivity.

We now extend the $<_r$ relation to range over subtrees in the obvious way. We thus have the following: *Definition of* $<_r$ *for subtrees:* Assume $b_1 <_{r_s} b_2$, where b_1 has the endpoints (s, s_1) and b_2 has the endpoints (s, s_2). Let t_1 be the subtree rooted at s_1 and t_2 be the subtree rooted at s_2. Then $t_1 <_{r_s} t_2$.

[2] Other, less satisfactory approaches are possible. We could use linear time, but introduce an explicit predicate *Causes* and thus eliminate the problems of material implication. Our axiom on knowledge preconditions for action sequences would then read:
(Know-how-to-perform(a,seq(act1,act2),s)
& Causes(act1,Know-how-to-perform(a,act2)))
⇒ Know-how-to-perform(a,seq(act1,act2),s) . But there are several problems with this strategy: We need to give a semantics to *Causes.* If we cannot, the theory is somewhat bogus; if we reduce *Causes* to material implication, the problems return through the back door. Moreover, sometimes the fact that one knows how to perform an action *act2* after performing an action *act1* does not mean that performing *act1 caused* the agent to know how to perform *act2.* One can imagine a situation in which I know that I will be told the combination of the safe at some point late in the day. In the meantime, I spend my day chopping wood. Now, it is perfectly plausible that I will know how to open the safe after I chop wood – but I would not want to say that the wood chopping *caused* me to know how to open the safe.

Another approach would be to develop an ontology using only "axiomatically possible worlds." The disadvantages here would be that it would be non-intuitive and hard to modify.

[3] We have imposed the condition of uniqueness for ease and simplicity of presentation but this condition is not strictly necessary. It is likely that in complex domains with varying degrees of granularity of representation, there can be several most preferred branches. For example, if Susan asks Bob for the combination, the branch in which he answers her orally and the branch in which he answers her in writing could both be most preferred branches. We deal with this in the longer version of this paper.

Figure 3: *relativized branching time: at each branching point, there exists a unique preferred branch (in bold). Note that since (so,s2) is more real than (s0,s19), the tree rooted at s2 is more real than the tree rooted at s19.*

See Figure 3 for examples of these definitions. Using this ontology, we can now introduce the concept of a state that is real relevant to some point in time. Specifically, we introduce the predicate *Real-wrt(s1,s2)*, which is given by the following metatheoretic definition:

Definition: $\models$ *Real-wrt(s1,s2)* iff *s2* is a point on b_j where b_j is the most real branch point originating from *s1*.

We extend *Real-wrt* to range over intervals in the obvious way. Specifically:

Definition: *Real-wrt(s1,(si,sj))* $\Leftrightarrow \forall s \in$ *(si,sj) Real-wrt(s1,s)*

Those causal rules which have action occurrences in their consequent must now be written in terms of this predicate. In general, where before we would have:

Holds(fluent,s1)
$\Rightarrow \exists$ *s2 Occurs(act,s1,s2)*

we would now have:

Holds(fluent,s1)
$\Rightarrow \exists$ *s2 Real-wrt(s1,s2)* & *Occurs(act,s1,s2)*

and where before we would have:

Occurs(act1,s1,s2)
$\Rightarrow \exists$ *s3 Occurs(act2,s2,s3)*

we would now have:

Occurs(act1,s1,s2)
$\Rightarrow \exists$ *s3 Real-wrt(s2,s3)* & *Occurs(act2,s2,s3)*.

In the above transformation rules, the term *Holds(fluent,s1)* is really just syntactic sugar; in fact, in our notation, the situation is just another argument to the predicate. Here is an example of a transformation: Where before we had

Goal(a,act,s) & *Can-perform(a,act,s)*
$\Rightarrow \exists$ *s2 Occurs(do(a,act),s,s2)*

we would now have:

Goal(a,act,s) & *Can-perform(a,act,s)*
$\Rightarrow \exists$ *s2 Real-wrt(s,s2)* & *Occurs(do(a,act),s,s2)*

We can now formalize the concept of relativized prediction as follows:

Axiom 4'

actors(pln) $\neq$ { *a* }& $\exists$ *s2 Real-wrt(s,(s,s2))* & *Occurs(pln,s,s2)*
$\Rightarrow$ *Vicarious-control(a,pln,s)*.

Using this axiomatization of the solution to the knowledge preconditions problem, we can build a theory of commonsense reasoning in which benchmark planning problems can be solved. As an example, we consider the example of section 3, in which Susan plans to learn the combination of a safe by asking a cooperative agent Bob. Consider a situation $S1$. Assume that Bob in $S1$ knows the combination of some safe and that Susan knows this fact in $S1$. Consider, further, a common set of social protocols governing agents' behavior, as discussed in [Morgenstern, 1988] or [Shoham, 1993]. Examples of such protocols are: that cooperative agents will accept one another's goals if possible, and that cooperative agents are constrained to tell the truth to one another. Assume that these protocols hold for Susan and Bob in $S1$, that Susan and Bob are aware of these facts, and that both obey the S4 axioms of knowledge. Then we have the following theorem:

Theorem:
Can-execute-plan(Susan, sequence(do(Susan,ask(Bob,comb)), do(Bob, tell(Susan,comb)))

We sketch the main points of the proof. Axiom numbers refer to the axioms listed in the appendix.

We first prove the following lemmas:

Lemma1: If A and B are cooperative agents, then A can tell P to B iff A knows P.

Proof: By Axiom 5, an agent A can perform the action of telling P to B iff the knowledge preconditions, the physical preconditions, and the social protocols are all satisfied. We assume for simplicity that the physical preconditions are satisfied (Axiom 6). Moreover, all agents always know how to perform the simple act of uttering a string. (Axioms 7 and 8). It remains to satisfy the social protocol. By Axiom 9, the social protocols are satisfied iff agent A tells the truth – i.e., if he knows P. Thus, if A knows P, the social protocols are satisfied, and since the knowledge and physical preconditions are satisfied, he can tell P to B. Conversely, if he can tell P to B, the social protocols must be satisfied, and thus he must know P. $\square$

Lemma2: Assume A and B are cooperative agents. If A asks B to do Act1, and B can do Act1, then B will subsequently perform Act1

Formally,

Cooperative(a,b,s1) &
Occurs(do(a,ask(b,act1)),s1,s2)
$\Rightarrow \exists$*s3 Real-wrt(s2,s3)* & *Occurs(do(b,.act1),s2,s3)*

Proof: Axiom 10 tells us that cooperative agents adopt one another's goals. That is, if A asks B, during some interval (s1,s2) to do some act, it is then B's goal in s2 to do this action. Moreover, we have from Axiom 11 that if an agent has a goal of performing a certain action, and he can perform that action, he will subsequently perform the action. □

Note that Axiom 11 explicitly uses the concept of relativized realness. Neither a stronger nor a weaker concept will suffice. If Axiom 11 had read:
Goal(a,act,s) & Can-perform(a,act,s)
$\Rightarrow \exists$ *s2 Real(s2) & Occurs(do(a,.act),s,s2)*
then it would be false. On the other hand, if Axiom 11 had read:
Goal(a,act,s) & Can-perform(a,act,s)
$\Rightarrow \exists$ *s2 Occurs(do(a,.act),s,s2)*
it would not be strong enough to prove Lemma 2.

Indeed the proof of Lemma 2 depends on the ontology developed here. It seems unlikely that it could be proven in a standard McDermott-type branching logic.

The proof of the theorem then goes as follows: By protocol (Axiom 14), agents can ask other cooperative agents for information. Moreover, the physical preconditions and knowledge preconditions are satisfied (Axioms 12 and 13). Thus, Susan can perform the first part of her plan. Thus, Susan can vicariously control the first part of her plan (Axiom 3). We must now show that if she performs this part, Bob will perform the second part. First we must show that Bob can perform the action of telling Susan the combination. By assumption, Bob knows the combination in S1. Moreover, agents do not forget (Axiom 15). Thus, Bob knows the combination in any situation subsequent to S1. Therefore, by Lemma 1, he *can* perform the action of telling Susan the combination in S1. Now, using Lemma 2, we can show that if Susan asks Bob the combination, he will subsequently tell it to her. This means that Susan vicariously controls the second part of the plan (Axiom 4'); by Axiom 2, Susan can execute the plan consisting of the sequence *(Do(Susan,ask(Bob,combination)),Do(Bob, tell(Susan,combination)))*. □
Again, this proof will not hold in a branching temporal logic.

Note, however, that the theory is not too powerful. In particular, it will not entail degenerate plans like Nancy Kerrigan's plan, above. Thus, the theory based on relativized branching time avoids both the problems of linear time and of standard branching time.

7 Conclusion and Further Directions

In the late seventies and early eighties, many researchers ([Allen, 1984], [McDermott, 1982]) argued for the importance of a correct ontology of time. The pendulum shifted somewhat subsequently, with McDermott [1984] arguing that some ontological distinctions were not all that crucial. In particular, he argued that the difference between linear and branching time was not that great, and would probably not make much of a difference in temporal reasoning.

We have shown that, contrary to McDermott's hopes, this distinction is crucial for theories of knowledge and planning, and that in fact, neither ontology is adequate for such theories. Linear time does not allow hypothetical reasoning, and thus cannot properly handle knowledge preconditions for action and plan sequences. Branching time can handle hypothetical reasoning, but it cannot handle prediction properly, especially in hypothetical reasoning contexts. (Recently, Pinto and Reiter [93] have also noted the problems of using standard branching time.) Thus, those who ignore the issue of ontology do so at their own peril: all researchers who have used standard ontologies for reasoning about knowledge and planning have developed theories that are inadequate in some respect.

We have developed a different ontology for time, relativized branching time, which allows for relativized realness. This allows prediction in hypothetical contexts, and thus allows the proper axiomatization of knowledge preconditions. The resultant theory can handle standard benchmark problems correctly, while avoiding the anomalies of previous theories.

Relativized branching time appears promising for other research areas as well. Because it supports certain types of hypothetical reasoning, it may be a suitable ontology for counterfactual reasoning. In particular, relativized branching time may help give structure to the rather vague concept of "most similar possible worlds" which has been used (see, e.g. [Lewis, 1963]), to explain the semantics of counterfactuals such as "If I had struck a match (at S1), it would have burst into flames." In our ontology, such a sentence can be analyzed as follows: it is true if given a (typically non-real) branch segment (S1,S2) during which the match is struck, it is true on the most real branch segment of S2 that the match burst into flames. *Most similar* can be understood as the most real subtree of the endpoint of a non-real branch. Such an analysis is very preliminary but suggests promising directions for future research.

8 Acknowledgements:

The author thanks the anonymous reviewers for comments on an earlier draft of this paper.

9 Bibliography

[Allen, 1984] Allen, James: Toward a General Theory of Action and Time, *Artificial Intelligence,* vol. 23, no. 2, 1984, pp. 123-154

[Davis, 1984] Davis, Ernest: *Representations of Commonsense Knowledge,* Morgan Kaufmann, Los Altos, 1990

[Konolige, 1982] Konolige, Kurt: A First Order Formalization of Knowledge and Action for a Multi-agent Planning System, J.E. Hays and D. Michie, eds. *Machine Intelligence 10*, 1982
[Lewis, 1963] Lewis, David: *Counterfactuals*, Oxford, 1963
[McDermott, 1984] McDermott, Drew: The Proper Ontology for Time, unpublished, 1984
[McDermott, 1982] McDermott, Drew: "A Temporal Logic for Reasoning About Processes and Plans," *Cognitive Science*, 1982
[Montague, 1963] Montague, Richard: Syntactical Treatments of Modality with Corollaries on Reflexion Principles and Finite Axiomatizability, in *Acta Philosophica Fennica,* fasc. 16, pp. 153-167, 1963
[Moore, 1980] *Reasoning About Knowledge and Action*, SRI TR 191, 1980
[Morgenstern, 1988] Morgenstern, Leora: *Foundations of a Logic of Knowledge, Action, and Communication*, NYU Ph.D. Thesis, Courant Institute of Mathematical Sciences, 1988
[Pinto and Reiter, 1993] Pinto, Javier and Raymond Reiter: Adding a Time Line to the Situation Calculus
[Shoham, 1993] Shoham, Yoav: Agent-Oriented Programming, *Artificial Intelligence*, 1993

10 Appendix:

Below, a list of the axioms and definitions used in the proofs of the lemmas and main theorem of Section 6. All variables are assumed to be universally quantified unless otherwise noted. Axioms 1 through 4' are taken from sections 4 through 6 of this paper.

Axiom 1:

$(\mathit{Knows\text{-}how\text{-}to\text{-}perform}(a,act1,s1)\ \&$
$(\mathit{Occurs}(do(a,.act1),s1,s2) \Rightarrow$
$\mathit{Knows\text{-}how\text{-}to\text{-}perform}(a,act2,s)))$
$\Rightarrow \mathit{Knows\text{-}how\text{-}to\text{-}perform}(a,\text{`}sequence(\wedge act\wedge, \wedge act2\wedge)\text{'},s1)$

Axiom 2:

$\mathit{Can\text{-}execute\text{-}plan}(a,\text{`}sequence(\ \wedge pln1\wedge,\wedge pln2\wedge)\text{'},s1)$
$\Leftrightarrow$
$\mathit{Know}(a,\text{`}Vicarious\text{-}control(@a, \downarrow pln1\downarrow, @s1)\text{'},s1)\ \&$
$\mathit{Know}(a,\text{`}Occurs(\wedge pln1\wedge, @s1,s2) \Rightarrow Vicarious\text{-}control(@a,\downarrow pln2 \downarrow,s2)\text{'},s1)$

Axiom 3:

$actors(pln) = \{\ a\ \}$
$\&\ \mathit{Can\text{-}perform}(a,\text{`}action(@a,\wedge pln \wedge\ )\text{'},s) \Rightarrow$
$\mathit{Vicarious\text{-}control}(a,pln,s)$

Axiom 4':

$actors(pln) \neq \{\ a\ \}\ \&\ \exists\ s2\ \mathit{Real\text{-}wrt}(s,s2)\ \&\ occurs(pln,s,s2)$
$\Rightarrow \mathit{Vicarious\text{-}control}(a,pln,s)$.

Axiom 5:

$\mathit{Can\text{-}perform}(a,act,s) \Leftrightarrow$
$\mathit{Know\text{-}how\text{-}to\text{-}perform}(a,act,s)\ \&\ \mathit{Physsat}(a,act,s)\ \&\ \mathit{Socialsat}(a,act,s)$

An agent can perform an action if the knowledge preconditions, the physical preconditions, and the social protocols are all satisfied.

Axiom 6:

$\mathit{Physsat}(a,\text{`}tell(@b,\downarrow p\downarrow)\text{'},s)$

For the sake of this paper, it is assumed that there are no physical preconditions for communicative actions. In reality, there are a variety of preconditions, including being at the same place as the hearer (or being connected in some way).

Axiom 7:

$\mathit{Primitive\text{-}act}(\text{`}tell\text{'})$

The simple locutionary action of just uttering a string is considered to be primitive, with correspondingly simpler knowledge preconditions.

Axiom 8:

$\mathit{Primitive\text{-}act}(f) \Rightarrow$
$\mathit{Know\text{-}how\text{-}to\text{-}perform}(a,\wedge f\wedge(\wedge x1,\cdots,\wedge xn)\text{'},s)$

where all of $x1 \cdots xn$ are constants.

An agent knows how to perform any primitive action if all the arguments are constant.

Axiom 9:

$\mathit{Cooperative}(a,b,s) \Rightarrow$
$\mathit{Socialsat}(a,\text{`}tell(@b,\downarrow p\downarrow)\text{'},s) \Leftrightarrow \mathit{Know}(a,p,s)$

Cooperative agents are constrained to tell the truth.

Axiom 10:

$\mathit{Cooperative}(a,b,s1) \wedge$
$\mathit{Occurs}(do(a,ask(b,info)),s1,s2)$
$\Rightarrow \mathit{Goal}(b,tell(a,info),s2)$

If one agents asks a cooperative agent for information, the second agent will subsequently have the goal of giving over the information. The above axiom has quite a bit of syntactic sugar in it. The term "info" is shorthand for what is really going on: Agent a is asking agent b to tell him a string of the form: 'Equal(term,p)', where p is a constant. Agent b adopts the goal of telling him a string of that form. In [Morgenstern 1988], this axiom is presented without any syntactic sugar.

Axiom 11:

$\mathit{Goal}(a,act,s)\ \&\ \mathit{Can\text{-}perform}(a,act,s)$
$\Rightarrow \exists\ s2\ \mathit{Real\text{-}wrt}(s,s2)\ \&\ \mathit{Occurs}(do(a,.act),s,s2))$

If an agent has the goal of performing an act, and can perform the act, he will perform the act. Note the crucial use of the *Real-wrt* predicate.

Axiom 12:

$\mathit{Primitive\text{-}act}(\text{`}ask\text{'})$

Asking is a primitive action.

Axiom 13:

$\mathit{Physsat}(a,ask(b,info),s)$

The physical preconditions of asking for information are always satisfied.

Axiom 14:

$\mathit{Cooperative}(a,b,s) \Rightarrow \mathit{Socialsat}(a,ask(b,info),s)$

If agents are cooperating, it is always all right to ask for information.

Axiom 15:

$\mathit{Know}(a,\downarrow p\downarrow,s) \Rightarrow \forall\ s2 \succeq s\ \mathit{Know}(a,\downarrow p\downarrow,s2)$

This is the axiom of perfect memory. Agents never forget.