Proceedings of the Second Conference on Artificial Intelligence Planning Systems
(AIPS-94), Chicago, IL, 13-15 July, 1994, pp. 43-48.
Reformatted May 26, 2010 (layout may have changed)

Arguing About Plans: Plan Representation and Reasoning for Mixed-Initiative Planning*

George Ferguson **James F. Allen**
Department of Computer Science
University of Rochester, Rochester, NY
{ferguson,james}@cs.rochester.edu

Abstract

We consider the problem of representing plans for mixed-initiative planning, where several participants cooperate to develop plans. We claim that in such an environment, a crucial task is *plan communication*: the ability to suggest aspects of a plan, accept such suggestions from other agents, criticize plans, revise them, *etc.*, in addition to building plans. The complexity of this interaction imposes significant new requirements on the representation of plans. We describe a formal model of plans based on defeasible argument systems that allows us to perform these types of reasoning. The arguments that are produced are explicit objects that can be used to provide a semantics for statements about plans.

Introduction

Mixed-initiative planning involves the cooperation of two or more agents collaboratively constructing and possibly executing a plan. This includes situations ranging from automated planning assistants, where a machine aids a human in constructing a plan, to situations with multiple autonomous agents attempting to co-ordinate their activities towards a common goal. In all these situations, the agents need to talk about plans. This paper develops a representation that supports such plan communication. While we have used this to support a natural language mode of communication, it applies equally well to more formally defined communication languages such as two machines might use, or as might be used as the back-end for a graphical human-machine interface.

The traditional representations of plans are inadequate for supporting communication about plans. Formal models of plans (*e.g.*, (McCarthy & Hayes 1969),(Chapman 1987)) typically define a plan as a sequence of actions. If agents communicated plans by simply listing the sequence of actions to be performed, then this might be adequate. But this is not what happens. To see how agents communicate plans, we designed an experiment that involves two people collaborating on forming plans in a transportation planning domain involving train scheduling. The subjects could not see each other, and could only communicate by speaking into a microphone. Over thirteen hours of interaction have been collected and transcribed (see (Gross, Allen, & Traum 1993)). In no case did one agent simply describe the plan by describing a sequence of actions. Rather, agents identified the overall goal, identified subgoals to focus on, identified important actions in the plan, stated relevant facts that would help in the development of the plan, identified problems with what the other agent proposed, confirmed what the other agent suggested, and requested clarification when suggestions were not fully understood. Many of the details of the plan are never mentioned at all, and yet are implicitly agreed upon by the agents.

Clearly, the representation of a plan as a sequence of actions accounts for very little of the actual interactions that occur in a mixed-initiative planning scenario. This is not to say that we must reject the notion of a plan as a sequence of actions. It only points out that a planning system that engages in mixed-initiative planning must use a much richer representation. Similar conclusions have been made by researchers interested in plan execution monitoring and plan modification and reuse (Hayes 1975; Drummond 1989; Kambhampati & Hendler 1992; Kambhampati 1992). For these applications, it is important to represent the reasons why an action is being performed, so that an appropriate response can be made if the action fails. It is also important to record which effects of an action are important for the plan in order to verify whether an action succeeded or failed in the first place. This same information is crucial in plan communication. It allows one agent to understand the other agent's motivation behind a suggested course of action, which helps in identifying the specific role that the suggested action should play in the plan.

Consider an example. Two agents, A and B, are cooperating to construct a plan to transport medical supplies to some location. To get the supplies there, the agents need to first move them overland to the port,

*This work is supported by ONR/ARPA research grant no. N00014-92J-15121 and Air Force – Rome Labs research contract no. F30602-91-C-0010.

and then carry them by ship. A cargo ships leaves every day at between 4:00 and 6:00. If the supplies are moved by train to the ship, they will arrive at 5:00. If they are moved by truck, they will arrive at 3:00. Moving them by truck, however, will be three times as expensive. Given this scenario, there are competing constraints on a solution. Ideally, one would like a plan that guarantees to get the supplies there as soon as possible, while spending the least amount of money. Such a solution is not possible here, however. Furthermore, there is no single, global evaluation measure on the worth of plans. Each of the agents may have different priorities in evaluating plans, possibly focusing on one factor and ignoring the others.

Given this setting, one possible interaction between the two agents is as follows:

1. Agent A suggests shipping the supplies by train;
2. Agent B points out that the ship might leave at 4:00, and thus the supplies would miss today's ship;
3. Agent A points out that the ship might not leave until 6:00, and thus the supplies would make today's ship.

At this point, the agents are at a standoff. They could continue to argue in many ways. A might argue that getting the supplies there today isn't important, or B might argue that the risk of missing the ship is unacceptable. Or one of them might propose another plan:

4. Agent B suggests moving the supplies by truck instead.

Assuming that agent A finds the cost of this option acceptable, there would appear to be no arguments against this plan, and thus it would be accepted by both agents.

There are many problems to address in order to support the type of interactions in the example. In this paper we are concerned with the formal representational underpinnings. The representation proposed involves applying techniques for representing arguments to a logic of time, events and action, which supports reasoning about attempting actions and their effects. We have explored the logic of time and action in depth elsewhere (Allen & Ferguson 1994) and have applied it to plan reasoning (Allen 1991). Here we explore the use of explicit argument structures in a direct inference system, and show how, with a suitable logic of time and action, they provide a very rich representation both for talking about plans and for doing plan reasoning itself. This representation is being used to support plan reasoning in the TRAINS system at Rochester, where it supports plan construction and plan recognition algorithms that are used by a planning system that cooperatively builds plans with a person.

The rest of this paper is organized as follows. First we define argument systems formally, then apply the model to representing planning knowledge. The preceding example is used to illustrate the approach. Finally we discuss some of the issues raised by this work.

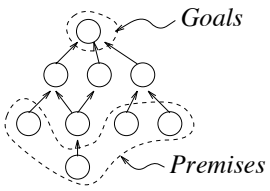

Figure 1: Graphical representation of an argument

Argument Systems

In this section we present a formal description of an argument system based on those of Loui (Loui 1987) and of Pollock (Pollock 1987; 1992).

Basic Definitions

We assume a logical language with an entailment relation $\models$, and allow a set KB of propositions that specify domain constraints against which arguments can be evaluated.

Definition 1 *An* ARGUMENT STEP *is a pair* $\langle \Phi, p \rangle$*, written "*$\Phi \rightarrow p$*," where* Φ *is a set of propositions (the* PREMISES*) and* p *is a single proposition (the* CONCLUSION*).*

An argument step is to be read something like "Φ is reason to believe p," or "The truth of Φ lends support to the truth of p."

Am agent's knowledge base includes a set of argument steps Δ that the agent uses in its reasoning. Note that in the case where we have large numbers of potential arguments, Δ may be specified schematically.

Definition 2 *An* ARGUMENT *is a set of argument steps* $\{\langle \Phi_i, p_i \rangle\}$ *drawn from* Δ*.*

Definition 3 *The* PREMISES *of an argument* A *are those propositions in the conclusions of steps with empty premises,* i.e.*,* $\{p | \langle \emptyset, p \rangle \in A\}$*. The* CONCLUSIONS *of an argument* A *are the union of the premises of steps with empty conclusions,* i.e.*,* $\bigcup\{\Phi | \langle \Phi, \epsilon \rangle \in A\}$*.*

We require that arguments be non-circular and that both $Premises(A)$ and $Goals(A)$ be non-empty for any argument A. In this case, arguments correspond to directed, acyclic graphs labeled with propositions, such as illustrated in Figure 1. An argument step $\langle \Phi, p \rangle$ corresponds to a node labeled by p with children each labeled by an element of Φ. The sources of the graph correspond to $Premises(A)$, the sinks to $Goals(A)$.

Conflict and Defeat

What makes argument systems interesting is the way arguments can conflict over the status of some proposition. In this case, we need to define how certain arguments are to be preferred over others.

Definition 4 *Two argument steps* $\langle \Phi, p \rangle$ *and* $\langle \Psi, q \rangle$ CONFLICT *if* p *and* q *are inconsistent.*

In Pollock's system, this form of conflict is called *rebuttal*, and a second form, called *undercutting*, is also permitted. The intuition is that in rebutting conflict the value of some proposition is at issue, while in undercutting conflict it is the applicability of a defeasible rule that is in doubt. In our approach, following Loui, we will impose certain criteria of defeat (below) which to some extent obviate the need for undercutting. We leave open the possibility, however, that it may be required in the future.

Given two arguments that conflict, we are interested in preferring certain arguments over others. The following definition, adapted from Loui, makes this explicit:

Definition 5 *An argument A is* AT LEAST AS SPECIFIC *as an argument B if there is a step $\langle \Phi, p \rangle \in A$ and a step $\langle \Psi, q \rangle \in B$ such that the steps conflict and $\Phi \models \Psi$. An argument is* MORE SPECIFIC *than another argument if it is at least as specific and the converse is not the case.*

Finally, we can put these pieces together and propose the following:

Definition 6 *An argument A* DEFEATS *an argument B if A conflicts with B and A is at least as specific as B.*

Definition 7 *An argument A is* UNDEFEATED *if there is no argument B that conflicts with A that is not itself defeated.*

There are complications in applying this somewhat circular definition to determine which arguments are undefeated. Pollock (Pollock 1992), for example, considers such phenomena as *collective defeat* and *self-defeat*. The details are not necessary to an appreciation of the formalism as regards planning.

Also, there might be other grounds for preferring one argument over another besides specificity. Loui (Loui 1987) considers *evidence*, *directness*, and *preferred premise*, for example. As well, there might be domain-specific criteria, such as resource use or time limits in a planning context. The specificity principle is generally accepted as appropriate and is sufficient for what follows.

Plan Reasoning Using Arguments

In this section we describe the application of argument systems to planning problems. We begin by describing the representation of causal knowledge using defeasible rules and consider the qualification problem in this light. We then present a formalization of the mixed-initiative planning example presented earlier. This is a short but interestingly complex example of a planning problem that highlights the incrementality of argumentation and the use of specificity. The example includes reasoning about uncertainty and external events.

The approach is to express our defeasible knowledge about actions and their preconditions and effects using rules in Δ. Arguments can then be constructed supporting the fact that executing certain actions will achieve certain goals. These arguments may be in conflict with or defeated by other arguments. Undefeated arguments represent plans that will succeed, given what we know and how long we searched for counter-arguments. Of course, plans can be compared or evaluated according to other factors besides defeat, such as resource usage or time constraints. Note that, among other things, this approach distinguishes goals from other effects and makes the assumptions underlying the plan explicit and thus available for reasoning.

Knowledge Representation

To keep the presentation as concise as possible, we will use a very simple representation of time, namely a discrete model where, given a time t, $n(t)$ is the next time. The representation can easily be generalized to either a situation calculus representation or the more expressive interval temporal logic (Allen & Ferguson 1994) that we actually use. An important point is that action attempts must be distinguished from their effects.

In order to do this, we introduce *events* as cognitive objects that correspond to "something that happened." Event predicates are used to organize our causal knowledge about the domain and are organized in a type hierarchy. For example, a predicate like $Load(e_t)$ is true if the event e_t describes a loading event that occurred at time t. Rules describe the necessary conditions given that an event of a particular type occurred.

In order to reason about the agent's abilities, we have to introduce *actions*. Unfortunately, actions and events are often confused, since, intuitively, for any action there is an event corresponding to that action being executed. To avoid this confusion, we will think of actions as sensory-motor *programs* that can be run by the agent; this is basically the concept of action used in the planning literature. There is a predicate $Try(\pi, t, e_t)$ which is true if program π is executed at time t (or over interval t in a more expressive logic) causing event e_t to occur. Rules specify sufficient conditions for a program's execution to result in an event of a certain type.

Causal Knowledge and Qualifications

The fundamental issue in representing knowledge for planning is how to capture the idea that execution of an action causes a certain effect, at least under certain conditions. The traditional planning view would be characterized something like the following:

$$Holds(preconds(a), t) \wedge Try(a, t, e_t) \supset Holds(effects(a), n(t)).$$

That is, if the preconditions of an action a hold at time t and we attempt action a, then the effects will hold at the next time point.[1] There are many problems with this definition, such as how to handle simultaneous actions, external events, and actions with duration. These problems can be resolved, however, by using a richer

[1] Traditional models wouldn't have events, but we include the e_t term for comparison with the later development.

temporal model, but for this paper such extensions are not important. The important idea is the intuition that actions do have preconditions, even if these are context-dependent and subject to revision.

In the context of reasoning about actual situations, what is troubling is the strength of the implication. It does not make sense with this definition, for example, to reason about execution of the action when only some of the preconditions are known to be true. Indeed, in the STRIPS model of planning, an operator cannot even be applied (*i.e.*, an action cannot be attempted) if all the preconditions do not hold. In a more philosophical sense, these axioms run into the *qualification problem*: the problem that no matter how many preconditions we write down, there will always be other things that might happen that would invalidate the axiom. If these axioms are based on the material conditional, the price of encountering an unconsidered qualification is inconsistency.

The obvious move then is to weaken the definition to use the defeasible connective "$\rightarrow$" rather than the material conditional. Then these definitions can be used in arguments that execution of an action has some effect, without saying finally that it must. We obtain something like the following, using the notation introduced in the previous section:

$$\{Holds(preconds(a), t), Try(a, t, e_t)\} \rightarrow Event(e_t)$$

where $Event$ is an event-type predicate. The definition of the event can continue to use the material conditional:

$$Event(e_t) \supset Holds(effects(e_t), n(t))$$

This is not quite enough, however, because it doesn't capture the notion that somehow the precondition *should* be true before attempting the action if it is to succeed, even if we don't know for sure that it this is the case. Clearly we shouldn't state that falsity of a precondition implies that the action won't succeed, since there might be other conditions (known or unknown) under which it would. But we can say that defeasibly, as in

$$\{\neg Holds(preconds(a), t), Try(a, t, e_t)\} \rightarrow \neg Event(e_t)$$

In fact, these two rules form the extremes of a continuum of rules relating knowledge of preconditions to successful execution of actions. Suppose we know that ϕ_1 and ϕ_2 are preconditions for action a. Then a plan that takes account of only ϕ_1 is at least reasonable, if not complete or even possible given other knowledge about the world. But it is important to be able to represent such partial or incorrect plans if we are interested in doing more than simply generating simple plans. In this case, we will need additional rules in Δ corresponding to these "partial" argument steps. In fact, for every action a with (conjunctive) preconditions $\phi_1, \phi_2, \ldots, \phi_n$ and effects ψ, there corresponds a set of defeasible rules organized in a lattice of specificity. The most specific rule is the one in which all preconditions are present. The least specific rule mentions no preconditions, *i.e.*, $Try(a, t, e) \rightarrow Event(e)$, indicating that an action can always be attempted, even if an argument that doesn't take note of preconditions may not be a very strong one. There is a similar lattice of rules regarding the falsity of preconditions and the non-occurrence of events.

Some comments are in order regarding this formulation of causal rules. First, it is independent of the underlying representation of time and action, so long as action attempts can be distinguished from their effects (*i.e.*, the Try predicate, above). Second, the approach is different from using a modal logic to express the possibility and necessity intuitions involved in reasoning about preconditions. The most modal logic could do is allow us to write that if a precondition is true then an action is possible, and if it is false then the action necessarily fails, for example. But in the defeasible reasoning approach, degrees of support are afforded through the use of defeasible rules with more or less specific antecedents. This is much closer to our intuitions about just what a precondition is, and also closer to the way people talk about preconditions in collaborative discourse about plans.

Finally, we believe that this approach to reasoning about preconditions and effects of actions is the best way to deal with the qualification problem. That is, the agent considers qualifications as long as it can or until it has considered all that it knows about, as reflected in its set of defeasible rules. If subsequent qualifications are encountered, they can be reasoned about given more time, without denying that the plan developed previously supported its conclusions, however tenuously.

Planning Example

Recall from the example presented in the Introduction that we have to get supplies X into ship S before it leaves port P, and that there are two ways of transporting the supplies. We will denote these by two actions (programs): $sendByTrain$ and $sendByTruck$. Then assuming these actions have no preconditions, their definitions are given by the following rules:

$$\{Try(sendByTrain(x, l), t, e_t)\} \rightarrow Transport(e_t, x, l, 5)$$
$$\{Try(sendByTruck(x, l), t, e_t)\} \rightarrow Transport(e_t, x, l, 3).$$

Time units are measured in hours. The event predicate $Transport$ is defined as

$$Transport(e_t, x, l, n) \supset At(x, l, t + n),$$

The action of loading ship s with supplies x has as preconditions that both the ship and the supplies must be at the dock, yielding the following definition:

$$\{At(x, l, t), At(s, l, t), Try(load(x, s, l), t, e_t)\} \rightarrow Load(e_t, x, s, l).$$

The definition of $Load$ is

$$Load(e_t, x, s, l) \supset In(x, s, t + 1).$$

Finally, we have the information that the ship S leaves port P between 4:00 and 6:00. Letting T_d denote the

time of departure, we have that $4 \leq T_d \leq 6$ and defeasible rules:

$$\begin{aligned} T_d > t &\rightarrow At(S, P, t) \\ T_d \leq t &\rightarrow \neg At(S, P, t). \end{aligned}$$

Agent A's initial plan, namely shipping the the supplies by truck, is shown in Figure 2(a). Obviously, this ignores the precondition that the ship be at the dock at the time of loading. Figure 2(b) shows agent B's a *more specific* argument that argues that the loading will fail if the ship leaves at 4:00. Figure 2(c) shows the equally specific argument supporting the successful loading of the supplies if the ship hasn't left yet. Since neither argument is more specific, neither emerges as undefeated and so, intuitively, neither plan is guaranteed. This is as it should be given the uncertainty in the scenario. Figure 2(d) shows the undefeated argument for using the truck rather than the train.

The point of this example is to demonstrate the incremental development of a plan when several agents are involved. By representing the interaction as a process of argumentation, we can account for interactions that point out of difficulties with a current proposal, or that propose alternative courses of action, as well as many other interactions beyond the scope of a traditional planning system. Further, the arguments that are built up explicitly include the types of information typically added to planners to support, *e.g.*, plan modification, while being formally well-motivated in terms of defeasible reasoning. Of course, the particular example used here is not particularly difficult for classical planning systems, although it does contain some complications involving uncertainty and external events that would cause problems for many frameworks.

Towards A Language for Plans

The description of plans as arguments allows us to define a language in which plans are objects in the ontology. Terms denoting plans are then given a semantics in terms of plan graphs, where different predicates impose different constraints on the interpretation (*i.e.*, on the graph). There are *structural* predicates, such as *ActionIn*, *Enables*, *Premise*, *Goal*, *etc.*, that are defined in terms of the structure of the plan graph. Then there are *evaluation* predicates. These include *absolute* predicates such as *Plausible* or *Impossible* and *relative* predicates that allow plans to be compared, for example, according to resource usage or time constraints. The view of mixed-initiative planning that emerges is one where agents post constraints on the shared plan using these predicates.

This language is being developed as part of the TRAINS project (Allen & Schubert 1991; Ferguson 1994), an effort to construct an intelligent planning assistant that is conversationally proficient in natural language. The plan description predicates are used in the interface between the language and discourse modules and the domain plan reasoner. The manager's utterances are result in queries being made of or constraints being posted on the plan. The plan reasoner communicates the results in terms of plan description predicates that satisfy the query or that needed to be added in order to connect an utterance to the plan. These are then used to generate responses or helpful suggestions, since often information beyond the literal content of an utterance must be added in order to incorporate it.

Discussion

There are many questions raised by this work, but space permits discussion of only a few. First, this work has clear similarities to the work on plan modification and replanning mentioned at the outset. The difference is that rather than developing data structures for traditional planners that retain information required for other tasks, we consider planning within the context of a general defeasible reasoning system. Thus, our work can be seen as a generalization and formalization of that work which can be applied to other problems involved in plan communication. Our approach also abstracts away from the underlying representation of action (*i.e.*, STRIPS).

Konolige (Konolige 1988) applies defeasible reasoning to reasoning about events, in particular to the Yale Shooting problem. The emphasis is on what types of defeasible arguments are important in reasoning about events as well as what information is necessary for adjudicating between these arguments. While the particular rules presented are largely an artifact of the simple representation of action, he argues convincingly for the relative merits of argumentation systems compared to indirect systems such as circumscription. He notes the ability to include knowledge about applicability of defeasible rules directly within the framework and the incremental nature of argumentation that we have also noted.

Konolige and Pollack (Konolige & Pollack 1989) directly apply the defeasible reasoning paradigm to plan recognition in the context of plans-as-intentions. The aim is to produce plausible arguments regarding the intentions of other agents from observations of their actions, and from the undefeated arguments to extract intentions that can be ascribed to those agents. The difference between our approaches is that we see the arguments as objects to be reasoned about rather than using argumentation to reason defeasibly about intentions. These are not irreconcilable viewpoints.

Conclusions

We have presented an explicit representation of plans as arguments that a certain course of action under certain explicit conditions will achieves certain explicit goals. We believe that in realistic mixed-initiative planning scenarios such a representation is necessary to capture the wide range of reasoning that agents do with plans. Basing the representation on argument systems provides a formal basis both for describing these forms of reasoning and for defining the semantics of plans.

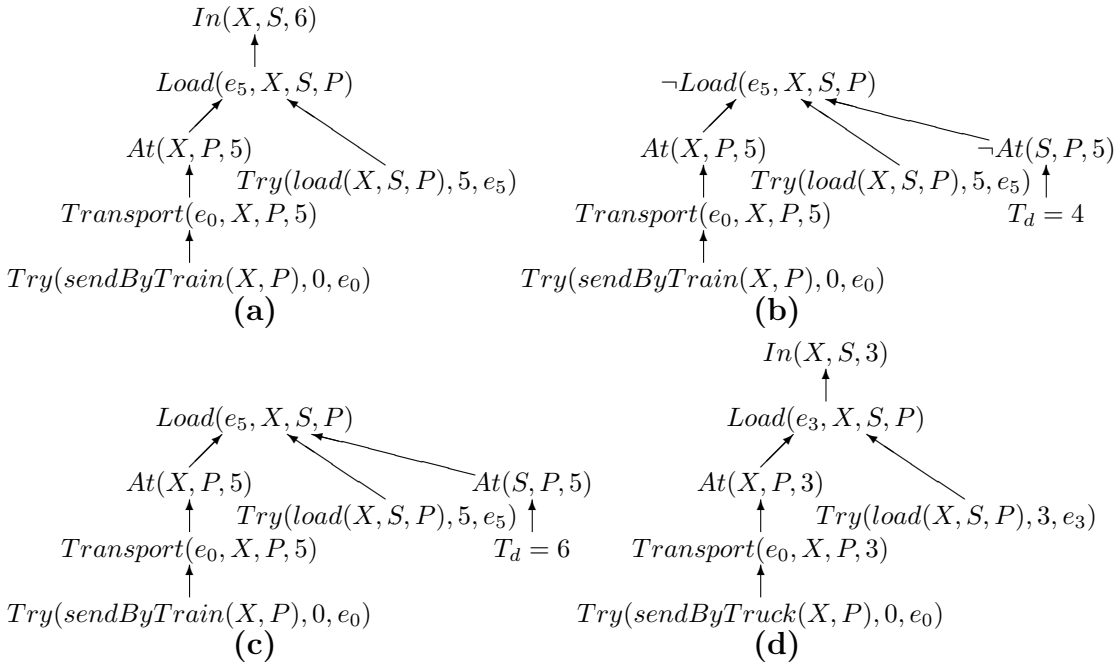

Figure 2: Four arguments about shipping supplies

References

Allen, J. F., and Ferguson, G. 1994. Actions and events in interval temporal logic. *J. Logic and Computation, Special Issue on Actions and Processes*. To appear.

Allen, J. F., and Schubert, L. K. 1991. The TRAINS project. TRAINS Technical Note 91-1, Dept. of Computer Science, U. of Rochester, Rochester, NY.

Allen, J. F. 1991. Temporal reasoning and planning. In *Reasoning about Plans*. San Mateo, CA: Morgan Kaufmann. 1–68.

Chapman, D. 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–377.

Drummond, M. 1989. Situated control rules. In *Proceedings of KR-89*, 103–113.

Ferguson, G. 1994. Domain plan reasoning in trains-93. TRAINS Technical Note 93-2, Dept. of Computer Science, U. of Rochester, Rochester, NY. To appear.

Gross, D.; Allen, J. F.; and Traum, D. R. 1993. The TRAINS-91 dialogues. TRAINS Technical Note 92-1, Dept. of Computer Science, U. of Rochester, Rochester, NY.

Hayes, P. J. 1975. A representation for robot plans. In *Proceedings of IJCAI-75*, 181–188.

Kambhampati, S., and Hendler, J. A. 1992. A validation-structure-based theory of plan modification and reuse. *Artificial Intelligence* 55:193–258.

Kambhampati, S. 1992. Characterizing multi-contributor causal structure for planning. In *Proceedings of AIPS-92*, 116–125.

Konolige, K., and Pollack, M. E. 1989. Ascribing plans to agents, preliminary report. In *Proceedings of IJCAI-89*, 924–930.

Konolige, K. 1988. Defeasible argumentation in reasoning about events. In Ras, Z. W., and Saitta, L., eds., *Methodologies for Intelligent Systems 3, Proc. of the Third Intl. Symposium*, 380–390. North-Holland.

Loui, R. 1987. Defeat among arguments: A system of defeasible inference. *Computational Intelligence* 3(2):100–106.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*. American Elsevier Publishing Co., Inc. 463–502.

Pollock, J. L. 1987. Defeasible reasoning. *Cognitive Science* 11:481–518.

Pollock, J. L. 1992. How to reason defeasibly. *Artificial Intelligence* 57:1–42.