Universal Planning: An (Almost) Universally Bad Idea

Matthew L. Ginsberg

There Are Too Many Universal Plans

To present a sharp criticism of the approach known as universal planning, I begin by giving a precise definition of it.

The key idea in this work is that an agent is working to achieve some goal and that

to determine what to do next in the pur-

suit of this goal, the agent finds its current situation in a large table that prescribes the correct action to take.

Of course, the action suggested by the table might simply be, "Think about your current situation and decide what to do next." This method is, in many ways, representative of the conventional approach to planning; however, what distinguishes universal plans from conventional plans is that the action suggested by a universal plan is always a primitive one that the agent can execute immediately (Agre and Chapman 1987; Drummond 1988; Kaelbling 1988; Nilsson 1989; Rosenschein and Kaelbling 1986; Schoppers 1987).

Definition 1.1. A universal plan

Several authors have recently suggested that a
possible approach to planning in uncertain
domains is to analyze all possible situationsis
is
to
beforehand and then store information about
what to do in each. The result is that a system
can simply use its sensors to examine its domain
and then decide what to do by finding its current
situation in some sort of a table.is

The purpose of this article is to argue that even if the compile-time costs of the analysis are ignored, the size of the table must, in general, grow exponentially with the complexity of the domain. This growth makes it unlikely that this approach to planning will be able to deal with problems of an interesting size; one really needs the ability to do some amount of inference at run time. In other words, an effective approach to acting in uncertain domains cannot be to look and then leap; it must always be to look, to think, and only then to leap. is an arbitrary function from the set of possible situations *S* into the set of primitive actions *A*.

I assume throughout this article that some fixed universal plan u describes the intended behavior of the agent being constructed; the question that I want to investigate is whether it is practical to provide the agent with an explicit description of u in its entirety as opposed to some method (such as

"invoke the planner") by which *u*(*s*) can be computed if the agent finds itself in situation *s*.

Given Definition 1.1. I can draw distinctions between the intentions of the various authors working with universal plans. Schoppers, who coined the term universal plan, seems to feel that it will be possible to equip an agent with a completely accurate description of a universal plan for acting in its particular environment. However, other authors (Agre and Chapman 1987; Drummond 1988; Nilsson 1989) concede that only some approximation to the universal plan u will be available. Rosenschein and Kaelbling (Rosenschein and Kaelbling 1986; Kaelbling 1988) seem to agree with Schoppers, although I argue that this apparent agreement misrepresents their actual beliefs.

Another choice needs to be made if an approximate universal plan is used. Agre and

0738-4602/89/\$3.50 ©1989 AAAI

Chapman and Drummond assume the agent using the approximate universal plan will continue to use it blindly in all situations, so that in some instances, the agent's actions will be unrelated to its overall goals. Nilsson, however, seems to assume that the agent will recognize these situations and do some additional planning when they are encountered. Of course, the only effective way to recognize situations in which the approximate plan differs from the optimal one is to label them explicitly; in other words, the action specified by the approximate plan must be to invoke the planner.

Thus, the three possible approaches to universal planning are as follows: First, assume that it will be possible to represent explicitly the universal plan, giving the agent's optimal action as a function of its situation (Schoppers 1987). I argue that this assumption is unlikely to be valid in practice.

Second, concede that a universal plan that always suggests a primitive action to be taken will occasionally suggest an action that is not related to achieving the agent's goals (Agre and Chapman 1987; Drummond 1988). I argue that this approach is only viable for precognitive activities such as locomotion.

Third, concede that a universal plan used to describe cognitive behavior must occasionally suggest that a planner be invoked to decide what to do next (Nilsson 1989). In this case, universal planning is only being used as a method for storing precompiled plans that the agent has available at run time.

All the arguments that I make rest on a careful examination of the techniques that have been suggested to actually represent a universal plan as data. Schoppers (1987), for example, suggests that a decision tree be constructed, and Nilsson (1989) and Rosenschein (Rosenschein and Kaelbling 1986) suggest that some sort of Boolean circuit be used. I restrict my attention to this latter approach because it is the somewhat more general of the two.¹

By a *gate*, I mean an element of the Boolean circuit being used to describe some particular universal plan. I assume that every gate has two inputs and a single output; note that because each gate has four possible input combinations, there are 16 possible gates types.

Now suppose that the domain includes n binary sensors that are used to examine the current situation and that there are a primitive actions that the planner might take. By a *circuit of size g*, I mean an arbitrary collection of g gates connected in arbitrary ways to each other, the output of the n sensors, and a actuators that correspond to decisions to take

A universal plan is an arbitrary function from the set of possible situations S into the set of primitive actions A.

each of the primitive actions. The only restriction placed on the circuit is that none of the outputs be connected to each other.²

Finally, I make two assumptions about the sensors. The first is that they are independent; so, every combination of sensor readings corresponds to a possible situation. The second is that the information given by the sensors is sufficient to determine the action the robot should take. These two assumptions in combination allow me to assume that there are 2^n distinct situations for which the universal plan must specify an action.

Proposition 1.2. There are $(2^a)^{(2^n)}$ distinct universal plans. There are at most $16^g(g + n)^{2g+a}$ distinct circuits of size g.

Proof. The proof is just a matter of counting. The number of functions (that is, universal plans) from a set of size 2^n (the set of situations) into a set of size 2^a (the set of possible combinations of actions to be taken) is $(2^a)^{(2^n)}$.

To describe a circuit consisting of g gates, I must specify, for each gate, which of the 16 possible types it is and, for each of the 2g + a input signals (one for each of the two inputs to each gate and one for each of the actuators), which of the g + n signals drives it (one for each gate's output and one for each sensor signal).

Proposition 1.3. The number of gates that can be expected to be needed to describe a universal plan in a domain with n sensors is on the order of 2^n .

Proof. This is simply the result of solving

$$\sum_{i=1}^{g} 16^{i} (i+n)^{2i+a} = (2^{a})^{(2^{n})}$$

for g.

In other words, the number of gates needed to describe a universal plan will, in general, grow exponentially with the size of the domain. Similar arguments can be made for Schoppers's decision-tree formulation.

Note that Proposition 1.3 holds independent of the value of a, the number of primitive actions that the agent can take. It might appear that even though it is impractical to construct a circuit that exactly captures the original universal plan u, you might construct a circuit that describes some alternative universal plan u' that takes an action that is approximately the same as the action described by u (perhaps the robot moves 1.5 inches instead of 1.6 in some situation). However, this approach is also, in general, impractical; if I lump all the actions into a single one, so that a = 1, and the circuit is simply trying to tell me whether I need to act I still need 2^n gates to make the decision.

Perhaps Most of the Universal Plans Don't Matter

If Schoppers and Rosenschein are to realize their intention of capturing the entire universal plan needed to act in some domain, they must argue that this universal plan can, for some reason, be described using far fewer gates than one would expect on average. This is certainly possible in principle because the various domain sensors could be sensing things such as, "You should take action 1 now."

Rosenschein and Kaelbling (1986) and Kaelbling (1988) take just this approach. Although the action component of their proposed architecture for intelligent agents is indeed expected to contain an exact description of the universal plan *u*, the perception component explicitly allows for arbitrary manipulations of the agent's primitive sensory input. These manipulations might well involve symbolic computations that do the planning themselves, turning the primitive input into forms such as, "Take action 1 now."

Schoppers, however, assumes that it will be possible to encode a universal plan without changing the sensory input. In simple domains such as the blocks-world examples considered in Nilsson (1989) and Schoppers (1987), this assumption is valid—the available sensors provide information that is near to what is needed to determine what action to take. (It is not surprising, for example, that a sensor telling you whether a particular block is clear is providing information that can be turned into an action decision with a minimum of effort.)

In general, however, the situation is not so simple. Consider the blocks-world problem of stacking nine blocks so that they spell "fruit-cake." This problem is difficult for universal planners because in a blocks world with nine blocks, there are in excess of 9! possible situations for which you need to specify actions, and this is impractical unless the available sensors somehow give you more useful information than simply, "The block labeled k is clear."

Nilsson (1988) managed to construct a universal plan for achieving this goal but only by introducing new sensors that were designed specifically for this problem.³ Nilsson assumes the existence of sensors that determine as primitives how much of "fruitcake" has been constructed so far and which block must be moved first to eventually clear the top of a given block. Given these sensors, it is hardly surprising that Nilsson can construct a universal plan for the goal in question!

More complex examples are worse still. The plan for using a tactical combination to win a chess game might be something like the following: "I'm going to attack his queen, and it'll have to go here, and then I can do this and he'll have to respond like that, and then I'll be able to checkmate him."

Well, maybe the opponent won't move his queen. A human player need not consider this possibility; he can simply assume that he will be able to win the game after the queen is captured. However, a universal planner can take no such luxuries; it needs to plan, in the most minute detail, how it will deal with every combination of moves (good and bad) by the opponent. In addition, because the move that should be selected in one chess position is likely to be very different from one that would be selected in a nearby position (perhaps where one of the pieces has been displaced by a square or changed color). there is no reason to believe that it will be possible to express the universal plan for this problem using a reasonable number of gates, decision nodes, or whatever.

What about other planning situations? In general, I am extremely pessimistic for reasons I give shortly. My pessimism aside, Schoppers is tacitly making an extremely strong claim—that the planning problems encountered by autonomous agents will be of the type for which universal plans can be represented. However, this claim needs to be defended, and he has yet to do so.

Why am I pessimistic? Consider for a moment Schoppers's original argument in favor of universal plans. He described a blocks-world domain in which a baby was wandering around, occasionally knocking down the blocks or relocating them. In an unpredictable domain, he argued, it was important to have planned for unexpected contingencies in advance.

Suppose, then, that I try to get two of Schoppers's blocks on top of one another in a particular location. I get the first block and set it down, and while I am getting the second block, the baby moves the first one. So I put the second block down, get the first block back where it belongs, and go back to the second block.

Unfortunately, the baby now moves the first block again. Sigh. I put the second block down, go back to the first block....

Or do I? In actuality, I am likely to do something fairly novel. I might try to carry the second block with me, so that I can bring both blocks to the target location simultaneously. If I can't do that, I might take the second block to the target location first, hoping that the baby won't move it. I might lock the baby in the bathroom while I move the blocks around.

The point is that the decision is a complex one. What I decide to do depends in subtle ways on the values of many sensors, not just the few indicating where the blocks are and whether they are clear. Where is the baby? Whose baby is it? After all, I can discipline my own baby more effectively than someone else's. Thus, even in this simple example, you see that deciding what to do can be a sensitive function of the sensory information available. What reason is there to believe that the goal is one for which a universal plan can be described in nonexponential form?

Agre, Chapman, and Drummond all appear to agree with the view that it is, in general, impossible to describe a universal plan in exact detail; I discuss the consequences of this conclusion in the next section. Before doing so, however, I want to discuss the implications of Proposition 1.2 on computational planning in general.

The proposition is of interest because a computer running a planning program is, after all, a collection of gates that have been wired together in some fashion. Is there any reason to believe the arguments I have made cannot be extended to imply that planning is, in general, an impossible task?

I argue there is reason because of an intuitive sense that it is possible to solve planning problems by "figuring out what to do." It is not unreasonable to hope that it will be possible to write programs that also "figure out what to do," although I would expect that such programs will suffer from the same difficulties humans do when encountering complex and unexpected situations.

Note, however, how delicate this argument is; the only reason I have for believing that a computer will be capable of solving planning problems is that I believe it will be possible for a machine to emulate my own solution methods. This assumption depends, in part, on the underlying belief that our environment behaves in predictable ways; the argument would obviously break down if there were no rhyme or reason to things.

Approximate Universal Plans

Instead of the exact universal plans discussed in the last section, Agre, Chapman, and Drummond assume that their agents will work with approximate universal plans that sometimes suggest suboptimal actions. Agre and Chapman (1987) argue that this approach is suitable if an autonomous agent is to perform tasks such as playing a video game.

This view seems entirely reasonable. The advantage of approximate universal plans is that they allow agents to react extremely quickly to their surroundings; the disadvantage is that their inflexibility makes it extremely difficult for an agent using such a plan to expend limited resources to improve its performance.

For noncognitive tasks, this choice seems fair. However, for tasks where one expects that an agent will be using its cognitive abilities to improve its performance, an approximate universal plan will, in general, be overly restrictive. (Indeed, the ability to enhance performance by expending additional mental resources seems to be a reasonable definition of the distinction between cognitive and noncognitive behavior.)

Nilsson (1989) suggests that this difficulty can be overcome by having one of the actions suggested by a particular universal plan be "plan." This approach is clearly a way around the problem. However, what Nilsson is doing is reducing the universal plan (in his case, a particular data structure he calls an action net) to a representation technique for conditional plans that are already known to the agent at run time.

If one of the actions is indeed "plan," then the contribution made by Nilsson's work must be evaluated in this light. The work on action nets is addressing a data-compaction problem, as opposed to previously recognized issues in planning or reasoning about action. Nilsson needs to present sharp arguments to the effect that first, action nets do provide compact representations of the precompiled plans available to an autonomous agent. Second, this issue is important. In other words, Nilsson must convince us (as members of the AI community) that the agent will have so many precompiled plans that they need to be compacted and that addressing this issue is as important as solving existing planning problems, such as developing computationally viable descriptions of action in a particular domain or inverting these descrip. . . because of an intuitive sense . . . it is possible to solve planning problems by "figuring out what to do."

... it is impractical for an agent to precompute its response to every situation in which it might find itself.

tions to construct a plan for acting when no precompiled plan can be applied.

Unfortunately, the only result in this direction is due to Schoppers (1989). He shows that a decision-tree description can reduce, by some 30 percent, the amount of time needed to find the precompiled plan assigned to the situation in which an agent finds itself. This savings is not that large. In addition, it seems likely that the principal cost of deciding what to do in an arbitrary situation is not finding this situation in a table of precompiled plans but rather deciding what to do if no such plan is available.

Summary

The principal point of this article is that it is impractical for an agent to precompute its response to every situation in which it might find itself. I have not argued and would not attempt to argue that it is impractical to plan for all the reasonably likely contingencies—or even for all the vaguely likely ones. In this situation, one must either expect to replan if a sufficiently unlikely or unexpected situation arises or expect the agent's abilities to be permanently limited by design decisions that were made when its approximate universal plan was constructed.

Acknowledgments

This work was supported by the Rockwell Palo Alto Laboratory, General Dynamics, and the National Science Foundation under grant number DCR-8620059. I would like to thank Andrew Baker, Ken Fertig, Leslie Kaelbling, Nils Nilsson, Stan Rosenschein, Marcel Schoppers, and David Smith for many enlightening discussions.

Notes

- 1. An expression of the form "If p then q else r" can be encoded by having the signal corresponding to p drive the output corresponding to q and passing this signal though an inverter before having it drive the output corresponding to r.
- 2. Note that I am allowing feedback loops, but I nevertheless assume that the circuit responds effectively instantly to its input. This assumption is reasonable because the circuit does not contain any explicitly time-dependent elements such as latches.

3. Note how similar this approach is to the perceptual modifications allowed by Rosenschein and Kaelbling.

References

Agre, P. , and Chapman, D. 1987. Pengi: An Implementation of a Theory of Activity. In Proceedings of the Sixth International Conference on Artificial Intelligence, 268–272. Menlo Park, Calif.: American Association for Artificial Intelligence.

Drummond, M. 1988. Situated Control Rules, Technical Report, NASA Ames Research Center. Also in 1989. Proceedings from the Rochester Planning Workshop: From Formal Systems to Practical Systems, University of Rochester, New York.

Kaelbling, L. P. 1988. Goals as Parallel Program Specification. In Proceedings of the Seventh National Conference on Artificial Intelligence, 60–65. Menlo Park, Calif.: American Association for Artificial Intelligence.

Nilsson, N. J. 1989. Action Networks, Technical Report, Dept. of Computer Science, Stanford Univ. Also in 1989. Proceedings from the Rochester Planning Workshop: From Formal Systems to Practical Systems, University of Rochester, New York.

Nilsson, N. J. 1988. Memorandum, internal to Stanford University.

Rosenschein, S. J., and Kaelbling, L. P. 1986. The Synthesis of Machines with Provable Epistemic Properties. In Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge, 83-98. San Mateo, Calif.: Morgan Kaufmann.

Schoppers, M. J. 1989. Representation and Automatic Synthesis of Reaction Plans. Ph.D. diss., Dept. of Computer Science, University of Illinois at Urbana-Champaign.

Schoppers, M. J. 1987. Universal Plans for Reactive Robots in Unpredictable Domains. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1039–1046. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Matthew L. Ginsberg received his Ph.D. in mathematics from Oxford University in 1980. He was on the faculty at Oxford until 1983, doing research in mathematical physics and computer science. During this time, he wrote a program that was successfully used to trade stock and stock options on Wall Street.

Ginsberg's continuing interest in AI brought him to Stanford University in 1983, where he is currently a senior research associate. His interests include multivalued logics, modal logics, real-time problem solving, nonmonotonic reasoning, and planning. He is the author of numerous publications in these areas and the editor of *Readings in Nonmonotonic Reasoning*, published by Morgan Kaufmann.