Abductive Planning with Sensing

Matthew Stone

Department of Computer and Information Science
University of Pennsylvania
200 South 33rd St, Philadelphia PA 19104-6389
matthew@linc.cis.upenn.edu

Abstract

In abductive planning, plans are constructed as reasons for an agent to act: plans are demonstrations in logical theory of action that a goal will result assuming that given actions occur successfully. This paper shows how to construct plans abductively for an agent that can sense the world to augment its partial information. We use a formalism that explicitly refers not only to time but also to the information on which the agent deliberates. Goals are reformulated to represent the successive stages of deliberation and action the agent follows in carrying out a course of action, while constraints on assumed actions ensure that an agent at each step performs a specific action selected for its known effects. The result is a simple formalism that can directly inform extensions to implemented planners.

Introduction

In this paper, we take a view of planning as ABDUCTION: A plan is (or at least comes with) a logical demonstration that a desired goal will be achieved, assuming the agent follows a specified course of action. To build a plan is simply to prove the goal, abductively assuming the occurrence of appropriate actions as necessary. In this framework, special-purpose planning algorithms, as in (McAllister and Rosenblitt, 1991; Penberthy and Weld, 1992), have been faithfully reconstructed and then extended to richer kinds of action using frameworks such as the event calculus (see e.g. (Shanahan, 1997)) and explanation closure (see e.g. (Ferguson, 1995)).

In existing abductive planning frameworks, and indeed most implemented planners, the plan shows that the agent can now commit to a specified sequence of actions that will achieve the goal. But a rational agent need not make all its decisions immediately. It can just as well defer choices of future actions to later steps of deliberation. Plans can and should guide these later steps of deliberation, but only if they anticipate the NEW reasons to act afforded by the agent's increased future information.

Traditional theories of action and knowledge (Moore, 1985; Morgenstern, 1987; Davis, 1994) suggest that searching for plans becomes vastly more complicated when this increasing information is taken into account. The problem is that these theories are based on NAMING plans, using object-level terms that must be specified in advance without reference to the agent's knowledge. This introduces two new and artificial search problems.

The first problem is that actions must be described indirectly in the plan. For example, suppose an agent plans to look up Bill in the phone book, then call him. From the agent's point of view, when it makes the call, it will just dial some number *n*. But since the value of *n* is settled in a future situation, *n* cannot be included in a term that specifies the plan fully in advance. Instead, the plan must include a characterization that indirectly describes this action, like *dialing Bill's phone number*. This means that even after the right action is found, a planner still has to search to find an independent description strong enough to show that the action achieves its intended effects.

The second problem is that planners must reason about FOLLOWING THE PLAN, not simply about acting in the world. Not every description corresponds to an action or plan that the agent can carry out: the description might appeal to a fact that the agent will not know. To avoid this, plan reasoning must map out the control structure of the plan in advance and compare the knowledge required by that control structure and the knowledge the agent can expect to have, at each step of execution.

One approach is to avoid these problems using heavy limitations on the syntax and semantics of parameterized actions (Levesque, 1996; Goldman and Boddy, 1996; Golden and Weld, 1996). This paper takes a very different approach. We simply add the idea of CHOICE directly into the characterization of achieving a goal. Any future situation offers the agent a number of concrete actions to take. To choose one of these, an agent simply consults its knowledge of these actions to find a good one. Thus, in our basic formulation, a plan is a demonstration that a goal state will follow a series of such feasible choices.

This definition allows plans to be constructed in which each choice is represented as it will be made. This is even true for the new, parametric actions that become available with more information. This account thus dispenses with object-level descriptions of actions and reasoning about following plans; instead, the account parameterizes actions using local Skolem constants, corresponding to the run-time variables of implemented planners. At the same time, the proof itself specifies how choices depend on one another. For example, conditional plans are realized as proofs that use case analysis to reason separately about alternative states of knowledge for the agent. Thus, proof search allows the control structure of the plan to be derived incrementally—in a way that mirrors the introduction and exploration of branches of alternative executions in implemented planners.

Choice and Future Reasons to Act

In this section, we introduce a characterization of reasons to act that explicitly refers to the successive stages of information in which an agent deliberates and chooses its future actions. This characterization can be formulated in intuitive language, as follows. A reason to choose a particular next action consists of a demonstration that this choice is the first step in a sequence of steps of deliberation and action—where the agent knows at each state what action to do next, and does it—which allows the agent to achieve its goals, thanks to a specified set of causal connections.

Choice

We derive and formalize our characterization by a running example, the bomb-in-the-toilet problem, which goes as follows: Given that one of two packages is a bomb, and that *R* can defuse a bomb-package by dunking it, how can *R* defuse the bomb? The solution is for *R* to successively dunk both packages (two actions); the one-action plan in which *R* dunks whatever package is the bomb is not a solution, because *R* cannot choose to carry it out.

We start with a purely temporal theory, in which a plan is a formal demonstration, constructed according to some theory of events and their consequences, that a sequence of actions will achieve some goal. We assume this demonstration takes the form of a deduction $\mathcal{D}$ with conclusion:

$$T, I, P \longrightarrow G \qquad (1)$$

This notation indicates that in the deduction $\mathcal{D}$, the formulas in *T*, *I* and *P* are used to derive the formula *G*. The deductive approach matches previous work on knowledge and action, and suggests an explanation-closure approach to reasoning about inertia, as in (Reiter, 1991; Scherl and Levesque, 1993) for example. *G* is a logical statement that some goal or goals hold at various points in the future. *T* is a theory describing the causal effects of actions in the domain. *I* describes the initial conditions for the planning problem (and perhaps further available information about future conditions and events). *P* records assumptions describing the occurrence and interactions of actions in the plan. In this framework, the basic problem of building a plan is to find an appropriate set of actions in *P* by abductively assuming premises, given the specification of *T*, *I* and *G*.

(1) provides a model in which the agent makes a SINGLE CHOICE OF ACTION, and evaluates the consequences of that choice of action in the different possibilities compatible with what it knows. Given a choice of *P*, we can make assumptions for the sake of argument; for example we can consider the different cases for which package is the bomb. However, assumptions in *P* are made once-and-for-all and cannot depend on what is assumed for the sake of argument; thus, *P* cannot name *whatever package is the bomb*.

A logic of knowledge can make the idea of choice explicit. We treat a single-choice, single-step plan as a proof of

$$[\mathrm{K}]T, [\mathrm{K}]I \longrightarrow \exists a[\mathrm{K}]([\mathrm{K}]\mathbf{h}a \supset [\mathrm{N}][\mathrm{K}]G) \qquad (2)$$

($[\mathrm{K}]p$ represents that the agent knows *p*; $[\mathrm{N}]p$, that *p* is true after one step of time; $\mathbf{h}a$ means that *a* is the next event to happen.) The goal formula says that the agent knows, of some concrete action *a*, that if *a* patently occurs, then as a result *G* will patently hold—in philosophical shorthand: the agent KNOWS WHAT WILL ACHIEVE *G* (Hintikka, 1971). The assumed theory *T* and facts *I* are now explicitly represented as part of the agent's knowledge. (2) matches (1) because of the wide scope of $\exists a$. Like *P* in (1), *a* here must specify a concrete action that cannot depend on assumptions in the argument assessing *a*'s known result.

Moore's definition of ability to act (Moore, 1985) also works by giving a quantifier wide scope over a modal operator. However, Moore's definition also includes a requirement that an agent knowingly select its concrete action under a given abstract description, *d*. Moore's condition can be reformulated in our notation for comparison:

$$[\mathrm{K}]T, [\mathrm{K}]I \longrightarrow \exists a[\mathrm{K}](a = d \wedge ([\mathrm{K}]\mathbf{h}a \supset [\mathrm{N}][\mathrm{K}]G)) \qquad (3)$$

The equation $a = d$ greatly increases the complexity of building a proof, by introducing cumbersome reasoning about known equalities between terms. Since we have grounded (2) in (1), we discover that it is not essential to derive an abstract description *d*, and then perform the equational reasoning to show $a = d$; naming the action abstractly does not help analyze an agent's ability to choose an appropriate action from among its concrete options.

Dependent Choice

The definition in (2) allows us to specify not only ambiguities in the state of the world but also ambiguities in the information that the agent might have. If what the agent knows is specified partially, proofs will permit an agent's chosen action to depend on its knowledge. By supporting correct reasoning about these dependent choices, proofs already allow us to describe conditional plans and plans with parameterized actions. There is thus no need for explicit, object-level constructs describing the structure of complex plans. Moreover, we continue to avoid the explicit abstract description of actions and plans using terms in the language. This allows us to maintain a very simple definition of achieving a goal by a sequence of choices.

To describe our representation of dependent choices, we return to the bomb-in-the-toilet scenario. If we suppose that the agent knows which package is the bomb, we can conclude that the agent knows enough to defuse the bomb. The agent can dunk the package it knows is the bomb.

Formally, this inference might play out in one of two ways. We can add the condition $[\mathrm{K}]b1 \vee [\mathrm{K}]b2$ to say that the agent knows whether package one is the bomb or whether package two is. Using case analysis, we can prove

$$[\mathrm{K}]T, [\mathrm{K}]I, [\mathrm{K}]b1 \vee [\mathrm{K}]b2 \longrightarrow \exists a[\mathrm{K}]([\mathrm{K}]\mathbf{h}a \supset [\mathrm{N}][\mathrm{K}]G)$$

If the agent knows package one is the bomb, it can conclude that dunking package one will defuse the bomb; otherwise, it must know that package two is the bomb and be able to conclude that dunking it will defuse the bomb.

This proof instructs the agent to make a CONDITIONAL choice of action, depending on what it knows. To see that this proof implicitly represents a conditional choice, imagine how the agent might use the proof directly to select an action while executing a plan. According to our specification, the agent will have one of two facts as part of its concrete knowledge: either $[\mathrm{K}]b1$ or $[\mathrm{K}]b2$. The proof maps out the reasoning that shows what to do in either case. Thus, the agent need only match its concrete knowledge against the cases in the proof to find which applies, then extract the appropriate component. For practical execution, we might want to use such analysis in advance, to recover an explicit conditional from a proof. Nevertheless, for efficient search, we must represent dependent choices implicitly rather than explicitly. Case analysis can be performed incrementally in proof search, so it is straightforward to derive the conditions for performing different actions piece-by-piece, as needed. Moreover, logical case analysis always interacts correctly with scope of quantifiers, so there is no possibility of proposing a conditional expression that could not form the basis of the agent's choice.

The other alternative is to add the condition $\exists x[\mathrm{K}]bomb(x)$, to say that the agent knows what the bomb is. Then we prove that the agent has a plan by picking a witness c that the agent knows is a bomb and showing that the agent knows dunking c defuses the bomb. We can regard this proof as instructing the agent to make a PARAMETERIZED choice of action, depending on what it knows.

Again, as with an abstract, symbolic description of a parameterized action, this proof has enough information for the agent to choose a successful action. If the partial specification of its knowledge is correct, its concrete knowledge includes a fact $[\mathrm{K}]bomb(x)$ for some object x. The proof spells out what to do with that value x: use it in place of the arbitrary witness c that the proof assumed. Doing so allows the agent to derive from the proof a concrete reason for a specific action. Again, by comparison with an explicit description, we see that the logical treatment, in terms of scope, naturally guarantees that only information the agent has can affect its choices. There could be no possibility of proposing a described action whose referent the agent did not know.

Sequenced Choice

We can call these arguments indirect assessments of an agent's plan. Indirect assessments allow an agent to determine the options available to itself in the future. Here is an example. Suppose we equip an agent R with a bomb-detector in the initial bomb-in-the-toilet scenario. R can describe what would hold after it used the bomb detector in the indefinite way just outlined: R would know which package is the bomb. Therefore, in the next step, R could choose to defuse the bomb. Thus, R already knows that in two steps of deliberation and action (choosing first to detect and then to dunk) the bomb will be defused.

This argument gives R an indirect reason to use the bomb detector now. The proofs we accept as plans must have a staged structure to reflect this staged introduction of future reasons to act. We should represent R's goal thus:

$$\exists a[\mathrm{K}]([\mathrm{K}]\mathbf{h}a \supset [\mathrm{N}]\exists a'[\mathrm{K}]([\mathrm{K}]\mathbf{h}a' \supset [\mathrm{N}][\mathrm{K}]G))$$

This fits R's argument. R first chooses a based on what R knows now. R's choice of a must enable R to choose a good action a' in the next step, based on what R knows then. There, R will choose a' by reasoning that a' brings about G. In all, G is nested under three $[\mathrm{K}]$ operators. Each inserts a boundary corresponding to a new stage of deliberation as R assesses its progress toward the goal. Each may be preceded by an existential quantifier for any action selected at that stage.

We can generalize this to longer plans using a recursive definition. At each step, we identify an action to do next based on information then available, and assume this action occurs; we then make sure that any remaining actions will be identified when needed, until the goal is finally achieved. We use $can(G, n)$ to denote the condition whose proof constitutes a plan to achieve the goal G in n further steps of action; $can(G, n)$ is defined inductively:

$$can(G, 0) \equiv [\mathrm{K}]G$$
$$can(G, n+1) \equiv \exists a_n[\mathrm{K}]([\mathrm{K}]\mathbf{h}a_n \supset [\mathrm{N}]can(G, n))$$

This recursive definition directly reflects the staged process by which successive actions are selected and taken.

In describing the knowledge an agent needs to follow a plan p, (Davis, 1994) uses a similar staged definition. Simplifying somewhat, and adapting the notation of (3), the agent satisfies $can(G,p)$ to follow p and achieve G:

$$\exists a_n[\mathrm{K}](a_n = next(p) \wedge ([\mathrm{K}]\mathbf{h}a_n \supset [\mathrm{N}]can(G, rest(p))))$$

As with Moore, this presupposes an overall abstract description of the course of action being carried out and appeals to complicated reasoning to determine the *next* action to match that course of action. We have seen that we can give a logical analysis of what an agent can choose to do without separately constructing or reasoning about such a description of a plan.

Formalizing Knowledge and Time

This section and the next section consider the construction of plan-proofs. For a (k-step) plan we need proofs of

$$[\mathrm{K}]T, [\mathrm{K}]I \longrightarrow can(G, k)$$

Such proofs can in fact be constructed in a very similar way to plan-proofs in a purely temporal theory.

This section describes the underlying logic. We adapt the model of (Moore, 1985) to introduce a type distinction between possible worlds and states in time. We capture the same inference schemes as Moore, but can apply recently-developed equational translation methods for efficient modal reasoning, e.g. (Ohlbach, 1991).

Each world in our models is populated by three sorts of entities. There are the ordinary INDIVIDUALS that are named by first-order terms in the language (like actions and objects). Then there are STATES, naming particular possible times in the history of the world. Finally there are UPDATES that describe possible ways a state could evolve in one moment of time. If s is a state and τ is an update, then $s;\tau$ names the state denoting the result of performing update τ in state s. Temporal operators are interpreted as quantifiers over updates; thus $[\text{N}]p$ is true at state s in world w if for any update τ in w, p is true at $(s;\tau)$ in w. Each world resolves all ambiguities about a particular state. But what could happen later is still up in the air. You can think of the states of the world as recording the REAL POSSIBILITIES of how things could turn out in that world: so each world is like a situation calculus model.

The worlds themselves are related by EPISTEMIC TRANSITIONS. If w is a world and α is an epistemic transition, then in w, for all the agent knows, it could be in $w;\alpha$. Thus, $[\text{K}]p$ is true at state s in world w if for any transition α, p is true in state s at world $w;\alpha$. By including an identity transition 1 with $w; 1 = w$, we ensure that no fact can be known unless it is true. We ensure that an agent is aware of the facts it knows by closing transitions under an operator of composition, $\star$: $w;(\alpha \star \beta) = w;\alpha;\beta$.

The entities in $w;\alpha$ include all the entities—individuals, states, and updates—in w, but may include others. A consequence of these increasing domains is the constraint of memory that $[\text{K}][\text{N}]p \supset [\text{N}][\text{K}]p$ but not vice versa. Here is why. The new entities indicate the agent's potentially limited knowledge about what exists—and what can happen. For all the agent knows now, the future might include any CONCEIVABLE POSSIBILITY—found by looking in each world $w;\alpha$ at states of the form $s;\tau'$ with τ' an update in $w;\alpha$. This corresponds to the semantics of $[\text{K}][\text{N}]p$. But as time passes, only the REAL POSSIBILITIES can actually be reached; the agent's knowledge in real possibilities is found by looking in each world $w;\alpha$ at states of the form $s;\tau$ with τ an update in w. This corresponds to the semantics of $[\text{N}][\text{K}]p$. Since there can only be more values for τ' than for τ, what an agent knows now about future alternatives is always less than what it will know about them.

The model is set up for PREFIX theorem-proving techniques for modal logic (Wallen, 1990; Ohlbach, 1991). Instead of proving that worlds and states are related by accessibility, prefix techniques use terms for worlds and states that directly encode accessibility. We follow (Ohlbach, 1991) in presenting prefix techniques by means of a semantics-based translation to classical logic (with sorts and equality), where modal operators are replaced by explicit quantifiers over transitions and updates. After translation, modal reasoning follows directly from the classical case.

The only trick in the translation is the handling of the increasing domains of individuals across possible worlds. We use compound terms $t@w$ where w names the world at which the referent for t is first defined—the DOMAIN of t. As arguments of relations involving individuals and states, any constant symbol or free variable t is immediately translated as $t@w_0$, where w_0 represents the real world. Bound variables are assigned an appropriate domain as quantifiers are translated. This translation depends on whether the quantifier is instantiated or Skolemized. At a world w, Skolemized quantifiers introduce a term $t@w$ that cannot be assumed to exist before w. So at Skolemized quantifiers we replace (argument occurrences of) the old bound variable x by a new term $x@w$. Other quantifiers are instantiated; at a world w they may take on any value $t@u$, provided that this t exists at w (given it first exists at u). To meet the proviso, we must find a path $u; v = w$, showing that u is a prefix of w, written $u \leq w$. So at instantiated quantifiers we replace the old bound variable x by a new term $x@u$ where u is a new restricted variable over worlds.

$$
\begin{aligned}
&[R(t_1,\ldots,t_k)]^{d,w,\pm} \equiv R(t_1,\ldots,t_k,d,w)\\
&[A \wedge B]^{d,w,\pm} \equiv [A]^{d,w,\pm} \wedge [B]^{d,w,\pm}\\
&[\neg A]^{d,w,\pm} \equiv \neg[A]^{d,w,\mp}\\
&[[\text{K}]A]^{d,w,\pm} \equiv \forall\alpha[A]^{d,w;\alpha,\pm}\\
&[[\text{N}]A]^{s@v,w,+} \equiv \forall\tau\forall u(u \leq w \supset [A]^{s;\tau@u,w,+})\\
&[[\text{N}]A]^{s@v,w,-} \equiv \forall\tau[A]^{s;\tau@w,w,-}\\
&[\forall x A]^{d,w,+} \equiv \forall e \forall u(u \leq w \supset [A[e@u/x]]^{d,w,+})\\
&[\forall x A]^{d,w,-} \equiv \forall e[A[e@w/x]]^{d,w,-}
\end{aligned}
$$

Figure 1: Translation $[\cdot]^{\cdot,\cdot,\pm}$ to classical logic

For completeness, the translation that we have just outlined informally is given precisely in Figure 1. The translation turns a modal formula A into a classical formula $[A]^{s_0@w_0,w_0,\pm}$ depending on the initial state s_0, the real world w_0 and whether A is assumed (+) or to be proved (-). It looks more complicated than it is: the translation just annotates terms and quantifiers with explicit domains and annotates atomic relations with an explict world and state of evaluation. The translation requires us to reason with the equations

$$E \equiv w;(\alpha \star \beta) = w;\alpha;\beta, \;\; w;1 = w$$

Using this translation, a plan is just a classical deduction with the following conclusion:

$$E, [[\text{K}]T, [\text{K}]I]^{s_0@w_0,w_0,+} \longrightarrow [can(G,k)]^{s_0@w_0,w_0,-}$$

Abduction

In this section, we recast this DEDUCTIVE approach to planning as an ABDUCTIVE problem, in which action occurrences are assumed as needed. The recursive definition of *can* already outlines a sequence of assumptions with a common content: at a particular stage of action and deliberation, the agent selects and performs an appropriate action. More precisely, proving $can(G,k)$ introduces, in lock-step with the introduction of temporal transitions, action assumptions that all take the form

$$\forall\beta.\mathbf{h}(\underbrace{e_i@u_i}_{\textit{real action}}, \underbrace{t_i@m_i}_{\textit{real state}}, \underbrace{m_i;a_i;\beta}_{\textit{known world}}) \qquad (4)$$

Here $u_i \leq m_i$. m_i represents the agent's view of what is REAL when the action is chosen; $t_i@m_i$ is a real state introduced at m_i by a goal quantifier; and e_i is some real action. Meanwhile, because the assumption is applicable at any world $m_i;a_i;\beta$, it can contribute only to what is KNOWN at m_i. By encoding the evaluation of a CONCRETE action for KNOWN effects, this form concisely distills the notion of choice.

Because the assumptions are indistinguishable, we can make them as needed. Thus, we can offer a purely abductive presentation of the proof search problem for building a plan to achieve G after k steps of deliberation and action. We simply prove $([\text{K}][\text{N}])^k[\text{K}]G$, making action assumptions of the form in (4) where necessary. This abductive approach eliminates ambiguities in proof search: there is only one way to assume a new action but there are many ways to match a sequence of uninstantiated actions (assumed independently). It also helps strengthen the connection between this theory and implemented planners: implemented planners also add actions one by one, as necessary.

The derivation of this abductive characterization in part depends on how formulas are represented using Skolem terms, logic variables and unification, according to a particular theorem-proving technique. We prefer to follow a LOGIC PROGRAMMING proof search strategy, as characterized in e.g. (Miller *et al.*, 1991; Baldoni *et al.*, 1993). In logic programming proofs, the first actions taken are always to decompose goals; this matches the strategies of special-purpose planning algorithms, and moreover allows modal operators in planning goals to be processed by introducing fresh constants independent of actions. (On the use of constrained constants for Skolem terms more generally, see (Bibel, 1982).) As we prove $can(G,k)$ by this strategy, the formula is decomposed level-by-level. Level n requires us to decompose a goal translated from $\exists a[\text{K}]([\text{K}]\mathbf{h}a \supset [\text{N}]\ldots)$; the implication introduces a new assumption of the form in (4).

This explains the source of the assumptions in (4). But is abduction sufficiently restricted? Suppose an assumption instantiates the sequence t_i to a particular time $s_j@w_j$. Then e_i first exists at some world $u_i \leq w_j$. If the assumption contributes to the ultimate proof of G, moreover, a_i can only equal α_j. Thus the instantiated assumption could just as well have been explicitly made in decomposing the formula $can(G,k)$.

Key Examples

The last three sections have outlined a logical approach to planning based on an analysis of an agent's ability to choose. To plan, an agent describes its goal in a form that indicates that the goal can be reached after a sequence of steps not only of action (corresponding to temporal updates) but also of deliberation and choice (corresponding to modal transitions). At each step, an agent must choose a concrete next action based on its known properties; this restriction corresponds directly to constraints which distinguish the possible worlds where actions and times are defined from the worlds where action assumptions can be used.

Run-time variables and knowledge preconditions

In this section, we first show how our framework allows the results of one action to provide parameters for later actions—unlike (Levesque, 1996; Goldman and Boddy, 1996). We return to the example of the bomb-in-the-toilet, formalized as in figure 2. The agent knows there is a bomb, knows it has a detector and knows it can dunk. The agent must defuse something in two steps. (In figure 2, $[\text{H}]p$ abbreviates that p is true indefinitely; explanation closure axioms are omitted as this proof goes through without them.) This translates

1 $[\text{K}]\exists b.bomb(b)$
2 $\exists a[\text{K}][\text{H}]\forall x(bomb(x) \wedge \mathbf{h}a \supset [\text{N}][\text{K}]bomb(x))$
3 $[\text{K}]\forall x\exists d[\text{K}][\text{H}](bomb(x) \wedge \mathbf{h}d \supset [\text{N}]defused(x))$

Figure 2: Bomb-in-the-toilet with detector.

into the goal:

$$defused(b(\alpha)@w_0;\alpha,\ s_0;\tau;\tau'@w_0;\alpha;\alpha',\ w_0;\alpha;\alpha';\beta)$$

$b(\alpha)$ Skolemizes b; other first-order terms will be Skolemized similarly; here, $b(\alpha)$ could also be found by unification during proof search. The proof requires two actions:

$$\forall\beta.\mathbf{h}(a@w_0,\ s_0@w_0,\ w_0;\alpha;\beta)$$
$$\forall\beta.\mathbf{h}(d(\alpha,b(\alpha))@w_0;\alpha,\ s_0;\tau@w_0;\alpha,\ w_0;\alpha;\alpha';\beta)$$

The first assumption considers the result of the immediate real action a of using the detector, assessed in worlds $w_0;\alpha$ compatible with what we know initially. The second assumption considers the result of dunking the hypothetical object $b(\alpha)$—a real action in world $w_0;\alpha$—in worlds $w_0;\alpha;\alpha'$ compatible with what we know after one step. The reader can readily flesh out this proof along the outlines suggested earlier, after first computing the translation and Skolemization of the clauses of figure 2.

Note how we represent the choice of dunking $b(\alpha)$ directly. The agent will learn from using the detector that $b(\alpha)$ is a bomb; the proof relies on the fact that the agent has this knowledge. Encoding this into the proof is enough—for, as we saw earlier, this is enough information to allow the agent later to extract what to do, by matching its concrete knowledge against the abstract knowledge the proof supposes. So we need not describe the dunking, as do (Moore, 1985; Morgenstern, 1987; Davis, 1994).

A comparison is instructive with a representative implemented planning language with similar plans, SADL (Golden and Weld, 1996). In their plans for such examples, sensing introduces a RUN-TIME VARIABLE storing the observed value; these run-time variables can be then appear as arguments to later actions. The terminology suggests some inherent departure from logic. On the contrary, such variables correspond exactly to Skolem terms like $b(\alpha)$, naming new abstract entities that exist only at remote worlds. Recognizing this logical status for run-time variables explains why such variables are treated existentially and why—in view of the "knowledge precondition" that only concrete actions can be chosen—they can serve as parameters only to actions chosen in future deliberation. At the same time, it confirms our contention that an agent's internal representation of its future actions need not be a timeless, abstract description of that action.

Knowledge preconditions for actions and plans

The next example shows, as (Golden and Weld, 1996) argue, that actions may have to be performed with different knowledge in different circumstances. However, it also shows that this variation is a natural component of a logical approach to planning—not an argument against it.

Consider a domain with a safe. If the agent dials the combination to the safe, the safe patently opens; if the agent

4 $[\text{K}]\forall sv\exists o[\text{K}][\text{H}](closed(s) \wedge combo(s,v) \wedge \mathbf{h}o \supset$
$[\text{N}][\text{K}]open(s))\wedge$
$[\text{K}][\text{H}](closed(s) \wedge \neg combo(s,v) \wedge \mathbf{h}o \supset$
$[\text{N}][\text{K}]closed(s))$
5 $[\text{K}](closed(d_0))$
6 $[\text{K}]([\text{H}]combo(d_0,n_0) \vee [\text{H}]\neg combo(d_0,n_0))$
7 $[\text{K}]\forall s[\text{H}]\neg(open(s) \wedge closed(s))$

Figure 3: Safe problem

dials something else, the safe patently remains closed. The safe starts out closed, has a constant combination, and can't be open and closed at once. We formalize the situation in figure 3. Suppose the agent wants to open the safe d_0 in one step—in our theory, to build this plan requires proving:

$$open(d_0@w_0,\ s_0;\tau@w_0;\alpha,\ w_0;\alpha;\alpha')$$

This cannot be proved abductively unless the agent knows the combination to the safe. Let's add that assumption:

$$\exists v[\text{K}]combo(d_0,v)$$

Then we can assume the real action of dialing this combination for assessment according to what the agent knows:

$$\forall\beta.\mathbf{h}(o(1,d_0,v)@w_0,\ s_0@w_0,\ w_0;\alpha;\beta)$$

This allows us to complete the plan straightforwardly, by applying the first rule of clause 4.

By comparison, suppose the agent merely wants to determine in one step whether the combination to the safe is n_0 or not. This goal is represented in modal logic as $[\text{K}]combo(d_0,n_0) \vee [\text{K}]\neg combo(d_0,n_0)$. It translates into a planning problem to prove

$$combo(d_0@w_0,\ n_0@w_0,\ s_0;\tau@w_0;\alpha,\ w_0;\alpha;\alpha';\beta)\vee$$
$$\neg combo(d_0@w_0,\ n_0@w_0,\ s_0;\tau@w_0;\alpha,\ w_0;\alpha;\alpha';\gamma)$$

This is a weaker statement than the goal for the previous problem—for starters, it contains disjunction. We can prove this abductively—without assuming knowledge of what the combination of the safe is—by considering the known consequences of attempting to dial n_0:

$$\forall\beta.\mathbf{h}(o(1,d_0,n_0)@w_0,\ s_0@w_0,\ w_0;\alpha;\beta)$$

(The proof is interesting. As the reader can work out, it uses nested case analysis to ANTICIPATE the agent's ability to correctly EXPLAIN the observed results of dialing.)

Conclusion

This paper has laid a new, clean groundwork for bridging formal and implemented accounts of sensing and planning. Further ties can be expected. Our account is compatible with partial-order representations of time (and modality) as in planners—provided path equations are represented and solved by corresponding constraints (Stone, 1997). Prospects are also good for reasoning about inertia efficiently by adapting the threat-resolution techniques used in implemented planners. The obvious starting point would be to use argumentation or negation-as-failure as in (Ferguson, 1995; Shanahan, 1997) on a world-by-world basis.

Acknowledgments

Thanks to Doug DeCarlo, Chris Geib, Leora Morgenstern, David Parkes, Mark Steedman and Rich Thomason for comments and discussion. Supported by an IRCS graduate fellowship and NSF grant IRI95-04372, ARPA grant N66001-94-C6043, and ARO grant DAAH04-94-G0426.

References

M. Baldoni, L. Giordano, and A. Martelli. A multimodal logic to define modules in logic programming. In *ILPS*, pp 473–487, 1993.

W. Bibel. *Automated Theorem Proving*. Vieweg, 1982.

E. Davis. Knowledge preconditions for plans. *J. of Logic and Comp.*, 4(5):721–766, 1994.

G. M. Ferguson. *Knowledge Representation and Reasoning for Mixed-Initiative Planning*. PhD thesis, University of Rochester, 1995.

K. Golden and D. Weld. Representing sensing actions: the middle ground revisited. In *Proc. of KR*, 1996.

R. P. Goldman and M. S. Boddy. Expressive planning and explicit knowledge. In *Proc. of AIPS*, pp 110–117, 1996.

J. Hintikka. Semantics for propositional attitudes. In Linsky, *Reference and Modality*, pp 145–167. Oxford, 1971.

H. J. Levesque. What is planning in the presence of sensing? In *Proc. of AAAI*, pp 1139–1146, 1996.

D. McAllister and D. Rosenblitt. Systematic nonlinear planning. In *Proc. of AAAI*, pp 634–639, 1991.

D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

R. C. Moore. A formal theory of knowledge and action. In Hobbs and Moore, *Formal Theories of the Commonsense World*, pp 319–358. Ablex, 1985.

L. Morgenstern. Knowledge preconditions for actions and plans. In *Proc. of IJCAI*, pp 867–874, 1987.

H. J. Ohlbach. Semantics-based translation methods for modal logics. *J. of Logic and Comp.*, 1(5):691–746, 1991.

J. S. Penberthy and D. S. Weld. UCPOP: a sound, complete partial order planner for ADL. In *Proc. of KR*, pp 103–114, 1992.

R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, *AI and Mathematical Theory of Computation*, pp 359–380. Academic Press, 1991.

R. B. Scherl and H. J. Levesque. The frame problem and knowledge-producing actions. In *Proc. of AAAI*, pp 689–695, 1993.

M. Shanahan. Event calculus planning revisited. In *Proc. of ECP 97*, LNAI 1348, pp 390–402. Springer, 1997.

M. Stone Efficient constraints on possible worlds for reasoning about necessity. Report IRCS 97-7, University of Pennsylvania.

L. A. Wallen. *Automated Proof Search in Non-Classical Logics*. MIT Press, 1990.