ELECTRONIC NEWSLETTER ON
REASONING ABOUT ACTIONS AND CHANGE

Issue 98071 Editor: Erik Sandewall 18.9.1998

Today

Submitting your article to ETAI's open review discussion is hard work, as Rob Miller commented in the general discussion a few months ago. David Poole is presently at the center of attention, and in today's issue he gives his answers to Ray Reiter's extensive invited review that appeared in the ENRAC Newsletter of 11.9. Both the review and the answer is quite substantial!

ETAI Publications

Discussion about received articles

Decision Theory, the Situation Calculus and Conditional Plans

by David Poole

Comment from: **David Poole**

Ray,

Thanks for your comments. I will try to explain everything so that it makes sense, and try to explain why I am doing what I am doing. I'll answer the questions in order, although it may help to read the guide to the perplexed first.

Technical issues

1. Examples 2.8 and 2.9.

The predicate $pickup\_succeeds(S)$ *in the first clause.*

(a) It is not parameterized by the thing – in this case key – being picked up. It should be $pickup\_succeeds(key, S)$.

What it says now is that the probability that picking up something (in this case a key) actually achieves the agent carrying the object doesn't depend on what is being picked up (nor does it depend on the position, as long as the robot and key are at the same position). You are right in that this probably isn't what you want. [This would really only make a difference if the key term was a variable.] As it stands now, $pickup\_succeeds(key, S)$ should probably be $pickup\_succeeds\_in\_carrying\_key\_when\_robot\_and\_key\_are\_at\_same\_position(S)$
.

> *Presumably, you need one of these predicates for each action type, e.g. putdown_succeeds, goto_succeeds, etc. Therefore, why not introduce a generic predicate* $succeeds(action, S)$*, in which case, you can write* $succeeds(pickup(key), S)$*,* $succeeds(goto(loc), S)$*, etc? Similarly, you can have a relation* $fails(action, S)$*.*

In this case *succeeds* would not only be a function of the action and the situation, but also a function of the condition being achieved (in this case that the key is being carried), and a context (in this case that the robot and the key are at the same position). But this is all that the rules say. The rules should be seen as axioms of conditional effects (see below).

> *Now,* C_0 *contains* $union(succeeds(pickup(key), S), fails(pickup(key), S))$*, where the union is over all ground instances of these atoms.*

I am not sure what you mean by the union, but $C0$ would contain infinitely many 2-element sets. (See the answer to the next point.)

> *Incidentally, your specifications of what sets are in* C_0*, e.g. bottom p. 13, is an abuse of notation, since it looks like an object level sentence, when in fact it is a metalevel statement for which universal quantification over situations is not defined.*

Yes. I mean for each situation term S the set $\{pickup\_succeeds(S), pickup\_fails(S)\}$ is in $C0$. That is, I am quantifying over terms in the language.

Here $C0$ is a set of 2-element sets of ground terms. In this case $C0$ is infinite but countable. I could have made it a set of open formulae, in which case it could be finite, but this would have complicated the presentation for no obvious gain.

> *(b) In axiomatizing at on p. 20, you do not include a goto_succeeds atom in the clause body. Does the atom* $would\_not\_fall\_down\_stairs(S)$ *play this role?*

Yes.

> *2. In the conditional plan of p. 21, don't you want* $at(key, r101)$ *instead of at_key?*

No. *at_key* is the sensor defined in Example 2.11 that (noisily) senses whether the robot is at the same position as the key. The robot can only condition on its sensed values. The robot doesn't know if the key is at room r101; it only knows what its sensors tell it. [You could also imagine that a robot could also condition on internal state variables, and what can be computed from internal state variables, which is hinted at in Section 2.10.]

> *3. Consider the axiom of Example 2.9 and your comment "Note that this implies that putting down the key always succeeds". Suppose not. Then is it the case that the axiom should be modified by replacing the atom* $A \neq putdown(key)$ *by something like*
>
> $$\neg(A = putdown(key) \wedge putdown\text{-}succeeds(key, S))$$
>
> *If so, perhaps this should be pointed out. (See my call below for a guide for the perplexed.)*

This is sort of correct, but it doesn't get the probabilities right. I would expect that the probability of keeping the key to be independent of of the other two cases. There are three free parameters to be assigned probabilities, so we need three choice alternatives (when we only have binary alternatives). See the guide for the perplexed below; I do this example in more detail.

> *4. In the axiom of Example 2.9, you include a choice $keeps\_carrying(key, S)$, which you introduce because "there may be a probability that the agent will drop the key". You then axiomatize this with $P0(keeps\_carrying(key, S)) = .95$ But I can't think of any realistic way to axiomatize this probability when only the parameter S is available.*
>
> *Basically, you want to describe the probability that the robot doesn't lose the key during the interval between the occurrence of the last action in S, when $carrying(key, S)$ was true, and the occurrence of the next action A. In other words, you want to describe the probability that the robot is still carrying the key at situation $do(A, S)$ given it is carrying it in situation S. But that depends, at the very least, on WHEN this next action A occurs. Now on one account of adding time to the sitcalc (Reiter, Proc KR'96), an action can take an additional temporal argument denoting that action's occurrence time, for example, $openDoor(t)$. Then one expects that $P0(keeps\_carrying(key, S))$ given that $openDoor(10)$ is the next action should be greater than this probability given that $openDoor(1000)$is the next action. So as far as I can tell, the choice $keeps\_carrying(key, S)$ should take another argument, namely A, or perhaps simply $time(A)$, where $time(A)$ is the occurrence time of the next action A.*

Yes. That is right. If the agent could drop the key at any time, we need to model (or learn) how the agent may drop the key as a function of time. Presumably this would be modelled as a Poisson process (assuming it is equally probable that the agent can drop the key at any time). As outlined on page 1, I haven't dealt here explicitly with time.

The axiomatization I gave assumes that the agent can hold on to the key tightly enough so whether it drops the key isn't a function of time.

The need for action preconditions

> *On p. 8, you claim "None of our representations assume that actions have preconditions; all actions can be attempted at any time." I'm sceptical of this claim. Consider your example domain, where you adopt the axioms $P0(pickup\_succeeds(S)) = .88$ Just prior to this, you say "$P0(pickup\_succeeds(S))$ reflects how likely it is that the agent succeeds in carrying the key* given that it was at the same position as the key and attempted to pick it up*". So you can't seriously write $P0(pickup\_succeeds(S)) = .88$ because here S is any situation, including one in which the agent is 1000 kms from the key. I take it that what you intended to write was something like*
>
> $$at(key, Pos, S) \wedge at(robot, Pos, S) \rightarrow P0(pickup\_succeeds(S)) = .88(*)$$
>
> *If this is the case, then it is exactly here where you are using action preconditions, namely, in characterizing those situations in which an action has a high probability of succeeding. It also appears that you are appealing to action preconditions in the axiom of Example 2.8*

> *(namely, $at(robot, Pos, S) \wedge at(key, Pos, S)$). Probably, these atoms should be omitted because their effects are realized in the above axiom (*).*

There are two separate issues here, first the meaning of preconditions, and secondly how the atomic choices act in contexts where they aren't applicable.

There are three things that could be meant by a precondition of an action:

> *A condition under which the action can be done. A condition under which some effect follows from the action. A condition under which the action should be done.*

It is only the second, which are often called conditional effects, that I represent. The first is what I meant as "preconditions", which I don't have. (These correspond to your *Poss* predicate, if I am not mistaken). This shouldn't really be seen as a feature, but as a necessary evil. It means that I have to axiomatize the effect of an action under all conditions (including the conditions under which you may say the action is not possible). I do this because, in general, an agent may carry out an action even if it doesn't know that the action is "possible" (in the normal sense). The third point, where the action should be done, is obtained because I want to build sensible agent that maximize utility, but I also want to be able to evaluate how good arbitrary agents are, not just good ones.

The second issue is a bit more subtle. Let's assume that the choice alternatives are all binary (non-binary alternatives are useful when a relation is functional). In this case, there are the same number of alternatives as there are free parameters (conditional probabilities that can be independently assigned) in the probability model. For example, when converting a Bayesian network to an ICL theory, there are the same number of alternatives as there are numbers to be assigned in the Bayesian network. You typically have, for each effect e, rules of the form:

$$e \leftarrow context_1 \wedge ac_1$$

$$\ldots$$

$$e \leftarrow expcontext_k \wedge ac_k$$

where the $context_i$'s are disjoint and covering. In this case we can assign the probabilities of the atomic choice aci as the conditional probability:

$$P0(aci) = P(e \mid context_i)$$

Your question on conditions for $P0$ could be recast as: What happens to the ac_i in the context when the corresponding rule is not applicable? Basically it gets marginalized out. We don't have to worry about it. The abductive characterization of the ICL shows that we only need to worry about the atomic choices that are needed to prove a goal (in our case the utility and the observations).

This should be seen in the context of an overriding theme of this work. We are not trying to see how much we can put into a formalism, but how little we can get away with. We don't need conditions on the probabilities of the atomic choices, so we don't have them.

> *Finally, you often talk about actions being "attempted", as in the above quotation. I found this particularly confusing since nothing in your ontology deals with action "attempts". Actions simply realize their effects with certain probabilities. You do not, for example, have an action try_pickup, as do other authors (e.g. James Allan) who do try to axiomatize action attempts. In this connection, I had a hard time making sense of footnote 9, since it motivates the axiom of Example 2.9 in terms of talk about "trying" to pick up the key. In fact, even after reading this material several times, I couldn't figure out footnote 9 until, much later, I began to think about how your approach compares with Bacchus, Halpern and Levesque. See my comments below on BHL.*

Perhaps the term "attempted" is misleading. What I mean is that the robot sends the motor control for the action, irrespectively of whether the robot can do it (in this sense it is attempting the action). By the pickup(key) action I mean a particular motor control (the one that would achieve carrying the key if it is at the same position as the key and the pickup succeeded). This action can be carried out in any context; it just doesn't achieve carrying the key in these contexts.

Section 3.3

> *I must confess that I couldn't make much sense of the discussion in Section 3.3, especially your arguments that the sitcalc is inherently unsuitable for modeling multiple, possibly reactive agents. It's true that Golog lacks the right control structures for doing this, but your comments on p. 26 and p. 27 (p. 26 "In order for a situation calculus program to ... and does the appropriate concurrent actions." p. 27 "When there are multiple agents ... or complex actions.") suggest to me that you are not aware of more recent work by De Giacomo, Lesperance and Levesque on Congolog that addresses precisely these issues (Proc. IJCAI 97). Congolog provides reactive rules (interrupts) and interleaving concurrency in a semantically clean way, all formulated within the sitcalc. I believe that Congolog addresses all your criticisms of the sitcalc, except, of course, it does not (yet) incorporate utilities.*

"It's true that Golog lacks the right control structures for doing this..." I am quite happy with the agent having simple control structures. I am concerned about modelling asynchronous external actions or events (by other agents or by nature).

It is not clear to me that Congolog addresses this, in the sense that all of the concurrency in the language is internal to the agent. It doesn't model exogenous actions (they generated externally). This would seem to mean that external events are restricted to occur at the situations defined by the primitive events. [I am looking forward to being told I am wrong.]

Moreover, with respect to the mix of the situation calculus and time, it is clear now that we can do what we want when we have both the situation calculus and time. But when you have statements saying when an event occurred, you don't need situations. Thus parsimony would suggest that we do away with situations. [I would expect that this should form a different thread.]

Comparison with Bacchus, Halpern and Levesque

So far as I know, there have only been two proposals for augmenting the sitcalc with probabilities, yours and BHL. As I indicate below, these two approaches are considerably different, and therefore I believe it is important to elaborate on these differences more than you do (top p. 25) especially the substantial ontological differences.

Yes, I do need to discuss this further (that is the joy of ETAI discussion of papers; we can retrospectively expand those parts that are of most interest).

First let me say that I am not claiming that mine is better than theirs, nor that it is different just for the sake of being different. We start from very different perspectives. In their introduction, they explicitly contrast their work with the work starting from Bayesian networks. In particular, they claim "...they [Bayesian networks] have difficulties in dealing with features like disjunction ...". I take a Bayesian perspective that disjunction is not a feature we want! What defines a Bayesian is that probability is a measure of belief, that any proposition can have a probability (both of which BHL would agree with) and that all uncertainty should be measured by probability. Thus you need to keep in mind that the different design choices could stem from this different perspective.

For BHL, there is no action_succeeds and action_fails fluents. Instead, an action whose outcome depend on nature is represented by the nondeterministic choice of simpler, deterministic actions corresponding to each different outcome. For example, the nondeterministic action flipaCoin is represented as the complex action

$$flipaCoin = flipHeads | flipTails$$

where | is GOLOG's operator for nondeterministically choosing between the two actions flipHeads and flipTails. For your ongoing example, pickup(key) would be represented by

$$pickup(key) = pickup\_succeeds(key) | pickup\_fails(key)$$

When an agent executes the action pickup(key) in situation S, nature steps in and selects which of the two actions pickup_succeeds(key) and pickup_fails(key) actually occurs, and it does so with frequencies determined by an axiom of the form:

$$P0(pickup\_succeeds(key), S) = p \leftrightarrow (suitable\ conditions\ on\ S\ and\ p)$$

In other words, your atomic choice of bottom p 13, and your axiom for P0 top of p 14 are represented by BHL in the way I have indicated above.

Except that BHL don't model probabilistic actions (in their IJCAI-95 paper, which is the only one I had seen) although I believe they could. All of their probabilities are within the agent, so such axioms would have to be part of the formula for updating belief. [I just found a new paper, Reasoning about Noisy Sensors and Effectors in the Situation Calculus, 1998, at Faheim Bacchus's web site that does this. I'll call this BHL98.]

One other important point to note is that the probabilistic part of the ICL is a restricted form of Bacchus's and Halpern's logics of probability as belief. The translation is that each atom that isn't an atomic choice is defined by Clark's completion of the rules defining that atom. We need statements that the elements of an alternative are exclusive and covering and that the atomic choices that aren't in the same alternative are probabilistically independent. Thus the translation doesn't require "suitable conditions on S and p".

A general framework for probability and action is intuitively straightforward (see Halpern and Tuttle (1993) for a general semantic framework). You can think about forward simulating the system. Nondeterministic/stochastic actions split the worlds and impose a probability over the resulting worlds. In the ICL we handle splitting the worlds using one simple mechanism: independent choice alternatives. I treat actions by the agent and other stochastic mechanisms (e.g., the ramifications of actions) in exactly the same way. BHL-98 split the worlds by nondeterministic actions (as you outline). They would then need a different mechanism for stochastic ramifications.

These are very different ontological and representational commitments than yours, and I think they deserve a deeper analysis than your discussion provides. Here are a few important differences these commitments lead to:

1. Primitive actions for BHL are entirely different than yours. BHL primitive actions would be things like $pickup\_succeeds(key)$, $flipHeads$ etc, i.e. they consist of the deterministic actions that nature can perform, together with those deterministic actions (if any) that the agent can perform. For you, primitive actions are typically nondeterministic, and correspond to those actions that the agent can perform (but with nondeterministic outcomes chosen by nature), e.g. $pickup(key)$, $flipaCoin$.

Yes.

2. Therefore, for BHL, situation terms describe the "actual" history selected by nature in response to the agent performed actions, e.g.

$$do(pickup\_fails(key), do(flipHeads, S0)).$$

For you, situations denote the history of agent performed actions, e.g.

$$do(pickup(key), do(flipaCoin, S0)).$$

The BHL axioms all refer to the former situations. In your ontology, only the latter are allowed.

Yes. I tried to be explicit about both of these points.

3. In BHL, all primitive actions ($pickup\_succeeds(key)$, $flipHeads$) are deterministic and successor state axioms (the causal laws + solution to the frame problem) are formulated only wrt these. For you, the causal axioms are formulated wrt the (nondeterministic) agent's actions ($pickup(key)$, $flipaCoin$); hence the need for your fluents $pickup\_succeeds(S)$ and $pickup\_fails(S)$. These function as random variables that determine the effects of the nondeterministic action $pickup(key)$. One consequence of this difference is that for BHL, successor state axioms are perfectly straightforward, whereas for you, because of these random variables, the causal laws become, at least conceptually, somewhat opaque (e.g. footnote 9).

Except I would think that my successor state axioms are perfectly straightforward. I get by with just Clark's completion, in much the same way as other logic programming representations of action. Frame and ramification axioms are treated the same (in fact I don't even think of them as different, and the paper doesn't distinguish them). I get by with Clark's completion because it is defined with respect to each world, where I just have a simple acyclic logic program. Maybe the guide to the perplexed below will help.

> *Moreover, BHL use "classical" action precondition axioms, in contrast to you (see above about action preconditions).*

So what happens if the agent doesn't know if the precondition holds, but still wants to do the action? This is important because, with noisy sensors, agents know the actual truth values of virtually nothing outside of their internal state, and most preconditions of actions are properties of the world, not properties of the agent's beliefs.

> *4. For BHL, your nondeterministic agent actions like pickup(key), are not part of the language of the sitcalc. Rather, they are nondeterministic GOLOG programs. Since you also introduce a notion of a program (Sections 2.8, 2.10), there are some natural comparisons to be made. The obvious (and perhaps only) important difference is in the nature of the primitive program actions. For BHL programs, the primitives are the deterministic actions described above (e.g. pickup_succeeds(key), flipHeads) whereas for you, they are the agent actions like pickup(key).*

One major difference is that in BHL, it is the agent who does probabilistic reasoning. In my framework, the agent doesn't (have to) do probabilistic reasoning, it only has to do the right thing (to borrow the title from Russell and Wefald's book). The role of the utility is to compare agents. This is important when we consider what Good calls type 2 rationality and Russell calls bounded rationality (see the work by I.J. Good, Eric Horvitz, and Stuart Russell for example), where we must take into account the time of the computation done by the agent in order to compute utility. I would expect that optimal agents wouldn't do (exact) probabilistic reasoning at all, because it is too hard! That is why we want a language that lets us model agents and their environment, and defines the expected utility of an agent in an environment. One of the reasons I didn't want to just add "time" into the framework is that I want a way to take into account the computation time of the agent (the thinking time as well as the acting time), and that makes it much more complicated (I wanted to get the foundations debugged first).

Suggestion: A Guide for the Perplexed

> *Your paper uses a single, ongoing example to illustrate the axiomatization style required by your approach. Personally, I couldn't abstract, from the example, a set of guidelines by which I could invent my own axiomatization for a different example of my choosing. I think it would considerably enhance the paper if you provided such guidelines. For example, for each action, tell the reader what axioms need to be specified. What choice of fluents needs to be made? (For example, there*

seem to be at least two categories of fluents – "normal" fluents like $carrying(X,S)$, and atomic choice fluents. How does one decide on the latter, when given the primitive actions and "normal" fluents of an application domain?) Similarly, for each fluent, what axioms need to be specified? Also, what needs to be axiomatized about the initial situation? Finally, what do all these axioms look like syntactically?

Thanks for the suggestion. The following is a paraphrase of how we tell people to represent knowledge in Bayesian networks, translated into the language of the ICL.

First I'll do the propositional (ground) case. We totally order the propositions. The idea is that we will define each proposition in terms of its predecessors in the total ordering. For each proposition e, a parent context is a conjunction of literals made up of the predecessors of e, such that the other predecessors are conditionally independent of e given the context. We find a set of mutually exclusive and covering set $\{context_1, ..., context_k\}$ of parent contexts. (The atoms appearing in one of the parent contexts are the parents of e in the corresponding Bayesian network). If e is always true when $context_i$ is true, we write the rule:

$$e \leftarrow context_i$$

If $P(e \mid context_i)$ isn't 0 or 1, we create a rule:

$$e \leftarrow context_i \wedge ac_i$$

and create an alternative $\{aci, naci\}$ (where these are both atoms that don't appear anywhere else), with the probability:

$$P0(ac_i) = P(e \mid context_i)$$

If the context is empty (e doesn't depend on any of its predecessors) we don't need to create a rule; we can just make e an atomic choice. When the probability given the context is 0, we don't write any rules.

The case for the ICL with the situation calculus is similar. Intuitively, we make the total ordering of the propositions respect the temporal ordering of situations. We write how a fluent at one situation depends on fluents (lower in the ordering) at that situation and on fluents at previous situations. Note that this rule automatically handles ramifications as well as frame axioms. The total ordering of the fluents guarantees the acyclicity of the rule base and that we don't have circular definitions. The initial situation is handled as any other, but the predecessors in the total ordering are only initial values of fluents (and perhaps atoms that don't depend on the situation).

The only thing peculiar about this is that we often have fluents that depend on values at the current as well as the previous situation. This is important when there are correlated effects of an action; we can't just define each fluent in terms of fluents at the previous situation. But this also means that we don't treat frame axioms and ramification axioms differently.

Let's look at how we would do Examples 2.8 & 2.9, with putting-down the key possibly failing. Lets consider when carrying would be true. Suppose carrying doesn't depend on anything at the same time, so we can ask what are the contexts on which $carrying(O, do(A, S))$ depends. It would seem there are three such contexts:

$A = pickup(O)$, and the robot and O are at the same position.

The action isn't $pickup(O)$, the robot was carrying O in situation S, and the action wasn't to put down O.

The robot was carrying O in situation S, and the action was to put down O.

In no other cases would the robot be carrying O.

The robot would be carrying O in the first context if the pickup succeeds (hence the name of the atomic choice in Example 2.8). This results in the rule:

$$\begin{aligned}&carrying(O, do(pickup(O), S)) \leftarrow\\&at(robot, Pos, S) \wedge\\&at(O, Pos, S) \wedge\\&pickup\_succeeds(O, S)\end{aligned}$$

The robot would be carrying O in the second context if it keeps carrying the key (hence the name of the atomic choice in Example 2.9). This results in the rule:

$$\begin{aligned}&carrying(O, do(A, S)) \leftarrow\\&carrying(O, S) \wedge\\&A \ = putdown(O) \wedge\\&A \ = pickup(O) \wedge\\&pickup\_succeeds(O, S)\end{aligned}$$

The robot would be carrying O in the third context if the putdown failed. Thus, to account for the chance that the putdown failed, we would write the extra clause:

$$\begin{aligned}&carrying(O, do(putdown(O), S)) \leftarrow\\&carrying(O, S) \wedge\\&putdown\_fails(O, S)\end{aligned}$$

where $putdown\_fails(O, S)$ is an atomic choice, with $P0(putdown\_fails(O, S))$ the conditional probability that the robot is still carrying O after it has carries out the $putdown(O)$ action.

Note that putdown may succeed in doing other things, it just fails in stopping the robot from carrying O. So maybe $putdown\_fails$ should be something more like $putdown\_fails\_to\_stop\_the\_robot\_from\_carrying$.

Footnote 9 is there to try to explain what the rules mean if the rule bodies aren't disjoint. We can interpret what these rules mean, but it usually isn't what you want (for the cases where both rules are applicable). When we write these rules, having non-disjoint rules is usually a bug (the implementation available from my web page warns when it finds non-disjoint rules.)

In connection with this last question, notice that on p. 20, the clause with head $at(X, P, S)$ resembles neither an effect axiom (as in Example 2.8) nor a frame axiom (as in Example 2.9).

It is a ramification of the robot moving. The above methodology handles these ramifications in the same way it handles frame axioms (and any other axioms).

Thanks for your comments and questions. I hope this made the paper clearer. I would be happy to answer any other questions you may have (or engage in debate about the usefulness of this).

David

This newsletter is issued whenever there is new news, and is sent by automatic E-mail and without charge to a list of subscribers, in plaintext format. The present postscript issue is a bona fide rendering of the same information in alternative typography. Instruction for contributors and other additional information is found at the following URL:

`http://www.ida.liu.se/ext/etai/actions/njl/`