$\mathcal{ESP}$: A Logic of Only-Knowing, Noisy Sensing and Acting

Alfredo Gabaldon
National ICT Australia
University of New South Wales
Sydney, NSW 1466
Australia
alfredo@cse.unsw.edu.au

Gerhard Lakemeyer
Dept. of Computer Science
RWTH Aachen
52056 Aachen
Germany
gerhard@cs.rwth-aachen.de

Abstract

When reasoning about actions and sensors in realistic domains, the ability to cope with uncertainty often plays an essential role. Among the approaches dealing with uncertainty, the one by Bacchus, Halpern and Levesque, which uses the situation calculus, is perhaps the most expressive. However, there are still some open issues. For example, it remains unclear what an agent's knowledge base would actually look like. The formalism also requires second-order logic to represent uncertain beliefs, yet a first-order representation clearly seems preferable. In this paper we show how these issues can be addressed by incorporating noisy sensors and actions into an existing logic of only-knowing.

Introduction

When reasoning about actions in realistic domains, the ability to cope with uncertainty often plays an essential role. This is true, for example, in cognitive robotics (Levesque and Lakemeyer 2007), where one is interested in giving a logic-based account of the actuators and sensors of robots.

There have been a number of approaches to address actions and uncertainty. For example, there are planners that deal with uncertainty like (N. Kushmerick et al. 1995; Weld et al. 1998) and there is the whole area of Markov Decision Processes (L. P. Kaelbling et al 1998). In order to fully capture the interplay between knowledge, action and uncertainty, more expressive languages are needed. In this regard, the situation calculus (McCarthy and Hayes 1969; Reiter 2001) has proven to be a very useful tool (Poole 1998; Bacchus et al. 1999; Boutilier et al. 2000; Thielscher 2001; Grosskreutz and Lakemeyer 2003). The starting point for our own investigations is Bacchus, Halpern and Levesque (1999) (BHL). As we will see, they have a compelling story about how to update beliefs when actions are uncertain, but there are also some problems with their framework which we will try to address in this paper.

To begin with, let us consider the following running example of a simple robot which is able to move towards the wall and which can use its sonar to measure the distance to the wall (Figure 1). All actions are noisy. In particular, suppose that the robot's sonar returns the true distance to the wall with probability .5 and and with probability .25 it is off by 1 unit in both directions. Also assume that initially the robot is 6 units away from the wall, but it is uncertain about its own position and believes that it might be 6, 5, or 4 units away, each with probability $\frac{1}{3}$.

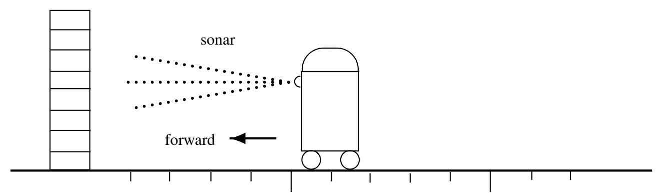

Figure 1: A simple robot

BHL model this using the situation calculus as follows: S_0 denotes the initial situation and a fluent $wd(s)$ represents the distance to the wall in each situation. For example, $wd(S_0) = 6$ says that initially the true wall distance is 6. To represent what the robot knows a fluent $K(s', s)$ is introduced, which captures, in possible-world fashion, the epistemically possible situations. In our example, there could be three such situations with $wd(s_1) = 6$, $wd(s_2) = 5$, and $wd(s_3) = 4$. If we have $K(s, S_0)$ iff s is one of the s_i, then the robot knows that wd is one of 4, 5, or 6 but does not know which. Here knowledge is simply truth in all accessible situations, abbreviated as $Knows(wd, S_0)$. To model uncertainty, BHL introduce another functional fluent $p(s', s)$ which is similar to K but also assigns a probability to the accessible situations s'. To indicate that the above s_i all have probability $\frac{1}{3}$, we would write $p(s_i, S_0) = \frac{1}{3}$. With p in place BHL define probabilistic belief in a sentence ϕ in S_0 (written $Bel(\phi, S_0)$) simply as the sum of the probabilities of those accessible situations where ϕ holds. For example, we get $Bel((wd = 6 \vee wd = 5), S_0) = \frac{2}{3}$.

A model of the noisy sonar can be obtained as follows. Imagine that the robot executes a noisy sensing action $nobs(x)$, which returns a value x. The action which is actually executed (chosen by nature) is $obs(x, y)$, where y is the actual value of wd. The uncertainty about which outcome is chosen by nature is measured by a probability (likelihood) l assigned to $obs(x, y)$. In our example, we would have $l(obs(x,x)) = .5$, $l(obs(x,x-1)) = .25$, $l(obs(x,x+1)) = .25$, and 0 otherwise. With this description, it follows that the belief that $wd = 6$ after executing $nobs(6)$ has probability

$$\frac{p(s_1, S_0) \cdot l(obs(6,6))}{\sum_{s'} p(s', S_0) \cdot l(obs(6, wd(s')))} = \frac{2}{3},$$

that is, after receiving the value 6 from the sonar, the belief that $wd = 6$ is sharpened, as it should be.[1] In a similar way it can be established that executing a noisy forward action would increase the robot's uncertainty about *wd*.

The way belief is updated in the BHL framework seems right, and it conforms to practice in robotics. Nevertheless there are at least two shortcomings which we would like to address in this paper. For one, under seemingly innocuous assumptions like the K-relation being transitive and Euclidean, and $p(s, s_1) = p(s, s_2)$ for all s, s_1, s_2, it follows under BHL that $\exists x. Knows(Bel(\phi) = x, S_0)$, suggesting that agents necessarily have *de re* knowledge about their degrees of belief. This is so because in each model of such BHL-theories there is exactly one probability distribution over situations. For another, there is also the technical problem that $Bel(\phi, s)$ is not a primitive of the language but defined in terms of equations like the above. In particular, representing summation requires second-order logic, and it seems like a heavy price to pay if a robot needs second-order sentences in its knowledge base.

Our solution to these problems is to reconstruct the BHL way of updating probabilistic belief in a recently proposed variant of the situation calculus (Lakemeyer and Levesque 2005), where situation terms are banned from the language. Instead they are used only as part of the possible-world semantics of the logic. To address the problem about what is known about probabilities, we will allow the robot to entertain many probability distributions over the worlds it considers possible, and these distributions may differ from an actual (objective) distribution over the set of all worlds. We will have summations over products of probabilities, but in contrast to BHL, these are pushed into the semantics. The language will have sentences of the form $HasP(\phi, p)$ as primitives to denote that ϕ has probability p. With that we are able to restrict knowledge bases to first-order sentences. Moreover, as the new logic supports the concept of only-knowing, we can precisely model what a robot knows and does not know given its knowledge base, something we cannot do in the BHL framework.

The rest of the paper is organized as follows. In the next section we briefly review the logic $\mathcal{ES}$. We then discuss the necessary changes to incorporate uncertainty into the logic and discuss some of the properties of the new logic. To illustrate the formalism, we then apply the logic to our robot example and end the paper with a brief conclusion.

The Logic $\mathcal{ES}$

The language of $\mathcal{ES}$ consists of formulas over symbols from the following vocabulary:

- first-order variables: $x_1, x_2, \ldots, y_1, y_2, \ldots, a_1, a_2, \ldots$;
- standard names: $n_1, n_2, \ldots$;
- fluent function symbols, rigid function symbols, fluent predicate symbols, and rigid predicate symbols of every arity;
- connectives and other symbols: $=$, $\wedge$, $\neg$, $\forall$, *Know*, *OKnow*, $\Box$, round and square parentheses, period, comma.

We assume that first-order variables, standard names, and function symbols come in two sorts, *action* and *object*. Constants are function symbols of 0 arity.[2] We let $\mathcal{N}$ denote the set of all standard names and $\mathcal{Z}$ denote the set of all sequences of standard names for actions, including $\langle\,\rangle$, the empty sequence. For sequences z and z', we let $z \cdot z'$ denote their concatenation.

Terms and formulas

The *terms* of the language are of sort *action* or *object*, and form the least set of expressions such that

1. Every standard name and first-order variable is a term of the corresponding sort;
2. If $t_1, \ldots, t_k$ are terms and h is a k-ary function symbol then $h(t_1, \ldots, t_k)$ is a term of the same sort as h.

By a *primitive term* we mean one of the form $h(n_1, \ldots, n_k)$ where h is a (fluent or rigid) function symbol and all of the n_i are standard names.

The *well-formed formulas* of the language form the least set such that

1. If $t_1, \ldots, t_k$ are terms, and H is a k-ary predicate symbol then $H(t_1, \ldots, t_k)$ is an (atomic) formula;
2. If t_1 and t_2 are terms, then $(t_1 = t_2)$ is a formula;
3. If t is an action term and α is a formula, then $[t]\alpha$ is a formula;
4. If α and β are formulas, and v is a first-order variable, then the following are also formulas: $(\alpha \wedge \beta)$, $\neg\alpha$, $\forall v.\, \alpha$, $\Box\alpha$, $Know(\alpha)$, $OKnow(\alpha)$.

We read $[t]\alpha$ as "α holds after action t", and $\Box\alpha$ as "α holds after any sequence of actions." As usual, we treat $(\alpha \vee \beta)$, $(\alpha \supset \beta)$, $(\alpha \equiv \beta)$, and $\exists v.\, \alpha$, as abbreviations. We call a formula without free variables a *sentence*. By a *primitive sentence* we mean a formula of the form $H(n_1, \ldots, n_k)$ where H is a (fluent or rigid) predicate symbol and all of the n_i are standard names.

The semantics

The language contains both fluent and rigid expressions. The former vary as the result of actions and have values that may be unknown, but the latter do not. Intuitively, to determine whether or not a sentence α is true after a sequence of actions z has been performed, we need to specify two things:

[1] See BHL for a discussion of how this is just a situation-calculus version of Bayesian conditioning.

[2] The standard names can be thought of as special extra constants that satisfy the unique name assumption and an infinitary version of domain closure.

a world w and an epistemic state e. We write $e, w, z \models \alpha$. A world determines truth values for the primitive sentences and co-referring standard names for the primitive terms after any sequence of actions. An epistemic state is defined by a set of worlds, as in possible-world semantics. More precisely:

- a world $w \in W$ is any function from the primitive sentences and $\mathcal{Z}$ to $\{0, 1\}$, and from the primitive terms and $\mathcal{Z}$ to $\mathcal{N}$ (preserving sorts), and satisfying the rigidity constraint: if r is a rigid function or predicate symbol, then $w[r(n_1, \ldots, n_k), z] = w[r(n_1, \ldots, n_k), z']$, for all z and z' in $\mathcal{Z}$.
- an epistemic state $e \subseteq W$ is any set of worlds.

We extend the idea of co-referring standard names to arbitrary ground terms as follows. Given a term t without variables, a world w, and an action sequence z, we define $|t|_w^z$ (read: the co-referring standard name for t given w and z) by:

1. If $t \in \mathcal{N}$, then $|t|_w^z = t$;
2. $|h(t_1, \ldots, t_k)|_w^z = w[h(n_1, \ldots, n_k), z]$, where $n_i = |t_i|_w^z$.

Truth is then defined as follows. Given $e \subseteq W$ and $w \in W$, we define $e, w \models \alpha$ (read: α is true) as $e, w, \langle\,\rangle \models \alpha$, where for any $z \in \mathcal{Z}$:

1. $e, w, z \models H(t_1, \ldots, t_k)$ iff $w[H(n_1, \ldots, n_k), z] = 1$, where $n_i = |t_i|_w^z$;
2. $e, w, z \models (t_1 = t_2)$ iff n_1 and n_2 are identical, where $n_i = |t_i|_w^z$;
3. $e, w, z \models [t]\alpha$ iff $e, w, z \cdot n \models \alpha$, where $n = |t|_w^z$;
4. $e, w, z \models (\alpha \wedge \beta)$ iff $e, w, z \models \alpha$ and $e, w, z \models \beta$;
5. $e, w, z \models \neg\alpha$ iff $e, w, z \not\models \alpha$;
6. $e, w, z \models \forall v.\, \alpha$ iff $e, w, z \models \alpha^v_n$, for every standard name n (of the same sort as v);
7. $e, w, z \models \Box\alpha$ iff $e, w, z \cdot z' \models \alpha$, for every $z' \in \mathcal{Z}$;
8. $e, w, z \models \mathit{Know}(\alpha)$ iff for every $w' \in e$, $e, w', z \models \alpha$;
9. $e, w, z \models \mathit{OKnow}(\alpha)$ iff for every w', $w' \in e$ iff $e, w', z \models \alpha$.[3]

When Σ is a set of sentences and α is a sentence, we write $\Sigma \models \alpha$ (read: Σ logically entails α) to mean that for every e and w, if $e, w \models \alpha'$ for every $\alpha' \in \Sigma$, then $e, w \models \alpha$. Finally, we write $\models \alpha$ (read: α is valid) to mean $\{\} \models \alpha$.

$\mathcal{ESP}$ = $\mathcal{ES}$ + uncertainty

The uncertainty we are adding to $\mathcal{ES}$ comes in two flavors. The first concerns the uncertainty about what is true initially. In particular, this will mean that the agent may believe that a sentence is true with some probability before any actions have occurred. The other concerns the uncertainty in the outcome of performing an action or sensing the value of a fluent.

[3] We remark that the original definition of knowledge in (Lakemeyer and Levesque 2005) was somewhat more complicated due to their treatment of (nonstochastic) sensing, which we ignore here.

Uncertainty about the initial situation

To start with, as we want to talk about probabilities, we need to include numbers in the language and the semantics. Normally, the reals are used for this purpose, but since we limit ourselves to a countably infinite domain, we use the rationals, which suffice for most practical purposes. Formally, we include the rationals as a new subsort of $\mathcal{N}_O$ and call it $\mathcal{N}_Q$, that is, there is a standard name for each rational number.

Recall that in BHL, uncertainty in the initial situation is captured by assigning probabilities to situations. It seems natural to try something similar in our semantics and assign probabilities to worlds. Unfortunately, there is a catch: there are an *uncountable* number of worlds, which would mean that any individual world has probability 0. Moreover, it would be nice to be able to restrict ourselves to discrete probabilities to simplify matters. Since for now we are only interested in what is true initially, distinguishing between each world is actually not necessary, as many worlds assign identical values to the primitive terms and formulas. So we could consider assigning probabilities to sets of worlds where the worlds in each set agree initially. But that is still not enough, since these sets are still uncountable because the universe is infinite. To address this issue, we assume that, as far as the agent is concerned, there are only a finite number of objects together with a finite number of predicate and functions symbols defined over these objects. Formally, we introduce a new sort $\mathcal{N}_U$ which is any finite subset of $\mathcal{N}_O$, and two finite sets of predicate symbols H_U, whose arguments are all of type $\mathcal{N}_U$, and a finite set of function symbols h_U, whose arguments and values are also of type $\mathcal{N}_U$. $\mathcal{N}_U$ together with H_U and h_U is called the agent signature $\mathcal{S}$.

With that we can now lump together worlds which agree on $\mathcal{S}$. In particular, for any world w, let

$$||w|| = \{w' \mid w'[h, \langle\,\rangle] = w[h, \langle\,\rangle] \text{ for all } h \in \mathcal{S}_P\},$$

where $\mathcal{S}_P$ is the set of all primitive formulas and terms mentioning only symbols from $\mathcal{S}$. Let $\mathcal{B} = \{||w|| \mid w \in W\}$. Clearly, $||w||$ defines an equivalence relation over W, and there are only finitely many equivalence classes, that is, $\mathcal{B}$ is finite. In our semantics we will consider probability distributions over $\mathcal{B}$ as a means to capture uncertainty about what is true initially.—Note that only domain-specific predicates and functions are restricted to be defined over the finite sort $\mathcal{N}_U$. We still allow infinitely many actions, and the set of values probabilities range over is also infinite.

Noisy actions and sensors

When adding uncertainty to actions we follow the ideas of BHL (see also (Reiter 2001)) and introduce *stochastic actions* which are associated with a number of outcomes (nature's choices) and a probability distribution over them. In the robot example, *forward* would become a stochastic action with three possible outcomes *adv*(0), *adv*(1), and *adv*(2), indicating that the robot initiates a forward action, but in fact he will either not move at all, correctly move one unit forward or overshoot by 1 unit. Each of the outcomes would happen with a certain probability, say, .25 for *adv*(0),

.5 for *adv*(1), and .25 for *adv*(2). Noisy sensors are modelled exactly the same way (see later when we return to the robot example). To simplify the presentation we assume, from now on, that all actions are stochastic. It is easy to see that this is no restriction as we can model ordinary actions as stochastic ones which have only a single choice.

The language

Here we summarize the changes and additions of the language when moving from $\mathcal{ES}$ to $\mathcal{ESP}$.

- In addition to the sorts $\mathcal{N}_Q$ and $\mathcal{N}_U$ for the rationals and the objects of the signature $\mathcal{S}$, we also introduce another sort $\mathcal{N}_S$, which contains a countably infinite set of standard names for stochastic actions. We continue to use $\mathcal{N}$ to refer to the set of all standard names.[4] We assume that the language has an appropriate number of function symbols for every sort. Action function symbols are rigid, those of type $\mathcal{N}_O$ can be rigid or fluent.
- We add a predicate *Choice*(a, n) which will be true just in case n is one of the choices of stochastic action a.
- We add a binary function symbol *prob* such that *prob*(a, n) is the probability of choice n of stochastic action a in the current situation.
- We introduce a new kind of modality *HasP*(ϕ, p), where ϕ is any static objective sentence[5] over the signature $\mathcal{S}$ and p is a rational number. It may be read as "ϕ has probability p." Note that *HasP* is not a predicate but a modal operator, mainly for practical reasons so that we do not need to reify formulas.
- In $\mathcal{ES}$ a formula $[n]\alpha$ means that α holds after n is performed. Since we are now dealing with actions as pairs we change this to $[(a, n)]\alpha$, which can be read as "after performing the choice n of stochastic action a α holds."
- We also include a new kind of modal operator $[[a]]$, where a is a stochastic action. $[[a]]\alpha$ is intended to mean that α holds after doing a regardless of which choice actually occurs.
- Formulas are formed in the usual way with the following restrictions, which help in keeping the semantics simpler:
  - for any $[(t, t')]\alpha$ and $[[t]]\alpha$, all function symbols occurring in t and t' are rigid;
  - for any $[[t]]\alpha$, all predicate and function symbols mentioned in α occur within a *HasP*(ϕ, p) expression. In other words, after performing a stochastic action we are only interested in statements about what is true or known about the probability of the outcomes.

The semantics

For the semantics we start by introducing a function $\mathcal{C} : \mathcal{N}_S \longrightarrow 2^{\mathcal{N}_A}$, which associates with each stochastic action a nonempty (possibly infinite) set of choices taken from the set of (ordinary) actions.

[4]Note that $\mathcal{N}_U$ is the only finite sort of the language.

[5]A static and objective sentence is one which does not mention modal operators of any kind.

Since actions now come in pairs, we change $\mathcal{Z}$ to mean the set of sequences of pairs (a, n), where a is a standard name of a stochastic action and n is either a standard name of an ordinary action or the wild card $*$ to indicate that the choice is left unspecified. Let $\mathcal{Z}_g$ be the subset of ground sequences of $\mathcal{Z}$, that is those which do not mention $*$.

Next we define worlds to give meaning to primitive terms and formulas as in $\mathcal{ES}$ except that we now use the set of ground sequences of action pairs from $\mathcal{Z}_g$ as the second argument. Note that this has no bearing on our definition of $\mathcal{B}$ above, where we only need the empty sequence of actions.

Since $\mathcal{B}$ is a finite set, we can easily define probability distributions μ over it. As we already mentioned, such μ assign weights to the various ways the world might look like initially.

To define the likelihood of action choices we make use of functions $\lambda : \mathcal{B} \times \mathcal{Z}_g \times \mathcal{N}_S \times \mathcal{N}_A \longrightarrow \mathbb{Q}$, where $\lambda(b, z, a, n) = p$ associates with each choice n of stochastic action a after any number of actions z starting in b a likelihood p; we assume that for each b, z, and a, λ is a probability distribution over the choices n. (For any n which is not a choice of a, λ is assumed to be 0.) Let Λ be the set of all such functions.

An epistemic state is then characterized by a triple $\epsilon = (e, m, l)$, where e is a set of worlds as before, $l \subseteq \Lambda$, and m is a set of probability distributions over $\mathcal{B}$ with the restriction that for each $\mu \in m$ and $w \notin e$, $\mu(||w||) = 0$. In other words, as far as the agent is concerned all worlds which are not considered possible are also improbable. Let $M_e = \{\mu \mid \mu(||w||) = 0 \text{ for all } w \notin e\}$

For the meaning of terms we can simply lift the definition of $|t|_w^z$ when $z \in \mathcal{Z}_g$. For z which are not ground we can at least give meaning to terms t all of whose function symbols are rigid by letting $|t|_w^z = |t|_w^{\langle\rangle}$. Since *prob* does not receive its meaning from worlds, we need to treat it specially: for a given $\lambda \in \Lambda$, $|prob(a, n)|_w^z = \lambda(||w||, z, a, n)$ where *prob*(a, n) is primitive and $z \in \mathcal{Z}_g$. (It will always be clear from the context which λ is meant.)

Given a function $\mathcal{C}$ as above, an epistemic state $\epsilon = (e, m, l)$, a world w, a probability distribution μ, and $\lambda \in \Lambda$, we define $\epsilon, w, \mu, \lambda \models \alpha$ (read: α is true) as $\epsilon, w, \mu, \lambda, \langle\rangle \models \alpha$, where $\epsilon, w, \mu, \lambda, z \models \alpha$ is inductively defined as follows:

For $z \in \mathcal{Z}_g$ we have:

1. $\epsilon, w, \mu, \lambda, z \models Choice(t, t')$ iff $|t'|_w^z \in \mathcal{C}(a)$ where $a = |t|_w^z$;
2. $\epsilon, w, \mu, \lambda, z \models H(t_1, \ldots, t_k)$ iff $w[H(n_1, \ldots, n_k), z] = 1$, where $n_i = |t_i|_w^z$;
3. $\epsilon, w, \mu, \lambda, z \models (t_1 = t_2)$ iff n_1 and n_2 are identical, where $n_i = |t_i|_w^z$.

For arbitrary $z \in \mathcal{Z}$ (including nonground z) we have:

4. $\epsilon, w, \mu, \lambda, z \models (\alpha \wedge \beta)$ iff $\epsilon, w, \mu, \lambda, z \models \alpha$ and $\epsilon, w, \mu, \lambda, z \models \beta$;
5. $\epsilon, w, \mu, \lambda, z \models \neg\alpha$ iff $\epsilon, w, \mu, \lambda, z \not\models \alpha$;
6. $\epsilon, w, \mu, \lambda, z \models \forall v. \alpha$ iff $\epsilon, w, \mu, \lambda, z \models \alpha_n^v$, for every standard name n (of the same sort as v);

7. $\epsilon, w, \mu, \lambda, z \models [(t, t')]\alpha$ iff $\epsilon, w, \mu, \lambda, z \cdot (a, n) \models \alpha$, where $a = |t|^z_w$ and $n = |t'|^z_w$;
8. $\epsilon, w, \mu, \lambda, z \models [[t]]\alpha$ iff $\epsilon, w, \mu, \lambda, z \cdot (a, *) \models \alpha$, where $a = |t|^z_w$;
9. $\epsilon, w, \mu, \lambda, z \models \Box\alpha$ iff $\epsilon, w, \mu, \lambda, z \cdot z' \models \alpha$, for every $z' \in \mathcal{Z}_g$;
10. $\epsilon, w, \mu, \lambda, z \models \mathit{Know}(\alpha)$ iff for every $w' \in e$, $\mu' \in m$, and $\lambda' \in l$, $\epsilon, w', \mu', \lambda', z \models \alpha$;
11. $\epsilon, w, \mu, \lambda, z \models \mathit{OKnow}(\alpha)$ iff for every $w', \mu' \in M_e, \lambda'$, $w' \in e, \mu' \in m, \lambda' \in l$ iff $\epsilon, w', \mu', \lambda', z \models \alpha$.

We remark that even though z must be ground for the cases (1)–(3), this gives us a complete specification of truth because of the restriction that predicate and function symbols following a $[[t]]$-operator must occur within a *HasP*-expression, whose semantics we now turn to.

In order to give meaning to sentences of the form $\mathit{HasP}(\phi, p)$, we need to introduce some abbreviations. For any $z \in \mathcal{Z}$ let $\mathit{gnd}(z)$ be the set of ground sequences, where each occurrence of $(a, *)$ is replaced by (a, n) for some $n \in \mathcal{C}(a)$. For any ground sequence z let $\mathcal{B}^z_\phi = \{b \in \mathcal{B} \mid \text{for all } w \in b,\ \epsilon, w, \mu, \lambda, z \models \phi\}$, that is $\mathcal{B}^z_\phi$ contains all those sets where ϕ is true everywhere after the actions in z have been performed. For any $z \in \mathcal{Z}_g$ with $z = (a_1, n_1) \cdot \ldots \cdot (a_k, n_k)$, let z_i be the prefix of the first i elements of z (with $z_0 = \langle\,\rangle$). Then

12. $\epsilon, w, \mu, \lambda, z \models \mathit{HasP}(\phi, p)$ iff

$$p = \frac{\sum_{\{z' \in \mathit{gnd}(z), b \in \mathcal{B}^{z'}_\phi\}} \mu(b) \cdot \prod_{i=0}^{k-1} \lambda(b, z'_i, a_{i+1}, n_{i+1})}{\sum_{\{z' \in \mathit{gnd}(z), b \in \mathcal{B}\}} \mu(b) \cdot \prod_{i=0}^{k-1} \lambda(b, z'_i, a_{i+1}, n_{i+1})},$$

provided the denominator is not 0, otherwise $p = 0$.

To better understand what is happening here, consider expanding the sequence z into a (situation) tree as follows: if $z = (a, n) \cdot z^+$ then add a node with an edge connected to the tree generated by z^+ and label the edge with (a, n); if $z = (a, *) \cdot z^+$ then add a node and for each (a, n) where n is one of the choices of a add an edge connected to a subtree generated by z^+ and label the edge with (a, n). In this tree each branch represents one of the possible executions of all the actions in z. One such tree is then associated with each $b \in \mathcal{B}$. Computing the probability of ϕ being true after z then amounts to computing the probability of each branch by multiplying the likelihoods of nature's choices along the edges of the branch, then summing up the probabilities of those branches where ϕ holds at the end (over all trees associated with the b's) and dividing over the sum of the probabilities of all branches, again over all trees. Note that the denominator is needed for normalization purposes.

Finally, for a fixed signature $\mathcal{S}$ we define logical implication in $\mathcal{ESP}$ for a set of sentences Σ and a sentence α as follows: $\Sigma \models \alpha$ iff for all $\mathcal{C}$, for all epistemic states ϵ, worlds w, probability distributions μ over $\mathcal{B}$, and $\lambda \in \Lambda$, if $\epsilon, w, \mu, \lambda \models \sigma$ for all $\sigma \in \Sigma$ then $\epsilon, w, \mu, \lambda \models \alpha$. As usual, α is valid if $\{\} \models \alpha$.

Some properties

A thorough investigation of the semantics of $\mathcal{ESP}$ is out of the scope of this paper. Instead we will focus on some of the properties regarding the relationship between uncertainty and knowledge.

1. $\models \mathit{Know}(\phi) \supset \mathit{Know}(\mathit{HasP}(\phi, 1))$;
2. $\models \mathit{Know}(\mathit{HasP}(\phi, p)) \supset \neg\mathit{Know}(\neg\phi)$ if $p > 0$;
3. $\not\models \mathit{HasP}(\phi, p) \supset \mathit{Know}(\mathit{HasP}(\phi, p))$;
4. $\not\models \mathit{Know}(\mathit{HasP}(\phi, p)) \supset \mathit{HasP}(\phi, p)$;
5. $\models \mathit{Know}(\exists x.\mathit{HasP}(\phi, x))$;
6. $\not\models \exists x.\mathit{Know}(\mathit{HasP}(\phi, x))$ where $\not\models \phi$.

Proof:

1. Recall that for every $\epsilon = (e, m, l)$ and $\mu \in m$ we have that $\mu(||w||) = 0$ for all $w \notin e$. Assuming that ϕ holds in all worlds of e, both the numerator and denominator in the definition of *HasP* are equal and hence the result is 1.
2. Let $\epsilon, w, \mu, \lambda \models \mathit{Know}(\mathit{HasP}(\phi, p))$ with $\epsilon = (e, m, l)$ and let $\mu \in m$. Then $\mu(||w'||) > 0$ for some $w \in e$ and $\epsilon, w, \mu, \lambda \models \phi$. Hence $\epsilon, w, \mu, \lambda \models \neg\mathit{Know}(\neg\phi)$.
3. The implication fails simply because if $\epsilon, w, \mu, \lambda \models \mathit{HasP}(\phi, p)$, then μ is not necessarily a member of m where $\epsilon = (e, m, l)$.
4. The reverse fails for the same reason.
5. This clearly holds because in any model, $\mathit{HasP}(\phi, p)$ is true for exactly one value of p.
6. The implication fails because for every $\epsilon = (e, m, l)$ each $\mu \in m$ may be different and hence the probability of ϕ may vary. ∎

We remark that it is because of properties (3) and (4) that we did not use BHL's notation $\mathit{Bel}(\phi, p)$, but opted for the more neutral $\mathit{HasP}(\phi, p)$. While they have a purely subjective view of probabilities (all distributions assign a probability of 0 to epistemically inaccessible situations), we take a more objective view in that we allow distributions which assign non-zero probabilities to $||w||$ where $w \notin e$, but these distributions are never considered by the agent.

The robot example revisited

Let us now model the robot example in $\mathcal{ESP}$. Recall that we have two stochastic actions, *forward* with choices *adv*(0), *adv*(1), and *adv*(2), and *nobs* with choices *obs*(x), where x ranges, say, between -1 and +20. The domain theory needs to specify these choices (Σ_{ch}), and the robot's knowledge base, consisting of a successor state axiom[6] for the only fluent *wd* (Σ_{post}), the likelihood of the action choices (Σ_{l}), and the robot's beliefs about the initial situation (Σ_0). We also need to assume that actions have unique names (Σ_{UNA}) and that the agent knows that. (Normally domain theories also include axioms about the executability of actions, an issue we ignore here for simplicity.)

Σ_{ch} consists of these axioms:[7]

[6]These were introduced by Reiter as a solution to the frame problem (Reiter 2001).

[7]In the following all free variables are implicitly universally quantified.

$Choice(forward, x) \equiv$
$(x = adv(0) \vee x = adv(1) \vee x = adv(2))$

$Choice(nobs, x) \equiv$
$\exists y. x = obs(y) \wedge (y = -1 \vee y = 0 \vee \ldots \vee y = 20)$

Σ_{post} has one axiom:

$$\Box[(a, x)]wd = y \equiv a = forward \wedge$$
$$[x = adv(1) \wedge wd = y + 1 \vee x = adv(2) \wedge wd = y + 2$$
$$\vee x \neq adv(1) \wedge x \neq adv(2) \wedge wd = y]$$
$$\vee a \neq forward \wedge wd = y$$

This axiom describes precisely how the value of *wd* changes or does not change after an action (a,x) has occurred. For example, in the case of $(forward, adv(1))$, *wd* will be 1 unit less than before the action.

Σ_{l} consists of these axioms:[8]

$$\Box prob(forward, adv(1)) = \textbf{if } wd > 0 \textbf{ then } .5 \textbf{ else } 0$$
$$\Box prob(forward, adv(0)) = \textbf{if } wd = 0 \textbf{ then } 1 \textbf{ else } .25$$
$$\Box prob(forward, adv(2)) = \textbf{if } wd > 1 \textbf{ then } .25 \textbf{ else } 0$$
$$\Box prob(nobs, obs(x)) =$$
$$\textbf{if } wd = x \textbf{ then } .5 \textbf{ else}$$
$$\textbf{if } wd = x + 1 \textbf{ then } .25 \textbf{ else}$$
$$\textbf{if } wd = x - 1 \textbf{ then } .25 \textbf{ else } 0$$

In other words, unless the robot is very close to the wall, the likelihood of advancing 1 unit is .5, whereas the likelihood of moving 2 units or not at all is .25. Observing the correct value of *wd* has likelihood .5 and being off by ± 1 has likelihood .25 each.

Σ_0 consists of these axioms:

$$HasP(wd = 6, \tfrac{1}{3}),\ HasP(wd = 5, \tfrac{1}{3}),\ HasP(wd = 4, \tfrac{1}{3})$$

These indicate that the robot finds it equally likely that *wd* is either 6, 5, or 4.

The domain theory is then defined as

$$\Sigma = \Sigma_{\text{UNA}} \wedge \Sigma_{\text{ch}} \wedge OKnow(\Sigma_{\text{UNA}} \wedge \Sigma_{\text{post}} \wedge \Sigma_{\text{l}} \wedge \Sigma_0)$$

(We assume that the signature $\mathcal{S}$ includes at least *wd* and all the numbers mentioned in Σ.)

The following sentences are logical consequences of Σ:

1. $\neg Know(wd = 6)$.
   Note that, when Σ is satisfied, the robot considers every world in W possible. Therefore it surely considers a world possible where $wd \neq 6$.
2. $Know(HasP(wd = 6, \frac{1}{3}))$.
   This is because $HasP(wd = 6, \frac{1}{3})$ is part of Σ_0.
3. $[[forward]]Know(HasP(wd = 5, \frac{1}{4}))$.
   After doing a noisy forward action the robot believes that $wd = 5$ with less confidence than before.
4. $[(nobs, obs(6))]Know(HasP(wd = 6, \frac{2}{3}))$.
   This is the way we model sensing: the agent performs a noisy sense action and receives a sensed value in return. Since the value is 6, the robot's belief that $wd = 6$ is now much stronger than before. The belief that $wd = 4$ is 0.
5. $[[forward]][(nobs, obs(5))]Know(HasP(wd = 5, \frac{11}{16}))$.
   While moving forward weakens the belief in $wd = 5$, observing a 5 afterwards strengthens it again.

[8] We use the abbreviation $f = \textbf{if } \phi \textbf{ then } p \textbf{ else } q$ to stand for $f = x \equiv \phi \wedge x = p \vee \neg\phi \wedge x = q$.

Conclusions

In this paper we reconstructed BHL's approach to noisy sensing and acting in a variant of the situation calculus. In contrast to BHL, an agent's epistemic state not only consists of a set of possible worlds but also of sets of probability distributions over both the initial situations and the outcome of stochastic actions after any number of actions. Also, no second-order logic is needed.

One of the questions we addressed is what only-knowing means in the presence of probabilities. We gave one answer, but there may be others. Indeed, one of the anonymous reviewers suggested to define only-knowing in terms of probabilities in the sense that an agent only-knows α iff the models of α are just the worlds with non-zero probability. While this may have interesting features, we conjecture that it does not have the following property, which holds in our case: only-knowing certain knowledge bases involving probabilities, including the one in the robot example, precisely determines what the agent knows and does not know, something we feel is very appealing from a KR point of view.

References

C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proc. of AAAI-00*, 355–362, 2000.

F. Bacchus, J.Y. Halpern, and H. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence 111(1-2)*, 1999.

H. Grosskreutz and G. Lakemeyer, Probabilistic Complex Actions in Golog. *Fundamenta Informaticae* 57(2–4), 167–192, 2003.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence 101(1-2*, 1998.

N. Kushmerick, S. Hanks, and D. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76:239–286, 1995.

G. Lakemeyer and H. J. Levesque, Semantics for a useful fragment of the situation calculus. IJCAI, 490–496, 2005.

H. J. Levesque and G. Lakemeyer, Cognitive Robotics, in F. van Harmelen, V. Lifshitz, B. Porter (Eds.) *The Handbook of Knowledge Representation*, Elsevier, to appear.

J. McCarthy and P. J. Hayes, Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer, D. Mitchie and M. Swann (Eds.) *Machine Intelligence 4,* Edinburgh University Press, 463–502, 1969.

David Poole. Decision theory, the situation calculus and conditional plans. *Linköping Electronic Articles in Computer and Information Science*, 3(8), 1998.

R. Reiter, *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press, 2001.

M. Thielscher, Planning with Noisy Actions. *Australian Joint Conference on Artificial Intelligence,* 495–506, 2001.

D S. Weld, C. R. Anderson and D. E. Smith, Extending Graphplan to Handle Uncertainty and Sensing Actions. *Proc. of AAAI*, 897–904, 1998.