OBSERVATIONS AND THE PROBABILISTIC SITUATION CALCULUS

Paulo Mateus[(1)], António Pacheco[(2)], Javier Pinto[(3)],

Abstract

In this article we propose a Probabilistic Situation Calculus logical language to represent and reason with knowledge about dynamical worlds in which actions have uncertain effects. Two essential tasks are addressed when reasoning about change in worlds: Probabilistic Temporal Projection and Probabilistic Belief Update. Uncertain effects are modeled by dividing an action into two subparts: a deterministic input (agent produced) and a probabilistic reaction (nature produced). The probability distributions of the reactions are assumed to be known.

Our logical language is an extension to Situation Calculae in the style proposed by Raymond Reiter. There are three aspects to this work. First, we extend the language to accommodate terms dealing with belief and probability. Second, we provide a operational semantics based on Randomly Timed Automata. Finally, we develop Monte-Carlo algorithms to efficiently interpret the probability and belief terms.

With the framework proposed we discuss how to develop a reasoning system in Mathematica capable of performing temporal projection and belief update in the Probabilistic Situation Calculus. Finally, we present a sound basis to set rewards and observation planning.

[(1)] Center for Logic and Computation, Departamento de Matemática, IST, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. **email:** `pmat@math.ist.utl.pt`. Supported by FCT SFRH/BPD/5625/2001 and the FibLog initiative.
[(2)] Applied Mathematics Center, Departamento de Matemática, IST, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. **email:** `apacheco@math.ist.utl.pt`
[(3)] Unfortunately J. Pinto passed away in an accident while this paper was being prepared. Formerly, he was at Bell Labs, Database Systems Research Dept., 600 Mountain Ave., New Jersey 07974, U.S.A.

OBSERVATIONS AND THE PROBABILISTIC SITUATION CALCULUS

Background and Objectives

Lately, there has been increasing interest in improving the expressive power of logical languages for representing knowledge about dynamic worlds with non-determinism and uncertainty (e.g., [9, 1, 10, 4, 2]). In this article, we extend previous work on the *Probabilistic Situation Calculus* [9, 6], a logical language for knowledge representation and reasoning about dynamic worlds in which actions have uncertain effects. In this article, we:

- Show how to represent knowledge about actions that have uncertain effects by exploiting the separation between *agent initiated actions* and nature's *random reaction*, first introduced in [8].

- Introduce observations as agent actions with uncertain effects. Thus, as in [1] observations are noisy; e.g., we may observe that the outside temperature is 25 degrees, but the reading may have an error with a Gaussian distribution.

As discussed in [5], there at least two essential tasks to be addressed when reasoning about change in worlds in which there are uncertain actions and noisy observations: [a] Probabilistic Temporal Projection: Given an initial state, a sequence of uncertain actions (an uncertain plan) and some fluent, determine the probability that the fluent will hold after the actions are executed in the initial state. [b] Probabilistic Belief Update: Given an initial state, a sequence of uncertain actions (an uncertain plan), an observation, and some fluent, determine the probability that the fluent will hold after the uncertain actions are executed in the initial state, and an uncertain (or *noisy*) observation is performed.

An important characteristic of the language of the probabilistic situation calculus is the hybrid nature of its semantics. In particular the logic includes the reals, and its usual operators, without axiomatizing them. Thus, we assume, at the semantic level, a fixed interpretation for them, and the satisfaction relation has to take this into account. When reasoning with such a theory, logical reasoning has to be combined with reasoning about the reals. The latter is done by appealing to an oracle (in our case, we implement this oracle using MATHEMATICA).

Furthermore, probability theory is assumed at the semantic level, and is not axiomatized. Our approach is radically different from approaches based on possible world semantics (e.g., [1]), as discussed in the full article.

State Transitions

Our work extends the Situation Calculus in the style presented in [11, 12], with the notion of uncertain actions. Recall that the situation calculus is a sorted logic with sorts $\mathcal{S}$ (situations), $\mathcal{A}$ (actions), $\mathcal{F}$ (fluents), etc. An essential component of any situation calculus theory is a description of how the world is affected by actions. This is done by means of effect axioms, which given a situation s, specify how the world would be in a situation $do(a, s)$. Here, a is an action term, while s and $do(a, s)$ are situation terms. The latter term denotes the situation that results from performing action a in situation s. For instance, if the action is $jump(x)$, meaning jump forward x meters, then one effect axiom could be:

$$holds(at(y - x), s) \supset holds(at(y), do(jump(x), s)). \tag{1}$$

At the heart of our proposal to integrating actions with uncertain outcomes, is the introduction of the sorts $\mathcal{I}$ (inputs) and $\mathbb{R}$ (reals), along with the requirement that any action corresponds to a pair $\langle i, r\rangle$ (an input and a real). The second component is referred to as a random *reaction*. For instance, we can introduce uncertainty in the previous example by having $jump_i(x)$ as an input, and the reaction being the actual distance jumped. Thus, the pair $\langle jump_i(3), 3.4\rangle$ would denote the action of jumping forward 3 meters but actually jumping 3.4 meters. The effect axiom (1) would now look like:

$$holds(at(y - r), s) \supset holds(at(y), do(\langle jump_i(x), r\rangle, s)). \tag{2}$$

As a second example, we model a dice throw as six different actions: one input (*throw*) and six possible reactions (the reals $1, 2, 3, 4, 5$ and 6). Pictorially: Thus, an agent's input (e.g., *throw*), has reactions that are indeterminate. Probability distributions are placed over the reactions. In the dice example, the distribution would be discrete, and in the *jump* example the distribution would be continuous. However, in order to have a uniform theory, without having to explicitly distinguish between continuous and discrete cases, we approximate discrete distributions by continuous ones that are described using the *Dirac Delta* function. Recall that the Dirac Delta of a real x is a function that is zero over $\mathbb{R}$ except in

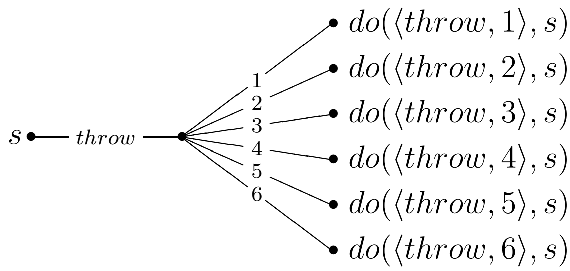

Figure 1: Throwing a dice, and possible reactions

an infinitesimal neighborhood around x, whose indefinite integral is 1. Thus, the probability density function in the case of the dice example, would be

$$dice(x) = \sum_{i=1}^{6} 1/6 \text{ DiracDelta}(i)(x).$$

This approach is completely general and allows for mixed distributions; i.e., inputs with reactions that have distributions that are continuous in some areas and discrete in others. Given that we have divided actions into its *input* and *reaction* components, we have decomposed the $Poss$ relation[1] into $iposs$ and $rposs$. Thus, we write:

$$Poss(\langle i, r\rangle, s) \equiv iposs(i, s) \wedge rposs(r, i, s). \tag{3}$$

$iposs$ is similar to the standard $Poss$ predicate. $rposs(r, i, s)$ is true when reaction r is possible after input i is performed in situation s. In the dice example,

$$rposs(r, throw, s) \equiv dice(r) > 0$$

As discussed in the full paper, we actually use the *cummulative distribution function (cdf)* instead of the density function. In general, we write $cdf(i, s)$ to denote the cdf of the reaction to input i in situation s. Thus $cdf(throw, s)(x)$ has value zero for values of x smaller than 0, value 1 for x greater or equal than 6, value $1/6$ when $0 \leq x < 1$, value $2/6$ when $1 \leq x < 2$, etc.

Notice that we are assuming the structure of the reals ($\mathbb{R}$) along with a very rich arsenal of real operators. We do not attempt to axiomatize $\mathbb{R}$, instead we assume standard interpretations for $\mathbb{R}$ and for the real operators. Furthermore,

[1] In the situation calculus, $Poss(a, s)$ is a relation that holds when action a can be executed in situation s.

for reasoning with the resulting theories, we assume the existence of an *oracle* for computations involving the reals. In particular, as shown in examples later, we take MATHEMATICA to be an implementation of this oracle. In essence, our approach mixes *logical reasoning* with *analytical calculus*.

Getting back to the dice example, if one wanted to compute the probability that that some fluent hold after input i is performed in a situation s, we would have to compute:

$$prob(f, i, s) = \int_{-\infty}^{+\infty} cf(holds(f, do(\langle i, x\rangle, s)))cdf(i, s)(dx), \qquad (4)$$

where the integral is of Lebesgue-Stieltjes type [13] and cf stands for characteristic function (1 if its argument is true, and zero otherwise).

Tree Structure

In the previous section, we concentrated on describing the approach to model uncertain actions, which derives from [9] (where only discrete distributions were considered), and extensions, presented in [6], to deal with continuous distributions. We have basically addressed the problem of describing the uncertainty that arises after inputs are performed in a known situation. Notice that a single input, in a fixed situation, gives rise to a potentially infinite set of possible successor situations. For instance, in formula (2), the distribution for the reaction r to a $jump(x)$ input can be a normal[2] $\mathtt{N}(0, 1)$; in which case the possible situations after a $jump(x)$ input would be infinite.

Let us consider now what happens when several inputs are performed in sequence[3]. For instance, when a dice is thrown several times in succession. Naturally we would be interested in computing the probability that some fluent holds after such a sequence is performed. In order to be able to pose such a query, we introduce input sequences as elements of discourse via a new sort $\mathcal{I}^*$ (a generic variable of type $\mathcal{I}^*$ will be $\vec{i}$). For instance, suppose that we define the fluent $Close$ as being at a distance of less than 1 from the origin:

$$holds(Close, s) \equiv holds(at(y), s) \wedge -1 < y < 1,$$

[2]We use $\mathtt{N}(\mu, \sigma)$ to denote the *cdf* of the Normal distribution with the usual parameters. Also, later, we use $U(\cdot, \cdot)$ to denote the cdf of the uniform distribution.

[3]In previous work, this problem was only addressed in the semantical and computational aspects

Then, we can ask what is the probability that $Close$ is true after some sequence of inputs (e.g., $\{jump_i(0.3), jump_i(0.3), jump_i(0.3)\}$) is performed.

A sequence of inputs and a situation define a *Markov Process*. Thus, we assume that the distribution of the outcomes resulting from an input depends only on the situation on which the input is performed. A situation s and an input sequence $\vec{i}$ define a subtree of the original situation structure. We can say that a situation s' is ***mp-accessible*** from a situation s and an input sequence $\vec{i}$, when s' is reached at the end of some evolution of the Markov process induced by performing the sequence of inputs $\vec{i}$ in the situation s. For instance, the situation $do(\langle jump_i(0.3), 0.25\rangle, do(\langle jump_i(0.3), 0.4\rangle, S_0))$ is ***mp-accessible*** from situation S_0, with the input sequence $\{jump_i(0.3),\ jump_i(0.3)\}$. The Markovian assumption lets us generalize (4). Thus we introduce an inductive definition of $prob$ on the input sequence $\vec{i}$.:

$$\begin{aligned}
&\vec{i} = \epsilon \supset prob(f, \vec{i}, s) = cf(holds(f, s)),\\
&(\exists\, i_0, \vec{i}_+).\vec{i} = (i_0; \vec{i}_+) \supset\\
&\qquad\qquad prob(f, \vec{i}, s) = \int_{-\infty}^{+\infty} prob(f, \vec{i}_+, do(\langle i_0, x\rangle, s))cdf(i_0, s)(dx).
\end{aligned} \tag{5}$$

Here x represents all the outcomes that can arise from performing i_0 in situation s. For each such outcome, we then consider all the outcomes that result from performing the remaining part of the input sequence.

Observations as Conditioning

In this section we define formally the *probabilistic temporal projection with observations*, which one might also call *Belief Updates*. The problem can be seen as solving probabilistic temporal projection (as informally defined in the previous section) but conditioned on one or more observations along the input sequence. To ease the presentation and for lack of space, we only consider a simplified case in which there is a single observation performed as a last input in the input sequence.

Observations are introduced into the logical language as a subsort $\mathcal{O}$ of $\mathcal{I}$. Thus, $\mathcal{O} \subseteq \mathcal{I}$, for convenience, we also introduce a sort predicate $observ = \mathcal{O}$. Notice that observations, together with their outcomes, are regular actions (inputs followed by reactions). For technical reasons (that we explain later), we introduce a fluent $Obs : \mathbb{R} \to \mathcal{F}$, such that if o is an observation (i.e., $o \in \mathcal{O}$):

$$holds(Obs(r), do(\langle o, r\rangle, s))$$

That is, Obs registers the outcome of the last observation performed. Furthermore, if $i \in \mathcal{I} \backslash \mathcal{O}$, then the only effect of observations is described with:

$$\neg observ(i) \supset holds(Obs(0), do(\langle i, r \rangle, s)).$$

Thus, after an input that is not an observation, the observed value is an arbitrary constant (e.g., 0). Notice that these axioms are domain independent.

Assuming that changes in the world are described using effect axioms, and the frame problem solved by using explanation closure axioms, then the resulting theory will lead to the conclusion that observations don't affect any fluents other than Obs. Indeed, the result of the completion should be that:

$$f \neq Obs(x) \supset holds(f, do(\langle o, x \rangle, s)) \equiv holds(f, s)$$

The axiom states that the state of the world does not change as result of performing observations. Note that all variables are implicitly universally quantified.

Notice that observations are handled in a radically different manner than in proposals based on epistemic logics (e.g., [1]). Basically, an observation action simply changes the nature of the distribution over the space of situations reached after performing actions. Thus, the distribution over the space of situations possible after input sequence $\vec{i}$ is changed after an observation o is made.

As in the previous example, the rest of the article considers only univariate outcomes for all the inputs. However, it is not difficult to extend this approach to multivariate cases (extending the presentation in [6] with observations).

Now, we extend the vocabulary of the probabilistic situation calculus to deal with updates based on observations (we deal with semantic issues in a later section). We introduce the operator $bel : \mathcal{F} \times \mathcal{I}^* \times \mathcal{S} \times \mathcal{O} \times \mathbb{R} \rightarrow \mathcal{P}$, with the following syntactic sugar: The term $bel(f, \vec{i}, s | o = r)$, denotes the probability that f holds, after input $\vec{i}$ is performed starting in s, and observing (at the end) reaction r to random observation o. We need to add an axiom to define[4]:

$$bel(f, \vec{i}, s | o = r) = \frac{prob(f \wedge Obs(r), (\vec{i}; o), s)}{prob(Obs(r), (\vec{i}; o), s)} \qquad (6)$$

In this expression we have used $f \wedge Obs(r)$ as syntactic sugar for $and(f, Obs(r))$; where $and : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{F}$ is a fluent operator. Here, if f_1, f_2 are fluent, $and(f_1, f_2)$ is a *defined* fluent (see [7]).

[4]This axiom is based on the definition of conditional probability $P(A|B) = P(A \wedge B)/P(B)$.

Note that to evaluate Bel in (6), we need to evaluate two $prob$ terms using (5). The expression (5) is a recursive definition for $prob$. By simple inspection, it turns out that to evaluate an expression like $prob(f, \vec{i}, s)$, where the input sequence $\vec{i}$ has dimension n requires solving an n-dimensional indefinite integral. Except for trivial cases, this integration is impossible to solve in exact terms. Before we discuss how we address this difficulty, we will discuss the semantics underlying the probabilistic situation calculus with observations.

Extended Probabilistic Situation Calculus Semantics

In this section, we introduce the semantics for the probabilistic extension to the situation calculus. We specify how the situation structure is generated and how the probability distributions are laid on top of this structure. The presentation of the semantics for the probabilistic situation calculus provided here is an extension to the Randomly Reactive Automata presented in [6]. In this extended abstract, we do not present the full semantics and leave standard aspects of Tarskian semantics out. In the full paper, the presentation is complete.

We provide a probabilistically sound especification for the computation of beliefs and probabilities based on the Randomly Reactive Automata. This should be understood as an *operational semantics*, in the sense that it specifies a mechanism to evaluate the prior and posterior probabilities (of fluents being true before or after certain observations). Notice that any system that reasons about these probabilities (like our oracle MATHEMATICA) has to obey this semantic specification.

As described formally below, in a Randomly Reactive Automaton with observations, actions are represented as pairs from $\mathcal{I}\mathbb{R}$, where $\mathcal{I}$ is the set of inputs (part of the specification of the Randomly Reactive Automaton), and $\mathbb{R}$ are the real numbers. These pairs correspond exactly to the input-reaction pairs of the last section. The situations are not explicitly introduced because they are simply sequences of $\mathcal{I}\mathbb{R}$ elements. Thus, we can correctly define the universe of situations as $S = (\mathcal{I}\mathbb{R})^*$. The empty string corresponds to the initial situation. We formally define *Randomly Reactive Automata* with observations as a tuple

$$\mathcal{A} = \langle I, O, F, T, M, g, h \rangle$$

where:

- I is a set of *inputs*. $O \subseteq I$ is a set of *observations*. F is a set of (proper) *fluents* (as specified in the full semantics, this set excludes the special fluent Obs).

- $T = \{T^f\}_{f\in F}$ where $T^f \subseteq S$ is such that

$$T^f_{\vec{i}\,s} = \{\vec{x} \in \mathbb{R}^n : si_1x_1 \dots i_nx_n \in T^f\}$$

  is Borel-measurable for all $\vec{i} = \langle i_1 \dots i_n\rangle \in I^*$ and $s \in S$. For each fluent $f \in F$, T^f represents the set of situations in which f holds.

- $M = \{M^{[s]}\}_{[s]\in[S]}$ where $M^{[s]} \subseteq I$ is the set of *possible inputs* at $[s]$, and $[S] = S/\approx$, $[s] = \{s' \in S : s \approx s'\}$ and $\approx$ is an equivalence relation of *indistinguishability* over S such that $s_1 \approx s_2$ iff $s_1 \in T^f \equiv s_2 \in T^f$ for all $f \in F$.

- $g = \{g_i^{[s]}\}_{i\in I\setminus O,[s]\in[S]}$, where $g_i^{[s]}$ is a generalized density function [6] of the *(hidden) reaction after performing input i at* $[s]$;

- $h = \{h_o^{[s]}\}_{o\in O,[s]\in[S]}$, where $h_o^{[s]}$ is a generalized density function of the *(observed) reaction after performing observation o at* $[s]$;

In order to completely specify the structures for probabilistic situation calculus interpretations, we need to add sets of objects for all domain sorts, along with sets of symbols for domain predicates and functions. Also, we need to specify the interpretation for the $prob$ and bel predicates, which are not fully axiomatized and play a special role in the theory. This latter part is completed in the next section. Armed with these interpretation structures it is relatively straightforward to provide a Tarskian style specification for the satisfaction relation between a Probabilistic Situation Calculus interpretation structure and a Probabilistic Situation Calculus formula. Moreover, as expected, first order reasoning is sound with respect to the proposed semantics. Both the Tarskian specification for the satisfaction and the soundness results are deferred to the full paper.

Belief

In this section we introduce the interpretation for the $prob$ and bel predicates in the light of randomly timed automata. In general, such interpretation requires solving an arbitrary n-dimensional indefinite integral, which is unfeasible via classical analytic calculus. To overcome this hurdle, we present a Monte-Carlo based method for approximating such integrals and discuss its efficiency for the applications in mind.

Note that, if no observation is made, the *belief* that a fluent f holds after executing a sequence of actions $\vec{i}$ at a situation s is just given by the probability

$prob(f, \vec{i}, s)$ as defined in (5). Given a Randomly Reactive Automata $\mathbf{A}$ we fix the value of $prob(f, \vec{i}, s)$ by calculating the probability of reaching a situation in T^f by a sequence of inputs denoted by $\vec{i} = i_1 \dots i_n$ from the situation denoted by s, more precisely:

$$[\![prob(f, \vec{i}, s)]\!]_{\mathbf{A}} = \int_{\mathbb{R}^n} 1_{T^f_{\vec{i}\,s}}(\vec{x}) \prod_{k=0}^{n-1} g_{i_{k+1}}^{[s\vec{i}_{k]}\vec{x}_{k]}]}(x_{k+1}) dx_n \dots dx_1 \tag{7}$$

where $1_{T^f_{\vec{i}\,s}}$ is the characteristic function of $T^f_{\vec{i}\,s}$ and $s\vec{i}_{k]}\vec{x}_{k]} = s_1 i_1 x_1 \dots i_k x_k$. This interpretation is sound with respect to the axiomatization presented in (5).

Unfortunately, solving this kind of integrals with all generality is an herculean task, and therefore methods of approximation are required. Literature, such as [3], clearly indicates that Monte-Carlo based methods are the most suitable for dealing with such probabilities. The overall idea is to sample over m independent random situations reached by $\vec{i}$ from s and take the rate of those that reach T^f to approximate the probability. In practice, this sampling can be easily obtained via a pseudo-random generator, which makes easy to encode this function in any language with this utility, like for instance, MATHEMATICA.

Proposition 1 The value $[\![\text{prob}(f, \vec{i}, s)]\!]_{\mathbf{A}}$ can be estimated via a Monte-Carlo approach by

$$\widehat{\text{prob}}_m(f, \vec{i}, s) = \frac{\sum_{k=1}^{m} 1_{T^f_{\vec{i}\,s}}(\vec{x}_k)}{m}$$

where $\{\vec{x}_k\}_{k=1,\dots,m}$ are m vectores of reactions obtained by a Monte-Carlo independent sampling of reactions obtained after executing input $\vec{i}$ from s. Moreover $\widehat{\text{prob}}_m(f, \vec{i}, s)$ is a consistent estimator of $\text{prob}(f, \vec{i}, s)$; that is, it converges with probability 1 to $[\![\text{prob}(f, \vec{i}, s)]\!]_{\mathbf{A}}$.

An advantage of the proposed Monte-Carlo algorithm is that, fixed s and $\vec{i}$, it is polynomial in the size of the sampling m. Therefore, if we want a more precise approximation, it is feasible just to consider more samples. An important addon result is that the (random) estimation error of this approximation has an asymptotic Gaussian distribution with mean 0 and standard deviation given by:

$$\sqrt{\frac{p(1-p)}{m}}$$

where $p = [\![\text{prob}(f, \vec{i}, s)]\!]_{\mathbf{A}}$.

A more precise notion of whether a fluent f holds or not might be achieved if an observation o is made after executing $\vec{i}$ from s. Naturally, we only gain any information with the observation if the latter is not *independent* from the fluent holding. If the outcome of observing o is independent from fluent f holding the following bad thing happens

$$bel(f, \vec{i}, s|\ o = r_1) = prob(f, \vec{i}, s),$$

meaning that observing o is totally useless to determine f. With our framework it is possible to measure the information gained by performing an observation o in order to know if a fluent f holds or not. This problem is of utmost relevance when observations are expensive and a plan of observations has to be set up. The full treatment of this problem is deferred to the full paper.

Independently of the value that an observation might provide to determine whether a fluent is holding or not, it is always possible to define in rigorous terms the *belief* of a fluent holding given an observation. Indeed, as presented for the discrete case in (6), for interpreting $bel(f, \vec{i}, s|o = r_1)$ we have to compute the conditional probability of reaching T^f by $\vec{i}$ from s given that observing o resulted in the value r_1. Hence, for a Randomly Reactive Automata $\mathbf{A}$ we have $[\![bel(f, \vec{i}, s|\ o = r_1)]\!]_{\mathbf{A}} =$

$$\lim_{y \to +\infty} \frac{\displaystyle\int_{-y}^{y} \cdots \int_{-y}^{y} 1_{T^f_{\vec{i}\,s}}(\vec{x}) h_o^{[s\vec{i}_{n]}\vec{x}_{n]}]}(r_1) \prod_{k=0}^{n-1} g_{i_{k+1}}^{[s\vec{i}_{k]}\vec{x}_{k]}]}(x_{k+1}) dx_n \dots dx_1}{\displaystyle\int_{-y}^{y} \cdots \int_{-y}^{y} h_o^{[s\vec{i}_{n]}\vec{x}_{n]}]}(r_1) \prod_{k=0}^{n-1} g_{i_{k+1}}^{[s\vec{i}_{k]}\vec{x}_{k]}]}(x_{k+1}) dx_n \dots dx_1}$$

Once again, solving the integrals presented above is an unfeasible task. However, contrasting with $prob$, approximation of conditional probabilities through Monte-Carlo methods is not straightforward, as we shall see.

Proposition 2 The value $[\![bel(f, \vec{i}, s|\ o = r_1)]\!]_{\mathbf{A}}$ can be estimated via a Monte-Carlo method by

$$\widehat{\text{bel}}_m(f, \vec{i}, s|\ o = r_1) = \frac{\sum_{k=1}^{m} 1_{T^f_{\vec{i}\,s}}(\vec{x}_k) h_o^{[s\vec{i}\vec{x}_k]}(r_1)}{\sum_{k=1}^{m} h_o^{[s\vec{i}\vec{x}_k]}(r_1)}$$

where $\{\vec{x}_k\}_{k=1,\ldots,m}$ are the first m vectors of reactions obtained by a Monte-Carlo independent sampling verifying $h_o^{[s\vec{i}\vec{x}_k]}(r_1) > 0$. Moreover $\widehat{\mathrm{bel}}_m(f, \vec{i}, s|\ o = r_1)$ is a consistent estimator of $bel(f, \vec{i}, s|\ o = r_1)$.

The need to guarantee m samples where there is positive probability density of obtaining r_1 from observing o is a shortcoming of Proposition 2. Observe that, if the reaction value r_1 has no probability (density) of being observed and it is, something very bad has happened, in particular, it is a clear evidence that one has misspecified the stochastic behavior of the agent. In this latter case it is impossible to obtain the required m samples, since observing the value r_1 is *inconsistent* with the specification, and therefore no *belief update* can be achieved. Putting aside such pathological behaviours, it still might be very hard to find situations where there is positive probability density of obtaining r_1 from observing o. In detail, for a starting situation s and sequence of inputs $\vec{i}$, the probability of reaching such situations is

$$p(\vec{i}, s, o, r_1) = \int_{\mathbb{R}^n} 1_{\{\vec{x}:h_o^{[s\vec{i}\vec{x}]}(r_1)>0\}}(\vec{x}) \prod_{k=0}^{n-1} g_{i_{k+1}}^{[s\vec{i}_k]\vec{x}_k]]}(x_{k+1}) dx_n \ldots dx_1.$$

Note, comparing with (7), that this value can be approximated via Monte-Carlo in the same way as $prob$. So, given s, $\vec{i}$, o and r_1, the next question arises: How big must a sample be in order to obtain m vectors in $\{\vec{x} : h_o^{[s\vec{i}\vec{x}]}(r_1) > 0\}$? It is intuitive that the expected value for the number of samples is $m/p(\vec{i}, s, o, r_1)$, further introspection leads to the following result.

Proposition 3 Let $M_m(\vec{i}, s, o, r_1)$ be the random variable defined as

$$M_m(\vec{i}, s, o, r_1) = \inf\left\{k\ :\ \sum_{j=1}^{k} 1_{\{\vec{x}:h_o^{[s\vec{i}\vec{x}]}(r_1)>0\}}(\vec{X}_j) \geq m\right\}$$

where $\vec{X}_j = (X_{1j}, X_{2j}, \ldots, X_{|\vec{i}|j})$ with $\vec{X}_1, \vec{X}_2, \ldots$ being independent sequences of reactions obtained after executing input $\vec{i}$ from s, and $|\vec{i}|$ denoting the cardinality of $\vec{i}$. Then $M_m \sim$ Negative Binomial $(m, p(\vec{i}, s, o, r_1))$; that is:

$$P(M_m(\vec{i}, s, o, r_1) = k) = \binom{k-1}{m-1}(1 - p(\vec{i}, s, o, r_1))^{k-m} p(\vec{i}, s, o, r_1)^m$$

for $k = m, m+1, \ldots$.

Thus, the random variable $M_m(\vec{i}, s, o, r_1)$ has expected value $m/p(\vec{i}, s, o, r_1)$ and variance $m(1 - p(\vec{i}, s, o, r_1))/p(\vec{i}, s, o, r_1)^2$. For large m, $M_m(\vec{i}, s, o, r_1)$ can be approximated by a normal distribution via

$$\frac{M_m(\vec{i}, s, o, r_1) - \frac{m}{p(\vec{i},s,o,r_1)}}{\sqrt{\frac{m(1-p(\vec{i},s,o,r_1))}{p(\vec{i},s,o,r_1)^2}}} \simeq \text{Normal } (0, 1)$$

where $\simeq$ stands for *approximate in distribution*.

Rewards can be introduced by considering a real valued (measurable) function from the set of indistinguishable situations. The reward of executing a plan of actions is obtained by approximating (via Monte-Carlo) the integral of such function over the probability measure associated to the plan. The information given by an observation for checking whether a fluent holds or not (value relevant for observation planning) can be approximated by this method, since the entropy of a sequence of actions given an observation can be obtained as a reward.

Conclusions

With this work we continued our work on the development of a logical language to support Knowledge Representation and Reasoning for dynamic domains that have uncertain actions. The main contribution of the research reported here (which should be considered upon those already established in [6]) is the effective treatment of *probabilistic temporal projection* and *probabilistic update belief* on agents having discrete, continuous or mixed probability distributions modeling their uncertain actions. In detail, we consider an extension of the Situation Calculus, which we call the Probabilistic Situation Calculus (PSC), and endow this logic with terms to deal with the probabilities related with beliefs. This language will be the ground to specifying an agent that executes uncertain actions. Since PSC is a first order logic, first-order reasoning is inherited and it is shown to be sound to a given semantics.

To deal with probabilistic reasoning we provide a Monte-Carlo algorithm as an efective interpretation of the terms dealing with beliefs, all other reasoning on real numbers relies on an oracle, that we implement using MATHEMATICA. Probabilistic Situation Calculus specifications can be translated to a conditional rewriting system, where it is also possible to encode the given Monte-Carlo algorithms (similar to what was done in [6] using MATHEMATICA, for a much simpler case). Due to lack of space, we were not able to present the reasoning capabilities of the rewriting system of MATHEMATICA endowed with our algorithms, nor were we given the space to provide convincing examples. Such gaps will be filled with the full paper. A detailed comparison with related works, as well as the treatment of observation planning and rewards is also to be included in the full version.

References

[1] F. Bacchus, J. Halpern, and H. Levesque. Reasoning about noisy sensors and effectors in the Situation Calculus. *Artificial Intelligence*, 111(1-2):171–208, 1999.

[2] C. Baral and L. Tuan. Reasoning about Actions in a Probabilistic Setting. In *Commonsense 2001, Fifth Symposium on Logical Formalizations of Commonsense Reasoning*, 2001. `http://www.cs.nyu.edu/faculty/davise/commonsense01/`.

[3] G. Fishman. *Monte-Carlo: Concepts, Algorithms, and Applications 2nd Ed.* Springer Verlag, 1996.

[4] H. Grosskreutz and G. Lakemeyer. Turning High-Level Plans into Robot Programs in Uncertain Domains. In *ECAI*, 2000.

[5] H. Grosskreutz and G. Lakemeyer. Belief Update in the pGOLOG Framework. In *Commonsense 2001, Fifth Symposium on Logical Formalizations of Commonsense Reasoning*, 2001. `http://www.cs.nyu.edu/faculty/davise/commonsense01/`.

[6] P. Mateus, A. Pacheco, J. Pinto, A. Sernadas, and C. Sernadas. Probabilistic situation calculus. *Annals of Mathematics and Artificial Intelligence*, 2001. To appear.

[7] J. Pinto. *Temporal Reasoning in the Situation Calculus.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, February 1994. URL = ftp://ftp.cs.toronto.edu/˜cogrob/jpThesis.ps.Z.

[8] J. Pinto, A. Sernadas, C. Sernadas, and P. Mateus. Non-determinism and uncertainty in the Situation Calculus. In *Proceedings of the 12th International Florida AI Research Symposium, FLAIRS'99*, pages 454–460. AAAI Press, 1999.

[9] J. Pinto, A. Sernadas, C. Sernadas, and P. Mateus. Non-determinism and uncertainty in the Situation Calculus. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 8(2), 2000.

[10] D. Poole. Decision theory, the Situation Calculus, and conditional plans. *Linköping Electronic Articles in Computer and Information Science*, 3(8), 1998. `http://www.ep.liu.se/ea/cis/1998/008/`.

[11] R. Reiter. *The Frame Problem in the Situation Calculus: A Simple Solution (sometimes) and a completeness result for goal regression*, pages 359–380. Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy. Academic Press, San Diego, CA, 1991.

[12] R. Reiter. Knowledge in action: Logical foundations for describing and implementing dynamical systems. Book Draft, available from `http://www.cs.utoronto.ca/cogrobo`, 2000.

[13] S.I. Resnick. *A Probability Path*. Birkhäuser Boston Inc., Boston, MA, 1999.