

Combining Probabilities, Failures and Safety in Robot Control

Alberto Finzi, Fiora Pirri

DIS - Università degli Studi di Roma “La Sapienza”

Via Salaria 113, 00198, Roma, Italy

{ finzi,pirri }@dis.uniroma1.it

Abstract

We present a formal framework for treating both incomplete information in the initial database and possible failures during an agent’s execution of a course of actions. These two aspects of uncertainty are formalized by two different notions of probability. We introduce also a concept of expected probability, which is obtained by combining the two previous notions. Expected probability accounts for the probability of a sentence on the hypothesis that the sequence of actions needed to make it true might have failed. Expected probability leads to the possibility of comparing courses of actions and verifying which is more safe.

1 Introduction

Uncertainty, both about the domain in which a robot has to operate and about the effective sequence of actions it is executing, is one of the most crucial problem that cognitive robotics has to face in robot control. In this paper we present a formal framework for dealing with uncertainty both in the initial database and during the execution of a course of actions: actions can fail. The failure of an action can lead to exogenous events, e.g. if the agent fails to pick up a block then the exogenous event that the block falls to the floor can take place. The complete framework is designed for autonomous manipulators endowed with perception. Perception, in fact, can be used to diagnose what really happened during a failed execution (a similar approach is taken in [9]). Here, leaving perception aside, we shall concentrate on the different aspects concerning uncertainty:

1. *Incomplete initial information.* The most prominent logical approaches to probability are [14; 4] and [11; 1; 21; 13; 7; 17]. We introduce in the Situation Calculus [15; 20] a formal theory, which turns out to be equivalent to the possible world approach, under certain restrictions (e.g. [13]). Instead of possible worlds, we propose a suitable axiomatization inducing possible state descriptions [4] therefore relying on a classical semantics. Probability of sentences is entailed by the axiomatization. We do not provide nested probabilities.

2. *Incomplete information for a course of actions, and exogenous events.* A logical approach to failures has been presented in [2]. In general, the problem of failures during the execution of plans is thoroughly treated in the planning community (see e.g. [16; 12]. To cope with the non deterministic effect of actions during the execution of a plan, say, we extend the classical language and axiomatization of the

Situation Calculus with stochastic actions and events so that different possible runs, for a given sequence of actions, can be obtained. However we preserve the core ontology of the Situation Calculus in which actions are deterministic and the whole dynamic of change is idealized. In fact, each run of stochastic actions is obtained as an expansion of a fixed sequence of deterministic actions. An expansion is formed by considering both probability of success and failure for each action, and exogenous events. We make a closed world hypothesis about exogenous events: the exogenous events are all and only those fixed in the theory. Probability of events is empirical, it accounts for the relative frequency of properties of events like the usual failures of an agent, due to its normal behavior or the circumstances under which it operates.

The probability of a sentence depends only on the amount of information available in the initial database, i.e. with state descriptions. Probability of actions, on the other hand, depends on properties of events, like a robot behavior. With the notion of expected probability we combine the two former notions (the logical and frequency ones) into a new probability that can account for the probability of a sentence after the execution of a course of actions, in the hypothesis that this sequence can fail in several different ways and exogenous actions can occur. This idea captures in some sense the dynamic execution under incomplete information ([8]). An example is the probability that the screwdriver is on the floor, given that the action the robot has to execute is to pick it up, and its goal is holding it in its hand. Expected probability, combining the two previous probabilities, gives a measure of safety of a sequence of actions chosen to achieve a goal. Therefore it can be used to determine whether a course of actions is more safe than another one. The notion of safety can be suitably extended to treat decision theory in the Situation Calculus (see e.g. [19; 3]).

2 Preliminaries

We consider the Situation Calculus (*SC*) ([15; 20]) as the core of our logic and language, and we suitably extend it to include new sorts, new symbols for its alphabet and new axioms. We refer the interested reader to [18] for a full presentation of the core logic and language and [9] for more details related to the extensions concerning probability on events. The core consists of the three sorts: *action*, *situation* and *object* = $worldEntities \cup genericObjects$. Here the sort *genericObject* is a catch-all that can include any possible additional sorts needed in an application, like the reals or sets. We have extended the core with three new

sorts: *outcome*, *stochasticAction*, *event*. The sort event consists of two sub-sorts, *event* = *condEvent* $\cup$ *seqEvent*. Observe that the sorted domain D could be infinite or uncountable even if some of its subsets is finite. For example we are interested in imposing a finite domain for world entities, while we would require probabilities to range over the reals. We shall use the following alphabet, with subscripts and superscripts, to denote terms: a , for variables of sort action; s , for variables of sort situation; sta , for variables of sort stochastic action; e , for variables of sort event; w for variables of the sub-sort *seqEvent* and u for variables of the sub-sort *condEvent*; p , for variables of sort object denoting probabilities and $x, y, z, \dots$, for variables of sort object (these might be further distinguished, e.g. for terms of sort *worldEntity*). The family of terms over SC and the set of formulae $\mathcal{L}_{SC}$ over SC are inductively defined as usual. In particular a *basic sentence* is one formed only by fluents, e.g. $\exists x \forall y (P(x, S_0) \rightarrow Q(x, y, S_0))$, in which fluents take as argument only elements of sort *worldEntities* (we). A *sentence uniform in σ* is one in which no variable, free or quantified, of sort situation is mentioned. E.g. $\exists s P(t, s)$ is not uniform and $\exists x (P(t, do(transmit(x), S_0)))$, is uniform in $\sigma = do(transmit(x), S_0)$.

Omitted quantifiers are always universal quantifiers.

3 The Basic Theory of Actions

The notions related to a basic theory of action with its set of foundational axioms and domain description axioms, are presented in detail in [20] and in [18]. These axioms represent an agent's prior knowledge about its domain of application, and its behavior, and are denoted by the following:

$$\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa} \cup \mathcal{D}_{una} \cup \mathcal{D}_{ap} \cup \Sigma.$$

Here, $\mathcal{D}_{S_0}$ is the initial database, describing what properties hold initially about the objects in the domain. $\mathcal{D}_{ssa} \cup \mathcal{D}_{ap}$ specify, respectively, the successor state axioms and the action precondition axioms. Σ is the set of foundational axioms for situations, and $\mathcal{D}_{una}$ ensures the uniqueness of each action. We present in the following a simple example to illustrate the basic ontology.

3.1 The basic ontology: a dispatching domain

An agent has to dispatch a string of bits from a source to a receiver and the information it has concerns some of the bits and the constraints that the string has to satisfy. The transmission can fail and thus an exogenous action that *distorts* the signal can take place. (A robotic oriented example is given in [9]).

Fluents $Source(x, n, s)$: At the source there is a bit $x \in \{0, 1\}$ in position n in situation s . E.g. $Source(1, 1, S_0) \wedge Source(0, 2, S_0)$ specifies that the string 10 is at the source in the initial situation S_0 .

$Receiver(x, n, s)$: At the receiver there is a bit $x \in \{0, 1\}$ in position n in situation s .

Control and Exogenous Actions $transmit(n)$. This is a control action with the following effect: the n -th bit at the source is transmitted.

$distort(n)$. This is an exogenous action with the following effect: the complement of the n -th bit at the source is transmitted.

The initial database $\mathcal{D}_{S_0}$ There are properties of the domain that the initial database has to represent. We list some of them, where $\bar{x}$ is complement of x :

1. $\neg Receiver(x, n, S_0)$
2. $Source(x, n, S_0) \rightarrow \neg Source(\bar{x}, n, S_0)$
3. $\exists n. (\forall n'. n' \leq n \rightarrow \exists y Source(y, n', S_0)) \wedge (\forall x \forall n''. n'' > n \rightarrow \neg Source(x, n, S_0))$

The first axiom says that no string of bits is at the *Receiver* in S_0 . The second says that at the source there is always one fixed bit in position n , the last specifies that the length of the string at the source is a priori fixed. Here for example we can fix $\forall x \neg Source(x, 4, S_0)$. The following constrains to the strings in *Source* are also in $\mathcal{D}_{S_0}$.

C_1 . If a bit b is in position $n > 2$, $\bar{b}$ is either in position $n - 1$ or in position $n - 2$:

$$Source(x, n, S_0) \wedge n > 2 \rightarrow Source(\bar{x}, n - 1, S_0) \vee Source(\bar{x}, n - 2, S_0)$$

C_2 . If a bit b is in position $n = 2$, $\bar{b}$ is in position $n = 1$:

$$Source(x, n, S_0) \wedge n = 2 \rightarrow Source(\bar{x}, n - 1, S_0)$$

For example: 011 is a legal string. 001 and 111 are not. Finally some axioms will state that bits, which we denote by b are limited to 1 and 0, and positions, which we denote by n , constitute a finite set of natural number. Both the sets are ordered and pairwise disjoint.

Successor State Axioms The successor state axiom for *Receiver* is:

$$Receiver(x, n, do(a, s)) \equiv (a = transmit(n) \wedge Source(x, n, S_0)) \vee (a = distort(n) \wedge Source(\bar{x}, n, S_0))$$

4 Incomplete initial information

If the initial database $\mathcal{D}_{S_0}$ is incomplete (see [20; 10]), we can assign a probability to those statements ψ , s.t. $\mathcal{D}_{S_0} \not\models \psi$. So, for example, if a goal G is given, we might be interested in finding a sequence σ of actions, such that $Prob(G(\sigma)) \geq 0.7$ is entailed by the theory. A classical, and certainly the most intuitive approach, is based on a possible world semantics, in which all worlds are assigned the same probabilities (see [2; 13] and the probability of a sentence ψ is obtained by summing up the probabilities of the worlds in which ψ is true. We take an analogous method but we avoid introducing possible worlds. In some sense our approach can be viewed as a way to treat worlds in the object language, under certain restrictions. The basic idea develops Carnap's [4] idea of defining state descriptions. In fact, we construct a set of sentences such that each sentence captures exactly one class of structures of the language. Here a class of structures is obtained by filtration: if two structures are isomorphic, w.r.t. the domain of sort we , then they belong to the same class. If the domain of sort we is finite, then we shall get a finite number of classes. As each state description identifies exactly a unique class of structures, we use these sentences to count those classes of structures which are models of $\mathcal{D}_{S_0}$. Having the number of classes of models of $\mathcal{D}_{S_0}$ we can finally assign probability to sentences, analogously as is done in the possible world approach. The advantage of our approach relies only on the fact that we filter the isomorphic models and we identify them directly in the language by the state description that we add to

the initial database. Observe that our approach is, however, possible just in case the domain of sort *we* has a fixed finite cardinality.

We shall now give few details about the formalization. Let $\mathcal{D}_{S_0}$ be the initial database, which we assume is a finite set of basic sentences, uniform in S_0 . The set of basic sentences uniform in S_0 is denoted, for short, by $\mathcal{L}^{we}$.

Given $\mathcal{D}_{S_0}$ we shall consider a set of *State Description* axioms (SD). SD is defined by the following set:

$$\begin{aligned} \{P_i^* = \exists x_1 \cdots \exists x_n \Phi_i(S_0) \wedge (x_1 \neq x_2 \wedge \cdots \wedge x_{n-1} \neq x_n) \wedge \\ \forall y(y = x_1 \vee \cdots \vee y = x_n)\}_{i \in I} \\ \text{and} \\ P_1^* \vee \cdots \vee P_q^* \end{aligned} \quad (1)$$

Here n is the cardinality of the domain of sort *we*, and $q \in I$ is an index which will be specified below (3). $\Phi_i(S_0)$, that appears in (1) is constructed as follows.

Definition 1 Let X be a set of variables. A function f_X is a variable renaming isomorphism iff f_X maps X onto X and is one-to-one. Let A and B be two sets of m -tuple of variables from X .

$$A =_{f_X} B \text{ iff for each } \langle x_1, \dots, x_m \rangle \in A \\ \langle f_X(x_1), \dots, f_X(x_m) \rangle \in B$$

f_X induces an equivalence relation between sets of m -tuples. An equivalence class of m -tuples is:

$$[A]_{f_X} = \{B \mid B =_{f_X} A\}$$

Example 1 Let $X = (x_1, x_2, x_3)$, $A = \{\langle x_1, x_1 \rangle, \langle x_2, x_1 \rangle\}$ and $B = \{\langle x_3, x_3 \rangle, \langle x_2, x_3 \rangle\}$. $f_X(x_1) = x_3$ and $f_X(x_2) = x_2$ is a variable renaming isomorphism, and $A =_{f_X} B$.

Let ϕ be a basic atom uniform in S_0 , of the form $F(\vec{x}, y)$ or $t(\vec{x}, y) = t'$, with $\vec{x}$ possibly empty, $y = S_0$ if it appears, and t, t' terms of sort *we*. Let $m \in \{a \mid \phi^a \in \mathcal{L}^{we}, a \text{ the arity of } \phi\}$. Let $X = \{x_1, \dots, x_n\}$:

$$\begin{aligned} C_m = \{\langle x_1, \dots, x_m \rangle_j \mid 1 \leq j \leq n^m, x_i \in X\} \\ \{A_m\}_{i \in I} \text{ a family of sets of } m\text{-tuples, s.t. for each } [B_m], \\ B_m \subset C_m, \text{ only one representative is in } \{A_m\}_{i \in I} \end{aligned}$$

$$h : \langle i, m \rangle \mapsto \{A_m\}_{i \in I} \quad h, \text{ is bijective and such that} \\ h(i, m) \in \{A_m\}_{i \in I}$$

The above definitions are used to form all the possible m -tuples out of n variables and ensure to gather only one representative for each equivalence class, under the renaming isomorphism. Let us order all the atoms ϕ_s , as defined above. Let ϕ_k be the k -th element in such ordering Ord , with $|Ord| = \alpha$. $\Phi_i(S_0)$ is defined as follows:

$$\Phi_i(S_0) = \bigwedge_{1 \leq k \leq \alpha} \left(\bigwedge_{j \in h(i, m)} \phi_k^m(\langle x_1, \dots, x_m \rangle_j) \wedge \right. \\ \left. \bigwedge_{j \in C_m \setminus h(i, m)} \neg \phi_k^m(\langle x_1, \dots, x_m \rangle_j) \right) \quad (2)$$

Observe that, given $n > 0$ the number of variables, and $m > 0$ the arity of the tuples, and $\phi_k \in Ord$,

$$\lambda_{\phi_k} = 2^{n^m} - \#f_X \text{ and } q = \prod_{\phi_k \in Ord} \lambda_{\phi_k} \quad (3)$$

Here $\#f_X$ is the number of renaming isomorphisms and $I = \{1, \dots, \lambda_{\phi_k}\}$.

To show that each $\Phi_i(S_0)$ and, consequently each P_i^* , is uniquely defined, many lemmata have to be used. Moreover

an equivalence relation, actually an isomorphism on structures, based only on the domain of sort *we*, and mirroring the renaming isomorphism, has to be defined. We skip all these details and leave only the main theorem.

Theorem 1 Let $\mathcal{M}$ be a structure of $\mathcal{L}_{SC}$ with domain of sort *we* with cardinality n . There exists a unique P_i^* such that $\mathcal{M} \models P_i^*$.

The above theorem tells us that each P^* is a possible state description and, by the uniqueness, there is a one-to-one correspondence between the P^* s and the classes of structures of $\mathcal{L}_{SC}$, with domain of sort *we* of fixed cardinality n .

Theorem 2 Let ϕ be a basic sentence, uniform in S_0 , such that $SD \cup \{\phi\}$ is consistent. There is an ordering on the P^* s, such that:

$$SD \models \phi \equiv (P_1^* \vee \cdots \vee P_q^*) \wedge \bigwedge_{k < j \leq q} \neg P_j^*$$

To count probabilities we need to extend $\mathcal{L}_{SC}$ with a number $p_1, \dots, p_q$ of parameters.

Lemma 1 Let $p_1, \dots, p_q$ be new parameters of the language, and let us add the following to $\mathcal{D}_{S_0}$:

$$(\bigwedge_i (p_i = 1 \equiv P_i^*) \wedge (p_i = 1 \equiv p_i \neq 0)) \quad (4)$$

Then there is an ordering on the P^* s such that $\mathcal{D}_{S_0} \cup SD$ entails both:

1. $\bigwedge \mathcal{D}_{S_0} \equiv P_1^* \vee \cdots \vee P_q^* \wedge \bigwedge_{j < k \leq n} \neg P_k^*$
2. $\bigvee_{1 \leq i \leq j} p_i = 1 \wedge \bigwedge_{j < k \leq n} p_k = 0$

Observe that the role of (4) is to assign to each model $\mathcal{M}$ of $\mathcal{D}_{S_0}$, which is also a model of a unique P_i^* a value 1 for the associated p_i . On the other if $\mathcal{M}_i$ is not a model of $\mathcal{D}_{S_0}$ the p_i associated with its unique state description P_i^* , will have everywhere value 0. We can, now, count the models of $\mathcal{D}_{S_0}$ as follows, where $\# \mathcal{M}$ is a new parameter:

$$\# \mathcal{M} = p_1 + \cdots + p_q \equiv \bigvee_{1 \leq i \leq j} p_i = 1 \wedge \bigwedge_{j < k \leq q} p_k = 0 \quad (5)$$

We are, now, ready to introduce the probability of sentences based on the equivalence classes of models of $\mathcal{D}_{S_0} \cup SD$. This can be done at the object level just by operating on the P^* s. First we define:

Definition 2 Let $\mathcal{D}_{S_0}$ be equivalent, for some ordering on the P^* s, to $\bigvee_{1 \leq i \leq j} P_i^* \wedge \bigwedge_{j < k \leq n} \neg P_k^*$, let $r \in \mathbb{R}$:

$$Prob(P_i^*) = \frac{1}{r} \text{ iff } r = \# \mathcal{M} \wedge P_1^* \vee \cdots \vee P_j^* \wedge \bigwedge_{j < k \leq n} \neg P_k^* \quad (i \neq k)$$

$$Prob(P_i^*) = 0 \text{ iff } \neg P_i^*$$

It follows, by Lemma 1 and Definition 5, that $\sum_i Prob(P_i^*) = 1$.

The following definition will give us the probability of a sentence with respect to the state descriptions.

Definition 3 For any basic formula ϕ , uniform in S_0 :

$$Prob(\phi) = \sum_{1 \leq i \leq n} Prob(P_i^*) \text{ iff } \phi \equiv \bigvee_{1 \leq i \leq j} P_i^* \wedge \bigwedge_{j < k \leq n} \neg P_k^* \quad (6)$$

Theorem 3 *The following properties are satisfied for each basic sentence ϕ , either uniform in S_0 , or equivalent to a sentence uniform in S_0 , in all models of $\mathcal{D}_{S_0} \cup SD$:*

1. $0 \leq \text{Prob}(\phi)$
2. $\text{Prob}(\top) = 1$
3. $\text{Prob}(\phi) = \text{Prob}(\phi \wedge \psi) + \text{Prob}(\phi \wedge \neg\psi)$
4. *If $\mathcal{D}_{S_0} \cup SD \models \phi \equiv \psi$ then $\text{Prob}(\phi) = \text{Prob}(\psi)$*

It follows that, if ψ is uniform in S_0 then:

$$\mathcal{D}_{S_0} \cup SD \models \psi \text{ iff } \text{Prob}(\psi) = 1.$$

Observe that differently from Bacchus and Halpern we do not allow any nesting of probabilities. In other words we do not admit sentences of the form: $\text{Prob}(\text{Prob}(\phi) \geq 0.3)$. However, given the above limitations, and the strict constraint that the domain of sort world entities has a fixed, finite, cardinality, the above theorem shows that our logic is equivalent to Halpern's logic, when to Halpern's axiomatization the finiteness axiom *FIN* is added.

Example 2 (Continued) Let $D_{we} = \{\{0, 1\}, \{1, 2, 3\}\}$. We assume that no information about the three bits in the source's string in S_0 is given, but the string is three bits and it satisfies C_1 and C_2 . The possible strings are:

$$st_1 = 100, st_2 = 101, st_3 = 010, st_4 = 011$$

Each string corresponds to a state description. E.g. 100 is captured by P_1^* with the following Φ_1 :

$$\begin{aligned} \Phi_1(s_0) = & \bigwedge_{i \in \{0,1\}} b_i = i \wedge \bigwedge_{1 \leq i \leq 3} n_i = i \wedge \bigwedge_{i,j} \neg \text{Receiver}(b_i, n_j, S_0) \\ & \wedge \text{Source}(b_1, n_1, S_0) \wedge \text{Source}(b_0, n_2, S_0) \wedge \text{Source}(b_0, n_3, S_0) \wedge \\ & \bigwedge_{(i,j) \notin \{(1,1), (0,2), (0,3)\}} \neg \text{Source}(b_i, n_j, S_0) \wedge \chi_1 \end{aligned}$$

Here χ_1 states the ordering defined respectively on bits and positions. P_2^*, P_3^* and P_4^* can be defined analogously. $\mathcal{D}_{S_0} \equiv P_1^* \vee P_2^* \vee P_3^* \vee P_4^* \wedge \bigwedge_{4 \leq i \leq q} \neg P_i^*$ where the P_j are all the other state descriptions admitted by the 5-structures of $\mathcal{L}_{we}$. Hence $\sharp\mathcal{M} = 4$ and $\text{Prob}(P_i^*) = \frac{1}{4}$. We want to evaluate the probability of the following sentence:

$$\Psi_s(S_0) \equiv \text{Source}(x, n, S_0) \wedge \text{Source}(\bar{x}, n+1, S_0)$$

Stating that *each bit b is followed by $\bar{b}$* , which is not entailed by $\mathcal{D}_{S_0}$. As $\Psi_s(S_0) \equiv P_2^* \vee P_3^* \wedge \bigwedge_{j \notin \{2,3\}} \neg P_j^*$ then $\text{Prob}(\Psi_s(S_0)) = \frac{1}{2}$.

5 A model for failures

The nice behavior of the Situation Calculus as a logic of change, is that once we have suitably defined truth and probabilities in the initial database, everything about uniform sentences in $\sigma > S_0$ (i.e. about the future of the initial database) can be easily computed in $\mathcal{D}_{S_0}$, using regression. Regression [18] is a proof method that, given a sentence uniform in a situation σ reduces such a sentence to an equivalent sentence in which the only situation mentioned is S_0 . Regression amounts to the following equivalence, given a sentence $\psi(\sigma)$:

$$\mathcal{D} \models \forall(\psi(\sigma) \equiv \mathcal{R}(\psi(\sigma)))$$

Here $\mathcal{R}(\psi(\sigma))$ is a sentence uniform in S_0 . Now, by the properties of probability shown in Theorem 3, regression allows us to compute the probability for any sentence uniform in σ .

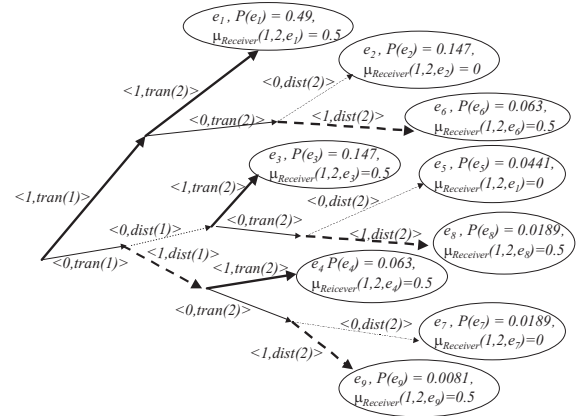


Figure 1: The tree $\text{tree}(\text{do}(\text{transmit}(2), \text{transmit}(1), S_0))$. The dotted arrow means the execution of an exogenous action.

Observe however that we do not have a method to compute the probability of any sentence, what so ever. For example we would not be able to compute the probability of a sentence of the form $\exists s \forall s' \exists s'' \exists x y (P(x, s) \wedge Q(x, y, s') \rightarrow R(y, x, s''))$. And since the definition of Prob is based on state descriptions, which are closed sentences, we cannot either express sentences of the form: $\forall x \text{Prob}(\phi(x))$.

Independently of probabilities, we have so far given an idealized representation of the world in which the effects of actions are exactly those intended, i.e. actions, even exogenous actions, are deterministic. However, due to various circumstances (e.g. noise in the channel), an agent might fail in its intended execution of an action and probabilities on sentences cannot capture this fact. To address this problem, we have extended the specification of an action theory with stochastic actions and events, therefore we introduce a complementary notion of probability as a relative frequency of events. We introduce in the sequel the functions, constants and relations of the extended $\mathcal{L}_{SC}$ that we use to model stochastic actions and probability of events, a more detailed description is in [9].

Stochastic Action. $\langle \cdot, \cdot \rangle : \text{outcome} \times \text{action} \mapsto \text{stochasticAction}$. E.g. $\langle v(a), a \rangle$. With $v : \text{action} \mapsto \text{outcome}$, e.g. $v(a) = 0$.

A *stochastic action* is a pair $\langle v(a), a \rangle$ with $v(a)$ taking values in the outcome space $\text{outcome} = \{0, 1\}$, where 0 means that a failed and 1 that it succeeded. Given a sequence of deterministic actions, this sequence expands to a tree of events due to all the possible outcomes of each action. See Figure 1.

To model the tree and the probability distribution we introduce the following notion of event.

Event. $\circ : \text{stochasticAction} \times \text{seqEvent} \mapsto \text{seqEvent}$. $| : \text{stochasticAction} \times \text{seqEvent} \mapsto \text{condEvent}$, and the special constant E_0 . E.g. $\langle 1, \text{distort}(\text{bit}2) \rangle \circ \langle 0, \text{transmit}(\text{bit}1) \rangle \circ E_0$ is a sequence of stochastic actions, i.e. is a term of sort seqEvent , hence a term of sort event. An event e is either a sequence event w of sort seqEvent or a conditional event u of sort condEvent , according to the following definition:

sequence event :

1. E_0 : the empty event.
2. $(\langle v(a), a \rangle \circ w)$: the event obtained as a result of executing a with value $v(a) \in \{0, 1\}$, after the sequence event w .

conditional event :

3. $(\langle v(a), a \rangle | w)$: the event obtained as a result of executing a with value $v(a) \in \{0, 1\}$, given the sequence event w

Probability of events. $\mathbf{P}: stochasticAction \times event \mapsto object$. E.g. $\mathbf{P}(\langle 0, distort(n) \rangle | E_0) = 0.9$. Probability on events, is a functional event fluent $\mathbf{P}$ which takes as arguments events and returns their probability: $\mathbf{P}$ maps events into $[0, 1]$.

Finally the following functions and predicates are used for the theory of events.

Measure. $\mu_P : object^n \times seqEvent \mapsto object$. E.g. $\mu_{Source}(bit, pos, E_0) = 0.7$. For each fluent P there is a belief measure μ_P .

$\prec: seqEvent \times seqEvent$. E.g. $E_0 \prec \langle v(a), a \rangle \circ E_0$.

Could. $stochasticAction \times event$. E.g. $Could(\langle 1, fall \rangle, S_0)$.

Axioms for probability $\mathbf{P}$ We shall not treat here axioms for events, we refer the reader to [9]. These axioms mirror the foundational axioms for situations, and for all events of sort sequence the analogous theorems as for situations [18] are derivable. The properties of $\mathbf{P}$ are:

1. $\mathbf{P}(e) \geq 0$, for any event e .
2. $\mathbf{P}(E_0) = 1$.
3. $\mathbf{P}(sta \circ e) = \mathbf{P}(sta | e) \times \mathbf{P}(e)$
4. $\mathbf{P}(\langle 0, a \rangle | e) = 1 - \mathbf{P}(\langle 1, a \rangle | e)$.

We define the following function:

$$trans(E_0) = S_0, \quad trans(\langle 0, a \rangle \circ e) = trans(e) \\ trans(\langle 1, a \rangle \circ e) = do(a, trans(e))$$

Let $\nu \in \{0, 1\}$. The following sets can be defined, with $ExAct$ a predicate that sorts out the exogenous actions:

$$tree(s) = \{ \langle \nu, a \rangle \circ w \mid ExAct(a) \wedge w \in tree(s) \vee \\ trans(\langle 1, a \rangle \circ w) \sqsubseteq s \} \\ leaves(s) = \{ w \mid w \in tree(s) \wedge \neg \exists w'. w' \in tree(s) \wedge w \preceq w' \}$$

Given the above axioms and the above functions, the following properties hold:

1. $\mathbf{P}(sta | e) = \frac{\mathbf{P}(sta \circ e)}{\mathbf{P}(e)}$
2. $\mathbf{P}(\langle 1, a \rangle | e) + \mathbf{P}(\langle 0, a \rangle | e) = 1$
3. Let $leaves(\sigma) = \{w_1, \dots, w_k\}$ then $\sum_i \mathbf{P}(w_i) = 1$.

5.1 Treating preconditions for stochastic actions

When a stochastic action sta is taken into account we need to know what holds at a given event in order to decide if sta is executable. In fact, we shall use the conditional probabilities to state that, if the preconditions are not satisfied, then the probability of success of the action will be 0. We need, therefore, suitable action preconditions axioms and successor state axioms concerning the dynamic of stochastic actions. Consider the dispatching domain and the control and exogenous actions given in paragraph 3.1: $transmit$ and $distort$. The corresponding stochastic actions are $\langle x, transmit(n) \rangle$ and $\langle x, distort(n) \rangle$, where x is a random variable on the values 1 or 0. Preconditions are defined as follows:

$$Could(\langle x, transmit(n) \rangle, e) \equiv \top \\ Could(\langle x, distort(n) \rangle, sta \circ e) \equiv sta = \langle 0, transmit(n) \rangle$$

The conditional probabilities for non exogenous action are:

$$\mathbf{P}(\langle 1, transmit(n) \rangle | e) = p \equiv Could(\langle 1, transmit(n) \rangle, e) \wedge \\ p = 0.7 \vee \neg Could(\langle 1, transmit(n) \rangle, e) \wedge p = 0 \quad (7)$$

The preconditions for the success of an exogenous action like $distort$ depends on the result of a previous action. Therefore the probability of an exogenous actions is conditioned on the *random effect* of stochastic actions.

$$\mathbf{P}(\langle 1, distort(n) \rangle | sta \circ e) = p \equiv \\ Could(\langle 1, distort(n) \rangle, e) \wedge sta = \langle 0, transmit(n) \rangle \wedge \\ p = 0.7 \vee p = 0 \wedge \neg Could(\langle 1, distort(n) \rangle, e) \quad (8)$$

Observe that, by the axioms for $\mathbf{P}$, we get a conditional probability for failure, for each of the above stochastic actions:

$$\mathbf{P}(\langle 0, a \rangle | e) = 1 - \mathbf{P}(\langle 1, a \rangle | e)$$

Successor state axioms for fluent measures. Since in the core language of SC fluents can take as argument only sequences of deterministic actions, by a principle of preservation, we prefer to extend the language with event fluents, namely μ -fluents, which we call *measures of fluents*. For each fluent $F(\vec{t}, s)$ we add a functional event fluent $\mu_F(\vec{t}, e)$, with the same arity as F and such that all arguments which are not of sort situation are the same as F and the argument of sort situation is replaced by an argument of sort event. The value of a μ -fluent is given as follows:

$$\mu_F(\vec{t}, E_0) = Prob(F(\vec{t}, S_0))$$

The successor state axioms for fluent measures capture the dynamic of the domain, in terms of measures, under the circumstances that stochastic actions occurred. These successor state axioms are strictly tied to the successor state axioms for fluents and they have to preserve the following property:

Property 1 For any μ -fluent $\mu_F(\vec{t}, e)$:

$$\mu_F(\vec{t}, e) = Prob(\mathcal{R}(F(\vec{t}, trans(e))))$$

According to the above property a simple method can transform a successor state axiom for a fluent into a successor state axiom for the corresponding μ -fluent. In particular, as a successor state axiom has the form $F(\vec{x}, do(a, s)) \equiv \Psi$, then we get $Prob(F(\vec{x}, do(a, s))) = Prob(\Psi)$. By suitable manipulations we get the required successor state axioms.

$$\mu_{Receiver}(x, n, sta \circ e) = p \equiv sta = \langle 1, transmit(n) \rangle \wedge \\ \mu_{Source}(x, n, S_0) = p \vee sta = \langle 1, distort(n) \rangle \wedge \\ p = 1 - \mu_{Source}(x, n, S_0).$$

5.2 Expected Probability and Safety

We want to combine the logical probability on sentences and the event probability defined on stochastic actions to get a third probability (see Theorem 4) that accounts for both incomplete information and possible failures of actions during execution. We are interested to know, given that $G(\sigma)$ is what an agent should achieve, what is the probability that G will hold at σ . We also want to know the probability that a sentence G' will hold at σ , given the evidence that G was to be achieved at σ . To capture these ideas we introduce the notion of *Expected Probability*. Let $Eset(\sigma', \sigma) = \{e_i : e_i \in leaves(\sigma), trans(e_i) = \sigma'\}$. Let $T(\sigma) = \{\sigma' \mid trans(e_i) = \sigma', e_i \in leaves(\sigma)\}$.

Definition 4 (Expected probability in S_0) Let ϕ and ψ be sentences uniform in σ . The expected probability, in S_0 , of ϕ , given evidence ψ is:

$$EP(\phi(\sigma) \mid \psi(\sigma)) = \\ \sum_{\sigma' \in T(\sigma)} (Prob(\mathcal{R}(\phi(\sigma')) \mid \mathcal{R}(\psi(\sigma)))) \times \sum_{e_i \in Eset(\sigma', \sigma)} \mathbf{P}(e_i)$$

Theorem 4 *EP satisfies the properties of Theorem 3.*

The above defined expected probability allows us to compare two sequences of actions and decide which one is safer:

Definition 5 (Safety) Let $\phi(\sigma_1)$ and $\phi(\sigma_2)$ be two sentence uniform, respectively, in σ_1 and σ_2 . Let $\preceq_{safe}$ be a ordering relation:

$$\phi(\sigma_1) \preceq_{safe} \phi(\sigma_2) \text{ iff } EP(\phi(\sigma_1)) \leq EP(\phi(\sigma_2))$$

Example 3 (Continued) Let $\Psi_r(\sigma)$ be the sentence $(Receiver(x, n, \sigma) \wedge Receiver(\bar{x}, n + 1, \sigma))$ where $\sigma = do(transmit(2), do(transmit(1), S_0))$. We consider the stochastic tree expansion of σ . Considering the set $\{\sigma' : \sigma' = trans(e), e \in leaves(\sigma)\}$, we get the following possible executions of σ , see Figure 1.

$$\begin{aligned} \sigma_1 &= do(transmit(2), do(transmit(1), S_0)) \\ \sigma_2 &= do(transmit(1), S_0), \sigma_3 = do(transmit(2), S_0) \\ \sigma_4 &= do(transmit(2), do(distort(1), S_0)), \sigma_5 = S_0 \\ \sigma_6 &= do(distort(2), do(transmit(1), S_0)) \\ \sigma_7 &= do(distort(1), S_0), \sigma_8 = do(distort(2), S_0) \\ \sigma_9 &= do(distort(2), do(distort(1), S_0)) \end{aligned}$$

By regression, $Prob(\mathcal{R}[\Psi_r(\sigma_i)]) = 1/2$, for $i \in \{1, 4, 6, 9\}$ and $Prob(\mathcal{R}[\Psi_r(\sigma_i)]) = 1$ for $i = \{2, 3, 5, 7, 8\}$. Considering the probability of events, from (7) and (8) we get:

$$\begin{aligned} P(\langle 0, distort(x) \rangle | E_0) &= 0.7 & P(\langle 0, transmit(x) \rangle | E_0) &= 0.3 \\ P(\langle 1, distort(x) \rangle | E_0) &= 0.3 & P(\langle 1, transmit(x) \rangle | E_0) &= 0.7 \\ P(e_1) &= 0.49 & P(e_2) &= 0.147 & P(e_3) &= 0.147 \\ P(e_4) &= 0.063 & P(e_5) &= 0.0441 & P(e_6) &= 0.063 \\ P(e_7) &= 0.0189 & P(e_8) &= 0.0189 & P(e_9) &= 0.0081 \end{aligned}$$

By Definition 4 we get $EP(\Psi_r(\sigma)) = 0.68795$.

6 Discussion

The notion of probability that we are concerned with, comes from the tradition of the *logical interpretations* of probability ([21]). The basic idea is that when assertions are not entailed it is always possible to conclude something about these assertions, a logical proximity as a *degree of deducibility*, treated as a probability on statements. The idea of attributing probabilities to logical statements was earlier introduced by Keynes's and illustrated in [14]. Carnap, in [4] addresses a distinction between *degree of confirmation* and *relative frequency*. In [13] Halpern proposes both a logic for statistical measure statements and probability statements. The former provides a probabilistic measure of events defined on a generic object: "the probability that a bird flies is 0.5". The latter is about probabilities of logical assertions: "the probability that Tweety flies is..". Following the tradition of *probability logic*, we have considered the Carnap's notion of degree of confirmation to declare a probability *Prob* on sentences and the relative frequency idea to declare probability *P* on actions. Bacchus and Halpern in [1; 13] presents two systems for both the statistical and confirmative approaches. A unified theory is presented in [7]. In particular, in [13; 1] axioms and rules of inference are presented and the logic is proved sound and complete (under given restrictions), w.r.t. a semantics based on possible worlds. Differently from Halpern's and Bacchus one, our theory is based on a classical Tarskian semantics. By suitably axiomatizing the initial database we get a semantical definition of probabilities analogous to the possible

worlds approach. We introduce axioms for state description, typical *extension formulas* used in finite model theory [5; 6]. Finally we have introduced a third probability, the *expected probability*, which can be used to compute the probability of a sentence given both incomplete knowledge and failures during the execution of a course of actions.

References

- [1] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press, Cambridge, Mass., 1990.
- [2] F. Bacchus, J. Halpern, and H. Levesque. Reasoning about noisy sensors in the situation calculus. *Artificial Intelligence*, 111:171–208, 1999.
- [3] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high level agent programming in the situation calculus. In *AAAI-2000*, pages 454–460. AAAI Press, 1999.
- [4] R. Carnap. *Logical Foundations of Probability*. University of Chicago Press, Chicago, 1950.
- [5] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41(1):50–58, 1976.
- [6] R. Fagin. Finite-model theory: a personal perspective. *Theoretical Computer Science*, 116:3–31, 1993.
- [7] R. Fagin and J. Halpern. Reasoning about knowledge and probability. *ACM*, 41(2):340–367, 1994.
- [8] Y. A. Feldman and D. Harel. A probabilistic dynamic logic. *Journal of Computer and System Sciences*, 28:193–215, 1984.
- [9] A. Finzi, F. Pirri, M. Pirrone, M. Romano, and M. Vaccaro. Autonomous mobile manipulators managing perception and failures. In *AGENTS'01, The Fifth International Conference on Autonomous Agents*, 2001.
- [10] A. Finzi, F. Pirri, and R. Reiter. Open world planning in the situation calculus. In *Proceedings of AAAI-2000, Seventeenth National Conference on Artificial Intelligence*, 2000.
- [11] H. Gaifman. Concerning measures in first-order calculi. *Israel Journal of Mathematics*, 2:1–18, 1964.
- [12] K. Z. Haigh and M. M. Veloso. Interleaving planning and robot execution for asynchronous user requests. *Autonomous Robots*, 5(1):79–95, 1998.
- [13] J. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.
- [14] J. M. Keynes. *A Treatise on Probability*. Macmillan, London, 1921.
- [15] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.
- [16] D. McDermott. Probabilistic projection in planning. In *Spatial and Temporal Reasoning. Dordrecht*. Dordrecht: Kluwer Academic, 1997.
- [17] J. Pinto, A. Sernadas, C. Sernadas, and P. Mateus. Non-determinism and uncertainty in the situation calculus. In *Proceedings of the FLAIRS'99 - the 12th International Florida AI Research Symposium*, pages 454–460. AAAI Press, 1999.
- [18] F. Pirri and R. Reiter. Some contributions to the metatheory of the situation calculus. *ACM*, 46(3):325–362, 1999.
- [19] D. Poole. Decision theory, the situation calculus, and conditional plans. *Linköping Electronic Articles in Computer and Information Science*, 3(8), 1998.
- [20] R. Reiter. *Knowledge in Action*. MIT press. To appear, www.utoronto.ca/cogrobo, 1998.
- [21] P. Roeper and H. Leblanc. *Probability Theory and Probability Semantics*. University of Toronto Press, 1999.