Reasoning about actions in a probabilistic setting

Chitta Baral and Le-Chi Tuan
Department of Computer Science and Engineering
Arizona State University
Tempe, Arizona 85287
chitta@asu.edu

Abstract

In this paper we present a language to reason about actions in a probabilistic setting and compare our work with earlier work by Pearl and (also briefly with) representations used in probabilistic planning. The main feature of our language is its use of static and dynamic causal laws, and use of unknown (or background) variables – whose values are determined by factors beyond our model – in incorporating probabilities. We also incorporate probabilities into reasoning with narratives.

1 Introduction and Motivation

One of the main goal of 'reasoning about actions' is to have a compact and elaboration tolerant [McC98] representation of the state transition due to actions. Many such representations – ETAI 98, vol 3 has several survey papers on these – have been developed in the recent literature. The common theme among them is to divide the description to four parts: the description of the effect of actions on fluents, the description of the causal relation between fluents, the description of the executability conditions for actions, and the description of the values of fluents at the initial time point. *But none of these elaboration tolerant representations consider probabilistic effect of actions.* When actions have probabilistic effects, the state transition due to actions is an MDP (Markov decision process). In an MDP we have the probabilities $p_a(s'|s)$ for all actions a, and states s' and s, which express the probability of the world reaching the state s' after the action a is executed in the state s'. *One of our main goal in this paper is to develop an elaboration tolerant representation for MDPs.*

There has been several studies and attempts of compact representation of MDPs in the decision theoretic planning community. Some of the representations that are suggested are probabilistic state-space operators (PSOs) [KHW95], 2 stage temporal Bayesian networks (2TBNs) [BDH95; BG96], sequential Effect trees (STs) [Lit97], and independent choice logic (ICL) [Poo97]. All these representation from the decision theoretic planning literature focus on planning only. Qualitatively, the two drawbacks of these representations are: (i) Although compact they do not aim at being elaboration tolerant. I.e., it is not easy in these formalisms to add a new causal relation between fluents or a new executability condition for an action, without making wholesale changes. (Poole in Section 1.1 of [Poo97] has a discussion on two views of knowledge representation; while 'elaboration tolerance' corresponds to his first view, 'compactness' corresponds to the second. He then mentions that his ICL is an instance of the second view.) (ii) They are not appropriate for reasoning about actions issues other than planning, such as: reasoning about values of fluents at a time point based on observations about later time points, and counter-factual reasoning about fluent values after a hypothetical sequence of actions taking into account observations. Pearl in [Pea99; Pea00] discusses this inadequacy at great length.

Besides developing an elaboration tolerant representation, *the other main goal of our paper* is to show how the other reasoning about action aspects of observation assimilation and counter-factual reasoning can be done in a probabilistic setting using our representation.

Our approach in this paper is influenced by [Pea99; Pea00]. Pearl argues that since Bayes nets do not require the DAGs to respect causality, they can not be used to represent and respond to external or spontaneous *changes*, and *first* suggests *Causal Bayesian Networks* – that require a stronger set of assumptions that rest on causal, and not merely associational knowledge – as a better alternative. Pearl then mentions two different ways of dealing with non-determinism: One based on Laplace's (1814) conception of natural phenomena, according to which nature's laws are deterministic, and randomness surfaces due to the reasoner's ignorance of the underlying boundary conditions. The second approach is to assume the world and nature's laws to be inherently stochastic, where determinism is a convenient approximation. Bayes nets are based on the second approach. Pearl argues that concepts such as: "the probability that event B occurred *due* to event A," and "the probability that event B would have been *different* if it were not for event A," can not be defined in terms of purely stochastic models, and require the Laplacian model. He also argues that the Laplacian model is more general in the sense that every stochastic model can be emulated by a Laplacian model (with stochastic inputs), but not the other way round, and the Laplacian model is more in tune with human intuition. With these arguments he proposes moving away from (Causal) Bayes nets to *functional causal models* where causal relationships are expressed in the form of deterministic, *functional* equations, and probabilities are introduced through the assumption that certain variables in the equations are unobserved.

As in the case of the functional causal models, *in this paper* we follow the Laplacian model in introducing probabilities through the assumption that certain variables are unobserved. (We call them 'unknown'[1] variables.) We differ from the functional causal models in two main ways: (i) We allow actions as first class citizens in our language, which allows us to deal with sequence of actions and (ii) In our formulation the relationship between fluents is given in terms of static causal laws, instead of structural equations. The static causal laws are more general, and more elaboration tolerant and can be compiled into structural equations. *Overall, our formulation can be considered as a generalization of Pearl's formulation of* causality *to a dynamic setting with a more elaboration tolerant representation.*

We now start with the syntax and the semantics our language PAL, which stands for *probabilistic action language.*

2 The Language PAL

The alphabet of the language PAL (denoting probabilistic action language) – based on the language $\mathcal{A}$ [GL93] – consists of three non-empty disjoint sets of symbols $\mathbf{F}$, $\mathbf{U}$, and $\mathbf{A}$. They are called the set of fluents, the set of unknown variables and the set of actions. A *fluent literal* is a fluent or a fluent preceded by $\neg$. An unknown variable literal is an unknown variable or an unknown variable preceded by $\neg$. A *literal* is either a fluent literal or an unknown variable literal. A *formula* is a propositional formula constructed from literals. A state (s) is an interpretation of fluents and unknown variables that satisfy certain conditions (to be mentioned while discussing semantics); an n-state (s) is an interpretation of only the fluents; and an u-state (s_u) is an interpretation of the unknown variables. For any state s, by s_u we denote the interpretation of the unknown variables of s. For any u-state s_u, $I(s_u)$ denotes the set of states s, such that $\mathsf{s}_u = s_u$. We say $\mathsf{s} \models s$, if the interpretation of fluents in s is same as in s.

PAL has four components: a domain description language PAL_D, a language to express unconditional probabilities about the unknown variables PAL_P, a language to specify observations PAL_O, and a query language.

[1] They are also referred to as [Pea99] 'background variables' and 'exogenous variables'. *They are variables whose values are determined by factors external to our model.*

2.1 PAL_D: The domain description language

Syntax

Propositions in PAL_D are of the following forms:

$$a \ \textbf{causes} \ \ \psi \ \textbf{if} \ \ \varphi \tag{2.1}$$

$$\varphi \ \textbf{causes} \ \ \psi \tag{2.2}$$

$$\textbf{impossible} \ \ a \ \textbf{if} \ \ \varphi \tag{2.3}$$

where a is an action, ψ is a fluent formula and φ is a formula of fluents and unknown variables. *Note that the above propositions guarantee that values of unknown variables are not affected by actions and are not dependent on the fluents. But the effect of an action on a fluent may be dependent on the unknown variables; also values of fluents in a state may be dependent on the unknown variables.*

Propositions of the form (2.1) describe the direct effects of actions on the world and are called *dynamic causal laws*. Propositions of the form (2.2), called *static causal laws*, describe causal relation between fluents and unknown variables in a world. Propositions of the form (2.3), called *executability conditions*, state when actions are not executable.

A *domain description* $\mathcal{D}$ is a collection of propositions in PAL_D.

Semantics of PAL_D: Characterizing the transition function

A domain description given in the language of PAL_D defines a transition function from actions and states to a set of states. Intuitively, given an action (a), and a state (s), the transition function (Φ) defines the set of states ($\Phi(a, \mathsf{s})$) that may be reached after executing the action a in state s. If $\Phi(a, \mathsf{s})$ is an empty set it means that a is not executable in s. We now formally define this transition function.

Let D be a domain description in the language of PAL_D. An *interpretation* I of the fluents and unknown variables in PAL_D is a maximal consistent literals of PAL_D. A literal l is said to be true (resp. false) in I iff $l \in I$ (resp. $\neg l \in I$). The truth value of a formula in I is defined recursively over the propositional connective in the usual way. For example, $f \wedge q$ is true in I iff f is true in I and q is true in I. We say that ψ holds in I (or I satisfies ψ), denoted by $I \models \psi$, if ψ is true in I.

A set of formulas from PAL_D is *logically closed* if it is closed under propositional logic (w.r.t. PAL_D).

Let V be a set of formulas and K be a set of static causal laws of the form φ **causes** ψ. We say that V is closed under K if for every rule φ **causes** ψ in K, if φ belongs to V then so does ψ. By $Cn(V \cup K)$ we denote[2] the least

[2] Note that a formula ψ can be equivalently represented as a static causal law *true* **causes** ψ.

logically closed set of formulas from PAL_D that contains V and is also closed under K.

A *state* s of D is an interpretation that is closed under the set of static causal laws of D.

An action a is *prohibited (not executable)* in a state s if there exists an executability condition of the form **impossible** a **if** φ in D such that φ holds in s.

The *effect of an action* a in a state s is the set of formulas $e_a(\mathsf{s}) = \{\varphi \mid D$ contains a law a **causes** ψ **if** φ and φ holds in $\mathsf{s}\}$.

Given the domain description D containing a set of static causal laws R, we formally define $\Phi(a, \mathsf{s})$, the set of states that may be reached by executing a in s as follows.

1. If a is not prohibited (i.e., executable) in s, then
$$\Phi(a, \mathsf{s}) = \{\mathsf{s}' \mid Cn(\mathsf{s}') = Cn((\mathsf{s} \cap \mathsf{s}') \cup e_a(\mathsf{s}) \cup R)\};$$
2. If a is prohibited (i.e., not executable) in s, then $\Phi(a, \mathsf{s})$ is $\emptyset$.

For other characterizations of $\Phi(a, \mathsf{s})$ and additional explanation and motivation behind the above definition please see the papers on causality in IJCAI 95 by Baral and McCain and Turner.

Every domain description D in a language PAL_D has a unique transition function Φ, and we say Φ is the transition function of D.

We now define an extended transition function (with a slight abuse of notation) that expresses the state transition due to a sequence of actions.

Definition 1 $\Phi([a], \mathsf{s}) = \Phi(a, \mathsf{s})$;
$\Phi([a_1, \ldots, a_n], \mathsf{s}) = \bigcup_{\mathsf{s}' \in \Phi(a_1, \mathsf{s})} \Phi([a_2, \ldots, a_n], \mathsf{s}')$

Definition 2 Given a domain description $\mathcal{D}$, and a state s, we write $\mathsf{s} \models_{\mathcal{D}} \varphi$ **after** $a_1, \ldots, a_n$, if φ is true in all states in $\Phi([a_1, \ldots, a_n], \mathsf{s})$.

(Often when it is clear from the context we may simply write $\models$ instead of $\models_{\mathcal{D}}$.)

2.2 PAL_P: Expressing probabilities of unknown variables

Syntax
A probability description $\mathcal{P}$ of the unknown variables is a collection of propositions of the following form:

$$\textbf{probability of } u \textbf{ is } n \tag{2.4}$$

where u is an unknown variable, and n is a rational number between 0 and 1.

Semantics
Each proposition above directly gives us the probability distribution of the corresponding unknown variable as: $P(u) = n$.
Since we assume (as does Pearl [Pea00]) that the values of the unknown variables are independent of each other defining the joint probability distribution of the unknown variables is straight forward.

$$P(u_1, \ldots, u_n) = P(u_1) \times \ldots \times P(u_n) \tag{2.5}$$

Note: $P(u_1)$ is a short hand for $P(U_1 = true)$. If we have multi-valued unknown variables then $P(u_1)$ will be a short hand for $P(U_1 = u_1)$.

Since several states may have the same interpretation of the unknown variables and we do not have any unconditional preference of one state over another, the unconditional probability of the various states can now be defined as:

$$P(\mathsf{s}) = \frac{P(\mathsf{s}_u)}{|I(\mathsf{s}_u)|} \tag{2.6}$$

2.3 PAL_Q: The Query language

Syntax
A query is of the form:

$$\textbf{probability of } [\varphi \textbf{ after } a_1, \ldots, a_n] \textbf{ is } n \tag{2.7}$$

where φ is a formula of fluents and unknown variables, a_i's are actions, and n is a rational number between 0 and 1. When $n = 1$, we may simply write: φ **after** $a_1, \ldots, a_n$, and when $n = 0$, we may simply write $\neg\varphi$ **after** $a_1, \ldots, a_n$.

Semantics: Entailment of Queries in PAL_Q
We define the entailment in several steps. First we define the transitional probability between states due to a single action. Since our transition function maps an action and a state to a set of states without any preference among the states, we have the following. (Later we discuss how the transition probability becomes non-uniform when we focus only on the fluent interpretations.)

$$\begin{aligned} P_{[a]}(\mathsf{s}'|\mathsf{s}) = P_a(\mathsf{s}'|\mathsf{s}) \quad &= \tfrac{1}{|\Phi(a,\mathsf{s})|} \text{ if } \mathsf{s}' \in \Phi(a, \mathsf{s}); \\ &= 0, \text{ otherwise }. \end{aligned} \tag{2.8}$$

We now define the (probabilistic) correctness of a single action plan given that we are in a particular state s.

$$P(\varphi \textbf{ after } a|\mathsf{s}) = \sum_{\mathsf{s}' \in \Phi(a,\mathsf{s}) \wedge \mathsf{s}' \models \varphi} P_a(\mathsf{s}'|\mathsf{s}) \tag{2.9}$$

Next we recursively define the transitional probability due to a sequence of actions, starting with the base case.

$$P_{[\ ]}(\mathsf{s}'|\mathsf{s}) = 1 \text{ if } \mathsf{s} = \mathsf{s}'; \text{ otherwise it is } 0. \tag{2.10}$$

$$P_{[a_1,\ldots a_n]}(\mathsf{s}'|\mathsf{s}) = \sum_{\mathsf{s}''} P_{[a_1,\ldots,a_{n-1}]}(\mathsf{s}''|\mathsf{s}) P_{a_n}(\mathsf{s}'|\mathsf{s}'') \tag{2.11}$$

We now define the (probabilistic) correctness of a (multi-action) plan given that we are in a particular state s.

$$P(\varphi \textbf{ after } \alpha|\mathsf{s}) = \sum_{\mathsf{s}' \in \Phi([\alpha],\mathsf{s}) \wedge \mathsf{s}' \models \varphi} P_{[\alpha]}(\mathsf{s}'|\mathsf{s}) \tag{2.12}$$

2.4 PAL_O: The observation language

Syntax

An observations description $\mathcal{O}$ is a collection of proposition of the following form:

$$\psi \textbf{ after } a_1, \ldots, a_n \tag{2.13}$$

where φ is a formula of fluents, and a_i's are actions.

When, $n = 0$, we simply write **initially** φ.

Semantics: assimilating observations in PAL_O

We now define the conditional probability of a state given that we have some observations.

$$P(\mathsf{s}_i|\mathcal{O}) = \frac{P(\mathcal{O}|\mathsf{s}_i)P(\mathsf{s}_i)}{\sum_{\mathsf{s}_j} P(\mathcal{O}|\mathsf{s}_j)P(\mathsf{s}_j}} \tag{2.14}$$

The above equation is based on Bayes' rule and the fact that for any state s and observation $\mathcal{O}$ either $\mathsf{s} \models \mathcal{O}$ or $\mathsf{s} \not\models \mathcal{O}$; i.e., there is no intermediate probability associated with this entailment.

2.5 Queries with observation assimilation

Finally, we define the (probabilistic) correctness of a (multi-action) plan given only some observations. This corresponds to counter-factual queries of Pearl [Pea00] when the observations are about a different sequence of actions than the one in the hypothetical plan.

$$P(\varphi \textbf{ after } \alpha|\mathcal{O}) = \sum_{\mathsf{s}\models\mathcal{O}} P(\mathsf{s}|\mathcal{O}) \times P(\varphi \textbf{ after } \alpha|\mathsf{s}) \tag{2.15}$$

Using the above formula, we now define the entailment between a theory (consisting of a domain description, a probability description of the unknown variables, and an observation description) and query:

Definition 3 $\mathcal{D} \cup \mathcal{P} \cup \mathcal{O} \models$ **probability of** $[\varphi$ **after** $a_1, \ldots, a_n]$ **is** n iff $P(\varphi$ **after** $a_1, \ldots, a_n|\mathcal{O}) = n$

3 Examples

We start with a simple example of the Yale shooting problem with probabilities. We have two actions *load* and *shoot*, and two fluents *loaded* and *alive*. To account for the probabilistic effect of the actions, we have two unknown variables u_1 and u_2. The effect of the actions *shoot* and *load* can now be described by $\mathcal{D}_1$ consisting of the following:

- *shoot* **causes** $\neg alive$ **if** $loaded, u_1$
- *load* **causes** *loaded* **if** u_2

The probabilistic effects of the action *shoot* and *load* can now be expressed by $\mathcal{P}_1$, that gives probability distributions of the unknown variables.

- **probability of** u_1 **is** p_1
- **probability of** u_2 **is** p_2

Now suppose we have the following observations $\mathcal{O}_1$.

- **initially** *alive*; **initially** $\neg loaded$

We can now show that $\mathcal{D}_1 \cup \mathcal{P}_1 \cup \mathcal{O}_1 \models Q_1$, where Q_1 is the query: **probability of** $[alive$ **after** $load, shoot]$ **is** $1 - p_1 \times p_2$.

3.1 Pearl's example of treatment and its effect on patients

In [Pea00], Pearl gives an example of a joint probability distribution which can be expressed by at least two different causal models, each of which have a different answer to a particular counter-factual question. We now show how both models can be modeled in our framework. In his example, the data obtained on a particular medical test where half the patients were treated and the other half were left untreated shows the following:

treated	alive	fraction
true	true	.25
true	false	.25
false	true	.25
false	false	.25

The above data can be supported by two different domain descriptions in PAL, each resulting in different answers to the following question involving counter-factuals. *"Joe was treated and he died. Did Joe's death occur due to the treatment. I.e., Would Joe have lived if he was not treated.*

Causal Model 1:

The domain description $\mathcal{D}_2$ of the causal model 1 can be expressed as follows, where the actions in our language are, *treatment* and *no_treatment*.

- *treatment* **causes** *action_occurred*
- *no_treatment* **causes** *action_occurred*
- $u_2 \wedge action\_occurred$ **causes** $\neg alive$
- $\neg u_2 \wedge action\_occurred$ **causes** *alive*

The probabilities of unknown variables $\mathcal{P}_2$ can be expressed as follows:

- **probability of** u_2 **is** 0.5

The probability of the *occurrence* of *treatment* and *no_treatment* is 0.5 each. (Our current language does not allow expression of such information. Although, it can be easily augmented, to accommodate such expressions, we do not do it here as it does not play a role in the analysis we are making.)

Assuming u_2 is independent of the occurrence of *treatment* it is easy to see that the above modeling agrees with data table given earlier.

The observations $\mathcal{O}_2$ can be expressed as follows:

- **initially** $\neg action\_occurred$ **initially** *alive*
- $\neg alive$ **after** *treatment*

We can now show that $\mathcal{D}_2 \cup \mathcal{P}_2 \cup \mathcal{O}_2 \not\models Q_2$, where Q_2 is the query: *alive* **after** *no_treatment*; and $\mathcal{D}_2 \cup \mathcal{P}_2 \cup \mathcal{O}_2 \models \neg alive$ **after** *no_treatment*

Causal Model 2:
The domain description $\mathcal{D}_3$ of the causal model 2 can be expressed as follows:

- *treatment* **causes** $\neg alive$ **if** u_2
- *no_treatment* **causes** $\neg alive$ **if** $\neg u_2$

The probabilities of unknown variables $\mathcal{P}_3$ is same as $\mathcal{P}_2$. The probability of occurrence of *treatment* and *no_treatment* remains 0.5 each. Assuming u_2 is independent of the occurrence of *treatment* it is easy to see that the above modeling agrees with data table given earlier.

The observations $\mathcal{O}_3$ can be expressed as follows:

- **initially** *alive* $\qquad$ $\neg alive$ **after** *treatment*

Unlike in case of the causal model 1, we can now show that $\mathcal{D}_3 \cup \mathcal{P}_3 \cup \mathcal{O}_3 \models Q_2$.

4 Other features of PAL and comparisons

In this section we point out several features of our formulation and compare our formulation with Pearl's formulation in [Pea00].

4.1 The transition function between n-states

As defined in (2.8) the transition probability $P_a(\mathsf{s}'|\mathsf{s})$ has either the value zero or is uniform among the s' where it is non-zero. This is counter to our intuition where we expect the transition function to be more stochastic. This can be explained by considering n-states and defining transition functions with respect to them. In our definition we will use the following notation. Let s be an n-state and s be a state.

We can then define $\Phi(a, s)$ where s is an n-state as:
$\Phi(a, s) = \{s' : s'$ is the interpretation of the fluents of a state in $\bigcup_{\mathsf{s} \models s} \Phi(a, \mathsf{s})\}$.

We can then define a more stochastic transition probability $P_a(s'|s)$ where s and s' are n-states as follows:

$$P_a(s'|s) = \sum_{\mathsf{s}_i \models s} (P(\mathsf{s}_i) \sum_{\mathsf{s}'_j \models s'} P_a(\mathsf{s}'_j|\mathsf{s}_i)) \qquad (4.16)$$

4.2 Impact of observations on the transition function

Observations have no impact on the transition function $\Phi(a, \mathsf{s})$ or on $P_a(\mathsf{s}'|\mathsf{s})$. But they do affect $\Phi(a, s)$ and $P_a(s'|s)$. Let us analyze why.

Intuitively, observations may tell us about the unknown variables. This additional information is monotonic in the sense that since actions do not affect the unknown variables there value remains unchanged. Thus, in presence of observations $\mathcal{O}$, we can define $\Phi_{\mathcal{O}}(a, s)$ as follows:

$\Phi_{\mathcal{O}}(a, s) = \{s' : s'$ is the interpretation of the fluents of a state in $\bigcup_{\mathsf{s} \models s \& \mathsf{s} \models \mathcal{O}} \Phi(a, \mathsf{s})\}$

As evident from the above definition, as we have more and more observations the transition function $\Phi_{\mathcal{O}}(a, s)$ becomes more deterministic.

On the other hand, as we mentioned earlier the function $\Phi(a, \mathsf{s})$ is not affected by observations. Thus, we can accurately represent two different kind of non-deterministic effects of actions: the effect on states, and the effect on n-states.

4.3 Comparing our notion of representing causality with Pearl's notion

There are several difference between our approach and Pearl's approach. We list some of them briefly here.

(**1**) Pearl represents causal relationships in the form of deterministic, functional equations of the form

$$v_i = f_i(pa_i, u_i) \qquad (4.17)$$

$pa_i \subset U \cup V \setminus \{v_i\}$, and $u_i \in U$, where U is the set of unknown variables and V is the set of fluents. Such equations are only defined for v_i's from V.

In our formulation instead of using such equations we use static causal laws of the form (2.2), and restrict ψ to only fluents. I.e., it does not contain unknown variables. A set of such static causal laws define functional equations which are embedded inside the semantics. The advantage of using such causal laws over the equations used by Pearl is the ease with which we can add new static causal laws. We just add them and let the semantics take care of the rest. (This is one manifestation of the notion of 'elaboration tolerance'.) On the other hand Pearl would have to replace his older equation by a new equation. Moreover, if we did not restrict ψ to be a a formula of only fluents, we could have written $v_i = f_i(pa_i, u_i)$ as the static causal law *true* **causes** $v_i = f_i(pa_i, u_i)$.

(**2**) We see one major problem with the way Pearl reasons about actions (which he calls 'interventions') in his formulation. To reason about the intervention which assigns a particular value v to a fluent f, he proposes to modify the original causal model by removing the link between f and its parents (i.e., just assigning v to f by completely forgetting the structural equation for f), and then reasoning with the modified model. This is fine in itself, except that if we need to reason about a sequence of actions, one of which may change values of the predecessors of f (in the original model) that may affect the value of f. Pearl's formulation will not allow us to do that, as the link between f and its predecessors has been removed when reasoning about the first action.

Since actions are first class citizens in our language we do not have such a problem. In addition, we are able to reason about executability of actions, and formulate indirect qualification, where static causal laws force an action to be in-executable in certain states. In Pearl's formulation, all interventions are always possible.

4.4 Comparison with other work

Earlier we mentioned the representation languages for probabilistic planning and the fact that their focus is not from the point of view of elaboration tolerance. We would like to add that even if we consider the Bayes net representations as suggested by Boutilier and Goldszmidt, our approach is more general as we allow cycles in the causal laws, and by definition they are prohibited in Bayes nets.

5 Extending PAL to reason with narratives

Our language PAL can be further generalized to incorporate narratives [MS94], where we have time points as first class citizens and we can observe fluent values and action occurrences at these time points and do tasks such as reason about missing action occurrences, make diagnosis, plan from the current time point, and counter-factual reasoning about fluent values if a different sequence of actions had happened in a past (not just initial situation) time point. Here, we give a quick overview of how PAL can be extended to allow narratives. We refer to this extended language as $PALN$.

$PALN$ has a richer observation language $PALN_O$ consisting of propositions of the following forms:

$$\varphi \ \mathbf{at} \quad t \tag{5.18}$$

$$\alpha \ \mathbf{between} \quad t_1, t_2 \tag{5.19}$$

$$\alpha \ \mathbf{occur\_at} \quad t \tag{5.20}$$

$$t_1 \ \mathbf{precedes} \quad t_2 \tag{5.21}$$

where φ is a fluent formula, α is a (possibly empty) sequence of actions, and t, t_1, t_2 are time points (also called situation constants) which differ from the current time point t_C.

A narrative is a pair $(\mathcal{D}, \mathcal{O}')$, where $\mathcal{D}$ is a domain description and $\mathcal{O}'$ is a set of observations of the form (5.18-5.21). Observations are interpreted with respect to a domain description. While a domain description defines a transition function that characterize what states may be reached when an action is executed in a state, a narrative consisting of a domain description together with a set of observations defines the possible histories of the system. This characterization is done by a function Σ that maps time points to action sequences, and a sequence Ψ, which is a finite trajectory of the form $\mathsf{s}_0, a_1, \mathsf{s}_1, a_2, \ldots, a_n, \mathsf{s}_n$ in which $\mathsf{s}_0, \ldots, \mathsf{s}_n$ are states, $a_1, \ldots, a_n$ are actions and $\mathsf{s}_i \in \Phi(a_i, \mathsf{s}_{i-1})$ for $i = 1, \ldots, n$. Models of a narrative $(\mathcal{D}, \mathcal{O}')$ are interpretations $\mathcal{M} = (\Psi, \Sigma)$ that satisfy all the facts in $\mathcal{O}'$ and minimize unobserved action occurrences. (A more formal definition is given in [BGP97].) A narrative is *consistent* if it has a model. Otherwise, it is *inconsistent*. When $\mathcal{M}$ is a model of a narrative $(\mathcal{D}, \mathcal{O}')$ we write $(\mathcal{D}, \mathcal{O}') \models \mathcal{M}$.

Next we define the conditional probability that a particular pair $\mathcal{M} = (\Psi, \Sigma) = ([\mathsf{s}_0, a_1, \mathsf{s}_1, a_2, \ldots, a_n, \mathsf{s}_n], \Sigma)$ of trajectories and time point assignments is a model of a given domain description $\mathcal{D}$, and a set of observations. For that we first define the weight of a $\mathcal{M}$ (with respect to $\mathcal{D}$ which is understood from the context) denoted by $Weight(\mathcal{M})$ as:

$$\begin{aligned} Weight(\mathcal{M}) \quad &= 0 \text{ if } \Sigma(t_C) \neq [a_1, \ldots, a_n]; \mathit{and} \\ &= P(\mathsf{s}_0) \times P_{a_1}(\mathsf{s}_1|\mathsf{s}_0) \times \ldots \times P_{a_n}(\mathsf{s}_n|\mathsf{s}_{n-1}) \\ &\text{otherwise.} \end{aligned} \tag{5.22}$$

Given a set of observation $\mathcal{O}'$, we then define

$$\begin{aligned} P(\mathcal{M}|\mathcal{O}') \quad &= 0 \text{ if } \mathcal{M} \text{ is not a model of } (\mathcal{D}, \mathcal{O}'); \\ &= \frac{Weight(\mathcal{M})}{\sum_{(\mathcal{D},\mathcal{O}') \models \mathcal{M}'} Weight(\mathcal{M}')} \text{ otherwise.} \end{aligned} \tag{5.23}$$

The (probabilistic) correctness of a multi-action plan from a time point t with respect to a model $\mathcal{M}$ can then be defined as

$$\begin{aligned} P(\varphi \ \mathbf{after} \ \ \alpha \ \mathbf{at} \ t|\mathcal{M}) \quad &= \sum_{\mathsf{s}' \in \Phi([\beta], \mathsf{s}_0) \wedge \mathsf{s}' \models \varphi} P_{[\beta]}(\mathsf{s}'|\mathsf{s}_0) \\ &\text{where } \beta = \Sigma(t) \circ \alpha \end{aligned} \tag{5.24}$$

Finally, we define the (probabilistic) correctness of a (multi-action) plan from a time point t given a set of observations. This corresponds to counter-factual queries of Pearl [Pea00] when the observations are about a different sequence of actions than the one in the hypothetical plan.

$$\begin{aligned} P(\varphi \ \mathbf{after} \ \ \alpha \ \mathbf{at} \ t|\mathcal{O}') \quad &= \sum_{(\mathcal{D},\mathcal{O}') \models \mathcal{M}} P(\mathcal{M}|\mathcal{O}') \\ &\times P(\varphi \ \mathbf{after} \ \ \alpha \ \mathbf{at} \ t|\mathcal{M}) \end{aligned} \tag{5.25}$$

One major application of the last equation is that it can be used for action based diagnosis [BMS00], by having φ as $ab(c)$, where c is a component. Due to lack of space we do not further elaborate here.

6 Conclusion and Future Work

In this paper we showed how to integrate probabilistic reasoning into 'reasoning about actions', and compare our work with Pearl's formulation and research in probabilistic planning. Among the future directions, we believe that our formulation can be transported to other formulations of reasoning about actions (such as situation calculus) and can be used in adding probabilistic concepts to other action based formulations (such as, diagnosis, and agent control), and execution languages (such as, GOLOG, and Congolog).

References

[BDH95] C. Boutilier, T. Dean, and S. Hanks. Planning under uncertainty: Structural assumptions and computational leverage. In *Proc. 3rd European Workshop on Planning (EWSP'95)*, September 1995.

[BG96] C. Boutilier and M. Goldszmidt. The frame problem and bayesian network action representations. In *Proc. of CSCSI-96*, May 1996.

[BGP97] C. Baral, M. Gelfond, and A. Provetti. Representing Actions: Laws, Observations and Hypothesis. *Journal of Logic Programming*, 31(1-3):201–243, May 1997.

[BMS00] C. Baral, S. McIlraith, and T. Son. Formulating diagnostic problem solving using an action language with narratives and sensing. *Proc. of KR 2000*, 311-322, 2000.

[GL93] M. Gelfond and V. Lifschitz. Representing actions and change by logic programs. *Journal of Logic Programming*, 17(2,3,4):301–323, 1993.

[KHW95] N. Kushmerick, S. Hanks, and D. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76(1-2):239–286, 1995.

[Lit97] M. Littman. Probabilistic propositional planning: representations and complexity. In *AAAI 97*, pages 748–754, 1997.

[McC98] J. McCarthy. Elaboration tolerance. In *Common Sense 98*, 1998.

[MS94] R. Miller and M. Shanahan. Narratives in the situation calculus. *Journal of Logic and Computation*, 4(5):513–530, October 1994.

[Pea99] J. Pearl. Reasoning with cause and effect. In *IJCAI 99*, pages 1437–1449, 1999.

[Pea00] J. Pearl. *Causality*. Cambridge University Press, 2000.

[Poo97] D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94(1-2 (special issue on economic principles of multi-agent systems)):7–56, 1997.