Representation and reasoning about evolution of the world in the context of reasoning about actions

Chitta Baral & Nam Tran
Department of Computer Science and Engineering
Arizona State University
Tempe, Arizona 85287
$\{$*chitta,namtran*$\}$*@asu.edu*

Abstract

The first step in reasoning about actions and change involves reasoning about how the world would evolve if a certain action is executed in a certain state. Most research on this assume the evolution to be only a single step and focus on formulating the transition function that defines changes between states due to actions. In this paper we consider cases where the evolution is more than just a single change between one state and another. This is manifested when the execution of an action may trigger execution of other actions, or when multiple agents act on the environment following certain strategies.

1 Introduction and Motivation

In the last one and half decade there has been tremendous progress in representing and reasoning about actions [Georgeff, 1994; Lifschitz, 1997; Sandewall, 1998] and its application to planning, hypothetical reasoning, reasoning about narratives, automatic control generation and verification, explanation and diagnosis. At an abstract level there are three main components of representing and reasoning about actions (a) Specifying the various possible ways the world could evolve, (b) Specifying a plan or execution program, and (c) Specifying a property over trajectories. For example, let us consider the simple language $\mathcal{A}$ [Gelfond and Lifschitz, 1993]. In it the specification for how the world could evolve includes statement of the form a **causes** f **if** $p_1, \ldots, p_n$ and statements of the form **initially** f. Together they define a transition function $\Phi : States \times Actions \rightarrow States$, and based on the transition function they define a set of possible trajectories of the form $s_0, a_1, s_1 \ldots$ such that s_0 satisfies all the statements of the form **initially** f, and $s_i = \Phi(s_{i-1}, a_i)$, for all $i > 0$. In $\mathcal{A}$ the language for plans – part (b) above, is a simple one: plans are simply sequence of actions. In $\mathcal{A}$, part (c) is about goals that hold in the final state, and are represented by fluent literals. Altogether, $\mathcal{A}$ defines the entailment relation $\models$, as in $D \models f$ **after** P, where D is a description corresponding to (a), f is fluent literal corresponding to (c), and P is of the form $a_1, \ldots, a_n$ corresponding to (b). In later languages more general (a), (b) and (c) are proposed. For example, in [McCain and Turner, 1995] the transition function is also dictated by static causal laws; and in [Baral *et al.*, 1997] observations about different time points play a role in defining valid trajectories. Similarly, in [Levesque, 1996; Son and Baral, 2001] more general plans such as conditional plans are considered and in [Baral *et al.*, 2001] more general languages for (c) are considered.

In this paper our goal is to consider some additional aspects of specifying how the world could evolve, beyond what has been considered so far. In this regard, most of the current proposals focus on defining what valid states of the world are, defining the transition between states of the world due to actions, and incorporating observations to determine valid trajectories. The additional aspects that we are concerned about in this paper involves the ability to express trigger conditions such as: (i) whenever a fluent f (or a fluent formula) is true in a state then a certain action *must* happen in that state; and (ii) a particular action *must* be followed by another particular action. Triggers of type (i) are often encountered in active databases [Widom and Ceri, 1996]. One also encounters triggers of types (i) and (ii) when modeling cell-signaling [Hancock, 1997], when a particular set of conditions trigger a particular event inside the cell, or one event in the cell triggers another event. We are also interested in multi-agent environments when each agent is following a particular policy and a particular agent would like to figure out how the world will evolve in response to its actions. For example consider an environment which has two agents A and B. Agent A is aware of how agent B would react to any action that agent A does. Now agent A would like to reason about the effect of its own action on how the world will evolve. In that case the evolution of the world not only depends on the action that A executes and the state transition due to that but also the action of B that is triggered by A's action.

To make our motivation more explicit let us go back to the language $\mathcal{A}$. In $\mathcal{A}$ a query is of the form f **after** $a_1, \ldots a_n$, where the agent wants to find out if f is true in the state (corresponding to the situation) reached after executing the sequence of actions $a_1, \ldots, a_n$ in the initial situation. If the state corresponding to the initial situation is s_0 and Φ is the transition function that specifies the transition between states due to an action then finding out if f **after** $a_1, \ldots a_n$ is true involves computing the state $\Phi(\Phi(\ldots \Phi(s_0, a_1) \ldots a_{n-1}), a_n)$ and evaluating if f is true in it. Now suppose the actions $a_1, \ldots, a_n$ are the actions that agent A is planning to take. But it knows that in response to its actions another agent (or several other agents) will take some action, or a particular series of actions. In that case just computing $\Phi(\Phi(\ldots\Phi(s_0, a_1)\ldots a_{n-1}), a_n)$ is not enough. Agent A needs to reason about the evolution of the world more thoroughly. Similarly, if $a_1, \ldots, a_n$ are each individual database transactions then reasoning about the impact of executing this series of transactions on an initial database also involves reasoning about what actions are triggered by these transactions and their impact. In the following section we describe a way to constrain the evolution of the world so as to take into account triggers, and reactions of other agents in a multi-agent world.

2 Constraining the evolution of the world: additional specifications

As mentioned in the previous section, our goal in this paper is to be able to specify or constrain the evolution of the world beyond just specifying what valid states are and how states transition due to actions. In this we propose to use linear temporal logic with future operators and operators denoting action occurrences. We now formally define this logic.

Formulas of LTL_A (denoting linear temporal logic with action occurrences) are built from the set of propositional symbols, the boolean operators, unary temporal operators $\bigcirc$ (denoting 'next'), $\Diamond$ (denoting 'eventually'), $\Box$ (denoting 'always'), **occurs** (denoting action occurrences), **occurs-only** , and the binary temporal operator $\cup$ (denoting 'until'). Intuitively, the difference between '**occurs-only** a' and '**occurs** a' is that in the former the only action that can occur is a, while in the later a is one of the actions that occur. We now show how LTL_A is intended to be used in constraining the evolution of the world.

1. Suppose we would like to specify that a state where f is true triggers the occurrence of action a. This can be expressed in LTL_A as

   $$\Box(f \Rightarrow \textbf{occurs-only } a)$$

   if no other actions are allowed to execute in parallel with a. If other actions are allowed to be executed in parallel with a then the correct way to express it is

   $$\Box(f \Rightarrow \textbf{occurs } a)$$

2. Similarly if we want to specify that action a_1 triggers action a_2 to be the next action then this can be expressed in LTL_A as

   $$\Box(\textbf{occurs } a_1 \Rightarrow \bigcirc \textbf{ occurs-only } a_2)$$

3. Similarly if we want to specify that the occurrence of action a_1, leading to a state where f is true makes another agent to execute the action a_2 then this can be expressed in LTL_A as

   $$\Box(\textbf{occurs } a_1 \wedge \bigcirc f \Rightarrow \bigcirc \textbf{ occurs } a_2)$$

The semantics of LTL_A is given with respect to trajectories. Intuitively, a trajectory describes an evolution of the world states due to actions. More formally, a trajectory τ is a sequence of states and sets of actions of the form $\tau = s_0 A_1 s_1 \ldots A_i s_i \ldots$, where the set of actions A_i when executed (in parallel) transform the world from the state s_{i-1} to the state s_i. Note that when an A_i is a singleton set $\{a_i\}$ often we simply write a_i instead of $\{a_i\}$.

We start with a definition of truth of an LTL_A formula with respect to a trajectory $\tau = s_0 A_1 s_1 \ldots A_i s_i \ldots$ and some reference state s_j. In the following p denotes propositional formulas, and f_is denote LTL_A formulas.

- $\langle s_j, \tau\rangle \models p$ iff p holds in s_j.
- $\langle s_j, \tau\rangle \models$ **occurs-only** a iff $A_{j+1} = \{a\}$.
- $\langle s_j, \tau\rangle \models$ **occurs** a iff $a \in A_{j+1}$.
- $\langle s_j, \tau\rangle \models \neg f$ iff $\langle s_j, \tau\rangle \not\models f$.
- $\langle s_j, \tau\rangle \models f_1 \wedge f_2$ iff $\langle s_j, \tau\rangle \models f_1$ and $\langle s_j, \tau\rangle \models f_2$.
- $\langle s_j, \tau\rangle \models f_1 \vee f_2$ iff $\langle s_j, \tau\rangle \models f_1$ or $\langle s_j, \tau\rangle \models f_2$.
- $\langle s_j, \tau\rangle \models \bigcirc f$ iff $\langle s_{j+1}, \tau\rangle \models f$.
- $\langle s_j, \tau\rangle \models \Box f$ iff $\langle s_k, \tau\rangle \models f$ for every $k \geq j$.
- $\langle s_j, \tau\rangle \models \Diamond f$ iff $\langle s_k, \tau\rangle$ for some $k \geq j$.
- $\langle s_j, \tau\rangle \models f_1 \cup f_2$ iff there exists $k \geq j$, such that $\langle s_k, \tau\rangle \models f_2$ and for all i, $j \leq i < k$ we have $\langle s_i, \tau\rangle \models f_1$.

We say that a formula f is true with respect to a trajectory (or holds in a trajectory) τ, written as $\tau \models f$, iff $\langle s_0, \tau\rangle \models f$.

3 Representing and reasoning about actions

In this section we show how the additional specifications about the evolution of the world, as described in the previous section, can be integrated with the other components of representing and reasoning about actions. We start with the specification of the evolution of the world.

3.1 Evolution Specification

For this part the language has three subparts:

- Specifying valid states and transitions.
- Specifying observations.
- Constraints on the evolution (or simply 'evolution constraints').

For specifying valid states and transitions we can use syntactic constructs such as:

- a **causes** f **if** $p_1, \ldots, p_n$ or the more general a **causes** ψ **if** φ;
- **executable** a **if** $q_1, \ldots, q_m$ or the more general **executable** a **if** φ; and
- $q_1, \ldots, q_m$ **causes** p or the more general θ **causes** ψ,

where as are actions (or sets of actions) p_is and q_i are fluent literals and φ and ψ are fluent formulas. Given a description consisting of statements of the above syntactic form, its semantics is defined by inferring (from the description) a transition function $\Phi : States \times 2^{actions} \rightarrow 2^{States}$. Since there are many papers (for example, [Gelfond and Lifschitz, 1993; Baral *et al.*, 1997; McCain and Turner, 1995]) which present what the transition function corresponding to a given description is, we do not give details of that aspect here. Note that some of these papers have slightly different syntax and slightly different semantics, but that is of not much consequence here. It is sufficient for us to assume that there is a way to characterize descriptions using a transition function, and in fact usually a description is a succinct way to specify the corresponding transition function.

Now let us consider the observations and the evolution constraints. We consider the observation language from [Baral *et al.*, 1997] where the observations are of the following form:

- t_1 **precedes** t_2;
- p **at** t_1; and
- α **occurs_at** t_1,

where t_is are situations (or time points), α is an action sequence and p is a fluent literal.

The evolution constraints are LTL_A formulas.

Now given a description D, a set of observations O and evolution constraints E, the evolution of the world is defined through a pair (τ, μ), where τ is a trajectory depicting how the world could evolve and μ is a mapping from situations (or time points) to points in the trajectory. For example, if the observations only refer to the situations $t_1, \ldots, t_4$, then we may have a (τ_1, μ_1), where $\tau_1 = s_0, A_1, s_1, A_2, \ldots, A_n, s_n$ and μ_1 is a mapping from $\{t_0, t_1, t_2, t_3, t_4\}$ to $\{0, \ldots, n\}$.

Definition 1 (trajectory interpretation) Given a description D, a set of observations O and evolution constraints E, a pair (τ, μ), with $\tau = s_0, A_1, s_1, A_2, \ldots, A_n, s_n$ is said to be a trajectory interpretation of (D, O, E) if

- $s_i \in \Phi(s_{i-1}, A_i)$, where Φ is the transition function corresponding to D,
- $\mu(t_0) = 0$;
- if t_1 **precedes** t_2 is in O then $\mu(t_1) \leq \mu(t_2)$;
- if p **at** t_1 is in O then p holds in $s_{\mu(t_1)}$;
- if $B_1, \ldots, B_k$ **occurs_at** t_1, then for $1 \leq i \leq k$, $B_i \subseteq A_{\mu(t_1)+i}$; and
- the formulas in E hold in the extended infinite trajectory $\hat{\tau} = s_0, A_1, s_1, A_2, \ldots, A_n, s_n, \emptyset, s_n, \emptyset, \ldots$.

In the above definition we define the notion of a trajectory interpretation. To define trajectory models, which express how the world could evolve, we incorporate the assumption that *no action occurs unless it is dictated by the observations or the evolution constraints.* We refer to this assumption as *No Superfluous Action Occurrence (NSAO) assumption.* The following example illustrates this point.

Example 1 Consider a domain with two fluents p and q. There are four possible states $s_0 = \{\}$, $s_1 = \{p\}$, $s_2 = \{q\}$, and $s_3 = \{p, q\}$. Suppose we have actions a_1 and a_2 and have descriptions $D_1 = \{a_1$ **causes** p, a_2 **causes** $q\}$. Now, suppose we have the observations $O_1 = \{\neg p$ **at** $t_0, \neg q$ **at** $t_0\}$ and the evolution constraint $E_1 = \{\neg p \wedge \neg q \Rightarrow$ **occurs** $a_1\}$. Now let us consider the various trajectory interpretations of (D_1, O_1, E_1). Let $\mu(t_0) = 0$. Let $\tau_0 = s_0$, $\tau_1 = s_0, a_1, s_1$, $\tau_2 = s_0, a_1, s_1, a_2, s_3$, $\tau_3 = s_0, \{a_1, a_2\}, s_3$. It is easy to see that (τ_1, μ), (τ_2, μ), and (τ_3, μ) are all trajectory interpretations of (D_1, O_1, E_1). But in τ_2 we have the action a_2 occurring in s_1 without being dictated or forced by O_1 or E_1. Similarly, in τ_3 we have the action a_2 occurring in s_0 without being dictated or forced by O_1 or E_1. Thus they both violate our non superfluous action occurrence assumption that 'no action occurs unless it is dictated by the observations or the evolution constraints.' On the other hand (τ_0, μ) is not a trajectory interpretation of (D_1, O_1, E_1), as the formula in E does not hold in $\hat{\tau_0} = s_0, \emptyset, s_0, \emptyset, \ldots$.

But (τ_1, μ) is a trajectory interpretation of (D_1, O_1, E_1) and at the same time it seem to satisfy the No Superfluous Action Occurrence (NSAO) assumption. Hence, (τ_1, μ) elucidates how the world evolves. □

In the above example we illustrated the role of the No Superfluous Action Occurrence (NSAO) assumption. We now incorporate this assumption to define trajectory models. But first we need the following definition to compare trajectories. In this definition and later for a trajectory $\tau = s_0, A_1, s_1, \ldots, A_n, s_n$, by $\alpha(\tau)$ we denote the sequence of action sets $A_1, \ldots, A_n$ of τ.

Definition 2 Let $\tau = s_0, A_1, s_1, \ldots, A_n, s_n$ and $\tau' = s'_0, B_1, s'_1, \ldots, B_m, s'_m$ be trajectories, such that $s_0 = s'_0$. We say that $\alpha(\tau) \leq \alpha(\tau')$ if there exists a subsequence[1] $[B_{i_1}, \ldots, B_{i_n}]$ of $\alpha(\tau')$ such that for every $A_k \in \alpha(\tau)$, $A_k \subseteq B_{i_k}$. □

Definition 3 (trajectory model) Given a description D, a set of observations O and evolution constraints E, a pair (τ, μ), with $\tau = s_0, A_1, s_1, a_2, \ldots, A_n, s_n$ is said to be a trajectory model of (D, O, E) if

- (τ, μ) is a trajectory interpretation of (D, O, E), and
- there does not exist a trajectory interpretation (τ', μ) of (D, O, E) such that $\alpha(\tau') < \alpha(\tau)$. □

Note that in this section we have limited the notion of trajectory interpretations and trajectory models to finite trajectories. In presence of evolution constraints that force continuous triggering there may not exist any trajectory models that are finite. We do not consider such cases in this paper and will consider them in the sequel.

3.2 Queries

In this section we present the query language and define entailment of queries from triplets (D, O, E). To motivate the query language that we propose, let us first recall the query language in $\mathcal{A}$ [Gelfond and Lifschitz, 1993] and $\mathcal{L}$ [Baral *et al.*, 1997]. In $\mathcal{A}$ queries are of the form p **after** $a_1, \ldots, a_n$ which intuitively asks whether a fluent literal p will become true if the sequence of actions $a_1, \ldots, a_n$ were to be executed in the initial situation. In $\mathcal{L}$, queries are of the form p **after** $a_1, \ldots, a_n$ **at** t which intuitively asks whether a fluent literal p will become true if the sequence of actions $a_1, \ldots, a_n$ were to be executed in the situation t. In $\mathcal{A}$ and $\mathcal{L}$ $a_1, \ldots, a_n$ can be thought of as actions that the agent is planning to execute and the agent wants to find out if f will be true

[1] Given a sequence $X = x_1, \ldots, x_m$, another sequence $Z = z_1, \ldots, z_n$ is a subsequence of X if there exists a strictly increasing sequence $i_1, \ldots, i_n$ of indices of X such that for all $j = 1, 2 \ldots, n$, we have $x_{i_j} = z_j$.

after that or not. Since we allow evolution constraints that can express triggers if the agent were to execute (or observe) the actions $a_1, \ldots, a_n$ it is possible that other actions may be triggered between a_i and a_{i+1}. Thus we no longer have a sequence of actions $a_1, \ldots, a_n$, but rather have a sampling of actions. We denote it by $a_1; a_2; \ldots; a_n$. The ';' between a_i and a_{i+1} indicates that there may be other actions that occur in between them. Also, instead of just fluent literals, we may ask about the trajectory starting from when a_1 is executed.

Thus in our query language a query is of the form f **during** $A_1; A_2; \ldots; A_m$ **at** t where f is an LTL formula[2], and t either appears in E, or is t_0 or is the current situation t_C. We now define the entailment between (D, O, E) and the above mentioned queries. Intuitively, we first compute the trajectory models of (D, O, E). For each trajectory model we find the state corresponding to t and then create a new (D, O', E) where the O' encodes what is true in the state corresponding to t as the fluent literals that hold in the initial situation. In addition O' encodes the occurrence of $A_1; \ldots; A_n$. f is then evaluated with respect to the trajectory models of (D, O', E). More formally, given (D, O, E) we say $(D, O, E) \models f$ **during** $A_1; A_2; \ldots; A_m$ **at** t if f holds in all trajectory models of the theories (D, O', E) constructed as follows:

- Let (τ, μ) be a trajectory model of (D, O, E), where $\tau = s_0, A_1, s_1, \ldots, A_n, s_n$.
- If $t = t_0$ then we assign k as 0; else if $t = t_C$ then we assign the value n to k; otherwise k is assigned the value $\mu(t)$.
- O' now consists of the following.
  - For all fluents p, if $p \in s_k$ then O' has p **at** t_0, otherwise it has $\neg p$ **at** t_0.
  - In addition the following are part of O'. (Here, $\{a_1, a_2, \ldots, a_l\}$ **occurs_at** t is a shorthand for the set $\{a_1$ **occurs_at** $t, \ldots, a_n$ **occurs_at** $t\}$.)
    A_1 **occurs_at** t_1,
    A_2 **occurs_at** $t_2, \ldots,$
    A_m **occurs_at** t_m,
    t_1 **precedes** t_2,
    t_2 **precedes** $t_3, \ldots,$
    t_{m-1} **precedes** t_m.

3.3 An example

Let us consider a domain with three fluents p, q and r. Then there are 8 possible states, which we refer as follows: $s_0 = \emptyset$, $s_1 = \{p\}$, $s_2 = \{q\}$, $s_3 = \{r\}$, $s_4 = \{p, q\}$, $s_5 = \{p, r\}$, $s_6 = \{q, r\}$, and $s_7 = \{p, q, r\}$. We have four actions a_1, a_2, a_3 and a_4 whose effects are described by

$D_2 = \{a_1$ **causes** p,
a_1 **causes** q,
a_2 **causes** r,
a_3 **causes** $\neg p$,
a_4 **causes** $\neg q\}$.

The evolution constraints are given by

$E_2 = \{\Box(q \Rightarrow$ **occurs** $a_2)$,
$\Box(p \Rightarrow$ **occurs** $a_3)$,
$\Box(\neg p \wedge q \wedge r \Rightarrow$ **occurs** $a_4)\}$.

The observations that we have are
$O_2 = \{\neg p$ **at** $t_0, \neg q$ **at** $t_0, \neg r$ **at** $t_0\}$.

Suppose our query is $q_2 = \Diamond\Box(\neg p \wedge \neg q \wedge r)$ **during** a_1. We now show that $(D_2, O_2, E_2) \models q_2$.

It is easy to see that the only trajectory model of (D_2, O_2, E_2) is (τ_2, μ_2), where $\tau_2 = s_0$, and $\mu_2(t_0) = s_0$. Now let us construct (D_2, O_2', E_2).

$O_2' = \{\neg p$ **at** $t_0, \neg q$ **at** $t_0, \neg r$ **at** t_0, a_1 **occurs_at** $t_1\}$.

The only trajectory model of (D_2, O_2', E_2) is (τ_2', μ_2'), where $\tau_2' = s_0, a_1, s_4, \{a_2, a_3\}, s_6, \{a_2, a_4\}, s_3$, and $\mu_2'(t_0) = s_0$, and $\mu_2'(t_1) = s_0$. This is because:

- $s_0 = \emptyset$, and we have $\neg p$ **at** $t_0, \neg q$ **at** $t_0, \neg r$ **at** t_0 in O_2'.
- Since we have a_1 **occurs_at** t_1 in O_2', and no other actions are dictated to be in between t_1 and t_2, we have a_1 following s_0 in τ_2'.
- Since we have a_1 **causes** p and a_1 **causes** q in D_2, the state following s_0, a_1 is $\{p, q\} = s_4$.
- Since we have $\Box(q \Rightarrow$ **occurs** $a_2)$, and $\Box(p \Rightarrow$ **occurs** $a_3)$ in E_2, the set of actions $\{a_2, a_3\}$ follow s_4 in τ_2'.
- Since we have a_2 **causes** r and a_3 **causes** $\neg p$ in D_2, the state following $s_4, \{a_2, a_3\}$ is $\{q, r\} = s_6$.
- Since we have $\Box(\neg p \wedge q \wedge r \Rightarrow$ **occurs** $a_4)$, and $\Box(q \Rightarrow$ **occurs** $a_2)$ in E_2, the set of actions $\{a_2, a_4\}$ follows s_6 in τ_2'.
- Since we have a_4 **causes** $\neg q$ in D_2, the state following s_6, a_4 is $\{r\} = s_3$.
- Since neither O_2' nor E_2's dictate any action to occur at s_3, by minimality we have $\tau_2' = s_0, a_1, s_4, \{a_2, a_3\}, s_6, a_4, s_3$.

Now to evaluate the goal part of the query q, which is $\Diamond\Box(\neg p \wedge \neg q \wedge r)$, we consider the extension of τ_2', which is $\hat{\tau_2'} = s_0, a_1, s_4, \{a_2, a_3\}, s_6, \{a_2, a_4\}, s_3, \epsilon, s_3, \epsilon, s_3, \ldots$. It is easy to see that $\Diamond\Box(\neg p \wedge \neg q \wedge r)$ holds in $\hat{\tau_2'}$. Hence, $(D_2, O_2, E_2) \models q_2$.

4 Augmenting with probabilities: representing strategies

In the earlier sections we discussed the incorporation of evolution constraints in reasoning about actions and its use in expressing triggers. Our next goal is to be able to express probabilistic triggers and randomized strategies. For example, we are interested in expressing that whenever p is true then with probability 0.7 the next action is a. Our motivation in representing and reasoning about such occurrences initially came from the papers [Poole, 1997] and [Pearl, 1999; 2000]. In [Pearl, 2000] there is a statement that says 'the probability of occurrence of treatment is 0.5. We are interested in being able to represent and reason about such statements. Similarly, in [Poole, 1997], there is a story about a goalie (goal keeper in soccer) and a kicker. The goalie can perform the actions $jump\_right$ (jr – meaning jumping to

[2]An LTL formula is an LTL_A formula which does not have the operators **occurs** and **occurs-only** .

its *own* right), and $jump\_left$ (jl) and the kicker can perform the actions $kick\_right$ (kr – meaning kicking to its *own* right), and $kick\_left$ (kl). The likelihood of scoring a goal is described by the following table:

	$kick\_left$ (kl)	$kick\_right$ (kr)
$jump\_left$ (jl)	0.9	0.15
$jump\_right$ (jr)	0.2	0.95

Using constructs from the action description language PAL [Baral *et al.*, 2002] this can be expressed by having inertial unknown variables u_1, u_2, u_3 and u_4 and having the following D_3: $\{jl, kl\}$ **causes** $goal$ **if** u_1.
$\{jl, kr\}$ **causes** $goal$ **if** u_2.
$\{jr, kl\}$ **causes** $goal$ **if** u_3.
$\{jr, kr\}$ **causes** $goal$ **if** u_4.
jl **causes** $\neg init$.
jr **causes** $\neg init$.
kl **causes** $\neg init$.
kr **causes** $\neg init$.
and the following probability information.
probability of u_1 **is** 0.9.
probability of u_2 **is** 0.15.
probability of u_3 **is** 0.2.
probability of u_4 **is** 0.95.
Let us assume that we have the following observations O_3 which basically says that in the initial situation $init$ is true and $\neg goal$ is true.
$init$ **at** t_0.
$\neg goal$ **at** t_0.
Let us have the following evolution constraints E_3 that says that in any state where $init$ is true one of the actions jl or jr occurs and also one of the actions kl or kr occurs. This can be expressed as follows:
$\Box(init \Rightarrow$ **occurs** $jl \vee$ **occurs** $jr)$
$\Box(init \Rightarrow$ **occurs** $kl \vee$ **occurs** $kr)$
Ignoring the probability information the above (D_3, O_3, E_3) has 64 trajectory models (16 combinations of u_1, u_2, u_3, u_4 and 4 combinations of $\{jl, jr\} \times \{kl, kr\}$), some of which are as follows:

1. $\{init, u_1, u_2, u_3, u_4\}\{jl, kl\}\{u_1, u_2, u_3, u_4, goal\}$
2. $\{init, u_2, u_3, u_4\}\{jl, kl\}\{u_2, u_3, u_4\}$
3. $\{init, u_1, u_2, u_3, u_4\}\{jl, kr\}\{u_1, u_2, u_3, u_4, goal\}$
4. $\{init, u_1, u_3, u_4\}\{jl, kr\}\{u_1, u_3, u_4\}$
5. $\{init, u_1, u_2, u_3, u_4\}\{jr, kl\}\{u_1, u_2, u_3, u_4, goal\}$
6. $\{init, u_1, u_2, u_4\}\{jr, kl\}\{u_1, u_2, u_4\}$
7. $\{init, u_1, u_2, u_3, u_4\}\{jr, kr\}\{u_1, u_2, u_3, u_4, goal\}$
8. $\{init, u_1, u_2, u_3\}\{jr, kr\}\{u_1, u_2, u_3\}$

Now using the probability associated with u_1, u_2, u_3 and u_4 we can compute the probability associated with each of the trajectory models. (Our computation is simplified because as u_1, u_2, u_3 and u_4 are unknown variables and hence are independent of each other. Also, we only have deterministic actions and our trajectories are of finite length.) We can compute the probability of goal as the sum of probabilities of trajectories whose final state has $goal$ in them. In this example, this comes out to $\frac{0.9+0.15+0.2+0.95}{4} = 0.55$. This is because the occurrence of jl and jr are equally likely and similarly the occurrence of kl and kr are also equally likely.

Now let us consider strategies where the goalie decides to jump left 40% of the time (and jump right 60% of the time) and the kicker decides to kick left 70% of the time (and kick right 30% of the time). The first issue is *how to express such strategies*?

Here, we can follow the ideas in [Baral *et al.*, 2002; Pearl, 1999; 2000] and again use unknown variables. Let us have unknown variables u_5 and u_6 and the following probability statements.
probability of u_5 **is** 0.4.
probability of u_6 **is** 0.7.
Now we modify the evolution constraints E_3 so as to incorporate u_4 and u_5 and have the following E_4.
$\Box(init \wedge u_5 \Rightarrow$ **occurs** $jl)$
$\Box(init \wedge \neg u_5 \Rightarrow$ **occurs** $jr)$
$\Box(init \wedge u_6 \Rightarrow$ **occurs** $kl)$
$\Box(init \wedge \neg u_6 \Rightarrow$ **occurs** $kr)$
Now although (D_3, O_3, E_4) will also have 64 trajectory models – as did (D_3, O_3, E_3), the trajectory models will not be exactly the same (because of new unknown variables u_5 and u_6) and the probability associated with them will also be different (because of E_4). Nevertheless, we can use the same method of summing the probabilities of trajectories whose final state has $goal$ in them, to compute the probability of $goal$. In this case it will be $(0.9 \times 0.4 \times 0.7 + 0.15 \times 0.4 \times 0.3 + 0.2 \times 0.6 \times 0.7 + 0.95 \times 0.6 \times 0.3) = 0.525$.

Above we have illustrated *through examples* how to represent 'strategies' (or triggered probabilistic occurrence of actions) and reason about them. The main idea is to use unknown variables (with associated probabilities) in evolution constraints. In addition we have an important assumption that the transition functions are *deterministic*. This allows us to focus on the initial state of a trajectory to compute the probability of that trajectory. But because of the observations we may have multiple trajectories with the same initial state. In that case we consider each such trajectory to be equally probable and compute the individual probability of a trajectory as the probability of its initial state divided by the number of trajectories with that same initial state. One drawback of our assumption that transition functions are deterministic is that it rules out non-inertial unknown variables that are used in PAL [Baral *et al.*, 2002]. Nevertheless, we still can represent and reason about many strategies. In this context it must be mentioned that in [Pearl, 1999] the unknown variables are all non-inertial.

5 Conclusion, related work and future work

In this paper we have shown how to incorporate evolution constraints to reason about the evolution of the world and how this allows us to express triggering of one action by another, and reason about actions of multiple agents following particular strategies. Since this is one of the initial attempt in this direction we have used a very general language (LTL_A) to express evolution constraints. Further study is necessary to decide whether this is a good idea or not, and whether an alternative where a very small subclass of LTL_A is used is better. For the later we need to identify important kinds of evolution constraints and have specific syntax for them. Overall, the main technical contribution of this paper are: (i) we introduce evolution constraints, (ii) we present a notion of minimization of trajectories that allows us to incorporate the NSAO (no superfluous action occurrence) assumption in presence of evolution constraints, (iii) we consider queries that allow gaps in between actions in the plan part of the query, and (iv) we show how to represent and reason about triggers, probabilistic occurrences of actions and probabilistic strategies.

In terms of related work this paper extends (and suggests extension to) the $\mathcal{A}$ [Gelfond and Lifschitz, 1993] class of high level action description languages such as $\mathcal{L}$ [Baral *et al.*, 1997] and PAL [Baral *et al.*, 2002]. The use of LTL_A to expressing evolution constraints is yet another use of linear temporal logic in reasoning about actions and change. This raises the following question: Should we forgo action description languages such as $\mathcal{A}$ completely and switch to temporal logics such as LTL or are there aspects where one is preferable to the other? In the past LTL has been used in expressing queries, in expressing the condition part ϕ of actions descriptions such as a **causes** f **if** ϕ, and in expressing the transition function Φ itself. The last one was recently proposed in [Calvanese *et al.*, 2002]. It is not clear if the last use is appropriate or not as [Calvanese *et al.*, 2002] shows that expressing executability conditions of actions is problematic in their temporal logic and the specification is not succinct. Also, since standard temporal logics use classical connectives it is difficult to express causality and static constraints in them. Thus, a preliminary answer to the above questions is that temporal logic is good for representing queries, and evolution constraints (together with minimality), but perhaps not for expressing transition functions and causality. For that the constructs from $\mathcal{A}$ and its successors are perhaps preferable. Another related work is the paper [Reiter, 1996] (and the Toronto situation calculus) which considers natural actions that occur at distinct times. But as mentioned in [Reiter, 1996] these approaches have the problem of premature minimization and hence have difficulty in expressing the NSAO assumption.

In terms of future work in the sequel we will consider infinite trajectories. Here we in essence assumed that the evolution constraints are such that trajectories are finite. We also plan to consider non-deterministic actions and presence of non-inertial variables when dealing with probabilities. Finally, we need to formalize the representation and reasoning about strategies which we only illustrated (due to lack of space) through an example in Section 4.

References

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

[Baral *et al.*, 2001] C. Baral, V. Kreinovich, and R. Trejo. Computational complexity of planning with temporal goals. In *IJCAI 2001*, 2001.

[Baral *et al.*, 2002] C. Baral, N. Tran, and L. Tuan. Reasoning about actions in a probabilistic setting. In *Proc. of AAAI'2002*, pages 507–512, 2002.

[Calvanese *et al.*, 2002] D. Calvanese, G. De Giacomo, and M. Vardi. Reasoning about actions and planning in ltl action theories. In *KR'02*, 2002.

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

[Georgeff, 1994] M. Georgeff, editor. *Journal of Logic and Computation, Special issue on Action and Processes*, volume 4 (5). Oxford University Press, October 1994.

[Hancock, 1997] J. Hancock. *Cell Signalling*. Longman, 1997.

[Levesque, 1996] H. Levesque. What is planning in the presence of sensing? In *AAAI 96*, pages 1139–1146, 1996.

[Lifschitz, 1997] V. Lifschitz, editor. *Special issue of the Journal of Logic Programming on Reasoning about actions and change*, volume 31(1-3), May 1997.

[McCain and Turner, 1995] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In *Proc. of IJCAI 95*, pages 1978–1984, 1995.

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

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

[Poole, 1997] 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.

[Reiter, 1996] R. Reiter. Natural actions, concurrency and continuous time in the situation calculus. In L. Aiello, J. Doyle, and S. Shapiro, editors, *KR 96*, pages 2–13, 1996.

[Sandewall, 1998] E. Sandewall. Special issue. *Electronic Transactions on Artificial Intelligence*, 2(3-4):159–330, 1998. http://www.ep.liu.se/ej/etai/.

[Son and Baral, 2001] T. Son and C. Baral. Formalizing sensing actions: a transition function based approach. *AI journal*, 125(1-2):19–93, 2001.

[Widom and Ceri, 1996] J. Widom and S Ceri, editors. *Active Database Systems - Triggers and Rules for advanced database processing*. Morgan Kaufmann, 1996.