

# Reasoning over Time

---

Reasoning about the past predicting the future.

## 1. Introduction

We live in a temporal dimension, our environment changes as times goes by. Up to now, we deal only within categories, substances and objects, providing us a static perception; we would like to represent a system that **evolves** along the temporal dimension.

We might need to take in account also a **dynamic dimension**, related to the passing time.

There exist two main approaches to describe the dynamical evolution:

- **Event Calculus.**
- **Allen's Temporal Logic.**

## 2. Propositional logic

The main idea is to use **propositional logic** to represent an agent's beliefs about the world. Introducing this kind of logic, it's possible to describe the current state of the world as set of **propositions**. Inside the set should appear only true propositions, thus representing what the agent believes to be true.

Different sets of propositions, put in some order, would represent the world evolution perceived by an agent.

### Example

A child has a bow and an arrow. He can shoot the arrow, throw away the bow, but he can also run, hide and crouch. Our agent observes the child, and it believes that:

$$KB^0 = \text{hasBow}^0, \text{hasArrow}^0$$

The child shoots the arrow, our agent observes the child and it belivies:

$$KB^1 = \text{hasBow}^0, \text{hasArrow}^0, \text{hasBow}^1, \neg \text{hasArrow}^1$$

Some observations are already coming out:

- 1<sup>st</sup>. The knowledge base at the time instant 0 still available for the knowledge base at time instant 1.
- 2<sup>nd</sup>. If we take, any time, all the knowledge of the previous steps, our agent can easily enter in an infinite search space.
- 3<sup>rd</sup>. We take this last assumption for granted, even though it is not a feasible observation in a real world application (we don't have the capability to store so much informations).

---

What's really interesting is given by the dynamic evolution that bring us from the  $KB^0$  to the  $KB^1$ . *What happened in between?*

There is a general concept about **action**, whose execution changes the world. We would like to describe which are the **effects** of the executed actions.

Additionally, there is a notion of **state** that captures the world at a certain instant. The evolution of the world is given by a sequence of states.

Given the clue about **action**, we need to define a formalism about its effects that could change the world, also called **effect axioms**.

## Example

What does it mean *shooting an arrow*?

$$\text{shoot}^t \rightarrow (\text{hasArrow}^t \Leftrightarrow \neg \text{hasArrow}^{(t+1)})$$

If the action **shoot** is executed at time instant  $t$ , this implies that at time instant  $t$  our agent has the arrow and at next time instant  $(t + 1)$  it will not have anymore the arrow.

$$\text{KB}^0 = \text{hasArrow}^0$$

$$\text{KB}^1 = \text{hasArrow}^0, \neg \text{hasArrow}^1$$

However, we are missing something. *Does the child still have the bow?* We don't know nothing about what happened to the bow at time instant 0.

From the previous observation, we formalize the **frame problem**: everything is untouched from the execution of an action should be moved to the next state. The effect axioms fail to state what remains unchanged after the execution of some actions and this approach is not possible; we have to describe an axiom for every unchanged state.

A solution would be the **frame axioms**, for each proposition not affected by the execution of an action, we will state that it is unchanged.

By the way, neither with frame axioms we can reach optimal solutions. Assuming  $m$  actions and  $n$  propositions, the set of frame axioms will be  $O(m \times n)$ .

## 3. Situation Calculus

Up to now, we focus on actions: a general concept used to express changes of the current environment. However, changing the viewpoint, focusing on the propositions that describe the world, we obtain a totally different solution.

In this case, we don't talk anymore about propositions, but instead about **fluents** (a special construct that can be true or false along the temporal dimension).

### Successor State Axioms

Each state is described by a set of fluents  $F$ . Then, we define the following axiom:

$$F^{t+1} \Leftrightarrow \text{ActionCauses}F^t \vee (F^t \wedge \neg \text{ActionCausesNot}F^t)$$

The set of fluents  $F$  holds at time instant  $(t + 1)$  if and only if:

- The action  $\text{ActionCauses}F^t$  that causes the set of fluents  $F$  was performed.
- The set of fluents  $F$  is already true at time instant  $t$  and no action performed change it.

### Example

$$\text{hasBow}^{(t+1)} \Leftrightarrow \text{pickUpBow}^t \vee (\text{hasBow}^t \wedge \neg \text{throwBow}^t)$$

Our agent has the bow if and only it performed the `pickUpBow` action or it already had the bow and did not throw away it.

**Situation calculus** has three different key features:

- 1<sup>st</sup>. The initial state is called **situation**.
- 2<sup>nd</sup>. Given an action  $a$  and a situation  $s$ , then  $\text{Result}(s, a)$  is a situation.
- 3<sup>rd</sup>. A function that changes from a situation to the next one is called **fluent**.

It introduces the notion of **preconditions** of an action: the preconditions are the properties which must hold before the execution of an action. Action's preconditions are defined by the **possibility axioms**:

$$\Phi(s) \rightarrow \text{Poss}(a, s)$$

Situation calculus adopts the **Successor State Axioms** following the possibility notion:

$$\text{Poss}(a, s) \rightarrow (F(\text{Result}(a, s)) \Leftrightarrow a = \text{ActionCauses}F \vee (F(s) \wedge a \neq \text{ActionCausesNot}F))$$

### Example

$$\text{Poss}(a, s) \rightarrow (F(\text{Result}(a, s)) \Leftrightarrow a = \text{shoot} \vee (\neg \text{hasArrow}(s) \wedge a \neq \text{reload}))$$

Despite the fact that through situation calculus we are able to determine what is true before and after the execution of an action, we cannot say anything about what is true during the action.

## 4. Event Calculus

Sergot and Kowalsky, in 1986, came out with **event calculus**, based on logics used to describe the evolution of a system. Without any distinction, all the fluents and events are treated like **terms**; the predicate

$$\text{HoldsAt}(\text{fluent}, \text{situation/action})$$

gives us the truth value of the fluent in any situation or action.

The formulation of event calculus is based on a fixed ontology and two different set of axioms:

1. Event calculus ontology.
2. Domain-independent axioms.
3. Domain-dependent axioms.

The **event calculus ontology** has six fixed predicates, which are:

- 1<sup>st</sup>  $\text{Holds}(F, T)$ . Fluents  $F$  holds at time instant  $T$ .
- 2<sup>nd</sup>  $\text{Happens}(E, T)$ . Event  $E$  happened at time instant  $T$ .
- 3<sup>rd</sup>  $\text{Initiates}(E, F, T)$ . Event  $E$  causes fluent  $F$  to hold at time instant  $T$ .
- 4<sup>th</sup>  $\text{Terminates}(E, F, T)$ . Event  $E$  causes fluent  $F$  to cease to hold at time instant  $T$ .
- 5<sup>th</sup>  $\text{Clipped}(T_1, F, T)$ . Fluent  $F$  has been made false between  $T_1$  and  $T$ .
- 6<sup>th</sup>  $\text{Initially}(F)$ . Fluent  $F$  holds at time instant 0.

Instead, the domain-independent axioms are:

- 1<sup>st</sup> Used to detect when a fluent is true.

$$\text{HoldsAt}(F, T) \leftarrow \text{Happens}(E, T_1) \wedge \text{Initiates}(E, F, T_1) \wedge (T_1 < T) \wedge \neg \text{Clipped}(T_1, F, T)$$

The fluent  $F$  is true at time instant  $T$  if happened the event  $E$  at time instant  $T_1$ , event  $E$  causes fluent  $F$  at time instant  $T_1$ , time instant  $T_1$  comes after time instant  $T$  and no one has changed the fluent  $F$  between  $T$  and  $T_1$ .

- 2<sup>nd</sup> Used to detect when a fluent is true.

$$\text{HoldsAt}(F, T) \leftarrow \text{Initially}(F) \wedge \neg \text{Clipped}(0, F, T)$$

The fluent  $F$  is true at time instant  $T$  if it is verified at the initial state and no one has changed the fluent  $F$  between time instant 0 and  $T$ .

- 3<sup>rd</sup> Used to define the clipping of a fluent.

$$\text{Clipped}(T_1, F, T_2) \leftarrow \text{Happens}(E, T) \wedge (T_1 < T < T_2) \wedge \text{Terminates}(E, F, T)$$

The fluent  $F$  has been made false between  $T_1$  and  $T_2$  if happened the event  $E$  at time instant  $T$ , time instant  $T$  comes after  $T_1$  and before  $T_2$  and event  $E$  causes fluent  $F$  to cease to hold at time instant  $T$ .

All the reasoning about **event calculus** is made by these three rules. Finally, the domain-dependent axioms are divided into:

- 1<sup>st</sup>  $\text{Initially}(F)$ . The fluent  $F$  holds at the beginning.
- 2<sup>nd</sup>  $\text{Initiates}(E, F, T)$ . The execution of the event  $E$  at time instant  $T$  makes the fluent  $F$  true.
- 3<sup>rd</sup>  $\text{Terminates}(E, F, T)$ . The execution of the event  $E$  at time instant  $T$  makes the fluent  $F$  false.

While situation calculus relies on deductive planning, event calculus is based on reactive rules, allowing us to represent the state of a system in logical terms, in particular in First Order Logic.

Despite we can easily implement event calculus by **First Order Logic**, it is not safe if a term contains variables and, also, another big problem is given by the representation of predicate  $\neg \text{Clipped}(\dots)$  by **negation as failure**.

## 5. Allen's Logic for reasoning over temporal aspects

Event calculus is based on the notion of event or, in other words, on the happening of things. *Are these events instantaneous or do they have a duration?*

Several observations could answer this query, but what is really true is the fact that do not exist any instantaneous action in a real world application, each of them has a duration.

Instead of thinking about **point in times**, we have to reason about **intervals** where properties are true or false. This is the basic idea about **Allen's logic**.

In Allen's view, intervals are more natural categories for humans, simpler to define when some properties or fluents hold (intervals as first entities).

Given an interval  $i$ , we know it has a beginning and an end; we can retrieve such informations by two different actions:

- **Begin( $i$ )**, that returns the initial time point.
- **Ends( $i$ )**, that returns the final time point.

Allen's logic is a powerful tool that allows us to describe the evolution of a system in terms of intervals. However, logic becomes easily undecidable when we have to deal with more complex entities, such as conjunction and disjunction of intervals.