32002: AI (Situation Calculus) **Spring 2014**

Lecture 20: February 20, 2014

Lecturer: K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

20.1 Situation Calculus

Example 20.1 *Situation-action*

The fluent $is\_carrying(o, s)$ can be used to indicate that the robot is carrying a particular object 'o' in a particular situation 's'. If the robot initially carries nothing, $is\_carrying(Ball, S_0)$ is false while, the fluent

$$is\_carrying(Ball, do(pickup(Ball), S_0))$$

is true. The location of the robot can be modeled using a functional fluent location(s) which returns the location (x, y) of the robot in a particular situation. □

Consider the blocks world where some blocks of equal size can be arranged into a set of towers on a table, using the robot hand, which can be directed to grasp any block that is on the top of a tower, and either add it to the top of another tower or put it down on the table to make a new tower.Following actions:

- $stack(x, y)$ - put block x on block y, provided the robot is holding x, and y top is clear. Being action, it is read "stack x on y", and shall not be confused with predicate.
- $unstack(x, y)$ - pick up block x from block y, provided the robot's hand is empty, x is on y, and x has top clear.
- $putdown(x)$ - put block x down on the table, provided the robot is holding x.
- $pickup(x)$ - pick up block x from the table, provided the robot's hand is empty, x is on the table and top clear.

To describe the effects of these actions, we can use the following relational fluents:

- $handempty$ - True in a situation if the robot's hand is empty.
- $holding(x)$ - True in a situation if the robot's hand is holding the block x.
- $on(x, y)$ - True in a situation if block x is on block y.
- $ontable(x)$ - True in a situation if block x is on the table.
- $clear(x)$ - True in a situation if block x is the top block of a tower.

For action $stack(x, y)$ to be performed in a situation, the fluent,

- $holding(x)$ must be true, and
- $clear(y)$ must be true.

Also, after $stack(x, y)$ is performed and it results to a new situation, the fluent,

- $on(x, y)$ will be true,
- $handempty$ will be true,
- $holding(x)$ will be false, and
- $clear(y)$ will be false.

The above action, fluents, and resulting action with fluents are formally expressed as,

$$\forall x \forall y[((holding(x) \wedge clear(y)) \rightarrow (stack(x, y) \\ \wedge\, handempty \wedge \neg holding(x) \neg clear(y))]$$

20.1.1 Formalizing the notions of context

Whenever we write an axiom, a critic can say the axiom is true only in a certain context. Consider axiomatizing "on" so as to draw appropriate consequences from the information expressed in the sentence, in a world of space-craft in a moon mission:

"The book is on the table."

The critic may propose to haggle about the precise meaning of 'on' causing difficulties about what can be between the book and the table or about how much gravity there has to be in a spacecraft in order to use the word "on" and whether centrifugal force counts. Thus, we encounter Socratic puzzles over what the concepts mean in complete generality and encounter examples that never arise in life. There simply is not a most general context !

A possible way out involves formalizing the notion of context is to add a context parameter to the functions and predicates in our axioms. There are a small number of predicates about contexts as well as dealing with time. One of the most important predicate is *holds*, which asserts that a property holds (i.e., is true) during a time interval. Thus

$$holds(p, t)$$

is true if and only if property p holds during t. As a subsequent axiom will state, this is intended to mean that p holds at every subinterval of t as well. Note that if we had introduced *holds* as a modal operator we would not need to introduce properties into our ontology.

Suppose that, whenever a sentence p is present in the memory of a computer, we consider it as in a particular context and as an abbreviation for the sentence

$$holds(p, C),$$

where C is the name of a context. There is a relation $c_1 \leq c_2$ meaning that context c_2 is more general than context c_1. We allow sentences like

$$holds(c_1 \leq c_2, C_O)$$

that even statements relating contexts can have contexts. The theory would not provide for any "most general context" any more than set theory provides for a more general set.

A logical system using contexts might provide operations of *entering* and *leaving* a context yielding what we might call *unnatural deduction* allowing a sequence of reasoning as given in the followings.

$$\begin{aligned}&holds(p, C)\\&ENTER\,C\\&p\\&...\\&...\\&q\\&LEAVE\ C\end{aligned}$$

This resembles the usual logical natural deduction systems.

Example 20.2 *Represent the following sentence in the formalism of situation calculus.*

"A man called Alex is standing at the courtyard of Yale University wearing a white T-shirt. Alex loads his gun, waits for few seconds, and shoots at squirrel."

Solution. The fluents, which are either true or false, in this sentence are:

standing(place): whether Alex is standing at a given place or not?
white: whether Alex is wearing white T-shirt or not?
loaded: whether the gun is loaded or not?

Following are the actions in above sentence:

load: Alex is loading the gun.
wait: Alex waits for few seconds.
shoot: Alex shoots his gun.

Now lets find out what holds at initial situation S_0.

$$holds(standing(courtyard), S_0)$$

$$holds(white, S_0)$$

$$holds(alive, S_0)$$

$\neg holds(loaded, S_0)$

Now we try to relate actions with situations. In other words, which fluents will hold after performing a certain action in a given situation?

1. If Alex shoots the gun and gun is loaded, squirrel is not alive.
2. If Alex shoots the gun, gun will become unloaded.
3. If Alex loads the gun, gun will be loaded.
4. If Alex waits on an loaded gun, the gun remain loaded.
5. If Alex waits on an unloaded gun, the gun remain unloaded.

Our first method for writing above sentences in formal way is:

$$\forall s[holds(f_1, s) \rightarrow holds(f_2, do(a_2, s))]$$

That is, f_2 (fluent) will be true by performing action a_2 in state s if f_1 is true in s.

Now consider the sentence: "if Alex shoots the gun and the gun is loaded, squirrel is not alive," which is written as:

1. $\forall s[holds(loaded, s) \rightarrow \neg holds(alive, do(shoots, s))]$

   The other sentences are also formally expressed as:

2. $\forall s[\neg holds(loaded, do(shoot, s))]$
3. $\forall s[holds(loaded, do(load, s))]$
4. $\forall s[holds(loaded, s) \rightarrow holds(loaded, do(wait, s))]$
5. $\forall s[\neg holds(loaded, s) \rightarrow \neg holds(loaded, do(wait, s))]$

References

[RNO05] S. Russel and P. Norvig, "Artificial Intelligence - A Modern Approach," *2nd Edition, Pearson Education, India*, 2005, Chapter 10.

[DOU95] Douglas B. Lenat, "CYC: A Large-Scale Investment in Knowledge Infrastructure", In *Communications of the ACM*, November 1995/Vol. 38, No. 11, pp. 33-38.

[SOW95] Sowa, J. F., "Distinctions, Combinations, and Constraints", In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, 1920 August, Montreal, Canada,1995.