To be published in:
Electronic News Journal on Reasoning about Actions and Change
Vol. 1: [*pages tbd*], 31.10.1997
http://www.ida.liu.se/ext/etai/rac/notes/1997/10/

Action Languages from $\mathcal{A}$ to $\mathcal{C}$: A Statement for the Panel on Ontologies

Vladimir Lifschitz
University of Texas at Austin

The ontology of the original action language $\mathcal{A}$ [Gelfond and Lifschitz1993] is more restrictive than the ontological assumptions of its "dialects" proposed later. In this note, I compare $\mathcal{A}$ with the recent proposal called $\mathcal{C}$ [Giunchiglia and Lifschitz1997]. (There is no discussion here of other work on action languages—that would require a long paper.)

The language $\mathcal{A}$, as well as some of its successors, has two components. Its "effect propositions" describe the effects of actions on fluents, and their meaning can be described by a transition diagram similar to the diagrams familiar from the theory of finite automata. They form the "action description part" of $\mathcal{A}$. The "value propositions" of $\mathcal{A}$ are conditions on a path in this transition diagram. They form the "query part" of $\mathcal{A}$. The task of designing the query component of an action language is pretty much orthogonal to the task of designing its description component [Lifschitz1997], and it is convenient to address these two problems separately. In accordance with this idea, $\mathcal{C}$ was designed as a pure *description* language; there is no counterpart of value propositions in it. The semantics of $\mathcal{C}$ is defined by showing how sets of propositions describe transition diagrams.

Here are some differences between the ontology of $\mathcal{C}$ and the ontology of the effect propositions of $\mathcal{A}$.

1. $\mathcal{A}$ is based on propositional logic; $\mathcal{C}$ uses full classical logic (even higher-order, if we wish). Accordingly, the ontology of $\mathcal{C}$ includes *families* of actions and fluents; they are represented by symbols with arguments. Furthermore, fluents can be non-Boolean; then they are represented by terms rather than formulas. We do not have to represent fluents by expressions like *Block5IsOnTable*, as in $\mathcal{A}$; instead, we can write $On(Block5, Table)$ or $Location(Block5) = Table$. We can also use variables and write expressions like $Location(x) = l$.

2. In $\mathcal{A}$, every action is executable in every state, and in exactly one way. In $\mathcal{C}$, actions can be nondeterministic. A transition diagram described in $\mathcal{C}$ looks like a transition diagram of a nondeterministic finite automaton: It can have several edges beginning in the same state and having the same action label. This set of edges can be

empty, which means that the action is impossible to execute in the given state.

3. There is no way to talk in $\mathcal{A}$ about the concurrent execution of actions. In $\mathcal{C}$, actions can be performed concurrently. Accordingly, an edge in a transition diagram is labeled by a *set* of actions, rather than a single action. In particular, this set of actions can be empty—the world can change even when we do nothing. In this way, we can describe causal relationships between states, as opposed to causal relationships between an action and a state. (Having \$100 in my bank account today causes the balance to become \$100.01 tomorrow even when I do nothing.) Syntactically, this is achieved by using *predicate* symbols to represent (families of) actions, instead of function symbols as in the situation calculus. The formula

$$Move(Block5, Table) \wedge \neg\exists l\, Move(Block6, l)$$

expresses that *Block5* is moved onto the table while *Block6* is not moved anywhere.

4. In $\mathcal{A}$, every fluent that has a name in the language is "inertial," that is to say, tends to keep the value that it had before. In $\mathcal{C}$, the assumption that a fluent is inertial can be easily expressed, but it is not "built-in." Some fluents are not inertial. For instance, the balance of my bank account tends to change in accordance with a certain formula. This is similar to the inertia assumption, but not quite the same. Whether a string on my guitar is producing a sound is not an inertial property either. It is "momentary," in the sense that it tends to have the value false.

Syntactically, the main feature of $\mathcal{C}$ is the use of "dynamic causal laws" of the form

$$\textbf{caused } F \textbf{ if } G \textbf{ after } H$$

("there is a cause for F to be true if G is true and H was true before"). The semantics of these expressions is defined by a translation into classical logic and is, in this sense, similar to circumscription. Its main idea comes from [McCain and Turner1997]. Here are some examples:

1. The effect of *Move* on *Location* can be expressed by

$$\textbf{caused } Location(x) = l \textbf{ after } Move(x, l)$$

which is shorthand for

$$\textbf{caused } Location(x) = l \textbf{ if } True \textbf{ after } Move(x, l).$$

2. The impossibility of moving a heavy object can be expressed by

$$\textbf{nonexecutable } Move(x, l) \textbf{ if } Heavy(x)$$

which is shorthand for

$$\textbf{caused } \mathit{False} \textbf{ if } \mathit{True} \textbf{ after } \mathit{Move}(x,l) \wedge \mathit{Heavy}(x).$$

3. The inertiality of *Location* can be expressed by

$$\textbf{inertial } \mathit{Location}(x) = l$$

which is shorthand for

$$\textbf{caused } \mathit{Location}(x) = l \textbf{ if } \mathit{Location}(x) = l \textbf{ after } \mathit{Location}(x) = l$$

("there is a cause for x to be at location l if x is there now and was there before.").

Two ontological ideas that I do not know how to express in $\mathcal{C}$ are

- continuous time, and
- causal relations between actions.

Do they require that we go beyond the framework of transition diagrams?

References

[Gelfond and Lifschitz1993] Michael Gelfond and Vladimir Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–322, 1993.

[Giunchiglia and Lifschitz1997] Enrico Giunchiglia and Vladimir Lifschitz. An action language based on causal logic. Unpublished draft. Available at `http://www.cs.utexas.edu/users/vl/mypapers/c-short.ps`, 1997.

[Lifschitz1997] Vladimir Lifschitz. Two components of an action language. *Annals of Mathematics and Artificial Intelligence*, 1997. To appear.

[McCain and Turner1997] Norman McCain and Hudson Turner. Causal theories of action and change. In *Proc. AAAI-97*, pages 460–465, 1997.