ELECTRONIC NEWSLETTER ON REASONING ABOUT ACTIONS AND CHANGE

Issue 98032 Editor: Erik Sandewall 31.3.1998

Today

Today, an answer by Rob Miller and Tony Kakas to the question that Francois Levy posed to them earlier this month. Also, the discussion about the ontology of time continues, with concurrently written contributions by Graham White, John McCarthy, and Pat Hayes.

ETAI Publications

Discussion about received articles

Reasoning about Actions, Narratives and Ramification

by Antonis Kakas and Rob Miller

Dear Francois,

Thanks again for your questions.

> *First, according to your view of ramifications, fluents can be initiated/terminated in two ways: either when an event occurs, or due to changing fluents in a constraint. The formal difference is that a fluent changing its value is not by itself an event. Do you consider it to rely on an ontological difference – i.e. in the process of modeling the real world, two kinds of objects of different nature have to be considered: events on the one side, (instantly) changing fluents on the other one. Or do you consider both similar, and make a difference on a purely technical ground (trigered events don't work to render this if the time line is dense)?*

In answer to your first question: we do have an ontological difference between actions and fluents, but we don't make a distinction between different types of fluent. For example, a fluent can be terminated both 'directly' via a c-proposition and 'indirectly' via an r-proposition (e.g. 'Switch2' in the electric circuit example.) The essential point is that all fluents are only initiated via initiation points and only terminated via termination points, and that all initiation and termination points are characterised by a (relevant) action occurrence (i.e. an event). In other words, in our framework all changes in fluent values have as their root cause an event (or a set of concurrent events). Both the types of change identified above are 'direct' in the sense that the effect of the corresponding event is 'instantaneous' (where 'instantaneous' has a slightly different interpretation for discrete time than for dense or continuous time).

So the choice of whether to use r-propositions as well as c-propositions when modelling a particular domain is partly pragmatic. "`-Switch2 whenever Relay`" can be read as "all the events which initiate Relay also terminate Switch2". The use of this r-proposition thus enables us to avoid writing a whole series of terminates propositions for Switch2 corresponding to each of the initiates propositions for Relay.

But there are other advantages in using r-propositions, as identified in the paper. Not least, it helps with succinctly and correctly capturing the effects of concurrent events. For example, the concurrent 'stuffy room' example in the paper is difficult to describe in $cal(E)$ without r-propositions, and difficult to describe in conventional Event Calculus (at least without making some sort of distinction between 'frame' and 'non-frame' fluents).

> *Second, as far as I understand, your predicate 'Whenever' embeds both a domain constraint and a notion of influence, in Michael Thielscher's sense in his AI97 paper. The domain constraint is what you call the static view – i.e. 'Whenever' being replaced by a material implication. The influence information is: in the formula* `L Whenever C`*, only* `L` *can be initiated, so one domain constraint yields as many* `Whenever` *formulas as fluents can be influenced in it. But Thielscher's Influence predicate is binary, and of course his cause imp effect propagation is a different technique. I tried shortly some example, and could'nt find a difference in the flow of causality. Do you agree with these remarks ? And do you believe that some formal correspondance could be established between your two formalisms ?*

In answer to your second question: yes, we broadly agree with these remarks, in that r-propositions act both as static domain constraints and as unidirectional propagators of change. You're right as well with your observation that one domain constraint can yield a number of r-propositions. For example, a definitional domain constraint such as
would be represented with the r-propositions

```
Alive whenever {-Dead}
-Alive whenever {Dead}
Dead whenever {-Alive}
-Dead whenever {Alive}
```

However, we feel that it would be difficult to establish a formal correspondence between our approach to ramifications and Michael Thielscher's, for the reasons outlined in our discussion section. There seems to be a difference in the approaches in that Michael's effect propagation is 'approximately' instantaneous, whereas ours is 'truly' instantaneous (this is not to say that either is right or wrong - just that they're modelling slightly different concepts). This difference manifests itself in domains such as Michael's 'light detector' example (see Section 5 of his AI97 paper). We'd model the introduction of the detector with the single r-proposition

```
Detect whenever {Light}
```

But we wouldn't get the same 'non-deterministic' behaviour of the detector that Michael gets - i.e. we wouldn't get the model in which the detector is activated when `Switch1` is connected. Indeed, this model wouldn't make sense in a narrative-based formalism with explicit time - the detector would have been activated even though there was no time-point at which the light was on. (Michael expands on the theme of 'approximately' verses 'truly' instantaneous effects in his related paper in the proceedings of Common Sense '98.)

Tony and Rob.

Debates

Ontologies for time

From: Graham White

John McCarthy wrote:

> *When my car accelerates, there is a time point at which it passes 65 miles per hour. It is awkward to describe this point in a language not providing for time points.*

Indeed so; but all this shows is that it's awkward to combine an interval-based approach to time with a point-based approach to other continua (such as, for example, velocity).

Graham White

From: John McCarthy

[1][McCarthy and Hayes (1969)] used time as a fluent on situations, i.e. time(s). One motivation was that people, and perhaps future robots, often do not know the time with sufficient resolution to compare two situations, e.g. Ray Reiter's recent message with times 1.41 and 3.14. A second motivation for making situations primary was to make it correspond to human common sense. Many people who can reason about the consequences of actions in situations perfectly well do not know about real numbers, and some don't know about numbers at all. The falling body example was also in that paper with time as a fluent. Galileo did know about real numbers.

It's not clear that either of these considerations is of basic importance for AI.

My previous message gave a reason for including time points in a theory of events and actions. The theory could be founded so as to regard them as degenerate intervals, but I don't see any advantage in that, although I suppose the idea stems from the fact that people and robots can't measure time precisely.

From: Pat Hayes

Responses to Ray Reiter and John McCarthy. Ray wrote:

> *In the sitcalc, a fluent (LightOn) has a truth value only with respect to a situation (= sequence of action occurrences). So, we might have*
>
> $$LightOn(do(switchOn, do(switchOff, S0)))$$
>
> *and*
>
> $$\neg LightOn(do(switchOff, do(switchOn, S0)))$$
>
> *In the sitcalc with explicit time, the first might become*

[1] Ref: http://www-formal.stanford.edu/jmc/mcchay69.html

$$LightOn(do(switchOn(3.14), do(switchOff(1.41), S0)))$$

> *meaning that as a result of the action history consisting of first switching off the light at time 1.41, then switching on the light at time 3.14, the light will be on. Notice that there is no way of expressing the claim that the light is, or is not on at time 3.14 (or 3.5), independently of the situation leading up to this time.*

I think of Ray's 'sequences of actions' as alternative ways the temporal universe might be, ie possible timelines (or histories, as Ray sometimes calls them.) The point/interval controversy is about reasoning within, or with respect to, one of these possible timelines; sitcalc gets this muddled up with reasoning about alternative futures for the partial timeline up to the present. (Think of the tree of accessible situations in a state's future: the distinction is between reasoning about a single branch, and comparing two different branches.)

> *On the other hand, time based formalisms do allow one to write* $LightOn(3.14)$*, without expicitly referencing, in their notation, the history leading up to the time 3.14 at which the fluent's truth value is to be determined. This seems to be the source of all the problems about open vs closed vs semi-open intervals and predicate truth values over these, and also why these seem to be non-issues for the sitcalc.*

This isn't where the difficulties lie. Even if there is only one possible future and only one thing that could happen at each situation, these conceptual problems about points and intervals would still arise and some solution for them would need to be found.

> *Now, one could rightly object to the above account because it provides only for fluent truth values at discrete time points, namely at the action occurrence times. So we are tempted to understand*

$$LightOn(do(switchOn(3.14), do(switchOff(1.41), S0)))$$

> *to mean that the light is on at time 3.14, but it tells us nothing about time 3.5 say. This is particularly bad for (functional) fluents that vary continuously with time, for example, the location of a falling object. To handle this, introduce a time argument for fluents, in addition to their situation argument.*

Hold on! What kinds of things are these 'times' supposed to be? They seem to be something like clock-times, ie temporal coordinates (maybe understood with respect to a global clock of some kind.) OK, but notice that this isn't what I mean by a 'timepoint'. There are at least six distinct notions of 'time' (physical dimension, time-plenum, time-interval, time-point, time-coordinate and duration.) I think the nearest thing in Reiter's ontology to what I call a time-point is something like the pairing of a clock-time with a situation ('3.14 in situation s').

For the light, one can write:

$$LightOnT(t, s) \leftrightarrow LightOn(s) \wedge t \geq start(s).$$

Here, start is defined by $start(do(a, s)) = time(a)$*, where* $time(a)$ *is the time at which the action* a *occurs in the history* $do(a, s)$*.*
An instance of this would be

$$LightOnT(t, do(switchOn(3.14), do(switchOff(1.41), S0))) \leftrightarrow t \geq 3.14$$

Here we have committed to the light being on at exactly the time of the switchOn action, and forever thereafter, relative to the history $do(switchOn(3.14), do(switchOff(1.41), S0))$*. In other words, provided* $switchOff(1.41)$ *and* $switchOn(3.14)$ *are the* **only** *actions to have occurred, then the light will come on at time 3.14, and remain on forever. Notice especially that we would have* **both**

$$LightOnT(3.14, do(switchOn(3.14), do(switchOff(1.41), S0)))$$

and

$$\neg LightOnT(3.14, do(switchOff(1.41), S0))$$

without contradiction...

This seems to be the half-open-interval solution, where intervals contain their endpoints but not their starting points. This makes sense for the sitcalc, which focusses on the *results* of actions, but seems ad-hoc and unintuitive in a broader context. (Also, BTW, the idea that one can ever say that some finite list of actions is **all** the actions that have occurred seems quite unrealistic. After all, people's fingers probably pushed the switch and something somewhere was generating electricity. Surely one should be able to actually infer this from a reasonably accurate common-sense description of light-switching.)

...This seems to be precisely the point at which purely time-based formalisms run into difficulties, and the sitcalc version of this problem illustrates the role that explicit situation arguments play in resolving these difficulties.

This isnt where the difficulties lie. These alternatives are obviously incompatible if they are asserted of the same timeline (the light can't have been both switched on and switched off at the same timepoint) and there is no contradiction is saying that p is true at time t in one possible timeline but not in another, as these are different timepoints.

The problem is that even if we stick to talking about a single timeline (eg the unique past, or one alternative future) there still seems to be an intuitive difficulty about timepoints like the time when a light came on. The solution I suggested - that is, truth at a point has to be defined relative to a reference interval containing the point (which is not my idea, let me add) - is similar in many ways to Reiter's , except it applies not just across timelines but also within a single one.

John wrote:

> *When my car accelerates, there is a time point at which it passes 65 miles per hour. It is awkward to describe this point in a language not providing for time points.*

Yes, I agree. Examples like this are what motivate the inclusion of both points and intervals as first-class objects. It is pretty awkward to do without points in any case if one wants to refer to the places (...that is, the times...) where (...that is, when...) intervals meet. However, points **can** be defined in terms of intervals , in principle, so having them around is essentially a matter of convenience more than a point of basic ontology. There are there, in a sense, whether one wants them or not.

The 65mph example is logically similar to the point at the top of a trajectory when the vertical velocity is zero. Examples like this appeal to a basic intuition about continuous change, that it has no 'jumps', so if it is $< x$ at $t1$ and $> x$ at $t2$, then it must $= x$ somewhere between $t1$ and $t2$. One can state this quite directly in the basic theory (For strict first-order syntax, replace (X...) by (value X ...)):

```
(continuous X i) =df
   (forall (y)(implies (between (X (begin i)) y (X (end y)))
                                (exists t) (and (in t i) (= y (X t)))  )

(strictlycontinuous X i) =df
   (forall (j) (implies (subint j i)(continuous X j)))
```

where `subint` is the Allen union `begins, inside, ends`. (This assumes that the timeline itself is dense; if not, strictlycontinuous is trivially true everywhere.) Other conditions like monotonicity and so forth also transcribe directly from their usual mathematical formulations.

Pat Hayes

This newsletter is issued whenever there is new news, and is sent by automatic E-mail and without charge to a list of subscribers, in plaintext format. The present postscript issue is a bona fide rendering of the same information in alternative typography. Instruction for contributors and other additional information is found at the following URL:

`http://www.ida.liu.se/ext/etai/actions/njl/`