Embedded Rules of Action Into Action Space

by Sven Nilsen, 2020

In this paper I introduce a logical theory for embedded rules of action into action space for agents.
I also represent an open problem for checking consistency of new rules without introducing unsoundness.

A rule of action is a tuple consisting of an observation and an action:

x : (observation, action)

An observation is something that the agent senses or thinks, which *almost* determines an action.

The semantics of a rule of action is slightly different from a function. Normally, it would look like this:

f : observation → action

A rule of action is not a function because not all actions are available at any time.

For example, a person drives a car toward a red light, but cannot stop since the car breaks are broken.

stop_at_red_light := (red_light, break)

red_light : observation
break : action

Therefore, a secondary rule of action might be used as a backup plan when the first rule fails:

use_horn_at_red_light := (red_light, use_horn)

use_horn : action

A rule of action must be followed at any time, if the observation is present and the action available.

An embedded rule of action into action space means that an agent is reflecting over choices to commit to new rules of action over time. One can think about this as a method of modifying the agent's behavior in realtime. Different kind of agents use different kinds of consistency reflection.

- Some agents permits modification of behavior over time (inconsistent)
- Some agents accepts only rules of actions that do not violate any previous moment (consistent)

As a logical theory, I suggest the study of maximizing of utility under the following assumptions:

- Perfect information about the environment
- Ability to go back in time

From these two assumptions, in order to guarantee consistency, the following two cases holds:

- A rule of action can be an a-priori axiom
- It is sufficient to choose an embedded rule of action at first opportunity

For example, the first time a person drives a car and see a red light, the person breaks.
From this moment on, the person breaks every time when seeing a red light, when possible.

Embedded rules of action are different from normal actions, because the action is motivated by an observation. A normal action is unmotivated by observations.

Here is an example of a car history that includes embedded rules of actions:

```
accelerate
break                   Unmotivated by an observation (no red light)
accelerate
(red_light, break)      Motivated by an observation (red light)
accelerate
break                   Motivated by an observation (red light)
```

The rule of action is only listed at the first opportunity, while following actions ignores the motivation.

When the rule of action is an a-priori axiom, the history looks as following:

```
(red_light, break)

accelerate
break                   Unmotivated by an observation (no red light)
accelerate
break                   Motivated by an observation (red light)
accelerate
break                   Motivated by an observation (red light)
```

Under maximization of utility and time travel, the optimal history of actions is independent of embedded rules of action when the utility cost of choices is zero.

```
cost_of_choices = cost_per_choice · |available_actions|
```

By adding a utility cost term for choices, an agent using embedded rules of action is motivated to include new rules.

An attempt to simplify consistency for a rule of action `(ho, ha) : (observation, action)` is to require a condition on previous history (plus a-priori rules of actions) such that it could not be chosen before:

```
can_pick_new_rule(ho : observation, ha : action) = ¬∃ h : prev_history {
        ∃ o : observations(h) { o == ho } ∧ ∃ a : available_actions(h) { a == ha }
}
```

It turns out that this is not enough to guarantee consistency, because of the complex semantics of rules.

Assume that a rule of action `$r_0$` is chosen at time `t`.
In the same moment, a distinct rule of action `$r_1$` is possible, but not chosen.

$r_0 := (o_0, a_0)$
$r_1 := (o_1, a_1)$

$o_0 \neg= o_1$
$a_0 \neg= a_1$

Later checks using `can_pick_new_rule` for `$r_1$` returns `false`, not necessarily because `$r_0$` is exclusive, but because of a *soundness error*. If the action `$a_0$` is unavailable, then `$r_1$` is no longer exclusive to `$r_0$`. Hence, there are some cases where `$r_1$` should be embedded in the action space even if the observation and available action was present at an earlier moment in the history of the agent.

The open problem is how to define a correct version of `can_pick_new_rule`.

Notice that because of the ability to go back in time, it is always possible to choose a new rule at the first opportunity it can be added, thereby minimize the future utility cost of choices. The difficulty is to do this without introducing unsoundness.