Function Decision Theory in Path Semantics

by Sven Nilsen, 2017

Newcomb's problem according to Function Decision Theory is a decision procedure that reasons about it's own output relative to a perfect predictor function (a path). Since this can get confusing for people, I show that there is no paradox, but contrary that the existence of a predictor function is necessary.

Thanks to Function Decision Theory *(Eliezer Yudkowsky and Nate Soares 2017)*, path semantics can now be extended to a unified theory of instrumental rationality in decisions and games. Link to paper: https://arxiv.org/abs/1710.05060

decision : [choice] → nat	Decision procedure
predictor : [choice] → real	Predictor function
reward : nat → real	Reward function

Function Decision Theory (FDT) in general can be formalized as following:

decision[id → reward] <=> predictor (1)
¬∃ i : nat { ∀ x : ∃reward{∃decision} { reward(i) > x } } (2)

The first line (1) states that the predictor is predicting the reward made by the choice of the decision procedure. According to the law of function identity in path semantics, it follows that the decision procedure can not imagine itself outputting any other choice than some giving the reward predicted by the predictor function.

The second line (2) states that there exists no other choice that gives a higher reward than the reward of the action decided by the decision function. Technically it allows a non-deterministic `decision` for equivalent rewards, since non-determinism can be thought of a family of rational decision procedures.

In path semantics, this means the decision procedure must contract in a certain way. It says that the reward, not the choice, for any rational decision must be predictable. At higher level, this means there exists a classifier function `rational` that determines whether any function `f` is rational by FDT:

f : [choice] → nat ∧ [rational] true	`f` is rational *iff* it satisfies (1) and (2)
rational : ([choice] → nat) → bool	

In the special case of Newcomb's problems, which "just" is a sub-type of all problems:

newcomb : [choice] → bool	Newcomb problem constraint

decision{newcomb}[id → reward] <=> predictor{newcomb}
¬∃ i : nat { ∀ x : ∃reward{∃decision{newcomb}} { reward(i) > x } }

This does not break the path, but only allows irrational decisions on non-Newcomb problems.