# Data Structure for Motivated Actions

by Sven Nilsen, 2018

*In this paper I formalize a data structure for motivated actions of potentially infinite complexity, which is possible by using axiom schemes in logic. As a result this formalization has an extremely simple form, because all the complexity is hidden in the machinery of logic itself.*

An ideal data structure for motivated actions can answer any logical question about motivation vs action. This is important for some domains in AI, such as higher order reasoning about goals. It takes the form of a logical axiom to provably answer any logical question, which other representations are derived from. The logical form is material implication. The left side is called the “motivation” and the right side is called the “action”:

$$\text{motivation} \rightarrow \text{action}$$

The motivation is an expression that evaluates to `true` or `false`. When the motivation is `true`, then the action is `true`. An action that is `true` does not mean that the action gets done, but rather that it is desirable to perform the action. One action can have multiple motivations, and one motivation can result in multiple actions.

Examples:

“cold outside” → “wear jacket”
“wind outside” → “wear jacket”
“cold outside” ∧ “wind outside” → “wear scarf”
“rain outside” → “wear umbrella”

A collection of motivated actions has the logical form:

$$\forall\ i\ \{\ \text{motivation}_i \rightarrow \text{action}_i\ \}$$

The reason of using a such simple logical form is that logic can be used to reason about questions formulated as logical statements. For example, given a specific motivation, one can answer which actions that are desirable to take. When an action is undesirable, one can figure out which motivations that are not present.

- The motivation must account for all situations when the action is not desirable
- It is not possible to derive a motivation from knowledge about an action only
- A motivation can be an observation, such as “it is cold outside”

Logic allows one to declare which properties a data structure has without needing to specify how the data structure is programmed, or even putting restrictions on the content of the data structure versus the environment.

For the is-ought problem proposed by David Hume, a data structure for motivated action makes it possible to derive “ought” (action) from “is” (motivation), but not “is” from “ought”.