Programming with Epistemic Logic

How to program what is perceived

Markus Eger
NC State University
meger@ncsu.edu

Chris Martens
NC State University
crmarten@ncsu.edu

Keywords Epistemic Logic, Dynamic Logic, State Representation

1. Motivation

The study of knowledge, epistemology, has a rich history, going back to the philosophers of ancient Greece (Plato 369BCE), with more recent research in the field of epistemic logic (Goldblatt 2003). The latter often uses possible world models with Kripke semantics to represent the agents' mental states. In this model, there is a collection of worlds, with one being marked as the actual world, and a relation R_a that holds between two worlds iff an actor a considers the second world possible when they are in the first world. We say an agent a knows p, written $\Box_a p$ iff p holds in all worlds accessible via R_a from the world in which we are evaluating the sentence. For example, if a car is red, but an agent does not know if it is red, blue or green, the actual world would contain "the car is red", with two other worlds that contain "the car is blue" and "the car is green" and R_a containing all (ordered) pairs of (not necessarily different) worlds, because the agent can not distinguish between any of the three worlds. Additionally, let's say the agent knows that the card is old, so "the car is old" is present in all worlds they consider possible. If p is "the car is old", then $\Box_a p$ holds, because p is present in all 3 worlds accessible from the actual world. A nice property of this model is that it is naturally recursive, allowing us to make statements about nested beliefs. For example, $\Box_a \Box_a p$ would be interpreted as "does $\Box_a p$ hold in all worlds accessible by R_a from the actual world", and then in each of these worlds we would determine whether p holds in all worlds accessible from there, i.e. $\Box_a \Box_a p$ holds iff p holds in every world that is "2 steps" of R_a away from the actual world. By introducing multiple relations R_a, R_b, R_c, ... it becomes possible to express sentences like "a knows that b knows that c does not know that ...".

OBT 2017 January 21, 2017, Paris, France

ACM ...$15.00

These models have been proving useful when describing what is true in the world, but in many practical applications, knowledge changes over time. There are many different approaches to this problem, like Dynamic Epistemic Logic (DEL), of which there are several variants, probably best summarized by Herzig et al. 2008. We will consider a version proposed by Baltag 2002, in which actions also have Kripke semantics, in that an action can appear as any of a number of other actions to different agents, and—as with possible worlds—it is possible to reason about appearances of appearances of actions, e.g. what agent a thinks that agent b thinks that the possible actions were that could have occurred. In the car example above, suppose agent b, who already knows everything about the car, (truthfully) told agent a "the car is red if and only if it is old". In Baltag's language, this action is modeled with the syntax $(?old \cdot ?red)^{*a,b} + (?\neg old \cdot ?blue)^{*a,b} + (?\neg old \cdot ?green)^{*a,b}$. The addition represents a non-deterministic choice, the multiplication sequential composition. A proposition prefixed with a question mark represents a truth test. Finally, the choices are wrapped in $(\cdot)^{*a,b}$, which means that after the action a and b will both know that the action happened, which choice was taken, as well as that the respective other agent knows that it happened and which choice was taken, and that the other agent knows that they know, etc. The action could therefore be read as "a and b learn *truthfully*, *mutually* and *introspectively* that the car is old and red, or not old and blue, or not old and green". The way this action is executed is that each of its deterministic options is executed in every state that agents consider possible, and those states where the action does not apply are pruned. In our case, since the agents know that the car is old, the last two options do not apply in any state, and the first option only applies in a state in which the car is both old and red, so after this action both agents will know that the car is old and red.

Dynamic Epistemic Logic can clearly express interesting situations, but the question remains if it is useful for anything of practical value. Baltag demonstrates the capabilities of his logic by solving a simple children's puzzle, which takes a whole page of formulas, and Van Benthem describes how to apply DEL to games (Van Benthem 2001), but his examples only show small fragments or abstractions of games. This

is of course no fault on the authors' behalf, it is simply the case that the formalisms, while sound, are not very "user friendly", and require very verbose descriptions of what is happening. On the flip side, procedures in programs can be viewed as compact instructions on how to change the world, as represented by boolean formulas. Having a programming language that can express not only factual change to the world, but also how agents perceive such change, as well as what agents learn about the world, and allows accessing these mental models in control structures, would facilitate the usage of DEL or similar formalisms to solve real world problems. We will call this paradigm *Programming with Epistemic Logic*.

2. Programming with Epistemic Logic

As a first step towards Programming with Epistemic Logic we have been working on a language called *Satori*, after the Japanese Buddhist term for "comprehension/understanding". It is a simple assignment based language with an if-construct, but it also includes a `tell` statement that supports communicating (true) information to a recipient or a group of recipients. This learning is done mutually and introspectively, as in the example above. In general, agents will be unaware of any action happening, but the action may change the actual world nonetheless. It is possible to use the knowledge modality $\Box_a$ in the conditions of `if` statements to make execution conditional on agents knowledge. It is also possible to mark any statement as `public` which has the effect of letting all agents or a subset thereof know that that statement is happening. This may, of course, cause inconsistencies or even contradictions in an agent's mental model, for example if an actor believes that $x = 3$, does not know about a later assignment statement that sets $x = 4$, but is subsequently told that the if-branch of an `if` statement happened which had $x == 4$ as its condition. Since DEL models gaining knowledge as eliminating worlds that contradict that knowledge, this may lead to the agent believing that no worlds are possible. If proper care is taken, though, this language allows the concise definitions of the game actions of real world games. As an example, Hanabi is a knowledge-based cooperative card game. In contrast to most card games, players don't see their own cards, but they do see everyone else's. Part of the game is giving hints to other players, that identify all cards of a particular color, or of a particular rank in their hand. Listing 1 shows how the action of giving a hint about all cards of a particular color can be encoded in our language.

Listing 1. The "hint about a color" action for Hanabi

```
hintcolor(player: Players,
          col: Colors)
  tell (player):
       Each i in HandIndex:
           color(at(player, i)) == col
```

The `tell` statement here has a mode in which a player is told the subset of $HandIndex$ for which the proposition following holds, in this case, for which subset of cards in their hand it is the case that they are of the given color. As can be seen, this is a relatively compact representation of a concept that would be quite cumbersome to express directly in DEL. In fact, our implementation of Satori compiles the language to Baltag, and the hint color action results in a formula with 125 terms.

There are still issues that we have not addressed yet with the current iteration of Satori. For example, while it is possible to tell someone something truthfully, we have no characterization of lying, or more generally defining appearances of statements or whole actions arbitrarily. As we mentioned above, the logic we are compiling to has support for this and other features from which a variety of applications would benefit, for example non-cooperative knowledge-based games like Poker.

Another prohibitive factor is the representation of mental models as a set of possible worlds itself. Consider the case of a standard card game with a standard deck of 52 cards, 5 of which are in each of the players' hands. Every player knows their own cards, but has to consider every permutation of the remaining cards possible, for a total of $47! > 2.58e59$ possible worlds. Giving a player information, or simply drawing cards is then a matter of determining which of these worlds are still possible and eliminating the others, which is not tractable. It should be possible to use more compact state representations, especially for such common cases as a large set of worlds about which (almost) no knowledge has been revealed. One such approach might be to use a more compact representation of possible truth assignments like Binary Decision Diagrams (Bryant 1986), or to use Answer Set Programming to represent what an agent considers possible.

References

"Theaetetus", Plato, in: "Plato in Twelve Volumes, Vol. 12", translated by Harold N. Fowler. Cambridge, MA, Harvard University Press; London, William Heinemann Ltd. 1921., `http://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.01.0172:text=Theaet.:section=142a`

Goldblatt, Robert. "Mathematical modal logic: a view of its evolution." Journal of Applied Logic 1.5 (2003): 309-392.

Herzig, Andreas, Hans van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. "Dynamic Epistemic Logic." (2008): 441-445.

Baltag, Alexandru. "A Logic for Suspicious Players: Epistemic Actions and BeliefUpdates in Games." Bulletin of Economic Research 54, no. 1 (2002): 1-45.

Van Benthem, Johan. "Games in Dynamic Epistemic Logic." Bulletin of Economic Research 53, no. 4 (2001): 219-248.

Bryant, Randal E. "Graph-based algorithms for boolean function manipulation." IEEE Transactions on computers 100, no. 8 (1986): 677-691.