Mental Actions and Explainability in Kripkean Semantics: What Else Do I Know? (Student Abstract)

Shikha Singh[1], **Deepak Khemani**[2]
Dept. of Computer Science & Engineering
Indian Institute of Technology Madras
IIT P.O., Chennai 600036, India
[1] cs16d008@smail.iitm.ac.in, [2] khemani@cse.iitm.ac.in

Abstract

The ability of an agent to distinguish the ramification effects of an action from its direct effects adds value to the explainability of its decisions. In this work, we propose to encode the ramification effects of ontic and epistemic actions as single-point update models in an epistemic planning domain modeled with Kripkean semantics of Knowledge and Belief. We call them "mental actions". We discuss a preliminary approach to realize our idea, and we conclude by pointing out some optimizations as our ongoing pursuit.

Introduction: Motivation & Related Work

The deductive and causal nature of declarative knowledge representation approaches in knowledge-based systems facilitate these systems to explain their decisions. In planning domains, a rich action theory enables an agent to answer *why it decided to act in a certain way*. Interestingly, actions often have indirect effects or *ramifications*, and reasoning and planning with ramifications is a well-studied problem in Knowledge Representation (Shanahan 1999; Thiébaux, Hoffmann, and Nebel 2005). We find a variety of approaches in the literature (Pinto 1999; Strass and Thielscher 2013; Muise, Belle, and McIlraith 2014) that handle this problem either by having the ramifications encoded as actions themselves, or by compiling them away as direct effects of actions.

Example. Consider a scenario wherein a burglar breaks into a house. The domestic robot *DBot* knows that the house owner is not around, and it has to act immediately. It hits a fake siren installed in the garden, hearing which the burglar runs away. We see many instances of inference making in this example, owing to the direct as well as the ramification effects of actions. The one which is of particular interest to us is where *DBot* considers the possibility of the burglar making an inference that *the police are nearby* if the burglar hears the siren. It is this inference that led *DBot* to hit the fake siren, and when asked by the neighbors, *DBot* should be able to explain why it did so.

This example also features another crucial ingredient of explainability: *Theory of Mind*[1] reasoning. The *possible worlds* semantics of Epistemic logic (Hintikka 1962), encoded as a Kripke model, provides an elegant mechanism for expressing agents' beliefs in terms of possible, impossible or imaginary worlds in a multi-agent setting.

This work is focused on handling ramifications in the *possible worlds model* which enables *DBot* to think in terms of possibilities and explain its inferences, such as: "I know that the police are not around...but *I inferred that hearing the alarm would create a false belief in the burglar that the police have arrived, and therefore, it may infer a threat to itself, which will make it run away.*".

Our Approach

We build on our previous work, (Singh and Khemani 2020) which solves epistemic planning problems based on an agent's subjective knowledge in the domains consisting of only *ontic* (world-changing) and *epistemic* (belief-changing) actions. We extend it by introducing *mental (inferencing) actions*. The idea is close to the earlier works which define the relationship between primitive and derived fluents using axioms (McDermott 2000). We start with the assumption that the direct effects of an action can be described in terms of primitive fluents, and the indirect effects, only in terms of derived fluents. Relaxing this constraint is a work in progress. Proceeding with this setup, we show ramifications as mental actions, and apply them in the states that trigger them. To the best of our knowledge, we are the first to encode *ramifications* as single-point update models in a planner that uses the *possible worlds* model to represent an agent's subjective knowledge.

Mental Actions as Single-Point Update Models

A mental action, like an axiom, encodes the relationship between primitive and derived fluents, or among several derived fluents. We first categorize the domain fluents $\mathcal{F}$ as: *primitive fluents* and *derived fluents*. Then we apply stratification (Thiébaux, Hoffmann, and Nebel 2005) on the set of derived fluents, say $\mathcal{F}_D$. A stratification on $\mathcal{F}_D$ induces stratification on the set of *mental actions:* $actions_M$ too. Applying mental actions in the lower stratum, say

[1]TOM reasoning is the ability to ascribe mental states, such as beliefs, goals, and intentions to others (Premack and Woodruff 1978).

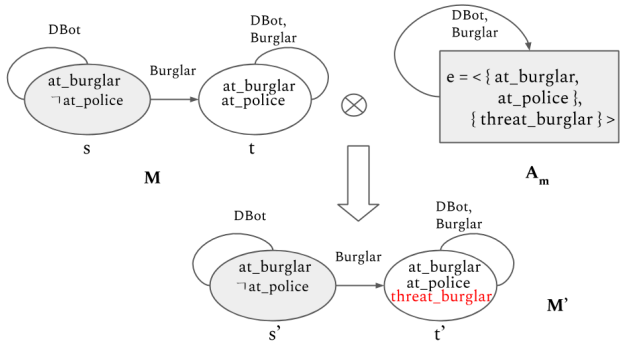

Figure 1: Mental action to infer threat to the burglar

$actions^{n}_{M}$, before applying mental actions in the next stratum, $actions^{n+1}_{M}$, leads to the same fixed point, had the mental actions in $actions_M$ been applied infinitely in any order.

Mental actions (for instance, A_m in Figure 1) are single-point update models (written as (A_m, e), for instance), with no observability restrictions, and are executable in only those worlds (such as world t in model M) in which the preconditions of the point (say, *at_burglar* and *at_police*) hold true. The model is then transformed in such a way that in the updated model (see world t' in model M'), the valuation of inferred (derived) fluent(s) (such as, *threat_burglar*) is set to true. Note that M is the resulting model after *DBot* hits the siren, leading to a false belief in the burglar that the police have arrived. M' is the resulting model after the ramification effect of hitting the siren is realized with the *mental action* A_m on M. We see that though *DBot* itself doesn't infer that there is a threat to the burglar, but it believes that the burglar would infer so.

As a preliminary approach, we let the planner[2] apply the stratified mental actions after every *ontic* and *epistemic* update (line no. 4-7, 15-18), as shown in Algorithm 1, and additionally store the inferences made after each update (not shown due to space constraints), which can be used for explanation generation at a later stage.

Conclusion and Future Work

The proposed approach increases the number of updates (or at least the applicability checks) in $O(|action_M| * |actions|^d)$ where d is the depth of the search tree at a particular time. We can optimize on the applicability checks by introducing the concept of *relevance* in the framework, such that instead of the states triggering mental actions, the ontic and epistemic actions trigger the relevant mental actions.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and encouraging feedback.

[2]The planning framework and the notations used in Algorithm 1 are described in detail in our previous work (Singh and Khemani 2020).

Algorithm 1: $KD45planner(PlanningProblemP = \langle(\mathcal{M}_0, Des), ..., actions, actions_M, self, goal\rangle)$

```
1  Initialize a queue, open = (M_0, Des)
2  while open is not empty do
3      (M, Des) = dequeue(open)
4      for i = 0; i < n; i = i + 1 do
5          foreach a_m in actions^i_M do
6              if a_m is applicable in (M, Des) then
7                  (M, Des) = (M, Des) ⊗ a_m
8      if (M, Des) ⊨ Goal then
9          SolutionPlan = TracePathToParent()
10         return SolutionPlan
11     else
12         for action ∈ actions do
13             if action is applicable in (M, Des) then
14                 (M', Des') = (M, Des) ⊗ action
15                 for i = 0; i < n; i = i + 1 do
16                     foreach a_m in actions^i_M do
17                         if a_m is applicable in
                               (M', Des') then
18                             (M', Des') = (M', Des')
                                   ⊗ a_m
19                 enqueue(open, (M', Des'))
20 return false
```

References

Hintikka, J. 1962. *Knowledge and belief: An introduction to the logic of the two notions*. Cornell University Press.

McDermott, D. M. 2000. The 1998 AI planning systems competition. *AI magazine* 21(2): 35–35.

Muise, C.; Belle, V.; and McIlraith, S. A. 2014. Computing contingent plans via fully observable non-deterministic planning. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Pinto, J. 1999. Compiling ramification constraints into effect axioms. *Computational Intelligence* 15(3): 280–307.

Premack, D.; and Woodruff, G. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences* 1(4): 515–526.

Shanahan, M. 1999. The event calculus explained. In *Artificial intelligence today*, 409–430. Springer.

Singh, S.; and Khemani, D. 2020. Planning with Subjective Knowledge in a Multi-Agent Scenario. In *11th Hellenic Conference on Artificial Intelligence*, 1–9.

Strass, H.; and Thielscher, M. 2013. A general first-order solution to the ramification problem with cycles. *Journal of Applied Logic* 11(3): 289–308.

Thiébaux, S.; Hoffmann, J.; and Nebel, B. 2005. In defense of PDDL axioms. *Artificial Intelligence* 168(1-2): 38–69.