Asking the Right Questions: Learning Interpretable Action Models Through Query Answering

Pulkit Verma, Shashank Rao Marpally, Siddharth Srivastava, Arizona State University

How would a non-expert assess the limits and capabilities of an AI system?

Introduction

Objective: Learn an interpretable model of a black-box agent by interrogating it.

Key technical challenge:
- Which sequence of queries to ask?

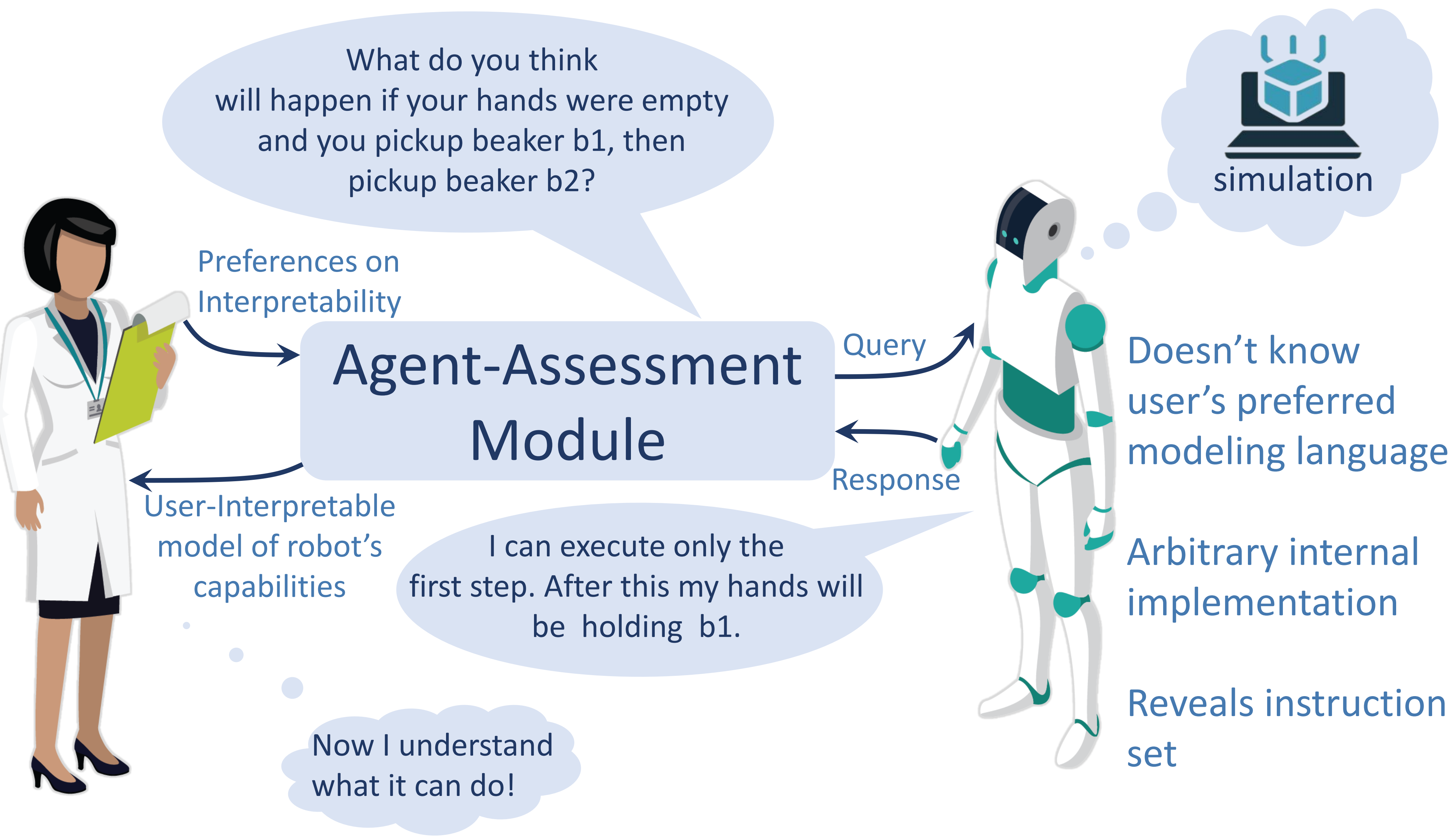
Algorithm (AIA)

1. Start with the most abstracted node in lattice.
2. Pick abstraction candidates in some order.
3. For each candidate, generate three models and for each pair of models:
4. • Generate a distinguishing query Q and pose it to the agent.
5. • Get the response R from the agent.
6. • Prune out the incorrect variants of candidate models.
7. • Repeat steps 3-6 till the model is fully estimated.
8. Return the final set of model(s).

Salient Features

- Efficiently learns internal model of an autonomous agent in a STRIPS-like form.
- Needs no prior knowledge of the agent model.
- Only requires an agent to have rudimentary query answering capabilities.
- Queries can be answered using a simulator.

Example of Agent Interrogation



What do you think will happen if your hands were empty and you pickup beaker b1, then pickup beaker b2?

Preferences on Interpretability

Agent-Assessment Module

Query

Response

User-Interpretable model of robot's capabilities

I can execute only the first step. After this my hands will be holding b1.

Now I understand what it can do!

simulation

Doesn't know user's preferred modeling language

Arbitrary internal implementation

Reveals instruction set

Abstraction in Space of Models

Abstracted model

```
(:action load_truck
 :parameters (?package ?truck ?location)
 :precondition (and (at ?truck ?location)
                    (+/-/∅) (at ?package ?location))
 :effect (and (not (at ?package ?location))
              (in ?package ?truck)))
```

abstraction ↑

Concrete model

```
(:action load_truck
 :parameters (?package ?truck ?location)
 :precondition (and (at ?truck ?location)
                    (at ?package ?location))
 :effect (and (not (at ?package ?location))
              (in ?package ?truck)))
```

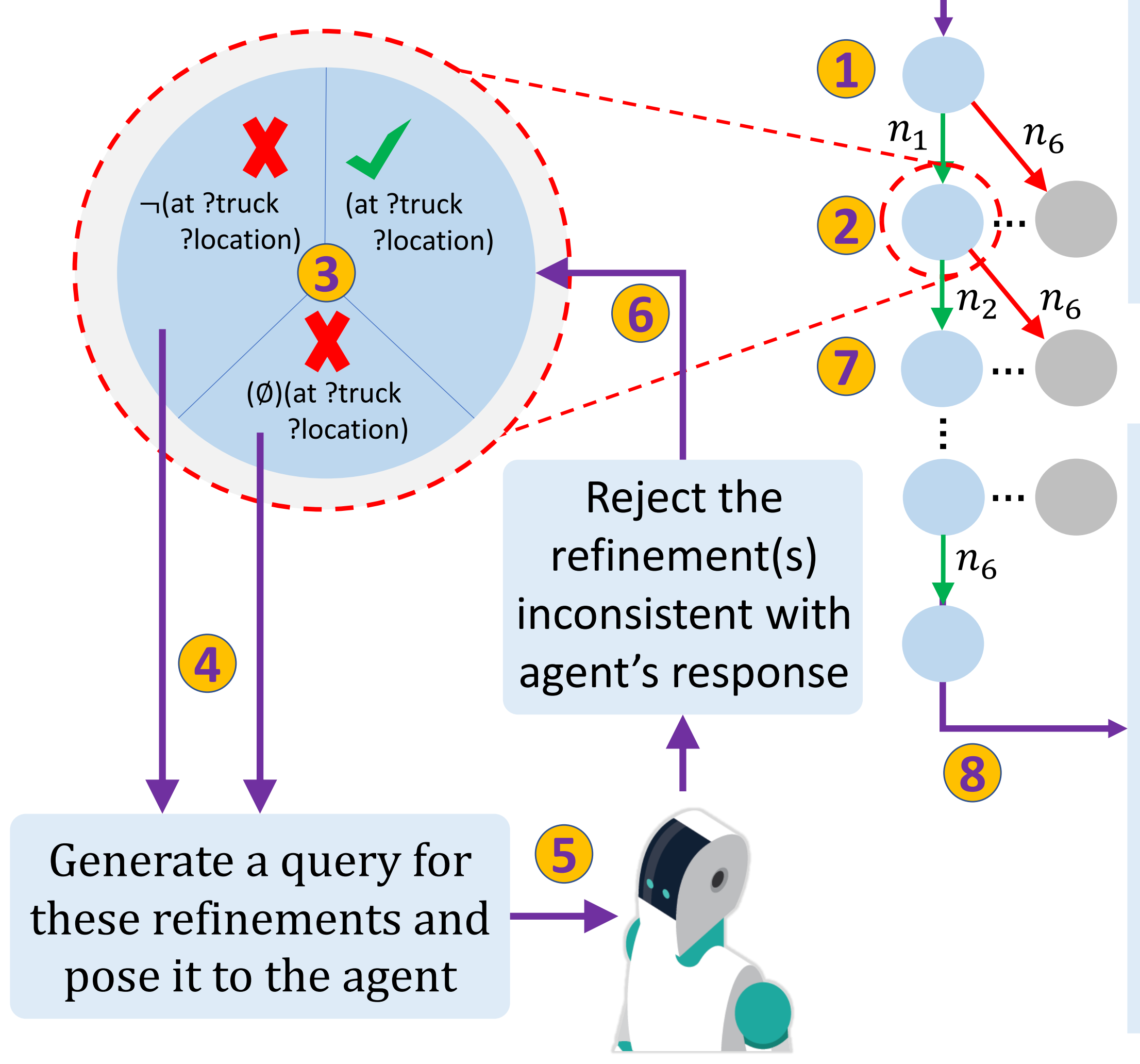
This predicate can appear in three forms:
- positive
- negative
- absent

Key Algorithmic Principle

Key feature of the algorithm
Each time we prune an abstracted model, we prune a very large number of models at the most concrete node.

```
(:action load_truck
 :parameters (?package ?truck ?location)
 :precondition (and (+/-/∅) (at ?truck ?location)      n_1
                    (+/-/∅) (in ?package ?truck)       n_2
                    (+/-/∅) (at ?package ?location))   n_3
 :effect (and (+/-/∅) (at ?package ?location)          n_4
              (+/-/∅) (at ?truck ?location)            n_5
              (+/-/∅)  (in ?package ?truck)))          n_6
```

Model estimate before querying



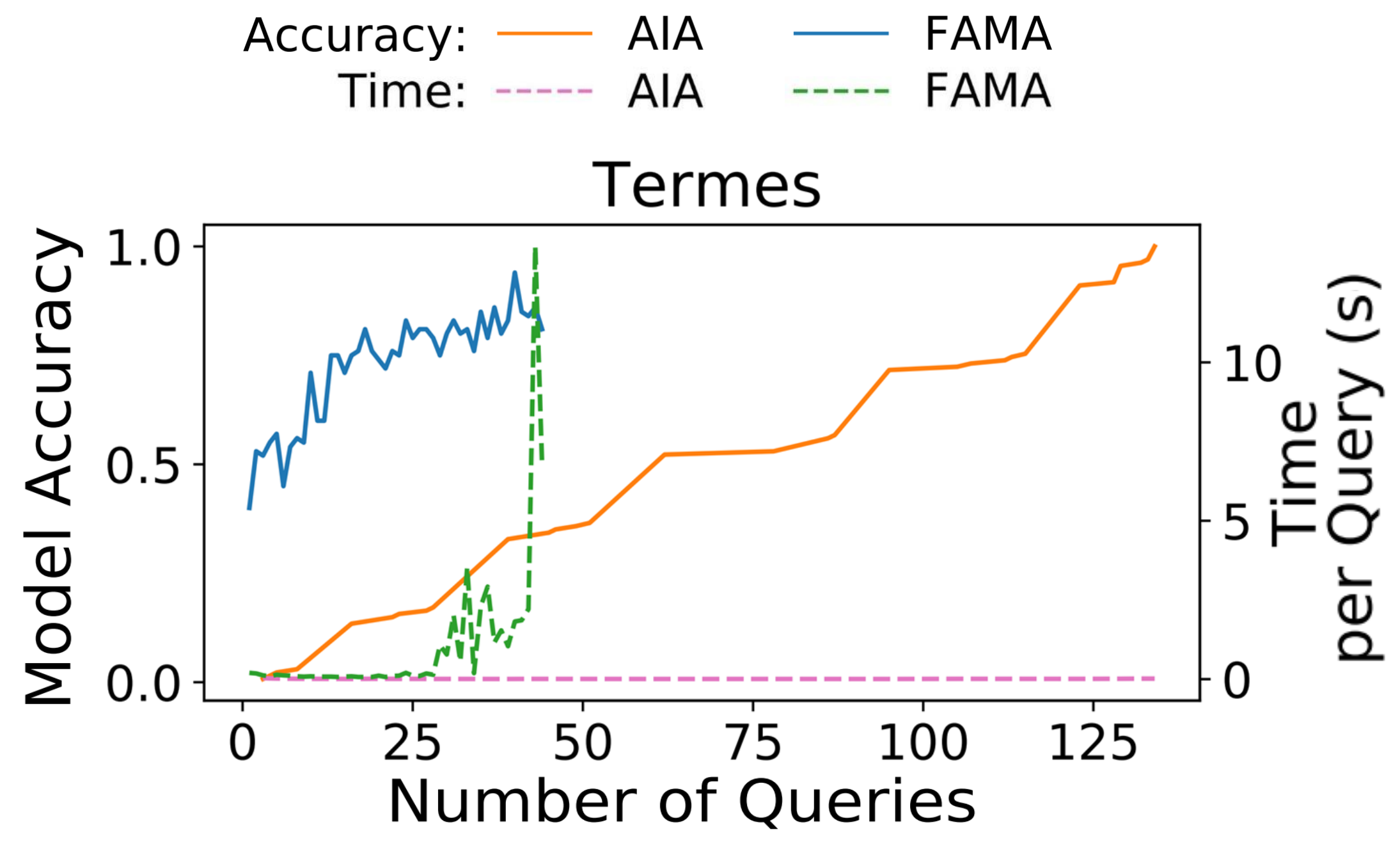
Reject the refinement(s) inconsistent with agent's response

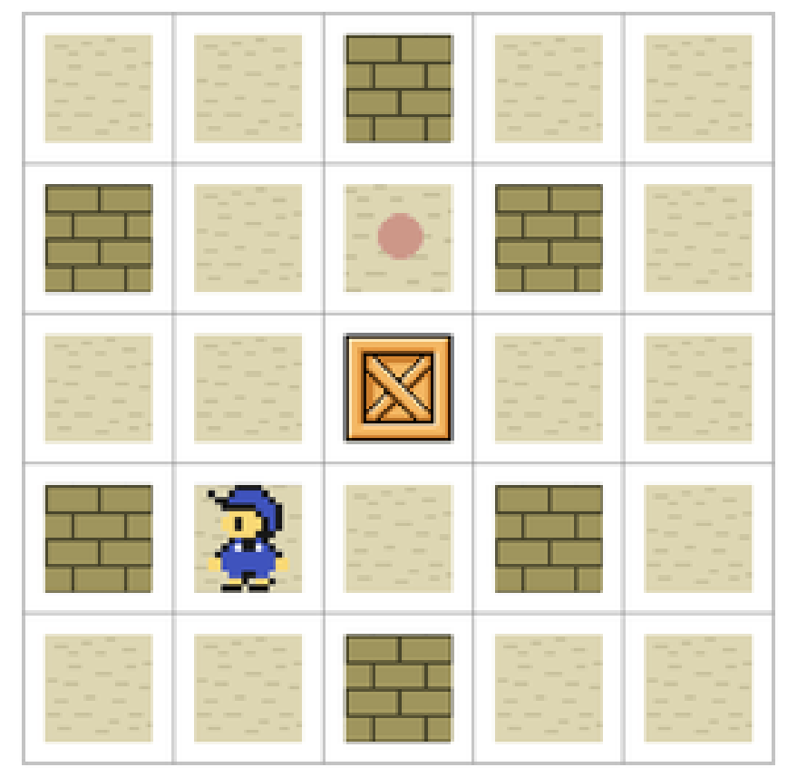
Generate a query for these refinements and pose it to the agent

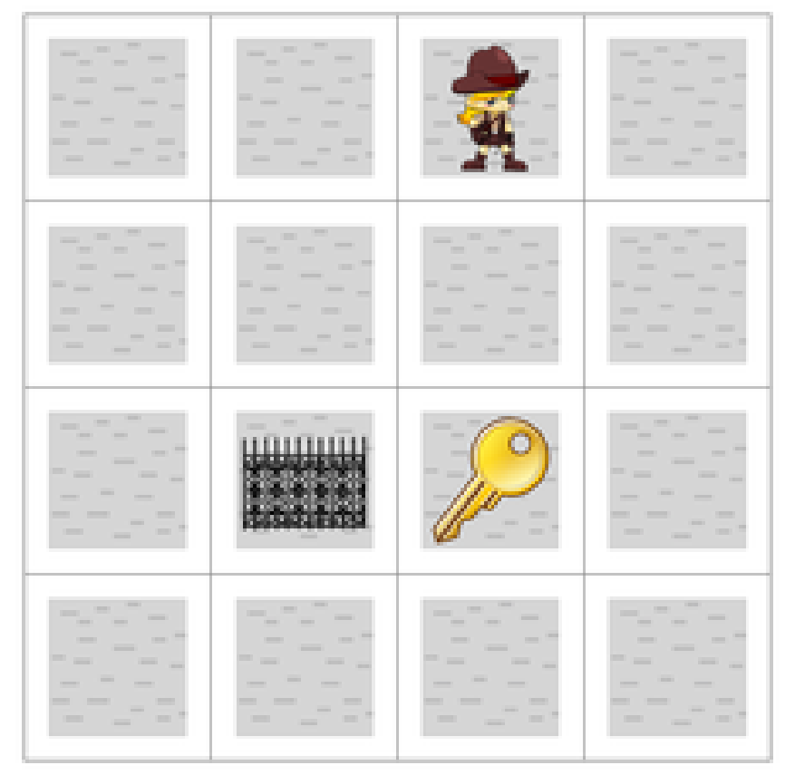
```
(:action load_truck
 :parameters (?package ?truck ?location)
 :precondition (and (at ?truck ?location)
                    (not (in ?package ?truck))
                    (at ?package ?location))
 :effect (and (not (at ?package ?location))
              (∅) (at ?truck ?location)
              (in ?package ?truck)))
```

Estimated Model after querying

Results







- AIA efficiently derives interpretable agent models for a range of agents.
- AIA is much faster than state of the art methods for deriving models based on passive observations.
- AIA offers better convergence guarantees.

Refer to the paper for detailed results

bit.ly/3p4cVRu