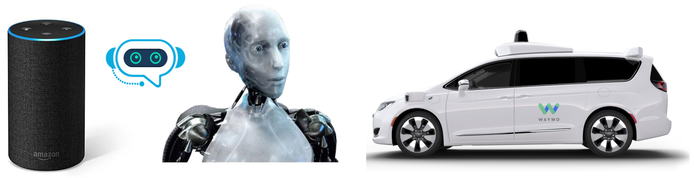
Learning Generalized Models by Interrogating Black-Box Autonomous Agents

Pulkit Verma and **Siddharth Srivastava**
Arizona State University, Tempe, AZ, USA

GenPlan'20

Introduction

Can a non-expert determine if an AI agent is reliable/safe for a task?



- Objective: Estimate an understandable model of a non-stationary black-box agent by interrogating it.
- Key technical challenge:
  - Which sequence of questions to ask?

Abstraction in Space of Models

```
(:action pickup
 :parameters (?ob)
 :precondition (and (handempty)
                    (+/-/∅) (ontable ?ob))
 :effect (and (not (handempty))
              (not (ontable ?ob)))
```

Abstracted model

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

abstraction

```
(:action pickup
 :parameters (?ob)
 :precondition (and (handempty)
                    (ontable ?ob))
 :effect (and  (not (handempty))
               (not (ontable ?ob)))
```

Concrete model

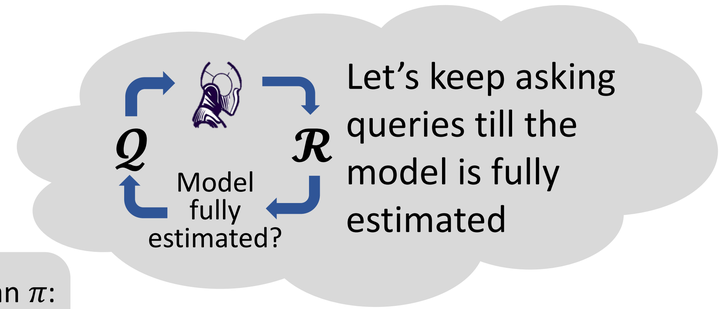
Algorithm

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).

Example of Agent Interrogation

Plan Outcome Query: Asks the outcome of a plan.

Query: Initial state, plan.
Response: Length of successful execution, final state.



What do you think will happen if you execute the plan π: ⟨pickup(b1), pickup(b2)⟩ starting in an initial state s_I: (ontable(b1) ^ handempty)?

I can execute only the first step, and the final state after executing one step was s_F:(¬ontable(b1) ^ ¬handempty ^ holding(b1)).

I think it needs to keep its hand empty before pickup

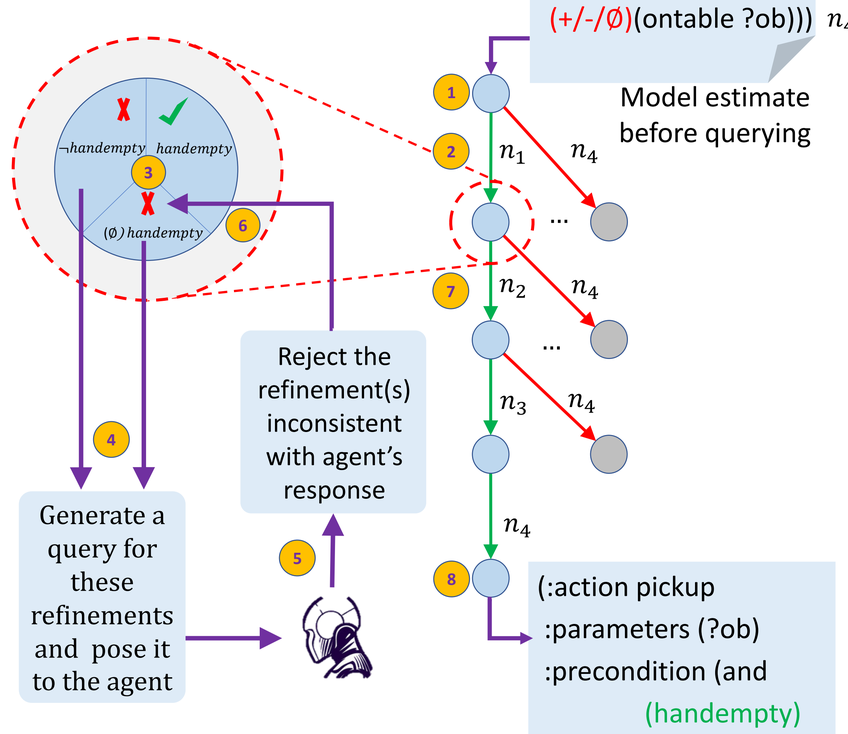
Key Algorithmic Principle

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 pickup
 :parameters (?ob)
 :precondition (and
  (+/-/∅)(handempty)      n1
  (+/-/∅)(ontable ?ob))   n2
 :effect (and
  (+/-/∅)(handempty)      n3
  (+/-/∅)(ontable ?ob)))  n4
```

Model estimate before querying



Estimated Model after querying

```
(:action pickup
 :parameters (?ob)
 :precondition (and
        (handempty)
        (ontable ?ob))
 :effect (and
    (not (handempty))
    (not (ontable ?ob))))
```

Results

- Randomly generate an agent and environment from the IPC benchmark suite.
- Algorithm estimates this agent's model.

Theorem: The algorithm will always terminate and return a set of models, each of which are functionally equivalent to agent's model.

Domain	$\lvert\mathbb{P}\rvert$	$\lvert\mathbb{A}\rvert$	$\lvert\mathcal{Q}_{naive}\rvert$	$\lvert\mathcal{Q}\rvert$	Time/$\mathcal{Q}$ (sec)
gripper	5	3	15×2^5	37	0.14
blocks-world	9	4	36×2^9	92	1.73
elevator	10	4	40×2^{10}	109	5.91
logistics	11	6	66×2^{11}	98	11.62
parking	18	4	72×2^{18}	173	12.01
satellite	17	5	85×2^{17}	127	19.53
openstacks	10	12	120×2^{10}	203	11.28

Number of queries generated in our approach vs naïve baseline. Results are averages of 10 random runs.

Salient Features

- Needs no prior knowledge of the agent model.
- Requires an agent to have only rudimentary query answering capabilities.
- Queries can be answered by the agent using a simulator.
- Works for non-stationary environments.

This work was supported in part by the NSF under grants IIS 1844325 and IIS 1909370