POMDPs.jl

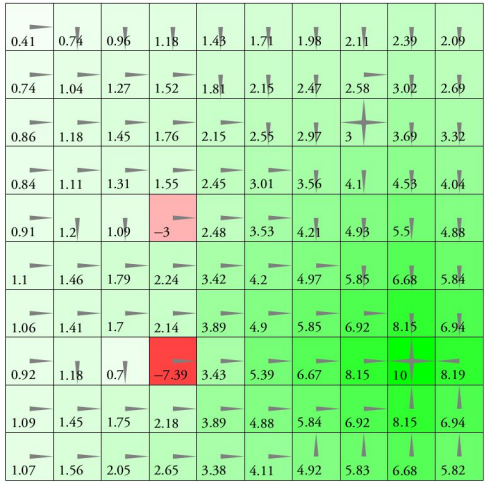
A unified framework for solving MDPs and POMDPs

What is an MDP?

A **Markov Decision Process** is a sequential decision making problem with

- A fully observable state
- Stochastic transitions based on the state and action
- Rewards based on the state and action

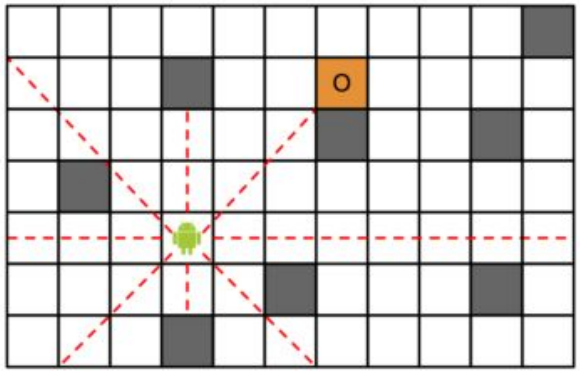
The goal is to find a **policy** that maps each **state to an action** that maximizes the expected sum of rewards.



What is a POMDP?

A **Partially Observable Markov Decision Process** is an MDP with

- States that are not directly observable
- Observations determined stochastically by the state and action

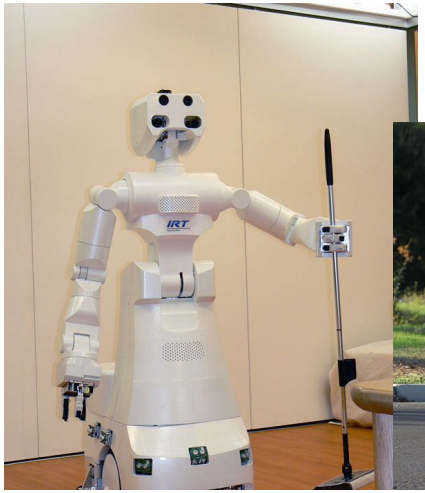


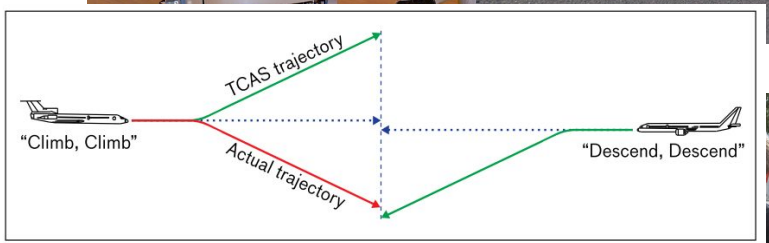
The goal is to find a **policy** that maps an **action-observation history** or **belief** to an action that optimizes the expected reward.

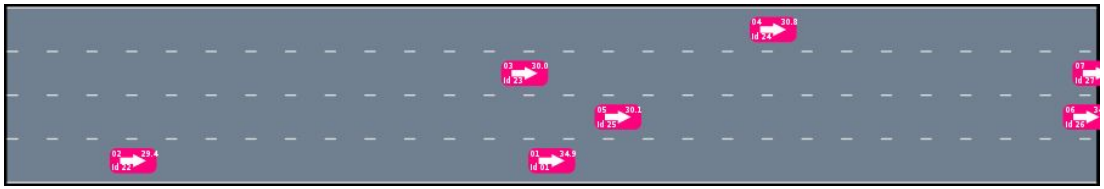
But **What** is a POMDP?

Answer: Almost everything

- Autonomous driving
- Aircraft collision avoidance
- Home assistance
- Medical treatment
- SLAM planning
- Reinforcement learning
- Disaster relief planning

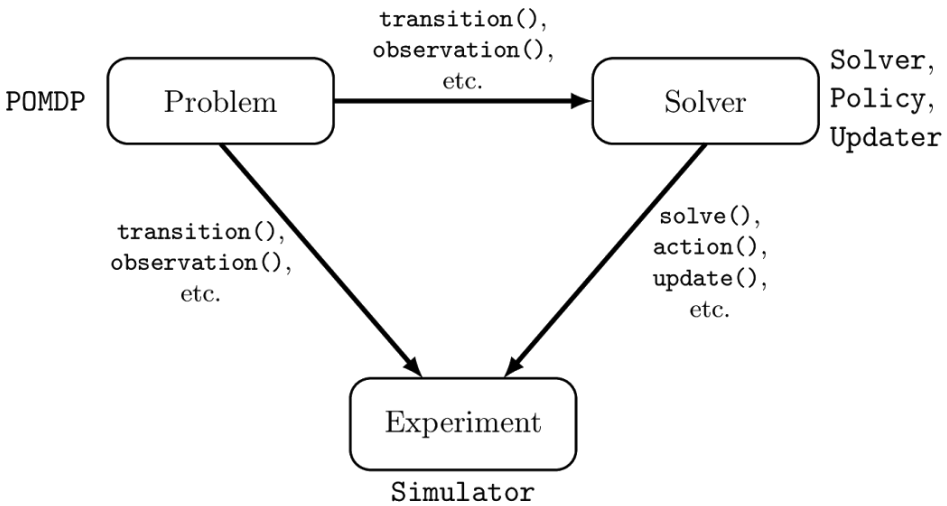






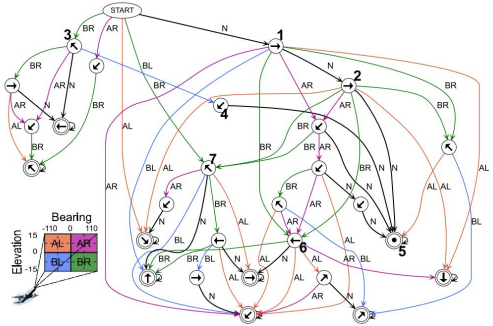
What is POMDPs.jl?

An interface for research collaboration and education.

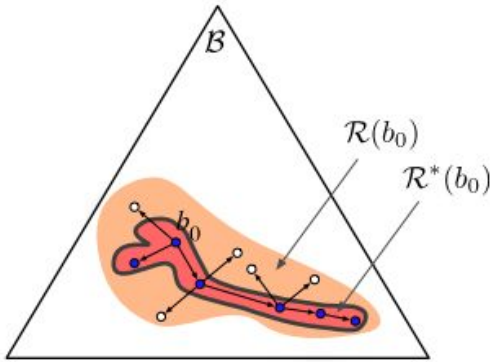


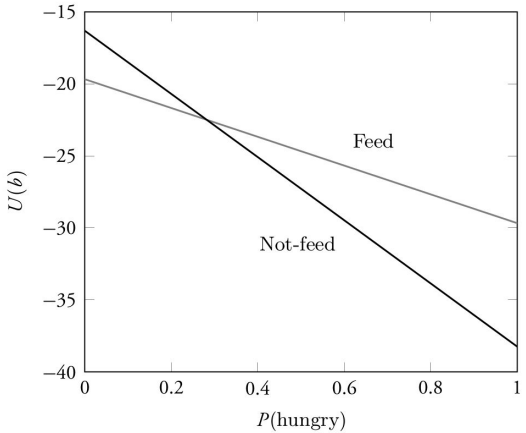
Challenge: Wide variety of problems and solvers

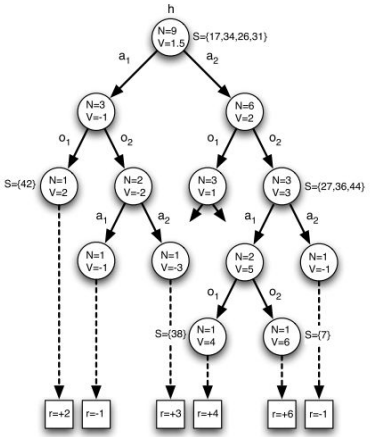
Problems can range from black-box simulators with continuous state and action spaces to small discrete problems with distribution tables.



Solvers can take a variety of forms





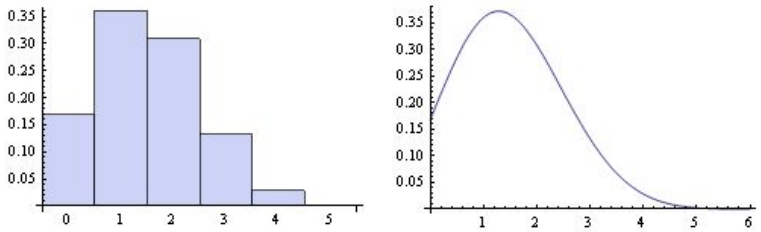


Two Interface Options Explicit and Generative

Explicit

Explicit definition of transition and observation probability distributions

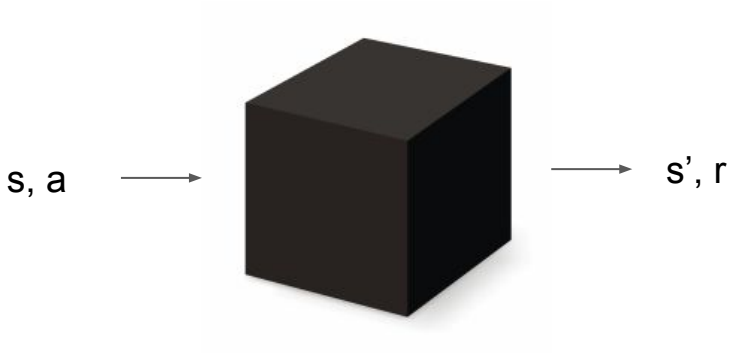
```
dist = transition(pomdp, s, a)
prob_s = pdf(dist, s)
sp = rand(rng, dist)
```



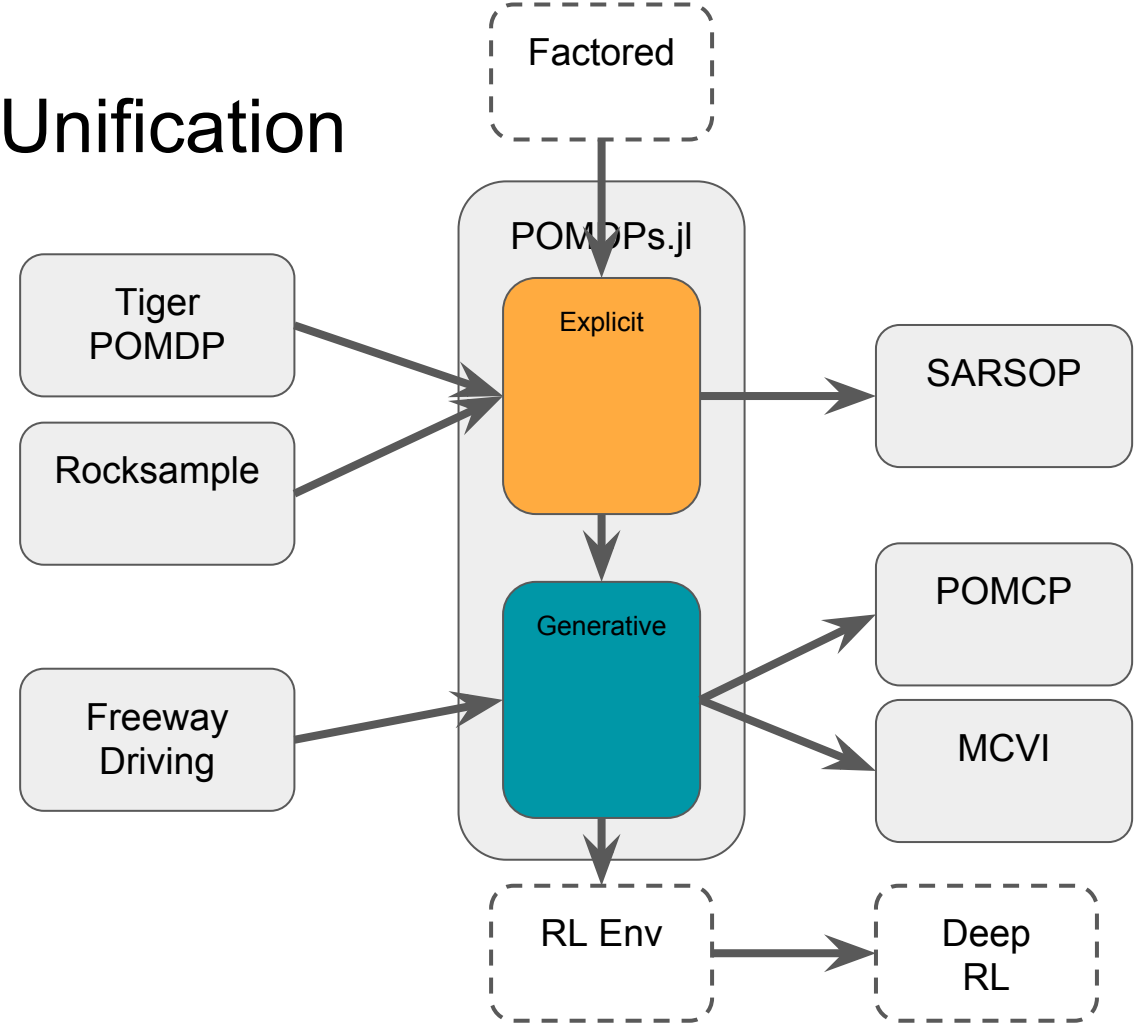
Generative

Generative model similar to OpenAI gym

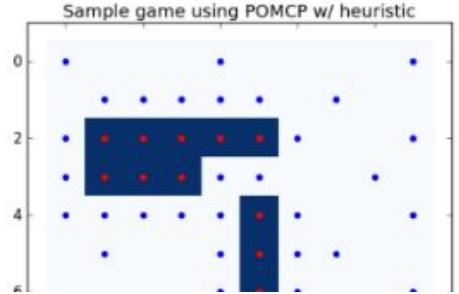
```
sp, r = generate_sr(pomdp, s, a, rng)
```

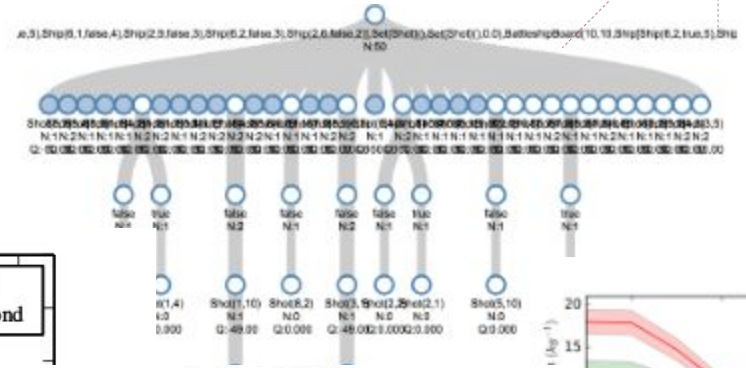


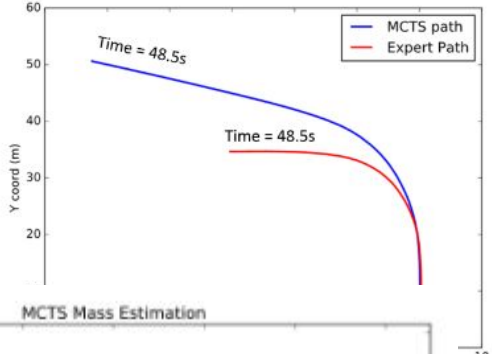
Interface Unification



POMDPs.jl: Use so far







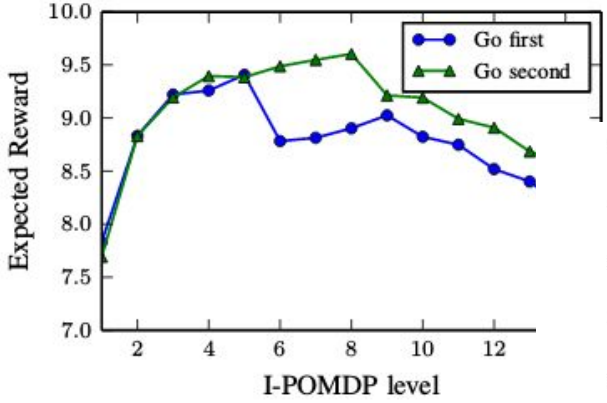


Fig. 3. Expected reward of I-POMDP solution in a PO $N = 3$, $r_w = 10$, and $r_u = -1$. Assumes all opponents at equally likely.