Parameter-Independent Strategies for pMDPs via POMDPs

Sebastian Arming UNIVERSITÄT SALZBURG

Ezio Bartocci TECHNISCHE UNIVERSITÄT WIEN

Krishnendu Chatterjee IST AUSTRIA Institute of Science and Technology

Joost-Pieter Katoen RWTH AACHEN UNIVERSITY

Ana Sokolova UNIVERSITÄT SALZBURG

RiSE Rigorous Systems Engineering

The problem

Finding **policies**

of a **parametric MDP**

that are **optimal**

over the **whole parameter space**.

The problem

Finding **policies**

of a **parametric MDP**

that are **expectation optimal**

(over the **whole parameter space**).

The solution

Finding **policies**

of a **parametric MDP**

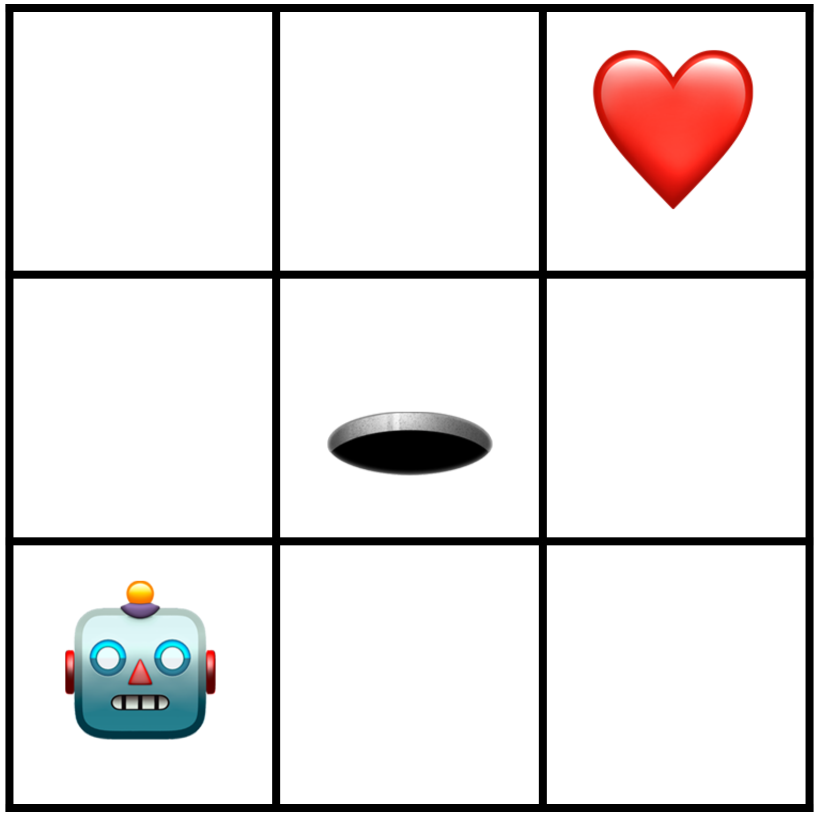
that are **expectation optimal**

(over the **whole parameter space**)

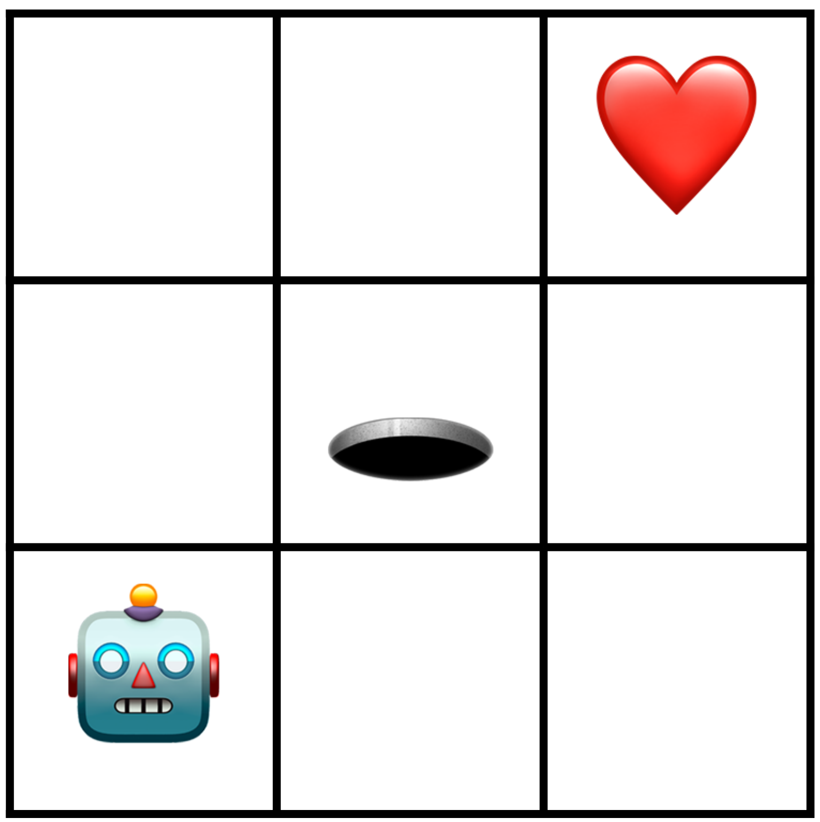
amounts to solving a **suitable POMDP**.

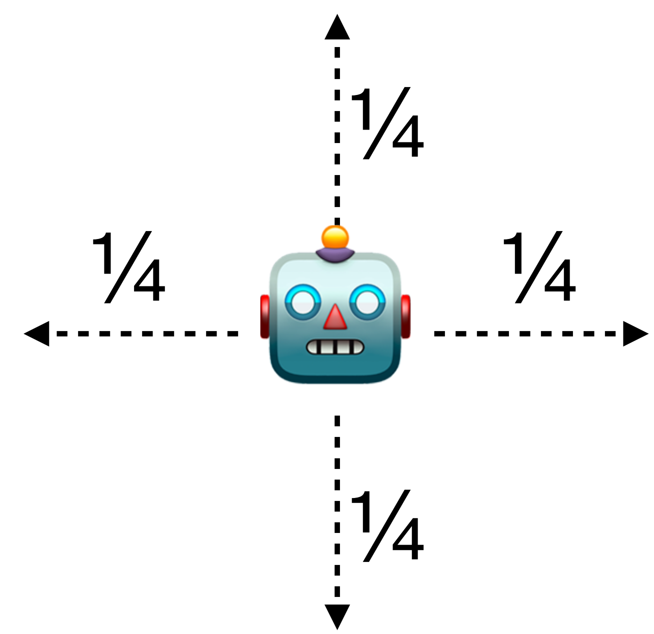
MDP: Markov Decision Process

Robot example

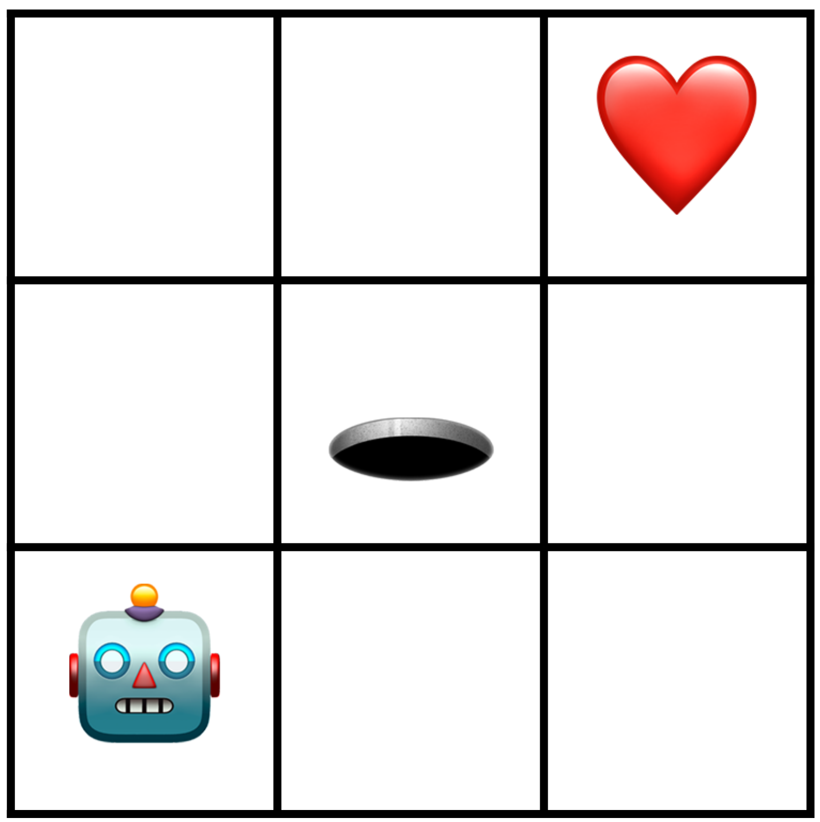


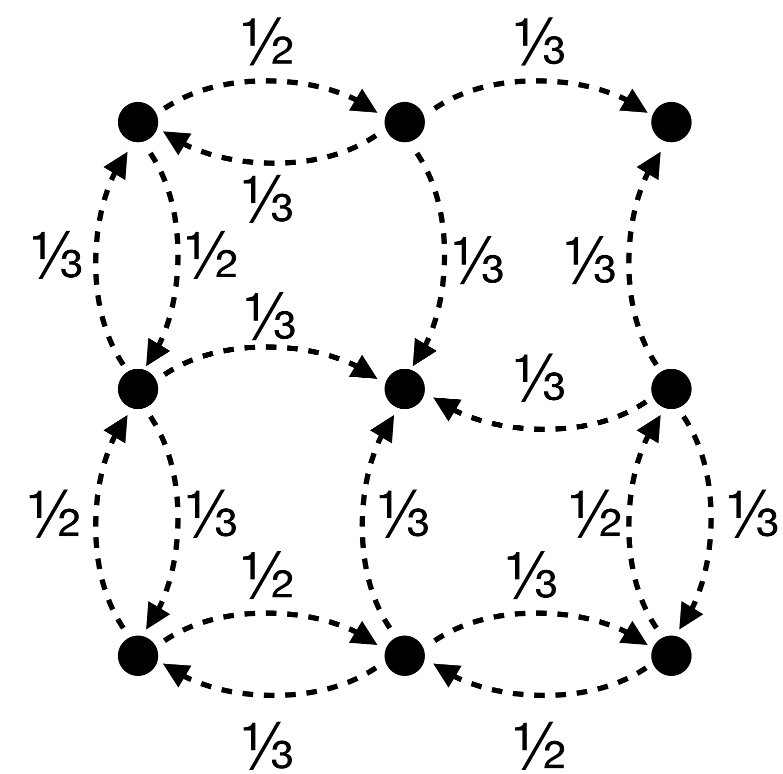
Robot example





Markov Chain MC



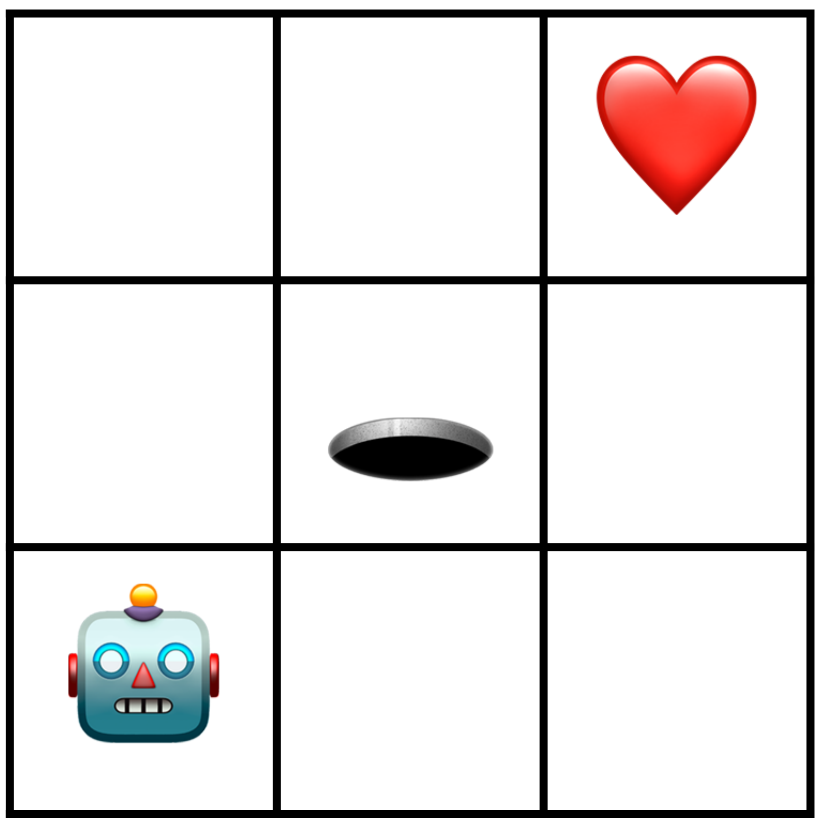


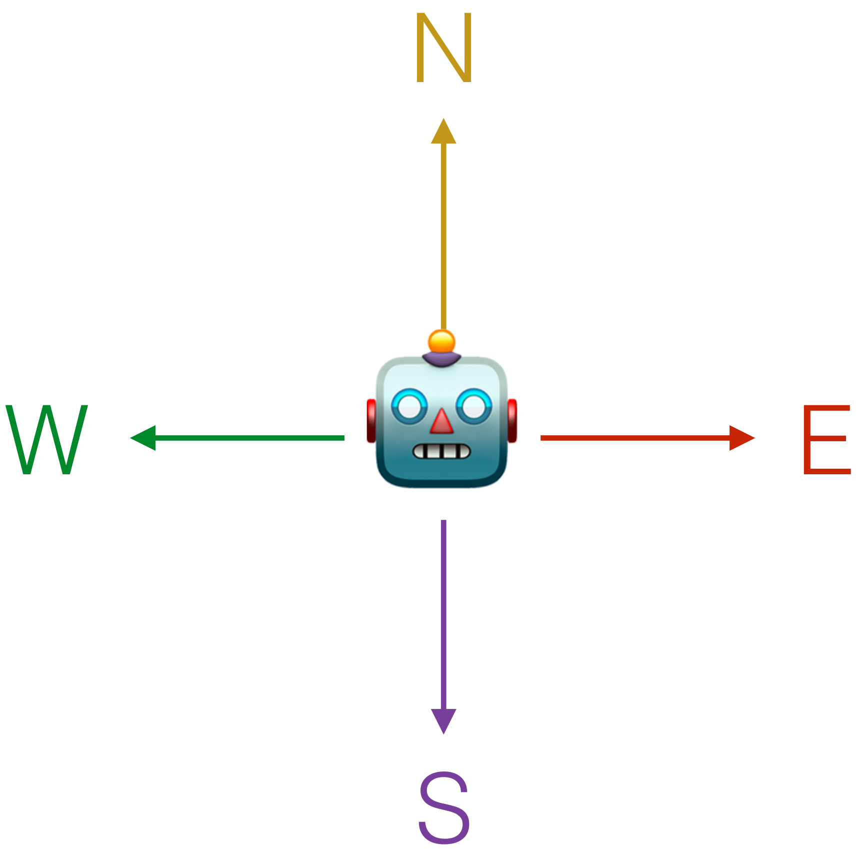
Pr(❤️)=1⁄7

A **MC** is a pair (S,T) where

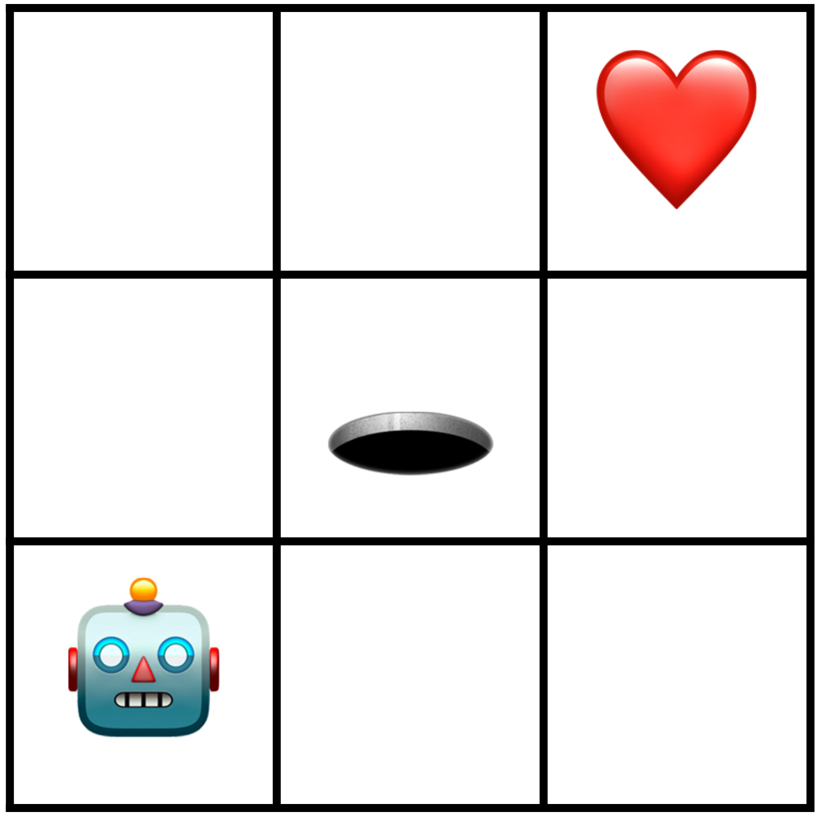
- S is a set of **states**
- $T: S \to \mathscr{D}S$ is a **transition function**

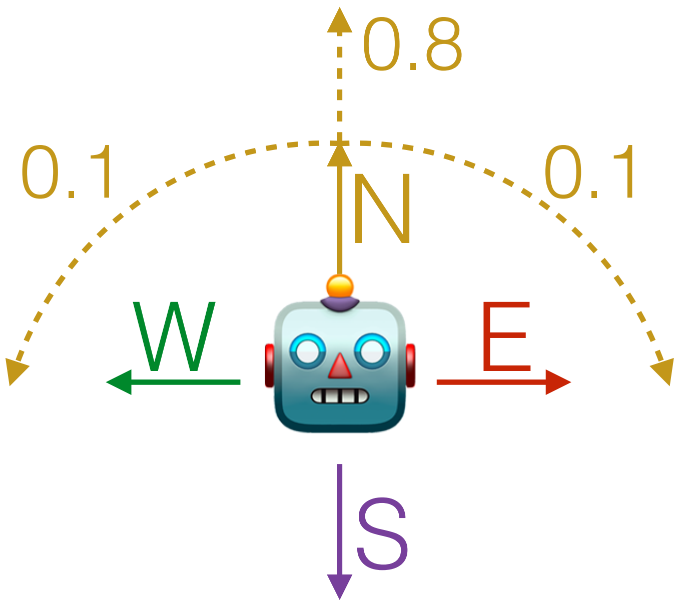
Robot example



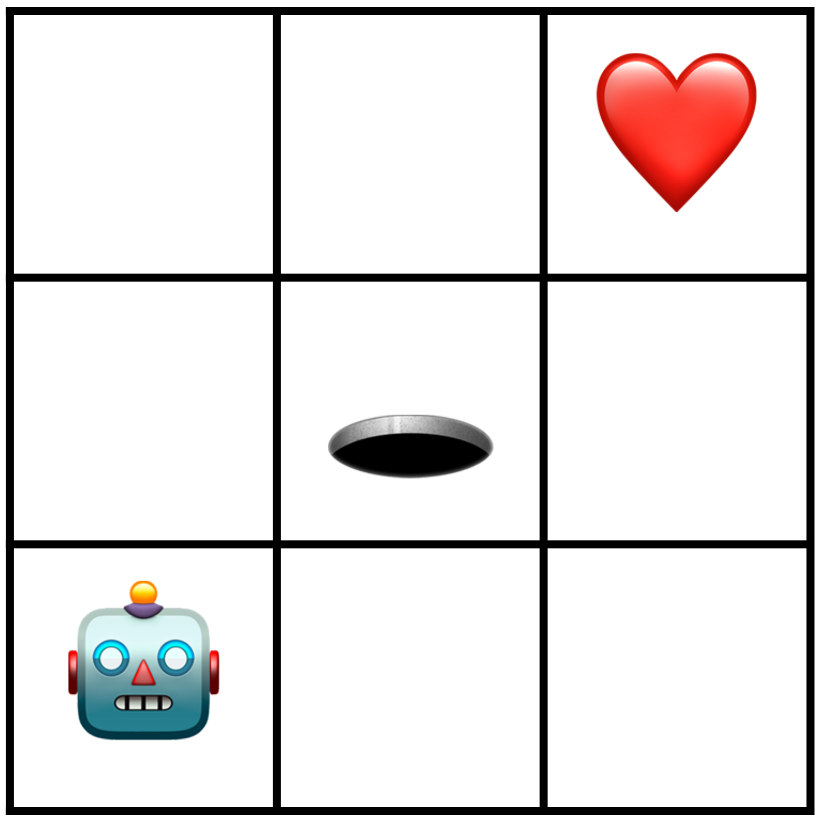


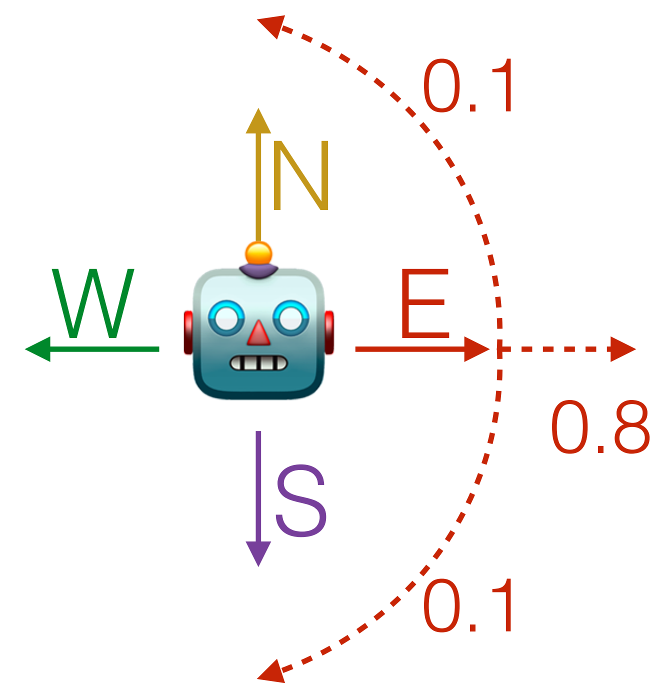
Robot example



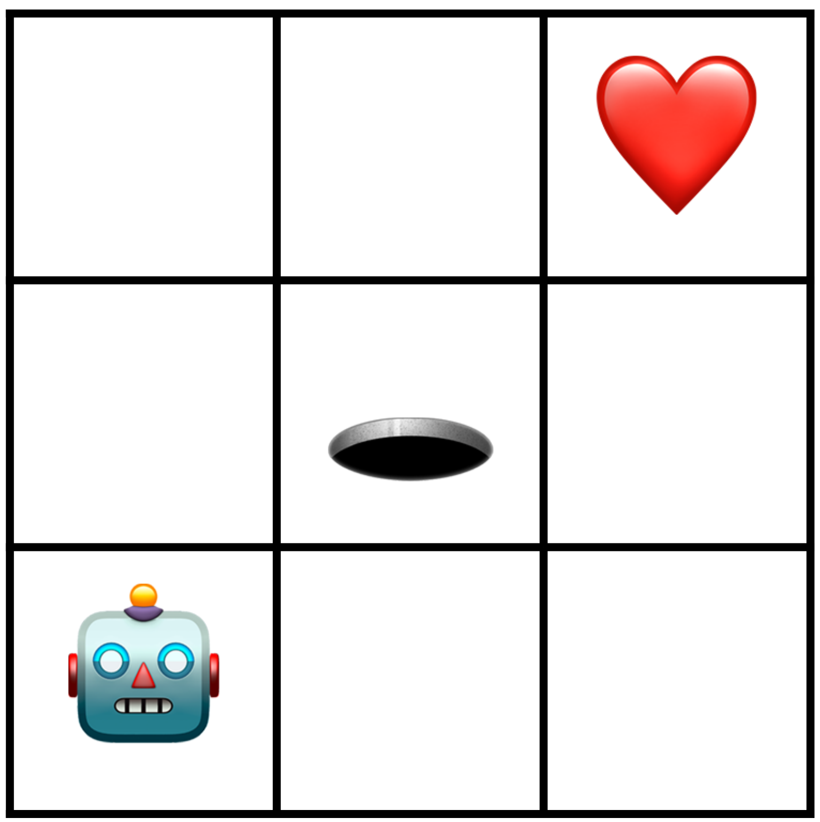


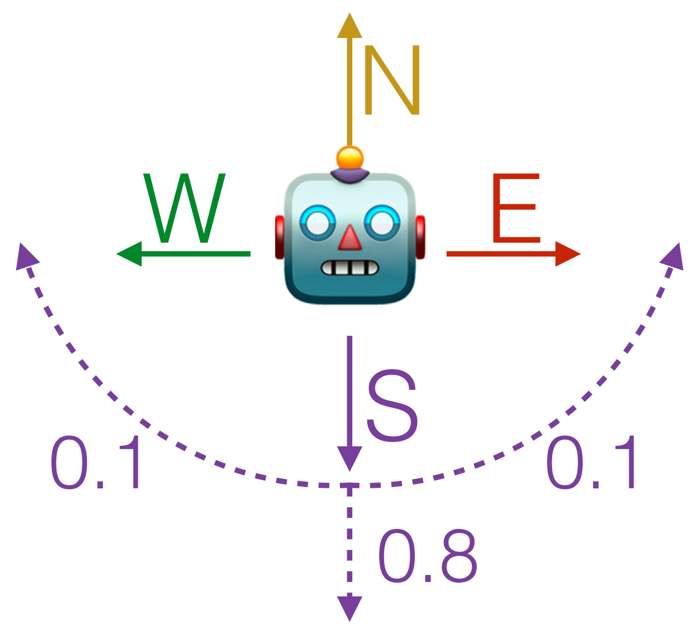
Robot example



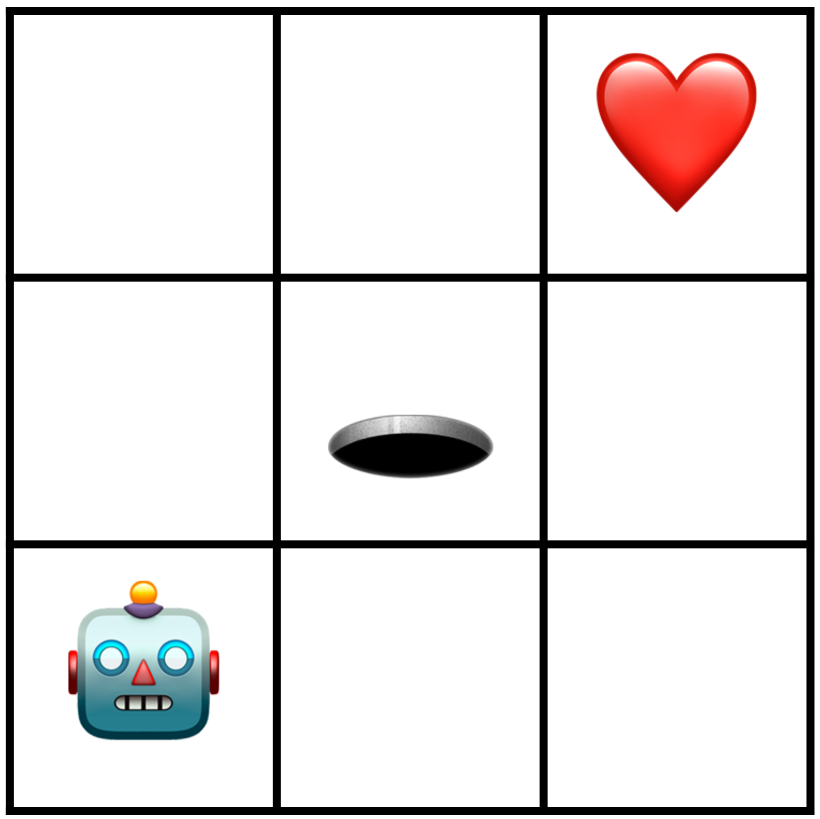


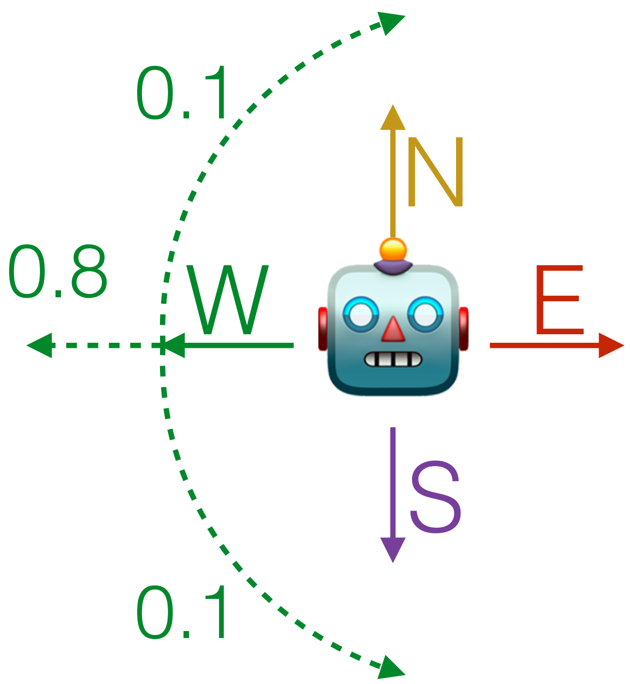
Robot example



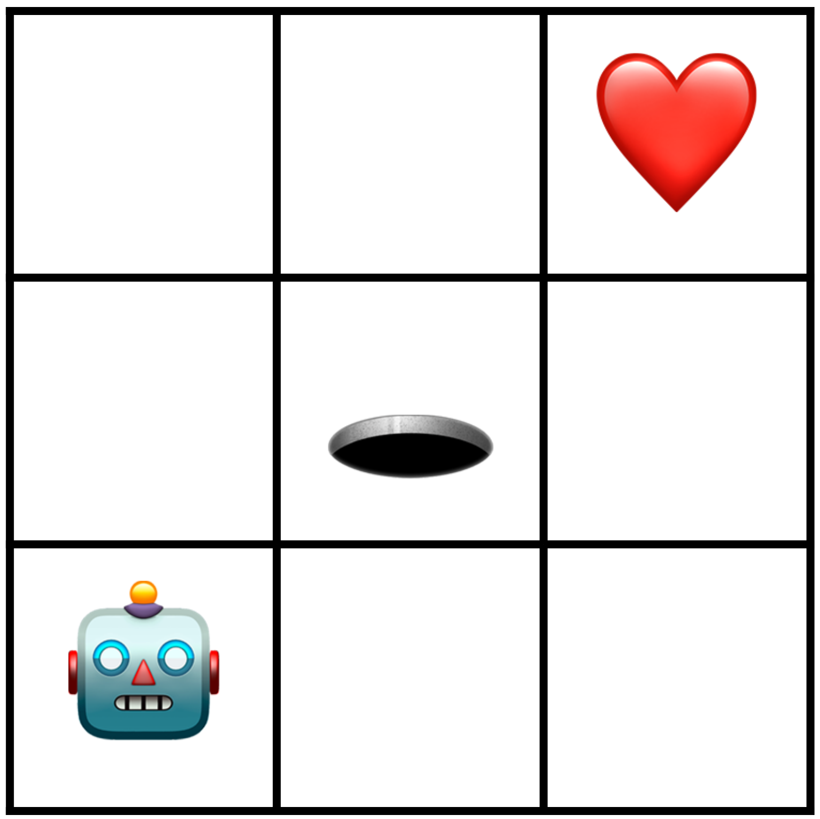


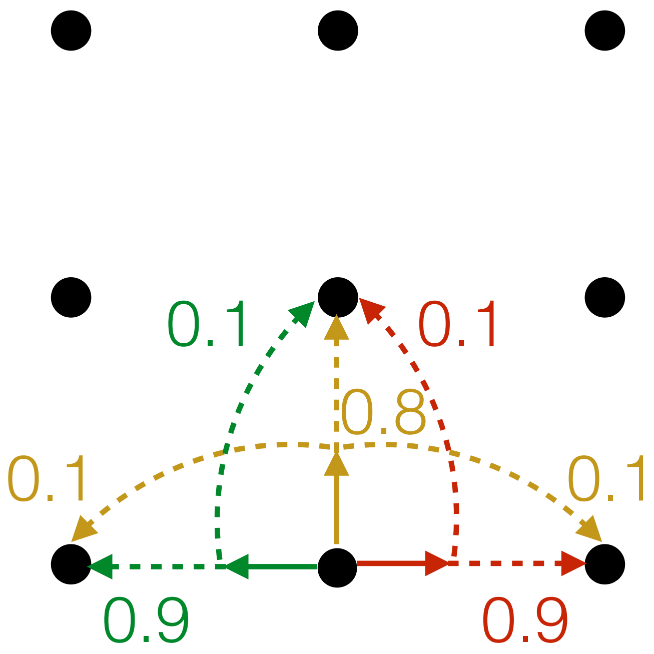
Robot example



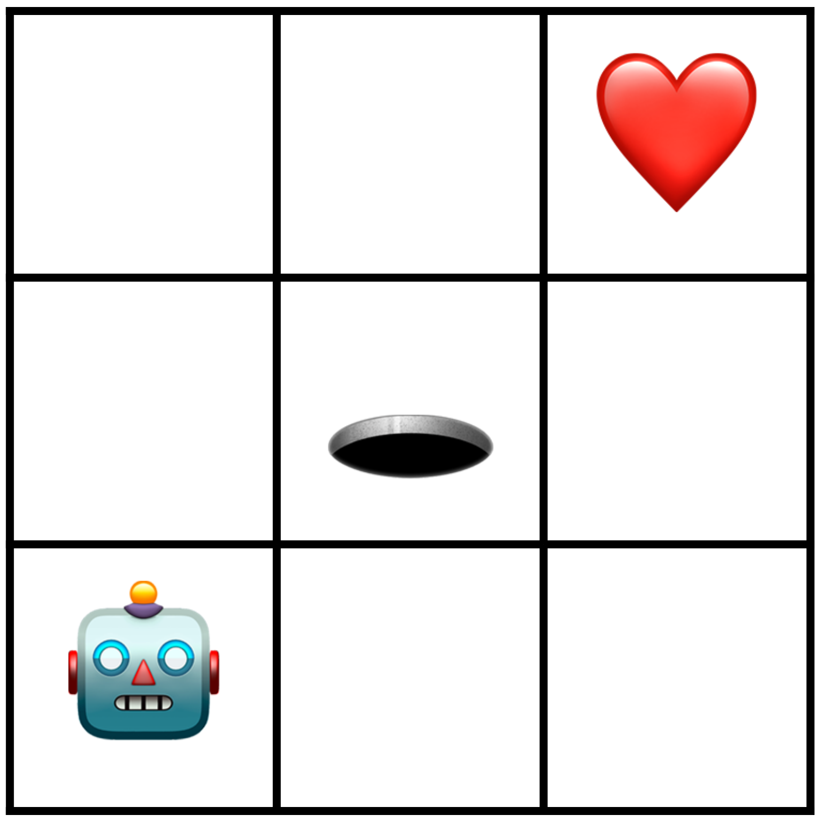


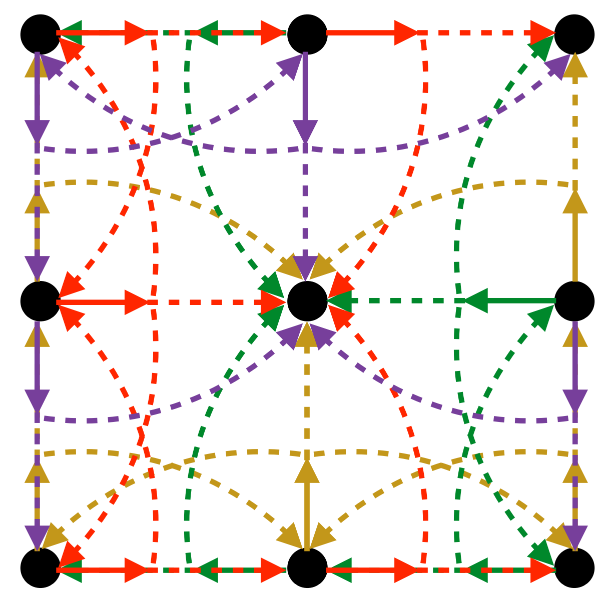
Markov Decision Process MDP





Markov Decision Process MDP

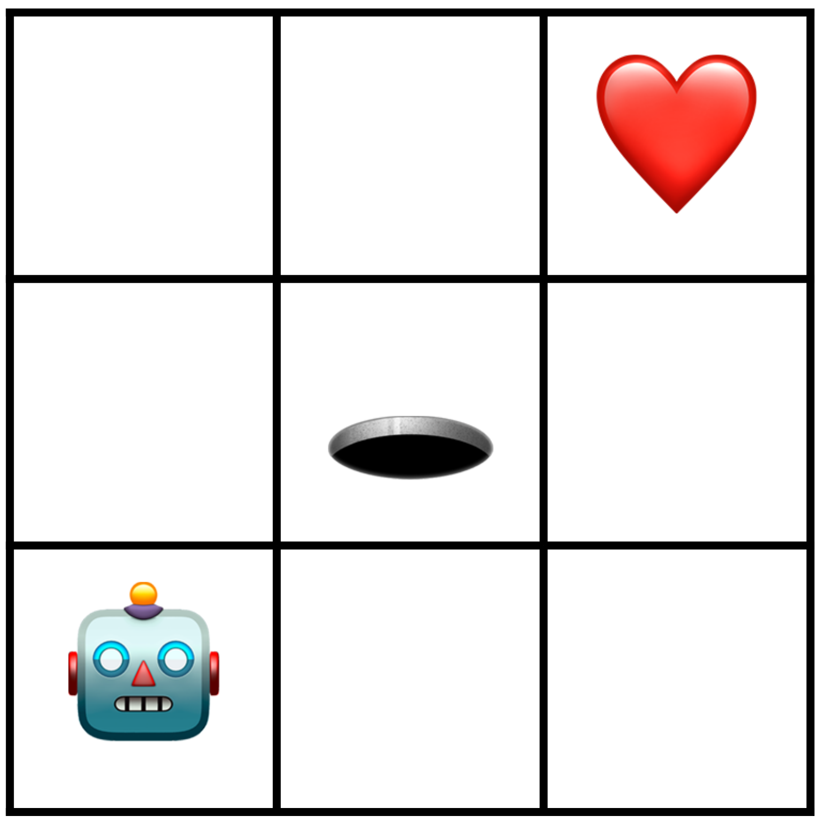


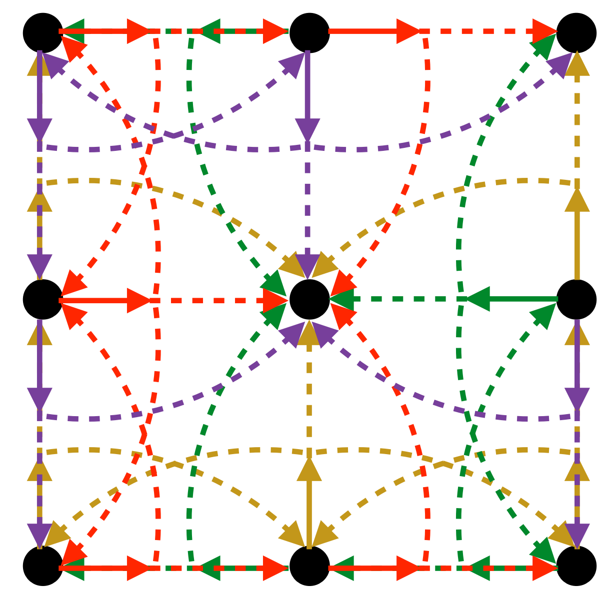


A **MDP** is a tuple *(S,A,T)* where

- *S* is a set of **states**
- *A* is a set of **actions**
- $T: S \times A \rightarrow \mathscr{D}S$ is a **transition function**

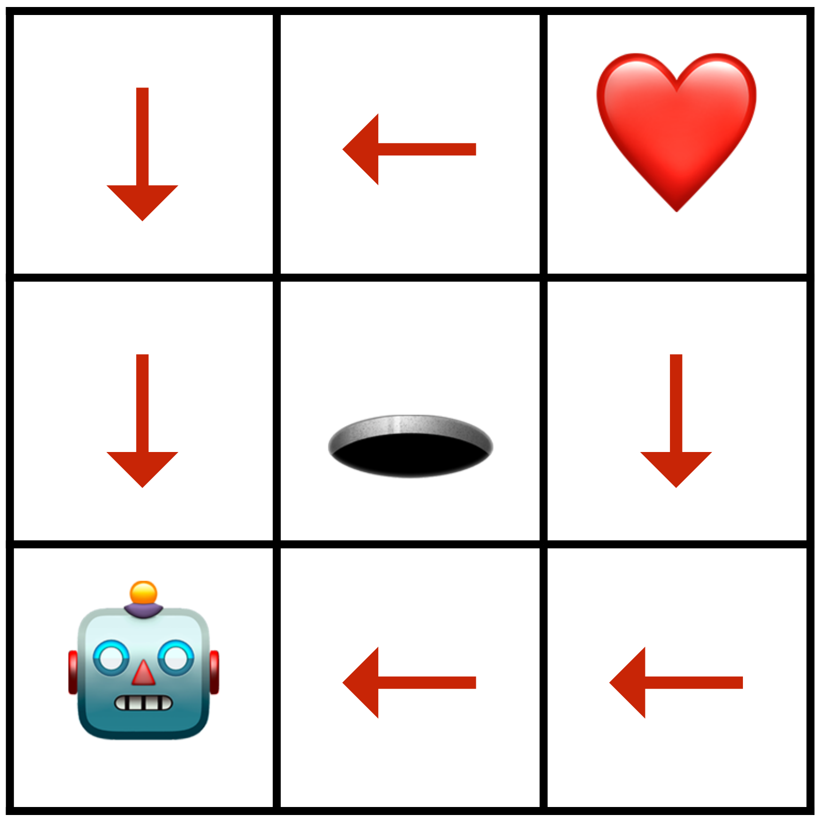
Markov Decision Process MDP

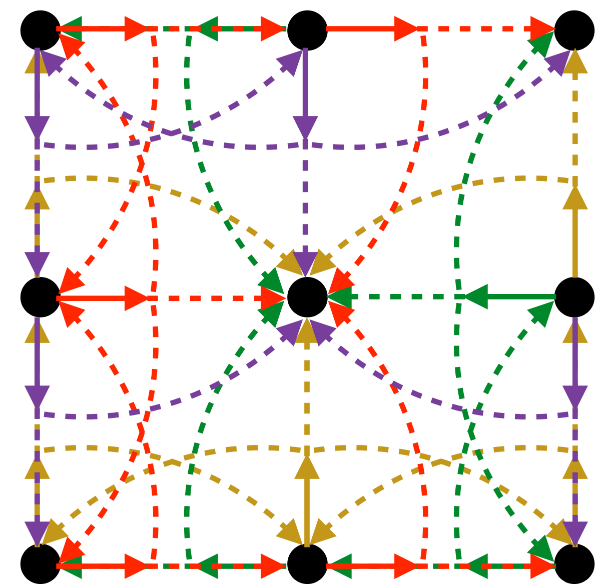




Pr(❤️)= f(🤖)

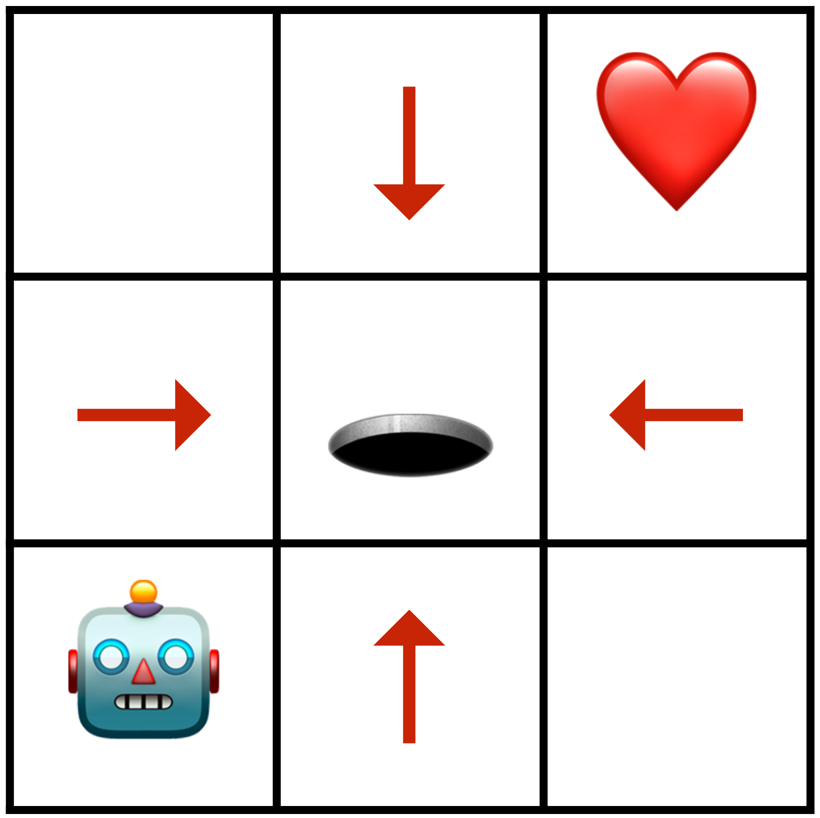
Markov Decision Process MDP

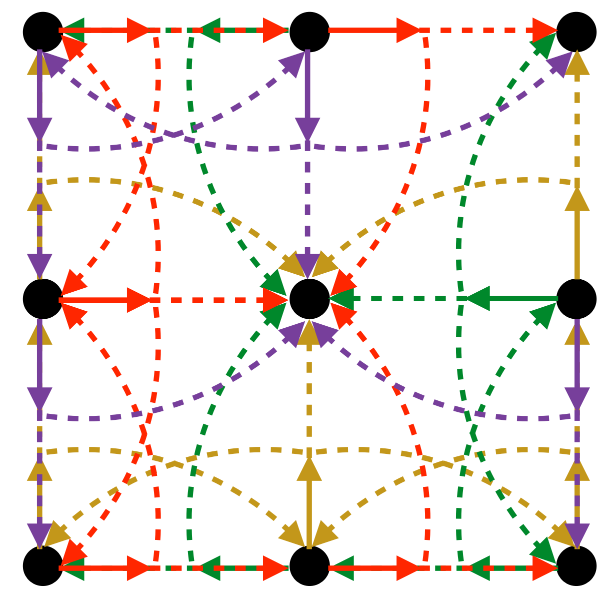




Pr(❤️)= f(🤖)

Markov Decision Process MDP





Pr(❤️)= f(🤖)

Policy

A **policy** π for an MDP (S, A, T) is a function

$$\pi: (S \times A)^* \times S \rightarrow \mathscr{D}A$$

deterministic: only Dirac distributions $\pi: (S \times A)^* \times S \rightarrow A$

memoryless: $\pi(\ldots s) = \pi(s)$ $\pi: S \rightarrow \mathscr{D}A$

simple: deterministic & memoryless $\pi: S \rightarrow A$

Wanted: a policy that optimizes an objective, e.g. reachability

Objective

A policy π and initial distribution *i* gives us a probability space

$$(\text{Runs}, \text{Cones}, \mathbf{P}_{\pi,i})$$

in the usual way.

Runs - all infinite runs in $(S \times A)^\omega$

Cones - the σ algebra generated by the sets of runs with a common finite prefix (history)

$\mathbf{P}_{\pi,\text{i}}$ - the usual measure on cones

A **Borel objective** is a measurable function

$$r: \text{Runs} \rightarrow \mathbb{R}$$

Objective

A **Borel objective** is a measurable function

$$r: \text{Runs} \rightarrow \mathbb{R}$$

Rewards R: $S \times A \rightarrow \mathbb{R}$ induce Borel objectives via

$$r_R(s_0,a_0,s_1,a_1,\ldots) = \sum_{i\geq 0} R(s_i,a_i)$$

Reachability objectives are a special case:

Reachability probability = Expectation of reachability objective

Optimal policy

An **optimal policy** is a policy π with

$$\mathbf{E}_{\pi,i}(r) = \sup_\sigma \mathbf{E}_{\sigma,i}(r)$$

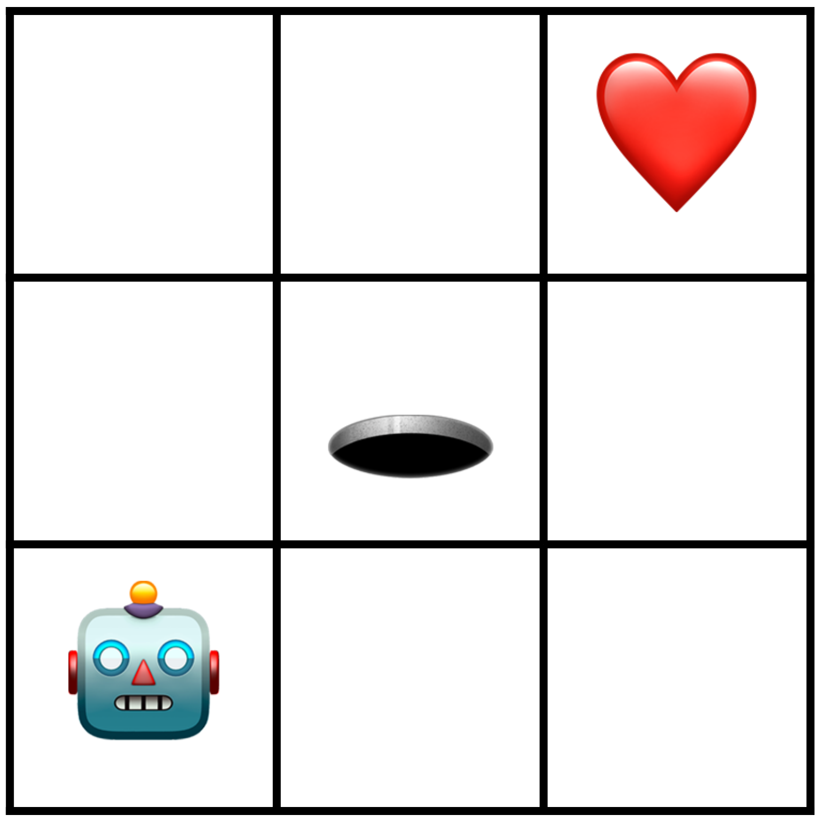
The **expectation** of π

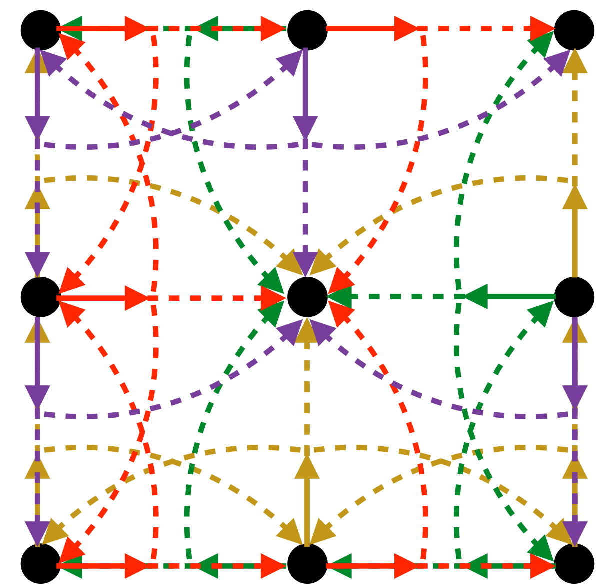
The **value** of r

An optimal policy not always exists, but **ε-optimal** do:

A policy is **ε-optimal if** $\mathbf{E}_{\pi,i}(r)$ is ε-close to $\sup_\sigma \mathbf{E}_{\sigma,i}(r)$

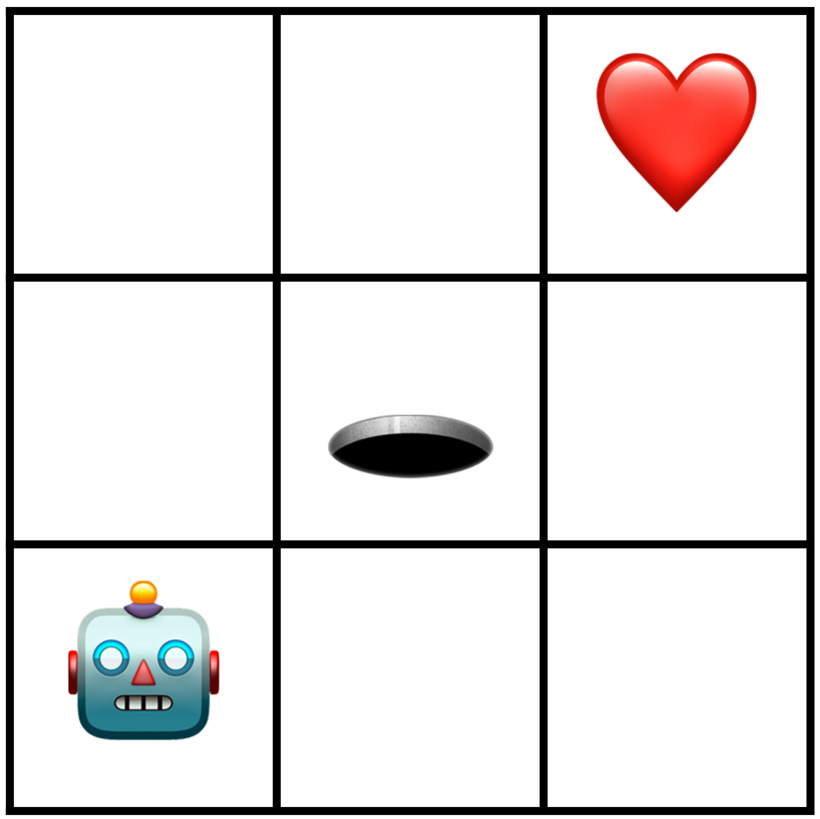
Markov Decision Process MDP

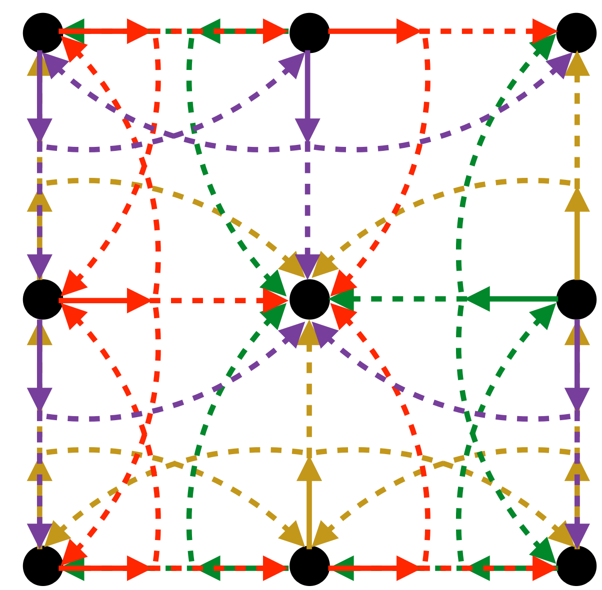




Pr(❤️)$= 0, \frac{3}{19}, \frac{4}{15}, \frac{3}{13}, \frac{3}{25}, \frac{9}{40}, \frac{1}{8}, \frac{18}{55}, \frac{9}{52}, \frac{9}{100}, \frac{27}{95}$

Markov Decision Process MDP



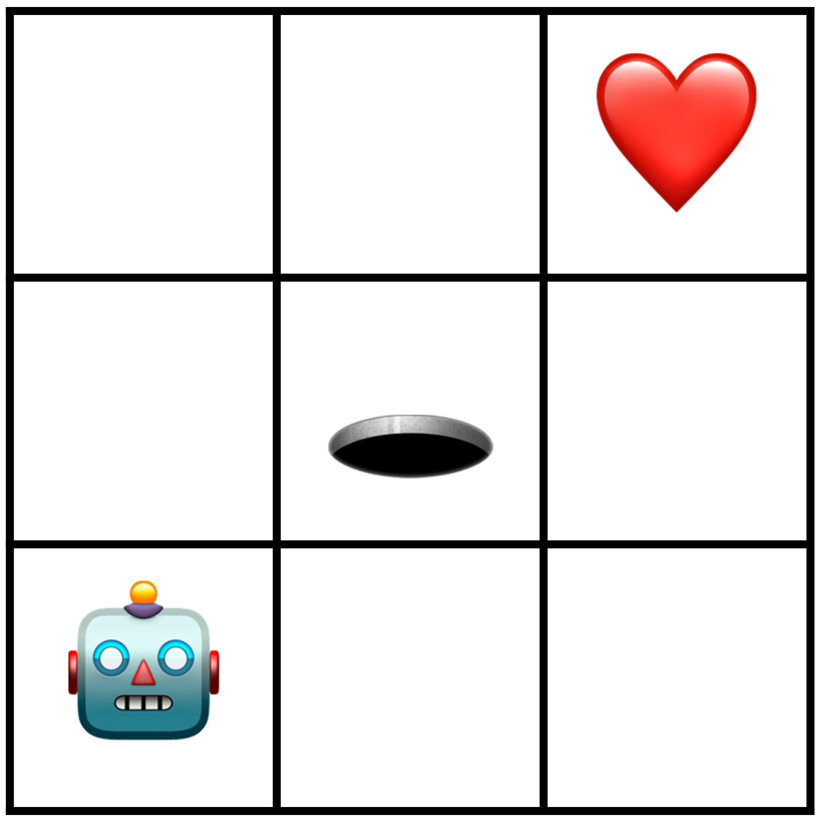


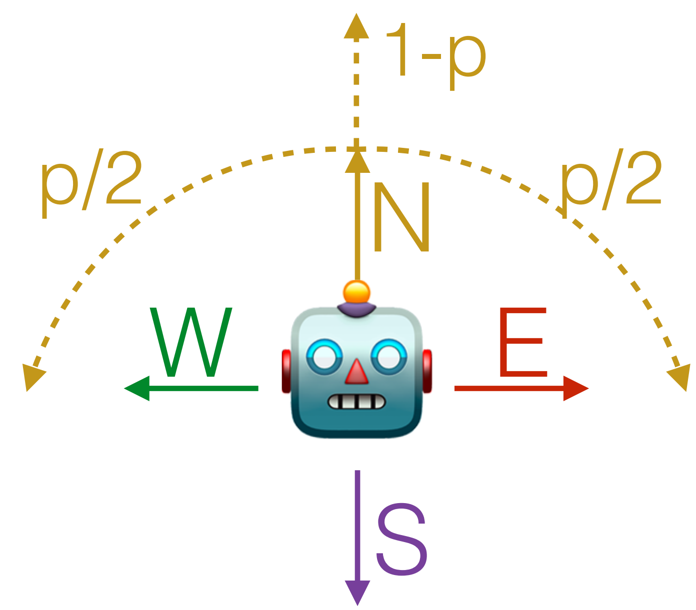
Pr$_{\textbf{sup}}$(❤️)= 18/55

Parameters

pMDP: parametric MDP

Parametric Markov Decision Process pMDP

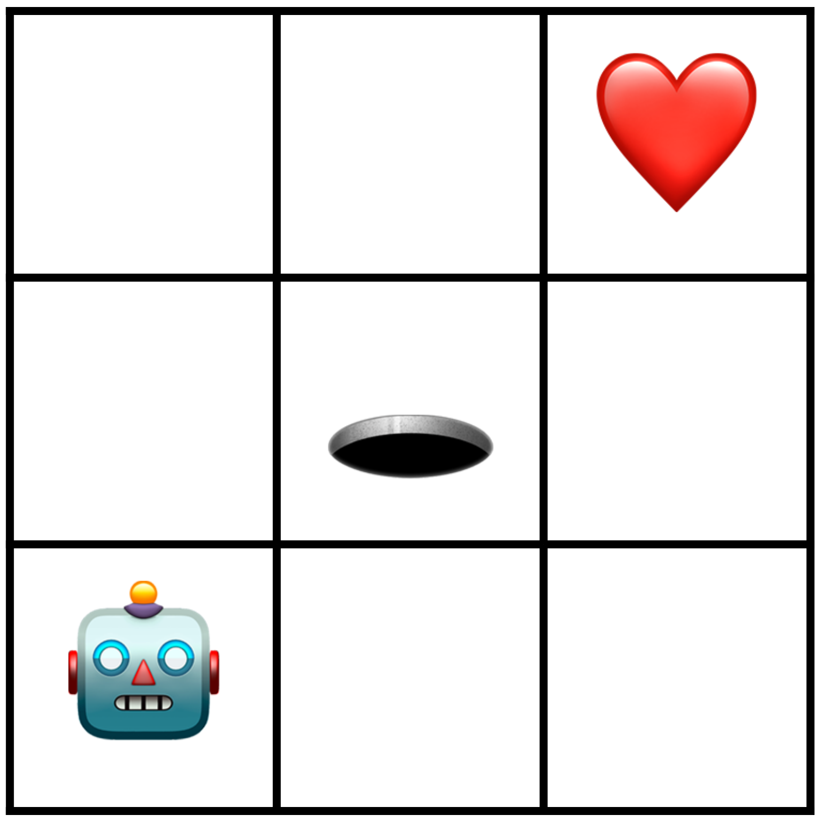


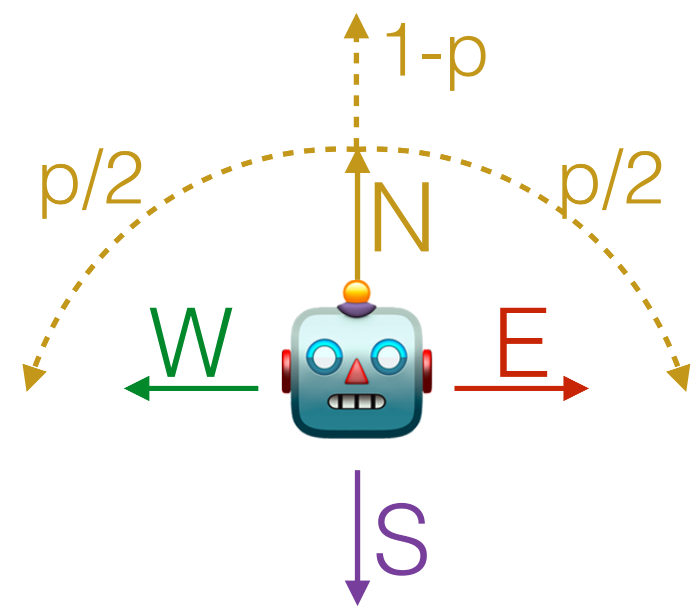


A **pMDP** is a tuple (S,A,X,T) where

- S and A are states and actions
- X is a a **parameter space**
- $T: S \times A \times X \rightarrow \mathscr{D}S$

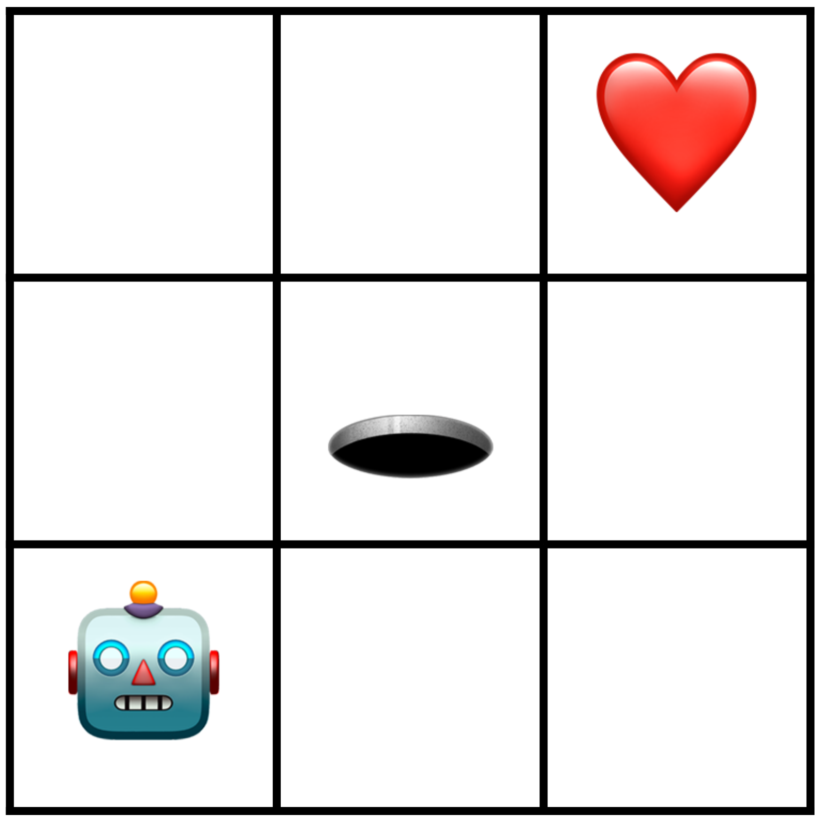
Parametric Markov Decision Process pMDP

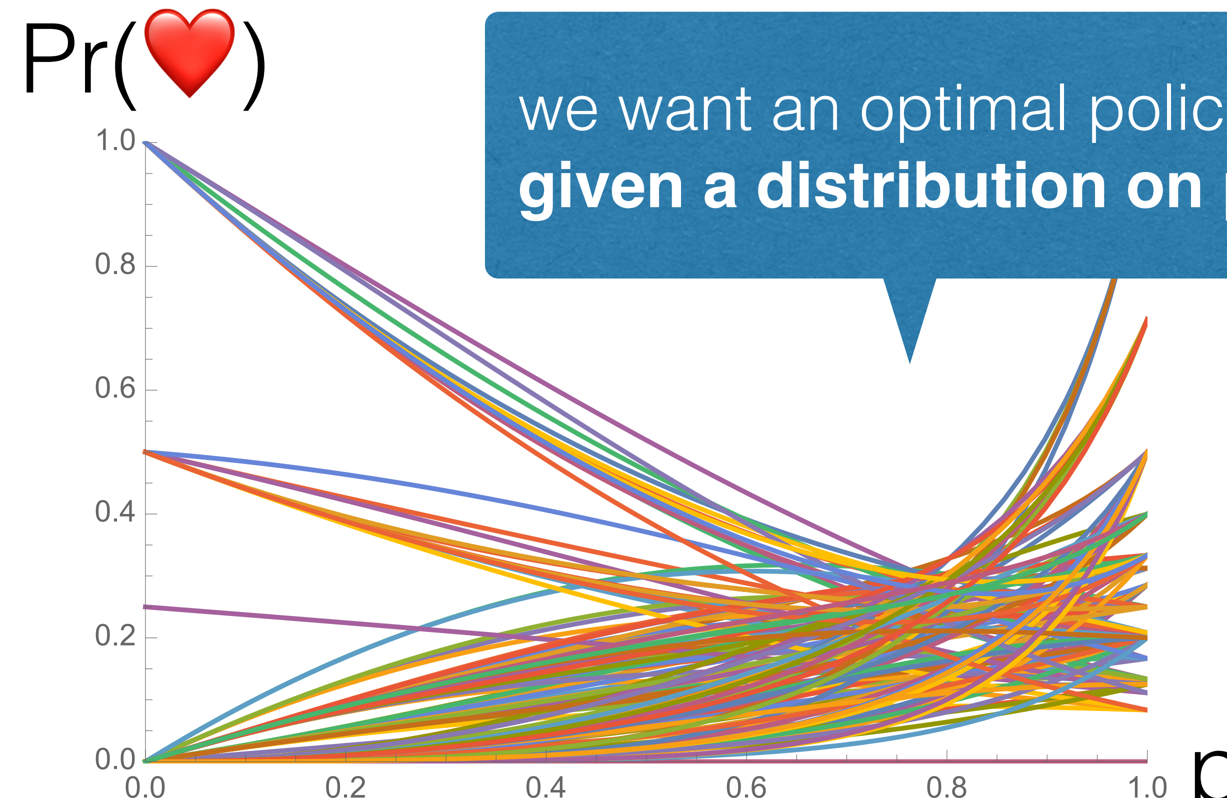




Pr(❤️)= $0, \frac{2p-p^2}{2p^2-4p+8}, -\frac{p^2}{2p-4}, \frac{2p-p^2}{4p^2-8p+8}, \frac{1}{8}, (2$

Parametric Markov Decision Process pMDP

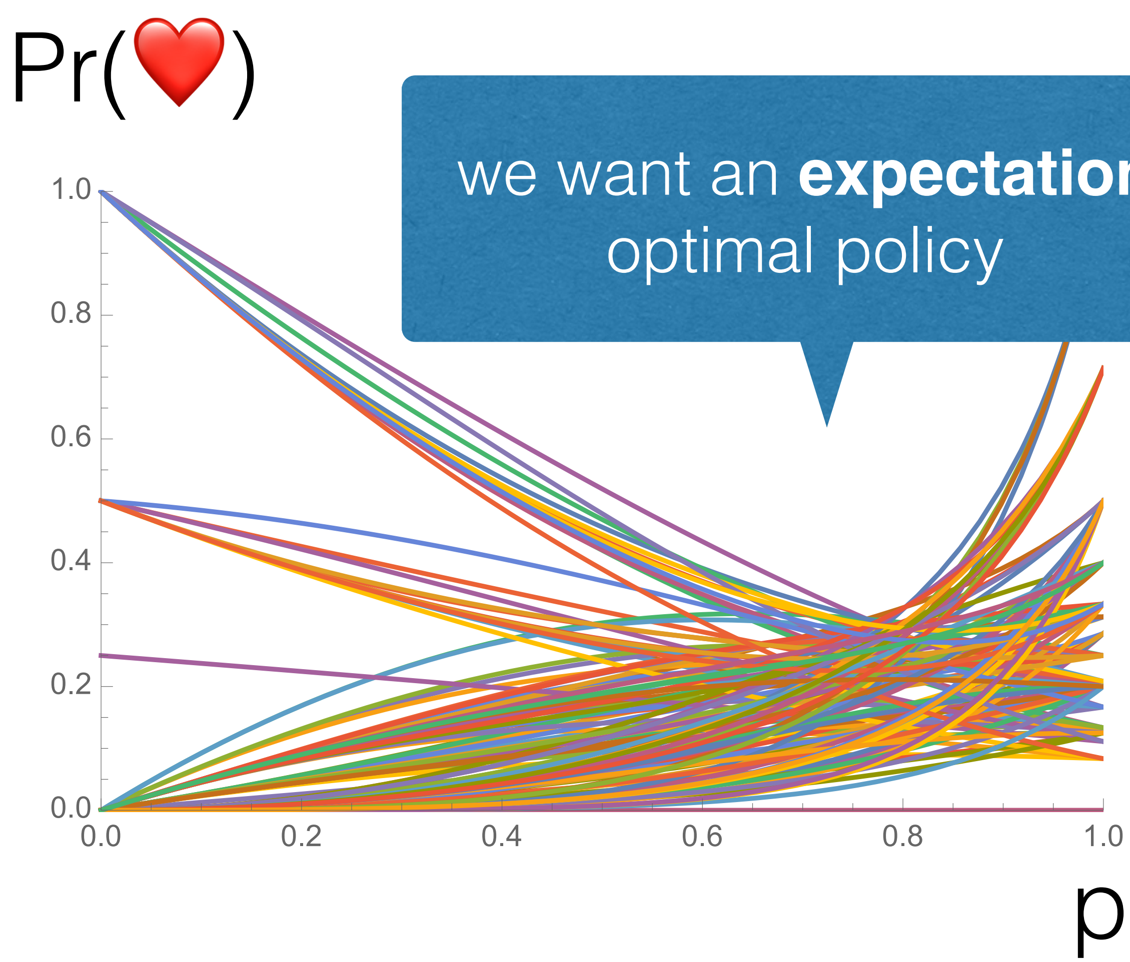




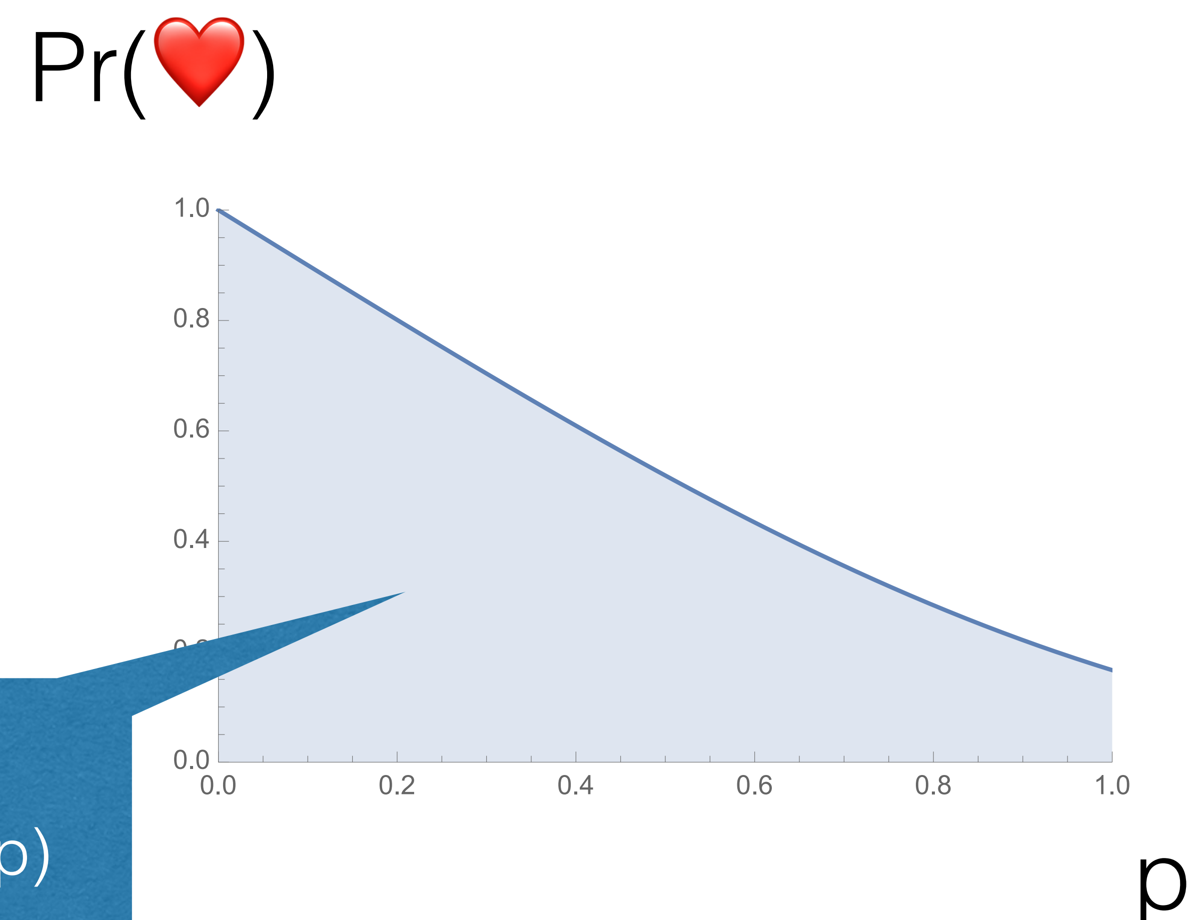
we want an optimal policy **given a distribution on p**

$$\textbf{Pr}(❤️) = 0, \frac{2p-p^2}{2p^2-4p+8}, -\frac{p^2}{2p-4}, \frac{2p-p^2}{4p^2-8p+8}, \frac{1}{8}, (2$$

Expectation optimality



we want an **expectation** optimal policy



optimizes the area
(for uniform distribution on p)

Expectation optimality

A policy π, initial distribution *i* and a distribution over the parameters *d*, gives us a parametric probability space

$(\text{Runs}_X, \text{Cones}_X, \mathbf{P}_{\pi,i,d})$.

Runs_X - disjoint union of the runs for all parameter values

Cones_X - the σ algebra generated by the disjoint union of cones for all parameter values

$\mathbf{P}_{\pi,i,d}$ - the d-convex combination of the individual measures on cones

Expectation optimal **policy**

An **expectation optimal policy** is a policy π with

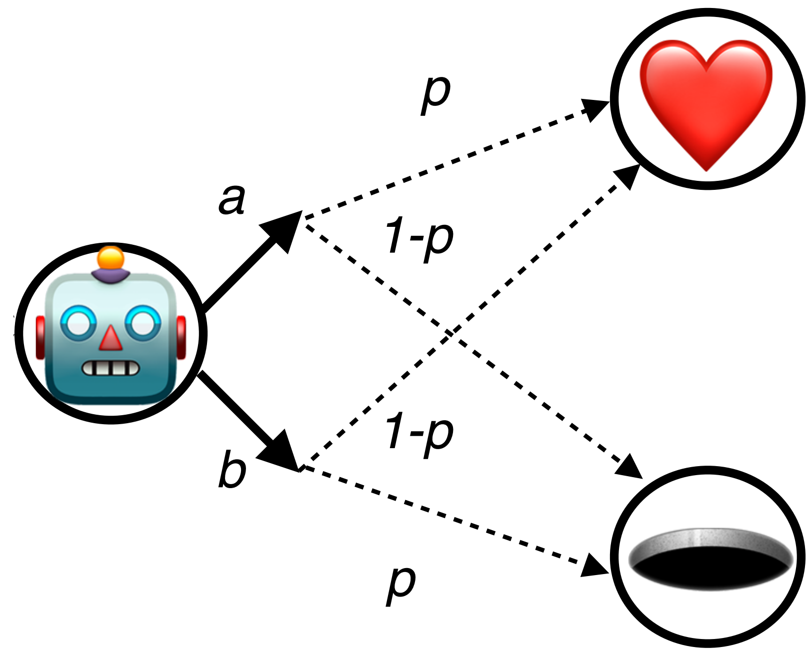
$$\mathbf{E}_{\pi,i,d}(r) = \sup_{\sigma} \mathbf{E}_{\sigma,i,d}(r)$$

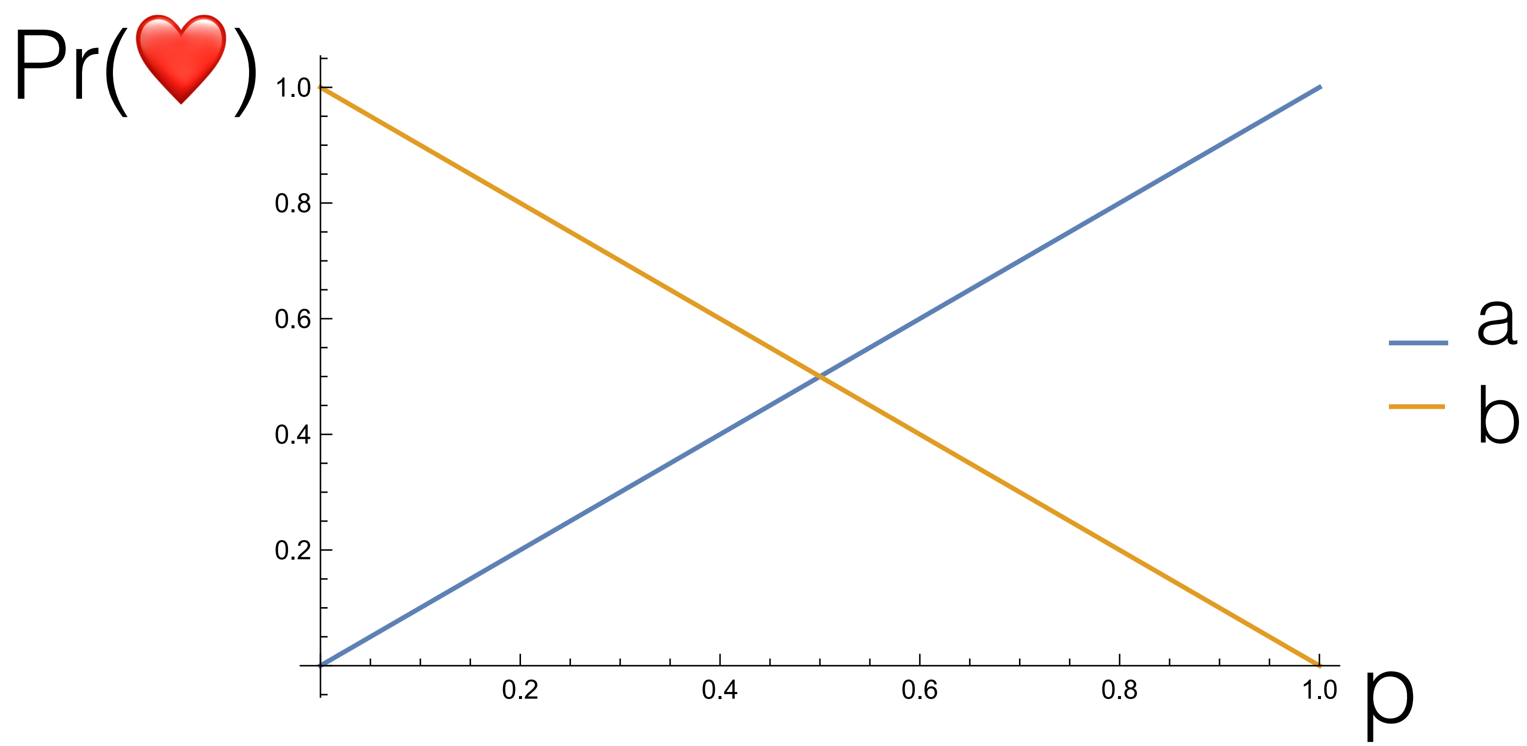
An optimal policy not always exists, but **ε-optimal** do:

A policy is **expectation ε-optimal** if

$$\mathbf{E}_{\pi,i,d}(r) \text{ is } \varepsilon\text{-close to } \sup_{\sigma} \mathbf{E}_{\sigma,i,d}(r)$$

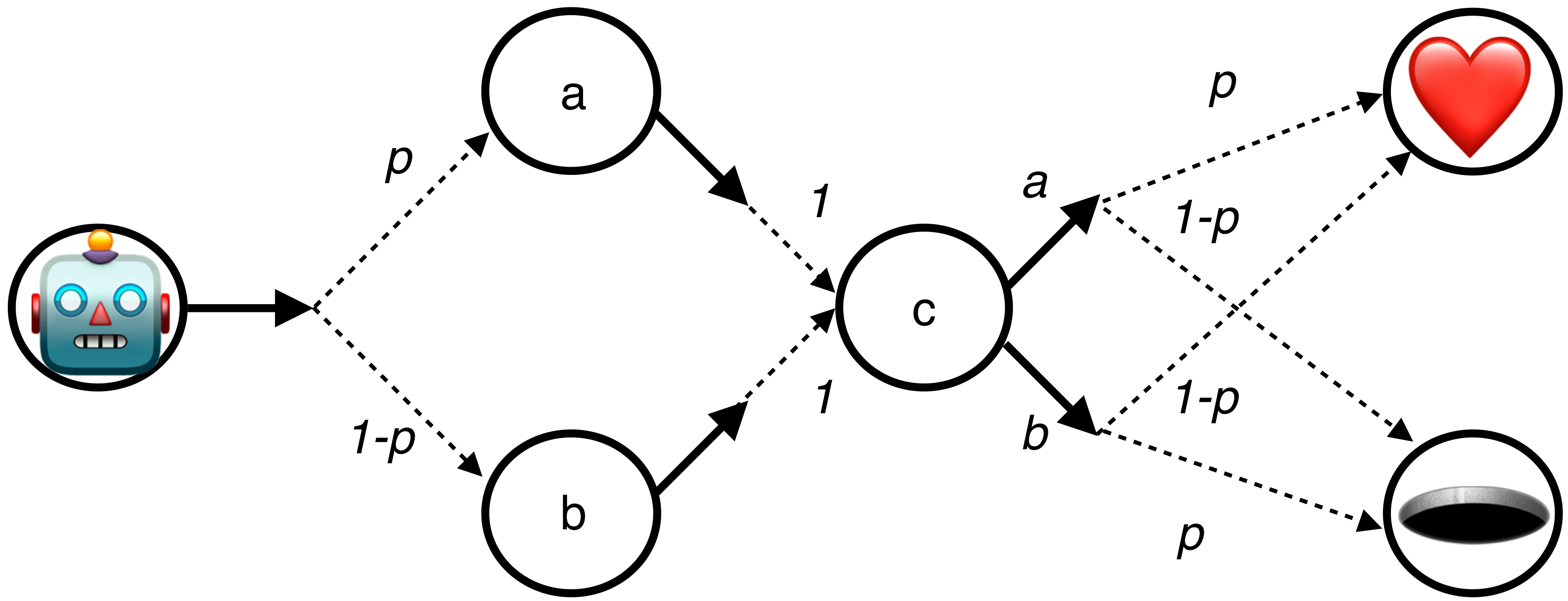
Learner Example

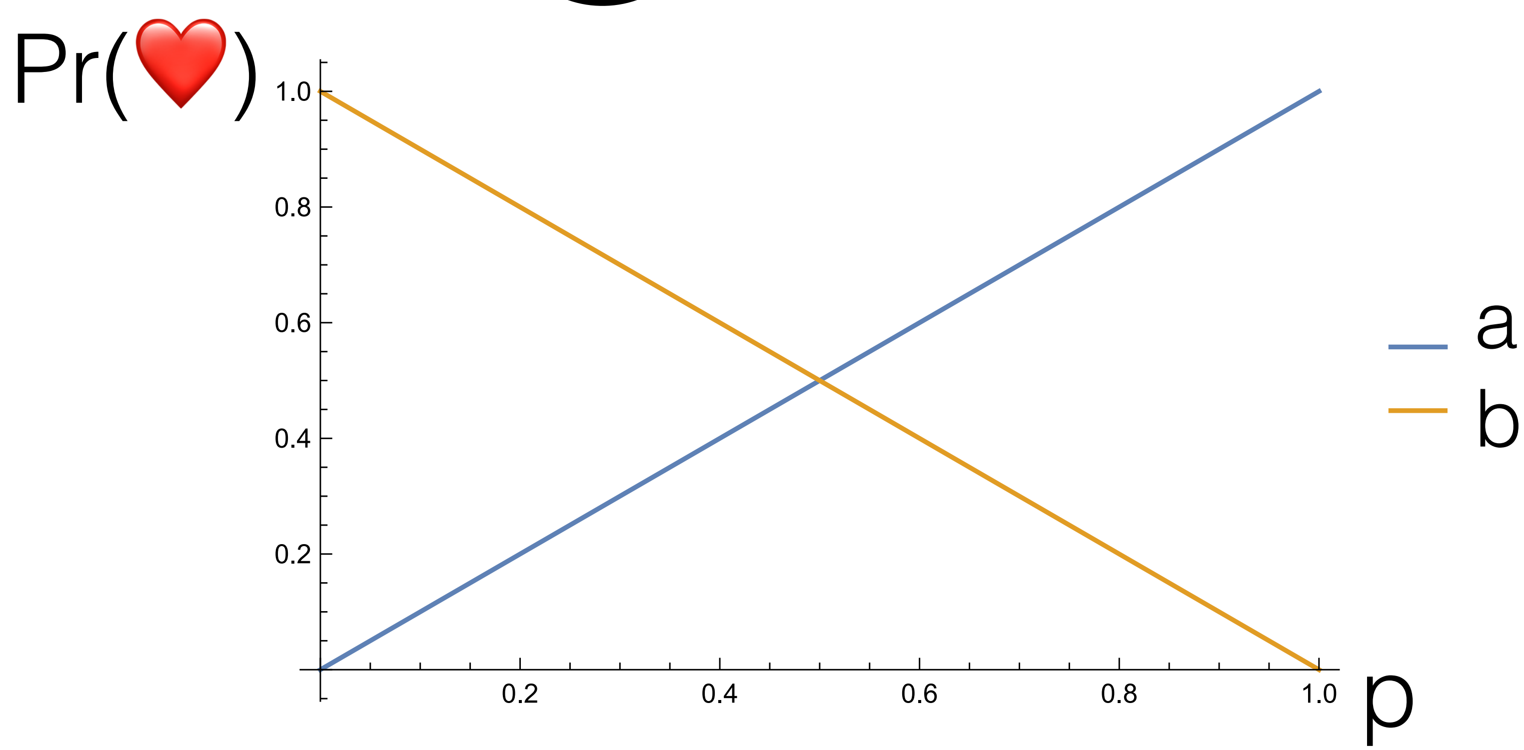




simple policies

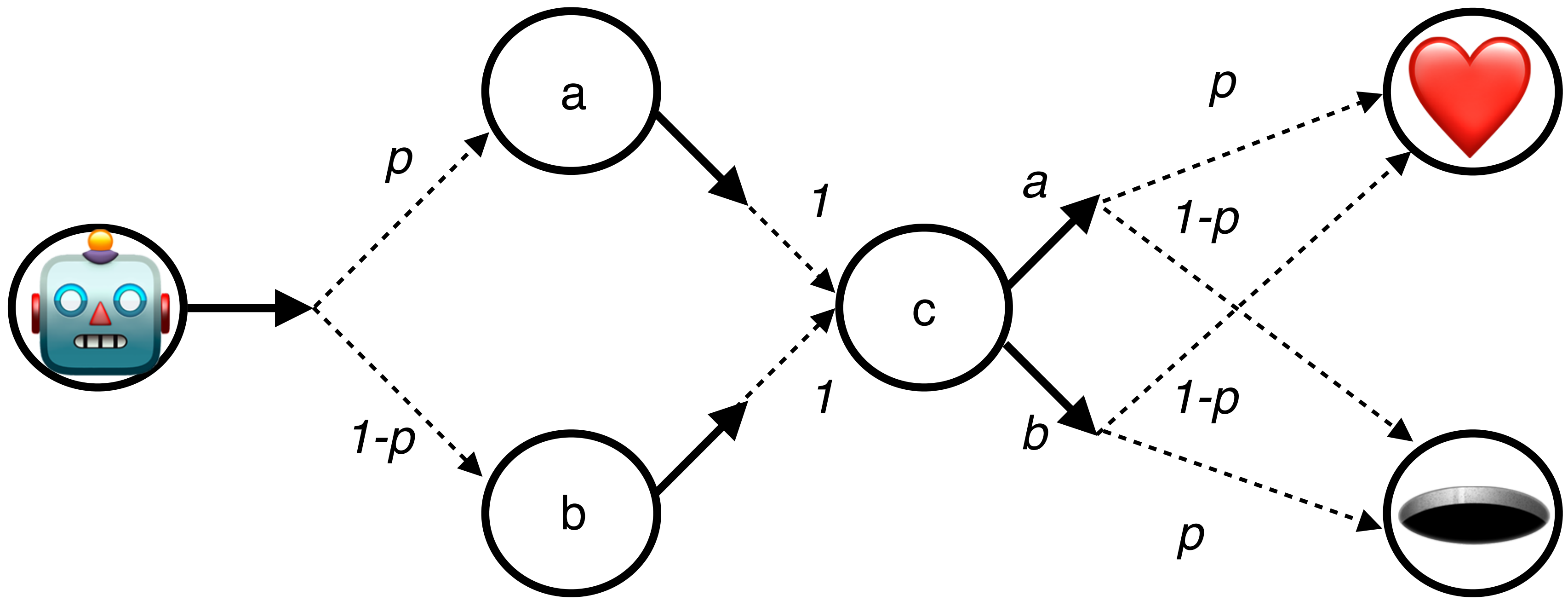
Learner Example

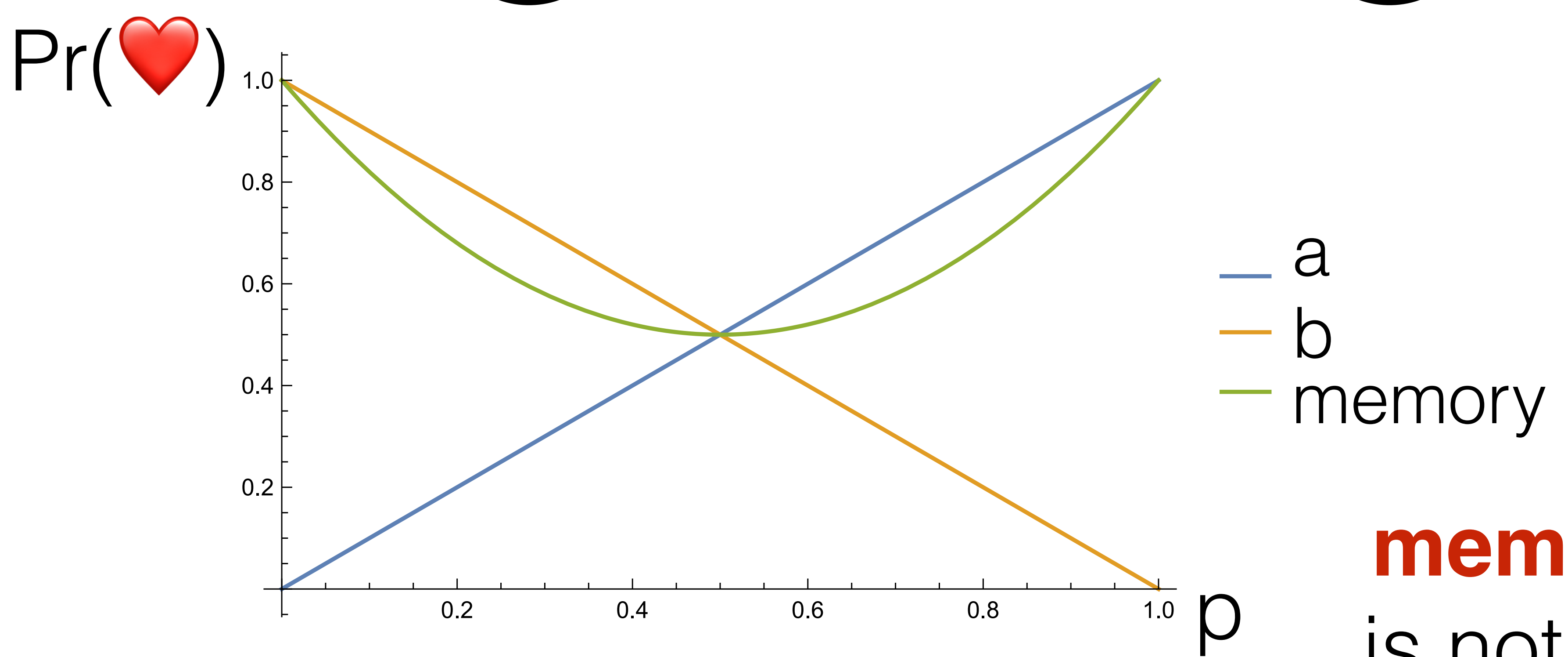




simple policies

Learner Example

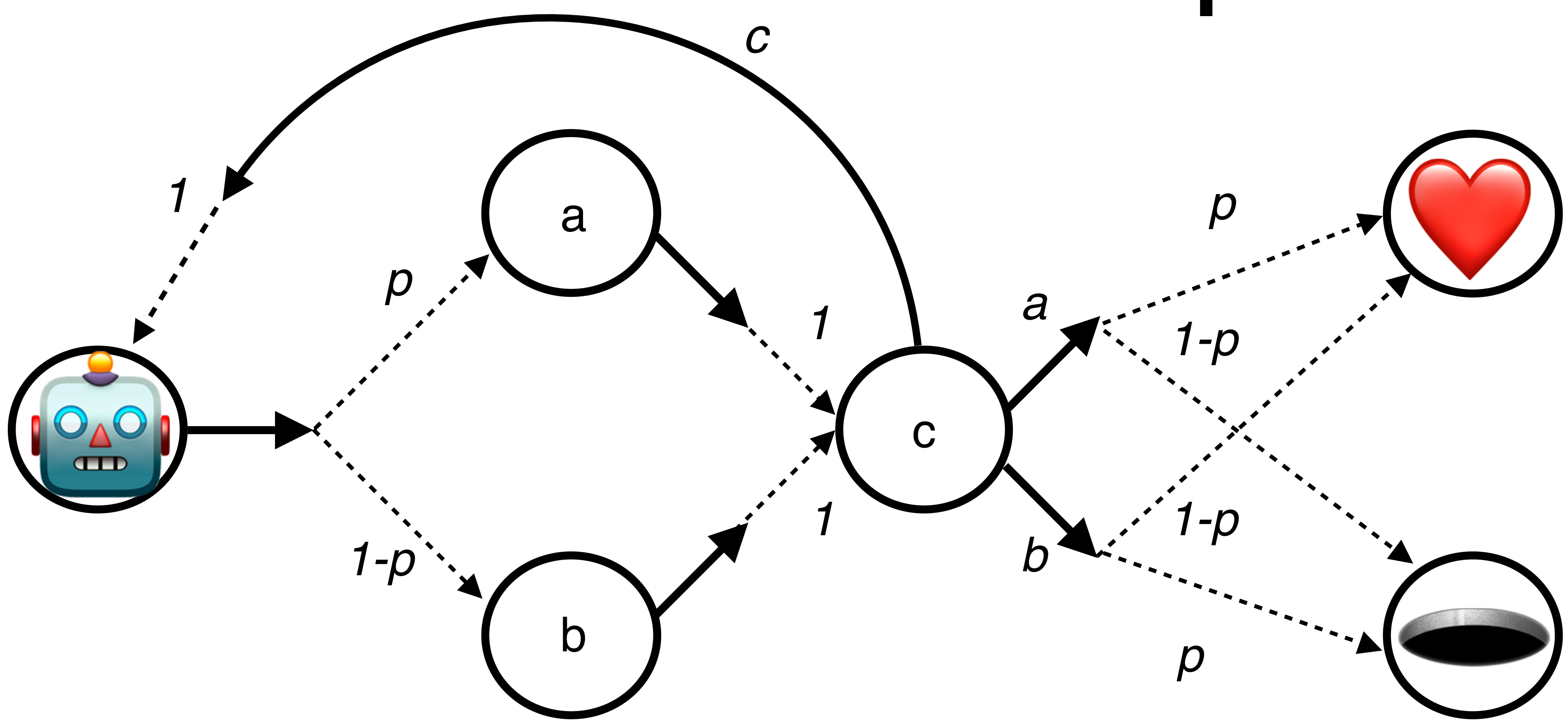


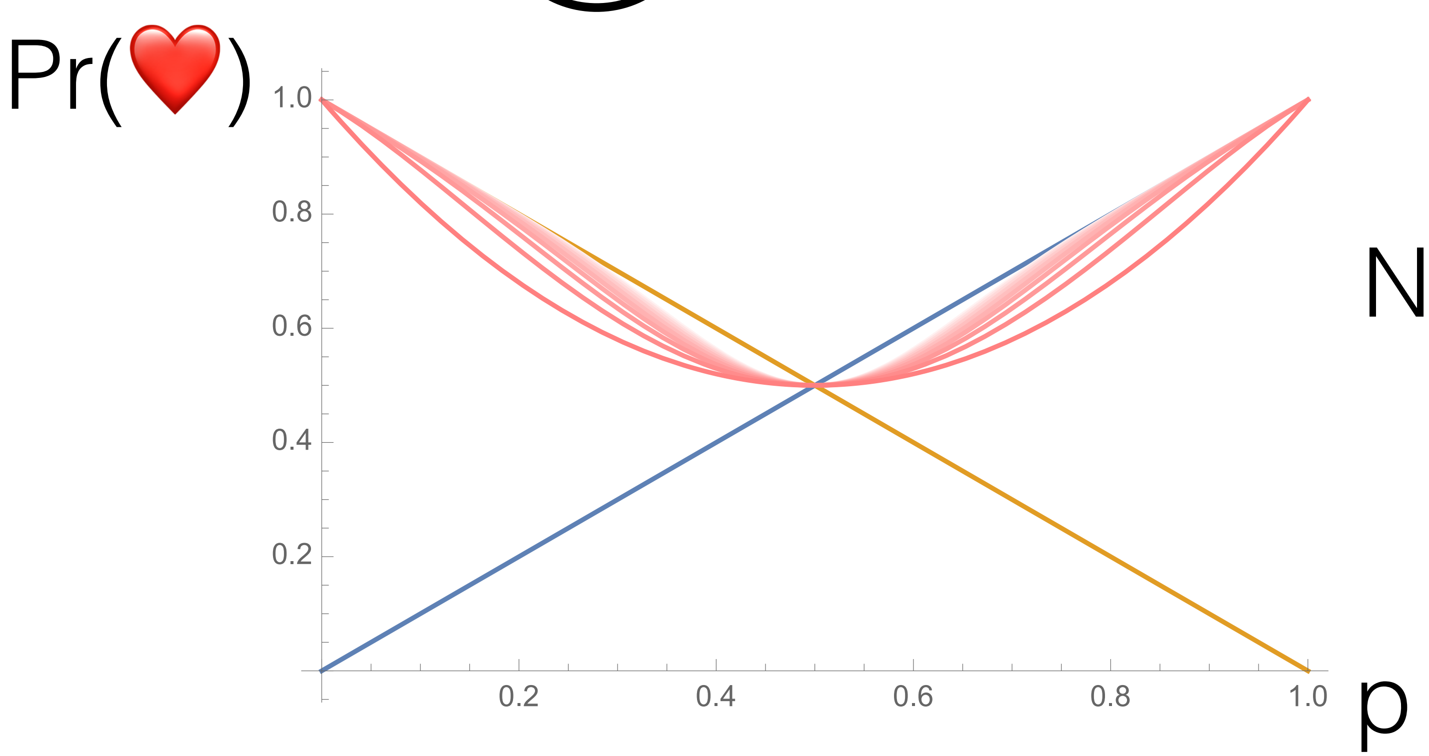


policies

memoryless
is not enough

Learner Example





policies

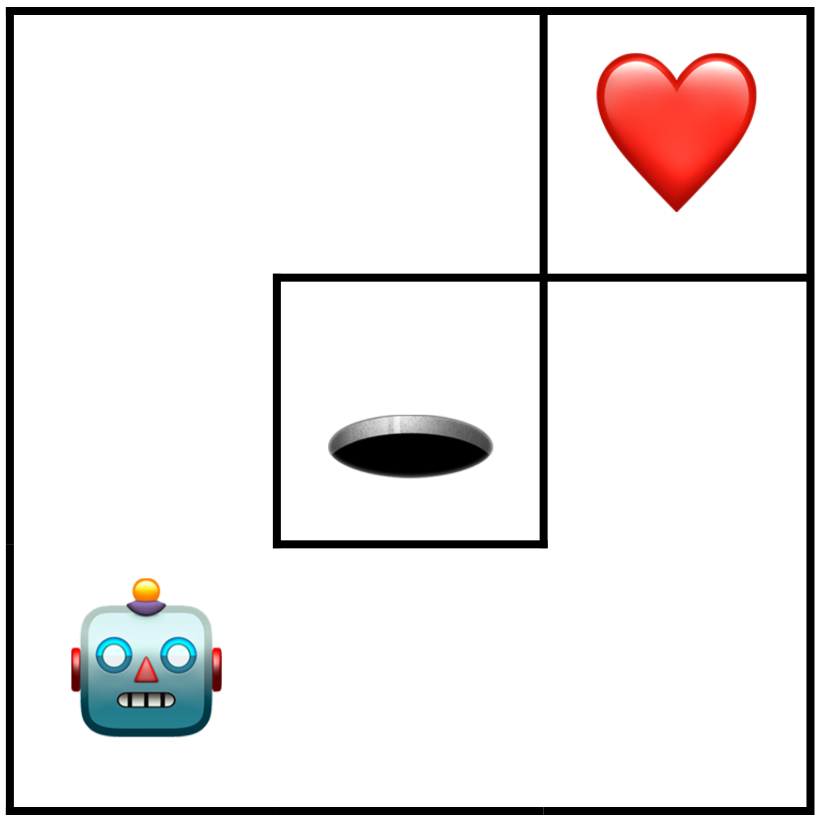
No **optimal** policy but ε-**optimal**

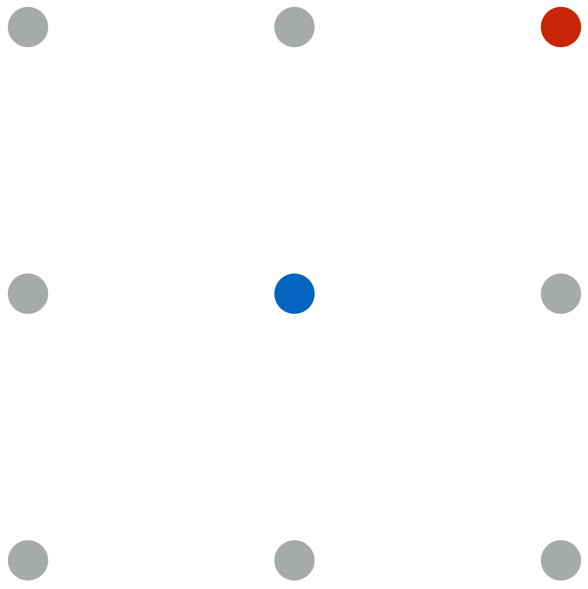
memoryless is not enough

How to compute these policies?

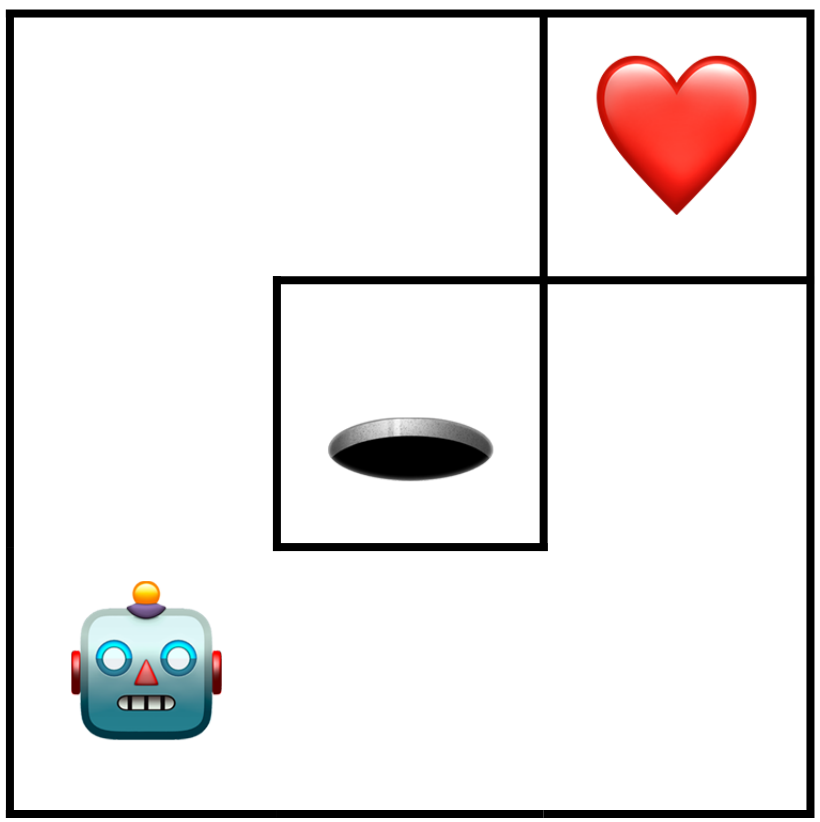
POMDP: Partially Observable MDP

Partially Observable MDP POMDP





Partially Observable MDP POMDP



A **POMDP** is a tuple (S,A,T,Ω,O) where

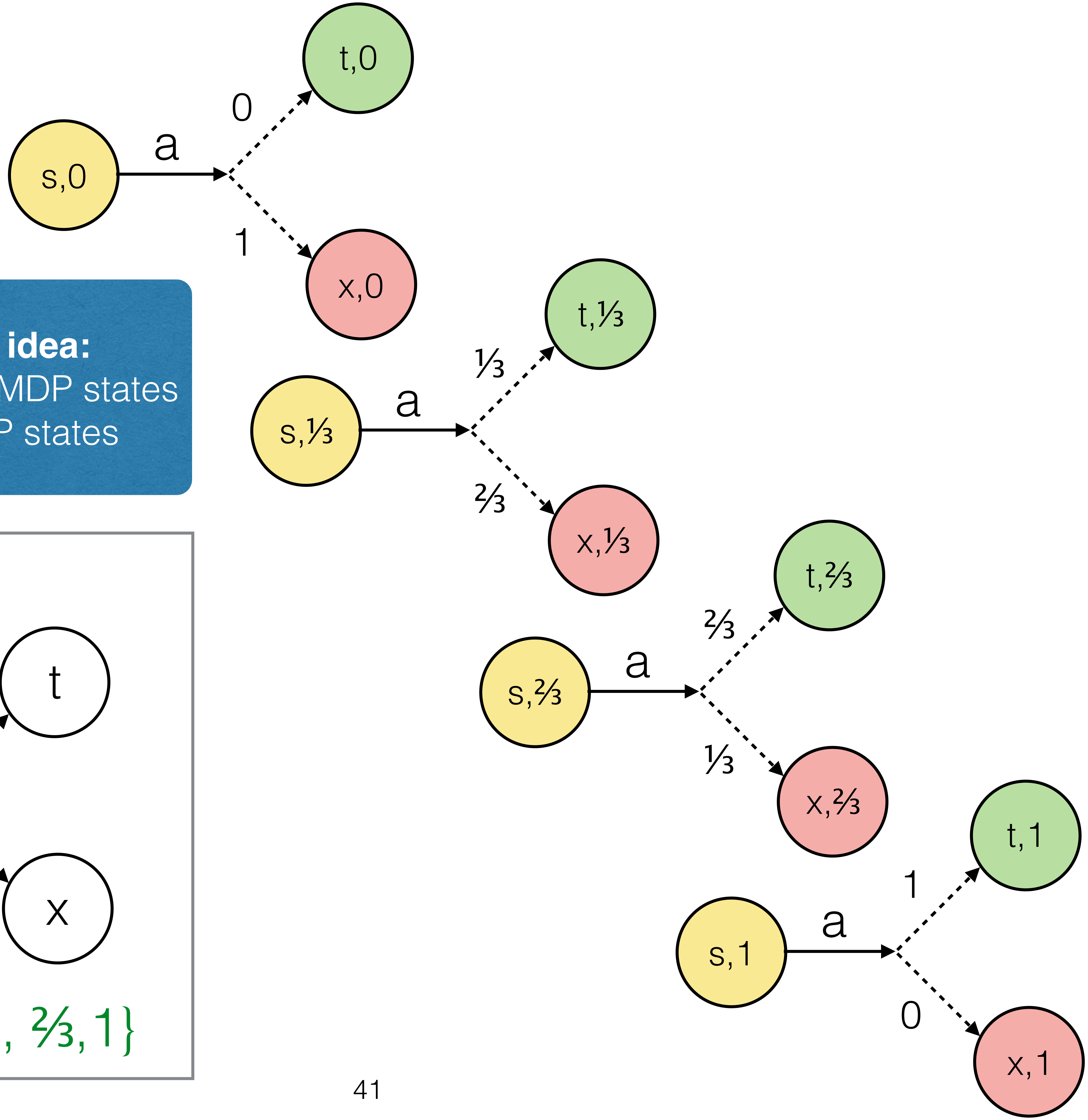
- (S,A,T) is an MDP
- Ω is a set of **observations**
- $O: S \to \Omega$ is the **observation function**

A **POMDP policy** π is a function

$\pi: (\Omega \times A)^* \times \Omega \to \mathscr{D}A$

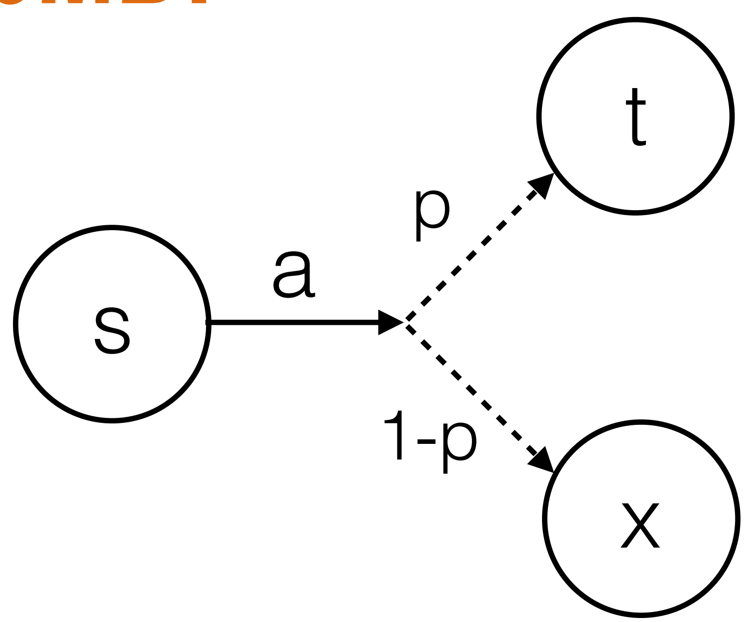
Encoding main idea:
put parameter into POMDP states
observe only pMDP states

POMDP



Encoding main idea:
put parameter into POMDP states
observe only pMDP states

pMDP



$p \in X = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$

The Encoding

Given a **pMDP M** (S,A,X,T) we construct the **POMDP M'** (S',A',T',Ω,O), where

$S' = S \times X$

$A' = A$

$T'((s,x),a)(s',x') = T(s,a)(x)(s') \cdot \delta_x(x')$

$\Omega = S$

$O((s,x)) = s$

Note: There is a 1-1 correspondence between the policies of **M** and **M'**.

Correctness

Hence we can use off-the-shelf POMDP tools to compute expectation optimal pMDP policies.

Theorem:
Given a **pMDP M** and its **POMDP** encoding **M'**:

every **ε-optimal policy** of **M'** is an **ε-expectation optimal policy** for **M**, and vice versa.

Tools

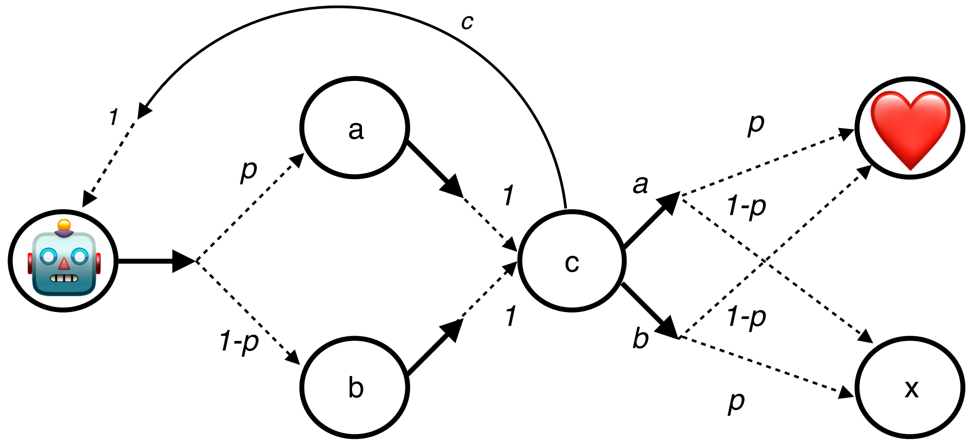
- Work for finite horizon reward objectives
- Online and **Offline** algorithms
- **AI-Toolbox**
  - Incremental Pruning (IP)
  - Point Based Value Iteration (PBVI)

computing ε-optimal policies for infinite horizon POMDPs is undecidable

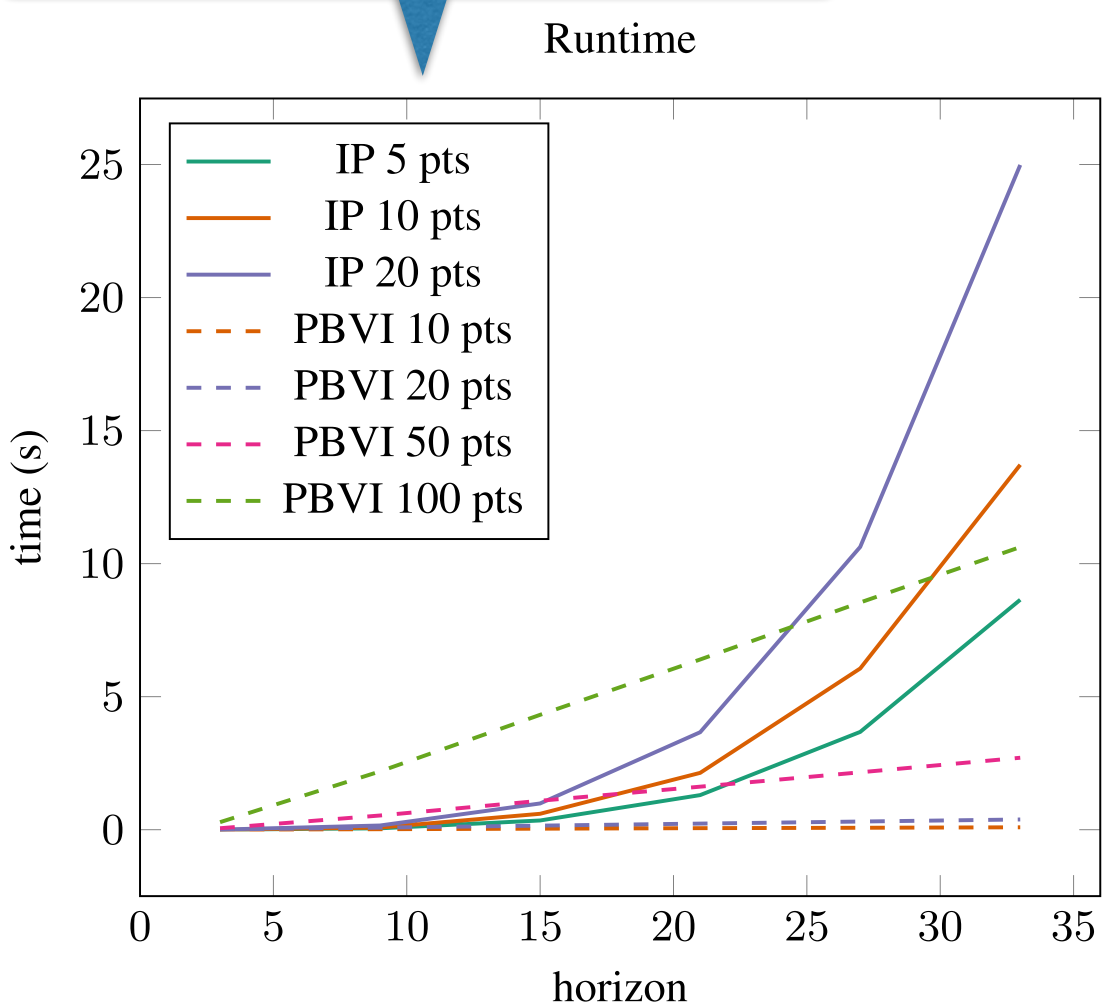
Prism model → STORM → AI-Toolbox via Python interfaces

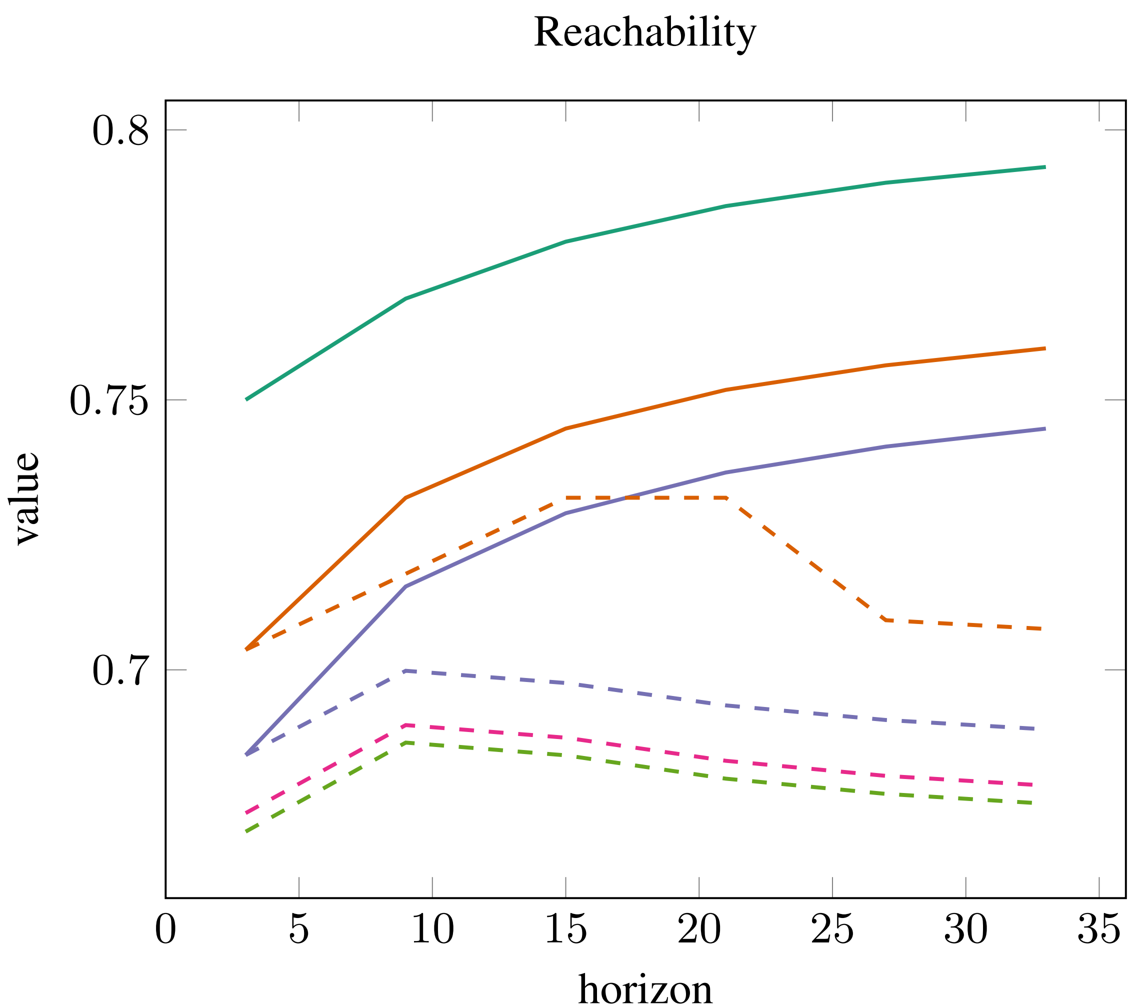
Experimental Results

Learner

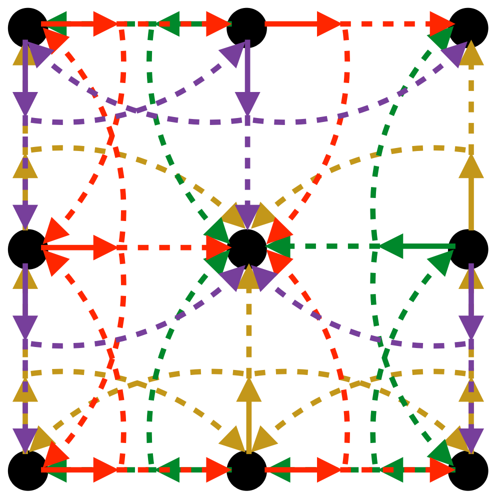


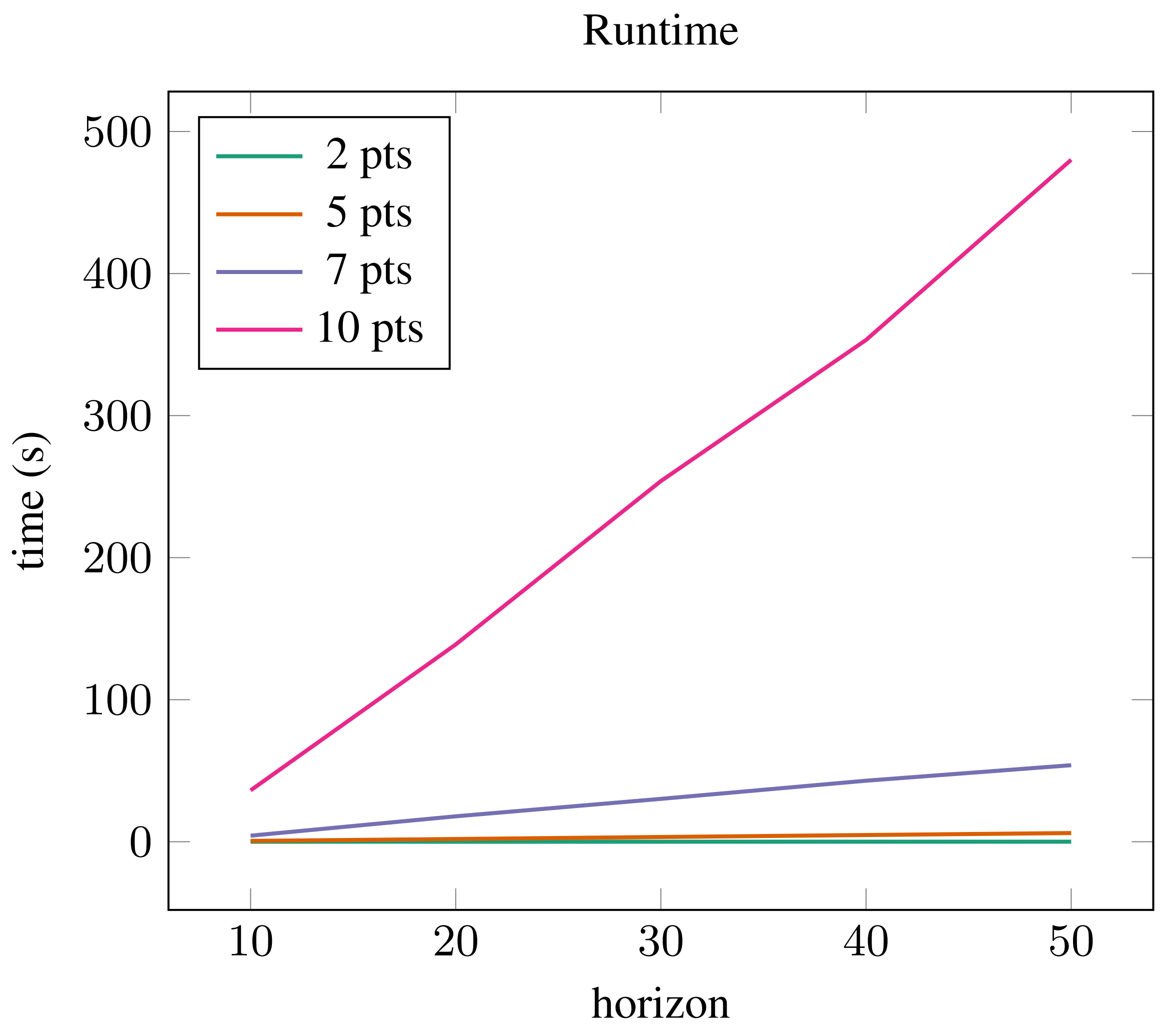
uniform distribution over equidistant points in [0,1]

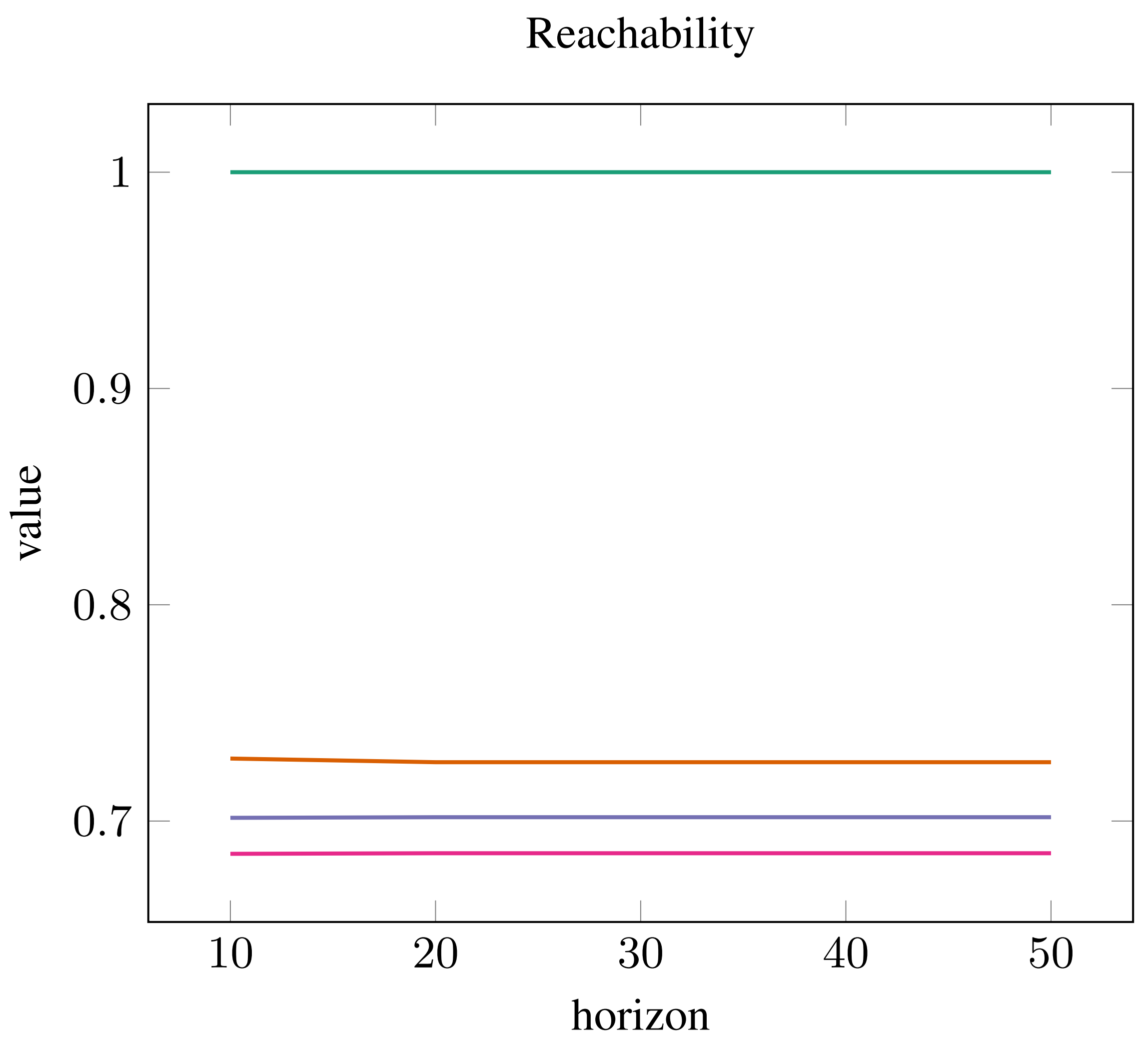




Robot




Runtime
500
400
300
200
100
0
time (s)
10
20
30
40
50
horizon
2 pts
5 pts
7 pts
10 pts


Reachability
1
0.9
0.8
0.7
value
10
20
30
40
50
horizon

Summary

Finding **policies**

of a **parametric MDP**

that are **expectation optimal**

(over the **whole parameter space**)

amounts to solving a **suitable POMDP**.

We have a
proof of concept implementation

github.com/sarming/pMDP-Toolbox

Thank You!