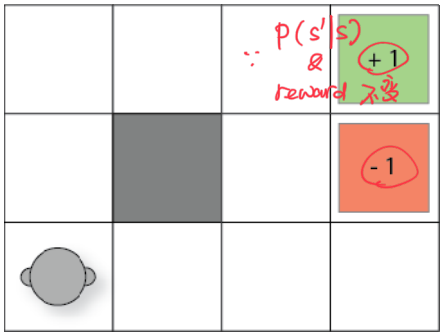
Probabilistic planning - Markov Decision Processes (MDPs)



An agent has a goal to navigate cells

- The grey square is a wall (like the edges of grid)
- The two coloured cells giving rewards: 1 (goal) and -1 (Bad goal)

Actions have **non-deterministic** outcomes (effects)!

- If the agent tries to move north, 80% of the time, this works as planned (provided the wall is not in the way)
- 10% of the time, trying to move north takes the agent east (provided the wall is not in the way)
- 10% of the time, trying to move north takes the agent west (provided the wall is not in the way);
- If wall is in the way of the cell that would have been taken, the agent stays put
- Similar for all other directions

MDPs:	Classical Planning:
Set of states S	Set of states S
Initial state I	Initial state I
Probabilistic state transitions: $\sum_{s'} P(s'\|s) = 1$	Transition function A (deterministic)
Reward function r(s, a, s') in Real	Goals G
Discount factor γ (*gamma*)	Costs

$s \xrightarrow{a} s'$

$s \xrightarrow{a=1} s'$

$s \xrightarrow{a=0} s'$ } MDP = classical planning

γ discount factor is $(0, 1)$ not inclusive

Discounted rewards

Reward at T time

$$G_t = r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 \ldots$$

$$= r_1 + \gamma(r_2 + \gamma(r_3 + \gamma(r_4 \ldots)))$$

$$= r_t + \gamma G_{t+1}$$

Modelling MDPs --- Probabilistic PDDL

```
(define (domain bomb-and-toilet)
    (:requirements :conditional-effects :probabilistic-effects)
    (:predicates (bomb-in-package ?pkg) (toilet-clogged)
                 (bomb-defused))

    (:action dunk-package
       :parameters (?pkg)


  :effect (and (when (bomb-in-package ?pkg)
                 (bomb-defused))
           (probabilistic 0.05 (toilet-clogged))))
```

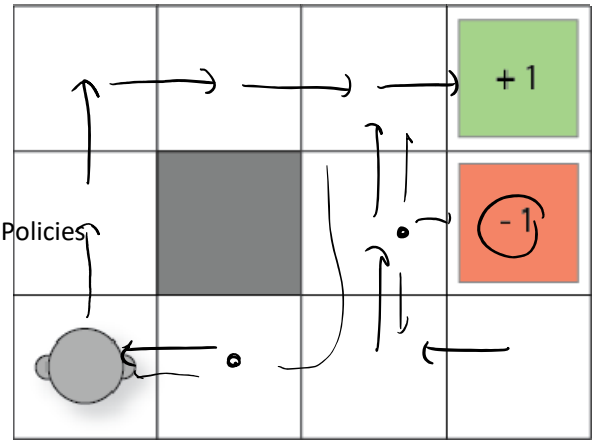
planning : action sequence

Mapping : states to actions

Solution for MDP is a *policy*: (function)

at(0,0) => move_up
at(0,1) => move_up
at(0,2) => move_right
at(1,0) => move_left
at(1,2) => move_right
at(2,0) => move_up
at(2,1) => move_up
at(2,2) => move_right
at(3,0) => move_left

Solutions to MDPs -- Policies



policies { $\pi(s) \to A$ deterministic — same action in the same state → subject focus
$\pi(s,a) \in [0,1]$ stochastic — diff action in the same state

Solving MDPs

Expected return exercise:
You can steal:

A) An iPhone, which you think you have a 20% chance of selling for $500, or an 80% chance of selling for $250.
B) A Samsung, which you think you have a 50% chance of selling for $500, or a 50% chance of selling for $200.

A: 0.2*500 + 0.8*250 = 300
B: 0.5*500 + 0.5*200 = 350

State (s) of Value ⇒ what action is the best

Bellman equation: (value func)

$$V(s) = \max_{a \in A} \sum_{s' \in S} \left[P_a(s'|s) \left[r(s,a,s') + \gamma V(s') \right] \right]$$

(γ: Discount; $r(s,a,s')$: immediate reward; $V(s')$: future reward)

all the possible s' we can transition to

expected reward action a

$Q(s,a)$ ↑ Quality