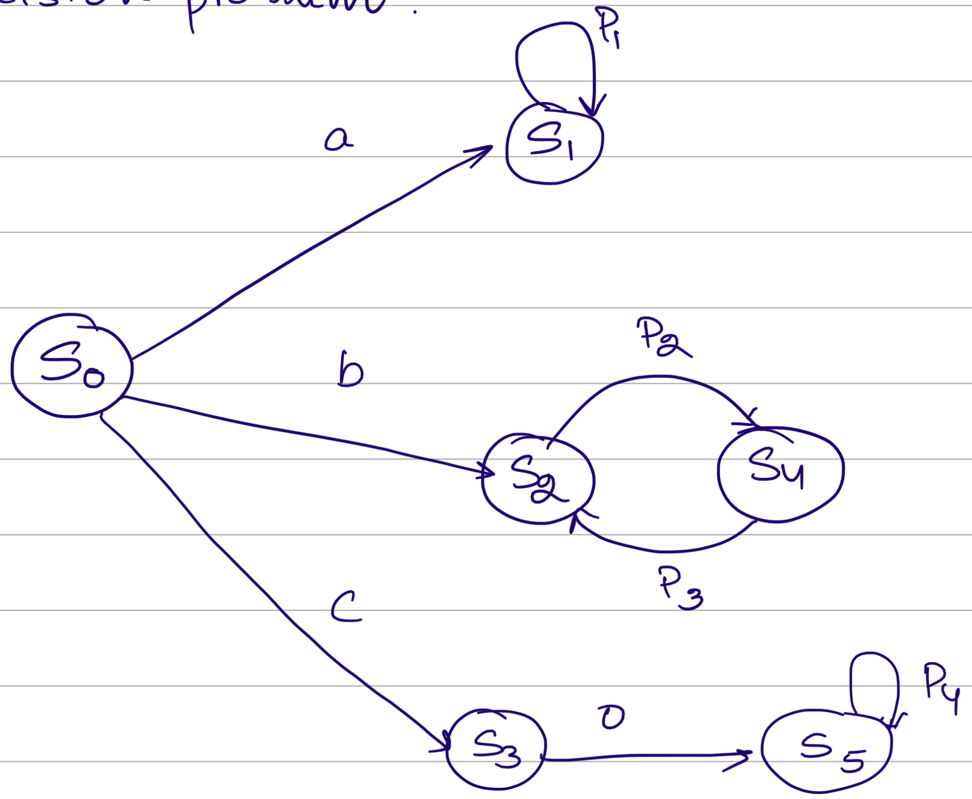
actions: a, b, c MDP

Decision problem:

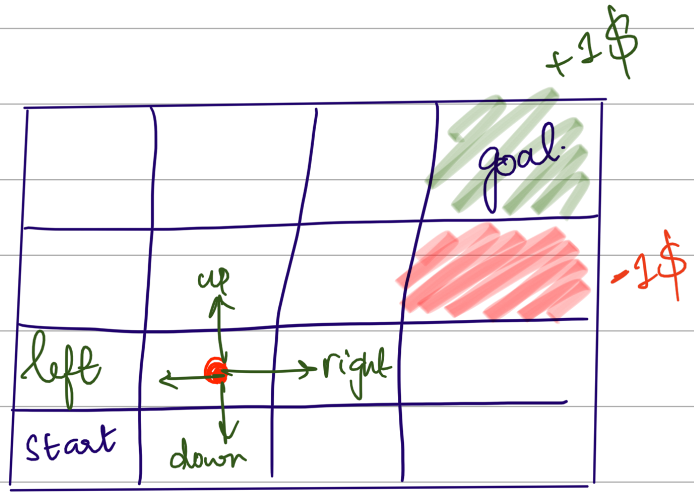


Rewards: P_1, P_2, P_3, P_4

Prob of ending after each step : $1-\gamma$

Discount factor $=\gamma$

Markov Decision Process.



Actions:
[up, down, left, right]

describe the world

states: S

model 1 (Physics of the world, rules)
(transition function)

$$T(S, a, S') = P(S' \mid S, a)$$

state action — state

Actions: A(S), A
things you can do in a particular state
commands that can be executed

scalar value for being in a state

Reward: R(S), R(S, a), R(S, a, S')
usefulness of entering into a state

reward for being in a state and taking an action

R(S, a, S') → reward for being in a state, taking an action and ending up in S'

Markovian Property:-

① $P(S' \mid S, a)$

only depends on "current" state S

$R(S) = R(S, a) = R(S, a, S')$

② $T(S, a, S')$ / Rules do not vary

States
model
Actions
Reward

} → defines a problem

solution to a MDP is a "policy"

Policy is a function that takes a state and returns an action

$\pi(s) \rightarrow a$

For any given state you are in, tells you the action you need to take

optimal policy π^* ↑ Maximizes your long-term expected reward

Find π^* when we have $T[s, a, s'], R[s, a]$

Algorithms {
↳ policy iteration
↳ value iteration