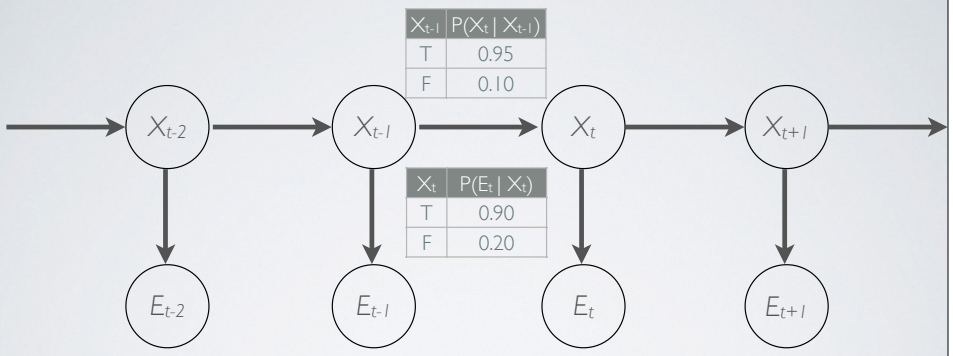
MARKOV DECISION PROCESSES

CSE 511A: Introduction to Artificial Intelligence

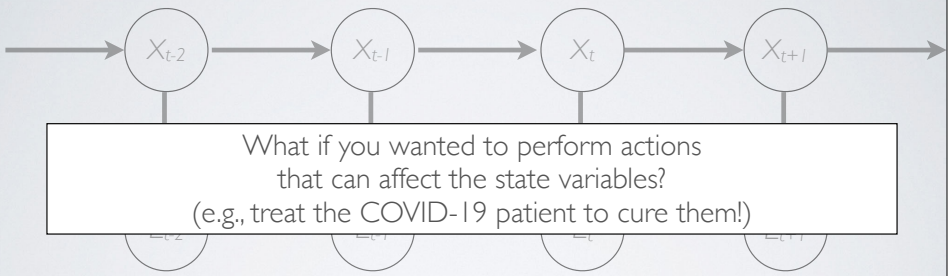
Some content and images are from slides created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.
All CS188 materials are available at http://ai.berkeley.edu.

HIDDEN MARKOV MODELS



- X_t's: State variables (variables that we want to infer) at each time step t.
  e.g., "have the coronavirus"
- E_t's: Evidence variables (variables that we can observe) at each time step t..
  e.g., "have fever"

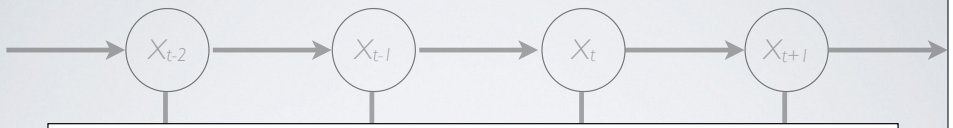
HIDDEN MARKOV MODELS



What if you wanted to perform actions
that can affect the state variables?
(e.g., treat the COVID-19 patient to cure them!)

- X_t's: State variables (variables that we want to infer) at each time step t.
  e.g., "have the coronavirus"
- E_t's: Evidence variables (variables that we can observe) at each time step t..
  e.g., "have fever"

HIDDEN MARKOV MODELS



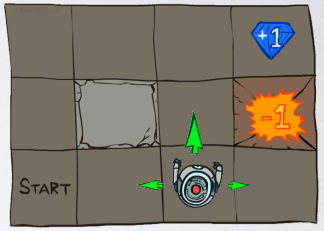
What if you wanted to perform actions
that can affect the state variables?
(e.g., treat the COVID-19 patient to cure them!)

Markov Decision Processes!

- X_t's: State variables (variables that we want to infer) at each time step t.
  e.g., "have the coronavirus"
- E_t's: Evidence variables (variables that we can observe) at each time step t..
  e.g., "have fever"

EXAMPLE: GRID WORLD

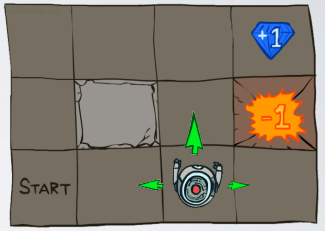
- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path



5

EXAMPLE: GRID WORLD

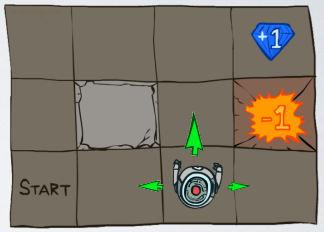
- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path
- Noisy movement: actions do not always go as planned
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put



6

EXAMPLE: GRID WORLD

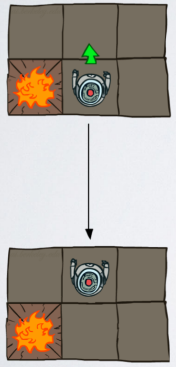
- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path
- Noisy movement: actions do not always go as planned
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- The agent receives rewards each time step
  - Small "living" reward each step (can be negative)
  - Big rewards come at the end (good or bad)
- Goal: maximize sum of rewards



7

EXAMPLE: GRID WORLD

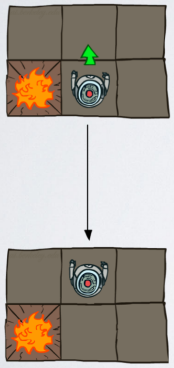
Deterministic Grid World



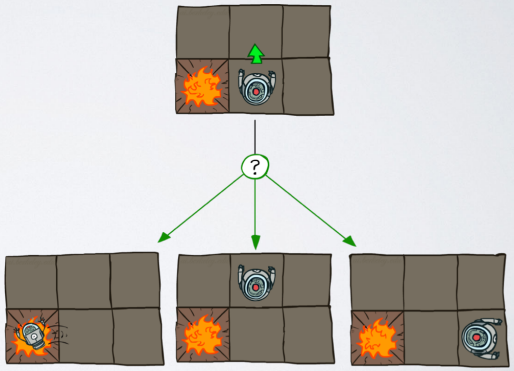
8

EXAMPLE: GRID WORLD

Deterministic Grid World



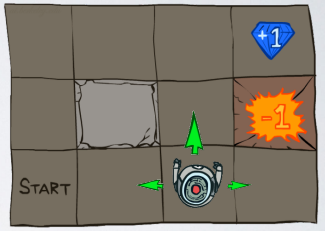
Stochastic Grid World



9

MARKOV DECISION PROCESSES

- An MDP is defined by
  - A set of states $s \in S$
  - A set of actions $a \in A$
  - A transition function $T(s, a, s')$
    - Probability that taking action a from s will lead to s'
    - i.e., $P(s' \mid s, a)$
  - A reward function $R(s, a, s')$
    - Reward of taking action a in s and ending up in s'
    - Sometimes just R(s), R(s,a), or R(s')
  - A discount factor γ
  - A start state
  - Maybe a terminal state

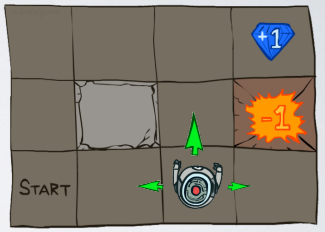


10

11

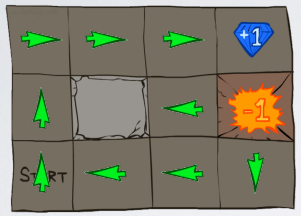
POLICIES

- In deterministic search problems, we wanted an optimal path (i.e., a sequence of actions with minimal cost) from start to goal
- In MDPs, we want an optimal policy $\pi^*: S \rightarrow A$
  - A policy π gives an action for each state
  - An optimal policy is one that maximizes expected utility if followed
  - Once you find an optimal policy, you just need to check what state you are in, and you execute the action for that state prescribed by the policy



12

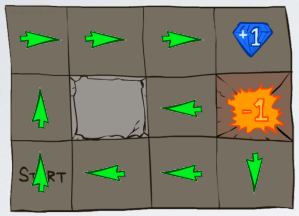
POLICIES



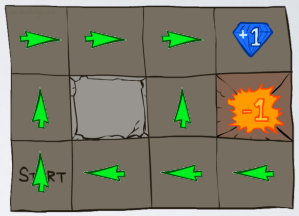
R(s) = -0.01

13

POLICIES



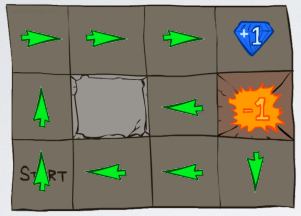
R(s) = -0.01



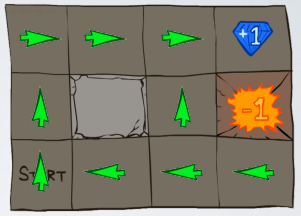
R(s) = -0.03

14

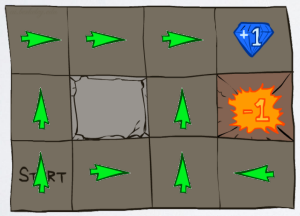
POLICIES



R(s) = -0.01



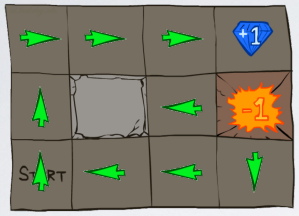
R(s) = -0.03



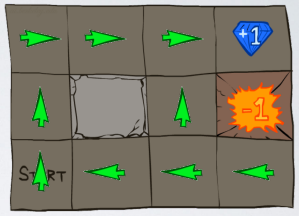
R(s) = -0.40

15

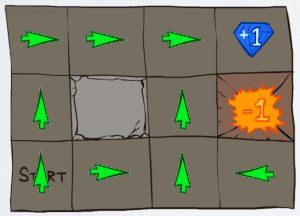
POLICIES



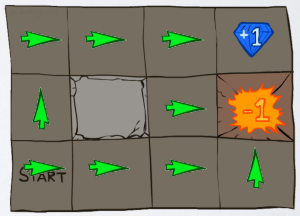
R(s) = -0.01



R(s) = -0.03



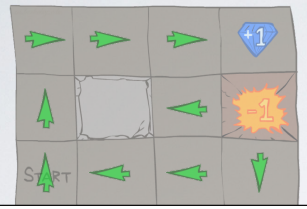
R(s) = -0.40

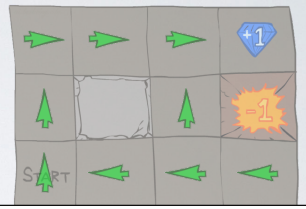


R(s) = -2.00

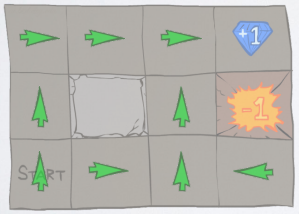
16

POLICIES

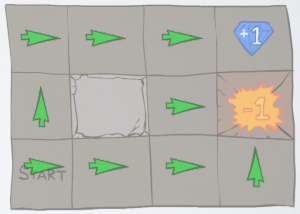




Question: What is the optimal policy if R(s) = 0.01?



R(s) = -0.40



R(s) = -2.00