Bayes Nets and MDPs

Russell and Norvig: Chapter 12, 13
CSE 240: Winter 2023
Lecture 14

Announcements

- Quiz 2 grading in progress
- Assignment 3 is due tonight (Thursday 2/23) at 5pm due to the power outage.
- Working on regrades

Agenda and Topics

- More on Bayesian Networks
- Introduction to Markov Decision Processes (MDP)

The Naïve Bayes Model

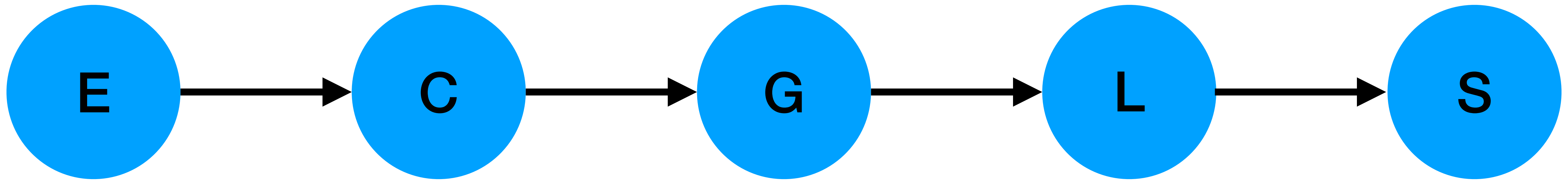
- The *Naïve Bayes Assumption*:
    - Assume that all features are independent **given the class label Y**.
- Equationally speaking:
    - $P(X_1, \ldots, x_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$

Why is This Useful?

- # of parameters for modeling $P(X_1, \ldots, X_n \mid Y)$:
  - $2(2^n - 1)$
- # of parameters for modeling $P(X_1 \mid Y), \ldots, P(X_n \mid Y)$:
  - $2n$

Conditional Independence

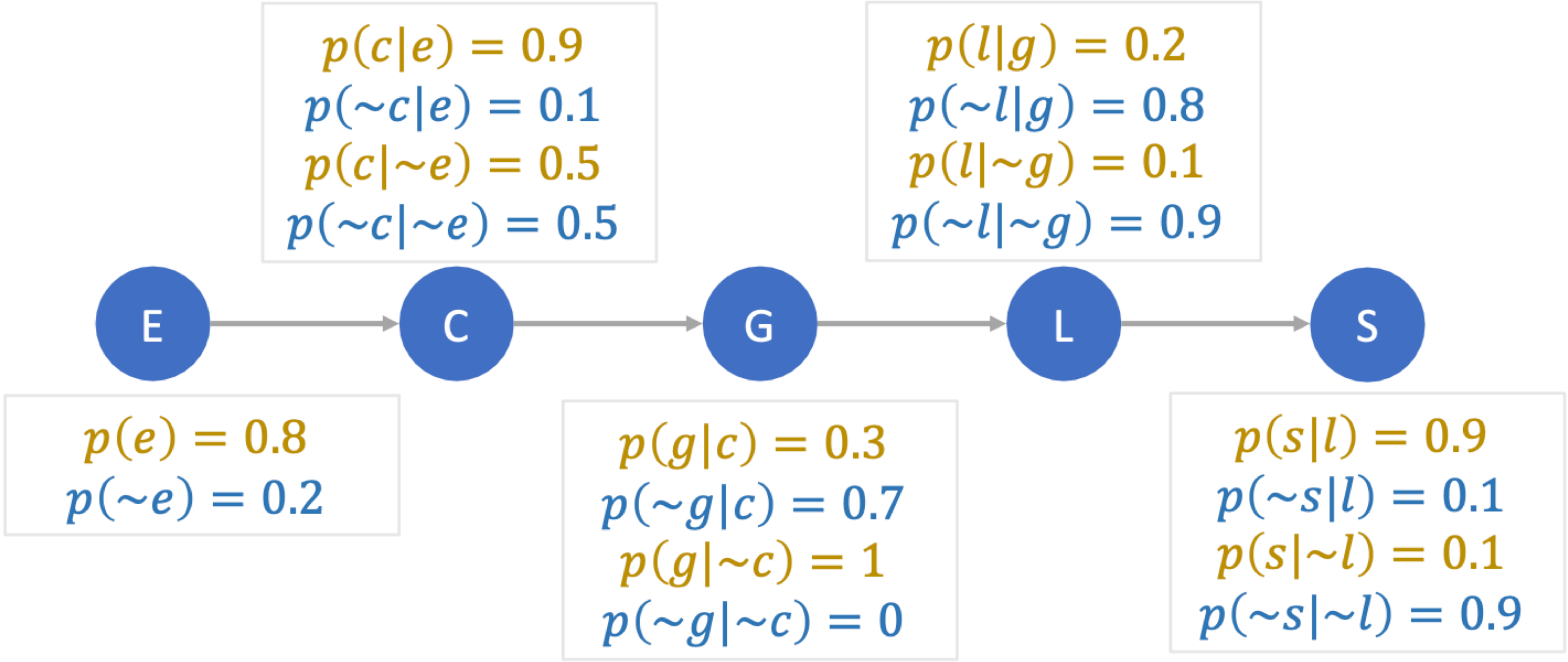
- By the chain rule (for any instantiation of S…E):
    - $P(S, L, G, C, E) = p(S \mid L, G, C, E)p(L \mid G, C, E)p(G \mid C, E)p(C \mid E)p(E)$
- By our independence assumptions:
    - $P(S, L, G, C, E) = p(S \mid L)p(L \mid G)p(G \mid C)p(C \mid E)p(E)$
- We can specify the full joint probability by specifying five local conditional probabilities
    - $p(S \mid L)$
    - $p(L \mid G)$
    - $p(G \mid C)$
    - $p(C \mid E)$ and $p(E)$



Example Quantification

Specifying the joint requires only 9 parameters

What is $p(g)$



Bayesian Networks

- The structure we just mentioned is a Bayesian network.
- Graphical representation of the direct dependencies over a set of variables + a set of conditional probability tables (CPTs) quantifying the strength of those influences.
- Bayesian Networks generalize the above ideas in very interesting ways, leading to effective means of representation and inference under uncertainty.

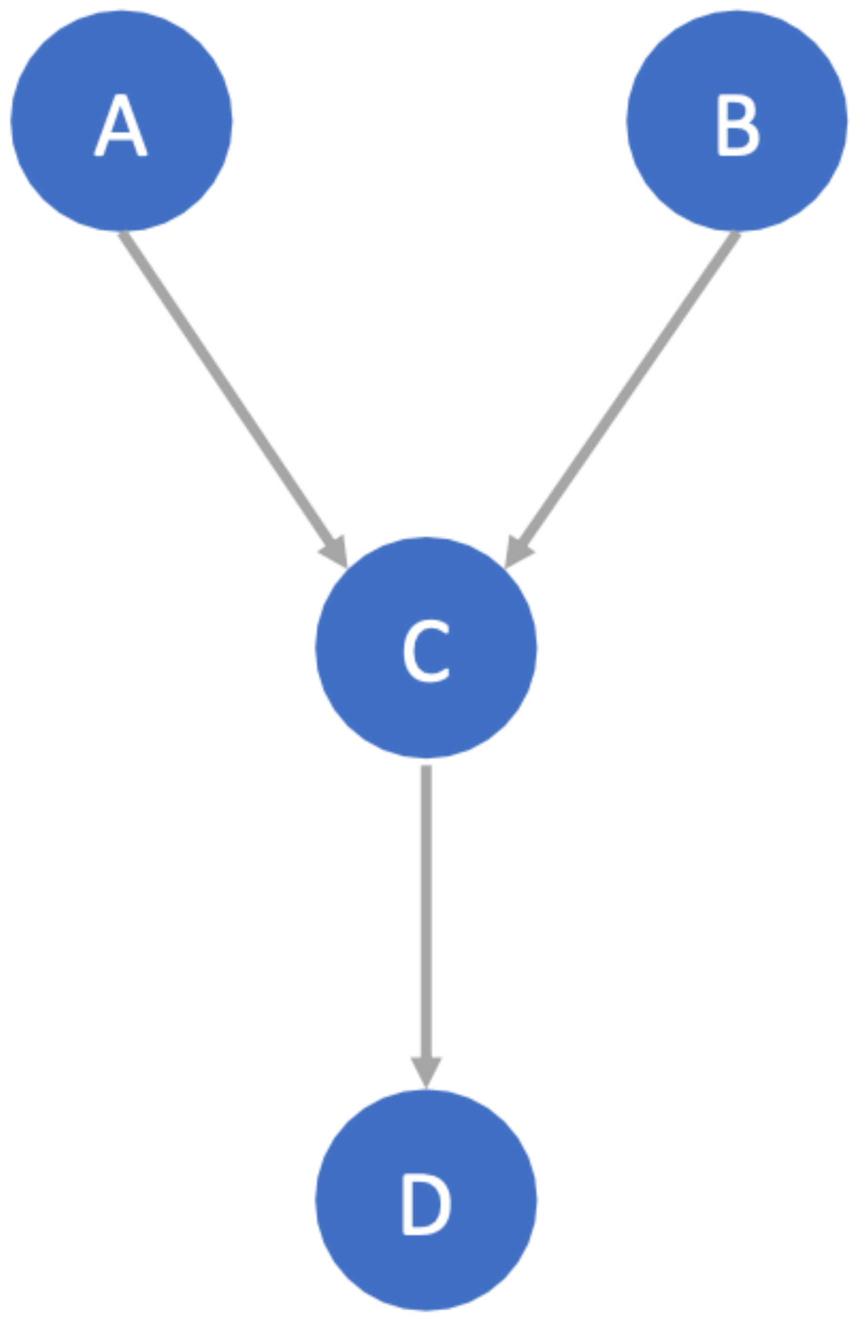
Bayesian Networks

- A simple, graphical notation for conditional independence assertions resulting in a compact representation for the full joint distribution
- Syntax:
  - a set of nodes, one per random variable
  - a directed, acyclic graph (link = ‘direct influences’)
  - a conditional distribution (CPT) for each node given its parents:

$$P(X_i|Parents(X_i))$$

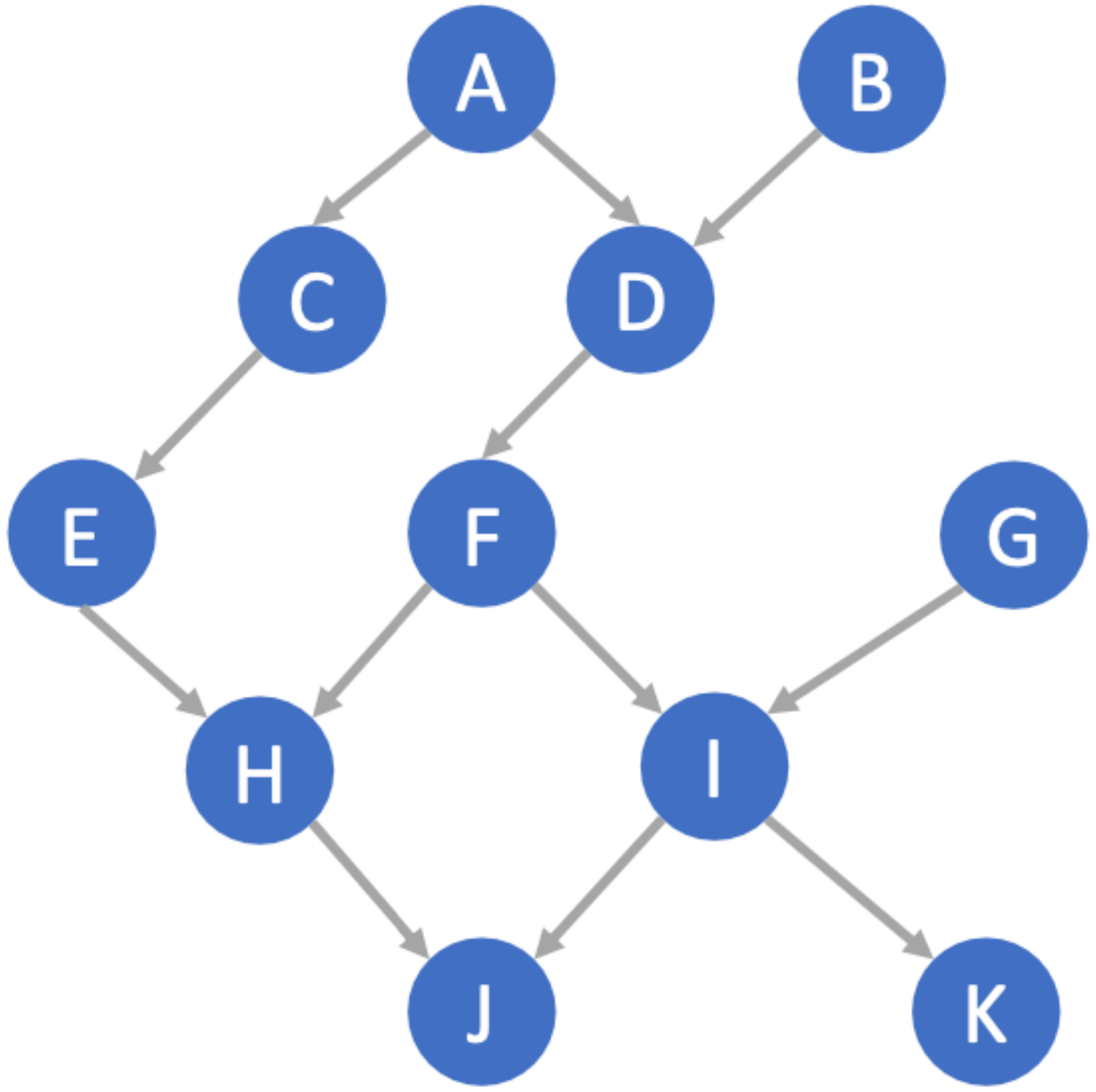
Key Notions

- Some definitions:
  - parents of a node → $par(C) = \{A, B\}$
  - children of node → $children(A) = \{C\}$
  - descendants of a node → $descendants(B) = \{C, D\}$
  - ancestors of a node → $ancestors(D) = \{A, B, C\}$
  - family: set of nodes consisting of x_i and its parents → $family(C) \ = \ \{C, A, B\}$
- CPTs are defined over families in the BN



An Example of a Bayes Net

- How many parameters do we need for the following BN?



Semantics of a Bayesian Network

- The structure of the BN means: every x_i is conditionally independent of all of its non-descendants given its parents:
    - $p(x_i \mid X \cup par(x_i)) = p(par(x_i))$
    - For any subset $S \subseteq non - descendants(x_i)$

How to build a Bayesian Network

1. Define a total order over the random variables: $(x_1, \dots, x_n)$
2. Apply the chain rule:
$$p(x_1, \dots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, \dots, x_{i-1})$$
3. For each x_i, select the smallest set of predecessors $par(x_i)$ such that:
$$p(x_i | x_1, \dots, x_{i-1}) = p(x_i | \, par(x_i))$$
4. Then we can rewrite
$$p(x_1, \dots, x_n) = \prod_{i=1}^{n} p(x_i | par(x_i))$$

How to build a Bayesian Network (2)

5. This is a compact representation of the initial JPD. Factorization of the JPD based on existing conditional independencies among the variables
6. Construct the Bayesian Net (BN)
   - ✓ Nodes are the random variables
   - ✓ Draw a directed edge from each variable in $par(x_i)$ to x_i
   - ✓ Define a conditional probability table (CPT) for each variable x_i:

$$p(x_i \mid par(x_i))$$

Example for BN Construction

- You want to diagnose whether there is a fire in a building
- You can receive reports (possibly noisy) about whether everyone is leaving the building
- If everyone is leaving, this may have been caused by a firealarm
- If there is a fire alarm, it may have been caused by a fire or by tampering
- If there is a fire, there may be smoke

Fire Diagnosis: Step 1

- Start by choosing the random variables for this domain:
  - Tampering (T) is true when the alarm has been tampered with
  - Fire (F) is true when there is a fire
  - Alarm (A) is true when there is an alarm
  - Smoke (S) is true when there is smoke
  - Leaving (L) is true if there are lots of people leaving the building
  - Report (R) is true if the sensor reports that lots of people are leaving the building

Fire Diagnosis: Step 2

- Define total ordering of variables.
- Let's choose an order that follows the causal sequence of events:
  - Fire(F), Tampering (T), Alarm (A), Smoke (S), Leaving (L), Report (R)

Fire Diagnosis: Step 3

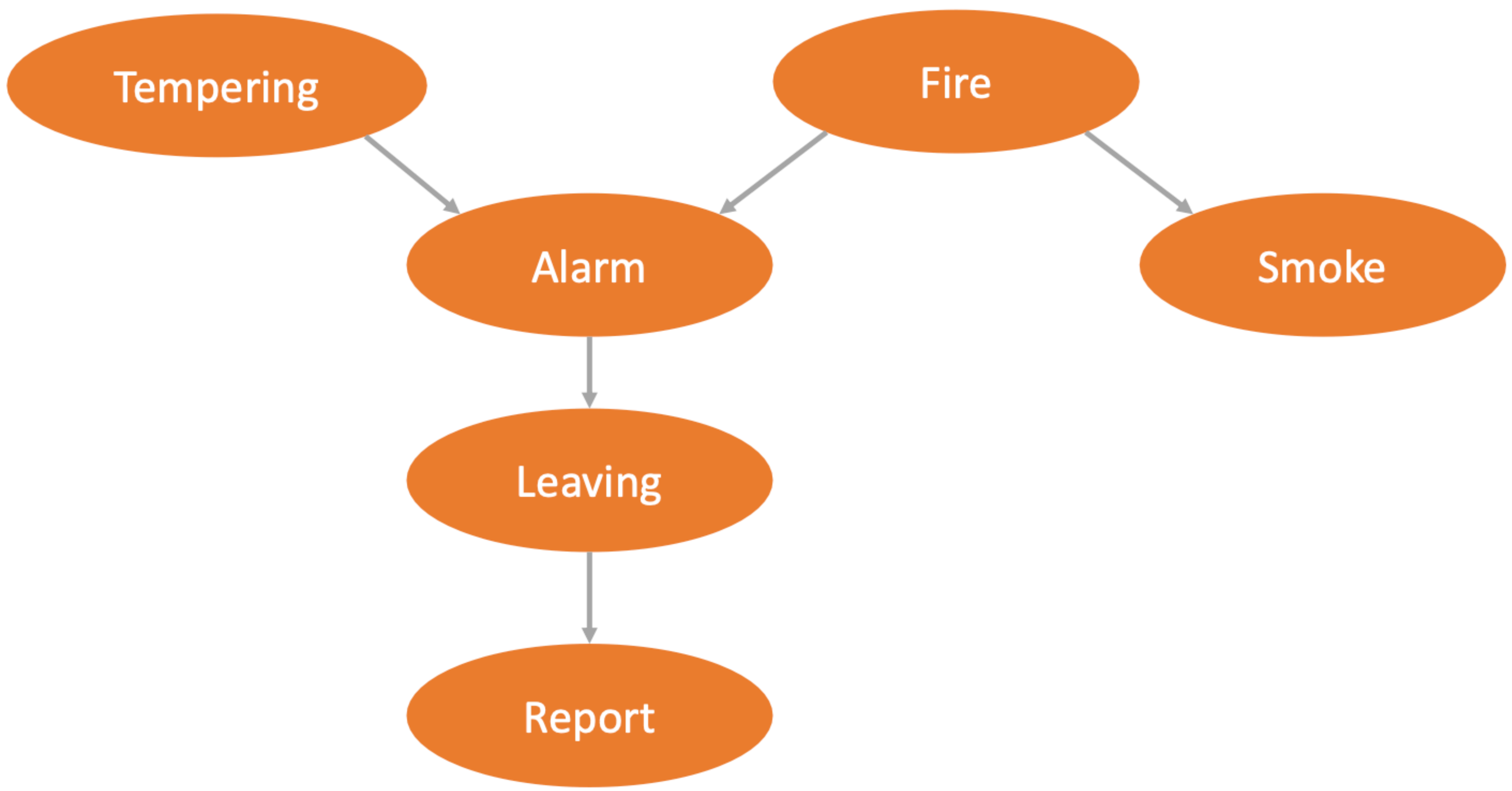
- Ordering:
  - Fire(F), Tampering (T), Alarm (A), Smoke (S), Leaving (L), Report (R)
- Apply the chain rule:
- $p(F, T, A, S, L, R) = p(F)p(T \mid F)p(A \mid F, T)p(S \mid F, T, A)p(L \mid F, T, A, S)p(R \mid F, T, A, S, L)$

Fire Diagnosis: Step 4 & 5

- $p(F, T, A, S, L, R) = p(F)p(T \mid F)p(A \mid F, T)p(S \mid F, T, A)p(L \mid F, T, A, S)p(R \mid F, T, A, S, L)$
- For each variable, x_i choose parents $par(x_i)$ and re-write the joint probability distribution:
  - $p(F, T, A, S, L, R) = p(F)p(T)p(A \mid F, T)p(S \mid F)p(L \mid A)p(R \mid L)$
- Now we need to build the BN based on the above JPD.

Fire Diagnosis: Drawing BN

- $p(F, T, A, S, L, R) = p(F)p(T)p(A \mid F, T)p(S \mid F)p(L \mid A)p(R \mid L)$



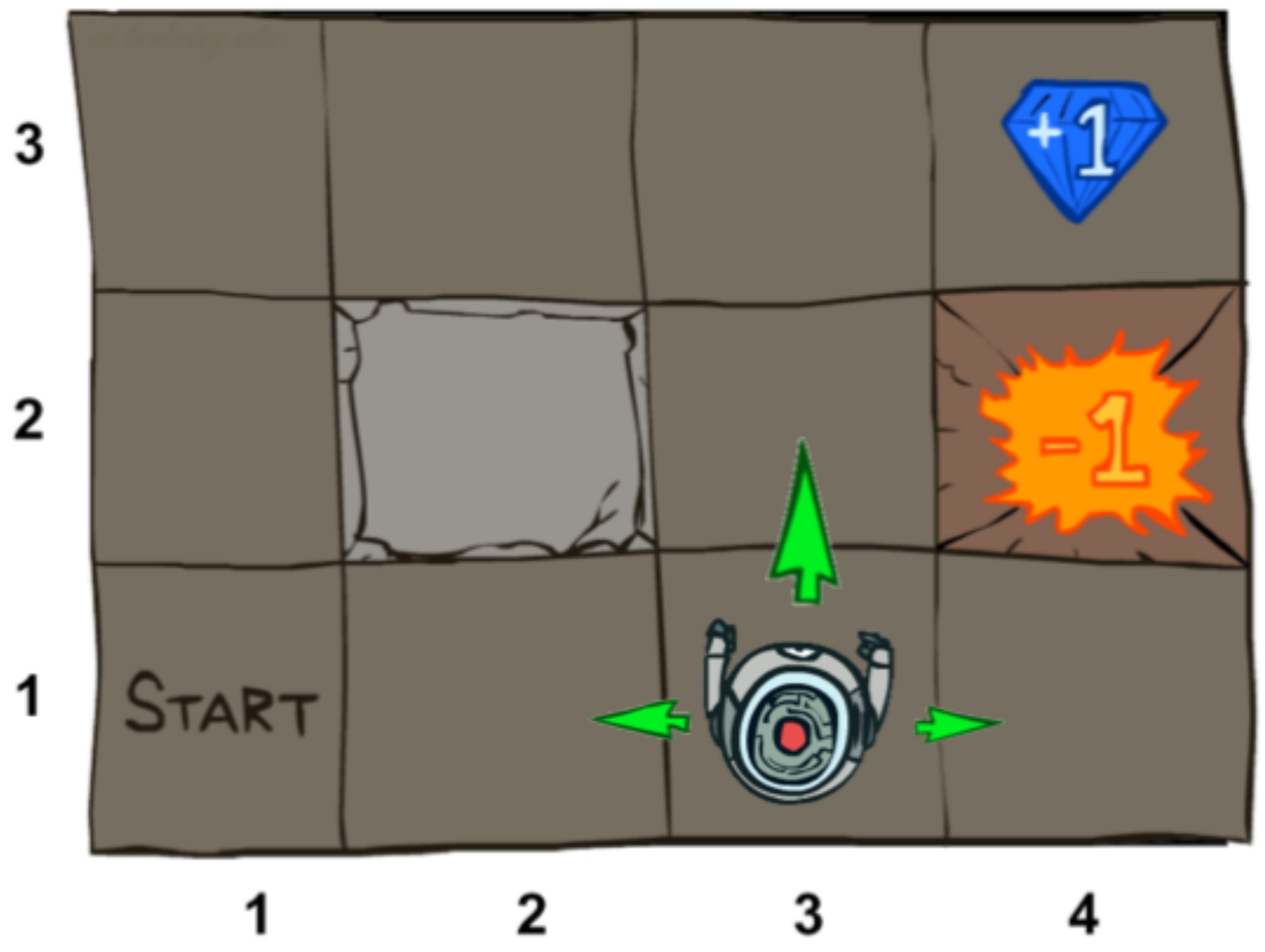
CE 14: Survey

- https://forms.gle/RVTQcREQoir9L3zp7

MDP: Non-Deterministic Search

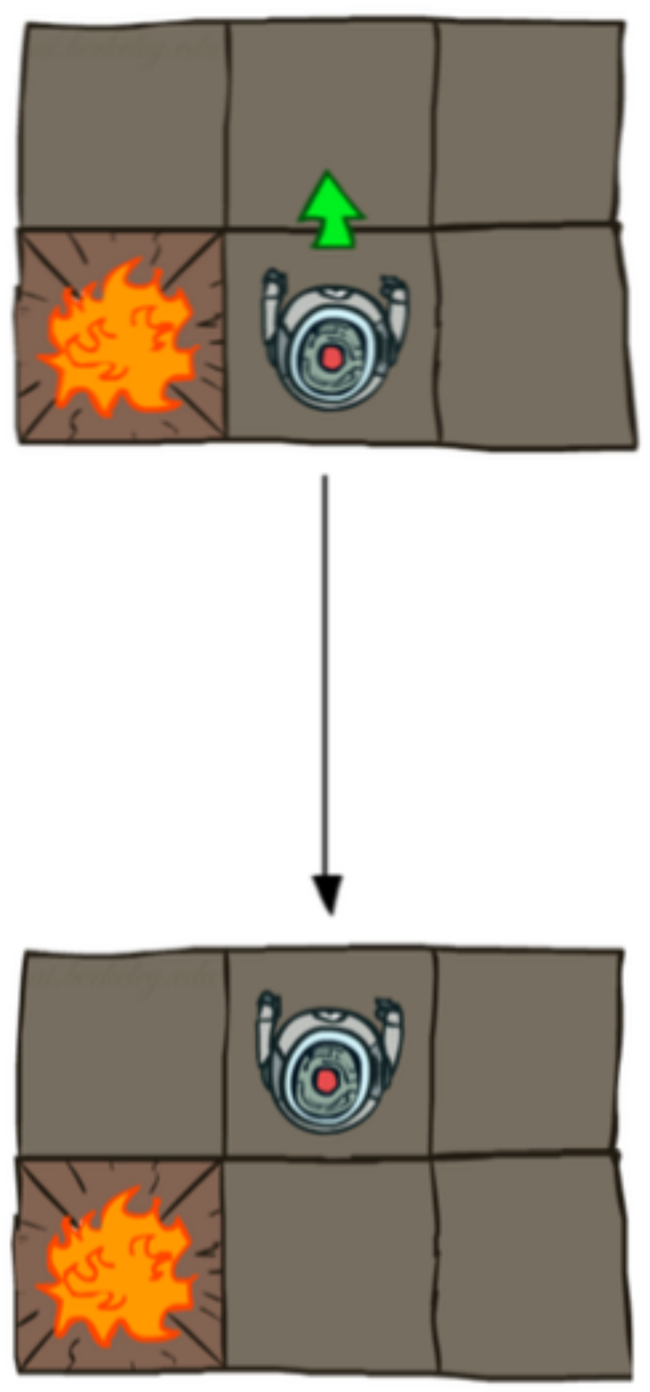
Example: Grid World

- 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
    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

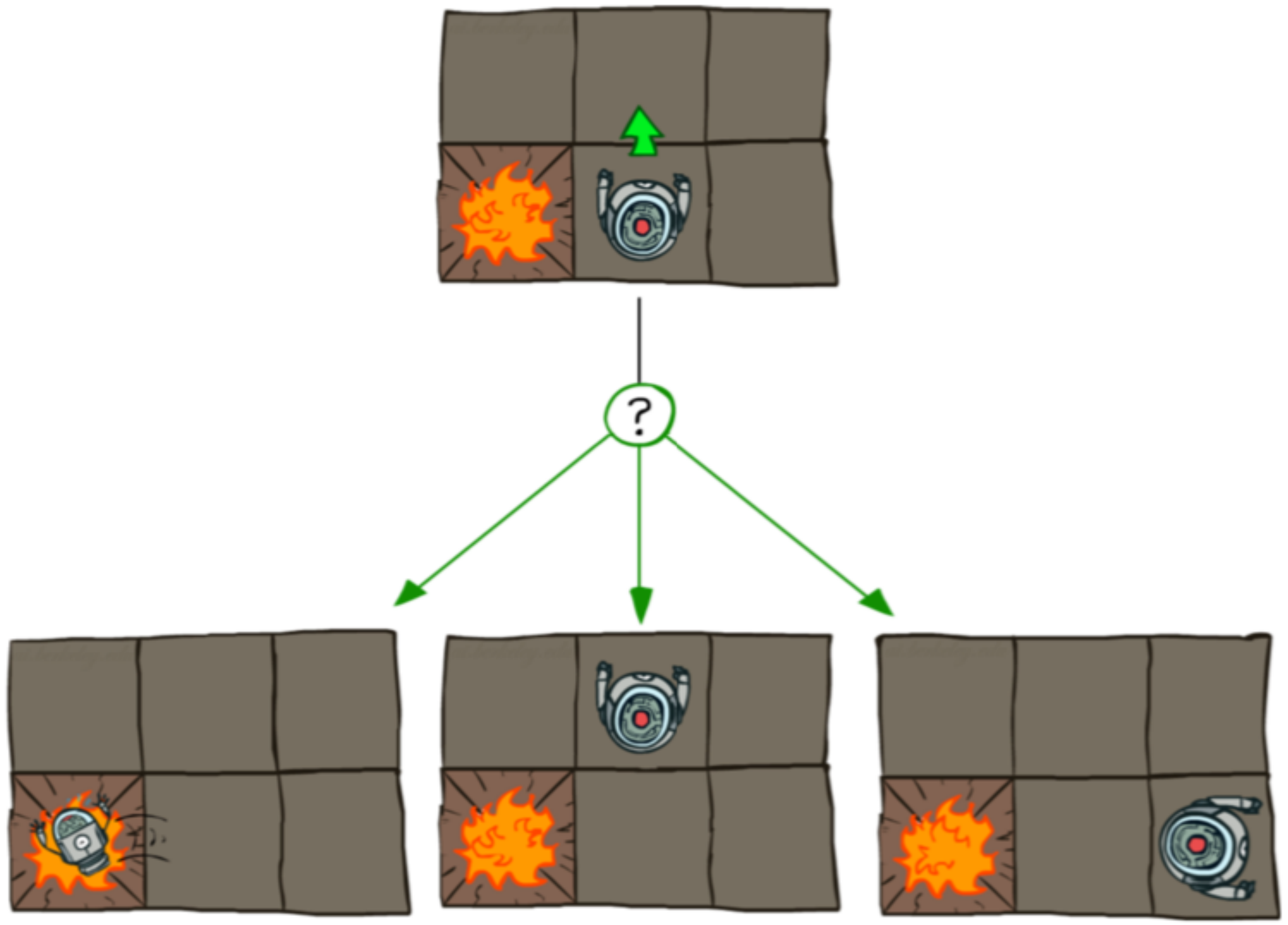


Grid World Actions

- Deterministic Grid World

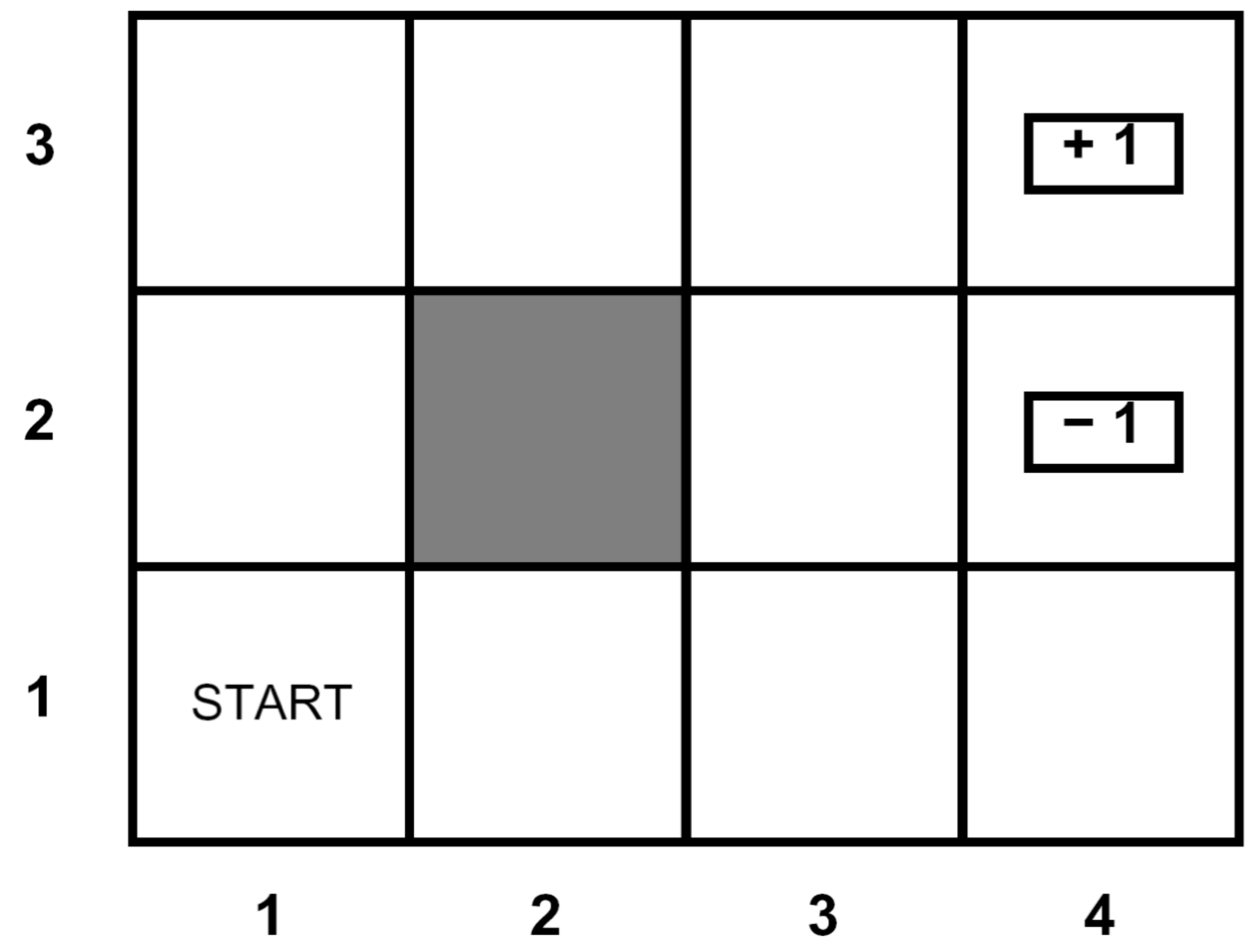


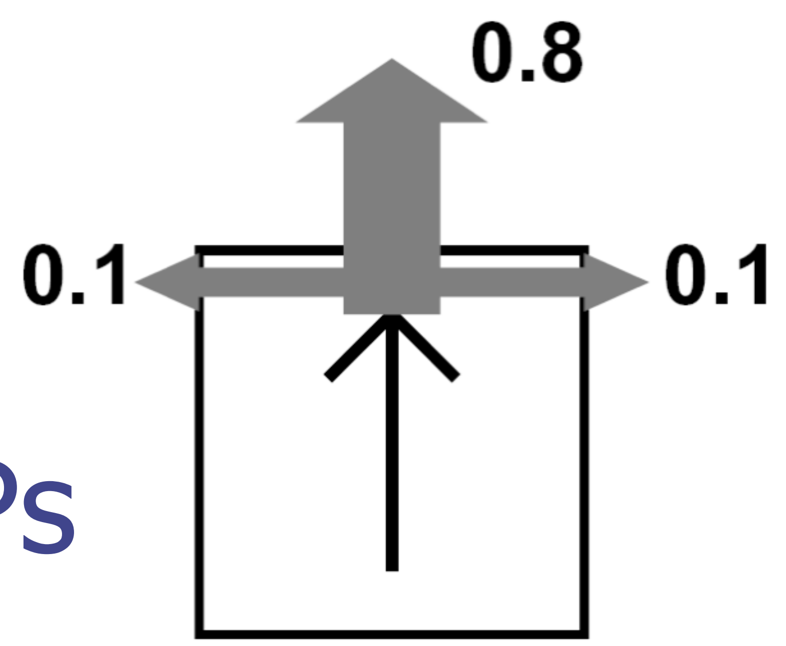
- Stochastic Grid World



Markov Decision Processes

- MDPs are a family of non-deterministic search problems
- An MDP is defined by:
  - A set of states s ∈ S
  - A set of actions a ∈ A
  - A transition function T(s,a,s')
    - Prob that a from s leads to s'
    - i.e., P(s' | s,a)
    - Also called the model
  - A reward function R(s, a, s')
    - Sometimes just R(s) or R(s')
  - A start state (or distribution)
  - Maybe a terminal state





Reinforcement learning: MDPs where we don't know the transition or reward functions

What is Markov about MDPs?

- Andrey Markov (1856-1922)
- “Markov” generally means that given the present state, the future and the past are independent
- For Markov decision processes, “Markov” means:

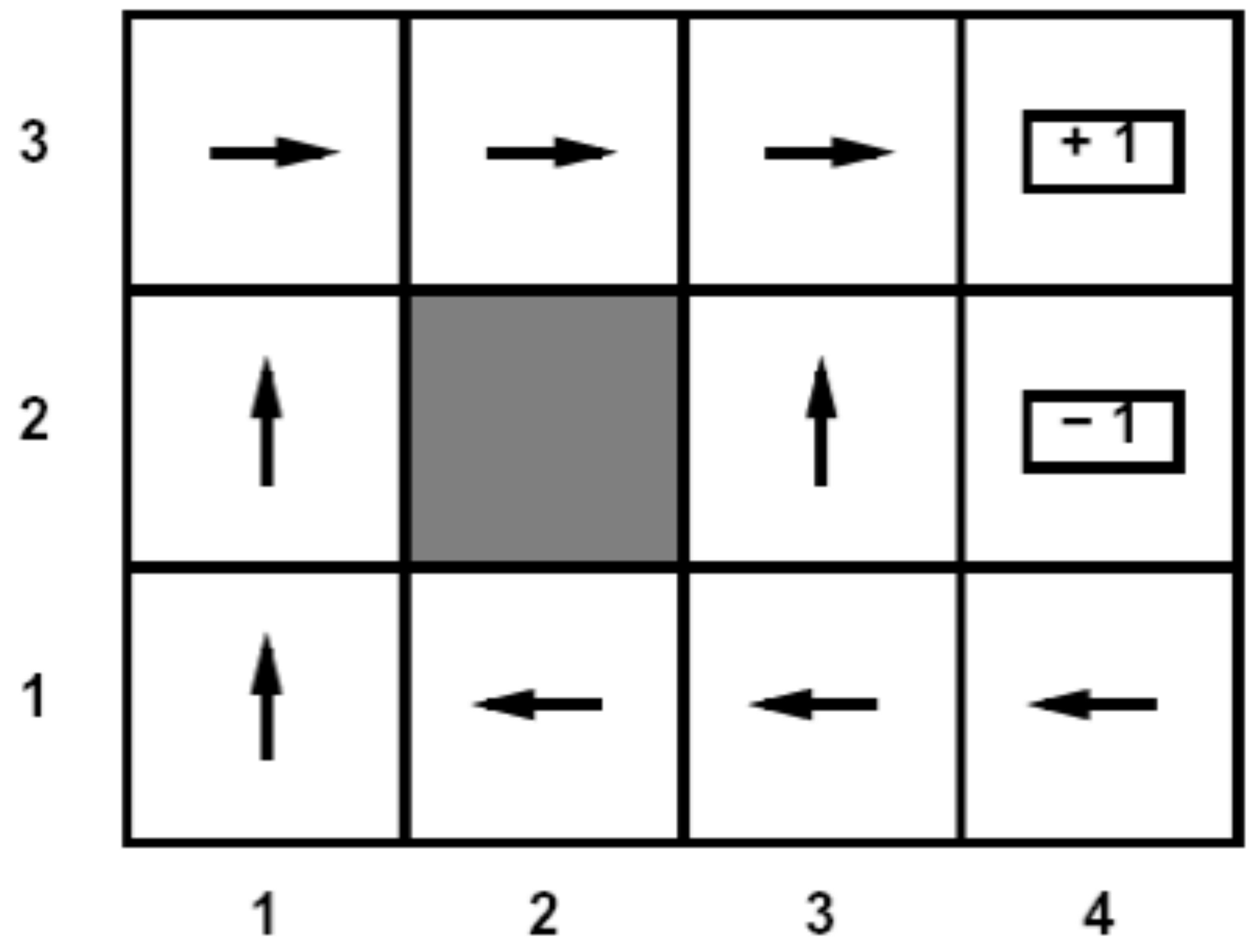
$$P(S_{t+1} = s' \mid S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, \ldots, S_0 = s_0, A_0 = a_0)$$
$$=$$
$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

Aside often use following shorthand:

$$P(s' \mid s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) = P(s' \mid s_t, a_t)$$

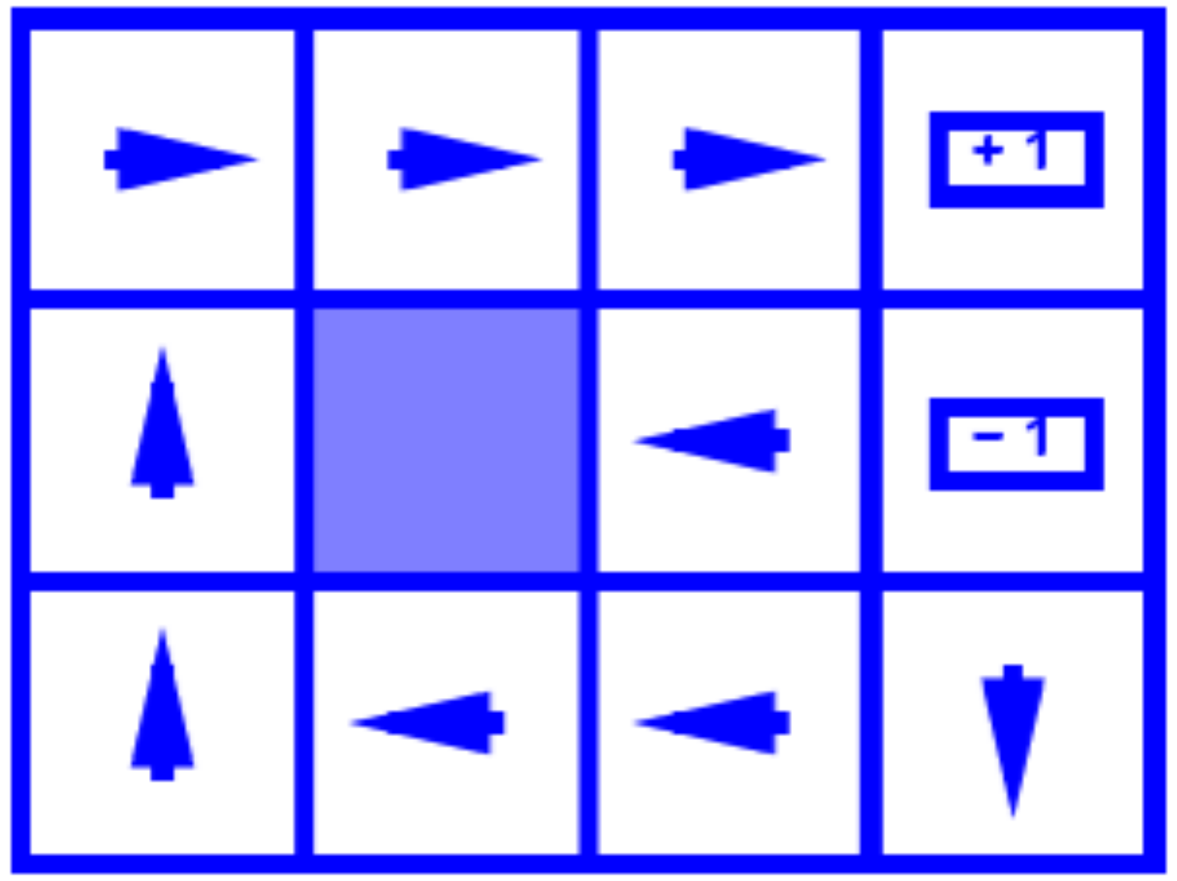
Solving MDPs

- In deterministic single-agent search problems, want an optimal plan, or sequence of actions, from start to a goal
- In an MDP, we want an optimal policy $\pi^*: S \rightarrow A$
  - A policy π gives an action for each state
  - An optimal policy maximizes expected utility if
  - An explicit policy defines a reflex agent

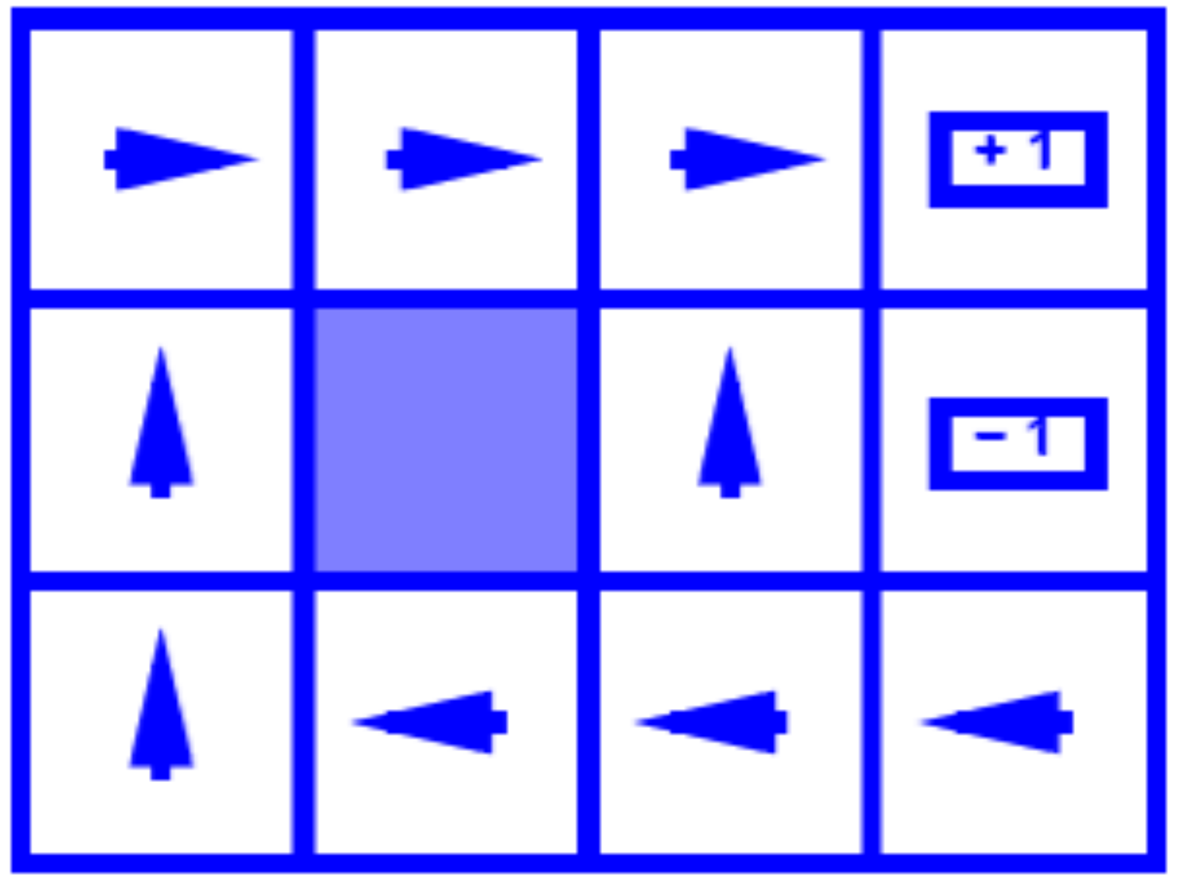


Optimal policy when R(s, a, s′) = -0.03 for all non-terminals s

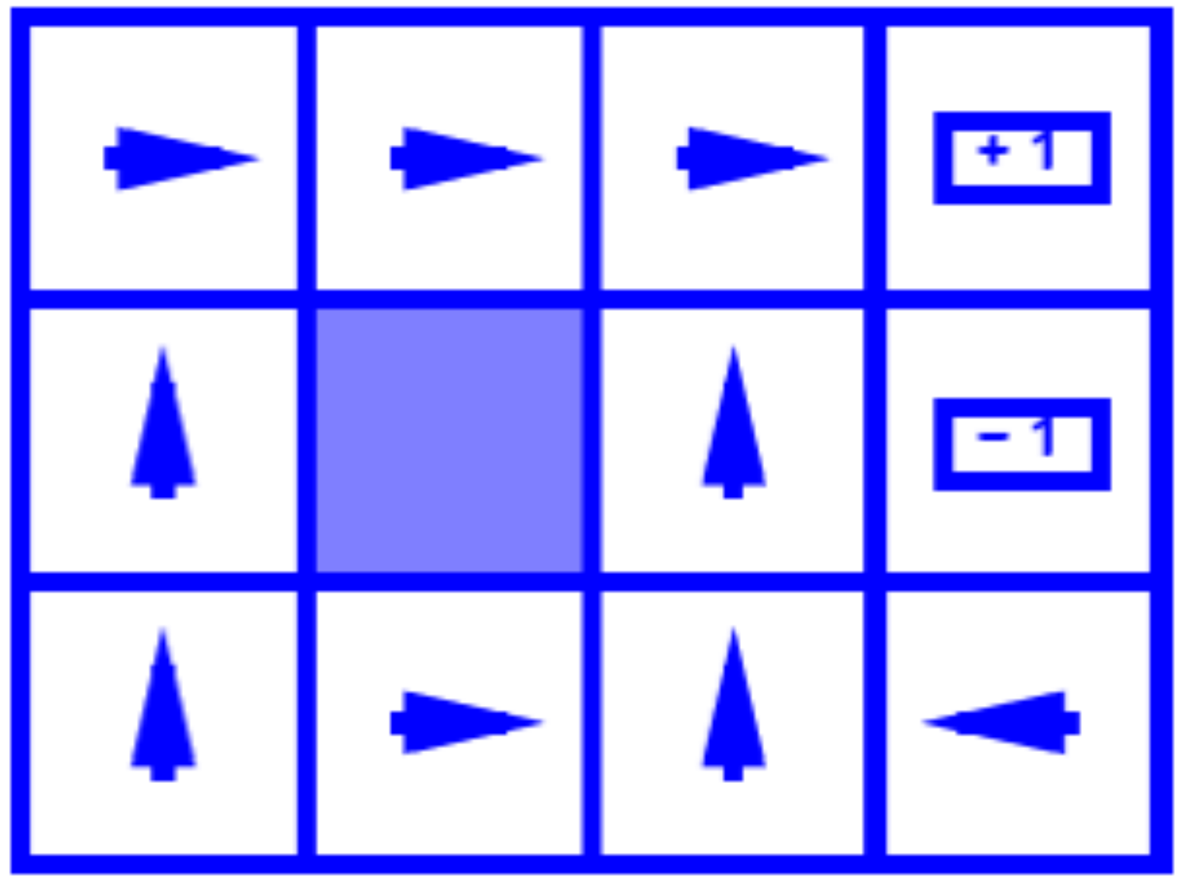
Example Optimal Policies



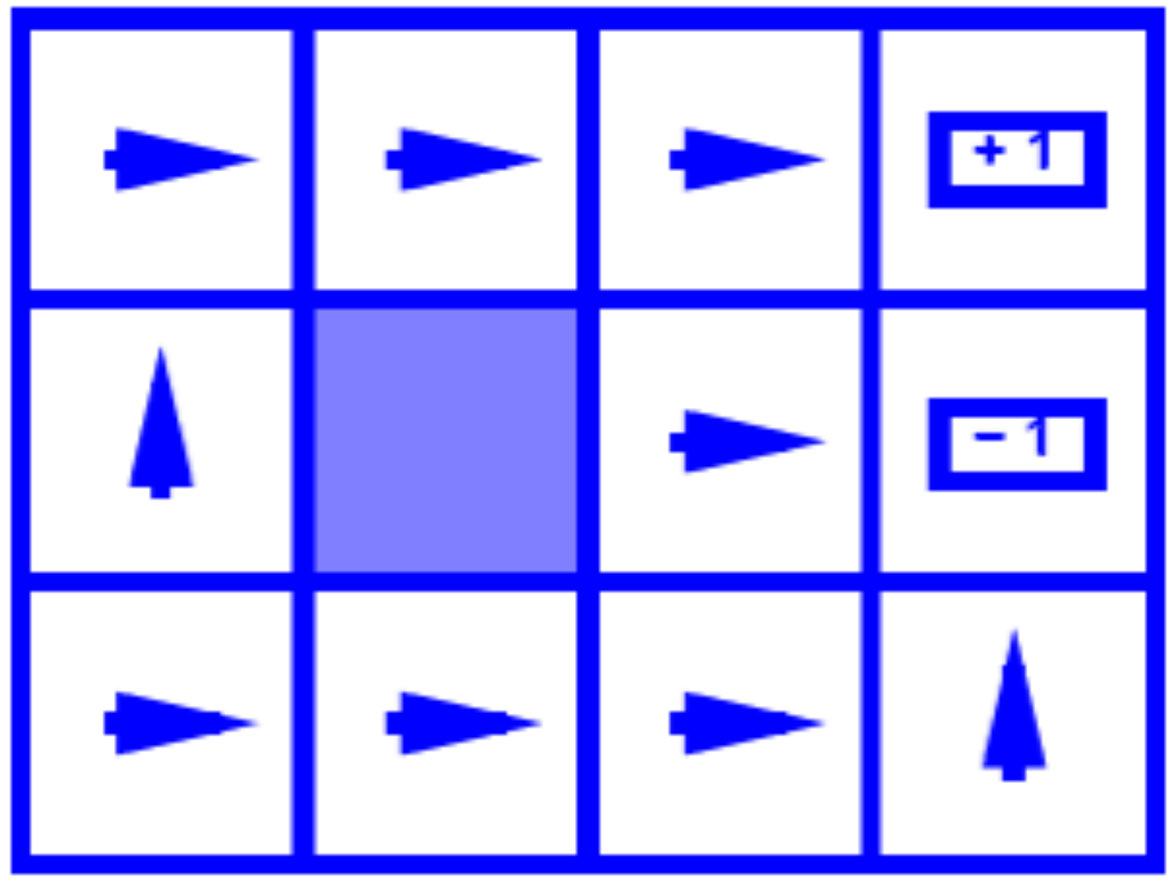
R(s) = -0.01



R(s) = -0.03



R(s) = -0.4



R(s) = -2.0

Utilities of Sequences

Utilities of Sequences

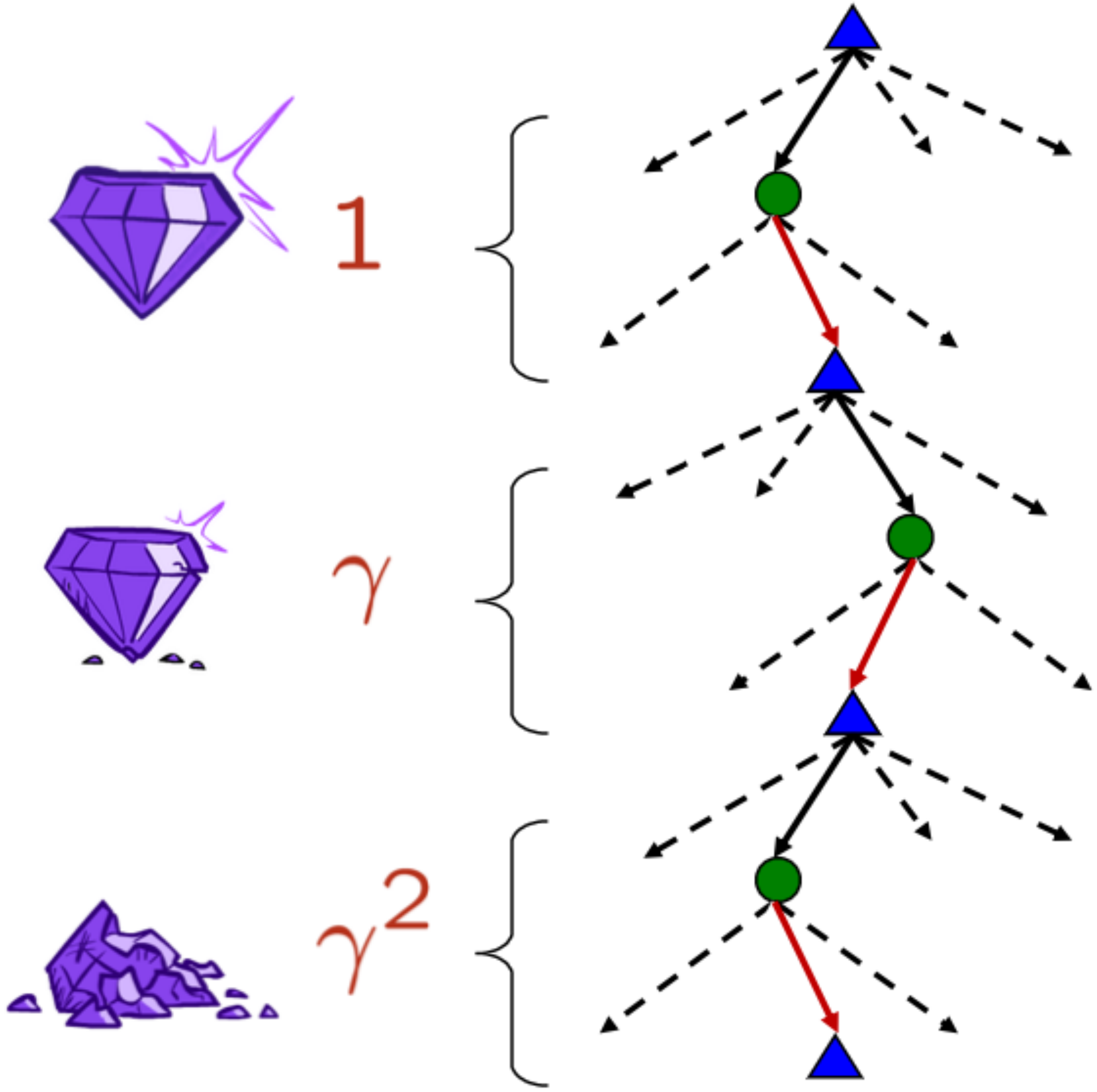
- What preferences should an agent have over reward sequences?
- More or less? [1, 2, 2] or [2, 3, 4]
- Now or later? [0, 0, 1] or [1, 0, 0]

Discounting

- It’s reasonable to maximize the sum of rewards
- It’s also reasonable to prefer rewards now to rewards later
- One solution: values of rewards decay exponentially

Discounting

- How to discount?
  - Each time we descend a level, we multiply in the discount once
- Why discount?
  - Think of it as a gamma chance of ending the process at every step
  - Also helps our algorithms converge
- Example: discount of 0.5
  - U([1,2,3]) = 1*1 + 0.5*2 + 0.25*3
  - U([1,2,3]) < U([3,2,1])



Summary and Next Time

- This week
  - Reasoning under uncertainty
    - Naïve Bayes
    - Bayes Networks
    - MDPs

- Next Week
  - Markov Decision Processes
  - Reinforcement Learning