- Project 2
  - Anticipated Score?
  - 3 min for 10 runs

- Normative vs. Descriptive
- Optimizer's Curse / Post-decision disappointment
  - Fig 16.3
  - 95% Confidence Interval
  - Mutual Funds, Winner's Curse
- Predictable Irrational (like economics!)
  - Anchoring: $1M plane voucher
  - Updating probabilities

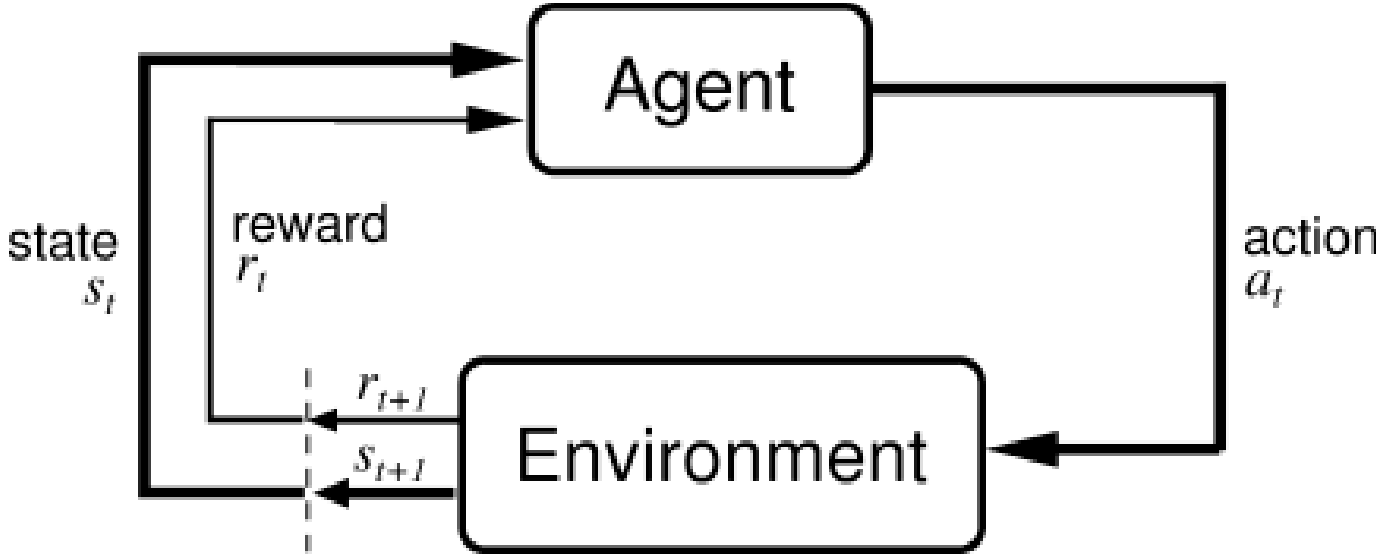
- http://www.cs.utexas.edu/~pstone/ComputersThought07/Movies/2005foxsports.mpg

- How similar to search?
- How different?

- How to make decisions?

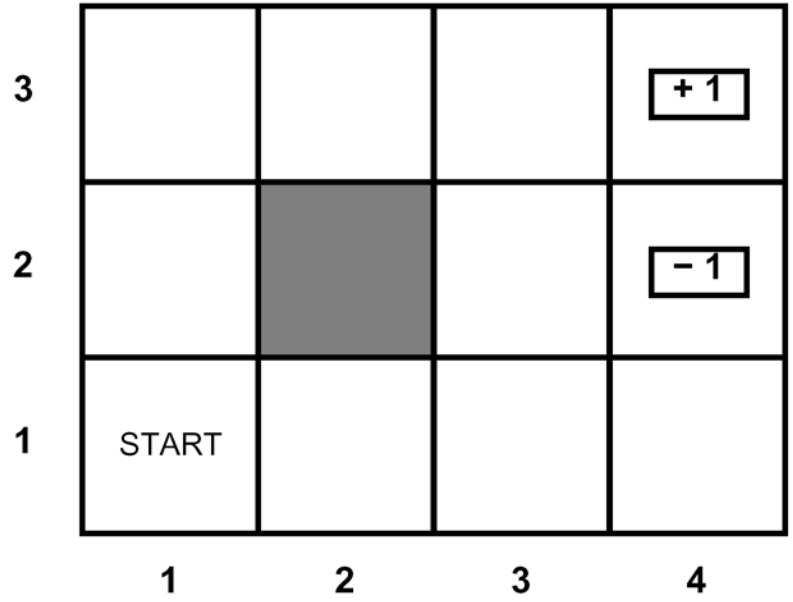
Reinforcement Learning

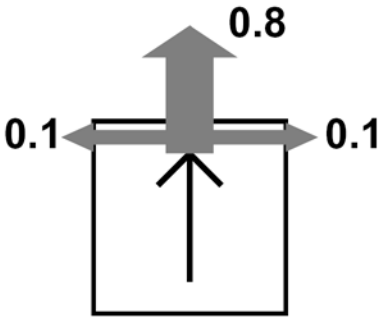
- Basic idea:
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must (learn to) act so as to maximize expected rewards



Grid World

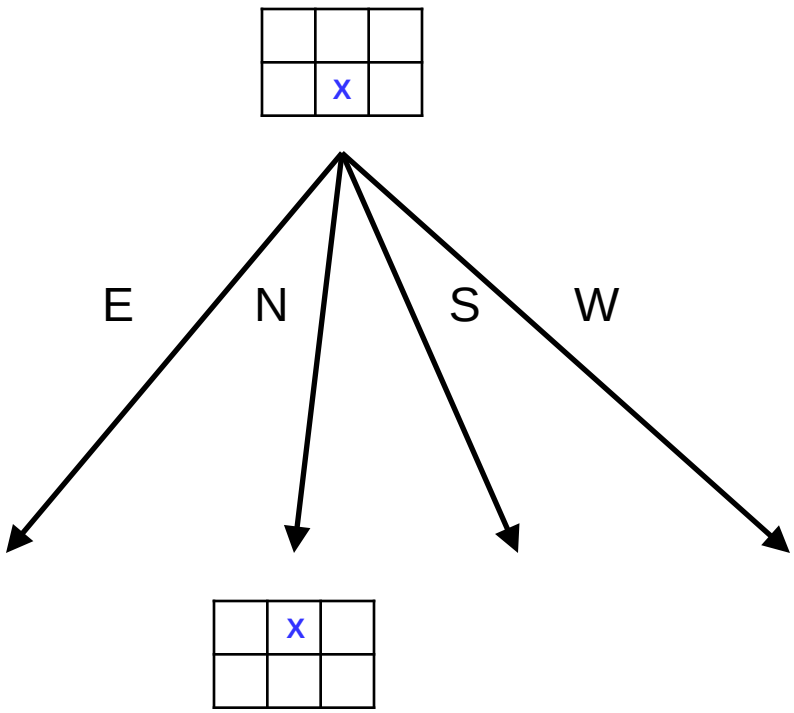
- The agent lives in a grid
- Walls block the agent's path
- The agent's actions do not always go as planned:
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent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



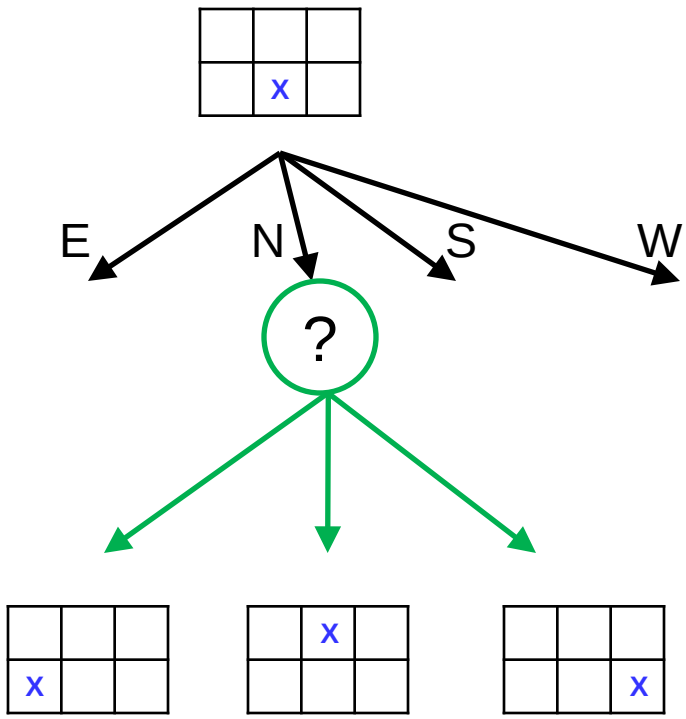


Grid Futures

Deterministic Grid World

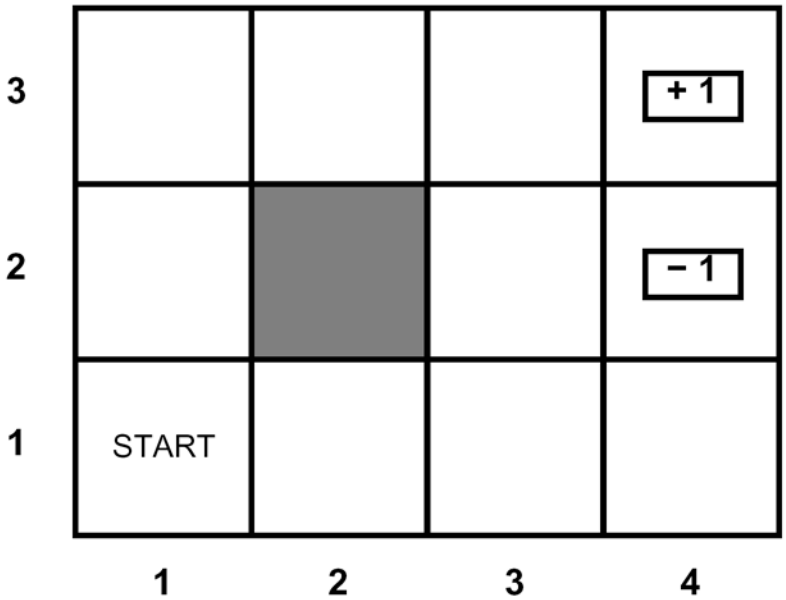


Stochastic Grid World

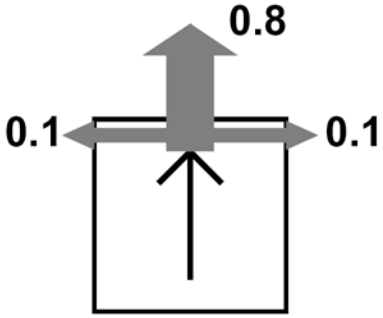


Markov Decision Processes

- An MDP is defined by:
  - A set of states $s \in S$
  - A set of actions $a \in A$
  - A transition function $T(s,a,s')$
    - Prob that a from s leads to s'
    - i.e., $P(s' \mid s,a)$
    - 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



- MDPs are a family of non-deterministic search problems
  - Reinforcement learning: MDPs where we don't know the transition or reward functions



Keepaway

- http://www.cs.utexas.edu/~AustinVilla/sim/keepaway/swf/learn360.swf

- SATR
- S_0, S_0

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:

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

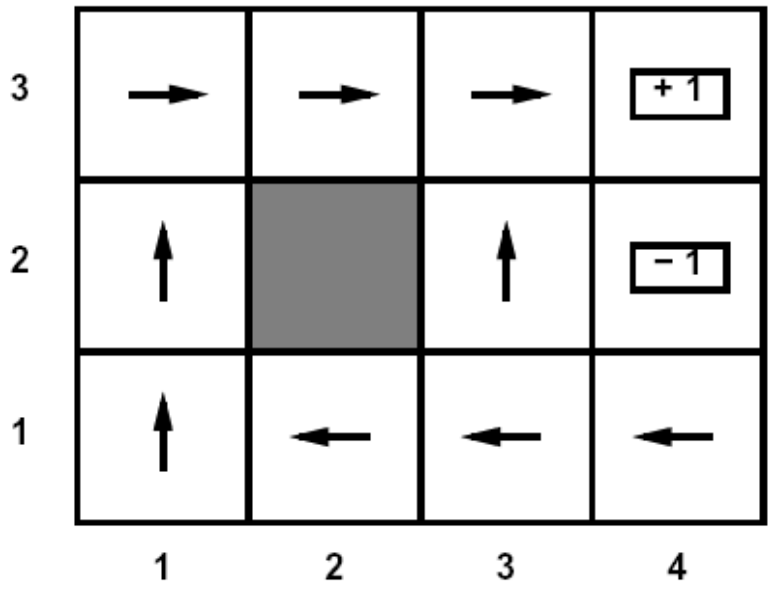
$$=$$

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

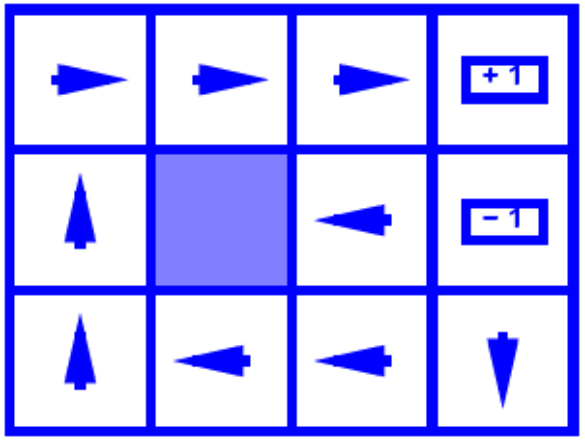
Solving MDPs

- 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 followed
  - Defines a reflex agent

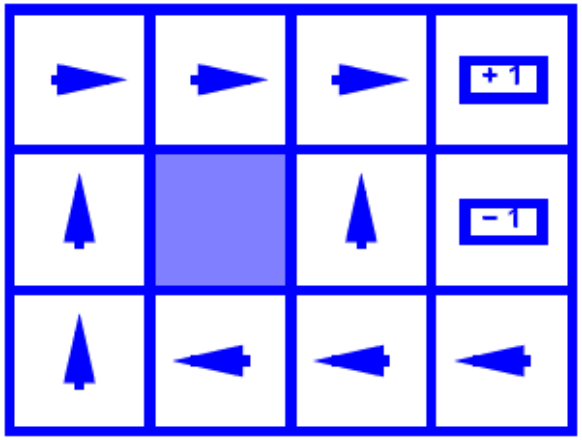
Optimal policy when R(s, a, s') = -0.03 for all non-terminals s



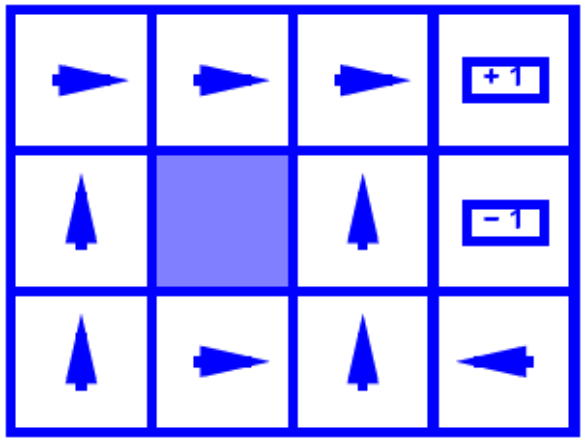
Example Optimal Policies



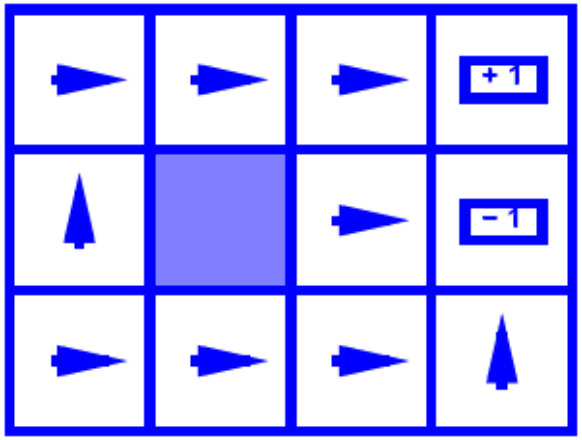
R(s) = -0.01



R(s) = -0.03



R(s) = -0.4



R(s) = -2.0