CS343
Artificial Intelligence

Prof: Peter Stone

Department of Computer Science
The University of Texas at Austin

Good Afternoon, Colleagues

Good Afternoon, Colleagues

Are there any questions?

Logistics

- Next week: RL

Logistics

- Next week: RL
- Multiagent assignment past due

Logistics

- Next week: RL
- Multiagent assignment past due
- RL assignment assigned...

Logistics

- Next week: RL
- Multiagent assignment past due
- RL assignment assigned... but due in 4 weeks

Logistics

- Next week: RL
- Multiagent assignment past due
- RL assignment assigned... but due in 4 weeks
  - Advice: start on it early. But if you get stuck, wait until after next week

Logistics

- Next week: RL
- Multiagent assignment past due
- RL assignment assigned... but due in 4 weeks
  - Advice: start on it early. But if you get stuck, wait until after next week
- Tournament details coming soon

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities
- **This week:** MDPs —

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities
- **This week:** MDPs — towards reinforcement learning

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function
  - Looking for a **policy** — optimal action from every state

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function
  - Looking for a **policy** — optimal action from every state
- **Next week:** Reinforcement learning

Some Context

- **First weeks:** search (BFS, A*, minimax, alpha-beta)
  - Find an optimal plan (or solution)
  - Best thing to do from the current state
  - Assume we know transition function and cost (reward) function
  - Either execute complete solution (deterministic) or search again at every step
- **Last week:** detour for probabilities and utilities
- **This week:** MDPs — towards reinforcement learning
  - Still know transition and reward function
  - Looking for a **policy** — optimal action from every state
- **Next week:** Reinforcement learning
  - Optimal policy without knowing transition or reward function

Pending Questions

- heuristic vs. eval. vs. reward vs. utility

Pending Questions

- heuristic vs. eval. vs. reward vs. utility
  - evaluation function f(n): a cost estimate of expanding a node - p.92
  - heuristic function h(n): estimated cost of the cheapest path from the state at node n to a goal state A*: f(n) = g(n) + h(n)
  - reward function r(s): one-step feedback in an MDP
  - utility function U(s): desirability of a state — final outcome or taking into account the future

Pending Questions

- heuristic vs. eval. vs. reward vs. utility
  - evaluation function f(n): a cost estimate of expanding a node - p.92
  - heuristic function h(n): estimated cost of the cheapest path from the state at node n to a goal state A*: f(n) = g(n) + h(n)
  - reward function r(s): one-step feedback in an MDP
  - utility function U(s): desirability of a state — final outcome or taking into account the future
- can AI make you rich (gambling)?
- axiom examples
- memory cost of non-stationary policies (Bert)
- how can you represent a policy in infinite state space?