Algorithmic Human-Robot Interaction

Planning Recap & Theory of Mind Papers

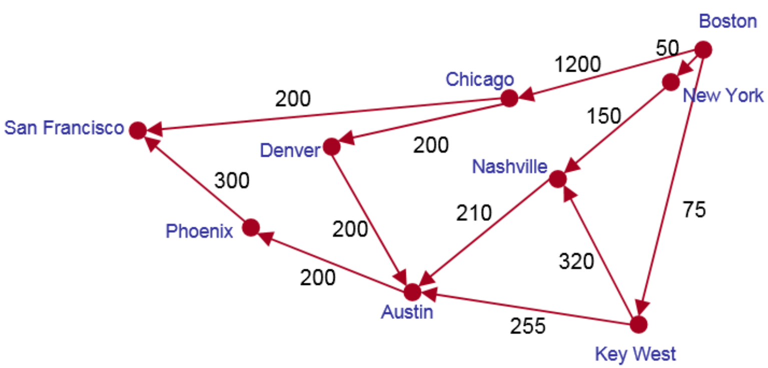
CSCI 7000

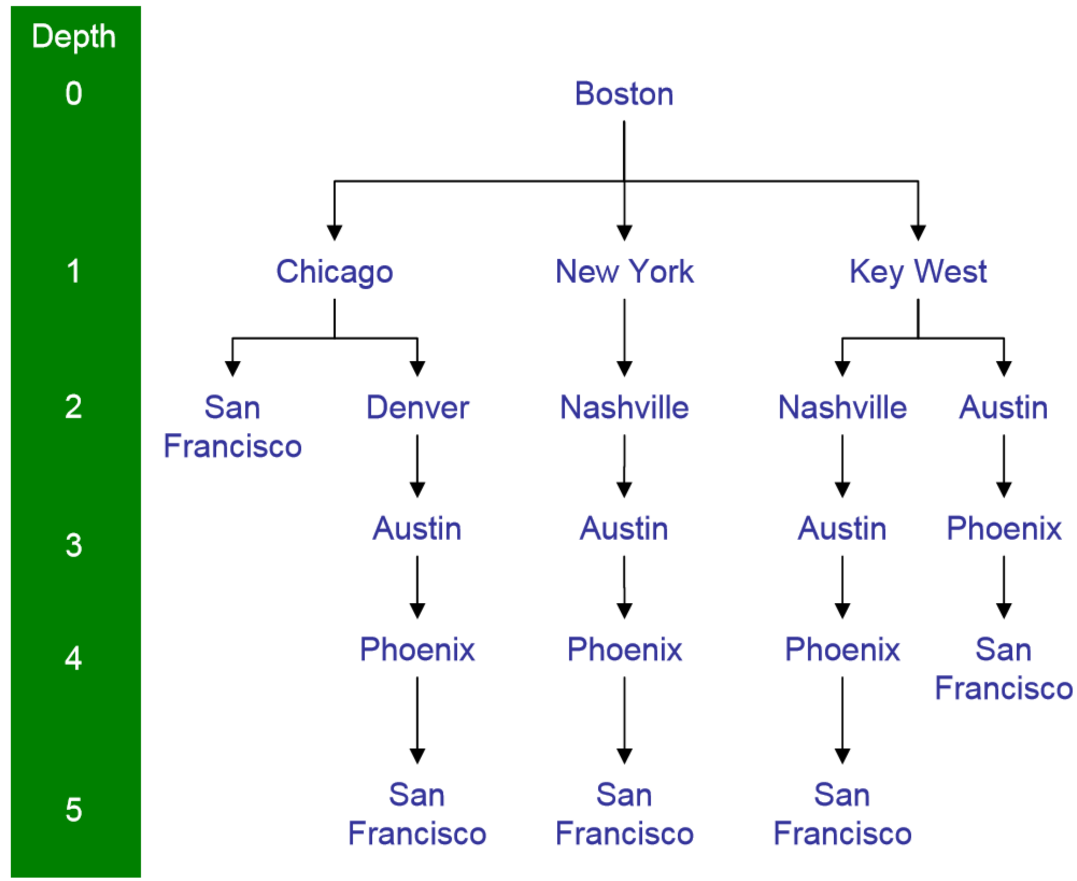
Prof. Brad Hayes

University of Colorado Boulder

Planning Recap

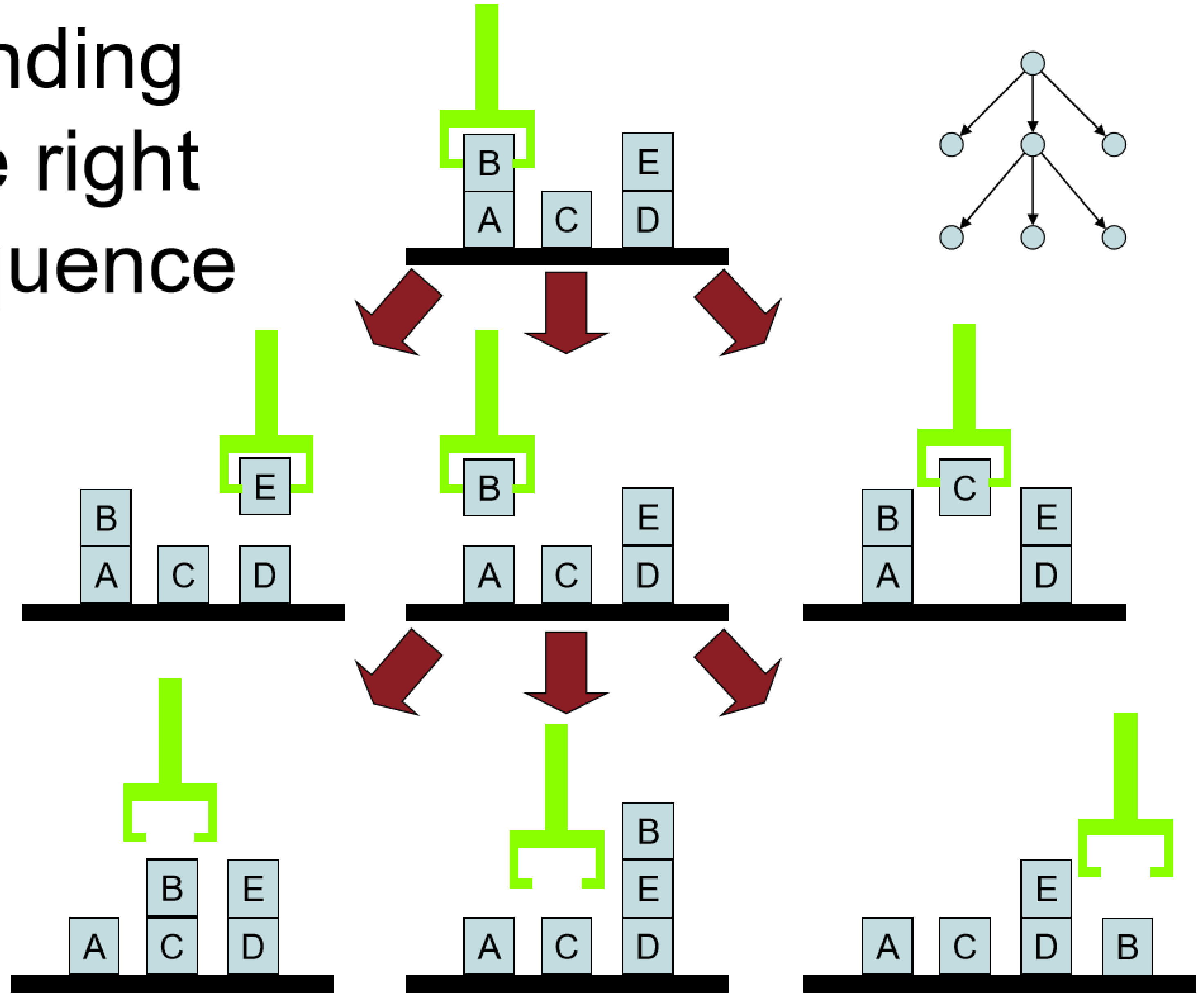
Search as a Problem-Solving Technique



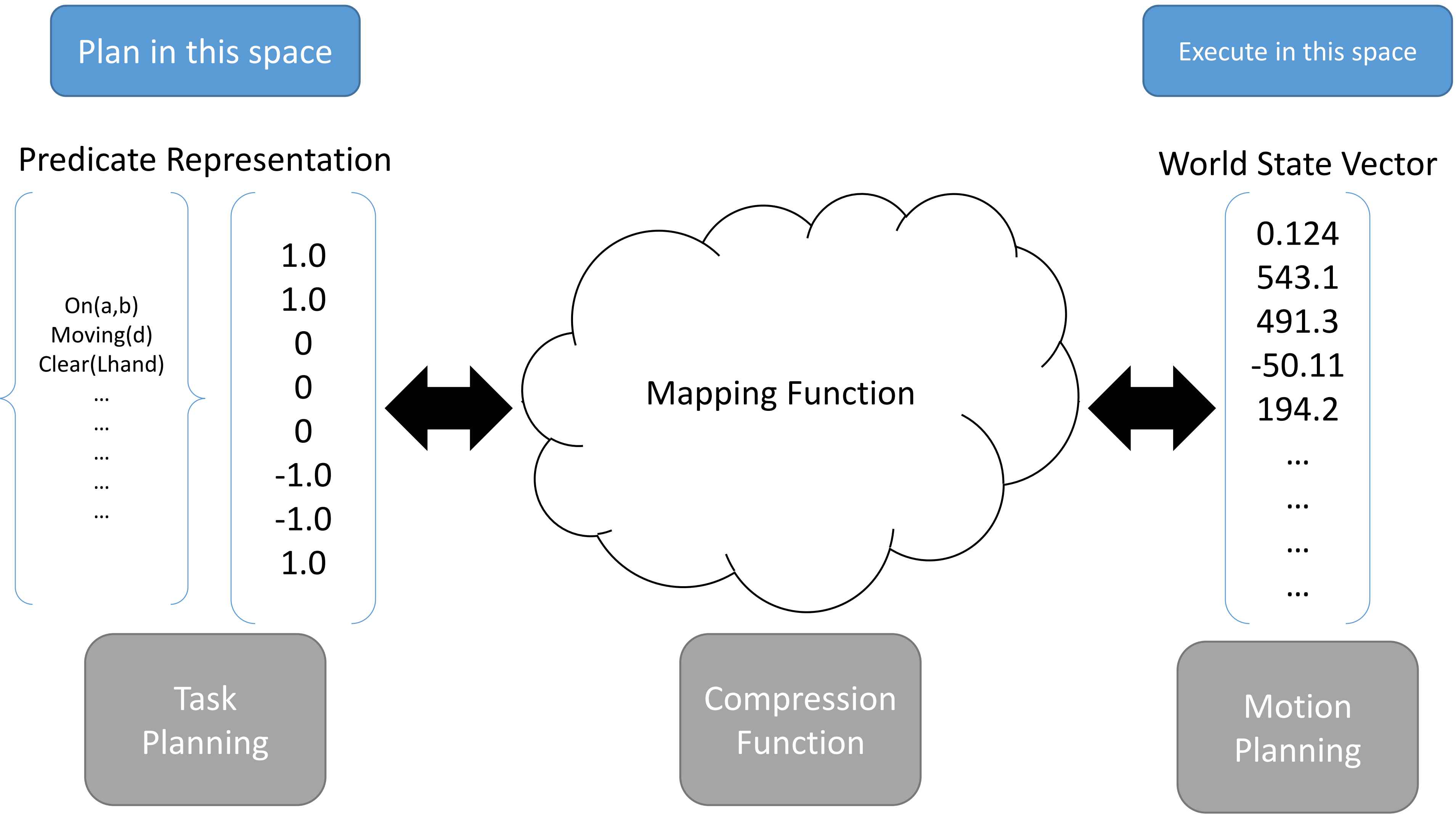


Branching Factor b=3

Finding the right Sequence

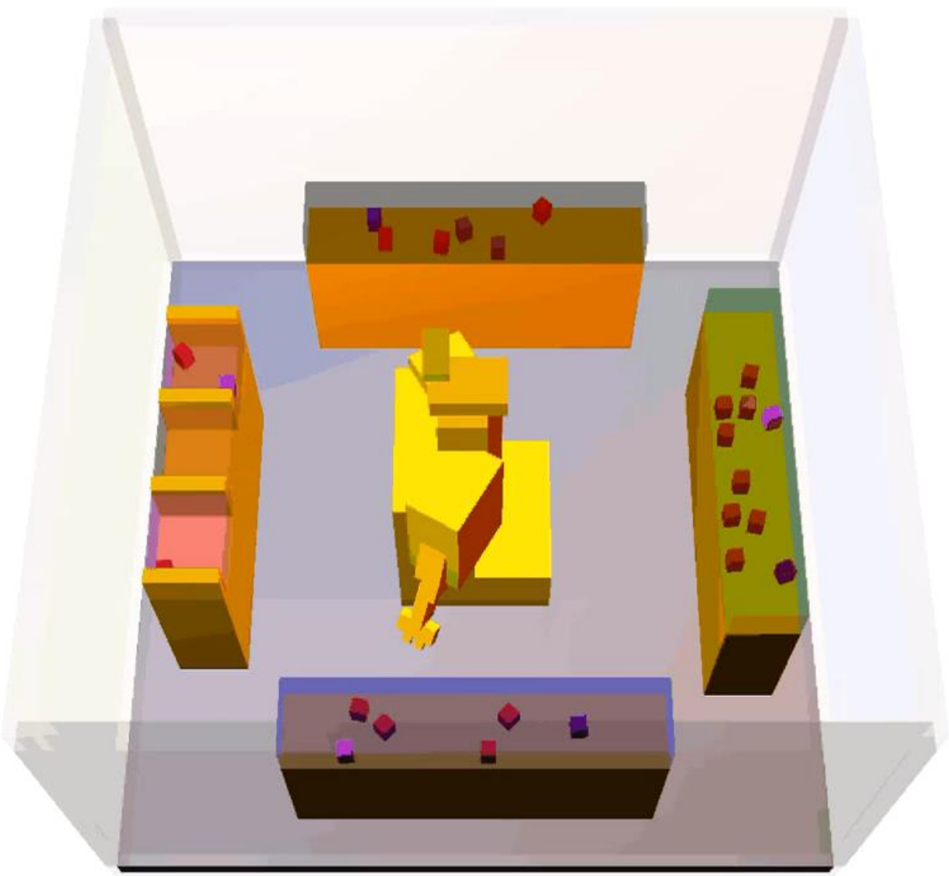


Lifted vs. Grounded Planning



Strategies for Tough Planning Problems

- Breadth-first search with sampled action parameterizations
- Pre-determined action parameterizations
  - e.g., Place(object,x,y,z)
- Plan for abstract 'areas'
  - **Lazy Grounding:**
    Sample parameterizations and add that action to plan if a valid parameterization is found
  - **Greedy Grounding:**
    Sample parameterizations and add that specific parameterization to the plan once a valid one is found



Planning Heuristics

FastForward Heuristic

- Remove deletions from all operators and run planner
- Resulting plan gives a heuristic for # steps from current state to goal
- FF Plan can provide ordering constraints on actions, even within partially specified plans

Learned Cost Function

- Learning Real-Time A* (LRTA*)
- Update heuristic dynamically: Value function V(s)
- Connection back to reinforcement learning: **V(s) and Q(s,a)** for search heuristic

Admissible Heuristics and Pruning Strategies

Admissible heuristics from delete-relaxation:

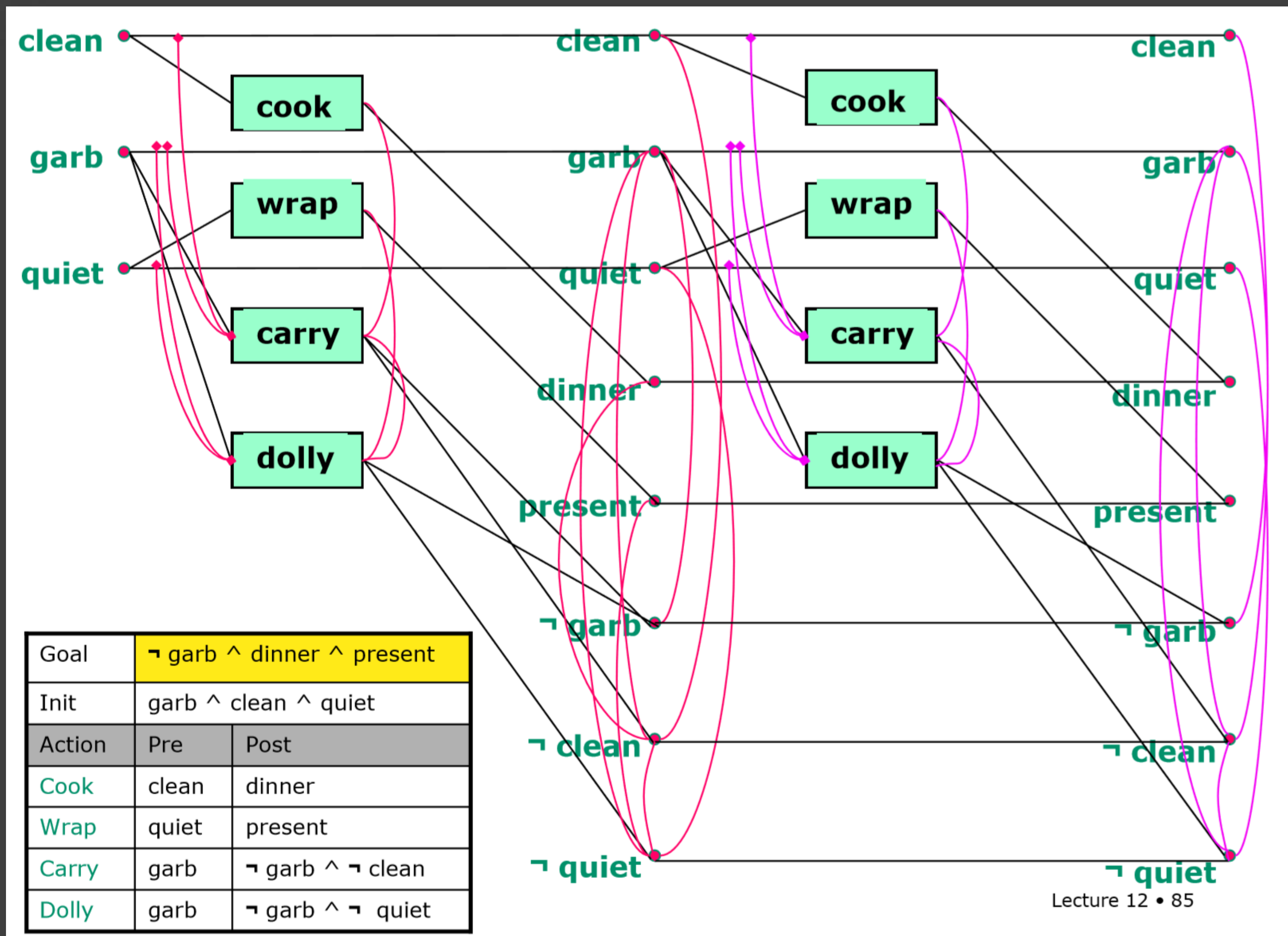
- $\boldsymbol{h_{max}}$ = i iff goal g is reachable in i steps from s
  - Admissible because of superposition (shown in GRAPHPLAN Algorithm)
- $\boldsymbol{h_{FF}}$ = number of **different** actions in $\pi(g)$

Search Pruning Strategies:

Novelty: Assign each state $novelty(s) = i$ if i predicates take on a value never seen in path from s_0 to s

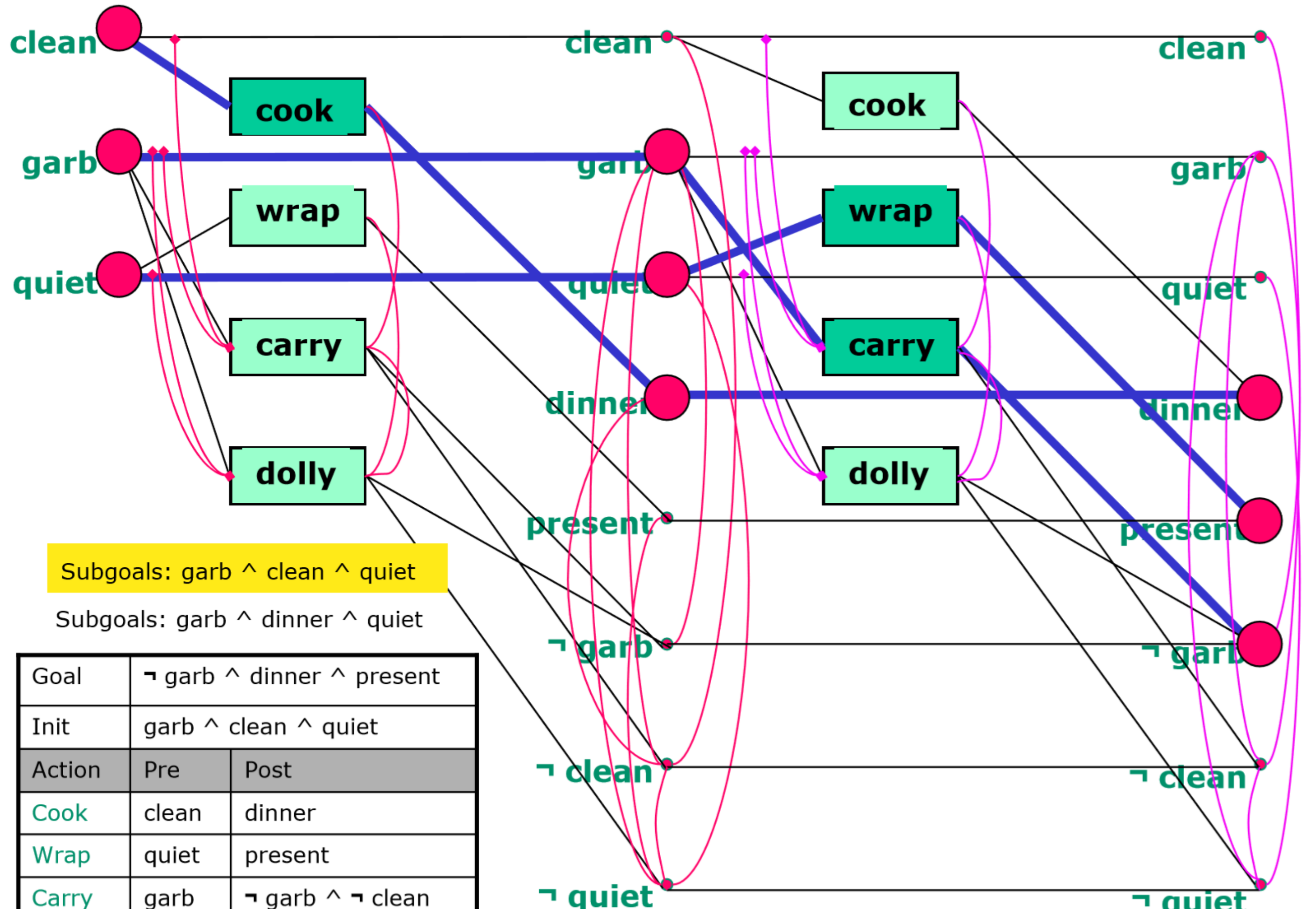
IW-search (Iterative Width): Ignore any nodes at the search frontier with $novelty(s) > i$.

GRAPHPLAN



Goal	¬ garb ^ dinner ^ present	
Init	garb ^ clean ^ quiet	
Action	Pre	Post
Cook	clean	dinner
Wrap	quiet	present
Carry	garb	¬ garb ^ ¬ clean
Dolly	garb	¬ garb ^ ¬ quiet

GRAPHPLAN Solution



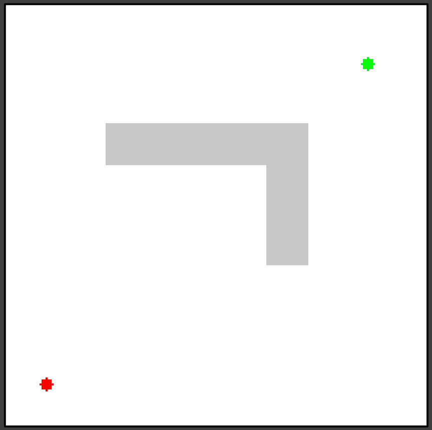
Subgoals: garb ^ clean ^ quiet

Subgoals: garb ^ dinner ^ quiet

Goal	¬ garb ^ dinner ^ present	
Init	garb ^ clean ^ quiet	
Action	Pre	Post
Cook	clean	dinner
Wrap	quiet	present
Carry	garb	¬ garb ^ ¬ clean
Dolly	garb	¬ garb ^ ¬ quiet

A* Search Algorithm

Adds heuristic to Dijkstra's Algorithm to bias exploration toward optimal path.

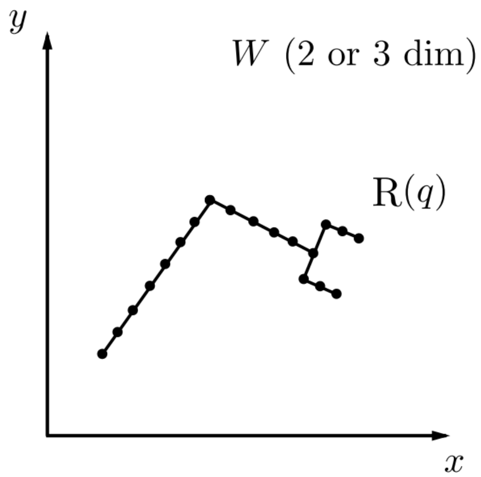


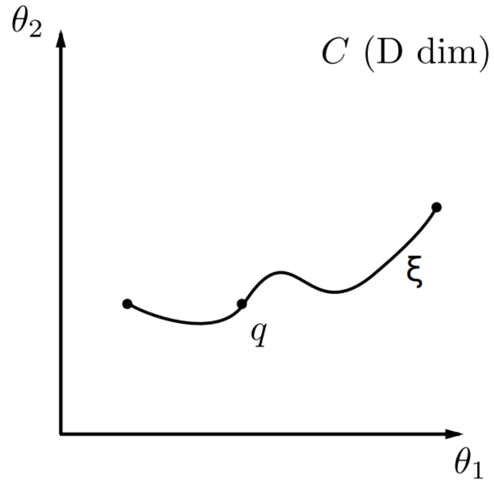
Insight

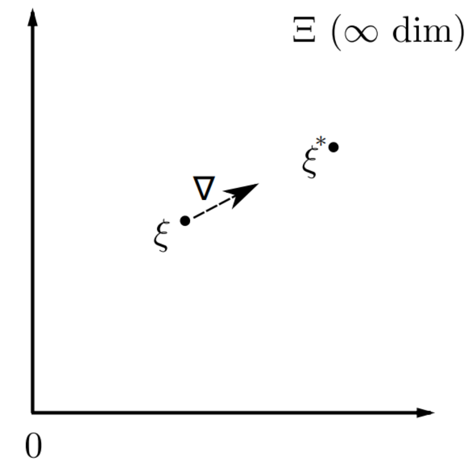
$$f(n) = g(n) + h(n)$$

$f(n) =$ Cost so far + Est. cost to go

Admissible Heuristic: An underestimate of the shortest possible path from node n to the goal.







World Space W(ℝ^3)

Configuration Space C($\#DoF$)

Trajectory Space $\Xi(\infty\ dim)$

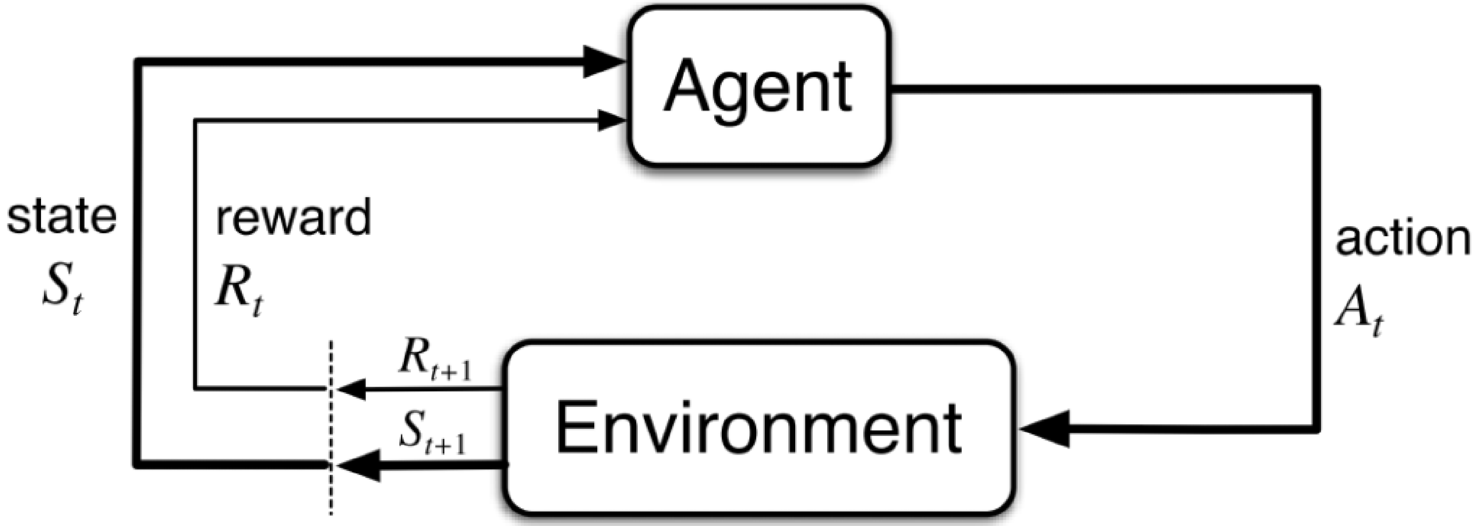
Robot pose in World Space (set of points)

Single point in Configuration Space

Trajectory through Configuration Space (set of points)

Single point in Trajectory Space

Rewind to January…



Rewind to January…

Reinforcement Learning

Function: $O \rightarrow A$

$\mathbf{o}$ $\pi_\theta(\mathbf{a}|\mathbf{o})$ $\mathbf{a}$

$\mathbf{s}_t$ – state

$\mathbf{o}_t$ – observation

$\mathbf{a}_t$ – action

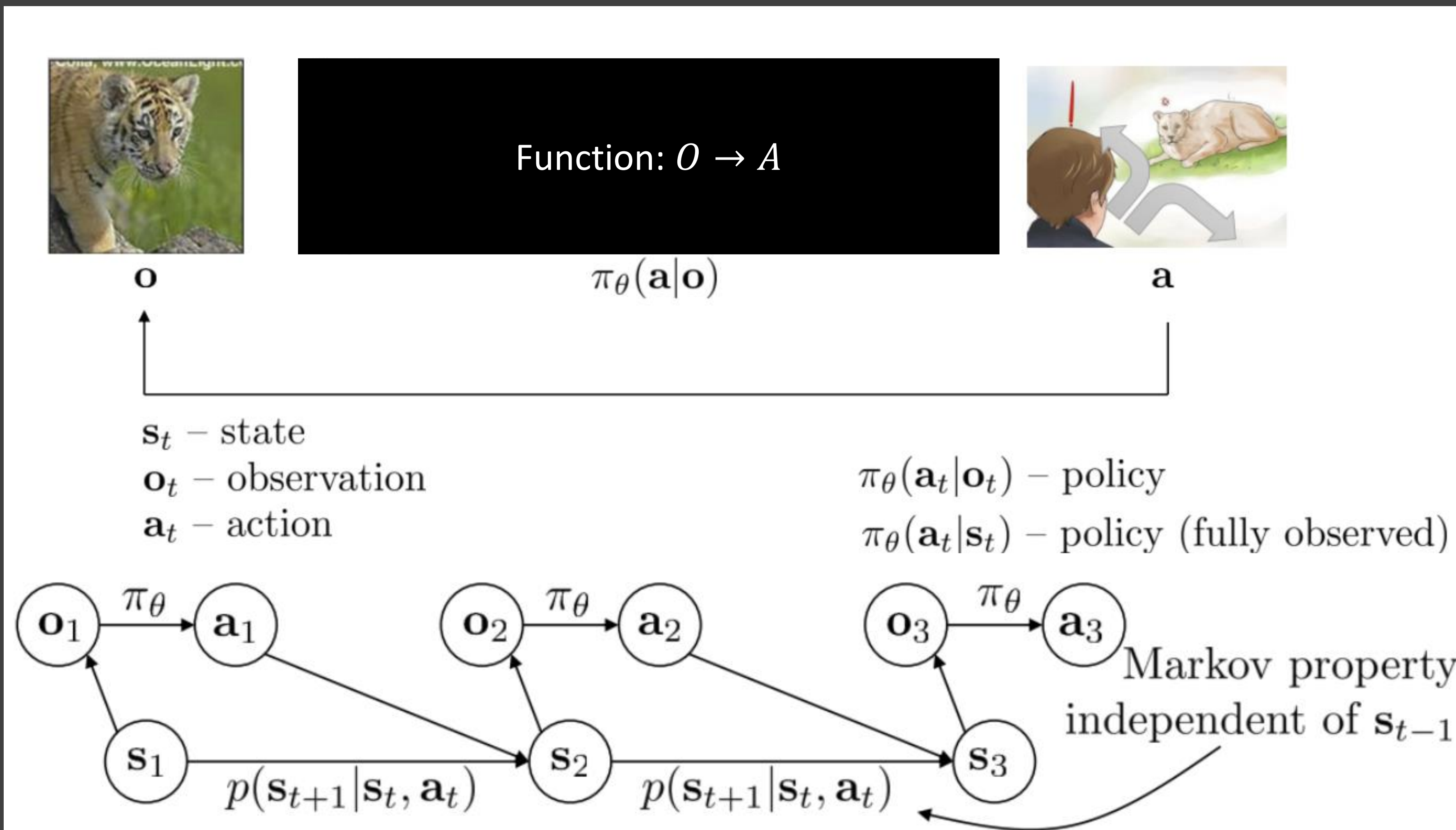
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy

$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

$\mathbf{o}_t$ – observation

$\mathbf{s}_t$ – state

Reinforcement Learning



Learning an Action-Value Function (Q-Learning)

- Q-value is the expected utility of taking an action in a state:

  Q(s,a) = Value of action ‘a’ in state ‘s’

- For a state s, choose action ‘a’ that maximizes Q(s,a)
- Q-values do not require an environment model:
  The transition function isn’t necessary, since it will be learned from experience.

$$Q(s,a) = (1-\alpha) * Q(s,a) + \alpha(\ r + \ \gamma\ * \max_{a'} Q(s',a'\,))$$

Reliance on existing vs. new Knowledge	Current understanding	reward	Discount factor	Best we think we can do in the future from here

Learning Real-Time A*

- Hill-climbing algorithm where the **best child node** is selected and others **discarded**, and the heuristic is updated dynamically.

A* Cost Formula: Dictates order of node expansion

- $f(s) = \; g(s) + \; h(s)$
- $g(s)$ remains as cost already paid to get here: $\sum_{i=0}^{n} c(s_i, a_i)$
- $g(s_{i+1}) = g(s_i) + c(s_i, a_i)$
- Replace $h(s_i)$ with $argmax_a \, Q(s_i, a)$

$$f(s) = g(s) + argmax_a \, Q(s_{i+1}, a)$$

$$Q(s,a) = \cancel{(1-\alpha) * Q(s,a) + \alpha}(\; r + \; \gamma \; * \max_{a'} Q(s', a' \,))$$

Today

Today's Papers

Planning + Predicting Human Behavior

Probabilistically Safe Robot Planning with Confidence-Based Human Predictions

Jaime Fisac et al.

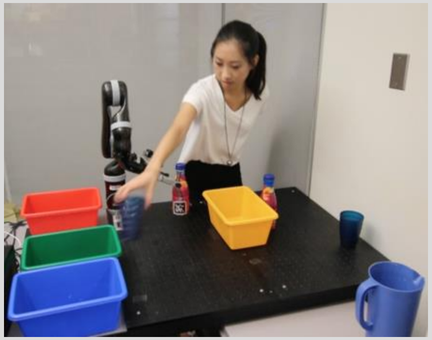
An Implemented Theory of Mind to Improve Human-Robot Shared Plans Execution

Sandra Devin and Rachid Alami

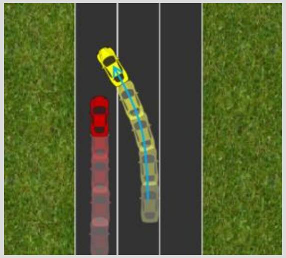
Papers for Next Thursday (2/21)

Modeling Capability and Effect

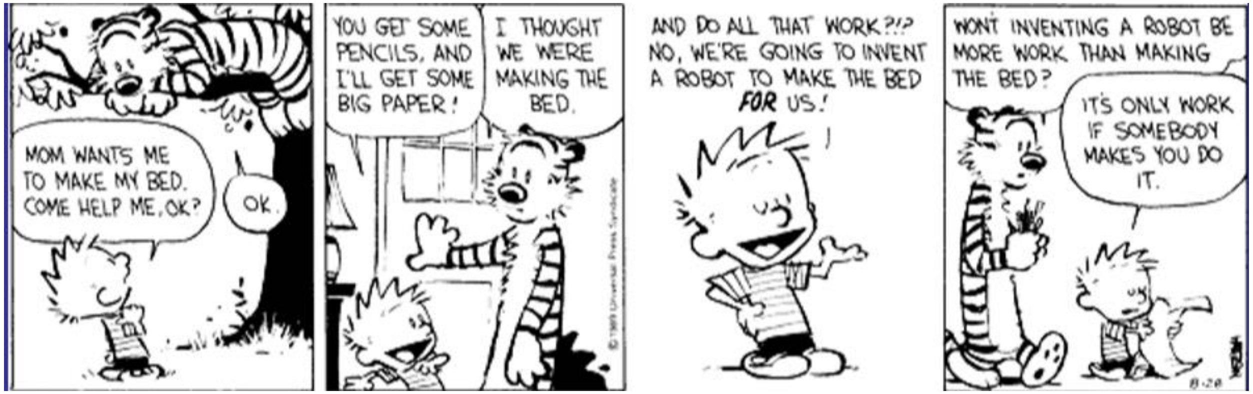
Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration
Stefanos Nikolaidis et al.



Planning for Autonomous Cars that Leverage Effects on Human Actions
Dorsa Sadigh et al.

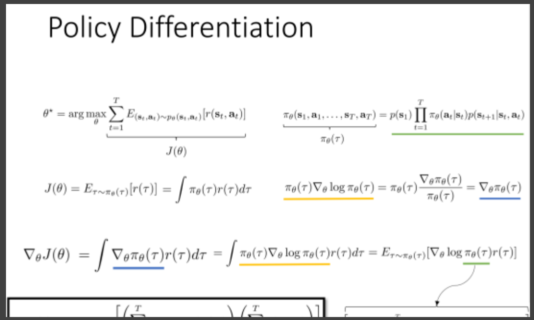


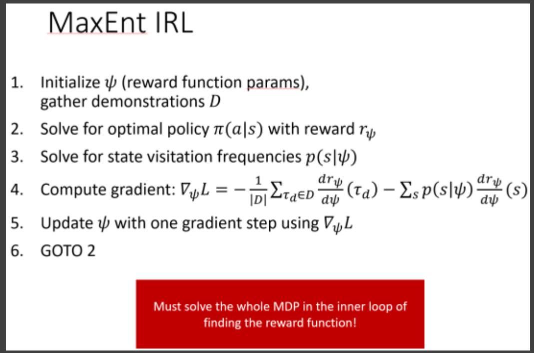
Final Project Outlines

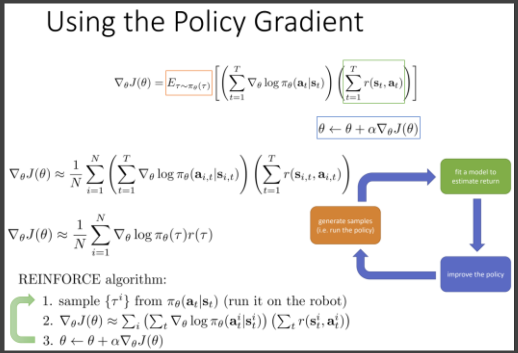


Due Friday (3/1), at 5:00pm via Moodle

- Major deliverables (+ assigned lead)
- Describe technical components (e.g., “face detector”)
- Describe evaluation
- Schedule for design, implementation, evaluation, writing
- Concerns / risks / challenges







Coming Soon

Project Workshops

Stochastic Gradient Descent for Policy Learning
(in continuous action spaces)

Inverse Reinforcement Learning