Regression Search

Regression

Progression: full initial state, partial goal state (set of goals)

- Initial node: set of states with true goal
- Applicable: at least one effect present, deletes not present, non-deleted preconditions present
- Child node: remove adds, add preconditions
- Goal nodes: subset of initial state

Lower branching factor
More relevant unification

Grocery World

Initial: At(Home), Sells(HD, Drill), Sells(MB, Milk), Sells(MB, Bananas)

Go (here, there)

Pre: At(here)

Post: At(there), ¬At(here)

Buy (store, x)

Pre: At(store), Sells(store, x)

Post: Have(x)

Goal: At(Home), Have(Drill), Have(Milk), Have(Bananas)

Comparison

- Forward: states
  - + exact state known
  - + strong heuristics
  - + reachable states
  - – branching factor
  - – irrelevant states
- Regression: sets of states
  - + relevant states
  - – unreachable states
  - – partial states: larger search space

More Planning

Planning assumes: static, deterministic, discrete time, single actions

- time, resource constraints
- concurrent actions
- abstraction: hierarchical planning
- uncertainty
- replanning
- continuous state
- multiple agents

Course Outline

1. Search: heuristics, CSPs, games
2. Knowledge representation: Propositional, FOL, resolution
3. Planning: PDDL, MDPs
4. Learning: RL, supervised, unsupervised
5. Uncertain knowledge: HMMs, Bayes nets

Markov Decision Processes

Probability

- propositional
- domain: discrete or continuous
- 0-1 chance of outcome, sum to 1
- distribution if continuous: density
- $E(X) = \int xP(X = x)dx$
- $P(X = x_1)$ written as $P(x_1)$, or $P(x)$ in boolean domain
- conditional (posterior): $P(x|y) = P(x \wedge y)/P(y)$

Markov Decision Processes – MDP

- **initial state**: s_0
- **transition model**: $T(s,a,s')$ = probability of going from state s to s' with action a
- **reward function**: $R(s)$ for arriving in state s
- **terminal state**: a sink state with no actions out, end of process

objective:

- **total reward**: reward over trajectory:
  - $R(s_0)+R(s_1)+\cdots+R(s_n)$
- **discounted reward**: penalize future rewards by γ
  - $R(s_0)+\gamma R(s_1)+\gamma^2 R(s_2)+\cdots+\gamma^n R(s_n)$

Markov Decision Processes – MDP

find:

- **policy**: $\pi(s) = a$
- **optimal policy**: π^*
- **proper policy**: reaches terminal state

Stochastic SampleWorld

- Actions: Up, Down, Left, Right, Sample
- Up action: .7 chance of going up, .1 of each other direction
- Rewards: +1 for sampling marked square, -1 for sampling unmarked square, -.05 for moves