asst 8 is posted
The Planning Graph

- Simple Heuristics
- Planning Graphs
- Cake World
- Break
- Relaxed Plan
- Comparison
- Heuristics
- Concurrent Actions
- EOLQs
1. \( h(n) = 0 \)
2. number of unachieved goals
3. reachability (’don’t delete’): \( H_1 \) max
4. \( H_1 \) sum
The ‘Planning Graph’

2 types of layers: fact and action
track both positive and negative grounded literals
‘no-op’ frame actions

actions $a$ and $b$ mutex iff:

**inconsistency:** $a$ deletes add of $b$

**interference:** $a$ deletes precondition of $b$

**competing needs:** inconsistent preconditions

literals $a$ and $b$ mutex iff:

**inconsistent:** $a$ is $\neg b$

**inconsistent support:** all ways of achieving them are mutex
Initial: Have(Cake)

**Eat:** Pre: Have(Cake)
Post: ¬Have(Cake), Eaten(Cake)

**Bake:** Pre: ¬Have(Cake)
Post: Have(Cake)

Goal: Have(Cake), Eaten(Cake)
Break

- project proposals
- asst 7
- asst 8
$H_1$ max too small, sum too large

Basic graph assumes parallelism: serial planning graph

building a relaxed plan:
- choose no-op when possible
- re-use previously chosen action when possible

optimal relaxed plan is admissible but NP-hard

need actions if optimizing costs (not makespan)
level-based heuristics

1. poor if many ‘concurrent’ actions at one level
   max vs sum

1. sum poor if positive interactions
   \( h^n \)

1. poor if negative interactions
1. 0
2. number of unachieved goals
3. $H_1$ max
4. $H_1$ sum
5. planning graph max
6. planning graph sum
7. relaxed plan
Concurrent Actions

$2^k$ vs incremental
What question didn’t you get to ask today?
What’s still confusing?
What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.

Thanks!