State-space Planning

Problem Types

- actions: serial or parallel
- actions: unit time or varying
- actions: unit cost or varying
- minimize makespan, cost, combination, or multi-objective
- just logical fluents or metric quantities (eg, resources)
- off-line or on-line planning
- world controlled or has autonomous (predictable) dynamics
- ‘single agent’ or other agents modifying state
- actions: deterministic or stochastic
- states: fully, partially, or not observable
- initial state known or unknown
- single goal state or set
- goals of achievement or maintenance
- action space: discrete or continuous
- state space: discrete or continuous

Frame Problems

- representational: how to represent what doesn’t change
- inferential: how to compute new state quickly
- qualification: how to represent preconditions

Planning Domain Definition Language

Operator schema:

Parameters: Move(block, src, dest)

Preconditions: On(block, src), Clear(block), Clear(dest)

Delete list: On(block, src), Clear(dest)

Add list: On(block, dest), Clear(src)

Assume everything else is static – closed world assumption

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)

Progression

- Initial state: initial truths
- Branch on all applicable actions
- Applicable: preconditions hold
- Effects: delete deletes, add adds
- Goal reached when all goal atoms are true.

This can be framed as a state-space search problem!

Forward search: start with initial conditions and apply valid actions until goal
Backward search: start with goal conditions and apply reversed actions until initial conditions

Heuristic Functions

Simple Heuristics

Reminder: heuristic functions estimate cost from current state to goal state by *relaxing the problem* – removing constraints

- $h(n) = 0$
- number of unachieved goals
- reachability – no deletes: h^+

Computing h^+

$h^+(I)$

```
$t \leftarrow 0$
$Q \leftarrow I$
while $Q \neq \varnothing$ and a goal is false do
    $Q' \leftarrow \varnothing$
    for each $l \in Q$ do
        for each $a$ that has $l$ as a precondition do
            if all of $a$'s preconditions are true then
                for each effect $e$ of $a$ do
                    if $e$ is not true then
                        record that $e$ became true at $t+1$
                        add it to $Q'$
    $t \leftarrow t+1$
    $Q \leftarrow Q'$
```

h^{max} and h^{add}

There are two ways we can calculate h^{+}:

- h^{max}: the maximum t required to satisfy all goals
- h^{add}: the sum of all t of goals

Are any of these admissible?

Cake World

Initial: Have(Cake)

Eat:

Pre: Have(Cake)

Post: Eaten(Cake), ¬Have(Cake)

Bake:

Pre: ¬Have(Cake)

Post: Have(Cake)

Goal: Have(Cake), Eaten(Cake)