Lecture03a: Hierarchical Search

Announcements
First Programming Assignment is due today
Up to midnight, but no later
First Reading Assignment is due Thursday

PA1
◦ What is your search space? How big is it?
◦ How do you do inference?
◦ How do you select possible actions?
◦ How do you avoid state repetition?
◦ What is your heuristic function?
◦ What does your search tree look like?

Making the Problem Harder...
Introducing wildcards to goal states
(is block3 wildcard1 on-top-of)
Some block is on top of block
(has wildcard1 color red)
The block on block3 must be red
(is wildcard1 wildcard2 side-by-side)

... and Still Harder
Introducing Geometry
◦ Table is a grid
◦ Blocks occupy one grid cell (they 1x1x1 in shape)
◦ Table is 10x10 surface, from 0,0 to 9,9
◦ Blocks on the table have height 0
◦ Blocks on other blocks have the height of the other block +1

Location is an initialized property (in the start state)
◦ (has block1 location xpos ypos zpos)
◦ All blocks have a starting location

Changing Locations
There are now a total of four commands:
(command slide block-id xdir ydir)
◦ Where xdir & ydir are -1, 0, or 1
◦ Preconditions:
  ◦ block-id height is 0
  ◦ Location block-id position + xdir, block-id position + ydir is empty
  ◦ No block is currently grabbed
◦ Postcondition
  ◦ Block id moves position+xdir, position+ydir
  ◦ Blocks on top of block-id move the same amount
Continued….

(command grab block-id)
Grabs the given block so that you can carry it
Precondition:
- No other block is grabbed
- No other block is on top of block-id
Postcondition
- Block-id is grabbed

(command release block-id)
Precondition:
- block-id is grabbed
- Block-id is on the table (height = 0) or another block
Postcondition:
- block-id is no longer grabbed

Still more…

Command carry block-id xdir, ydir, zdir)
Xdir, ydir and zdir are all -1, 0 or 1
Preconditions
- Block-id is grabbed
- Block-id zposition + zdir >= 0
- Block-id xposition + xdir, yposition + ydir, zposition+zdir is empty
Postcondition
- Block is now at xposition + xdir, yposition + ydir, zposition+zdir

What happened to relations?

Initial states contain block properties
- Locations
- Colors

Goal states include
- Locations
- Colors
- Relations

Relations can be inferred
- Blocks are side-by-side if their location is different by 1 in either x position or y position (but not both), and their heights are the same
- Blocks are on-top-of if x and y positions match, but one is 1 higher than the other.

Notes

Relations are like wildcards. The goal state could be:

(has block3 location 5 5 0)
(is block3 wildcard1 side-by-side)
(is block1 wildcard1 on-top-of)

This says that some other block must be at one of the four available sides of block3.

More Notes

When blocks are stacked, sliding moves multiple blocks with one command
- Highly efficient

Hierarchical Planning

The actions in PA2 are low level
- Slide, grab, carry, release

If you search (plan) at the level of actions, the search space quickly explodes

If you plan at the level of goals, the problem becomes hierarchical
- In this task, goals are relations
Example

Here is a goal state:
- (has block3 location 5 5 0)
- (is block3 wildcard1 side-by-side)
- (is block1 wildcard1 on-top-of)

Planning at the level of relations, your first goal might be
- (has block3 location 5 5 0)

Your next goal might be
- (has block2 location 5 6 0)
  - Satisfies the 2nd relation in goal state
  - Of course, there are many choices of block and 4 choices of location

Final goal:
- (has block1 location 5 6 1)

Example (cont.)

Of course, that’s not a plan

Now you need a lower level planner to satisfy location goals
- This is a path planner
- Sub-goals don’t interact

Downsides
- Missed opportunities for efficiency
  - E.g. stack block1 on block2, slide both toward block3
- Sub-goals may fail or need to be solved recursively
  - Block 1 may be surrounded by 8 other blocks, plus one on top
  - In which case, you can’t move it.
  - Need to plan to clear it first…