Informatics 2D: Reasoning and Agents

Alex Lascarides

School of informatics

THE UNIVERSITY OF EDINBURGH

Lecture 16c: Examples in PDDL

Where are we?

Last time . . .

- Planning Domain Definition Language (PDDL)

Now:

- Some examples

Blocks world example

- Given: A set of cube-shaped blocks sitting on a table
- Can be stacked, but only one on top of the other
- Robot arm can move around blocks (one at a time)
- Goal: to stack blocks in a certain way
- Formalisation in PDDL:
  - $On(b, x)$ to denote that block b is on x (block/table)
  - $Move(b, x, y)$ to indicate action of moving b from x to y
  - Precondition for this action requires $Clear(z)$: nothing stacked on z.

Blocks world example

- Action schema:

$$Action(Move(b,x,y),$$
$$\text{Precond:}On(b,x) \wedge Clear(b) \wedge Clear(y)$$
$$\text{Effect:}On(b,y) \wedge Clear(x) \wedge \neg On(b,x) \wedge \neg Clear(y))$$

- Problem: when $x = Table$ or $y = Table$ we infer that the table is clear when we have moved a block from it (not true) and require that table is clear to move something on it (not true)
- Solution: introduce another action

$$Action(MoveToTable(b,x),$$
$$\text{Precond:}On(b,x) \wedge Clear(b)$$
$$\text{Effect:}On(b, Table) \wedge Clear(x) \wedge \neg On(b,x))$$

Does this Work?

- Interpret $Clear(b)$ as "there is space on b to hold a block" (thus $Clear(Table)$ is always true)
- But without further modification, planner can still use $Move(b, x, Table)$:
  - Needlessly increases search space (not a big problem here, but can be)
- So part of solution is to also add $Block(b) \wedge Block(y)$ to precondition of $Move$

Summary

- We have now defined a language for expressing planning problems
- Blocks world example as a famous application domain
- Discussed how to address some specific problems in representing states and actions
- Next time: Algorithms for planning!
  State-Space Search and Partial-Order Planning