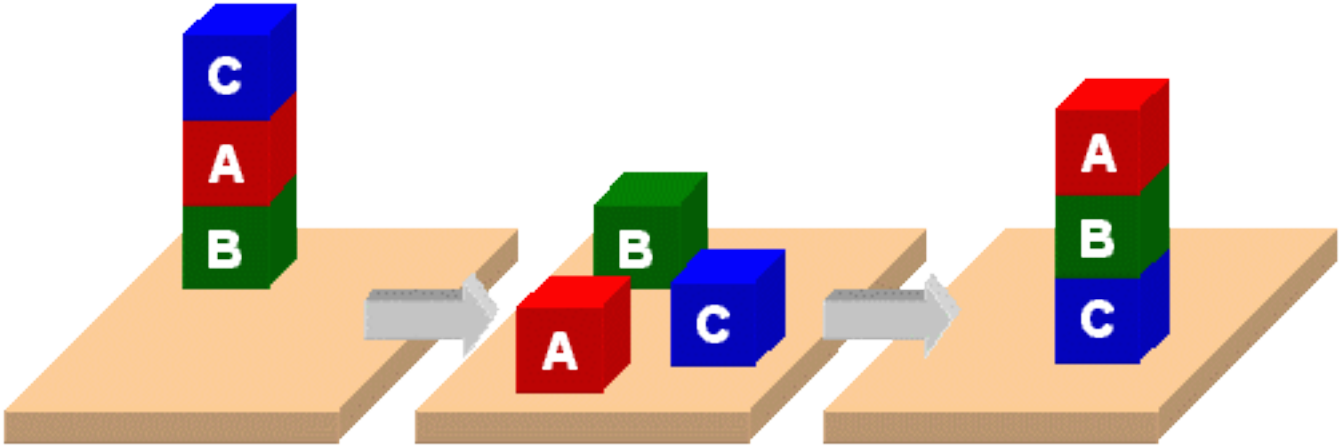
HW3: Planning



PDDL

- Planning Domain Description Language
- Based on STRIPS with various extensions
- Originally defined by Drew McDermott (Yale) and others
- Used in the biennial International Planning Competition (IPC) series
- Many planners use it as a standard input

PDDL Representation

- A task specified via two files: **domain file** and **problem file**
- **Problem file** gives objects, initial state, and goal state
- **Domain file** gives predicates and operators; these may be re-used for different problem files
- **Domain file** corresponds to the transition system, the **problem files** constitute instances in that system

Blocks Word Domain File

```
(define (domain blocksworld)
 (:predicates (clear ?x)
    (holding ?x)
    (on ?x ?y))
 (:action stack
   :parameters (?ob ?underob)
   :precondition (and (clear ?underob) (holding ?ob))
   :effect (and (holding nil) (on ?ob ?underob)
                     (not (clear ?underob)) (not (holding ?ob)))
)
...
```

Domain File (partial)

```
(define (domain prodigy-bw)
 (:requirements :strips)
 (:predicates (on ?x ?y)
    (on-table ?x)
    (clear ?x)
    (arm-empty)
    (holding ?x))
 (:action pick-up
    :parameters (?ob1)
    :precondition (and (clear ?ob1)
        (on-table ?ob1)
        (arm-empty))
    :effect
    (and (not (on-table ?ob1))
           (not (clear ?ob1))
           (not (arm-empty))
           (holding ?ob1)))
...)
```

http://bit.ly/SJPNBw

Problem File

```
(define (problem bw-reversal4)
 (:domain prodigy-bw)
 (:length (:parallel 8) (:serial 8))
 (:objects a b c d)
 (:init (arm-empty)
   (on a b)
   (on b c)
   (on c d)
   (on-table d)
   (clear a))
 (:goal (and
   (on d c)
   (on c b)
   (on b a)
   (on-table a)
   (clear d))))
```

http://bit.ly/SJQdrP

Blackbox planner

- Blackbox planner converts STRIPS-like problems into Boolean satisfiability problems
- Input given in PDDL (domain and problem)
- Solves with a variety of satisfiability engines
- Do *blackbox -help* for options
- Open source and executables for Linux, Mac, Windows
- http://cs.rochester.edu/~kautz/satplan/blackbox/

Blackbox planner

```
bw> ls
README     bw-large-a.pddl         bw-large-d.pddl          bw-sussman.pddl
_notes     bw-large-b.pddl         bw-reversal4.pddl        domain.pddl
bw-12step.pddl         bw-large-c.pddl                      bw-simple.pddl

bw> blackbox -o domain.pddl -f bw-reversal4.pddl
blackbox version 43
...
Begin plan
1 (unstack a b)
2 (put-down a)
3 (unstack b c)
4 (stack b a)
5 (unstack c d)
6 (stack c b)
7 (pick-up d)
8 (stack d c)
End plan
...
bw>
```

Extend the domain: new objects

- Paint sprayers. Each sprayer can only paint in one color (e.g., red, green, blue).
- Paint cans. A paint can holds only only color of paint.
- Brushes. A brush can either be clean or loaded with paint of a particular color.
- Water bucket. A water bucket is used to wash brushes.

Extend the domain: new actions

- painting an object with a sprayer
- painting an object with a brush and can
- loading a paint with paint of a given color
- washing a brush, making it clean

Extend the domain: constraints

- To paint an object, it must be on the table and clear
- Painting with a sprayer: just pick it up and spray
- To paint something a color with a brush, it must be loaded with paint of that color.
- To load a paint bush with a color, you must be holding the brush, brush must be clean and there must be a paint can with that color which is clear. When a brush is loaded with a color it is not clean.
- To wash a brush, making it clean, you have to have a water bucket that has nothing on it (i.e., is clear) and you have to be holding the brush

Problem p1.ppd

```
;; There is only one block, A, which is on the table.  A can with
;; red paint is on the table.  There is a clean brush on the
;; table.  Our goal is to have A be red and the arm empty.

(define (problem 1)
 (:domain hw6)
 (:objects ....  )
 (:init (arm-empty)
  ... block A on the table with nothing on it ...
  ... a red paint can on the table with nothing on it ...
  ... a clean brush is on the table with nothing on it ...
        )
 (:goal (and (arm-empty)
       ... A is red …    )))
```

Problem p6.ppd

```
;; Block A is on the table, B is on A and C on B.  Cans of red, green
;; and blue paint are on the table along with three clean brushes.
;; There is no water bucket.  The goal is to make A red, B green and C
;; blue and to have A on B, B on C and C on the table and the arm
;; empty.

(define (problem 6)
  (:domain hw7)
  (:objects A B C sprayer can1 can2 brush)
  (:init (arm-empty) ... )
  (:goal (and (arm-empty) ...)))
```