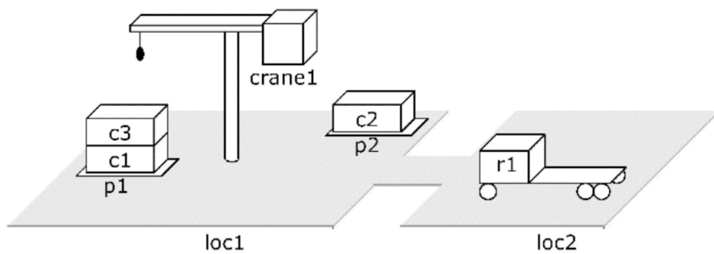
Planning

---

1. (Combinatorics)
   Show that the total number of states for the domain corresponding to the picture below is $8n(n!)$ if there are $n > 0$ containers.



COMP4418 17s2 • Week 13 • Sample Solutions

Planning

(a) (Combinatorics)
There are $n!$ different ways to sort n containers into a specific order $c_1, \ldots, c_n$. Each of these orderings can be configured in the following ways:

- There are $n+1$ different ways to distribute $c_1, \ldots, c_n$ onto `p1` and `p2`:
  - all on `p1`
  - $c_1, \ldots, c_{n-1}$ on `p1` and c_n on `p2`
  - $c_1, \ldots, c_{n-2}$ on `p1` and c_{n-1}, c_n on `p2`
  - ...
  - all on `p2`
- There are n different ways to distribute $c_1, \ldots, c_{n-1}$ onto `p1` and `p2`, with c_n held by the crane.
- There are n different ways to distribute $c_1, \ldots, c_{n-1}$ onto `p1` and `p2`, with c_n loaded onto the cart.
- There are $n-1$ different ways to distribute $c_1, \ldots, c_{n-2}$ onto `p1` and `p2`, with c_{n-1} held by the crane and c_n loaded onto the cart.

Taken together, we obtain $4n(n!)$ configurations. The cart can be at either `loc1` or `loc2`, which results in a total of $8n(n!)$ different states.

COMP4418 17s2 • Week 13 • Sample Solutions to Class Exercises

Planning

---

1. **Blocks World**

   (a) Predicates:

`on(x,y)`	block `x` is on block `y`
`table(x)`	block `x` is on the table
`clear(x)`	block `x` is clear
`holding(x)`	the robot arm is holding block `x`
`handempty`	the robot arm is free

   (b) Operators:

   ```
   unstack(x,y)
      precond:  handempty, clear(x), on(x,y)
      effect:   ¬handempty, holding(x), ¬clear(x), ¬on(x,y), clear(y)

   stack(x,y)
      precond:  holding(x), clear(y)
      effect:   ¬holding(x), handempty, on(x,y), clear(x), ¬clear(y)

   pickup(x)
      precond:  handempty, table(x), clear(x)
      effect:   ¬handempty, holding(x), ¬clear(x), ¬table(x)

   putdown(x)
      precond:  holding(x)
      effect:   ¬holding(x), handempty, clear(x), table(x)
   ```

   (c) Solution plan:
   ⟨putdown(d), unstack(c, a), putdown(c), pickup(b), stack(b, c), pickup(a), stack(a, b)⟩

2. **Variable Assignment Domain**

   (a) A shortest path to a solution (written as sequence of states $[value(a, \_), value(b, \_), value(c, \_)]$:
   $[3, 5, 0] \rightarrow [3, 5, 5] \rightarrow [3, 3, 5] \rightarrow [5, 3, 5]$ (3 actions)

   (b) Without loop-checking, there are infinite paths. With loop-checking, the longest paths have 7 actions, for example,
   $[3, 5, 0] \rightarrow [3, 3, 0] \rightarrow [3, 0, 0] \rightarrow [3, 0, 3] \rightarrow [0, 0, 3] \rightarrow [0, 3, 3] \rightarrow [0, 3, 0] \rightarrow [0, 0, 0]$