Week 5: Modelling in STRIPS and PDDL
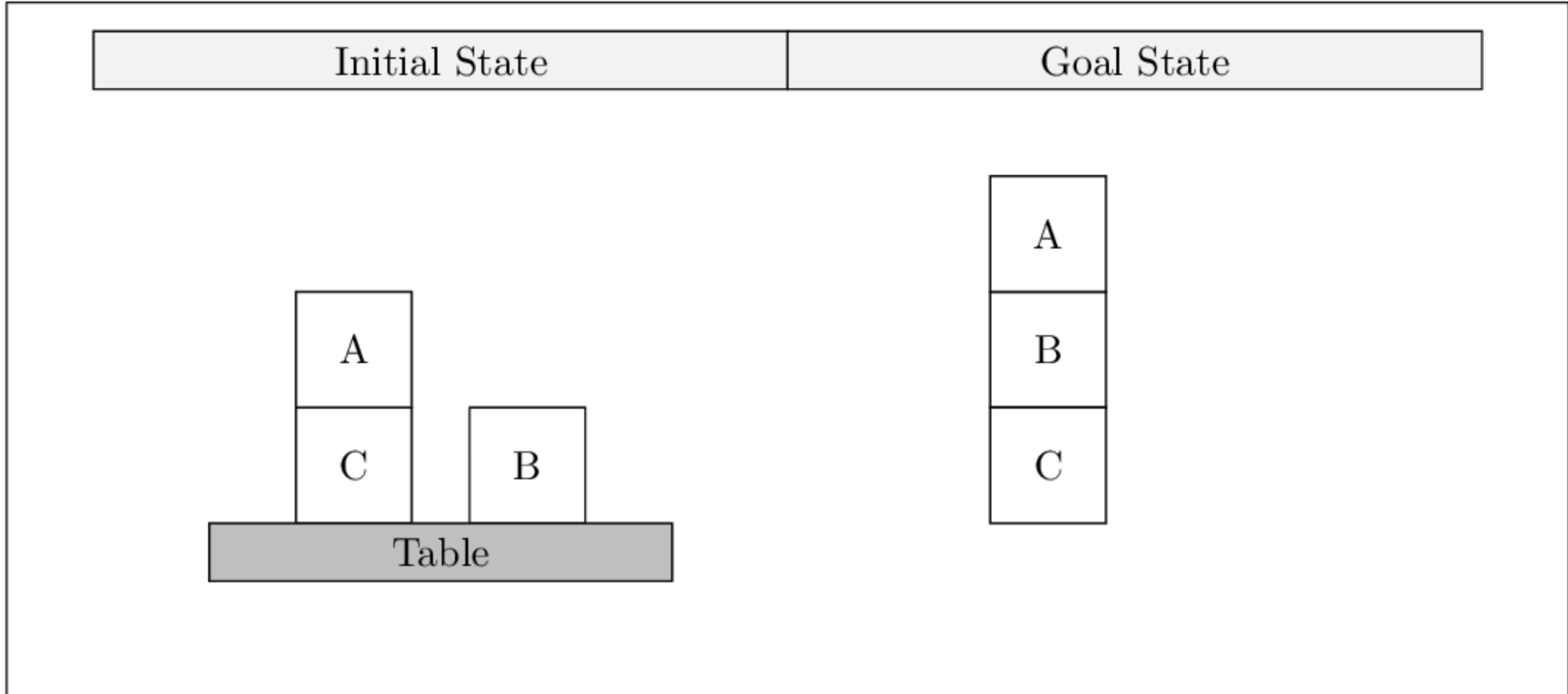
COMP90054 – AI Planning for Autonomy

Thao Le

Key concepts

- How to model a problem in STRIPS
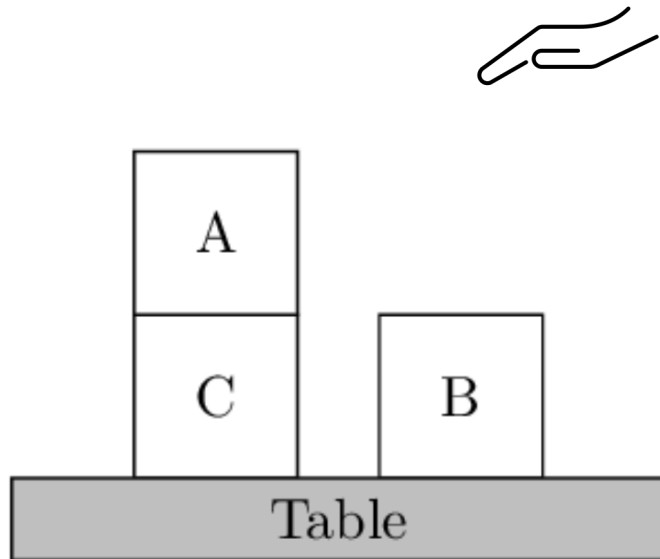- PDDL (Problem Domain Definition Language)

Problem 1: STRIPS

Model Blocks-World as a STRIPS problem $P = \langle F, O, I, G \rangle$. You need to define the set of facts F, the set of operators O, the goal facts G and the initial facts I. You must also define the pre, add, and del functions.

Problem 1

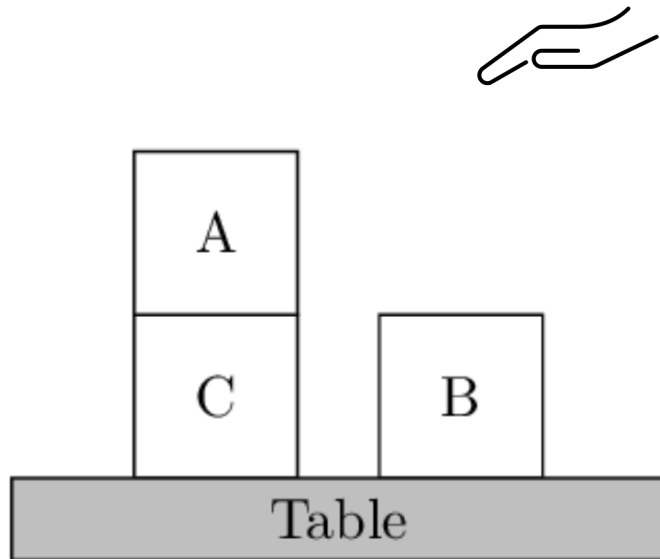P = <F, O, I, G>

F = { on(x, y),
    onTable(x),
    clear(x),
    holding(x),
    handEmpty |
        x, y ∈ $\{A, B, C\}$
}

Problem 1

P = <F, O, I, G>

F = { on(x, y),
        onTable(x),
        clear(x),
        holding(x),
        handEmpty |
            x, y ∈ $\{A, B, C\}$
    }

I = {on(A, C), onTable(C), onTable(B), clear(A), clear(B), handEmpty}

Problem 1

Goal State
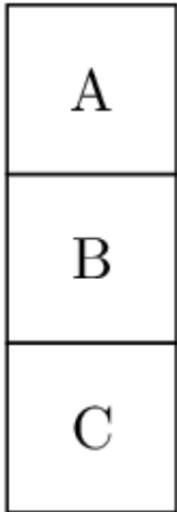
F = { on(x, y),
    onTable(x),
    clear(x),
    holding(x),
    handEmpty |
        x, y $\in \{A, B, C\}$
}

G = {on(A,B), on(B,C)}

Problem 1

Initial State
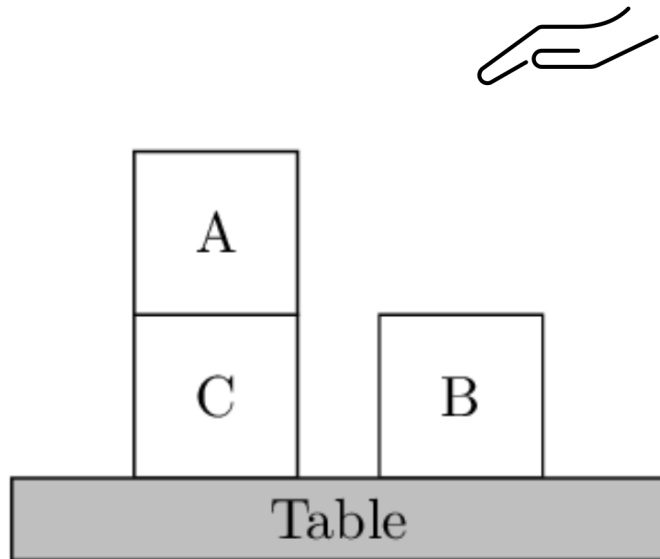
F = { on(x, y), onTable(x), clear(x), holding(x), armFree |
x, y ∈ {A, B, C}}

Operator o ∈ O represented by
1. the **Add** list Add(o) ⊆ F
2. the **Delete** list Del(o) ⊆ F
3. the **Precondition** list Pre(o) ⊆ F

Define Operators

1. Pick up a block from the table **pickup(x)**
2. Pick up a block from another block **unstack(x, y)**
3. Put down a block on the table **putdown(x)**
4. Put down a block on another block **stack(x, y)**
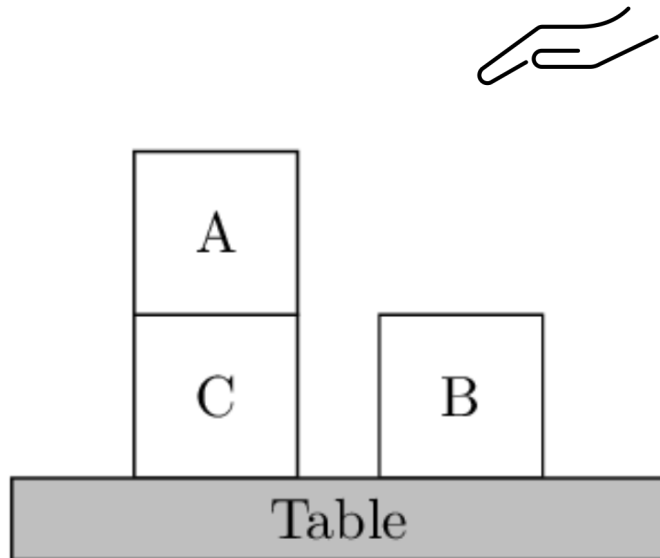
Thao Le

Problem 1

Initial State

P = <F, O, I, G>

F = { on(x, y), onTable(x), clear(x), holding(x), handEmpty | x, y ∈ {A, B, C}}

Operator o ∈ O represented by
1. the **Add** list Add(o) ⊆ F
2. the **Delete** list Del(o) ⊆ F
3. the **Precondition** list Pre(o) ⊆ F

Define Operators

1. Pick up a block from the table **pickup(x)**

- Prec: onTable(x), clear(x), handEmpty
- Add: holding(x)
- Del: onTable(x), clear(x), handEmpty

Problem 1

P = <F, O, I, G>
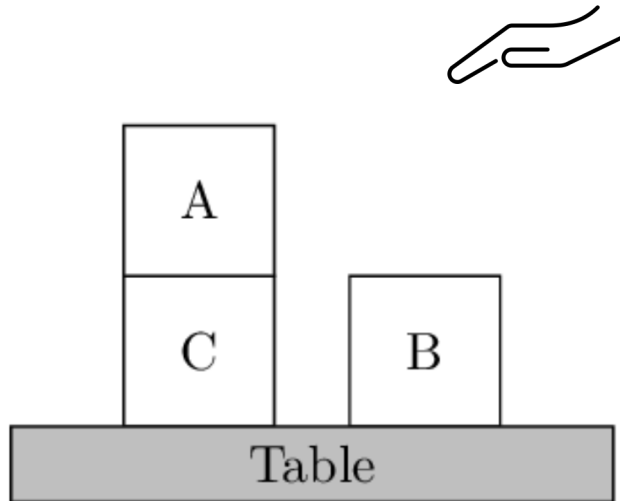
F = { on(x, y), onTable(x), clear(x), holding(x), handEmpty | x, y ∈ $\{A, B, C\}$}}

Operator o ∈ O represented by
1. the **Add** list Add(o) ⊆ F
2. the **Delete** list Del(o) ⊆ F
3. the **Precondition** list Pre(o) ⊆ F

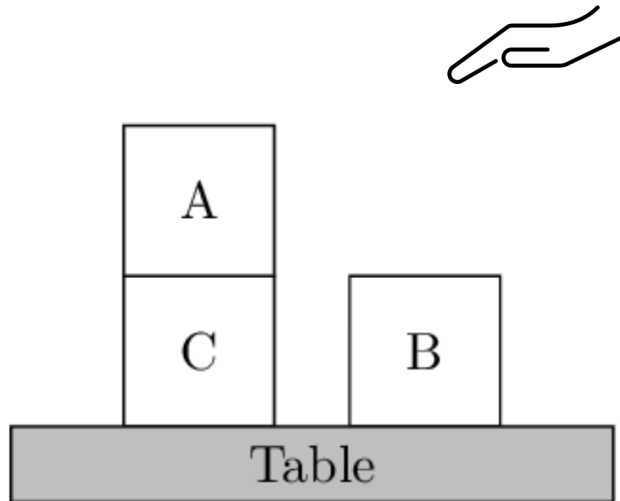**Define Operators**

2. Pick up a block from another block **unstack(x,y)**

- Prec: on(x, y), clear(x), handEmpty
- Add: holding(x), clear(y)
- Del: on(x, y), clear(x), handEmpty

Problem 1

P = <F, O, I, G>

F = { on(x, y), onTable(x), clear(x), holding(x), handEmpty | x, y ∈ $\{A, B, C\}$}}

Operator o ∈ O represented by
1. the **Add** list Add(o) ⊆ F
2. the **Delete** list Del(o) ⊆ F
3. the **Precondition** list Pre(o) ⊆ F

Define Operators

3. Put down a block on the table **putdown(x)**

- Prec: holding(x)
- Add: clear(x), onTable(x), handEmpty
- Del: holding(x)

Problem 1
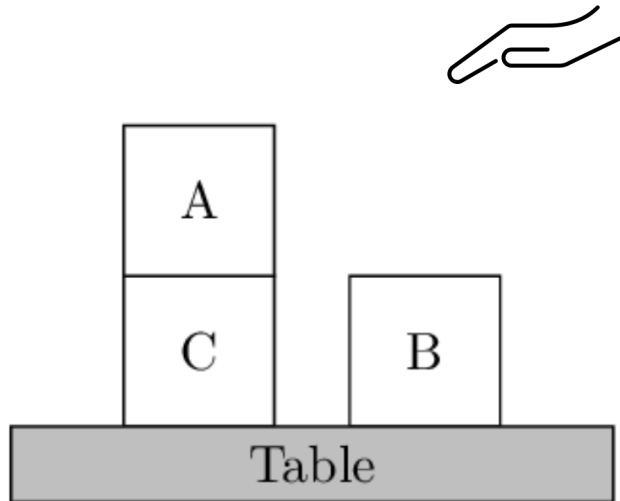
Initial State

P = <F, O, I, G>

F = { on(x, y), onTable(x), clear(x), holding(x), handEmpty |
x, y ∈ {A, B, C}}

Operator o ∈ O represented by
1. the **Add** list Add(o) ⊆ F
2. the **Delete** list Del(o) ⊆ F
3. the **Precondition** list Pre(o) ⊆ F

Define Operators

4. Put down a block on another block **stack(x, y)**

- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handEmpty
- Del: clear(y), holding(x)

Problem 1

Initial State
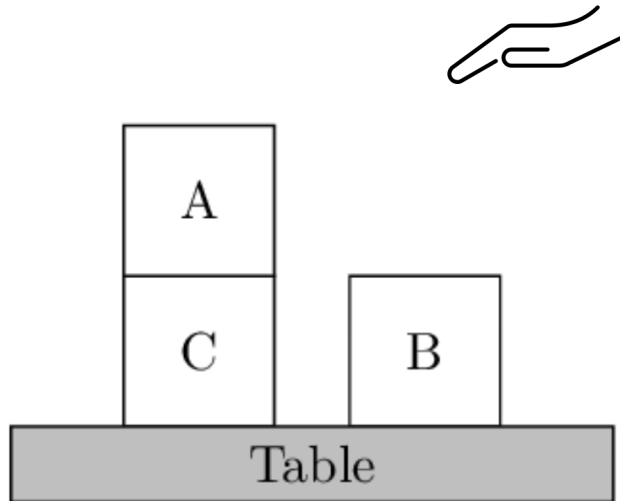
Define Operators

O = {
pickup(x)
- Prec: onTable(x), clear(x), handEmpty
- Add: holding(x)
- Del: onTable(x), clear(x), handEmpty

unstack(x, y)
- Prec: on(x, y), clear(x), handEmpty
- Add: holding(x), clear(y)
- Del: on(x, y), clear(x), handEmpty

putdown(x)
- Prec: holding(x)
- Add: clear(x), onTable(x), handEmpty
- Del: holding(x)

stack(x, y)
- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handEmpty
- Del: clear(y), holding(x)
}

Thao Le

Problem 2: PDDL

1. A domain file that has predicates and actions
2. A problem file that has objects, initial state and goal

TSP: http://editor.planning.domains/#read_session=VpDqyltSHg

Problem 2: PDDL

```
1  (define (domain tsp)
2      (:requirements :typing)
3      (:types node)
4
5      ;; Define the facts in the problem
6      ;; "?" denotes a variable, "-" a type
7      (:predicates
8          (move ?from ?to - node)
9          (at ?pos - node)
10         (connected ?start ?end - node)
11         (visited ?end - node)
12     )
13
14     ;; Define the action(s)
15     (:action move
16         :parameters (?start ?end - node)
17         :precondition (and
18             (at ?start)
19             (connected ?start ?end)
20         )
21         :effect (and
22             (at ?end)
23             (visited ?end)
24             (not (at ?start))
25         )
26     )
27 )
```

F (lines 7–11)
O (lines 15–25)

```
1  (define (problem tsp-01)
2      (:domain tsp)
3      (:objects Sydney Adelade Brisbane Perth Darwin - node)
4
5      ;; Define the initial situation
6      (:init  (connected Sydney Brisbane)
7              (connected Brisbane Sydney)
8              (connected Adelade Sydney)
9              (connected Sydney Adelade)
10             (connected Adelade Perth)
11             (connected Perth Adelade)
12             (connected Adelade Darwin)
13             (connected Darwin Adelade)
14             (at Sydney)
15     )
16     (:goal
17         (and
18             (at Sydney)
19             (visited Sydney)
20             (visited Adelade)
21             (visited Brisbane)
22             (visited Perth)
23             (visited Darwin)
24         )
25     )
26 )
```

I (lines 6–13)
G (lines 16–24)

Problem 2

F = { on(x, y),
    onTable(x),
    clear(x),
    holding(x),
    handEmpty | x, y $\in \{A, B, C\}$
}

I = {on(A, C), onTable(C), onTable(B), clear(A), clear(B), handEmpty}

G = {on(A,B), on(B,C)}

O = {

pickup(x)
- Prec: onTable(x), clear(x), handEmpty
- Add: holding(x)
- Del: onTable(x), clear(x), handEmpty

unstack(x, y)
- Prec: on(x, y), clear(x), handEmpty
- Add: holding(x), clear(y)
- Del: on(x, y), clear(x), handEmpty

putdown(x)
- Prec: holding(x)
- Add: clear(x), onTable(x), handEmpty
- Del: holding(x)

stack(x, y)
- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handEmpty
- Del: clear(y), holding(x)
}

Thao Le

Planimation Plugin

https://github.com/planimation/documentation

Solution with 4 actions

http://editor.planning.domains/#read_session=iOEg2OeV24