Tutorial 1

Week 3

TA: *Tas*bolat Taunyazov
CSxx46
AY23-24, Sem 1

Logistics

- Contacts:
  - Use canvas forums to post questions
  - TA email: tasbolat@comp.nus.edu.sg
  - Canvas message
- Participation:
  - Attendance (graded!)
  - Ask and discuss questions

Recap

Observability

- **Fully:** agent *has complete* information about the current state of the environment.
- **Partially:** agent *does not have complete* information about the current state of the environment.

Agents

- **Single:** *one agent* interacts with the environment.
- **Multi + cooperative:** agents interact with each other to achieve *aligned* goals.
- **Multi + competitive:** agents interact with each other to achieve *mismatched* goals.

Recap

Stochasticity

- **Deterministic:** the next state of the environment is solely determined by the current state of the environment and the actions selected by the agents.
- **Stochastic:** the agent will not always enter the same state after taking an action from a starting state.

Episodic vs Sequential

- **Episodic:** every action depends only on the current state.
- **Sequential:** information about previous states and actions is needed to model the environment properly.

Problem 1: Autonomous taxi service

Question: Comment on the observability aspect of the environment.

Partially Observable

Fully Observable

Problem 1: Autonomous taxi service

Question: Comment on the observability aspect of the environment.

Partially Observable | Fully Observable

Taxi does not have complete information of the environment. E.g: traffic conditions, pedestrians and so on

Problem 1: Autonomous taxi service

Question: Is the environment a single-agents, collaborative multi-agent or competitive multi-agent environment?

Single agent | Multi + Collaborative | Multi + Competitive

Problem 1: Autonomous taxi service

Question: Is the environment a single-agents, collaborative multi-agent or competitive multi-agent environment?

Single agent	Multi + Collaborative	Multi + Competitive
	On the road cars cooperate	*Taxi companies compete to pick more passengers*

Problem 1: Autonomous taxi service

Question: Will the environment be deterministic or stochastic?

Deterministic

Stochastic

Problem 1: Autonomous taxi service

Question: Will the environment be deterministic or stochastic?

Deterministic

Stochastic

Examples:

- *Pedestrians appearance on the road*
- *Rainy weather makes road slippery -> chance of accidents*
- *Traffic breakage*

Problem 1: Autonomous taxi service

Question: Would you model the environment to be episodic or sequential?

Episodic

Sequential

Problem 1: Autonomous taxi service

Question: Would you model the environment to be episodic or sequential?

Episodic Sequential

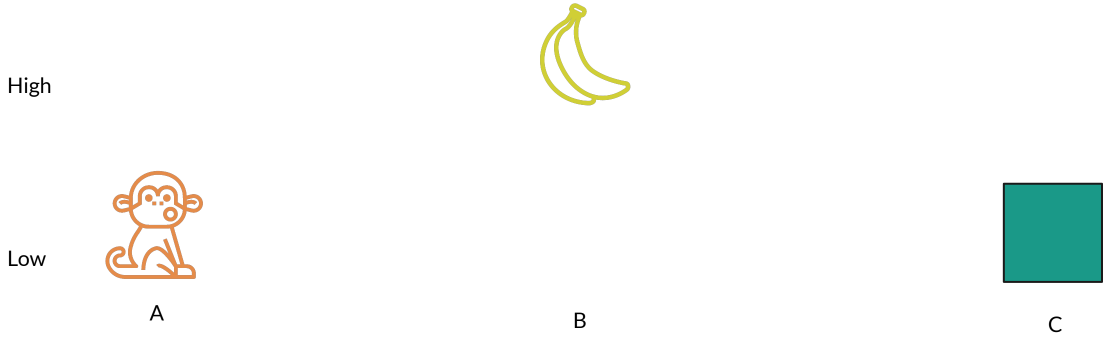
Example: temporal information about the past state of the traffic light might affect if taxi accelerates or decelerate.

Recap: PDDL and STRIPS

PDDL

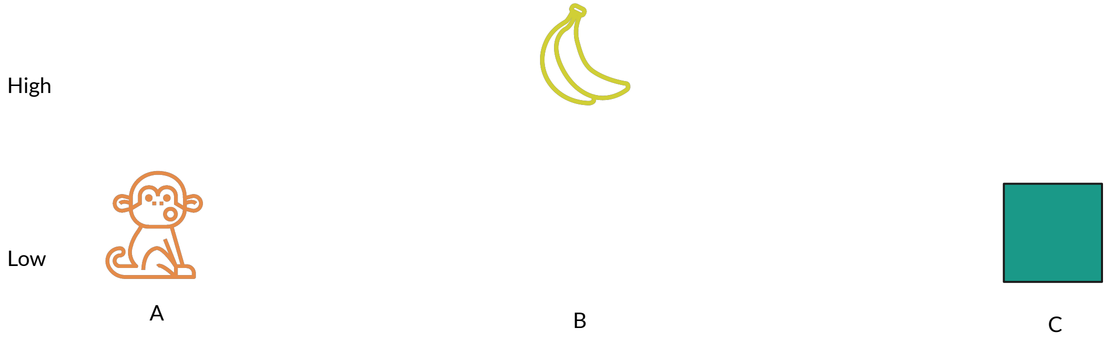
- Goal states are specified as conjunctions of **positive** literals
- Actions are specified as **pre-conditions** and **effects**
- Anything left unspecified in a state is assumed **false**

Problem 2



Write down the initial state description.

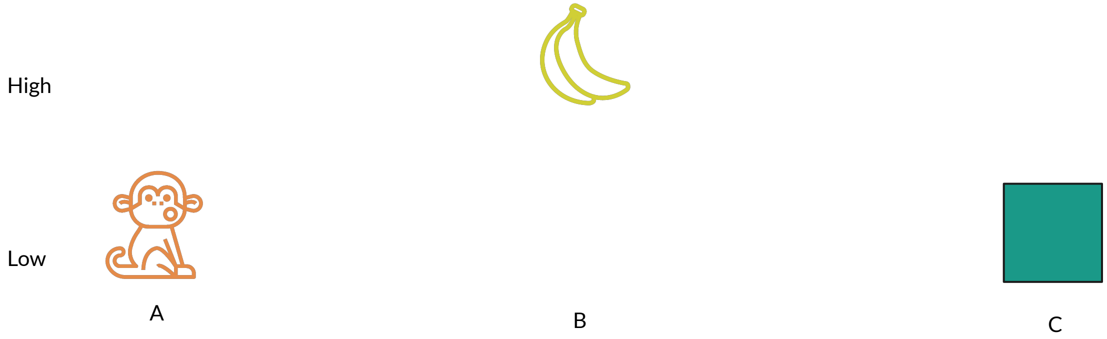
Problem 2



Write down the initial state description.

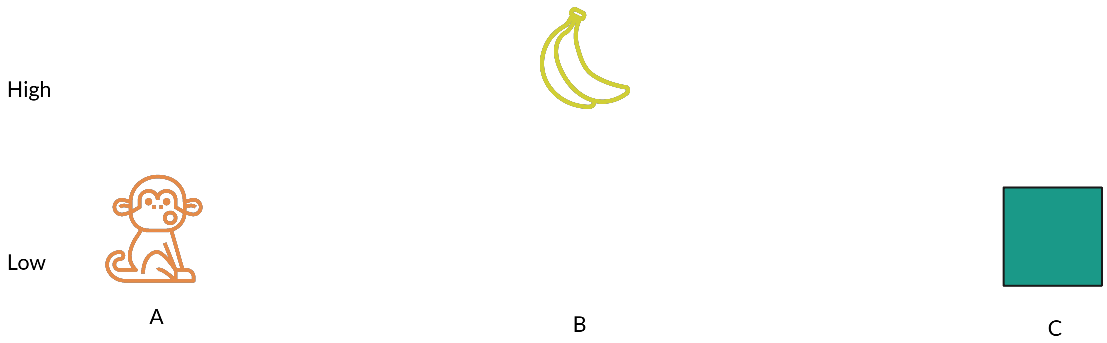
At(Monkey, A) $\wedge$ At(Bananas, B) $\wedge$ At(Box, C) $\wedge$ Height(Monkey, Low) $\wedge$ Height(Bananas, High) $\wedge$ Height(Box, Low) $\wedge$ Pushable(Box) $\wedge$ Climbable(Box) $\wedge$ Graspable(Bananas)

Problem 2



Write the six action schemas.

Problem 2

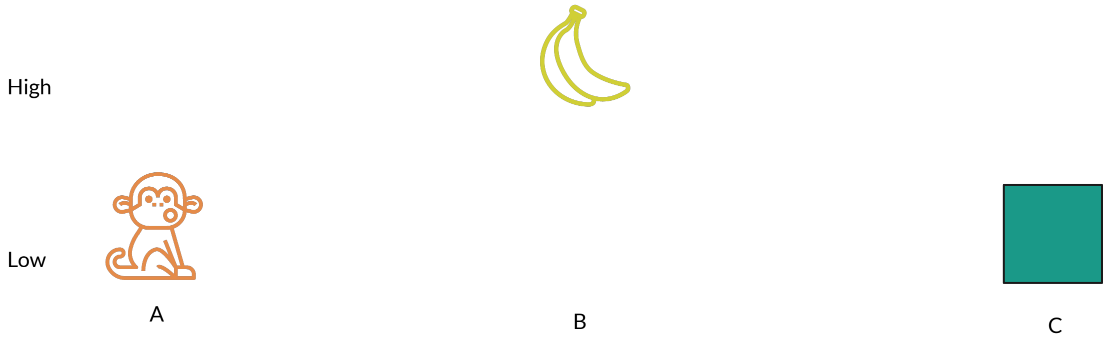


Write the six action schemas.

Action(Action Go(x, y),
Precond: At(Monkey, x),
Effect: At(Monkey, y) ∧ ¬At(Monkey, x))

Action(Action Push(b, x, y),
Precond: At(b, x) ∧ At(Monkey, x) ∧ Pushable(b) ∧ Height(Monkey, Low) ∧ Height(b, Low)
Effect: At(b, y) ∧ At(Monkey, y) ∧ ¬At(b, x) ∧ ¬At(Monkey, x)

Problem 2

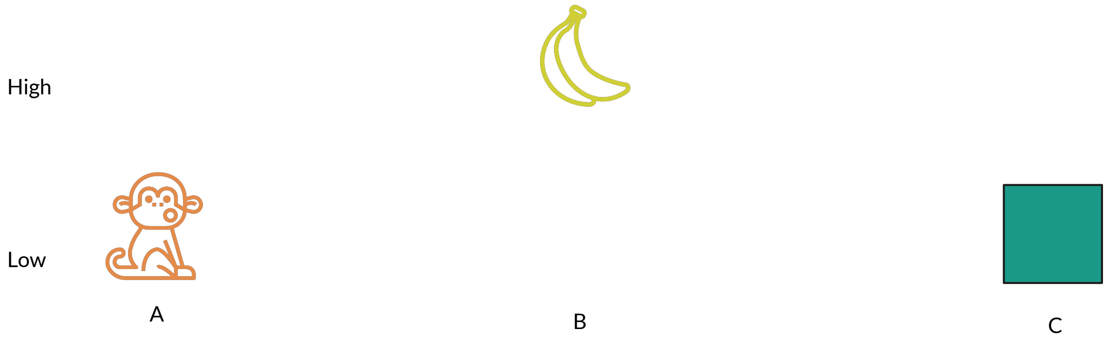


Write the six action schemas.

Action(Action ClimbUp(b, x),
Precond: At(Monkey, x) ∧ At(b, x) ∧ Climbable(b) ∧ Height(Monkey, Low) ∧ Height(b, Low)
Effect: On(Monkey, b) ∧ Height(Monkey, High) ∧ ¬Height(Monkey, Low))

Action(Action ClimbDown(b, x),
Precond: On(Monkey, b)
Effect: Height(Monkey, Low) ∧ ¬Height(Monkey, High) ∧ ¬On(Monkey, b)

Problem 2



Write the six action schemas.

Action(Action Grasp(a, x, h),

Precond: At(Monkey, x) ∧ At(a, x) ∧ Graspable(a) ∧ Height(Monkey, h) ∧ Height(a, h)

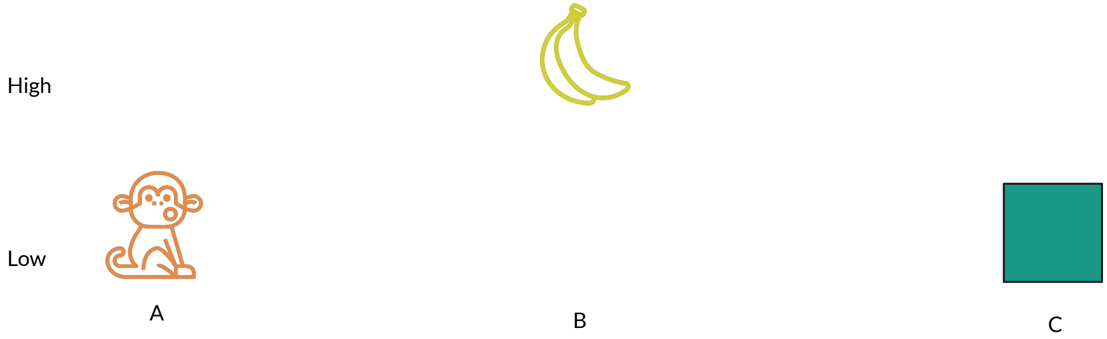
Effect: Have(Monkey, a) ∧ ¬At(a, x) ∧ ¬Height(a, h))

Action(Action UnGrasp(a, x, h),

Precond: Have(Monkey, a) ∧ At(Monkey, x) ∧ Height(Monkey, h)

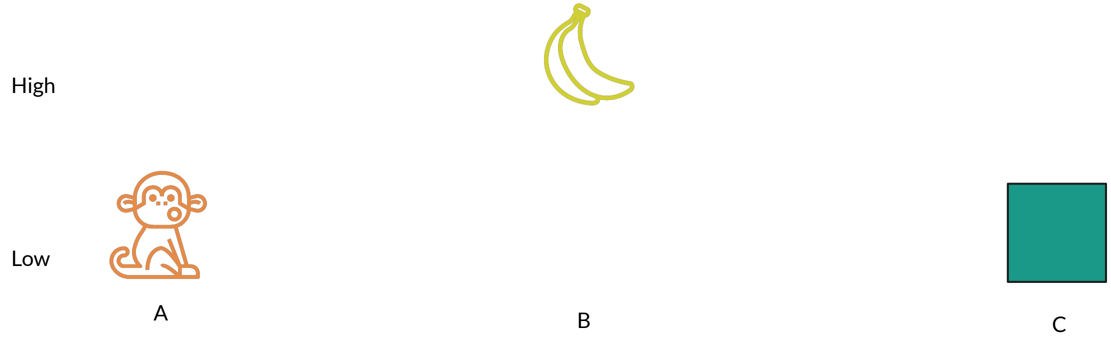
Effect: ¬Have(Monkey, a) ∧ At(a, x) ∧ Height(a, h)

Problem 2



- New goal: Get the bananas ***and*** put the box back at C
- Can we do this with PDDL or another STRIPS-style system?

Problem 2

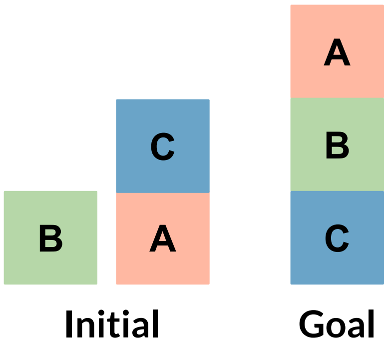


- New goal: Get the bananas ***and*** put the box back at C
- Can we do this with PDDL or another STRIPS-style system?

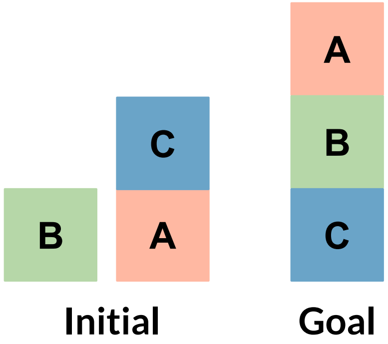
No! STRIPS doesn't allow relations in goal states, everything must be fully evaluated

Problem 3

- **Question:** What is a formal definition for the problem?
  - Only one block can moved at a time.



Problem 3



- **Question:** What is a formal definition for the problem?
  - Only one block can moved at a time.

Initial State: On(B, Table) ∧ On(C, A) ∧ On(A, Table) ∧ Clear(B) ∧ Clear(C)

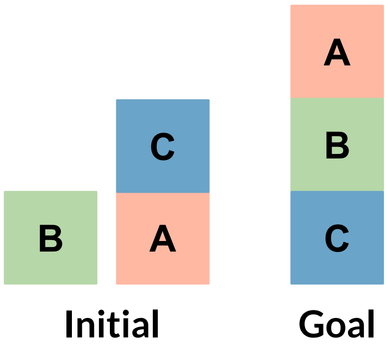
Goal: On(C, Table) ∧ On(B, C) ∧ On(A, B)

Action(Action Move(oj, from, to),
  Precond: Clear(obj) ∧ Clear(to),
  Effect: On(obj,to) ∧ Cleabr(from) ∧ ¬ Clear(to))

Action(Action MovetoTable(oj, from),
  Precond: Clear(obj),
  Effect: On(obj,Table) ∧ Cleabr(from)

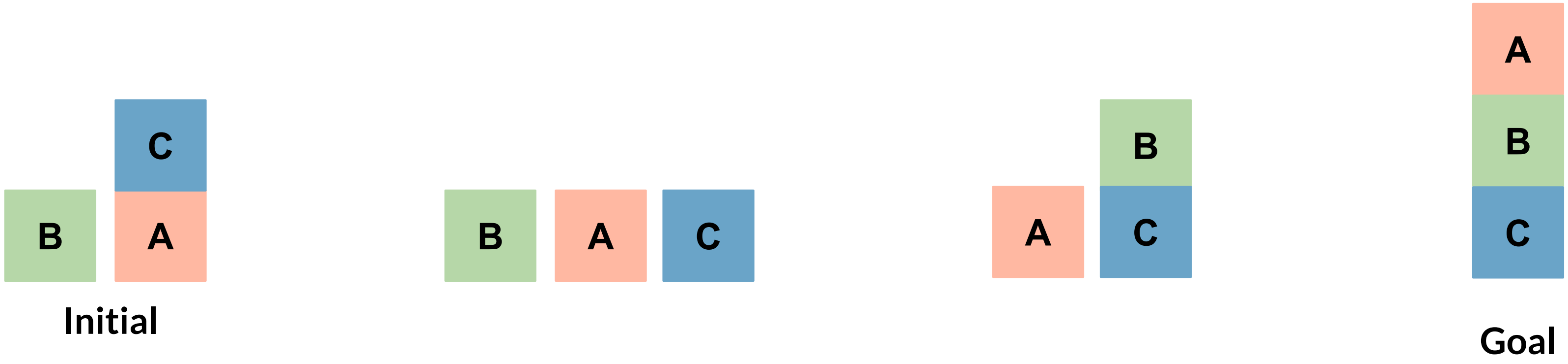
Problem 3

Solve the problem, either by hand or with a planning program



Problem 3

Solve the problem, either by hand or with a planning program

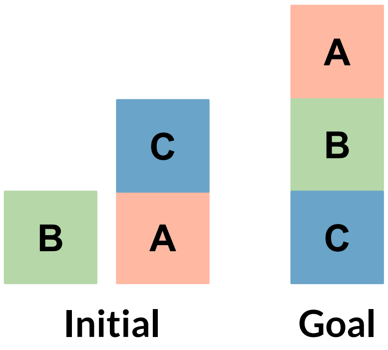


1. MoveToTable(C, A)

2. Move(B, Table, C)

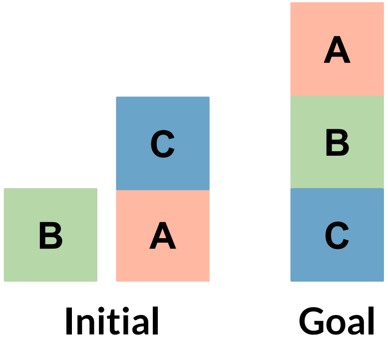
3. Move(A, Table, B)

Problem 3



- We can only move one block at a time
- Why can't non-interleaved planners solve this?

Problem 3



- We can only move one block at a time
- Why can't non-interleaved planners solve this?

Our goal has multiple subgoals. To search for the best solution, we have to consider solutions that undo at least one subgoal along the way. Only interleaved planners allow this.