AI - CT3 - Notes

Planning: Task of coming up with a sequence of actions that will achieve a goal is called planning.

Planning Environment

- Classical
  - Fully observable, Deterministic, Finite, static, discrete
- Non-classical
  - Partially observable, stochastic and agent based.

Difficulty in PSA

→ perform irrelevant actions

→ finding heuristic function (lack autonomy).

→ No-problem decomposition.

> A planning agent overcomes all problems by representing the goals as conjunction of goals.

Planning problem

The procedure to from next state to goal state.

It involves two things, how and when.

Planning problem
- → domain model
- → initial state
- → ~~fin~~ goal state (next state)

Domain model : defines actions along with objects. Necessary to specify operators too. Information about actions and state constrains while acting should be given

Initial state : State where any action is yet to take place

Final state : The state where the plan is intended to achieve at the end of execution.

Planning problem

- Finds sequence of actions - achieves a given goal from a given initial world state.
  - set of operator descriptions.
  - an initial state description.
  - goal state description or predicate.
- compute a plan
  - sequence of operator instances
  - execution of them to change states
- Goals → specified as a conjunction of subgoals

planning agent → constructs plans to achieve goals then executes them.

- Sub-goals → planned independently, it reduces problem complexity (kind of divide and conquer).

	Prob. Sol.	Planning
States →	data structure	logical sentences
Actions →	code	preconditions / outcomes
Goals →	code	logical sentences
Plans →	sequences from s_0	constraints on actions.

[Problem solving agents + knowledge base = Planning agents]

Problem solving agents ↓ plan the consequences

knowledge base ↓ select actions based on explicit logical represent.

Planning problem using state-space search approach

initial state → initial solution

goal-test predicate → goal state description

successor function → ~~successo~~ computed from set of ops.

once goal found → solution plan is the sequence of ops in path from start node to goal node.

Algorithm of simple planning agent

→ generate a goal to achieve

→ construct a plan to achieve goal from current state

→ execute plan until finished

→ begin again with new goal.

Goal → plan → current state → goal state → if found then end

else

Key ideas behind planning

i) to open up representation of state, goals and operators so that a reasoner can more intelligently select actions when needed.

ii) planner is free to add actions to plan whenever needed rather than in an incremental fashion.

iii) each part of problem world is independent of most other parts which can be solved by divide-and-conquer.

Planning languages

planning languages must represent:

→ states
→ goals
→ actions.

languages must be
→ expressive for ease of representation
→ flexible for manipulation

State representation

Represented as a conjunction of positive literals.

→ using logical propositions (poor ∧ unknown)
→ FOL literals: At(plan1, OMA) ∧ At(Plan2, JFK)
→ FOL literals must be ground & function free.
→ Closed world assumption (what is not stated are assumed false)

Goal → partially satisfied state

Action representation

Action schema → name, pre-conditions, effects

Languages

STRIPS

[Standford Research Institute Problem Solver]

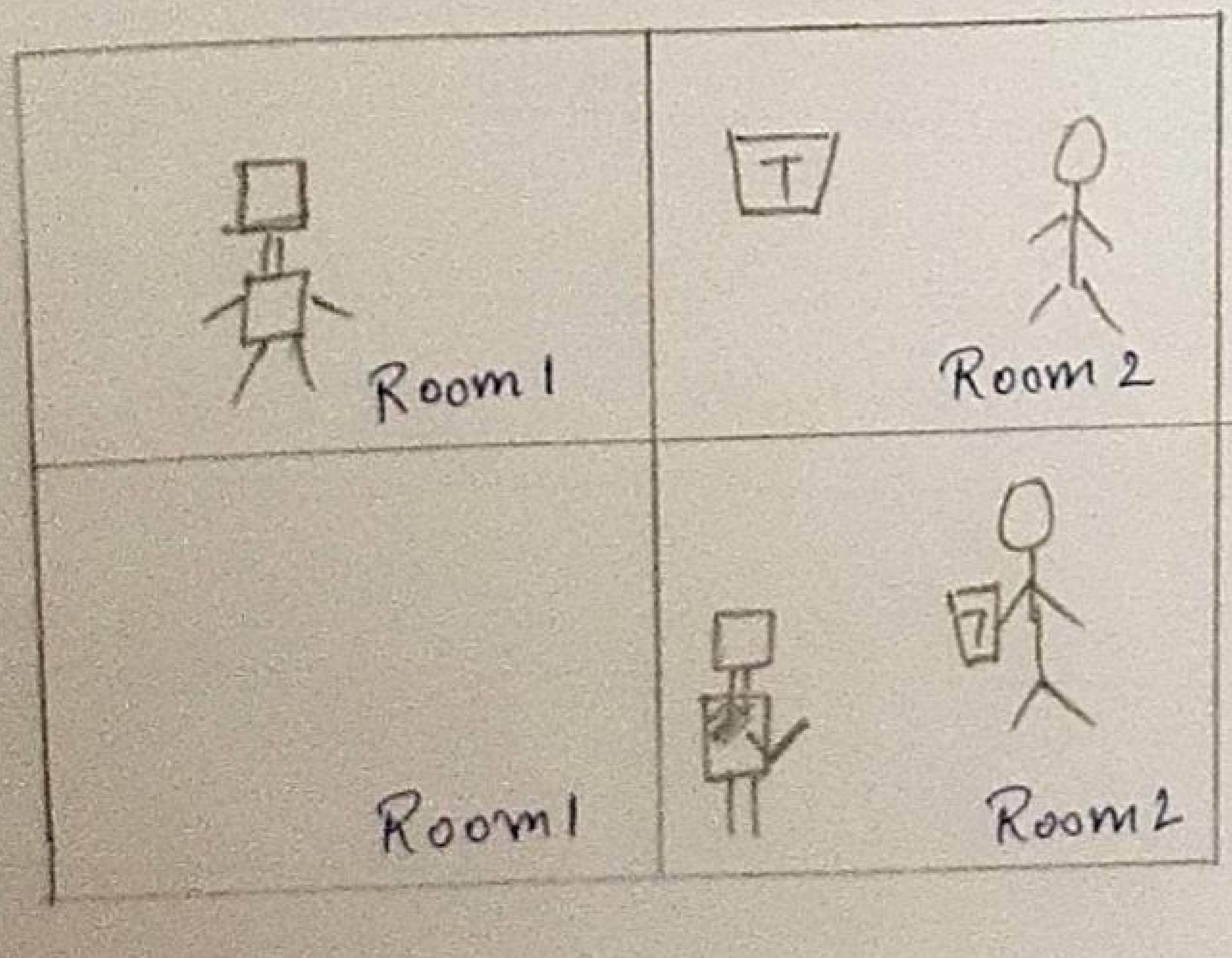
ADL → [Action Description Lang]

PDDL → [planning domain definition language]

STRIPS

↓ uses first-order predicate.
allow function-free literal.

ROBOT - TEA - GUEST problem



State representation

1. in (robot, room1) ^
   in (tea, room 2) ^
   in (guest, room 1)

Action representation

Action 1: move-to-room2

precondition: in(robot, room 1)

post-condition: add-List (robot, room2)

delete-list : in(robot, room 1)

Block World

There are 'N' number of blocks resting on table with specific sequence.

Goal : Arrange in desired sequence.

Available moves : put block on table
put block on another block top.

State : represented using sequence of blocks in current position.

STRIPS

→ initial state S
→ goal state G
→ set of STRIPS actions

[precondition must be true, effect depends on actions]

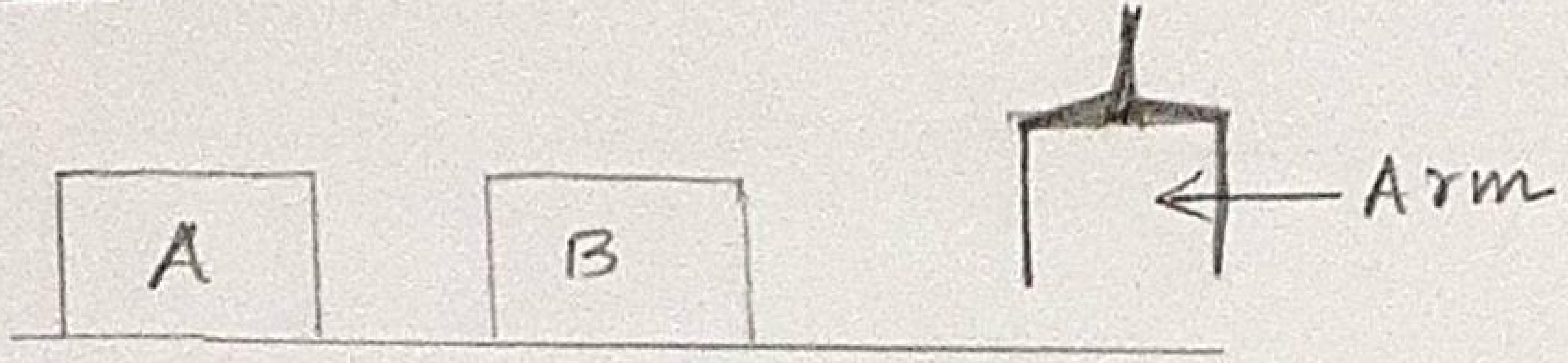
Block World Problem

Action list :

S. No.	Action	Precondition	Effect
1.	Pickup(x)	Arm empty On (x, Table) clear(x)	Holding(x)
2.	Putdown (x)	Holding (x)	Arm Empty On(x, Table) Clear(x)
3.	Stack (x, y)	Holding (x) Clear (y)	On (x, y) Clear(x) Arm Empty

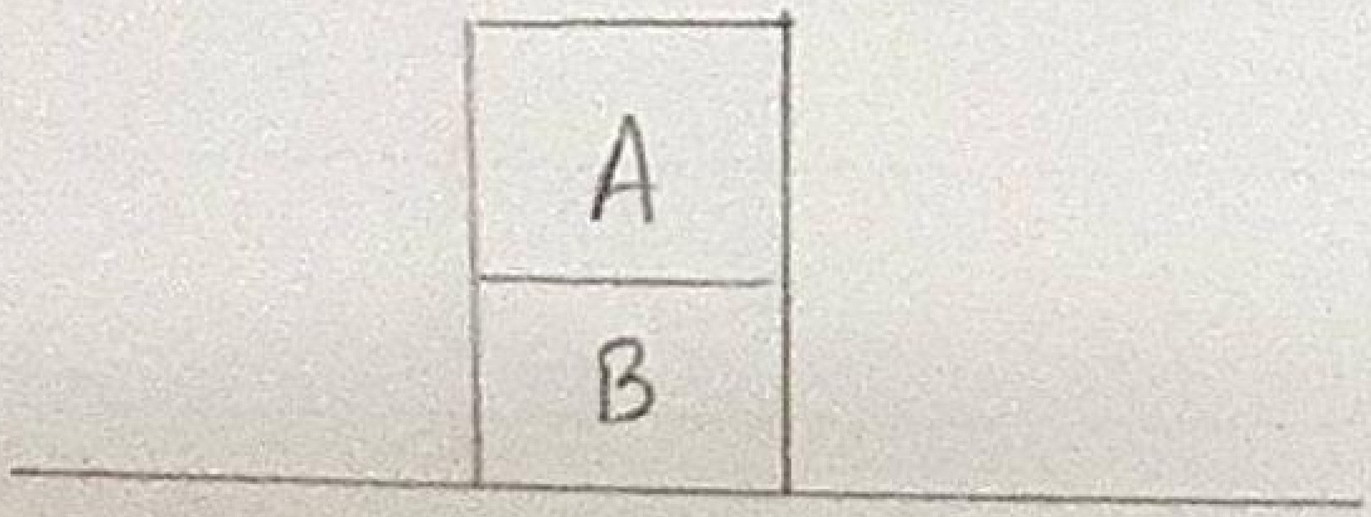
4	Unstack (x,y)	On (x, y) Clear (x) Arm Empty	Holding (x) Clear (y)

Start state :



On (A, Table)
On (B, Table)

Goal state :



On (A, B)

Solution : Stack (A, B)

Preconditions :

× Holding (A) → Pickup (A)
✓ Clear (B)

Goal stack planning → Handle interactive compound goals.

operators → ADD, DELETE, PREREQUISITES,

database maintaining current situation for each operator

Means- Ends Analysis

- → search strategies reason forward or backward.
- → mixed strategy - solve major part of problem first and smaller parts later.
- → limits searching process.
- → centres around finding difference between current state and goal state.
- → Means- ends can be applied recursively for a problem. It is a strategy to control search in problem- solving.

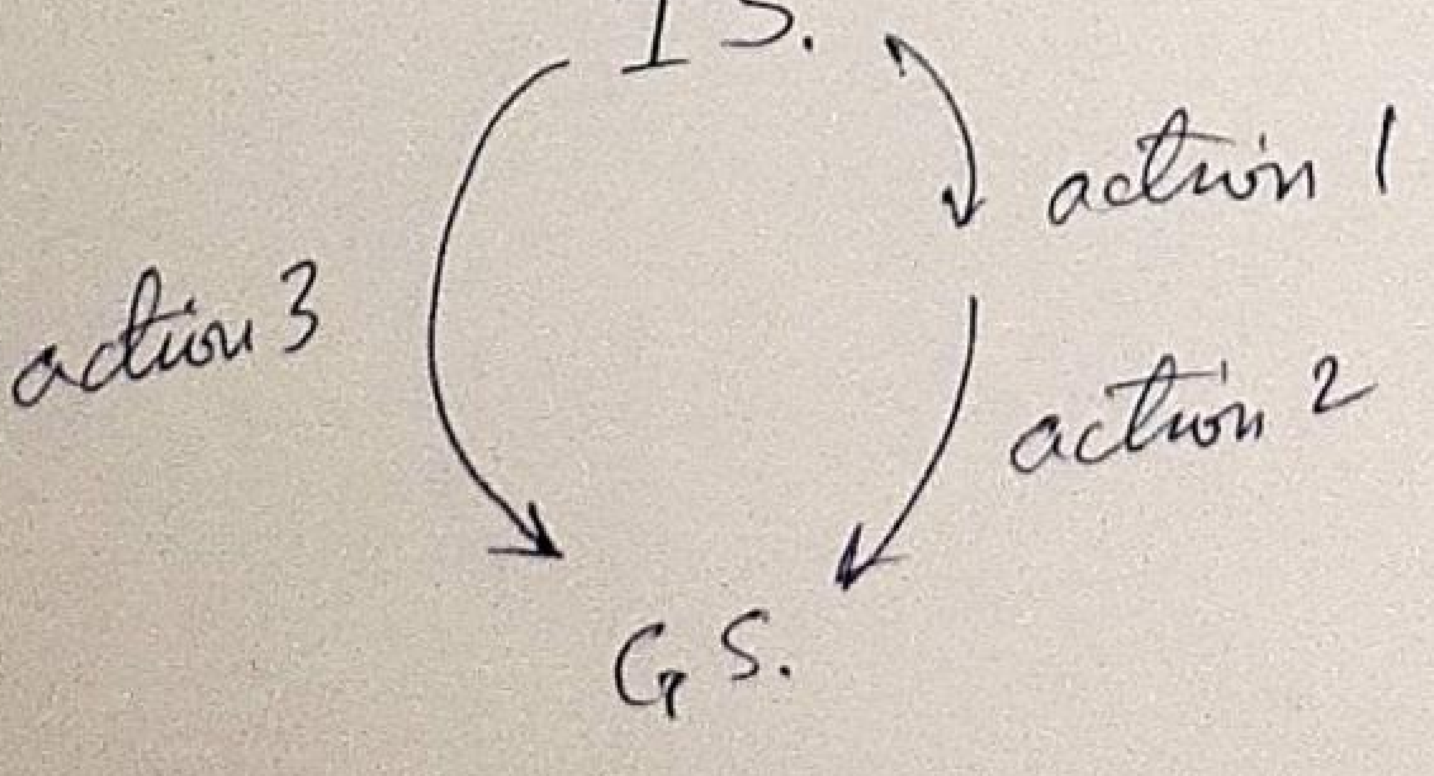
Steps in MEA:

- → First evaluate diff. between initial and final state
- → Select various operators which can be applied for each difference.
- → Apply the operator at each difference, which reduces the difference between the current state and goal state.

Solution:
Move.
Delete
Expand

Non- linear planning

plan → subproblems → solved simultanal
→ non- linear plan.



→ op addition
→ ordering them
→ binding the vars with ops