GROUP MEMBERS

① UTKARSH THUSOO (M20AIE318)

② SURESH BOJJAM (M20AIE313)

Assignment
Artificial Intelligence - 2
AY 2021-22, Semester - I

Executive MTech in AI

Instructions:

1. This is a handwritten assignment. Scan and submit in the Google Classroom.
2. You are free to work in a groups of three. Write down names and roll numbers and who has done which problem in the top of the page. However, good understanding of all the problems is expected and highly encouraged. Working in group is allowed only to encourage peer learning.
3. Maximum points: 40.

Important Dates:

1. Release of assignment: Nov 5, 2021
2. Group Formation: Nov 7, 2021
3. Early Bird Submission: Nov 18, 2021: Midnight (full points)
4. Late Submission: Nov 30, 2021: Midnight (10% penalty in each day late)

Problem 1: A robot moves into rooms R_1 and R_2 and switch the bulbs B_1 and B_2 on/off. The following are the action schema:

1. goto(r, x1, x2) : robot r go to x2 from x1
2. switchON(s): switchON the bulb s
3. switchOFF(s): switchOFF the bulb s

1. Write down preconditions and effects of the above actions. 2. Consider the following: (i) Initial state: $< R_1, \bar{R_2}, \bar{B_1}, \bar{B_2} >$: Robot is at Room R_1 not in Room R_2 and both bulbs are off.

(ii) Goal state: $< R_2, B_1, B_2 >$: Robot is at Room R_2 and both bulbs are ON.

Draw state space diagram for the above by drawing to all possible states.

Problem 2: What is Sussman anomaly. Give example and discuss.

Problem 3: The blocks world is one of the most famous planning domains in artificial intelligence. Write down action schema and preconditions and effects of the actions in the block world problem.

Problem 4: Draw the graph-plan graph for a depth-two plan given the following action descriptions. Starting state is: not have-keys, not open, not painted. Goal state is: open, painted. Show all mutexes.

- Get Keys: (preconditions:) (Effect: have-keys)
- Open Door: (Preconditions: not-open, have keys) (Effect: Open)
- Paint Door: (Preconditions: not painted) (Effect: painted)

Problem 5: Consider the following domain:

- $Action_1$: Effect: C, A
- $Action_2$: Effect: D, $\neg A$

The goal is $C \wedge D$. What solution is returned by partial order plan? What solution is returned by Graph Plan? (If there are multiple answers, give them.) What does this example reveal about the expressive power of the solution descriptions in the two algorithms?

Problem 6: What is situation calculus. Explain in detail. Show examples which were not discussed in the class.

Problem 1

1) Preconditions & effects are as below:

i) goto (r, R1) : robot r go to room R1

ii) switch ON(B1) : the bulb B1 will be switched ON

iii) Switch OFF(B1) : the bulb B1 will be switched OFF

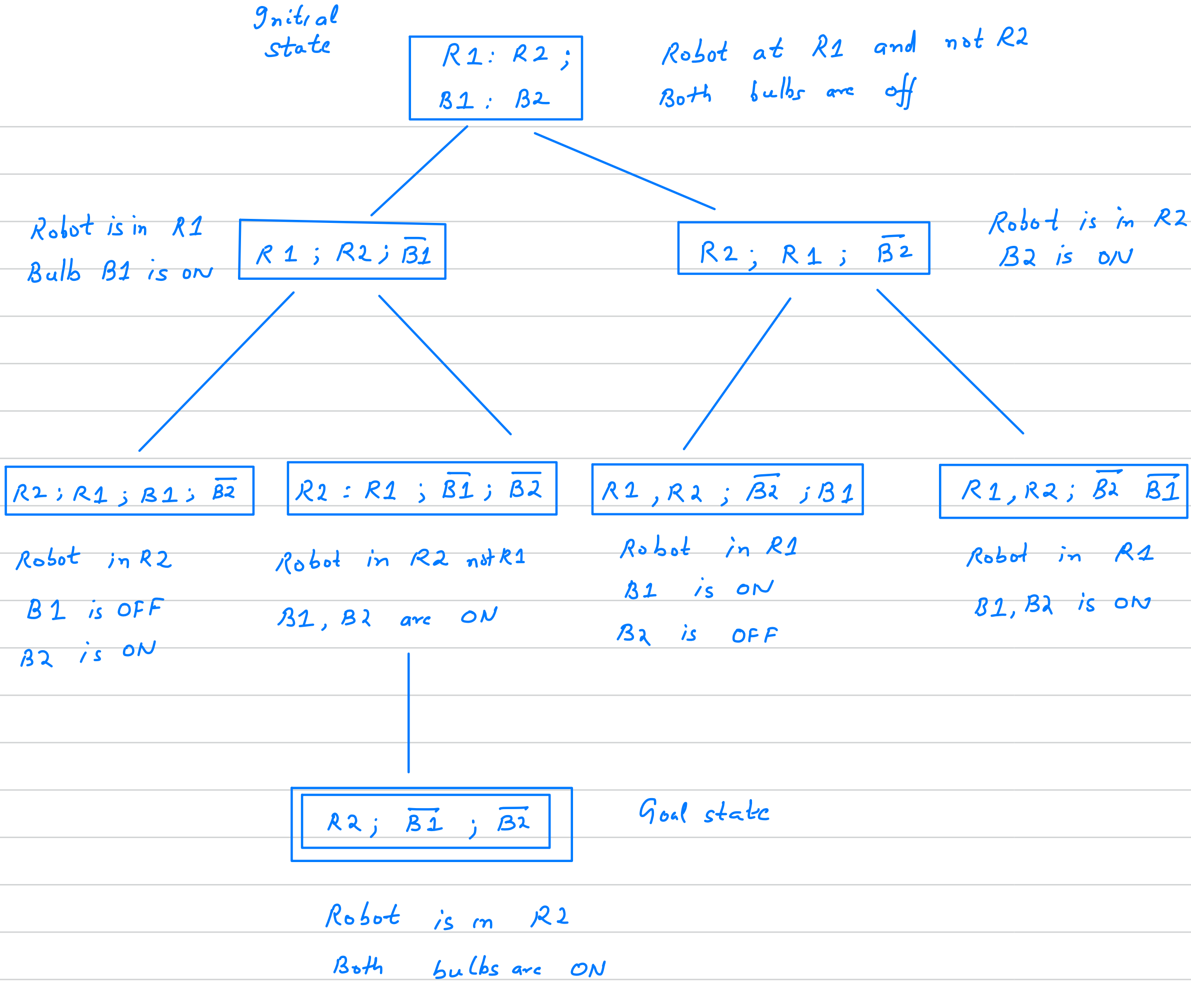
iv) goto (r, R2, R1) : Robot r goto room R2 from R1

v) switch ON(B2) : the bulb B2 will be switched ON

vi) switch OFF(B2) : the bulb B2 will be switched OFF

2) Assuming $\overline{B1}$ is switch ON on bulb B1
$\overline{B2}$ is witch ON on bulb B2

State space diagram:


Initial state
R1: R2 ;
B1 : B2
Robot at R1 and not R2
Both bulbs are off
Robot is in R1
Bulb B1 is ON
R 1 ; R2 ; $\overline{B1}$
R2 ; R 1 ; $\overline{B2}$
Robot is in R2
B2 is ON
R2 ; R1 ; B1 ; $\overline{B2}$
R2 : R1 ; $\overline{B1}$; $\overline{B2}$
R1 , R2 ; $\overline{B2}$; B1
R1 , R2 ; $\overline{B2}$ $\overline{B1}$
Robot in R2
B1 is OFF
B2 is ON
Robot in R2 not R1
B1, B2 are ON
Robot in R1
B1 is ON
B2 is OFF
Robot in R1
B1, B2 is ON
R2 ; $\overline{B1}$; $\overline{B2}$
Goal state
Robot is in R2
Both bulbs are ON

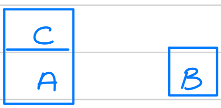
Problem 2

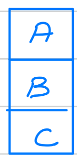
Statement : "When the operators or actions used to solve one sub-problem may interfere with the solution to a previous sub-problem and we have to undo the actions then this problem is known as Sussman Anamoly".

ex. If we have a problem X which has two parts X_1 & X_2 where X_2 depends on X_1. To solve X_2 we have to undo all actions on X_1

Block World problem

Initial State | **Goal State**





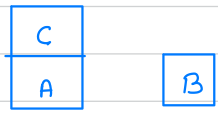
Given : A, B block on table | C block on table
C block on A block | B block on C
| A block on B

→ Divide Goal state into Sub Goal

1. on (A, B) → here A is on B
2. on (B, C) here B is on C

1. Suppose, here we satisfy "on (A, B)" first

Initial State



Goal State

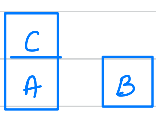
Here, to satisfy on (A, B) we will have to do 2 steps
a) table (C) → put C on table
b) on (A, B) → put A on B

Now, the problem is we cannot satisfy on (B, C) because current A is on B.

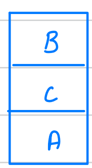
To perform subgoal on (B, C) we need to undo on (A, B)

2. Suppose if we try to satisfy "on (B, C)" first

Initial State



Goal State



Now, we cannot perform on (A, B) subgoal because B, C are on A block and we have to undo actions of on (B, C) sub goal to satisfy on (A, B) subgoal

Hence, we can say that we have to undo all the actions of the subgoal to solve other sub-goals then this problem is called sussman anamoly.

Problem 3

→ **Block World Problem**

↳ There are 'N' number of Blocks resting on table with specified sequence

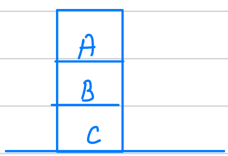
↳ GOAL : arrange them in desired sequence

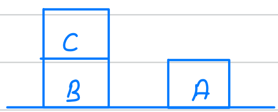
↳ Available moves :

a) Put a block on table
b) Put a block on another block
c) only move one block at a time.

↳ State is represented using of blocks in current position

ex.



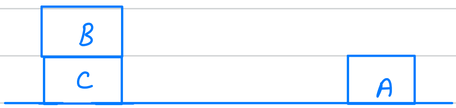


Goal state   Initial State

(i) Put 'C' on table top

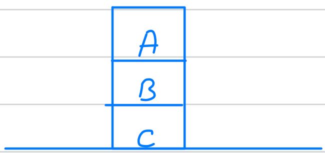
Intermediate state

(ii) Put 'B' on 'C'



Intermediate state

(iii) Put 'A' on top of 'B'



← Goal state

The same can be written as

- pickup (C) → pickup block 'C'
- put down (C) → put down 'C' on table
- stack (B, C) → put 'B' on 'C'
- stack (A, B) → put 'A' on 'B'

→ States and goal here are sentenses of first order logic

→ Operators have

<u>name</u> : the name of the operator

<u>pre condition</u> : sentence which describes the preconditions so that an operation can be executed

<u>effect</u> : describes how the world has changed because of operator execution. Consists of adding and delete operations.

Problem 4

Goal: open, painted

Initial-state:

not have-keys
not open
not painted

Getkeys

Pre:
Effe: have-keys

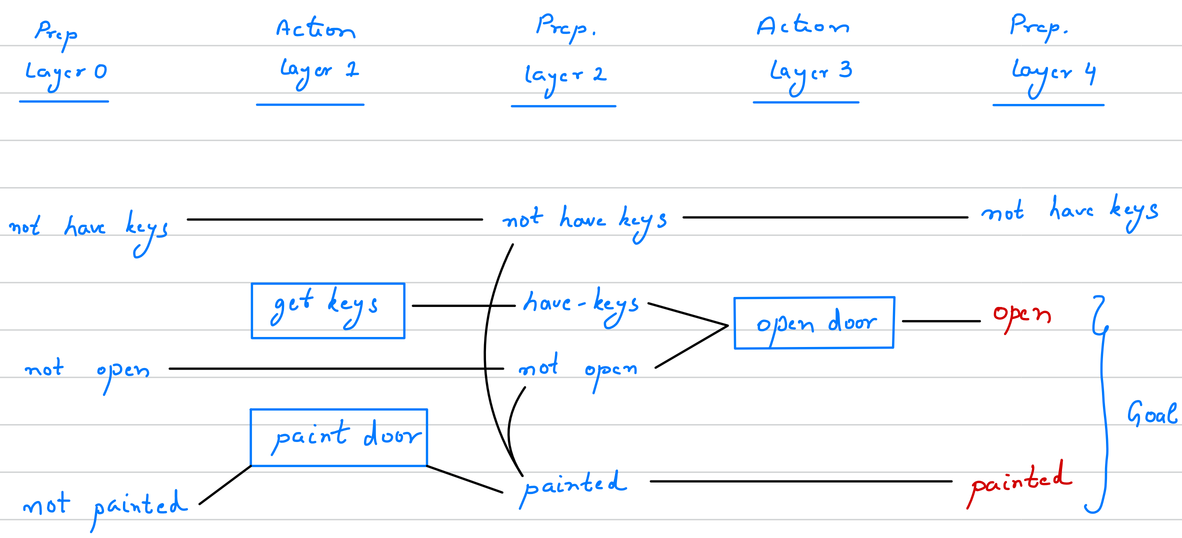
Open door

Pre: not open
have-keys
Effe. open

Paint door

Pre: not painted
Eff: painted

Graph plan



Problem 5

a) Graph plan will have two plans

Plan 1 : Action 1
Action 2

Plan2 : Action2
Action 1

Partial Order planner will give a single layer plan which would correspond to actions, Action1 and Action2 can run in parallel

We have difference because GraphPlan declares actions with inconsistent effects mutex while partial order planner will not. Hence the results of partial order plan will be more expressive as they can accomodate more possibilities.

Problem 6

The idea behind situation calculus is that (reachable) states are definable in terms of the actions required to reach them. These reachable states are called situations. What is true in a situation can be defined in terms of relations with the situation as an argument. Situation calculus can be seen as a relational version of the feature based representation of actions.

Here we only consider single agents, a fully observable environment and deterministic actions.

Situation calculus is defined in terms of situations. A situation is either

- init, the initial situation or
- do (A, S), the situation resulting from doing action A in situation S, if it is possible to do action A in situation S

example: Consider the robot delivery domain and the task of finding a path from one location to another.

We can model a state space search problem where the states are locations.

example: Consider the robot delivery domain and the task of finding a path from one location to another.

We can model a state space search problem where the states are locations.

Here we assume we have a robot at a certain location and the idea is it has to pick a key from store room. and a package from storage.

suppose initial position, init, the robot is 100 i.e LOC (100)

key K1 is at the storage room.

An action do (move (Robert, LOC (100), LOC (200), init) is the situation resulting from Robert moving from position LOC(100) init to LOC (200). In this situation, Robert is at LOC (200), the key K1 is still in store room and the package is in storage

The situation

do (move (Robert (LOC (200), store),
do (move (Robert, LOC (100), LOC(200), init))

is one in which the robot has moved from position LOC (100) to LOC (200) to move and is currently at store. Suppose Robert then picks key K1

The resulting situation is

do (pickup (Robert , k1),
do(move (Robert , LOC (200), store),
do (move (Robert , LOC (100), LOC (200), init)))

Here robert is at position store carrying key K1

A situation can be associated with a state. The differences between situation & state

- Multiple situations may refer to the same state if multiple sequences of actions lead to the same state. That is, equality between situations is not the same as equality between states.

- Not all states have corresponding situations. A state is reachable if a sequence of actions exist that can reach that state from the initial state. States that are not reachable do not have a corresponding situation.

Some do (A, S) terms do not correspond to any state. However sometimes an agent must reason about such a situation without knowing if A is possible in a state S, or if S is possible.