**1. Do Russell & Norvig problem 10.4, page 397. Note that PDDL is described on page 367.**

Action (ACTION:Go(x, y),

PRECOND: At(Shakey, x) ∧ In(x, r) ∧ In(y, r),

EFFECT: At(Shakey, y) ∧ ¬(At(Shakey, x)))

Action( ACTION:Push(b, x, y),

PRECOND: At(Shakey, x) ∧ isPushable(b),

EFFECT: At(b, y) ∧ At(Shakey, y) ∧ ¬At(b, x) ∧ ¬At(Shakey, x))

Action (ACTION:ClimbUp(b),

PRECOND: At(Shakey, x) ∧ At(b, x) ∧ isClimbable(b),

EFFECT: On(Shakey, b) ∧ ¬On(Shakey, Floor))

Action( ACTION:ClimbDown(b),

PRECOND: On(Shakey, b),

EFFECT: On(Shakey, Floor) ∧ ¬On(Shakey, b))

Action (ACTION:TurnOn(l),

PRECOND: On(Shakey, b) ∧ At(Shakey, x) ∧ At(l, x),

EFFECT: TurnedOn(l))

Action (ACTION:TurnOff(l),

PRECOND: On(Shakey, b) ∧ At(Shakey, x) ∧ At(l, x),

EFFECT: ¬TurnedOn(l))

The initial state is:

In(Switch1,Room1) ∧ In(Door1,Room1) ∧ In(Door1,Corridor)

In(Switch1,Room2) ∧ In(Door2,Room2) ∧ In(Door2,Corridor)

In(Switch1,Room3) ∧ In(Door3,Room3) ∧ In(Door3,Corridor)

In(Switch1,Room4) ∧ In(Door4,Room4) ∧ In(Door4,Corridor)

isClimbable(Box1) ∧ isClimbable(Box2) ∧ isClimbable(Box3) ∧ isClimbable(Box4)

isPushable(Box1) ∧ isPushable(Box2) ∧ isPushable(Box3) ∧ isPushable(Box4)

In(Box1,Room1) ∧ In(Box2,Room1) ∧ In(Box3,Room1) ∧ In(Box4,Room1)

In(Shakey,Room3) ∧ At(Shakey,XS)

At(Box1,X1) ∧ At(Box2,X2) ∧ At(Box3,X3) ∧ At(Box4,X4)

TurnedOn(Switch1) ∧ TurnedOn(Switch4)

A plan to achieve the goal is:

Go(XS,Door3)

Go(Door3,Door1)

Go(Door1,X2)

Push(Box2,X2,Door1)

Push(Box2,Door1,Door2)

Push(Box2,Door2, Switch2)

**2. Suppose that you have a one-way rocket (fuel for one trip) with the capacity to carry a payload of two satellites. Operators are LOAD(I), which loads an item I onto the rocket, UNLOAD(I) which unloads I from the rocket, and FLY(A,B), which flies the rocket from A to B. Assume reasonable preconditions and effects. Initially, satellites S1, S2, and the rocket are all on Mars. Your goal is to move S1 and S2 from Mars to the earth, i.e., for LOCATION(S1) and LOCATION(S2) to be earth.**

**a) What is a plan for accomplishing this goal?**

**b) Can STRIPS generate a plan to solve this problem? (You can review the STRIPS algorithm on page 14 of the classical planning slides covered in class.) If so, explain the steps it would follow. If not, explain why not.**

**c) Can the partial-order planner described in class solve this problem? If so, explain the steps it would follow. If not, explain why not.**

a) Load S1 and S2 on the rocket.

Fly from mars to earth

Unload S1 and S2 from rocket.

b)

Yes, STRIPS can generate a plan to solve this problem.

Initial State:

LOCATION(S1) = Mars ∩ LOCATION(S2) = Mars ∩ LOCATION(Rocket) = Mars

Goal State:

LOCATION(S1) = Earth ∩ LOCATION(S2) = Earth ∩ LOCATION(Rocket) = Earth

It will perform the following steps,

operator(LOAD(S1),

Precond [LOCATION(S1) = mars , LOCATION(Rocket) = mars],

Add [LOCATION(S1) = Rocket],

Delete [LOCATION(S1) = mars],

Constraint [S1 ≠ Rocket, S1 ≠ mars, Rocket ≠ mars])

operator(LOAD(S2),

Precond [LOCATION(S2) = mars, LOCATION(Rocket) = mars],

Add [LOCATION(S2) = Rocket],

Delete [LOCATION(S2) = mars],

Constraint [S2 ≠ Rocket, S2 ≠ mars, Rocket ≠ mars])

operator(FLY(mars,earth),

Precond [LOCATION(Rocket) = mars, FUEL(Rocket)],

Add [LOCATION(Rocket) = earth, ⌐FUEL(Rocket)],

Delete [LOCATION(Rocket) = mars, FUEL(Rocket)],

Constraint [earth ≠ mars ≠ Rocket])

operator(UNLOAD(S2),

Precond [LOCATION(S2) = Rocket, LOCATION(Rocket) = earth],

Add [LOCATION(S2) = earth],

Delete [LOCATION(S2) = Rocket],

Constraint [earth ≠ Rocket ≠ S2])

operator(UNLOAD(S1),

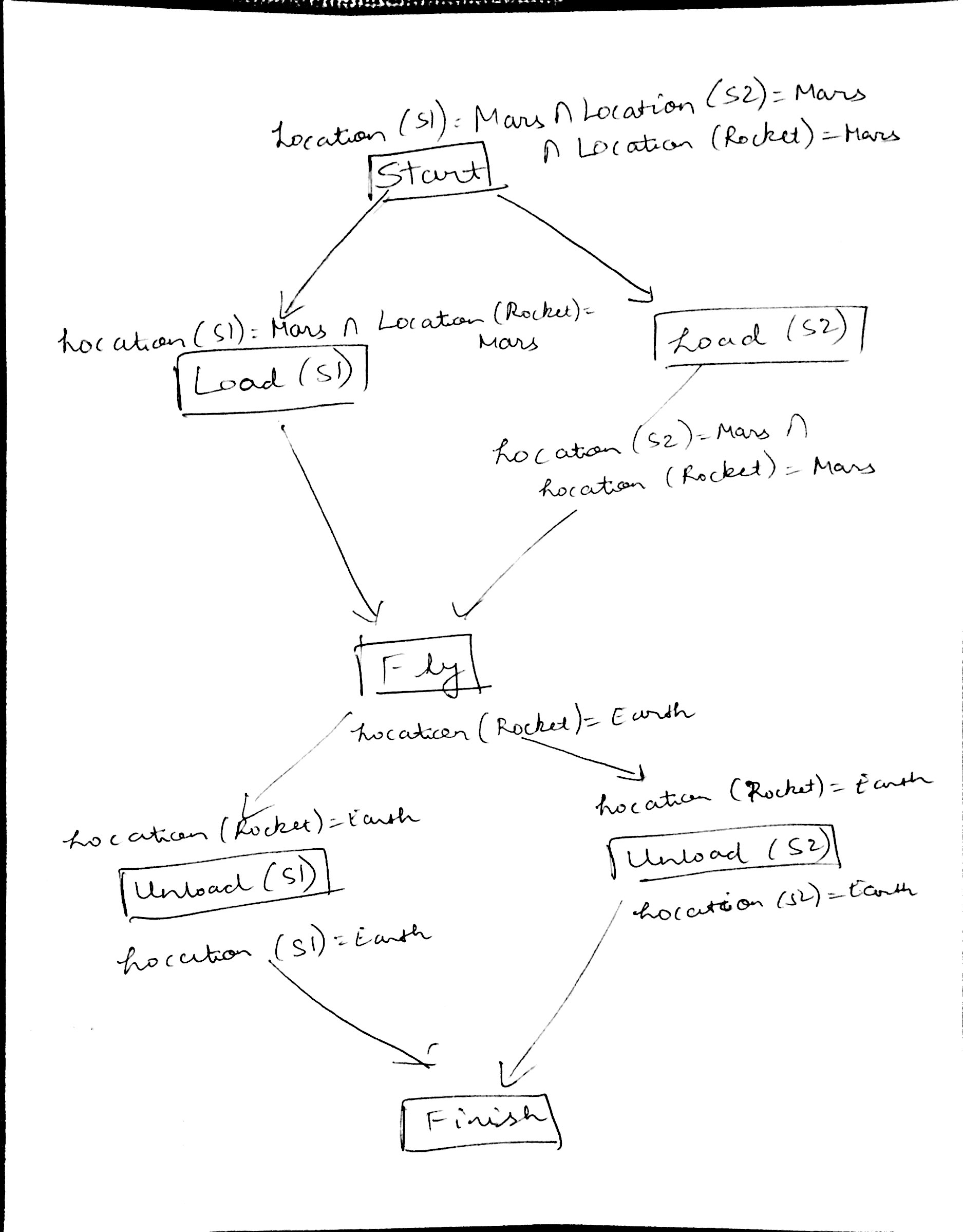
Precond [LOCATION(S1) = Rocket, LOCATION(Rocket) = earth],

Add [LOCATION(S1) = earth],

Delete [LOCATION(S1) = Rocket],

Constraint [earth ≠ Rocket ≠ S1])

c) Yes partial order can solve this problem.



**3.** **Do some runs of your CBR system and your original system to explore the tradeoff between solution generation efficiency and solution quality (measured by the length of the solution path). Describe what you find, and substantiate your observations with sample quantitative data from your tests.**

I took the following case and performed the cbr:

InitialState=makeState("blank", 3, 6, 2, 1, 5, 4, 7, 8)

Time taken for Informed Search(Manhattan) with case based : 0.0249 Seconds

numruns=66

Time taken for Informed Search(Manhattan) without case based : 0.004 Seconds

numruns =11

InitialState=makeState(2, 8, 3, 1, "blank", 5, 4, 7, 6)

Time taken for Informed Search(Manhattan) with case based : 0.009 Seconds

numruns=0

Time taken for Informed Search(Manhattan) without case based : 0.03 Seconds

numruns =9

**4. In case-based reasoning, as in human reasoning, the order in which problems are processed matters. For the task of route planning (deciding a sequence of road segments to reach a destination), starting from an empty case base, describe a problem sequence for which CBR would be maximally useful in one problem order, but much less useful in the reverse order. Assume the basic CBR approach described above.**

C--A -- B---D

A person wants to go from place A to place B. He travels the path and stores in the case base. Assume he wants to go from place C to place D. In this scenario, he will perform an analysis and determines that place A is nearer to place C and place D is nearer to place B. He uses this case and travels the route. In this scenario, Case Based Reasoning is most effective.

Now consider the other scenario.

A person knows the path from place C to the place D. The person wants to travel from place A to place B. He performs analysis and determines that place C is nearer to place A. He goes to place C, and based on the case base travels to place D. He then travels to place B from place D since place D is nearer to place B. In this scenario, Case Based Reasoning is least effective.

**5. For what types of task domains would a CBR system be preferable to a rule-based system? Explain and give examples.**

The representation of knowledge in rules based system is in the form of rules and in case based system is in the form of cases.

Rules based system are more suitable to system wherein the rules do not change periodically. Case based system uses historical experience to solve the new problems and is based on assumption that similar problems have similar solutions.

Given a problem multiple rules trigger for rules based system . Case based reasoning focuses on the retrieval of the optimal case for the problem and uses this case to solve it.

Case based system are mosre flexible, adaptive compared to rules based system.

Rule bases system can be used to select tactical moves to play game.

Case-based reasoning could be used by the diagnostic assistant to help users diagnose problems on their computer systems.