Mid-sem Answer Scripts

For each of the following statements indicate TRUE / FALSE. For the FALSE ones, write a corrected statement by modifying ONLY the underlined part.

a) The following statement is <u>valid</u>: $[\exists x \forall y\, P(y) \Rightarrow R(y,x)] \Rightarrow [\forall x P(x) \Rightarrow \exists y R(x,y)]$

→ TRUE

b) Exploration of a *max node, J,* in a game tree can be stopped when its value <u>equals or falls below the maximum current value of all the *min ancestors* of *J*</u>.

→ FALSE [equals or exceeds the minimum current value of all the min ancestors of J]

c) The notion of using *pathmax* with algorithm A* is to update the cost of a node *n* with parent *m* using the following expression: $\underline{f(n) = g(n) + \max\{h(n),\ h(m) + c(m,n)\}}$

→ FALSE [$\underline{f(n) = g(n) + \max\{h(n),\ h(m) - c(m,n)\}}$]

Mid-sem Answer Scripts

For each of the following statements indicate TRUE / FALSE. For the FALSE ones, write a corrected statement by modifying ONLY the underlined part.

d) In the algorithm AO*, when the cost of a node *n* changes, we <u>need not</u> revise the cost of every parent of the node *n*.

→ **TRUE**

e) With the use of an inadmissible heuristic function, the algorithm A* will find the path to the minimum cost goal node, *n*, if *f(n)* is <u>greater than</u> the cost, *f(m)*, of every node *m* on the path from the start node to *n*.

→ **TRUE**

Mid-sem Answer Scripts

Convert the following statement to clause normal form and write only the final form without existential quantifiers. Use Skolem functions wherever necessary:

$$[\exists x \forall y\, P(y) \Rightarrow R(y,x)] \Rightarrow [\forall x P(x) \Rightarrow \exists y R(x,y)]$$

$$[\exists x \forall y\, P(y) \Rightarrow R(y,x)] \Rightarrow [\forall x\, P(x) \Rightarrow \exists y\, R(x,y)]$$

$$= [\exists x_1 \forall y_1\, P(y_1) \Rightarrow R(y_1,x_1)] \Rightarrow [\forall x_2\, P(x_2) \Rightarrow \exists y_2\, R(x_2,y_2)]$$

$$= [\exists x_1 \forall y_1\, \neg P(y_1) \vee R(y_1,x_1)] \Rightarrow [\forall x_2\, \neg P(x_2) \vee \exists y_2\, R(x_2,y_2)]$$

$$= [\forall x_1 \exists y_1\, P(y_1) \wedge \neg R(y_1,x_1)] \vee [\forall x_2\, \neg P(x_2) \vee \exists y_2\, R(x_2,y_2)]$$

$= \sim\sim$ ~~$\forall x_1\, P(y_1) \wedge \neg R(y_1$~~

$$= \forall x_1 [P(f(x_1)) \wedge \neg R(f(x_1),x_1)] \vee \forall x_2 [\neg P(x_2) \vee R(x_2, g(x_2))]$$

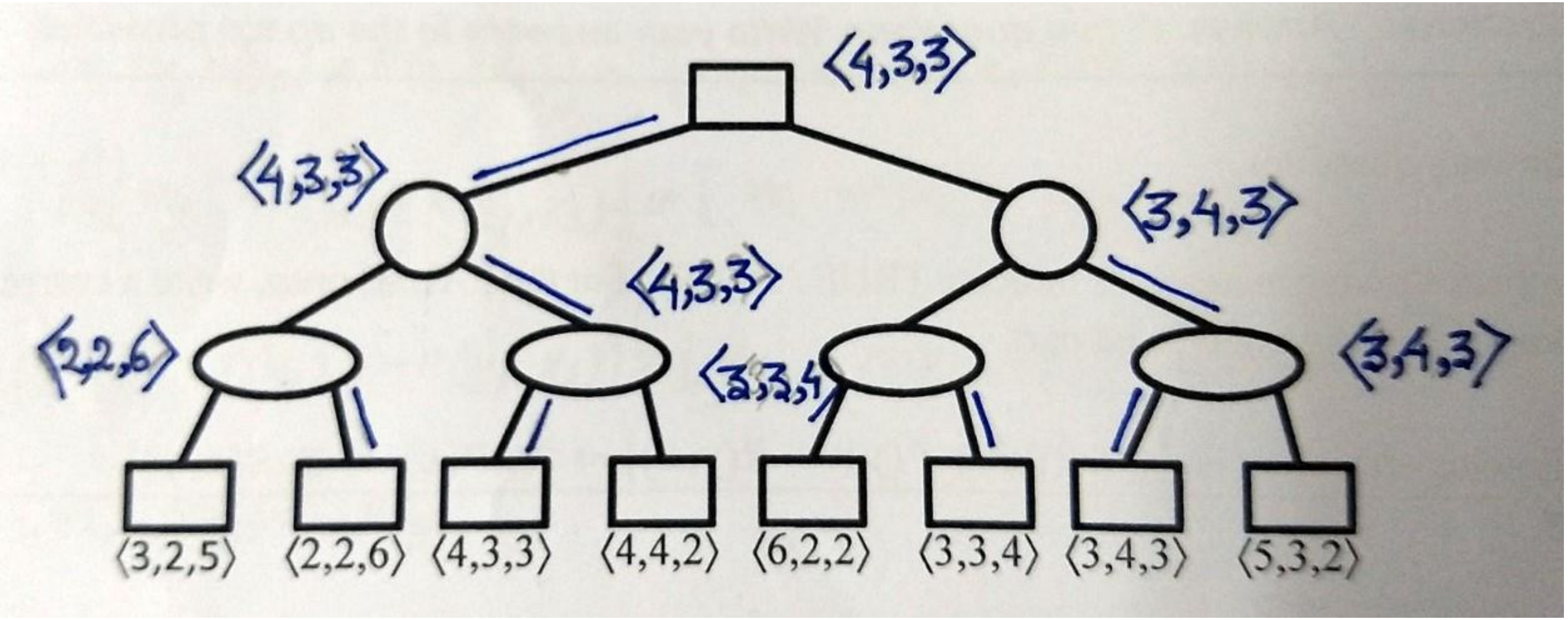
$$= \left([P(f(x_1)) \vee \neg P(x_2)] \wedge [\neg R(f(x_1),x_1) \vee \neg P(x_2)]\right) \vee R(x_2, g(x_2))$$

$$= [P(f(x_1)) \vee \neg P(x_2) \vee R(x_2, g(x_2))] \wedge [\neg R(f(x_1),x_1) \vee \neg P(x_2) \vee R(x_2, g(x_2))]$$

Mid-sem Answer Scripts

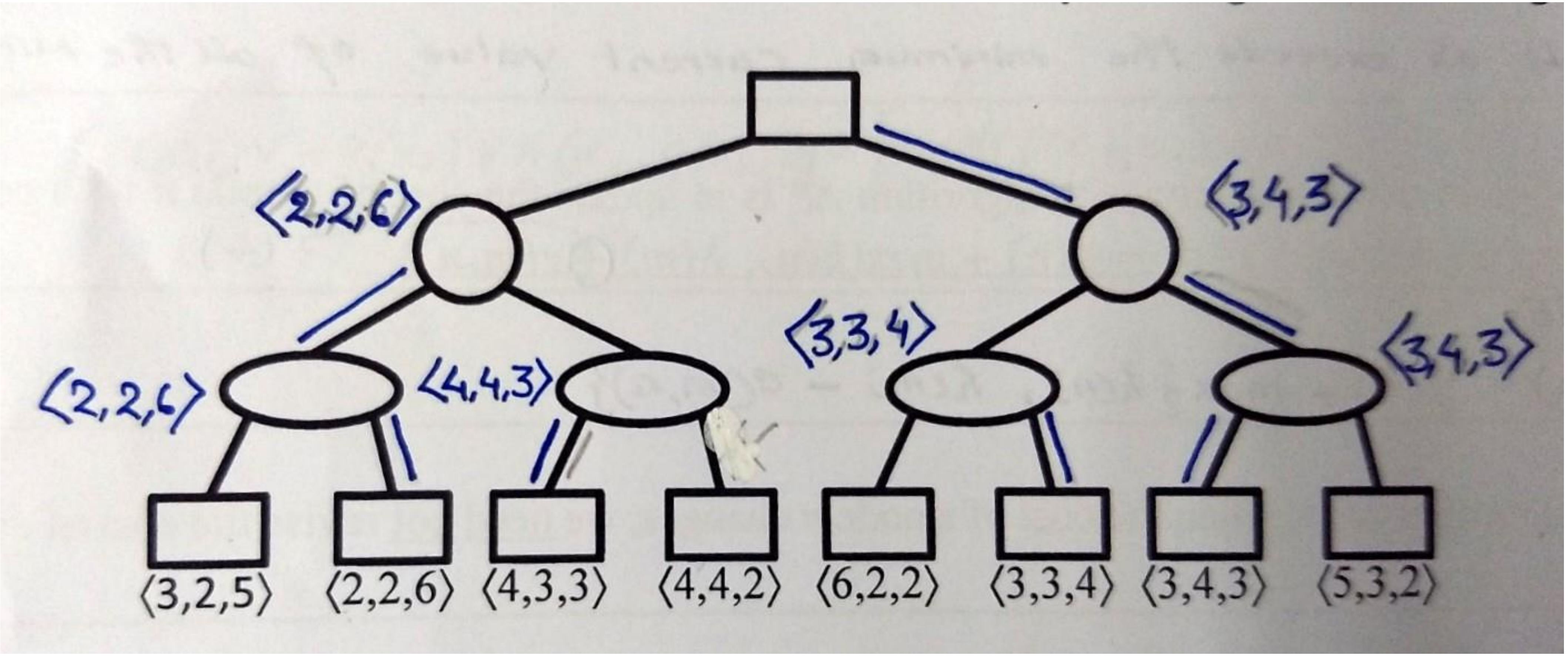
2. The following game trees are for a three player game. The vector ⟨*x, y, z*⟩ in each terminal node represents the payoffs for Player-1, Player-2, and Player-3 respectively. The square nodes represent moves for Player-1, the circular nodes represent moves of Player-2 and the oval nodes represent moves for Player-3.

a) The three player game may assume that the opponents are rational, that is, each player's sole motive is to maximize the payoff of that player. Under this assumption, show the preferred choice of successor at each node by marking the relevant edges.



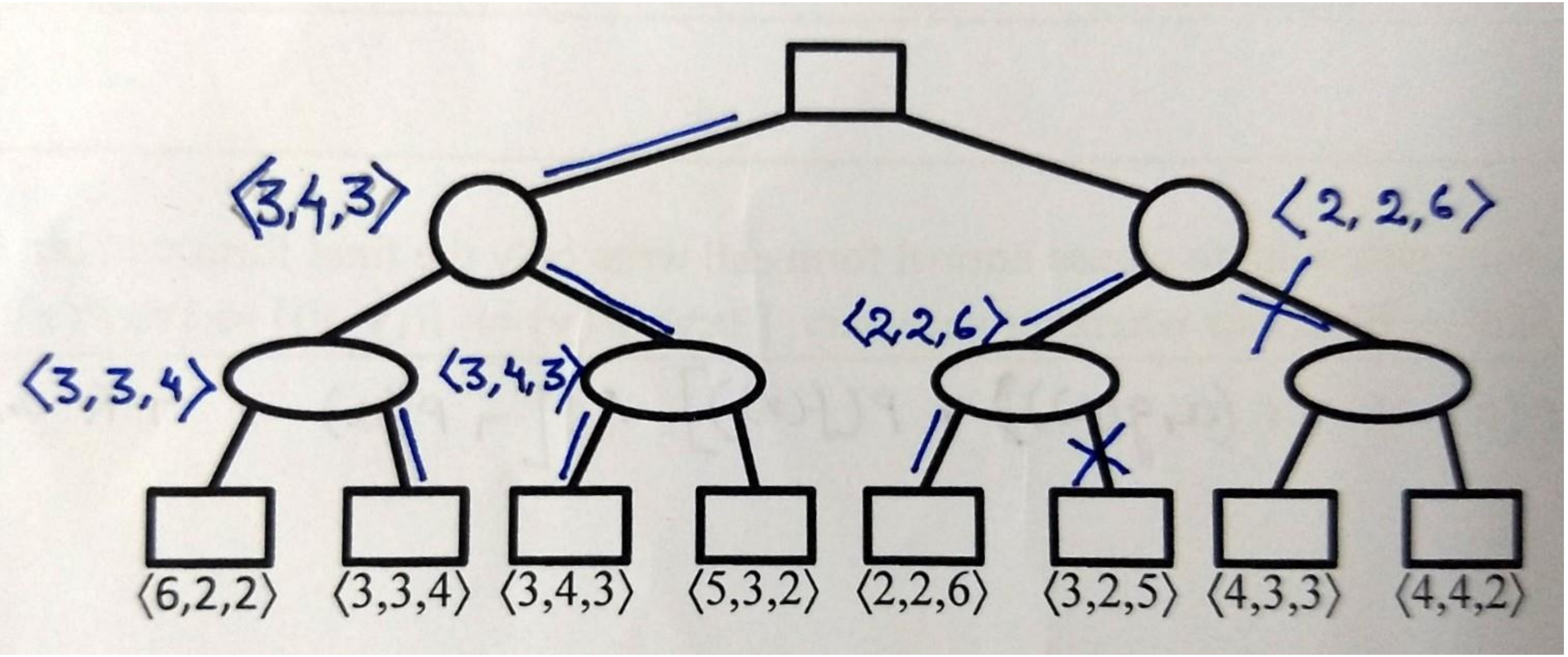
Mid-sem Answer Scripts

b) A more conservative approach for Player-1 is to choose the move without making the assumption of rational opponents. This situation allows Player-2 and Player-3 to conspire to minimize the payoff of Player-1. If this is the case, then show the preferred choice of successor at each node by marking the relevant edges.



Mid-sem Answer Scripts

c) Design a cut-off criterion similar to αβ-pruning for the situation where Player-2 and Player-3 conspire against Player-1 to minimize the payoff for Player-1. Show the pruning on the following game tree. You don't have to state the pruning criterion.



Mid-sem Answer Scripts

Consider the problem of swapping the contents of two registers, A and B. For a programmer, this is very easy, but suppose we wish to ask a robot to figure out how to write such a code. Suppose we pose it as the following planning problem in STRIPS:

Op(ACTION: Start,

EFFECT: Contains(A, X) $\wedge$ Contains(B, Y)) *// Register A contains X, Register B contains Y*

Op(ACTION: Finish,

PRECOND: Contains(B, X) $\wedge$ Contains(A, Y))

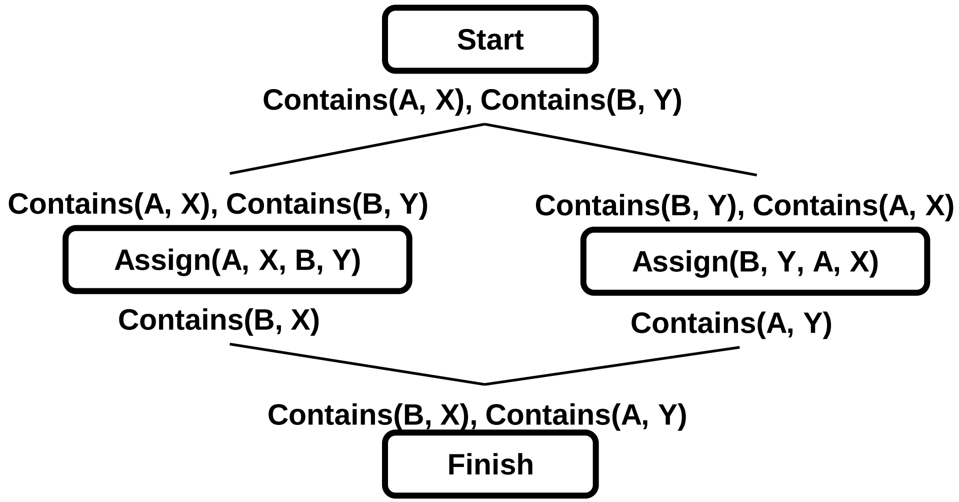
Op(ACTION: Assign(r1, v1, r2, v2), *// Assigns the content v1 of register r1 to register r2 which contained v2*

PRECOND: Contains(r1, v1) $\wedge$ Contains(r2, v2),

EFFECT: Contains(r2, v1))

Mid-sem Answer Scripts

A partial order planner produces the following plan.

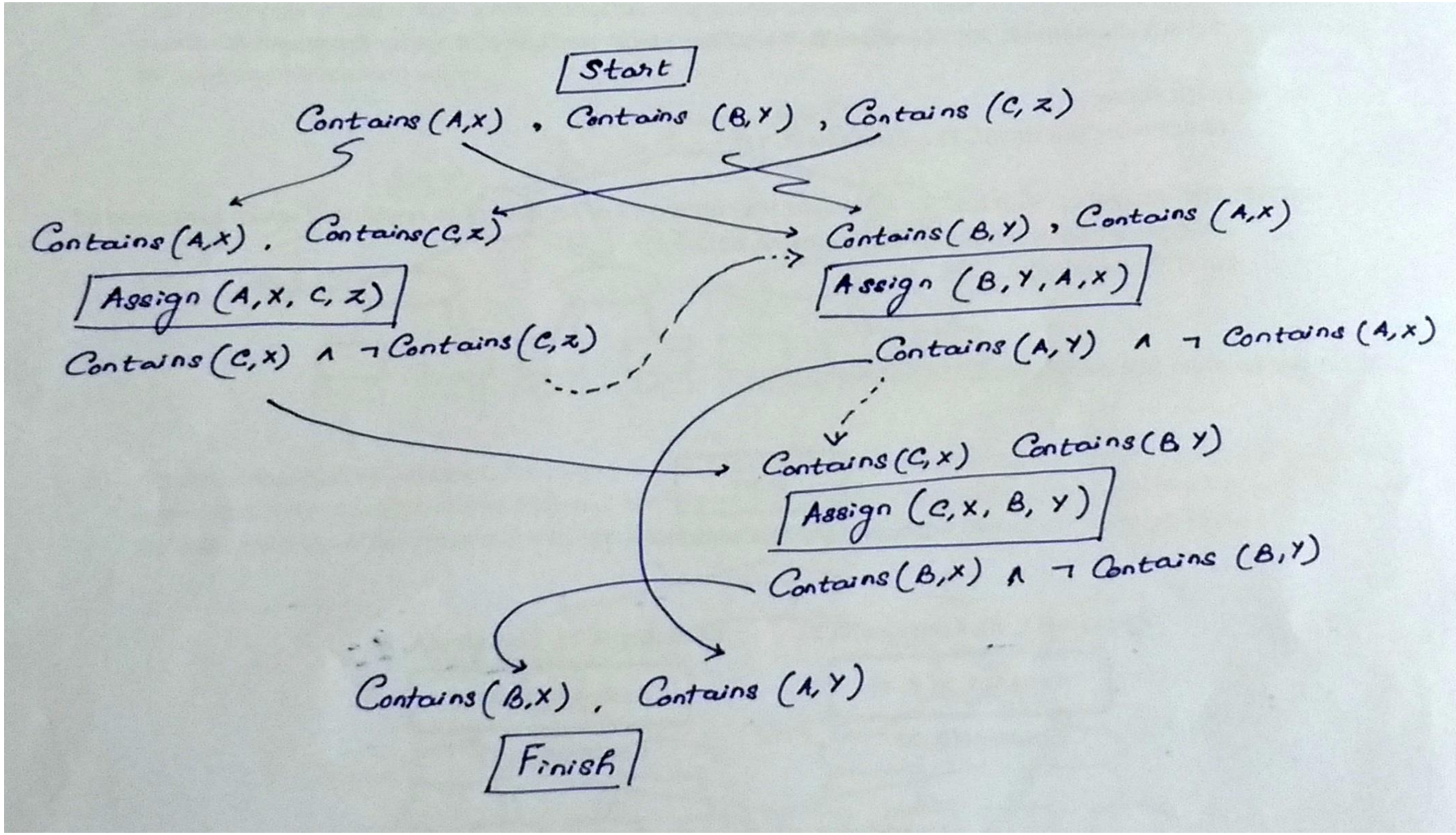


Observe that the steps of the plan cannot be executed in any order to achieve the swapping the contents of the registers. The robot is not at fault, since it was not told that assigning the contents of register r1 to register r2 destroys the previous content of register r2. Can you rewrite the action so that the correct consequence of the action is captured?

Op(ACTION: Assign(r1, v1, r2, v2)
PRECOND: Contains(r1, v1) $\wedge$ Contains(r2, v2),
EFFECT: Contains(r2, v1)) $\wedge$! Contains(r2, v2),

Mid-sem Answer Scripts

b) After the modification of part (a), observe that no totally ordered plan exists corresponding to the plan shown. Now suppose we have a third register, C. Draw a partial order plan for swapping A and B using C and show that it can then be totally ordered.



Mid-sem Answer Scripts

4. Complete the following statements in First Order Logic by filling in the blanks.

a) All students are smart: $\forall x\ (\text{Student}(x) \Rightarrow \text{Smart}(x))$

b) There exists a student: $\exists x\ \text{Student}(x)$

c) There exists a smart student. $\exists x\ (\text{Student}(x) \wedge \text{Smart}(x))$

d) Every student loves some student. $\forall x\ (\text{Student}(x) \Rightarrow \exists y\ (\text{Student}(y) \wedge \text{Loves}(x,y)))$

e) Every student loves some other student. $\forall x\ (\text{Student}(x) \Rightarrow \exists y\ (\text{Student}(y) \wedge \neg(x=y) \wedge \text{Loves}(x,y)))$

f) There is a student who is loved by every other student.

$\exists x\ (\text{Student}(x) \wedge \forall y\ (\text{Student}(y) \wedge \neg(x=y) \Rightarrow \text{Loves}(y,x)))$