CS121 – Introduction to Artificial Intelligence – Winter 2011

Homework #5
(Task Planning and Constraint Satisfaction)
Out: 2/14/11 — Due: 2/22/11 (at noon)

How to complete this HW: First copy this file; then type your answers in the file immediately below each question; start each question on a separate page, write your name on every page. Finally, print this file and return it in the drawer marked CS121 of a file cabinet located in the entryway of the Gates building next to office Gates 182 (see red arrow in http://ai.stanford.edu/~latombe/cs121/2011/map.jpg) no later than Tuesday 2/22 at noon.

Your name: ……………………………………………………………………………………

Your email address: ………………………………………………………………………

Note on Honor Code: You must NOT look at previously published solutions of any of these problems in preparing your answers. You may discuss these problems with other students in the class (in fact, you are encouraged to do so) and/or look into other documents (books, web sites), with the exception of published solutions, without taking any written or electronic notes. If you have discussed any of the problems with other students, indicate their name(s) here:
……………………………………………………………………………………………………
Any intentional transgression of these rules will be considered an honor code violation.

General information: Justify your answers, but keep explanations short and to the point. Excessive verbosity will be penalized. If you have any doubt on how to interpret a question, tell us in advance, so that we can help you understand the question, or tell us how you understand it in your returned solution.

Grading:

Problem#	Max. grade	Your grade
I	25	
II	25	
III	25	
IV	25	
Total	100	

Name ……………………………………………………………………..

I. Action Planning

A robot Robot operates in a world made of three rooms, R1, R2, and R3. There are also an apple Apple and an orange Orange in this world. The initial state is described by:

$$\text{IN(Apple, R1)} \wedge \text{IN(Orange, R2)} \wedge \text{IN(Robot, R1)}.$$

The goal of the robot is to bring the apple and the orange into room R3. It is described by:

$$\text{IN(Apple, R3)} \wedge \text{IN(Orange, R3)}.$$

The robot's five possible action schemas are described as follows in the STRIPS language:

Go(x,R1)
- P: IN(Robot, x) $\wedge$ ($x \neq$ R1)
- D: IN(Robot, x)
- A: IN(Robot, R1)

Go(x, R2)
- P: IN(Robot, x) $\wedge$ ($x \neq$ R2)
- D: IN(Robot, x)
- A: IN(Robot, R2)

Go(x, R3)
- P: IN(Robot, x) $\wedge$ ($x \neq$ R3)
- D: IN(Robot, x)
- A: IN(Robot, R3)

Pick(w,x)
- P: IN(Robot, x) $\wedge$ IN(w,x) $\wedge$ ($w \neq$ Robot)
- D: IN(w,x)
- A: HOLDING(w)

Drop(w,x)
- P: IN(Robot, x) $\wedge$ HOLDING(w)
- D: HOLDING(w)
- A: IN(w,x)

The inequality in the first action schema is treated as follows. The parameters x and y may be instantiated by any constant (Apple, Orange, R1, R2, R3, and Robot) to obtain an action. This action is applicable to a state S, if IN(Robot,x) where x has been instantiated is a proposition in S and if the two constants instantiating x and y are different. The inequality in Pick(w,x) is treated in a similar way; w must be instantiated to a constant other than Robot.

1. Construct the planning graph (S_0, A_0, S_1, A_1, …) of the initial state until it levels off. Indicate how you recognize in this particular case when it levels off. [*Don't consider "mutual exclusions", even if you know what these are. You need not present the planning graph in a "graph" form; you may just write down the contents of* S_0, A_0, S_1, A_1, *etc… as sets of propositions or actions*.]

   Write your answer as a sequence:
   S_0 = ...
   A_0 = ...
   S_1 = ...

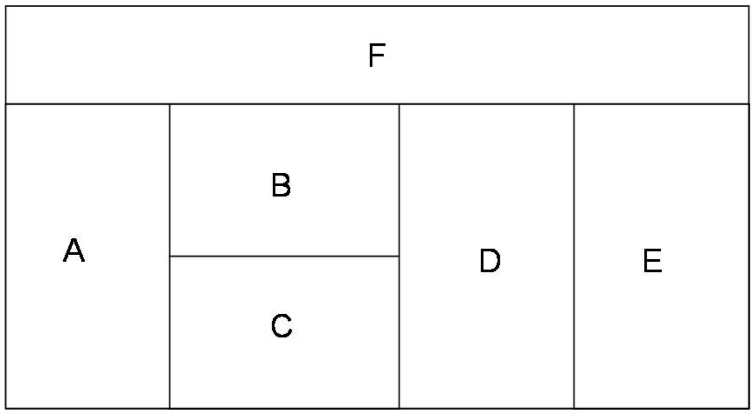
A_1 = ...
etc ...
where each S_i, i = 0, 1, 2, ..., is a set of propositions, such as IN(Apple,R1) and IN(Robot,R1), and each A_i is a set of actions, such as Go(R1,R3) and Pick(Apple,R1).

2. Give the general definition of the level cost of a goal in a planning graph. In the above planning problem, what is the level cost of IN(Apple, R3) $\wedge$ IN(Orange, R3)?
3. From now on, assume the robot plans its actions using a backward planning method. What are the actions relevant to achieving the goal IN(Apple, R3) $\wedge$ IN(Orange, R3)? Why? [*See slide #43 of lecture on Action Planning.*]
4. Compute the regression of the goal IN(Apple, R3) $\wedge$ IN(Orange, R3) through each of the relevant actions that you have identified at Question 3.
5. The robot's planner uses the level cost as a heuristic to decide which regressed goal to expand (assume that all actions have unit cost). Of the regressed goals that you have computed at Question 4, which one it will decide to expand (i.e., to regress further through relevant actions)? Why?
6. The description of the actions allows the robot to hold several objects simultaneously. Using the additional proposition HANDEMPTY, give the description of the actions so that the robot can hold at most one object. What is the new description of the initial state?
7. With the modified descriptions of the actions and the initial state, would the decision of the planner at Question 5 be different? Why? [The explanation should be brief. No need to reconstruct the entire planning graph.]

Name …………………………………………………………………..

II. Constraint Satisfaction

The following diagram represents the map of a country made of 6 states. Each state must be colored in red, green or blue, so that no two adjacent states get the same color.



We represent this map-coloring problem as a constraint satisfaction problem with 6 variables (A, B, C, D, E, and F), each having the same domain {red, green, blue} of values.

To solve this problem, we use the CSP-BACKTRACKING algorithm presented in class with the forward-checking operation performed each time a value is assigned to a variable. **No** other constraint-propagation operation (such as AC3) is performed. The algorithm uses the most-constrained-variable, the most-constraining-variable, and the least-constraining-value heuristics to select the variables and their values. Whenever several variables are tying for selection, the algorithm selects the first in alphabetic order. Whenever several values are tying for selection, the algorithm selects them in the following order: red, green, blue.

1. Which variable will be selected first by the algorithm? Why?

2. Which value will be assigned to this variable? Why? Which values of which variable domains does the forward-checking operation then remove?

3. Which variable will be selected next? Why?

4. Which value will be assigned to this variable? Why? Which values of which variable domains does the forward-checking operation then remove?

5. Use your answers to the above questions to count the number of complete valid assignments for this problem. Tell us how you proceed to do the counting.

Name ………………………………………………………………..

III. CSP: Train Scheduling

Two trains, the A-line and the B-line, use the simple rail network shown in Figure 2. We must schedule a single daily departure for each train from its respective departing terminus (terminus T1 for the A train and terminus T3 for the B train). Each departure will be on an hour between 1pm and 7pm, inclusive. The trains run at identical constant velocities. Each line consists of 5 segments (see figure) and each segment takes a train one hour to cover.

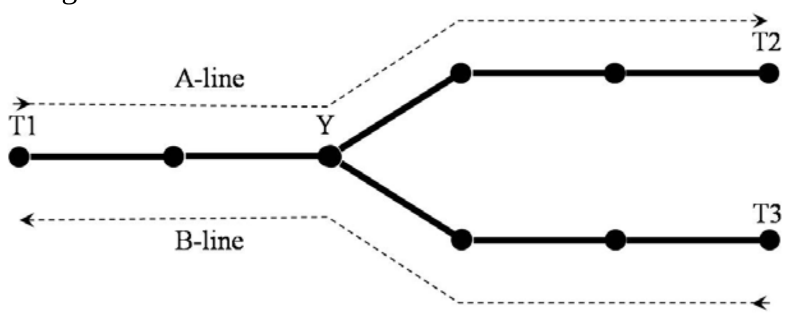


Figure 2: Train line tracks for problem IV

We formulate this scheduling problem as a constraint satisfaction problem in which the two variables, *A* and *B*, represent the departure times of the trains from their initial stations. The domain of each variable is {1, 2, 3, 4, 5, 6, 7}. If $A = 1$ and $B = 3$, them the A-line train leaves at 1pm and the B-line train leaves at 3pm.

The two trains cannot pass each other in the two track segments that they share (between terminus T1 and intersection Y). The only points in the shared region where the two trains can pass safely are T1 and Y. For example, if the A-line train leaves at 3pm and the B-line train at 2pm, then the two trains will be at the intersection Y at 5pm where they can pass each other safely.

1. If the B-line train leaves at 1pm, list the times that the A-line train can safely leave.

2. State the binary constraint between the variables *A* and *B*. You should state them in the form of mathematical inequalities and Boolean connectives (e.g., OR, AND), not in the form of a statement in English that the trains should not be in the shared track segments at the same time.

3. Assume now that the A-line train must leave at 4pm or 5pm and the B-line train must leave between 1pm and 7pm, inclusive. Write down the new domains of *A* and *B* taking into account the unary constraints stated in the previous sentence. Which values on each domain is removed by applying the AC3 algorithm using the constraints of question #2?

Name ……………………………………………………………………..

IV. Tiling with Dominoes

Consider the problem of tiling a planar region R with n dominoes. Each domino is a 2×1 rectangle. R is an arbitrary collection of $2n$ 1×1 squares. Figure 1 shows one example of such a region. The squares are numbered 1 through $2n$. R is described by the set of all the pairs (a,b), $a, b \in \{1, 2, \ldots, 2n\}$, $a < b$, such that square a and square b are edge-connected (i.e., have an edge in common). In this representation, let $R = \{p_1, p_2, \ldots, p_r\}$, where each p_k, k = 1 to r, is a pair of edge-connected squares.

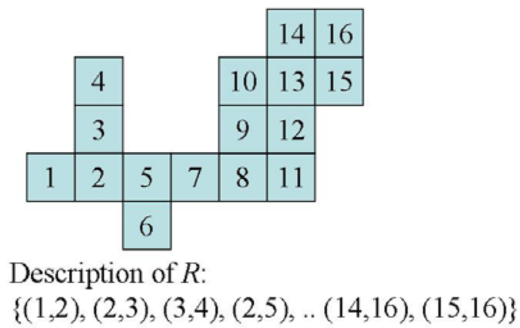


Figure 1: Example of a region R where $2n$ = 16.

1. Formulate this problem as a constraint-satisfaction problem (CSP) where the dominoes are the variables, that is, define the variable domains and the constraints. [*Note: The dominoes are just 2×1 rectangle, each made of 2 identical squares. The number or dots in each square are irrelevant to this problem. We are not "playing" dominoes here.*]
2. Formulate this problem as a CSP where the squares are the variables. In this formulation, keep the size of the domains as small as possible (4 values at most) by noticing that it does not matter which particular domino goes on a given pair of squares.
3. Construct two examples with 6 squares each, such that the constraint-graph of the CSP formulated as in the above Question 2 is tree-structured for one example and not tree-structured (but connected) for the other.
4. Consider the region R shown in Figure 1. We want you to solve the corresponding CSP formulated as in Question 2 using the backtracking search algorithm with forward checking and the most-constrained-variable heuristic. We ask you to report what each step of the algorithm performs in the following format:

   Step 1 (choice of variable and value): *Here tell us which variable is selected and which value is assigned to it.*

   Step 2 (forward checking): *Here tell us what are the variables whose domains are reduced and what are their new domains.*

   Step 3 (choice of variable and value): …

   Step 4 (forward checking): …

   Etc …

Name ……………………………………………………………………..

When there is a tie among several variables using the most-constrained-variable heuristic, select the variables in increasing numerical order (of the squares). When several values can be assigned to a variable, select the values in increasing numerical order. We ask you to only fill in the first 12 steps. For Steps 1 and 2, briefly justify your answer. For the other steps, just give your answer.