CS4341 Introduction to Artificial Intelligence.   A Term 2017

SOLUTIONS Exam 2 - September 26, 2017

Solutions by Michael Sokolovsky, Samuel Ogden, Ahmedul Kabir, and Prof. Ruiz
Department of Computer Science
Worcester Polytechnic Institute
Answers provided in green font.

Problem I. Constraint Satisfaction Problems (CSP) [25 Points]

This constraint satisfaction problem is a simplified version of Sudoku in a 4x4 matrix. The goal is to fill in each cell in the matrix with a number between 1 and 4 in such a way that no number is repeated on the same column or on the same row. To save you time, some cells have already been filled in with a value. The remaining ones have been named with a letter for easy reference. These letters, A, B, C, D, E, F and G, are the variables in the constraint satisfaction problem.

2	A	3	B
4	C	1	2
1	D	E	F
3	G	4	1

Variables: A, B, C D, E, F, and G.

Domain: The domain of each variable is {1, 2, 3, 4}.

Constraints: There is a constraint between each pair of cells P and Q that belong to the same column or to the same row of the matrix stating that the values assigned to the two cells cannot be equal.

Answer the questions below as if you were an agent following the CSP algorithms we studied in class.

1. [5 Points] Fill in the table below (some values are provided as examples to guide you. For instance, A has two remaining values, 1 and 4, and it has constraints with four other variables B, C, D, G.):

Variable	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Remaining values	1, 4	**4**	**3**	2, 3, 4	**2**	**3,4**	**2**
# of constraints with other variables	four	**Two (A,F)**	**Three (A,D,G)**	five	**Two (D,F)**	**Three (B,D,E)**	**Three (A,C,D)**

2. [5 points] Using the Minimum Remaining Values (MRV) heuristic, list the variable that the CSP search algorithm will select next. If there are ties, list all the variables that have the same MRV.

   Looking at the top row of the table, you can see that the Minimum # of remaining values for the variables is one. The MRV = 1. Variables B,C, E, and G all have remaining values equal to this MRV

3. [5 points] If the above was a tie, use the degree heuristic (i.e., variable with the most constraints on remaining variables) to break the tie. What variable would be selected? If a tie still remains, provide a systematic way to deal with the tie so that only one variable is selected. Explain your work.

   We need to select a variable from (B, C, E, G) as we found in question #2. B and E have 2 constraining variables whereas C and G have 3 constraining variables. We should then choose either C or G. We can break this tie arbitrarily using say alphabetical order. In this case, C is chosen.

4. [10 Points] Starting from the following possible values, use forward checking to propagate constraints. Show the propagation of just one constraint at a time neatly on a separate row in the table below, until no more constraints can be propagated. An example is provided on the 3rd row. You may not need all the rows provided here.

   This is one possible way of filling out the table. You could fill out the table in three steps of propagating constraints

Constraint Propagation	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Possible values	1, 4	4	3	2, 3, 4	2	3, 4	2
Constraint between: ***A and B***	1	4	3	2, 3, 4	2	3, 4	2
Constraint between: **F and B**	**1**	**4**	**3**	**2, 3, 4**	**2**	**3**	**2**
Constraint between: **D and E**	**1**	**4**	**3**	**3, 4**	**2**	**3**	**2**
Constraint between: **D and F**	**1**	**4**	**3**	**4**	**2**	**3**	**2**

Problem II. Propositional Logic [30 Points]

1. [10 points] Convert each of the following sentences to clausal form.
   a) [2 Points] $P \Rightarrow Q$
      Write the resulting clause below:
      C1: $\neg \boldsymbol{P} \vee \boldsymbol{Q}$
   b) [2 Points] $(L \wedge M) \Rightarrow P$
      C2: $\neg(\boldsymbol{L} \wedge \boldsymbol{M}) \vee \boldsymbol{P} \equiv \neg \boldsymbol{L} \vee \neg \boldsymbol{M} \vee \boldsymbol{P}$
   c) [2 Points] $(B \wedge L) \Rightarrow M$
      C3: $\neg(\boldsymbol{B} \wedge \boldsymbol{L}) \vee \boldsymbol{M} \equiv \neg \boldsymbol{B} \vee \neg \boldsymbol{L} \vee \boldsymbol{M}$
   d) [2 Points] $(A \wedge B) \Rightarrow L$
      C4: $\neg(\boldsymbol{A} \wedge \boldsymbol{B}) \vee \boldsymbol{L} \equiv \neg \boldsymbol{A} \vee \neg \boldsymbol{B} \vee \boldsymbol{L}$
   e) [1 Points] A
      C5: $\boldsymbol{A}$
   f) [1 Points] B
      C6: $\boldsymbol{B}$

2. [10 points] Use **resolution** to prove the sentence Q (that is, prove that Q is true) from the 6 clauses above. Show your work. *Hint:* Remember that the first step of the process is to negate the sentence that you want to prove.

C7: $\neg Q$ We start by assuming this to show that it will cause a contradiction.

C8: $\frac{\neg P \vee Q}{\neg P}$ (using C7 and C1) Because we are assuming $\neg Q$ then it would mean that it must be the case that $\neg P$ is true

C9: $\frac{\neg L \vee \neg M \vee P}{\neg L \vee \neg M}$ (using C8 and C2))Because $\neg P$ is true, then P is false so for this statement to be true then it must be the case that $\neg L \vee \neg M$ is true.

C10: $\frac{\neg B \vee \neg L \vee M}{\neg L \vee M}$ (using C6 and C3) Because from C6 we know that B is true, so it must be that $\neg L \vee M$ is true

C11: $\frac{\neg L \vee M}{\neg L}$ (using C9 and C10) Because since we know from C9 that $\neg L \vee \neg M$ must be true we can conclude that $\neg L$ must be true

C12: $\frac{\neg A \vee \neg B \vee L}{\neg A \vee \neg B}$ (using C4 and C11) Because we know $\neg L$ is true then L must be false meaning that $\neg A \vee \neg B$ must be true

C13: $\frac{A}{\neg B}$ (using C12 and C5) Because we know A to be true (C5) then it must be that $\neg B$ is true

C14: $\frac{B}{\{\}}$ (using C13 and C6) Because we know that B is true then we have a contradiction and so Q must in fact be true!

3. [10 points] This problem is independent from the two problems above. Determine whether the following propositional sentence is valid, unsatisfiable, or neither. Justify your answer with either truth tables, equivalent rewrites of the sentence, or both. Show your work.

$$((\neg\neg C) \Rightarrow \neg D) \Rightarrow (D \Rightarrow \neg C)$$

$(C \Rightarrow \neg D) \Rightarrow (D \Rightarrow \neg C)$ by removing the double negation

$(\neg C \vee \neg D) \Rightarrow (\neg D \vee \neg C)$ by rewrite the implication to an equivalent form, following order of operations (note: at this point this can be rearranged to $(\neg C \vee \neg D) \Rightarrow (\neg C \vee \neg D)$ which can be seen to be always true and then you're done!)

$\neg(\neg C \vee \neg D) \vee (\neg D \vee \neg C)$ by rewriting the big implication in the middle

$(C \wedge D) \vee (\neg D \vee \neg C)$ by using de Morgan's law to distribute the negation into the parentheses

$\big((C \wedge D) \vee \neg D\big) \vee \big((C \wedge D) \vee \neg C\big)$ by distributing the main $\vee$: in other words, breaking out the second sentence into two parts, each combined with the first term

$\big((C \vee \neg D) \wedge (D \vee \neg D)\big) \vee \big((C \vee \neg C) \wedge (D \vee \neg C)\big)$ by distributing $\vee$ over the $\wedge$ in both sentences

$(C \vee \neg D) \vee (\neg C \vee D)$ by removing $\big((D \vee \neg D)\big)$and $\big((C \vee \neg C)\big)$ which are always true

$C \vee \neg C \vee D \vee \neg D$ by removing parentheses (all connectives are the same $\vee$ so parentheses are not needed) and rearranging the sentence so like complementary literals are together

And finally this statement is always true then the initial propositional sentence is valid!

Problem III. First Order Logic [20 Points]

Consider the Wumpus world discussed in class and in the textbook. We focus here only on the presence of pits and the breeze that they cause in adjacent cells. Each cell is given a letter name for easy reference. Remember that capital letters are used for constants and lower case letters for variables.

A	B	C	D
E **Pit?**	F	G	H
I **Breeze**	J	K	L
M	N $\neg$ **Breeze**	O	Q

Goal: The job for your intelligent agent is to determine whether there is a pit in cell E. In other words, your agent needs to determine whether Pit(E) is true.

Knowledge Base Facts:

Fact 1. Breeze(I)

Fact 2. $\neg$ Breeze(N)

Fact 3. Adjacent(E, I)

Fact 4. Adjacent(J, I)

Fact 5. Adjacent(M, I)

Fact 6. Adjacent(M, N)

Fact 7. Adjacent(J, N)

Knowledge Base Rules:

We will use three rules from the Wumpus world. The first two rules state that "*a cell in this world is breezy if and only if one of its adjacent cells contains a pit*". To keep it simple, the first one of these rules has already been instantiated to the cell of interest, namely cell I, and its adjacent cells, E, J, and M. The third rule states that the Adjacent relationship is commutative. Variables x and y in the following statements are assumed to be universally quantified.

Rule 1. Breeze(I) $\Rightarrow$ Pit(E) $\vee$ Pit(J) $\vee$ Pit(M)

Rule 2. $\neg$ Breezy(y) $\wedge$ Adjacent(x, y) $\Rightarrow$ $\neg$ Pit(x)

Rule 3. Adjacent(x, y) $\Rightarrow$ Adjacent(y, x)

Other Logical Rules:

You can use any standard logical equivalences (e.g., $\vee$ is commutative; and $\vee$ is associative) and other valid sentences like the following, where α and β are any first order sentences:

$$(\alpha \vee \beta) \wedge \neg\beta \Rightarrow \alpha$$

[20 points] Using **forward chaining**, show how your intelligent agent would derive that there is a pit in cell E. That is, prove Pit(E) using forward chaining.

Show your work clearly and neatly. Justify each step of your derivation. State which Facts and/or Rules you use at each step of your forward chaining derivation. Number each new Fact (e.g., *Fact 8, Fact 9, Fact 10, Fact 11*, …) that you obtain during your derivation so that you can refer to it easily.

From Fact 1, applying Rule 1, we get

Fact 8: Pit(E) $\vee$ Pit(J) $\vee$ Pit(M)

From Fact 2 and Fact 6, applying Rule 2, we get

Fact 9: $\neg$ Pit(M)

From Fact 2 and Fact 7, applying Rule 2, we get

Fact 10: $\neg$ Pit(J)

From Facts 8 and 9, applying rule $(\alpha \vee \beta) \wedge \neg\beta \Rightarrow \alpha$ we get

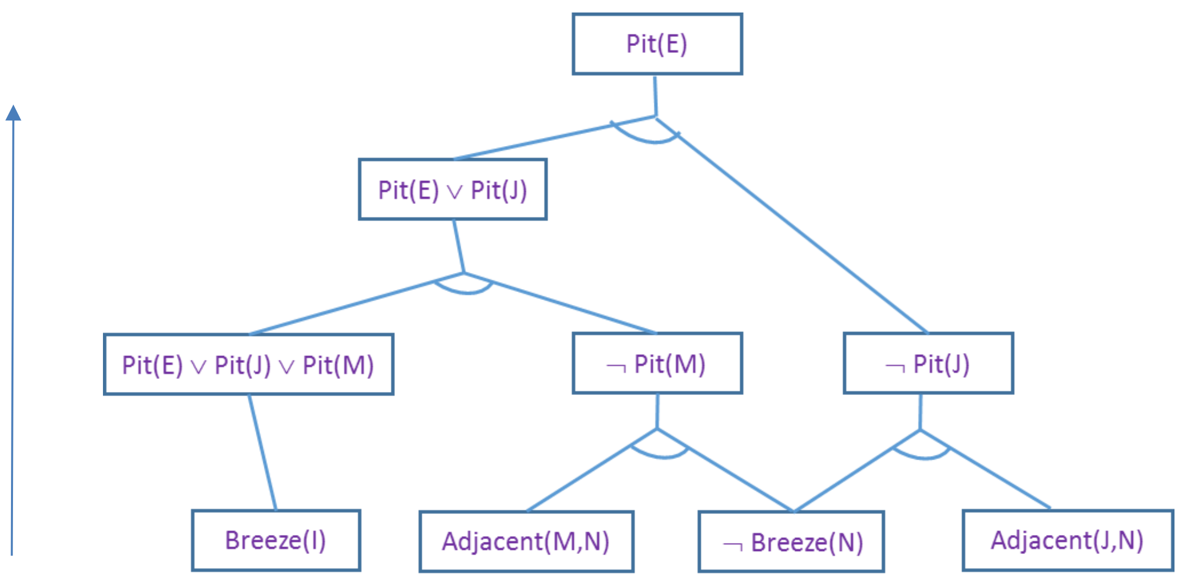
Fact 11: Pit(E) $\vee$ Pit(J)

From Facts 10 and 11, applying rule $(\alpha \vee \beta) \wedge \neg\beta \Rightarrow \alpha$ we get

Fact 12: Pit(E) [proved]

Note: Forward chaining can create facts not directly leading to the goal. So if you show the correct derivation of more facts than the ones given, that is perfectly fine.

A graphical representation of the forward chaining is given below:



Problem IV. Planning [25 Points]

Consider an extension of the planning problem discussed in class. An intelligent robot needs to devise a plan to achieve the goal of having its socks, pants, and shoes on, starting from an initial, naked state. The actions available to the robot are below, where "side" is a variable with possible values "Left" and "Right":

Action: *Put-Sock-On(side)*
  Precondition: *none*
  Effect: Has-Sock-On(side)

Action: *Put-Pants-On*
  Precondition: Has-Sock-On(Right) $\wedge$ Has-Sock-On(Left)
  Effect: Has-Pants-On

Action: *Put-Shoe-On(side)*
  Precondition: Has-Sock-On(side) $\wedge$ Has-Pants-On
  Effect: Has-Shoe-On(side)

Starting from the partial plan provided on the next page answer the following questions below.

1. Take the open condition (i.e., currently unsatisfied subgoal) *Has-Shoe-On(Left)*.
   a. [1 points] Is there a step (= action) in the current plan that satisfies this open condition? Yes or No?
      Your answer: __No___ The only steps in the plan so far are Put-Sock-On(Left) and Put-Shoe-On(Right), which don't cause Has-Shoe-On(Left) to be true.

   b. [4 points] If your answer is "Yes", add a link between that step and this open condition in the partial plan. If your answer is "No", add a new step to the plan satisfy this open condition.
      See the link on the next page.
2. Take any one of the occurrences of the open condition *Has-Pants-On*.
   a. [1 points] Is there a step in the current plan that satisfies this open condition? Yes or No?
      Your answer: __No___ The only steps in the plan so far are Put-Sock-On(Left), Put-Shoe-On(Right) and Put-Shoe-On(Left), which don't cause Has-Pants-On to be true.

   b. [4 points] If your answer is "Yes", add a link between that step and this open condition in the partial plan. If your answer is "No", add a new step to satisfy this open condition.
      See the link on the next page.
3. [5 points] Continue selecting open conditions in the plan and adding links from steps (= actions) currently in the plan to satisfy those conditions or introducing new steps as needed until you obtain a complete plan. Show your work on the next page.
   See the complete plan on the next page.
4. [5 points] Can the introduction of a new step (= action) in the plan cause any threats (i.e., interferences)? Yes or No? Explain.
   No. The three given actions in this problem are such that they don't undo the effect of any of the other actions. So not threats can occur when these actions (steps) are introduced into the plan. If we had other actions, like for instance removing a shoe, or a sock, or pants, then threats may occur.

5. [5 points] **When several options are available to satisfy an open condition**, the search algorithm explores these options in a hill-climbing manner (if a heuristic is provided) or in depth-first search manner (if a heuristic is not provided), allowing for backtracking, until a complete plan is formed, or the search fails. Would backtracking be needed for this particular planning problem with the given initial and goal situations, and actions above? Explain.
   No. Note that here there is just one action available to achieve a given sub-goal (e.g., the only way for the robot to get a sock on is to use the action Put-Sock-On). Hence, alternative ways of achieving a sub-goal wouldn't be available to store in the search queue for exploration if a search path fails.
   Note that the planning search algorithm can introduce actions (=steps) to achieve sub-goals in any order as long as it ends up with a complete plan. It doesn't need to introduce the actions in the order they will be performed when the plan is executed. Hence, introducing say the Put-Pants-On step before introducing the Put-Sock-On(Right) step in the plan doesn't imply that backtracking is needed.

In the figure below, each action in the plan is depicted inside a rectangle, with its preconditions above and its effects below the rectangle. Remember that at the START, the robot is naked - that is, it has no socks, shoes or pants on.