CS 188
Fall 2019

Final Review CSPs

Q1. Encrypted Knowledge Base

We have a propositional logic knowledge base, but unfortunately, it is encrypted. The only information we have is that:

- Each of the following 12 boxes contains a propositional logic symbol (A, B, C, D, or E) or a propositional logic operator and
- Each line is a valid propositional logic sentence.

□ 1 □ 2

□ 3 □ 4 □ 5

□ 6

□ 7 □ 8 □ 9

□ 10 □ 11 □ 12

(a) We are going to implement a constraint satisfaction problem solver to find a valid assignment to each box from the domain $\{A, B, C, D, E, \wedge, \vee, \neg, \Rightarrow, \Leftrightarrow\}$.

Propositional logic syntax imposes constraints on what can go in each box. What values are in the domain of boxes 1-6 after enforcing the unary syntax constraints?

Box	Remaining Values
1	
2	
3	
4	
5	
6	

(b) You are given the following assignment as a solution to the knowledge base CSP on the previous page:

$$\begin{array}{l} \neg\ A \\ B \Rightarrow A \\ D \\ C \lor B \\ D \lor E \end{array}$$

Now that the encryption CSP is solved, we have an entirely new CSP to work on: finding a model. In this new CSP the variables are the symbols $\{A, B, C, D, E\}$ and each variable could be assigned to *true* or *false*.

We are going to run CSP backtracking search with forward checking to find a propositional logic model M that makes all of the sentences in this knowledge base true.

After choosing to assign C to false, what values are removed by running forward checking? On the table of remaining values below, cross off the values that were removed.

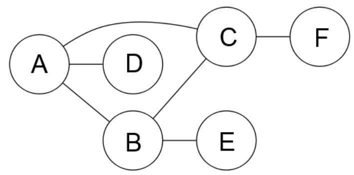
Symbol	Remaining Values
A	F
B	T F
C	F
D	T
E	T F

(c) We eventually arrive at the model $M = \{A = False, B = False, C = True, D = True, E = True\}$ that causes all of the knowledge base sentences to be true. We have a query sentence α specific as $(A \lor C) \Rightarrow E$. Our model M also causes α to be true. Can we say that the knowledge base entails α? Explain briefly (in one sentence) why or why not.

Q2. CSPs

(a) The graph below is a constraint graph for a CSP that has only binary constraints. Initially, no variables have been assigned.

For each of the following scenarios, mark all variables for which the specified filtering might result in their domain being changed.



(i) A value is assigned to A. Which domains might be changed as a result of running forward checking for A?

○ A ○ B ○ C ○ D ○ E ○ F

(ii) A value is assigned to A, and then forward checking is run for A. Then a value is assigned to B. Which domains might be changed as a result of running forward checking for B?

○ A ○ B ○ C ○ D ○ E ○ F

(iii) A value is assigned to A. Which domains might be changed as a result of enforcing arc consistency after this assignment?

○ A ○ B ○ C ○ D ○ E ○ F

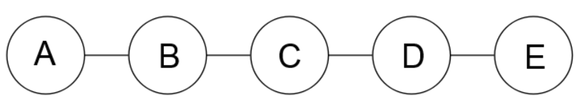
(iv) A value is assigned to A, and then arc consistency is enforced. Then a value is assigned to B. Which domains might be changed as a result of enforcing arc consistency after the assignment to B?

○ A ○ B ○ C ○ D ○ E ○ F

(b) You decide to try a new approach to using arc consistency in which you initially enforce arc consistency, and then enforce arc consistency every time you have assigned an even number of variables.

You have to backtrack if, after a value has been assigned to a variable, X, the recursion returns at X without a solution. Concretely, this means that for a single variable with d values remaining, it is possible to backtrack up to d times. For each of the following constraint graphs, if each variable has a domain of size d, how many times would you have to backtrack in the worst case for each of the specified orderings?

(i)

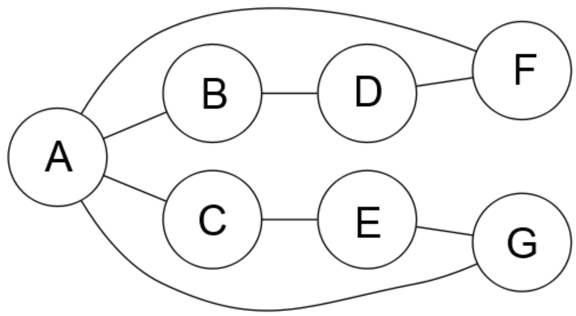


A-B-C-D-E: ______________________

A-E-B-D-C: ______________________

C-B-D-E-A: ______________________

(ii)



A-B-C-D-E-F-G: ____________________

F-D-B-A-C-G-E: ____________________

C-A-F-E-B-G-D: ____________________