CS188 Fall 2015 Section 3: CSPs and Propositional Logic

1 Course Scheduling

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time.

The classes are:

1. Class 1 - Intro to Programming: meets from 8:00-9:00am
2. Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
3. Class 3 - Natural Language Processing: meets from 9:00-10:00am
4. Class 4 - Computer Vision: meets from 9:00-10:00am
5. Class 5 - Machine Learning: meets from 10:30-11:30am

The professors are:

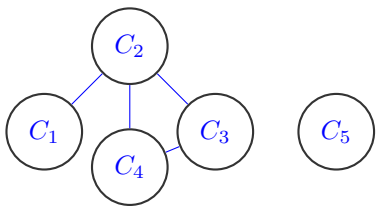
1. Professor A, who is qualified to teach Classes 1, 2, and 5.
2. Professor B, who is qualified to teach Classes 3, 4, and 5.
3. Professor C, who is qualified to teach Classes 1, 3, and 4.

1. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

Variables	Domains (or unary constraints)
C_1	{A, C}
C_2	{A}
C_3	{B, C}
C_4	{B, C}
C_5	{A, B}

   Binary Constraints
   $C_1 \neq C_2$
   $C_2 \neq C_3$
   $C_2 \neq C_4$
   $C_3 \neq C_4$

2. Draw the constraint graph associated with your CSP.



3. Your CSP should look nearly tree-structured (where any two variables are connected by only one path). Briefly explain (one sentence or less) why we might prefer to solve tree-structured CSPs.

   Minimal answer: we can solve them in polynomial time. If a graph is tree structured (i.e. has no loops), then the CSP can be solved in $O(nd^2)$ time as compared to general CSPs, where worst-case time is $O(d^n)$. For tree-structured CSPs you can choose an ordering such that every node's parent precedes it in the ordering. Then after enforcing arc consistency you can greedily assign the nodes in order, starting from the root, and will find a consistent assignment without backtracking.

4. Unfortunately, our CSP is not quite tree-structured. Explain how could still use methods for solving tree-structured CSPs to solve our nearly-tree-structured CSP.

   We can use cutset conditioning. After we remove C_4 from the graph, the remainder is tree-structured. So we can instantiate C_4 with B and C in turn, remove variables from the domains of the neighbors of C_4 that are inconsistent with the assignment made to C_4, and use the tree CSP solver on the remainder.

5. Solve your CSP using the method from question 4. List all the steps which you would perform.

   First we assign $C_4 = B$. This leaves us with the following domains:

Variables	Domains (or unary constraints)
C_1	{A, C}
C_2	{A}
C_3	{C}
C_4	{B}
C_5	{A, B}

   Now we need to ensure pairwise arc consistency.

   First we check that C_2 is arc-consistent with C_3, i.e., that for every value in the domain of C_2, there exists some value in the domain of C_3 such that the binary constraints are satisfied. We can see that they are already arc-consistent. We also check that C_1 is arc-consistent with C_2, and thereby remove A from the domain of C_1.

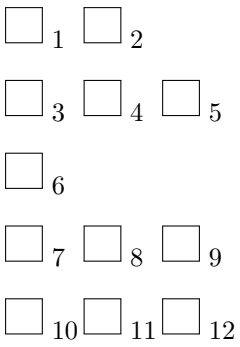
   Now we make consistent assignments to C_1, C_2, and C_3 in order. Since each variable contains only one value in its domain, we have $C_1 = C$, $C_2 = A$, $C_3 = C$, $C_4 = B$, and $C_5 = A$ or B.

   If we had first assigned $C_4 = C$, we would have followed analogous steps.

2 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. 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	$\neg$
2	A B C D E
3	A B C D E $\neg$
4	$\wedge$ $\vee$ $\neg$ $\Rightarrow$ $\Leftrightarrow$
5	A B C D E
6	A B C D E

2. You are given the following assignment as a solution to the knowledge base CSP:

$$\neg\ A$$
$$B \Rightarrow A$$
$$D$$
$$C \vee B$$
$$D \vee E$$

   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

Forward checking removes the value false from the domain of B. Forward checking does not continue on to make any other arcs consistent.

3. 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 $(A \vee C) \Rightarrow E$. 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.

   No, the knowledge base does not entail α. There are other models for which the knowledge base could be true and the query be false. Specifically $\{A = False, B = False, C = True, D = True, E = False\}$ satisfies the knowledge base but causes the query α to be false.

4. Instead of using model-checking for inference, we can also use theorem-proving methods to see whether our knowledge base entails a query sentence. To use these methods, it is useful to convert our knowledge base to *conjunctive normal form* (CNF), which satisfies:

   - The sentence is a conjunction of (one or more) clauses.
   - Each clause is a disjunction of literals.
   - Each literal is a symbol or a negated symbol.

   (a) Which sentences in the knowledge base are not already in conjunctive normal form? Convert them to CNF.

   $B \Rightarrow A$ is converted to $\neg B \vee A$.

   (b) Write the entire knowledge base as a single sentence in CNF.

   After taking the conjunction of all sentences, we get

   $$\neg A \wedge (\neg B \vee A) \wedge D \wedge (C \vee B) \wedge (D \vee E)$$

5. Describe the steps necessary for converting $(A \wedge B) \vee (C \wedge D)$ to CNF.

   $$\begin{aligned} &(A \wedge B) \vee (C \wedge D) \\ &((A \wedge B) \vee C) \wedge ((A \wedge B) \vee D) && \text{distribute } \vee \text{ over } \wedge \\ &((A \vee C) \wedge (B \vee C)) \wedge ((A \vee D) \wedge (B \vee D)) && \text{distribute } \vee \text{ over } \wedge \\ &(A \vee C) \wedge (B \vee C) \wedge (A \vee D) \wedge (B \vee D) && \text{associativity of } \wedge \end{aligned}$$