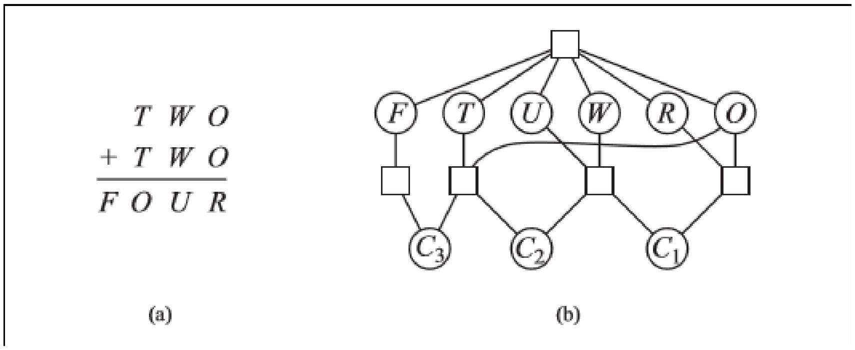
David Lee

Lee2173

2/5/2019

CS 426 HW 2

1. Solve the cryptarithmetic problem in Figure 6.2 by hand, using the strategy of back tracking with forward checking and the MRV and least-constraining-value heuristics



O + O = R + 10 * C1
C1 * W + W = U + 10 * C2
C2 + T + T = O + 10 * C3
F = C3
In addition C1, C2, and C3 can only hold a value of 0 or 1. However, since F cannot hold 0, F = C3 holds F = 1 and C3 = 1.
Substitute for C3 = 1, then
C2 + 2T – 10 = O
We need to assign a value greater than 5 to T so that there exists possible values for O, which isn't negative and also greater than 0. Therefore, with the least constraining heuristic, T = 6.
Plug in T = 6, then O = C2 + 2 we know that C2 carries a 0 or 1 so we set the value of C2 to be 0. This makes O = 2.
Currently, we have F= 1, T = 6, O = 2.
Plugging in, O to the top equation we get
2 + 2 = R + 10*C1
C1 has to be 0 here because R cannot be negative, then R = 4.
Then we know that U = 2W. U can only be an even number which is 0 or 2. However, since 2 is already assigned to O we can backtrack to O and assign O = 3.
We backtrack to have F = 1, T = 6, O = 3. When O = 3, C1 = 0 and then R = 6. But since 6 is already assigned to T, we backtrack to T to equal 7.
Plugging in T = 7 to X2 + 2T = O + 10 * 1, we get O = 4. When O = 4, 2O = R + 10 * C1 Which makes R = 8.
We can continue to check until we get to U, since there are no more backtracks.
If we keep plugging in by using backtracking and forward checking with MRV and lsv heuristics, we get the value of constraints: F = 1, T = 7, O = 4, R = 8, W = 3, and U = 6.

2. Show how a single ternary constraint such as A+B=C can be turned into three binary constraints by using an auxiliary variable. You may assume finite domains. (Hint: Consider a new variable that takes on values that are pairs of other values, and consider constraints such as X is the first element of the pair Y). Show how constraints with more than three variables can be treated similarly.
   I want to introduce a new variable X as a set of pairs to express the ternary constraint in binary. We will assume that X's pairs has a finite domain of numbers.
   Constraints:
   - The value of A must be equal to the first element of the pair X.
   - The value of B must be equal to the second element of the pair X.
   - The sum of the pair of numbers in X must be equal to the value of C.

   With this, we can say that any constraints with more than 3 variables can be treated similar because just like how we reduced to binary by reducing A, B, and C to binary constraints by adding a variable X.

3. Which of the following are correct?
   a. False |= True
      Correct, everything is a model of true.
   b. True |= False
      Incorrect, no model can make true into false.
   c. $(A \land B) \models (A \Leftrightarrow B)$
      Correct, we can see that $A \Leftrightarrow B$ is true when $A \land B$ is true by truth table.
   d. $A \Leftrightarrow B \models A \lor B$
      Incorrect, if A and B are both false $A \Leftrightarrow B$ is true while $A \lor B$ is false.
   e. $A \Leftrightarrow B \models \neg A \lor B$
      Correct, since even when $\neg A \lor B$ has one more true statement than $A \Leftrightarrow B$, everything is a model of true, and since the left side of the entails is false it is still considered correct.

f. $(A \land B) \Longrightarrow C \models (A \Longrightarrow C) \lor (B \Longrightarrow C)$
Correct, truth table.

a	b	c	$((a \land b) \rightarrow c)$
F	F	F	**T**
F	F	T	**T**
F	T	F	**T**
F	T	T	**T**
T	F	F	**T**
T	F	T	**T**
T	T	F	**F**
T	T	T	**T**

a	b	c	$((a \rightarrow c) \lor (b \rightarrow c))$
F	F	F	**T**
F	F	T	**T**
F	T	F	**T**
F	T	T	**T**
T	F	F	**T**
T	F	T	**T**
T	T	F	**F**
T	T	T	**T**

g. $(C \lor (\neg A \land \neg B)) = ((A \Longrightarrow C) \land (B \Longrightarrow C))$
Correct, truth table.

a	b	c	$(c \land (\neg a \land \neg b))$
F	F	F	**F**
F	F	T	**T**
F	T	F	**F**
F	T	T	**F**
T	F	F	**F**
T	F	T	**F**
T	T	F	**F**
T	T	T	**F**

a	b	c	$((a \rightarrow c) \land (b \rightarrow c))$
F	F	F	**T**
F	F	T	**T**
F	T	F	**F**
F	T	T	**T**
T	F	F	**F**
T	F	T	**T**
T	T	F	**F**
T	T	T	**T**

h. $(A \lor B) \land (C \lor \neg D \lor E) \models (A \lor B)$
Correct, for the left to be true, A or B needs to be true on the left, which makes A or B true on the right as well.

i. $(A \lor B) \land (\neg C \lor \neg D \lor E) \models (A \lor B) \land (\neg D \lor E)$
Incorrect, if we have A or B is true, C is false, D is true, E is false, then the left side will be true but not the right.

j. (A ∨ B) ∧ ¬(A ⟹ B) is satisfiable
(A ∨ B)) ∧ ¬(¬A ∨ B) = (A ∨ B)) ∧ (A ∧ ¬B)
Correct, when A is true and B is false.

k. (A ⟸⟹ B) ∧ (¬A ∨ B) is satisfiable
(¬A ∨ B) ∧ (¬A ∨ B)
Correct, when A is false and B is true.

l. (A ⟸⟹ B) ⟸⟹ C has the same number of models as (A ⟸⟹ B) for any fixed set of proposition symbols that includes A, B, C
Correct, the number of models will double for every symbol added, but it doesn't change the fact that the first condition has to be make a true model in either way.

4. According to some, a person who likes cats (C), likes dogs (D) if he/she likes all animals (A), but otherwise does not. Which of the following are correct representations of this assertion?
   a. (C ∧ D) ⟸⟹ A

A	C	D	((C ∧ D) ↔ A)
F	F	F	**T**
F	F	T	**T**
F	T	F	**T**
F	T	T	**F**
T	F	F	**F**
T	F	T	**F**
T	T	F	**F**
T	T	T	**T**

When C is true and D is true, A is true.
When C is true and D is false, A is false.
correct.

b. C ==⇒ (D ⇐⇒ A)

A	C	D	(C → (D ↔ A))
F	F	F	**T**
F	F	T	**T**
F	T	F	**T**
F	T	T	**F**
T	F	F	**T**
T	F	T	**T**
T	T	F	**F**
T	T	T	**T**

incorrect. Someone who likes all animals, cannot dislike cats and dogs.

c. C ==⇒ ((A ==⇒ D) ∨ ¬D)

A	C	D	(C → ((A → D) ∨ ¬D))
F	F	F	**T**
F	F	T	**T**
F	T	F	**T**
F	T	T	**T**
T	F	F	**T**
T	F	T	**T**
T	T	F	**T**
T	T	T	**T**

incorrect. Someone who likes all animals, cannot dislike cats and dogs.

5. This question considers representing satisfiability (SAT) problems as CSPs.
   a. Draw the constraint graph corresponding to the SAT problem $(\neg X1 \vee X2) \wedge (\neg X2 \vee X3) \wedge \ldots \wedge (\neg Xn-1 \vee Xn)$ for n = 5
      x1 - x2 - x3 - x4 – x5 will be the constraint graph since the problem's relationship is grouped to the xi for xi-1 and xi+1 of the number.
   b. How many solutions are there for this general SAT problem as a function of n?
      There should be n+1 solutions, since all the nodes the following xi would be true if xi is true.

c. Suppose we apply BACKTRACKING-SEARCH (page 215) to find all solutions to a SAT CSP of the type given in (a). (To find all solutions to a CSP, we simply modify the basic algorithm so it continues searching after each solution is found.) Assume that variable are ordered X1, …, Xn and false is ordered before true. How much time will the algorithm take to terminate? (Write an O() expression as a function of n.)
I believe it would be O(n^2) because if we have a loop from X1 to Xn, we need to have another loop to make sure the rest of the trailing nodes of the graph are also true.