Artificial Intelligence 2007 Spring
Homework 3 Solution

April 15, 2007

1. **Constrain Satisfaction Problem**

   We have only four variables: A, B, C and D and each of them have only two legal values, which we will write as: A1, A2 (for variable A), B1, B2 (for variable B), C1, C2 (for variable C) and D1, D2 (for variable D). The **only** legal assignments for each pair of variables are:

   A-B: A1-B1, A2-B1

   A-C: A1-C1, A2-C2

   B-D: B1-D1

   C-D: C2-D1

   B-C: No constraint.

   A-D: No constraint.

   No other combination of variable values is legal. Let's say that that "an assignment is generated" every time a variable in the problem gets a new (tentative) assignment. We assume that ***the variables are examined in alphabetical order and the values in numerical order***. Below, we ask you to solve this problem using pure backtracking and also by using backtracking with forward checking. Stop when a valid solution is found.

   The search tree for this problem is given below. Each node (except the root) is labeled with the value involved in the assignment, the variable involved is obvious given the value. Your answers will be a space-separated sequence of these values involved in the assignments as they are generated during the appropriate search. For example, A1 B1 etc.

   (a) Pure backtracking: How many total assignments are made before finding an answer?

   (b) Pure backtracking: Show the assignments in order.

   (c) Backtracking with forward checking: How many assignments are made before finding an answer?

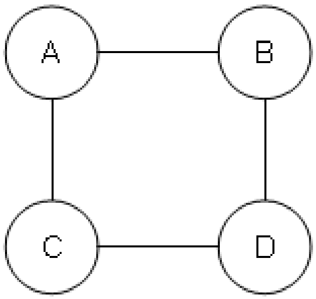
   (d) Backtracking with forward checking: Show the assignments in order.

   Ans.

   To formulate this problem as a CSP. We can represent it using a graph as Figure 1:

   (a),(b) The steps that examine these variables in alphabetical order and assign values in numerical order to these variables using pure backtracking are in Table 1:

   (c),(d) Using backtracking with forward checking can find illegal states earlier than pure backtracking. The steps are shown in Table 2.



Constraints:

A-B: A1-B1, A2-B1

A-C: A1-C1, A2-C2

B-D: B1-D1

C-D: C2-D1

A, B, C, D ∈{1,2}

Figure 1: Formulate this problem as a CSP.

step1	A1				
step2	A1	B1			
step3	A1	B1	C1		
step4	A1	B1	C1	D1	×
step5	A1	B1	C1	D2	×
step6	A1	B1	C2	×	
step7	A1	B2	×		
step8	A2				
step9	A2	B1			
step10	A2	B1	C1	×	
step11	A2	B1	C2		
step12	A2	B1	C2	D1	DONE!

Table 1: Assign values using pure backtracking strategy

						A	B	C	D
init						1,2	1,2	1,2	1,2
step1	A1					1	1	1	1,2
step2	A1	B1				1	1	1	1
step3	A1	B1	C1	×		1	1	1	
step4	A2					2	1	2	1,2
step5	A2	B1				2	1	2	1
step6	A2	B1	C2			2	1	2	
step7	A2	B1	C2	D1	DONE!	2	1	2	1

Table 2: Assign values using backtracking with forward checking