Assignment-2 Artificial Intelligence

S Nishok Kumar 15CS30024

Q1) Consider assigning colors to a checkerboard so that squares that are adjacent vertically or horizontally do not have the same color. We know that this can be done with only two colors, say red (R) and black (B). We will limit our discussion to five squares on a 3x3 board, numbered as follows: 1 | 2 | 3

---

4 | 5 |

| |

We consider the CSP formulation of this problem. Let the squares be the variables and the colors be the values. All the variables have domains { R, B }.

(a) If we run full constraint propagation on the initial state, what are the resulting domains of the variables?

Ans: None of the variable domains change:

1 = {R, B} ; 2 = {R, B} ; 3 = {R, B} ; 4 = {R, B} ; 5 = {R, B}

(b) Say, instead, the initial domain of variable 5 is restricted to { B }, with the other domains as before. If we now run full constraint propagation, what are the resulting domains of the variables?

Ans: The resulting domains of the variables are

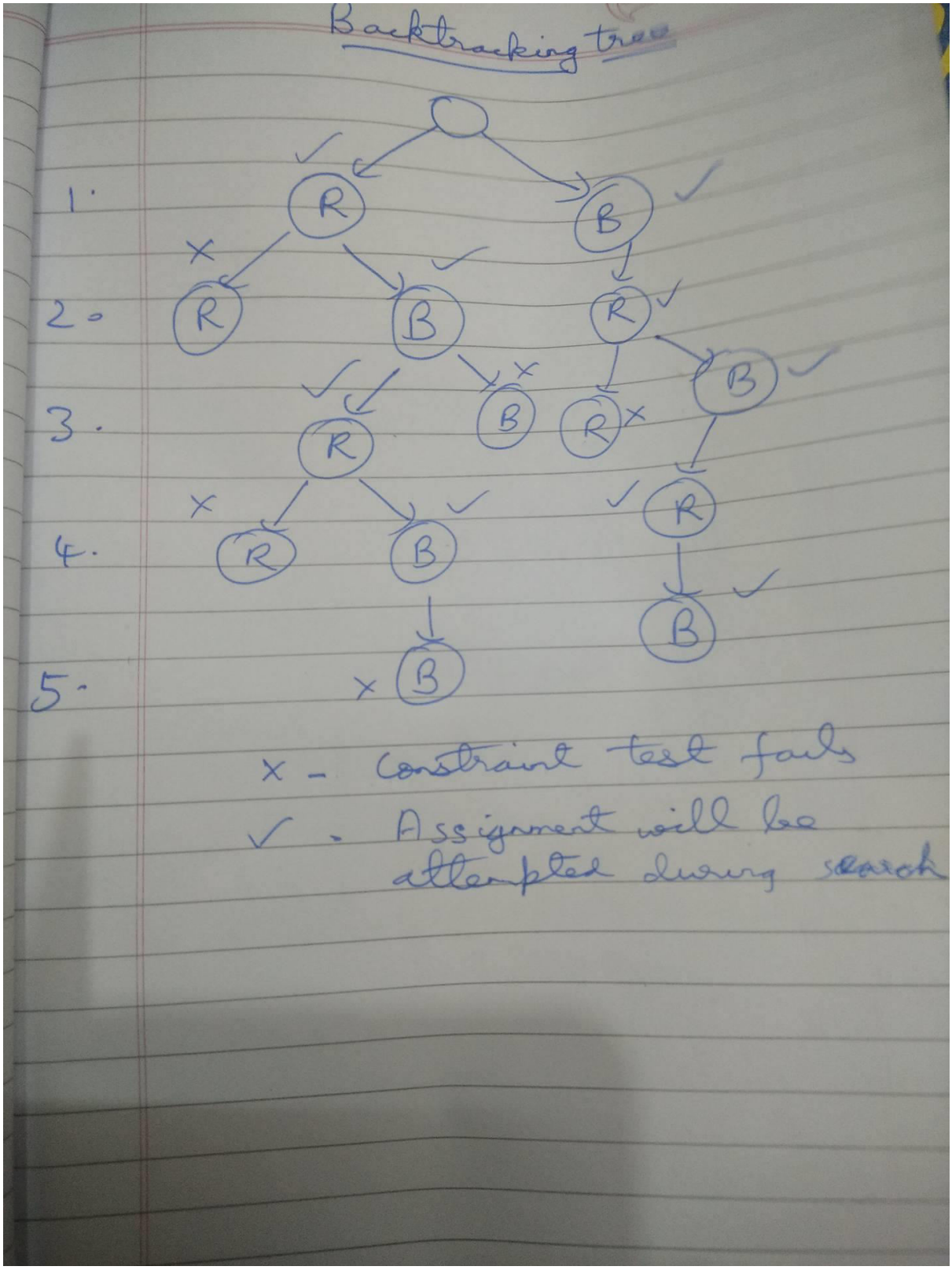
1 = {B}; 2 = {R} ; 3 = {B} ; 4 = {R} ; 5 = {B}

(c) If in the initial state (all variables have domains { R, B }), we assign variable 1 to R and do forward checking, what are the resulting domains of the other variables?

Ans: Forward checking is defined as a single iteration of constraint propagation only on those edges that terminate at the variable whose value was just set, and that do not originate from variables which have already been set. Therefore, after we set 1 = R, forward checking affects the domains of variables 2 and 4 since they are adjacent to variable 1 (and have not yet been assigned). 1 = {R} 2 = {B} 3 = {R, B} 4 = {B} 5 = {R, B} Forward checking only does one step of propagation, only to the immediate neighbors of the assigned variable.

(d) Assume that during backtracking we first attempt assigning variables to R and then to B. Assume, also, that we examine the variables in numerical order, starting with 1. Also, let the domain of variable 5 be { B }, the other domains are { R, B }. Draw the pure backtracking tree that will be generated.

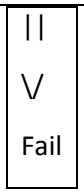
Ans:



(e) If we use backtracking with forward checking in this same situation, give a list of all the assignments attempted, in sequence. Use the notation variable = color for assignments, for example, 1=R.

Ans: We must keep track of the variable domains as we search since forward checking modifies these domains based on the current assignment, and we will need to restore the domain of earlier search nodes if we have to backtrack to them. We fail at a node if (1) the current assignments violate some constraint, or (2) if forward checking after the present assignment causes the domain of some variable to become empty. The following lists (in order from left to right) each attempted assignments and the resulting variable domains after forward checking.

Assignment	None	1 = R	2 = B	1 = B	2 = R	3 = B	4 = R	5 = B
Domain of 1	{R, B}	R	R	B	B	B	B	B
Domain of 2	{R, B}	{B}	B	{R}	R	R	R	R
Domain of 3	{R, B}	{R, B}	{R}	{R,B}	{B}	B	B	B
Domain of 4	{R, B}	{B}	{B}	{R}	{R}	{R}	R	R
Domain of 5	{B}	{B}	{}	{B}	{B}	{B}	{B}	B



Note that when we fail at 2 = B, since there are no further values to try in the 4 domain of variable 2, we backtrack to the assignment of variable 1. When this happens, we restore the domains from before variable 1 was assigned, i.e. the ones listed above under “None”.

Q2) Explain why it is a good heuristic to choose the variable that is most constrained but the value that is least constraining in a CSP search.

Ans: The most constrained variable is the variable that is most likely to get constrained to no possibilities and force backtracking. The least constrained value is the value that leaves open the maximum possibilities in the subtrees (and so the least chance of backtracking). Hence it is a good heuristic to choose the variable that is most constrained but the value that is least constraining in a CSP search.

Q3) Consider the CSP with Variables = A,B, C Domains = {1,2,3,4} Constraints = A<B, B<C. Using the Arc Consistency Algorithm, determine the new domains for A,B and C when the algorithm terminates with all consistent arcs. Show detailed steps.

Ans:

dom(A) = {1, 2, 3, 4}; dom(B) = {1, 2, 3, 4}; dom(C) = {1, 2, 3, 4}

1)Suppose you first select the arc {A, A < B}.

a) Remove A = 4 from the domain of A.
b) Add nothing to TDA.

2)Suppose that {B, B < C} is selected next.

a) Prune the value 4 from the domain of B.
b) Add {A, A < B} back into the TDA set.

3)Suppose that {B, A < B} is selected next.

a) Prune 1 from the domain of B.

b) Add no element to TDA.

4)Suppose the arc {A, A < B} is selected next.

a) The value A = 3 can be pruned from the domain of A.

b) Add no element to TDA.

5)Select {C, B < C} next.

a) Remove 1 and 2 from the domain of C.

b) Add {B, B < C} back into the TDA set .

The other two edges are arc consistent, so the algorithm terminates with dom(A) = {1, 2}, dom(B) = {2, 3}, dom(C) = {3, 4}.

Q4) Give precise formulations for each of the following as constraint satisfaction problems:

(a) 8 queens problem: The eight queens puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other. Thus, a solution requires that no two queens share the same row, column, or diagonal

Ans: Let us have 8 variables Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8. Qi contain the value of the row in which queen Qi is placed. Subscript i corresponds to the column in which queen Qi is placed. 8 variables correspond to columns 1 – 8. Therefore the set of values for each of these variables will be {1,2,3,4,5,6,7,8}.

Constraint is formulated as Qi cannot attack Qj (i!=j) ie Qi is a queen to be placed in column i,Qj is a queen to be placed in column j. The value of Qi and Qj are the rows in which the queens are to be placed in.

This can be represented as follows:

Between every pair of variables (Qi,Qj) (i!=j) we have a constraint Cij.

For each Cij, an assignment of values to the variables Qi = A and Qj = B satisties this constraint iff

1)A != B

2)| A – B | != | i – j |

(b) Class scheduling: There is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes that he or she can teach.

Ans: The four variables in this problem are: Professors, Subjects, Classrooms and Time slots.

We can use two constraint matrices, Pij and Sij . Pij represents a professor in classroom i at time j. Sij represents a class being taught in classroom i at time j. The domain of each Pij variable is the set of professors. The domain of each Sij variable is the set of classes offered. Let's denote by D(t) the set of classes that professor named t can teach.

The constraints are: Pij != Pkj ,k != i which enforces that no professor is assigned to two classes which take place at the same time. There is a constraint between every Sij and Pij , denoted Cij (t, s) that ensured that if professor t is assigned to Pij , then Sij is assigned a value from D(t).

An example for the constraint C is C(Pij , Sij) = {(Dick, 6a),(Dick, 171),(Dick, 175a),(Sam, 171),(Sam, 278),(Irani, 6a), . . .}

In general C (Pij , Sij) = {(t, s)| professor t can teach class s}