

ICS 270A: Solutions to Homework 3

1. Give a precise formulation of the following constraint satisfaction problems:

- (a) 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.

Answer:

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

We can use two constraint matrices, T_{ij} and S_{ij} . T_{ij} represents a teacher in classroom i at time j . S_{ij} represents a subject being taught in classroom i at time j . The domain of each T_{ij} variable is the set of teachers. The domain of each S_{ij} variable is the set of subjects. Let's denote by $D(t)$ the set of subjects that teacher named t can teach.

The constraints are:

$$T_{ij} \neq T_{kj} \quad k \neq i$$

which enforces that no teacher is assigned to two classes which take place at the same time. There is a constraint between every S_{ij} and T_{ij} , denoted $C_{ij}(t, s)$ that ensured that if teacher t is assigned to T_{ij} , then S_{ij} is assigned a value from $D(t)$. An example for the constraint C is

$$C(T_{ij}, S_{ij}) = \{(\text{Dechter}, 6a), (\text{Dechter}, 171), (\text{Dechter}, 175a), (\text{Smyth}, 171), (\text{Smyth}, 278), (\text{Irani}, 6a), \dots\}$$

In general $C(T_{ij}, S_{ij}) = \{(t, s) \mid \text{teacher } t \text{ can teach subject } s\}$

- (b) The rectilinear floor-planning problem: Find non-overlapping places in a large rectangle for a number of small rectangles.

Answer:

Assume the large rectangle has width \mathcal{W} and height \mathcal{H} . Each rectangle R_i is parameterized by four variables, x, y, w, h , which define its position, width and height. All smaller rectangles have the set of constraints

$$\begin{aligned} R_{i,x} &\geq 0 \\ R_{i,x} + R_{i,w} &\leq \mathcal{W} \\ R_{i,y} &\geq 0 \\ R_{i,y} + R_{i,h} &\leq \mathcal{H} \end{aligned}$$

which confines each rectangle to the interior of the large rectangle. In addition, there is a set of constraints, C_{ij} , between rectangles i and j , $i \neq j$, which are

$$\begin{aligned} R_{i,x} + R_{i,w} \leq R_{j,x} \quad \text{or} \quad R_{i,x} \geq R_{j,x} + R_{j,w} \\ R_{i,y} + R_{i,h} \leq R_{j,y} \quad \text{or} \quad R_{i,y} \geq R_{j,y} + R_{j,h} \end{aligned}$$

which enforces the constraint that no two rectangles overlap each other.

2. Consider the following binary-constraint network: There are 4 variables: X_1, X_2, X_3, X_4 , with the domains:

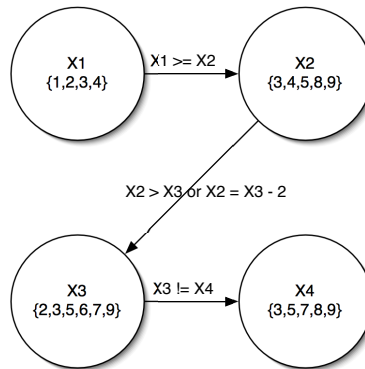
$$D_1 = \{1, 2, 3, 4\}, D_2 = \{3, 4, 5, 8, 9\}, D_3 = \{2, 3, 5, 6, 7, 9\}, D_4 = \{3, 5, 7, 8, 9\}.$$

The constraints are: $X_1 \geq X_2$, $X_2 > X_3$ or $X_3 - X_2 = 2$, $X_3 \neq X_4$.

- (a) Write the constraints in a relational form and draw the constraint graph.

Answer:

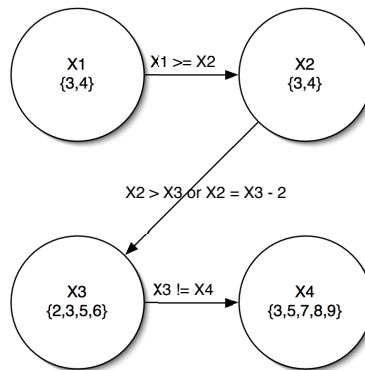
Relation	Domain
$X_1 \geq X_2$	(3,3), (4,3), (4,4)
$X_2 > X_3$ or $X_3 - X_2 = 2$	(3,2), (4,2), (4,3), (5,2), (5,3), (8,2), (8,3), (8,5), (8,6) (8,7), (9,2), (9,3), (9,5), (9,6), (9,7), (3,5), (4,6), (5,7)
$X_3 \neq X_4$	(2,3), (2,5), (2,7), (2,8), (2,9), (3,5), (3,7), (3,8), (3,9) (5,3), (5,7), (5,8), (5,9), (6,3), (6,5), (6,7), (6,8), (6,9) (7,3), (7,5), (7,8), (7,9), (9,3), (9,5), (9,7), (9,8)



- (b) Is the network arc-consistent? If not, compute the arc-consistent network.

Answer:

No, it is not arc consistent. The arc-consistent constraint network is:



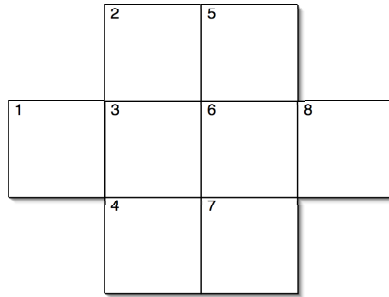
(c) Is the network consistent ? If yes, give a solution.

Answer:

Yes, it is consistent. A solution is:

$$X_1 = 3, X_2 = 3, X_3 = 2, X_4 = 3$$

3. The task is to label the boxes with the numbers 1-8 such that the labels of any pair of adjacent squares (i.e. horizontal vertical or diagonal) differ by at least 2 (i.e. 2 or more).

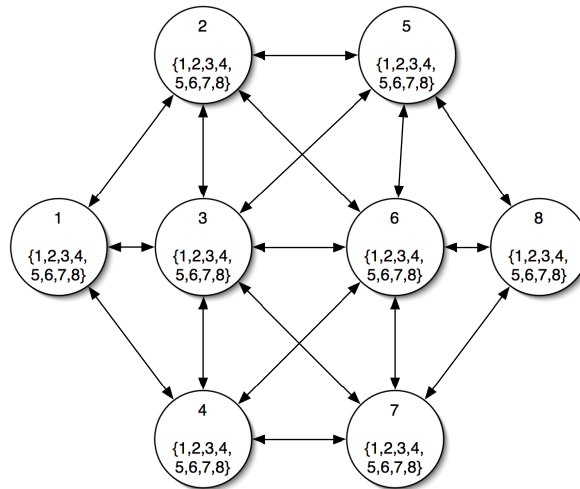


(a) Write the constraints in a relational form and draw the constraint graph.

Answer:

Given the identifying numbers in the above diagram, for any neighboring pair of squares i and j such that $i \neq j$,

Relation	Domain
$ X_i - X_j \geq 2$	(1,3), (1,4), (1,5), (1,6), (1,7), (1,8) (2,4), (2,5), (2,6), (2,7), (2,8) (3,1), (3,5), (3,6), (3,7), (3,8) (4,1), (4,2), (4,6), (4,7), (4,8) (5,1), (5,2), (5,3), (5,7), (5,8) (6,1), (6,2), (6,3), (6,4), (6,8) (7,1), (7,2), (7,3), (7,4), (7,5) (8,1), (8,2), (8,3), (8,4), (8,5), (8,6)



(b) Is the network arc-consistent ? if not, compute the arc-consistent network.

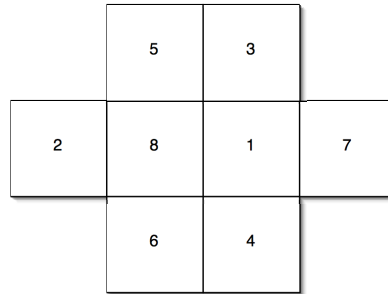
Answer:

Yes, it is arc-consistent.

(c) Is the network consistent ? If yes, give a solution.

Answer:

Yes, a solution is given by

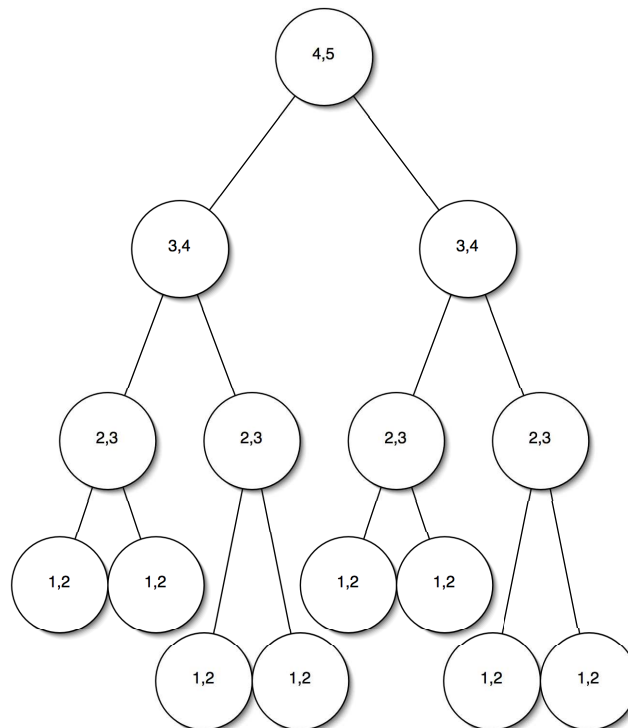


4. Consider a tree constraint problem, with 15 variables $X_1, \dots, X_{15}, D_1, \dots, D_{15} = \{1, 2, 3, 4, 5\}$, the constraint graph is a perfect binary tree with degree 2, and the binary constraints C_{ij} are $C_{ij} = X_i > X_j$ where i is the parent of j in the tree (The root is X_1).

(a) Enforce arc-consistency.

Answer:

The following constraint graph is arc-consistent,



(b) Is the network consistent ? If yes, compute a solution.

Answer:

Yes, it is consistent. A solution is:

$$\begin{array}{cccc} X_1 = 5 & X_2 = 4 & X_3 = 4 & X_4 = 3 \\ X_5 = 3 & X_6 = 3 & X_7 = 3 & X_8 = 1 \\ X_9 = 1 & X_{10} = 2 & X_{11} = 1 & X_{12} = 2 \\ X_{13} = 1 & X_{14} = 1 & X_{15} = 2 & \end{array}$$

(c) Can you suggest an ordering for which backtrack search will be backtrack-free?

Answer:

Do an in-order traversal: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

(d) Bound the complexity of solving any problem whose constraint graph is the same as for this problem.

Answer:

Since this is a tree the complexity is $O(nd^2)$ when n is the number of variables and d is the domain size.

5. Solve the cryptarithmic problem in Figure 5.2 by hand, using backtracking, Forward checking and MRV and least constraining value heuristics.

Answer:

Given the problem

$$\begin{array}{rcccc} & & T & W & O \\ + & & T & W & O \\ \hline F & O & U & R & \end{array}$$

and the constraints

$$\begin{array}{rcl} O + O & = & R + 10 \cdot X_1 \\ X_1 + W + W & = & U + 10 \cdot X_2 \\ X_2 + T + T & = & O + 10 \cdot X_3 \\ X_3 & = & F \end{array}$$

along with the Alldiff(F, O, U, R, T, W) constraint and no leading zeros being allowed in any of the quantities. We then have the domains

$$\begin{array}{rcl} D_F, D_{X_3} & = & \{1\} \\ D_{X_2}, D_{X_1} & = & \{0, 1\} \\ D_T & = & \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ D_W, D_O, D_U, D_R & = & \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \end{array}$$

A trace leading to a solution is shown below. Some state expansions were skipped when the MRV heuristic would have chosen a state with only a single value in its domain.

Node	State ($F, O, U, R, T, W, X_1, X_2, X_3$)	Action
0	(-, -, -, -, -, -, -, -, -)	Select F next via MRV.
1	(1, -, -, -, -, -, -, -, -)	Assign $F = 1$. Remove 1 from all domains by forward checking. Select X_3 next via MRV.
2	$D_{X_2}, D_{X_1} = \{0, 1\}$ $D_T = \{2, 3, 4, 5, 6, 7, 8, 9\}$ $D_W, D_O, D_U, D_R = \{0, 2, 3, 4, 5, 6, 7, 8, 9\}$ (1, -, -, -, -, -, -, -, 1)	Assign $X_3 = 1$. Remove $\{2, 3, 4\}$ from D_T by forward checking. Select X_2 using MRV.
3	$D_{X_2}, D_{X_1} = \{0, 1\}$ $D_T = \{5, 6, 7, 8, 9\}$ $D_W, D_O, D_U, D_R = \{0, 2, 3, 4, 5, 6, 7, 8, 9\}$ (1, -, -, -, -, -, -, 0, 1)	Assign $X_2 = 0$ using LCV. Remove $\{3, 5, 7, 9\}$ from D_O and $\{5, 6, 7, 8, 9\}$ from D_W by forward checking. Select X_1 using MRV.
4	$D_{X_1} = \{0, 1\}; D_T = \{5, 6, 7, 8, 9\}$ $D_W = \{0, 2, 3, 4\}; D_O = \{0, 2, 4, 6, 8\}$ $D_U, D_R = \{0, 2, 3, 4, 5, 6, 7, 8, 9\}$ (1, -, -, -, -, -, 0, 0, 1)	Assign $X_1 = 0$ using LCV. Remove $\{6, 8\}$ from D_O and odds from D_U by forward checking. Select D_O using MRV.
5	$D_T = \{5, 6, 7, 8, 9\}; D_W = \{0, 2, 3, 4\}$ $D_U = \{0, 2, 4, 6, 8\}; D_O = \{0, 2, 4\}$ $D_R = \{0, 2, 3, 4, 5, 6, 7, 8, 9\}$ (1, 2, -, -, -, -, 0, 0, 1)	Assign $O = 2$ using LCV. Remove 2 from all domains and set $D_T = \{6\}$ and $D_R = \{4\}$. Select W by MRV.
6	$D_T = \{6\}; D_O = \{2\}; D_R = \{4\}$ $D_U = \{0, 4, 6, 8\}; D_W = \{0, 3, 4\}$ (1, 2, -, 4, 7, -, 0, 0, 1)	Assign $W = 3$ by LCV. No Solution. Backtrack to node 3
7	$D_U = \{0, 4, 8\}; D_W = \{0\}$ (1, -, -, -, -, -, 1, 0, 1)	Assign $X_1 = 1$. Remove $\{0, 2, 4\}$ from D_O and evens from D_U by forward checking. Select D_O using MRV.
8	$D_T = \{5, 6, 7, 8, 9\}; D_W = \{0, 2, 3, 4\}$ $D_U = \{3, 5, 7, 9\}; D_O = \{6, 8\}$ $D_R = \{0, 2, 3, 4, 5, 6, 7, 8, 9\}$ (1, 6, -, -, -, -, 1, 0, 1)	Assign $O = 6$. Remove 6 from all domains and set $D_T = 8$ and $D_R = 2$. Pick W next.
9	$D_T = \{8\}; D_W = \{0, 2, 3, 4\}$ $D_U = \{3, 5, 7, 9\}; D_O = \{6\}$ $D_R = \{2\}$ (1, 6, 7, 2, 8, 3, 0, 0, 1) $D_U = \{3, 5, 7, 9\}; D_W = \{0, 3, 4\}$	Assign $W = 3$ by LCV. Found Solution!

A solution is:

$$\begin{array}{r}
 8 \ 3 \ 6 \\
 + \ 8 \ 3 \ 6 \\
 \hline
 1 \ 6 \ 7 \ 2
 \end{array}$$

6. Show how a single ternary constraint such as $A+B = C$ can be turned into 3 binary constraints by using auxiliary variables. You may assume finite domains. Next show how constraints with more than 3 variables can be treated similarly. Finally show how unary domains can be eliminated by altering the domains of variables. This completes the demonstration that any CSP can be transformed into a CSP with binary constraints only.

Answer:

Create a new constraint variable c whose domain is the set of all possible 3-tuples satisfying $A+B=C$. Now define a new set of constraints.

$$\begin{aligned}
 c[A] &\in A \\
 c[B] &\in B \\
 c[C] &\in C
 \end{aligned}$$

which says that a binary constraint is satisfied only if the tuple t in the domain of c is compatible with a value x in the domain of A (for the first constraint) such that $t[A] = x$.

Any constraint with n variables is reduced to a collection of n binary constraints between a new constraint variable and the constitutive variables.

Any unary constraint can be eliminated by removing any values from the variables domain that are inconsistent with the unary constraint. For instance, if the domain of Y is $\{1, 2, 3, 4, 5, 6, 7\}$ and we have the unary constraint,

$$Y \geq 4$$

then we alter the domain of Y to be $\{4, 5, 6, 7\}$ and the unary constraint can be eliminated.

7. Use the min-conflict (local search) method to solve the Four-Queens problem. Start with the queens on a main diagonal. Break ties randomly.

Answer:

A sequence of moves leading to the solution is shown below.

