CS486: Artificial Intelligence

Homework 2 (15 pts)
Constrained Search
Due 14 Sep @ 1630

Instructions

This is an individual assignment; however, *you may receive assistance and/or collaborate without penalty, so long as you properly document such assistance and/or collaboration in accordance with DAW.*

Answer the questions below and submit a hardcopy with DAW coversheet and acknowledgment statement to your instructor by the due date.

Problem 1: AC-3 Algorithm

Consider the following CSP:

$$
\begin{aligned}
&\text{Variables :}\\
&V = \{A, B, C, D\}\\
\\
&\text{Domains :}\\
&\forall x\ D_x = \{0, 1, \ldots, 9\}\\
\\
&\text{Constraints :}\\
&\langle (A),\ A \neq v\ \forall v \in \{0, 2, 3, 4, 6, 7, 8, 9\}\rangle\\
&\langle (B),\ B \neq v\ \forall v \in \{1, 3, 5, 7, 8, 9\}\rangle\\
&\langle (C),\ C \neq v\ \forall v \in \{0, 1, 3, 4, 5, 6, 7\}\rangle\\
&\langle (D),\ D \neq v\ \forall v \in \{0, 2, 3, 4, 5, 6, 7, 9\}\rangle\\
&\langle (A, B),\ A < B\rangle\\
&\langle (A, C),\ C - A \leq 3\rangle\\
&\langle (A, D),\ A \neq D\rangle\\
&\langle (C, D),\ C > D\rangle
\end{aligned}
$$

a. List the domains for each variable after applying all unary constraints.

$D_A = \{1, 5\},\ D_B = \{0, 2, 4, 6\},\ D_C = \{2, 8, 9\},\ D_D = \{1, 8\}$

b. The table below represents a *queue* containing all of the initial arcs for the CSP constraint graph as (arc-tail, arc-head) pairs. The top of the table is the head of the queue, and the bottom is the tail. Suppose the AC-3 algorithm were enforcing consistency of these arcs in the order received from the queue. If an arc is inconsistent, indicate which variable's domain will lose which values (e.g.: "remove 3 and 7 from the domain of D") and add any arcs that will need to be re-checked to the tail of the queue in alphabetical order. If an arc is consistent, leave the rightmost column blank. *Hint: the number of blank rows in the table is the number of arcs that will need to be re-checked.*

(A, B)	
(A, C)	
(A, D)	
(B, A)	remove 0 from D_B[1]
(C, A)	remove 9 from D_C, add (D, C) to queue[2]
(C, D)	
(D, A)	
(D, C)	remove 8 from D_D, add (A, D) to queue
(D, C)	
(A, D)	remove 1 from D_A, add (B, A) and (C, A) to queue
(B, A)	remove 2 and 4 from D_B
(C, A)	

c. List the domain values remaining for each variable after AC-3 is done.

Problem 2: Backtracking Search

For the CSP in Problem 1, show the variable assignments made by Backtracking Search using MRV and LCV. For MRV, use the "degree heuristic" to break ties (i.e., choose to assign the variable with the largest number of neighbors). If there is no value that can be assigned to a variable at a step, write "backtrack". The following is an (incorrect) example of what your answer might look like

[1] The reverse arc (A,B) is *not* added to the queue by the AC-3 algorithm. This may not be completely clear from the slides and video lectures, but several authoritative sources will support this.

[2] Because we're using a queue, (D, C) ends up in the queue twice. A better choice would be to use a set, but I wanted to enforce an ordering.

if you detected no options at B after assigning $D = 1, C = 2, A = 5$ (in that order) and had to try a different value at A to proceed at B:

$$\begin{aligned}
&D \to 1 \\
&D \to 1,\ C \to 2 \\
&D \to 1,\ C \to 2,\ A \to 5 \\
&D \to 1,\ C \to 2,\ A \to 5,\ B \to \text{backtrack} \\
&D \to 1,\ C \to 2,\ A \to 1,\ B \to 2
\end{aligned}$$

MRV and LCV can (and should, for this problem) be used *together*. Neither forward checking, nor maintaining arc consistency was specified, so they should not have been used. This might have been confusing since, to determine the LCV, you do actually have to do a "what-if" check that looks like forward checking, but *does not change any of the domains*. In practice, if you have to do that work for LCV, you might as well implement forward checking, too, but here it was not specified to do so.

$A \to 1$ [3]
$A \to 1, D \to 8$ [4]
$A \to 1, D \to 8, C \to$ backtrack [5]
$A \to 1, D \to$ backtrack
$A \to 5$
$A \to 5, D \to 1$ [6]
$A \to 5, D \to 1, C \to 2$
$A \to 5, D \to 1, C \to 2, B \to 6$

What if we'd have chosen 5 for A at step 1?
$A \to 5$
$A \to 5, D \to 1$ [7]
$A \to 5, D \to 1, C \to 2$
$A \to 5, D \to 1, C \to 2, B \to 6$

[3] Or 5. A and D both have 2 values remaining by MRV, but by the degree heuristic, A is chosen. For value selection, selecting 1 will result in $D_B = \{2, 4, 6\}, D_C = \{2\}, D_D = \{1\}$. On the other hand, selecting 5 will result in $D_B = \{6\}, D_C = \{2, 8\}, D_D = \{1, 8\}$. Which is least constraining? Well, that's a great question, and we didn't talk about it in class. Both eliminate four possibilities and leave five possibilities remaining, across all their neighbors, but choosing 1 reduces the domains of C and D to one value, whereas choosing 5 only reduces the domain of B to a single value . . . . Let's see which works better.

[4] D has MRV. Backtracking search doesn't try $D \to 1$ since that's known to be invalid if $A \to 1$.

[5] C has MRV, but none of them satisfy the constraints, given current assignments.

[6] 1 or 8 would work, but 1 is clearly least constraining here.

[7] By MRV and LCV.

No backtracking required! So, maybe defining the "least constraining value" requires a bit of thought . . . .