



# HOMEWORK

UNIVERSITY OF SOUTH CAROLINA

COMPUTER SCIENCE AND ENGINEERING

---

## CSCE 580: Artificial Intelligence CSPs

---

*Author:*

Your Name (ID: Your USC ID)

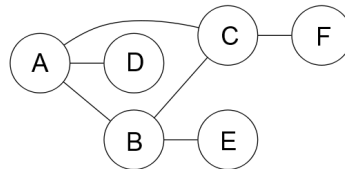
Release Date: February 18, 2019

**Due Date: Friday Feb 15, 11:59 pm**

---

**CPSs (1)** The graph below is a constraint graph for a CSP that has only binary constraints. Initially, no variables have been assigned.

For each of the following scenarios, mark all variables for which the specified filtering might result in their domain being changed.



1. A value is assigned to A. Which domains might be changed as a result of running forward checking for A? **(10 points)**

**B&C&D.** Forward checking for A only considers arcs where A is the head. This includes  $B \rightarrow A$ ,  $C \rightarrow A$ ,  $D \rightarrow A$ . Enforcing these arcs can change the domains of the tails.

2. A value is assigned to A, and then forward checking is run for A. Then a value is assigned to B. Which domains might be changed as a result of running forward checking for B? **(10 points)**

**C&E.** Similar to the previous part, forward checking for B enforces the arcs  $A \rightarrow B$ ,  $C \rightarrow B$ , and  $E \rightarrow B$ . However, because A has been assigned, and a value is assigned to B, which is consistent with A or else no value would have been assigned, the domain of A will not change.

3. A value is assigned to A. Which domains might be changed as a result of enforcing arc consistency after this assignment? **(10 points)**

**B&C&D&E&F.** Enforcing arc consistency can affect any unassigned variable in the graph that has a path to the assigned variable. This is because a change to the domain of X results in enforcing all arcs where X is the head, so changes propagate through the graph. Note that the only time in which the domain for A changes is if any domain becomes empty, in which case the arc consistency algorithm usually returns immediately and backtracking is required, so it does not really make sense to consider new domains in this case.

4. A value is assigned to A, and then arc consistency is enforced. Then a value is assigned to B. Which domains might be changed as a result of enforcing arc consistency after the assignment to B? **(10 points)**

**C&E&F.** After assigning a value to A, and enforcing arc consistency, future assignments and enforcing arc consistency will not result in a change to A's domain. This means that D's domain won't change because the only arc that might cause a change,  $D \rightarrow A$  will never be enforced.

---

**CPSs (2)** After years of struggling through mazes, Pacman has finally made peace with the ghosts, Blinky, Pinky, Inky, and Clyde, and invited them to live with him and Ms. Pacman. The move has forced Pacman to change the rooming assignments in his house, which has 6 rooms. He has decided to figure out the new assignments with a CSP in which the variables are Pacman (**P**), Ms. Pacman (**M**), Blinky (**B**), Pinky (**K**), Inky (**I**), and Clyde (**C**), the values are which room they will stay in, from 1-6, and the constraints are:

**Table 1: Constraints**

- |  |                             |
|--|-----------------------------|
| i) No two agents can stay in the same room | vi) <b>B</b> is even        |
| ii) $P > 3$                                | vii) <b>I</b> is not 1 or 6 |
| iii) <b>K</b> is less than <b>P</b>        | viii) $ I - C  = 1$         |
| iv) <b>M</b> is either 5 or 6              | ix) $ P - B  = 2$           |
| v) $P > M$                                 |                             |

1. **Unary constraints** On the grid below cross out the values from each domain that are eliminated by enforcing unary constraints. (10 points)

**Table 2**

P	<del>1</del>	<del>2</del>	<del>3</del>	4	5	6
B	<del>1</del>	2	<del>3</del>	4	<del>5</del>	6
C	1	2	3	4	5	6
K	1	2	3	4	5	6
I	<del>1</del>	2	3	4	5	<del>6</del>
M	<del>1</del>	<del>2</del>	<del>3</del>	<del>4</del>	5	6

The unary constraints are ii, iv, vi, and vii. ii crosses out 1,2, and 3 for P. iv crosses out 1,2,3,4 for M. vi crosses out 1,3, and 5 for B. vii crosses out 1 and 6 for I. K and C have no unary constraints, so their domains remain the same.

2. **MRV** According to the Minimum Remaining Value (MRV) heuristic, which variable should be assigned to first? (10 points)

**M** has the fewest value remaining in its domain (2), so it should be selected first for assignment.

3. **Forward Checking** For the purposes of decoupling this problem from your solution to the previous problem, assume we choose to assign P first, and assign it the value 6. What are the resulting domains after enforcing unary constraints (from part i) and running forward checking for this assignment? (15 points)

Table 3

P						6
B	<del>1</del>	2	3	4	<del>5</del>	<del>6</del>
C	1	2	3	4	5	<del>6</del>
K	1	2	3	4	5	<del>6</del>
I	<del>1</del>	2	3	4	5	<del>6</del>
M	<del>1</del>	2	3	<del>4</del>	5	<del>6</del>

In addition to enforcing the unary constraints from part i, the domains are further constrained by all constraints involving P. This includes constraints i, iii, v, and ix. i removes 6 from the domains of all variables. iii removes 6 from the domain of K (already removed by constraint i). v removes 6 from the domain of M (also already removed by i). ix removes 2 and 6 from the domain of B.

4. **Iterative Improvement** Instead of running backtracking search, you decide to start over and run iterative improvement with the min-conflicts heuristic for value selection. Starting with the following assignment:

P:6, B:4, C:3, K:2, I:1, M:5

First, for each variable write down how many constraints it violates in the table below.

Then, in the table on the right, for all variables that could be selected for assignment, put an x in any box that corresponds to a possible value that could be assigned to that variable according to min-conflicts. When marking next values a variable could take on, only mark values different from the current one. (25 points)

Variable	# violated		1	2	3	4	5	6
P	0	P						
B	0	B						
C	1	C		x				
K	0	K						
I	2	I		x		x		
M	0	M						

Both I and C violate constraint viii, because  $|I-C|=2$ . I also violates constraint vii. No other variables violate any constraints. According to iterative improvement, any conflicted variable could be selected for assignment, in this case I

---

and C. According to min-conflicts, the values that those variables can take on are the values that minimize the number of constraints violated by the variable. Assigning 2 or 4 to I causes it to violate constraint i, because other variables already have the values 2 and 4. Assigning 2 to C also only causes C to violate 1 constraint.