CS 340 Home Work # 4 Solution
CLO 2
Total marks: 80

Q 1. (10 Marks) Mom, Dad, Baby, Student, Teacher, and Guide are lining up next to each other in six linear spots labeled 1 to 6, one to a spots. Baby needs to line up between Mom and Dad. Student and Teacher need to be next to each other. Guide needs to be at one end, in spot 1 or 6. Formulate this problem as a CSP: list the variables, their domains, and the constraints. Encode unary constraints as a constraint rather than pruning the domain. (No need to solve the problem, just provide variables, domains and constraints.)

- Variables: $\{M, D, B, S, T, G\}$
- Domains: $\{1, 2, 3, 4, 5, 6\}$ for all variables
- Constraints: alldiff(M, D, B, S, T, G), $|B - M| = 1$, $|B - P| = 1$, $|S - T| = 1$, $G \in \{1, 6\}$.

Q2. (10 marks) Consider the problem of constructing (not solving) crossword puzzles: fitting words into a rectangular grid. The grid, which is given as part of the problem, specifies which squares are blank and which are shaded. Assume that a list of words (i.e., a dictionary) is provided and that the task is to fill in the blank squares by using any subset of the word list. Formulate this problem precisely in two ways:

(a) As a general search problem. Choose an appropriate search algorithm and specify a heuristic function. Is it better to fill in blanks one letter at a time or one word at a time?

(b) As a constraint satisfaction problem. Should the variables be words or letters? Which formulation do you think will be better? Why?

a. Crossword puzzle construction can be solved many ways. One simple choice is depth-first search. Each successor fills in a word in the puzzle with one of the words in the dictionary. It is better to go one word at a time rather than one letter at a time, to minimize the number of steps.

b. As a CSP, there are even more choices. You could have a variable for each box in the crossword puzzle; in this case the value of each variable is a letter, and the constraints are that the letters must make words. This approach is feasible with a most-constraining value heuristic. Alternately, we could have each string of consecutive horizontal or vertical boxes be a single variable, and the domain of the variables be words in the dictionary of the right length. The constraints would say that two intersecting words must have the same letter in the intersecting box. Solving a problem in this formulation requires fewer steps, but the domains are larger (assuming a big dictionary) and there are fewer constraints. Both formulations are feasible.

Q3. (15 Marks) Give precise formulations for each of the following as constraint satisfaction problems:

(a) Rectilinear floor-planning: find non-overlapping places in a large rectangle for a number of smaller rectangles.

(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 they can teach.

(c) Hamiltonian tour: given a network of cities connected by roads, choose an order to visit all cities in a country without repeating any.

a. For rectilinear floor-planning, one possibility is to have a variable for each of the small rectangles, with the value of each variable being a 4-tuple consisting of the x and y coordinates of the upper left and lower right corners of the place where the rectangle will be located. The domain of each variable is the set of 4-tuples that are the right size for the corresponding small rectangle and that fit within the large rectangle. Constraints say that no two rectangles can overlap; for example if the value of variable R_1 is $[0, 0, 5, 8]$, then no other variable can take on a value that overlaps with the $0, 0$ to $5, 8$ rectangle.

b. For class scheduling, one possibility is to have three variables for each class, one with times for values (e.g. MWF8:00, TuTh8:00, MWF9:00, ...), one with classrooms for values (e.g. Wheeler110, Evans330, ...) and one with instructors for values (e.g. Abelson, Bibel, Canny, ...). Constraints say that only one class can be in the same classroom at the same time, and an instructor can only teach one class at a time. There may be other constraints as well (e.g. an instructor should not have two consecutive classes).

c. For Hamiltonian tour, one possibility is to have one variable for each stop on the tour, with binary constraints requiring neighboring cities to be connected by roads, and an AllDiff constraint that all variables have a different value.

Q4. (10 Marks) Solve the cryptarithmetic problem in Figure 6.2 by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics. Show the trace of each variable choice and assignment.

The exact steps depend on certain choices you are free to make; here are the ones I made:

a. Choose the X_3 variable. Its domain is $\{0, 1\}$.

b. Choose the value 1 for X_3. (We can't choose 0; it wouldn't survive forward checking, because it would force F to be 0, and the leading digit of the sum must be non-zero.)

c. Choose F, because it has only one remaining value.

d. Choose the value 1 for F.

e. Now X_2 and $\mathbf{X}_1$ are tied for minimum remaining values at 2; let's choose X_2.

f. Either value survives forward checking, let's choose 0 for X_2.

g. Now X_1 has the minimum remaining values.

h. Again, arbitrarily choose 0 for the value of X_1.

i. The variable O must be an even number (because it is the sum of $T + T$ less than 5 (because $O + O = R + 10 \times 0$). That makes it most constrained.

j. Arbitrarily choose 4 as the value of O.

k. R now has only 1 remaining value.

l. Choose the value 8 for R.

m. T now has only 1 remaining value.

n. Choose the value 7 for T.

o. U must be an even number less than 9; choose U.

p. The only value for U that survives forward checking is 6.

q. The only variable left is W.

r. The only value left for W is 3.

s. This is a solution.

This is a rather easy (under-constrained) puzzle, so it is not surprising that we arrive at a solution with no backtracking (given that we are allowed to use forward checking).

Q5. (5 Marks) Consider a CSP with variables X; Y with domains {1, 2, 3, 4; 5, 6} for X and {2, 4, 6} for Y , and constraints X < Y and X +Y > 8. List the values that will remain in the domain of X after enforcing arc consistency for the arc X ->Y (which prunes the domain of X, not Y).

Solution. The resulting domain of X is {3, 4, 5}.

Q6. (15 Marks) Are the following statements true or false?

(a) Running forward checking after the assignment of a variable in backtracking search will ensure that every variable is arc consistent with every other variable.

(b) In a CSP constraint graph, a link (edge) between any two variables implies that those two variables may not take on the same values.

(c) Two different CSP search algorithms may give different results on the same constraint satisfaction problem.

(d) A CSP can only have unary and binary constraints.

(e) If forward checking eliminates an inconsistent value, enforcing arc consistency would eliminate it as well.

(f) If enforcing arc consistency eliminates an inconsistent value, forward checking would eliminate it as well.

(g) When enforcing arc consistency in a CSP, the set of values which remain when the algorithm terminates depends on the order in which arcs are processed from the queue.

Solution.

(a) False. Forward checking only checks variables that are connected to the variable being assigned.

(b) False. A link represents any constraint, including, e.g., equality!

(c) True. A CSP may have many solutions, and which one is found first depends on the order in which the space of assignments is searched. If there is only one solution, then all algorithms should agree.

(d) False. There are several examples in the book and elsewhere of constraints with more than two variables, including cryptarithmetic and sudoku problems. (It is possible to reduce any n-ary constraint into a set of binary constraints.)

(e) True.

(f) False.

(g) False. Arc consistency eliminates values which cannot be possible, it does not make choices when there are several possibilities, so any order will end up with the same results.

Q7. (5 Marks) 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.

The most constrained variable makes sense because it chooses a variable that is (all other things being equal) likely to cause a failure, and it is more efficient to fail as early as possible (thereby pruning large parts of the search space). The least constraining value heuristic makes sense because it allows the most chances for future assignments to avoid conflict.

Q8. (10 Marks) Sudoku has been introduced in text as a CSP to be solved by search over partial assignments because that is the way people generally undertake solving Sudoku problems. It is also possible, to attack Sudoku problems with local search over complete assignments. How well would a local solver using the min-conflicts heuristic do on Sudoku problems?

It is certainly possible to solve Sudoku problems in this fashion. However, it is not as effective as the partial-assignment approach, and not as effective as min-conflicts is on the N-queens problem. Perhaps that is because there are two different types of conflicts: a conflict with one of the numbers that defines the initial problem is one that must be corrected, but a conflict between two numbers that were placed elsewhere in the grid can be corrected by replacing either of the two. A version of min-conflicts that recognizes the difference between these two situations might do better than the naive min-conflicts algorithm.