CSC242: Homework 2.1
AIMA Chapter 6.0–6.4

1. What are the three components of a constraint satisfaction problem?

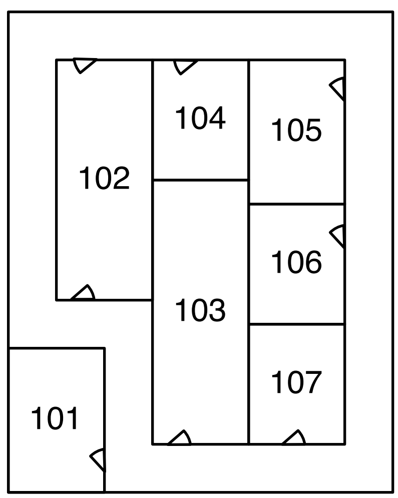
   ANSWER:

   - Variables: $\{X_1, \ldots, X_n\}$
   - Domains (one for each variable): $\{D_1, \ldots, D_n\}$. Each domain D_i consists of a set of values: $\{v_{i,1}, \ldots, v_{i,k_i}\}$.
   - Constraints that specify allowable combinations of values

2. What is a answer to a constraint satisfaction problem?

   ANSWER: A complete, consistent assignment of values to variables. Complete: every variable is assigned a value. Consistent: no constraints are violated by the assigned values.

3. Suppose you are running a hotel with seven rooms on one floor:



   Three families want to book all seven rooms. However people in the same family don't get along, so they cannot be in adjacent rooms.

   (a) Formulate this problem as a constraint satisfaction problem.

   ANSWER: Have to assign families to room subject to the constraints on adjacent rooms:

- Variables: $\{X_{101}, \ldots, X_{107}\}$ corresponding to the rooms
- Domains: $\{f_1, f_2, f_3\}$ corresponding to the families, for all variables
- Constraints: For each pair of adjacent rooms there is a binary "$\neq$" constraints between the corresponding variables. For example: $X_{102} \neq X_{103}$, and so on.

Note that this problem is isomorphic to the Australia map coloring problem (Room 103 is South Australia).

(b) How many solutions are there to this CSP?

ANSWER: Start with room 103, which can have any of the three families. Then, moving clockwise, room 102 can have either of the other two families, and then everything else is strictly determined. That makes 6 possibilities for rooms 102–107, times the three possibilities for room 101. Total: 18.

(c) How many solutions are there if there are four families?

ANSWER: Four choices for Room 103, 3 for Room 102, then two each for rooms 104, 105, 106, and 107, times 4 for room 101 = $4 \times 3 \times 2 \times 2 \times 2 \times 2 \times 4 = 768$.

(d) How many solutions are there if there are only two families?

ANSWER: No solutions with just two families. This could only work if no room was adjacent to more than one other room, which clearly isn't the case in this hotel.

4. Alice, Betty, and Carol are in a book club. They're trying to figure which of five different books they should read next. The books are: *Dreams From My Father* by Barack Obama, *Lord of the Rings* by J.R.R. Tolkein, *Artificial Intelligence: A Modern Approach* by Stuart Russell and Peter Norvig, *Harry Potter and The Sorceror's Stone*, by J.K. Rowling, and *The Fabric of the Cosmos: Space, Time, and the Texture of Reality* by Brian Greene.

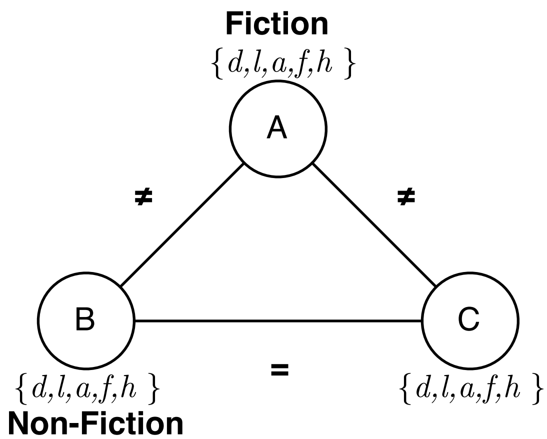
   In this book club, they don't all have to read the same book. Alice only likes fiction, while Betty only likes non-fiction. Furthermore, Alice won't read whatever either Betty or Carol are reading, while Betty and Carol always read the same book.

   (a) Formulate this as a constraint satisfaction problem.

   ANSWER:

   - Variables: A, B, C (first letter of person's name).
   - Domains for all $\{d, l, a, h, f\}$ (first letter of titles of books).
   - Unary constraints: A fiction $\{l, h\}$; B non-fiction $\{d, a, f\}$.
     Binary constraints: $A \neq B$, $A \neq C$, $B = C$.

   Picture follows:



Thus a answer is an assignment of a book to each person, consistent with the constraints.

(b) Propagate unary constraints and show the results.

ANSWER: Domain of A exactly $\{l, h\}$. Domain of B exactly $\{d, a, f\}$.

(c) Solve the CSP using a combination of search and constraint propagation. At each step (assignment or propagation), show the state of the problem. (Hint: You should only need two iterations if you've done it right.)

ANSWER: One possible solution and trace:

- Search: choose $A = l$.
- Propagate constraints: B unchanged, $C = \{d, a, f, h\}$.
- Search: choose $B = d$ (or a or f).
- Propagate constraints: A unchanged, $C = d$ (or a or f, to match B).

Answer: $A = l, B = d, C = d$. Also possible: $\langle l, a, a\rangle$, $\langle l, f, f\rangle$, $\langle h, d, d\rangle$, $\langle h, a, a\rangle$, $\langle h, f, f\rangle$.

5. Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of the partial assignment $\{WA = green, V = red\}$ for the Australia map coloring problem.

ANSWER: Trace of AC-3 for one possible ordering of the arcs:

- Start with all arcs (binary constraints) in the queue.
- Remove SA–WA, delete $green$ from SA.
- Remove SA–V , delete red from SA, leaving only $blue$.
- Remove NT–WA, delete $green$ from NT.
- Remove NT–SA, delete $blue$ from NT, leaving only red.
- Remove NSW–SA, delete $blue$ from NSW.
- Remove NSW–V , delete red from NSW, leaving only $green$.

- Remove Q–NT, delete red from Q.
- Remove Q–SA, delete $blue$ from Q.
- Remove Q–NSW, delete $green$ from Q, leaving no domain for Q.

6. Consider the problem of creating (not solving) crossword puzzles. Assume you are given a rectangular grid with blank and shaded squares, and a list of words. Your goal is to fit the words into the grid satisfying the constraints of a crossword puzzle about shared letters.

   Formulate this precisely as a constraint satisfaction problem. Comment on the strengths or weaknesses of your formulation(s).

   ANSWER: This obvious approach is to have a variable for each blank cell in the grid. The domain of all variables is the set of possible letters $\{a, \ldots, z\}$. The constraints are that the letters in successive horizontal or vertical cells must make words. Given a grid, you could figure out what these spans are. And given a list of words, you could check possible assignments against the list.

   An alternative approach "compiles" the grid into the horizontal and vertical spans. There is one variable for each span. The domains of the variables are the words from the list (of the right length). The constraints are the shared letters between spans (words), which again you could compile out of the grid at the outset.

   The first formulation has many more variables (empty cells), but only 26 values in each variable's domain. The latter formulation has fewer variables (word spans), but many more possible values (assuming a large dictionary). There are fewer constraints in the second formulation, since the first formulation includes all the adjacency constraints explicitly. So probably the second formulation would be faster to check constraints. Either way, it would require some careful data structure design.