

HW2: Constraint Satisfaction

1 Crossword Puzzles

(This is a variant of exercise 5.4 from the book.)

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 (i.e., can contain letters) and which squares are shaded (i.e., cannot contain letters). Assume also that a list of words (i.e., a dictionary) is provided. The task is to fill in the blank squares using any subset of the list.

1. Formulate this problem as a CSP where the variables are *words*. List all the variables and constraints.
2. Formulate this problem as a CSP where the variables are *letters*. List all the variables and constraints.
3. Suppose we wish to make sure the the crossword puzzle we generate doesn't contain multiple words that are "too similar." We'll say that words w_1 and w_2 are too similar if *either* one can be obtained from the other by changing exactly one character *or* one is a substring of the other. For instance "dog" and "dogs" are too similar (by the second constraint) as are "dog" and "dog" (again, by the second constraint). Similarly, "cats" and "cots" are too similar (by the first constraint). How would you specify these additional constraints in both the by-word and by-letter formulations?

2 Arc Consistency

While skiing at Alta last week, Alice (our venerable computer science student) decides that she wants to get a minor in math. She has already taken many of the requirements as part of her CS degree; all that remains are the following courses. She must take both Analysis (A) and Analysis II (B). She then has three electives of which she must take two; these are: Linear Algebra (C), Number Theory (D) and Modern Algebra (E)¹. Alice has three more semester of study to go. Fortunately, every course is offered every semester, but she can take at most two math courses during any given semester.

Of course, there are constraints on how she must take the courses. Obviously she must take Analysis (A) before Analysis II (B). She must also take Linear Algebra (C) before Modern Algebra (D). She can only take Number Theory (D) once she's taken Modern Algebra (E). Sadly, Analysis (A) and Linear Algebra (C) are always offered at exactly the same time, so she cannot take both in the same semester.

We will now formulate Alice's scheduling problem as a CSP in which each variable corresponds to a class Alice must take and the values correspond to the numbering of the remaining semesters: 1 for junior Spring, 2 for senior Fall and 3 for senior Spring; ∞ for "does not take this course."

1. Find an assignment of the variables to values (just by trial and error) that satisfies the constraints.
2. State the domain of the variables and the constraints in mathematical notation using the following symbols: $=, \neq, <, >, \leq, \geq$.
3. We now solve this CSP using backtracking search with arc consistency. Namely, we will run the AC-3 algorithm as a preprocessor *and* after each variable assignment. When there is ambiguity as to which variable to assign next, assign a value to the variable corresponding with the course letter that comes first alphabetically. Furthermore, when the next value to assign to a variable is ambiguous, break ties by assigning the earliest value first.
 - (a) What are the remaining domains after enforcing arc consistency on the initial CSP with no assignments?
 - (b) What are the remaining domains after assigning $A = 1$ and re-enforcing arc consistency?
 - (c) What about after assigning $A = 1$ and $B = 2$?
4. **(CS 6300 only)** Suppose we're running AC-3 but trying to get Alice to finish in only two semesters instead of one. Sketch a complete run of AC-3 under these new constraints and say precisely when it will fail and how many states are expanded before failing.

¹This is not exactly the requirements for the math minor at the U, but is close.