CSCE476/876 **Spring 2010**

Homework 6

Assigned on: Monday, March 22, 2010.

Due: Monday, March 29, 2010.

Points: 87 points + 10 Points bonus

Contents

Alert: If you submit your homework handwritten, it must be *absolutely neat* or it *will not* be corrected. If you type your homework (preferable), submit using webhandin.

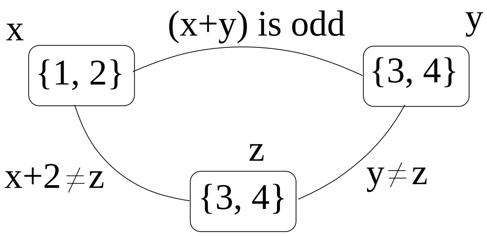
---

1 Dynamic Variable Ordering (8 points)

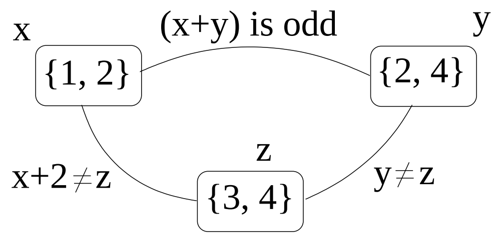
1. Explain the principle for dynamic variable ordering in Backtrack Search. (2 points)
2. Describe three heuristics that implement this principal. (6 points)

2 Constraint propagation (6 points)

Consider the CSPs represented by the constraint networks below:



CSP I



CSP II

For each CSP,

1. State whether or not the CSP is arc-consistent.
2. If it is, explain why.
3. If it is not, explain which domain must be reduced to make the CSP arc-consistent, and specify the constraint that can be used to reduce the domain.

3 Simple scheduling problem (I) (10 points)

Courtesy of Rina Dechter

Consider the problem of scheduling five tasks: T_1, T_2, T_3, T_4, and T_5, each of which takes one hour to complete. The tasks may start at 1:00, 2:00, 3:00. Any number of tasks can be executed simultaneously provided the following restrictions are satisfied.

- T_1 must start after T_3.
- T_3 must start before T_4 and after T_5.
- T_2 cannot execute at the same time as T_1.
- T_2 cannot execute at the same time as T_4.
- T_4 cannot start at 2:00.

1. Formulate the problem as a CSP by stating: the variables, their domain, and the applicable constraints. (4 points)
   Hints: focus on the start time of a task.
2. Draw the constraint graph. (2 points)
3. Apply arc-consistency to each constraint in the CSP until no values can be ruled out (i.e., the CSP becomes arc-consistent). (4 points)

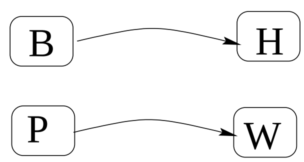
4 Simple scheduling problem (II) (15 points)

Courtesy of Daphne Koller

Mr. Smith is the owner of a factory that is about to produce solar power flashlights. He wishes to see if he can make a torch in four hours. The construction of the torch consists of four tasks:

1. constructing the bulb: **B** duration time is 2 hours
2. making the solar panel: **P** duration time is 1 hour
3. doing the wiring: **W** duration time is 2 hours
4. assembling the housing: **H** duration time is 1 hour

If Mr. Smith could perform all the tasks simultaneously, he would be able to make a torch in 2 hours. However, some of the tasks need to be completed before others can proceed. The housing cannot start before the bulb has been built; the wiring cannot be started before the panel is made. This information is summarized in the precedence graph below:



Moreover, the only person qualified to construct the bulb and panel is Constance, the electrical engineer. Therefore, the bulb and panel cannot be made simultaneously.

1. Formulate this scheduling problem as a CSP. (10 points)
   Hints: Choose the variables to be time point at which a task is started. State the initial domain of each variable as a set of discrete values, one value per hour. Specify the unary constraints that restrict the start time of a task given its duration to the duration of the entire job (i.e., four hours). For example, a task whose duration is 1 hour cannot start at 'time point' 4. Specify the binary constraints that link the variables as an algebraic constraint. The constraint expressions we are looking for are of the form $S_b + 2 < S_w$ (this example expression is only to give you an idea of what to write: it is *not* one of the constraints listed above.)

2. Draw the *complete* search tree generated by a backtrack-search procedure *with forward-checking* using a static variable ordering of your choice. Specify, as best you can, the domains filtered after each forward-checking step. (5 points)

5 N-Queen Problem as a CSP (14 points)

Consider the 4-queens problem where each queen is associated with a row and can be assigned to any column in the row.

1. Define this problem as a CSP. Specify the variables and their domain, and each binary constraint by 'extension.' (1 point)

2. Define a binary constraint C_{Q_i,Q_j} between two variables Q_i and Q_j by 'intension.' (4 points)

3. What is the size of this CSP (which is the size of the search tree it may yield)? (1 point)

4. Draw the constraint graph. (2 points)

5. Arc-consistency of a binary constraint C_{Q_i,Q_j} between two variables Q_i and Q_j ensures that every value for the variable Q_i has a support (at least one consistent value) in the domain of Q_j and vice-versa. Run manually arc-consistency on the 4-Queens problem. Can you remove any value? At the end of the operation the CSP is said to be arc-consistent. (2 points)

6. Arc-consistency is also called 2-consistency because it considers all combinations of two variables at the same time. Let's consider all combinations of 3 variables at the same time and let's check whether or not every value in the domain of a given variable has a support in the domain of the two other variables (simultaneously). If it does not, the value can be removed. Can you remove any value? This consistency property is called (1,2)-consistency. (4 points)

6 Consistency checking (10 points)

Courtesy of Rina Dechter, UC-Irvine.

Consider the simple coloring problem shown in Figure 1.

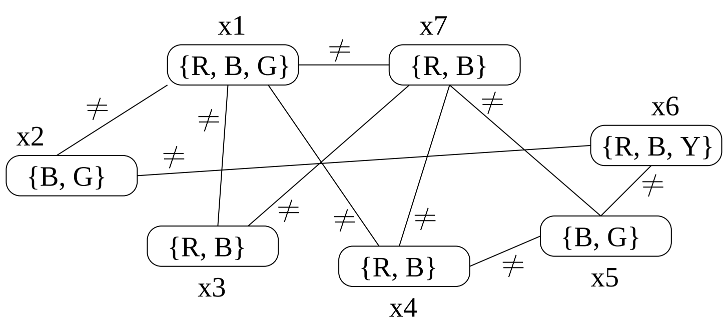


Figure 1: *A simple CSP.*

1. Compute the equivalent network that is *strong* 3-consistent. As a reminder, *strong* 3-consistency requires that the network be both arc-consistent (i.e., 2-consitent) and path-consistent (i.e., 3-consitent). Also note that path-consistency requires considering *all* combinations of 3 variables, replacing all univeral constraints with tighter ones.

2. Find a solution to the CSP.

7 Latin square (Bonus 10 points)

Adapted from of Daphne Koller, Stanford University.

A Latin Square is a $N \times N$ array filled with colors, in which no color appears more than once in any row or column. Finding a solution to a 4×4 Latin Square can be formulated as a CSP, with a variable for each cell in the array, each having a domain of $\{r, g, b, y\}$, and a set of constraints asserting that any pair of cells appearing in the same row must have different colors, and that any pair of cells appearing in the same column must have different colors.

1 **r**	2 **g**	3 **b**	4 **y**
5 **g**	6 **y**	*r g b y*	r g b y
7 **b**	r g b y	r g b y	r g b y
r g b y	r g b y	r g b y	r g b y

Figure 2: *Current state of search.*

At this point in the search, seven of the cells have been instantiated (displayed in **boldface** in Figure 2), and the initial domains of the remaining cells are shown. (The next cell to instantiate has the domain values in *italics*, but this information is irrelevant for this exercise.)

Re-execute (from scratch) the same 7 first assignments in the specified order and, at *each* assignment, draw the Latin Square while filtering the domains of the relative future variables. Do this process for the two following look-ahead strategies:

- the partial look-ahead strategy, forward checking (FC)
- the full look-ahead strategy, maintaining arc consistency.

Indicate eliminated values by crossing them out or just erasing them.

8 Reduction of 3SAT into a CSP (24 points)

1. Formulate 3SAT as a CSP. (15 points)
   Indications: Your formulation should be as general as possible and should represent each of the elements of the 3SAT and its question in the terminology of a CSP. Consider X, the set of Boolean variables of a 3SAT instance. What are the values that a variable can take? Use this to define the variables of the CSP and their values. A clause is a disjunction of literals. How to represent a clause in the CSP formalism? A 3SAT sentence is a conjunction of clauses. How is the sentence represented in the CSP formalism? Finally, state how the question of 3SAT is reduced as a question to the CSP

and prove that a solution to 3SAT exists if and only if a solution to the corresponding CSP exists.

2. What is the arity of the constraints of the resulting CSP? (2 points)

3. As a direct application of your reduction, transform the following 3SAT problem into a CSP. Specify the variables, their domains, define the constraint in extension, and draw the corresponding constraint graph: (4 points)

$$(c_1 \vee c_2 \vee c_3) \wedge (c_2 \vee c_3 \vee c_4) \wedge (\neg c_1 \vee c_5) \wedge (c_1 \vee c_4 \vee c_5)$$

4. Knowing how a 3SAT clause (which is a disjunction of exactly 3 literals) is represented in the CSP, how do you propose to represent a clause of SAT (which has an arbitrary number of literals in the clause)? (3 points)