CSCE476/876 **Spring 2015**

Homework 6

Assigned on: Monday, March 30, 2015.

Due: Monday, April 13, 2015.

Points: 144 points + up to 37 bonus for ugrads, 156 points + up to 25 bonus for grads

You can choose to do either **Choice A**: Section 7 (worth 100 points) *or* **Choice B**: Sections 8 to 14 (worth 100 points + 20 bonus). It is your decision. If you choose to do both options, you will receive the *maximum* of the grades of the two sections, not the sum.

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.

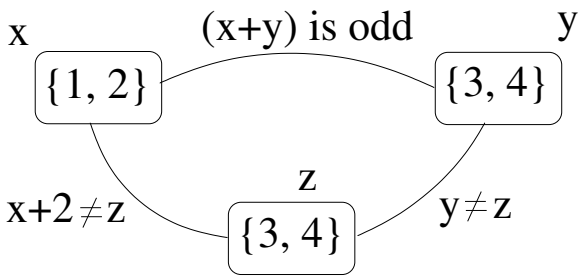
---

1 Dynamic Variable Ordering (8 points)

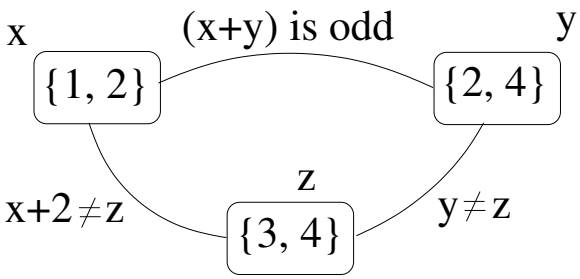
1. Explain the principle for dynamic variable ordering in Backtrack Search. (2 points)
2. Describe three heuristics that implement this principal. (6 points)

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.

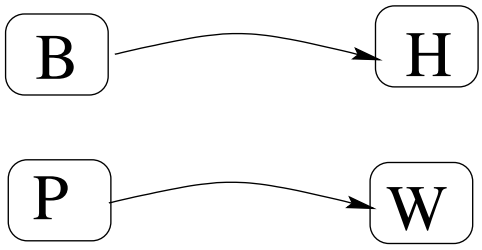
3 Simple scheduling problem (15 points)

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)

4 Consistency checking (10 points)

Courtesy of Rina Dechter, UC-Irvine.

Consider the simple coloring problem shown in Figure 1.

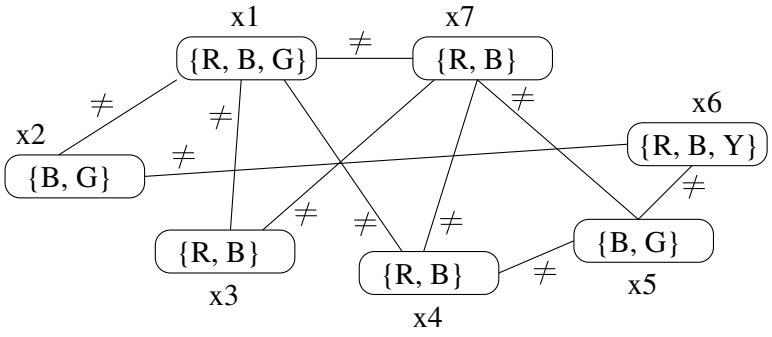


Figure 1: *A simple CSP.*

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.

5 Latin square (5 points + 5 bonus)

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 <u>*each*</u> 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) (5 points)
- the full look-ahead strategy, maintaining arc consistency. (5 bonus points)

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

6 Grad students: Reduction of 3SAT into a CSP (12 points)

(Mandatory for graduate students, bonus for undergrads 12 points)

1. Formulate 3SAT as a CSP. (8 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? (1 point)

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: (2 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)? (1 point)

7 A: Implementation, Solving SAT (100 points)

Write a search algorithm to determine the satisfiability of a SAT instance. You can either write:

- A DPLL procedure (backtrack search),
- A local search procedure.

You must

- Clearly describe, in addition to your code, your data structures, how your search algorithm operates, and the improvements, if any, that you have included in your code.
- We recommend that you use the standard file for input files known as the 'simplified version of the DIMACS format':
  http://www.satcompetition.org/2009/format-benchmarks2009.html
- Test the performance of your algorithm on some non trivial uniform random instances taken from the SAT Competition. For example:
  http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html

Alert: many implementations exist in the literature and on the web. We expect you to do your *own* implementation.

8 B: AIMA, Exercise 7.1, page 279. (16 points)

9 B: AIMA, Exercise 7.7, page 281. (6 points)

10 B: Truth Tables (8 points)

Use truth tables to show that each of the following is a tautology.

1. $(p \wedge q) \rightarrow \neg(\neg p \vee \neg q)$
2. $[Mary \wedge (Mary \rightarrow Susy)] \rightarrow Susy$
3. $\alpha \rightarrow [\beta \rightarrow (\alpha \wedge \beta)]$
4. $(a \rightarrow b) \rightarrow [(b \rightarrow c) \rightarrow (a \rightarrow c)]$

11 B: AIMA, Exercise 7.10, page 281. (16 points)

Only b, c, d, e, f, and g.

12 B: Logical Equivalences (8 points)

Using a method of your choice, verify:

1. $(\alpha \rightarrow \beta) \equiv (\neg\beta \rightarrow \neg\alpha)$ contraposition
2. $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ de Morgan
3. $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \gamma) \vee (\alpha \wedge \beta))$ distributivity of $\wedge$ over $\vee$

13 B: AIMA, Exercise 7.22, page 284. (18 points + 20 bonus)

Parts a, b, and c are required. Parts d, e, and f are bonus.

14 B: Proofs (28 points)

Give the explantions of each step if the steps are given, and give both the explanation and step if they are not.

- If $q \wedge (r \wedge p), t \to v, v \to \neg p$, then $\neg t \wedge r$.

  Proof **Explanations**

  1. $q \wedge (r \wedge p)$ Given
  2. $t \to v$ Given
  3. $v \to \neg p$ Given
  4. $t \to \neg p$
  5. $(r \wedge p)$
  6. r
  7. p
  8. $\neg\neg p$
  9. $\neg t$
  10. $\neg t \wedge r$

- If $p \to (q \wedge r), q \to s$, and $r \to t$, then $p \to (s \wedge t)$.

  Proof **Explanations**

  1.
  2.
  3.
  4.
  5.
  6.
  7.

- **Prove by contradiction.**

  If $\neg(\neg p \wedge q), p \to (\neg t \vee r), q$, and t, then r.

  Proof **Explanations**

  1. $\neg(\neg p \wedge q)$ Given
  2. $p \to (\neg t \vee r)$ Given
  3. q Given
  4. t Given
  5. $\neg r$ Negation of Conclusion
  6.
  7.
  8.
  9.
  10.
  11.
  12.