CS 70 Spring 2005 — Discrete Mathematics for CS — Clancy/Wagner — HW 4

Due Thursday, February 17th

Coverage: This assignment involves topics from the February 8 and 10 lectures (labeled as lectures 6, 7, and 8 in the notes).

Administrative reminders: We will accept only unformatted text files or PDF files for homework submission. Include your name, login name, section number, and partner list in your submission. Give the command `submit hw4` to submit your answers to this assignment.

Homework exercises:

1. **(23 pts.) Boolean decision trees**

DECISION TREE

A **decision tree** is a variant of binary trees in which each *internal* node is labelled with a Boolean variable. Each leaf is labelled with an atom T or F. The notation $\langle P, t_1 \bullet t_2 \rangle$ denotes a decision tree whose root is labelled with P and whose left and right subtrees are t_1 and t_2. A decision tree t represents a Boolean function as follows, where m is a model (i.e., an assignment that assigns each Boolean variable to true or false):

If t is an atom, $eval(t,m) = t$.
$eval(\langle P, t_1 \bullet t_2 \rangle, m) = eval(t_1, m)$ if P is true in m
$eval(\langle P, t_1 \bullet t_2 \rangle, m) = eval(t_2, m)$ if P is false in m

(a) Draw a decision tree that represents the same Boolean function as $A \implies B$.

(b) How large is the smallest decision tree (i.e. the tree with the fewest nodes) that represents the disjunction of n variables? Briefly justify your answer.

(c) Prove that every Boolean function can be represented by a decision tree.

(d) Draw a decision tree that represents the following function of X_1, X_2, X_3: T if at least two of the three variables are true, otherwise F.

(e) Write a recursive mathematical definition for the function $kn(k,n)$ that returns a decision tree on n variables $X_1, \ldots, X_n$; the decision tree returns T iff at least k of the n variables are true.

2. **(27 pts.) Timetabling**

Consider the following exam timetabling problem. There are n student candidates, $c_1, c_2, \ldots, c_n$; m disjoint time slots, $t_1, t_2, \ldots, t_m$; and ℓ subject exams $e_1, e_2, \ldots, e_\ell$. For each candidate, we are given $E(c_i)$, the subset of the exams that the candidate has to take. A *schedule* is any assignment of each of the exams to a time slot. A schedule is *feasible* if no candidate is expected to take two exams at the same time. Our task is to come up with a feasible schedule. Note that the problem is only interesting when $\ell > m$, i.e., the number of exams exceeds the number of slots. Otherwise we can just schedule all exams in disjoint slots and there will be no clashes.

Let's see how to cast this problem into a logical reasoning framework, by constructing a Boolean expression S that is satisfiable if and only if a feasible schedule exists. Obviously S will depend on the problem input (i.e., on the particular set of candidates we are given, as well as on n, m and ℓ). The variables of the formula S will be X_{ij}, one for each exam e_i and time slot t_j, where X_{ij} is true iff exam e_i is scheduled in time slot t_j.

(a) Show first how to write a formula S' that expresses just the fact that the variables X_{ij} encode a proper, complete schedule (ignoring feasibility for students, in true Sproul Hall tradition). You may use any of the standard syntax for Boolean expressions given in class, including the operators $\wedge, \vee, \neg, \implies$. You will probably also want to use notation such as $\bigvee_{i=1}^{n} Y_i$, which stands for $Y_1 \vee Y_2 \vee \ldots Y_n$ in the same way that $\sum_{i=1}^{n} y_i$ stands for $y_1 + y_2 + \cdots + y_n$. [Hint: Deal with each exam separately. What do you have to ensure about each exam?]

(b) Now show how to add in further constraints that require the schedule to be feasible. Hence deduce how to construct the entire formula S.

(c) Verify your construction by showing (i) how to construct a satisfying assignment for S given any feasible schedule; and (ii) how to construct a feasible schedule given any satisfying assignment for S.

(d) Is there a bijection (i.e., a perfect one-to-one correspondence) between satisfying assignments and feasible schedules? Briefly defend your answer.

3. (10 pts.) Setting up a minesweeper problem

Consider the following minesweeper problem:

2			
1	1	2	1
	1	2	3

(a) List the CNF expressions corresponding to the local constraints $N_{1,1}$, $N_{2,1}$, $N_{3,1}$, arising from squares $(1,1)$, $(2,1)$, $(3,1)$.

(b) Construct a truth table for the problem; it should have 8 rows. Add columns for the expressions $N_{1,1}, N_{2,1}, N_{3,1}$.

(c) Mark those rows that correspond to satisfying assignments for this set of constraints.

(d) Deduce what you can about the unknown squares.