CSE 431
Spring Quarter 2001
Assignment 2
Due Friday, April 13

All solutions should be neatly written or type set. All major steps in proofs and algorithms must be justified.

1. (10 points) Using implementation descriptions show that the class of decidable languages is closed under union, intersection, complement. For this problem show how to design Turing machines a to decide $L_1 \cup L_2$, $L_1 \cap L_2$, and $\overline{L_1}$ given Turing machines to decide to decide L_1 and L_2. Assume that the Turing machines to decide L_1 and L_2 are of the one tape variety. Your Turing machines can be of the multitape variety.

2. (10 points) Using high level descriptions show that the class of decidable languages is closed under concatenation and Kleene star. For this problem show how to design algorithms to decide $L_1 L_2$ and L_1^* given algorithms to decide L_1 and L_2.

3. (10 points) A special form of Boolean formulas is called *conjunctive normal form* or simply *CNF*. For example,

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$

is such a CNF formula. You will want to review pages 14-15 in the book. We say such a formula is *satisfiable* if there is some way to assign the Boolean variables to 1 (true) and 0 (false) so that the formula evaluates to true. For the example, the assignment $x_1 = 1, x_2 = 0, x_3 = 0$ satisfies the formula. So the formula is satisfiable. The following is called the CNF-SAT problem.

   - Input: A CNF formula F.
   - Property: F is satisfiable.

   (a) Design an encoding for CNF formulas in a finite alphabet.

   (b) Design a nondeterministic multitape Turing machine (using an implementation description, not a formal description) that accepts the set of encodings of satisfiable CNF formulas in your encoding. Use nondeterminism wisely to avoid the your Turing machine from searching through all possible assignments to find a satisfying one.