Automata and Formal Languages Comprehensive Exam

Fall 2004

Problem 1 (10 points)

Give context-free grammars generating the following languages over the alphabet $\{0, 1\}$ (you need not prove the correctness of your grammars):

(a) $\{a^i b^j a^{i+j+k} b^k \,:\, i, j, k \geq 0\}$;

(b) all strings with an equal number of a's and b's.

Solution:

(a)

$$\begin{aligned} S &\rightarrow AC && (1)\\ A &\rightarrow aAa && (2)\\ A &\rightarrow B && (3)\\ B &\rightarrow bBa && (4)\\ B &\rightarrow \epsilon && (5)\\ C &\rightarrow aCb && (6)\\ C &\rightarrow \epsilon && (7) \end{aligned}$$

(b)

$$\begin{aligned} S &\rightarrow aSbS && (8)\\ S &\rightarrow bSaS && (9)\\ S &\rightarrow \epsilon && (10) \end{aligned}$$

Problem 2 (15 points)

Decide whether the following statements are TRUE or FALSE. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) If L_1 and L_2 are both non-regular, then $L_1 \cap L_2$ must be non-regular.

(b) Suppose there is a polynomial-time reduction from the language L_1 to the language L_2. It is possible that L_1 is solvable in polynomial time but L_2 is not even in NP.

(c) Suppose there is a polynomial-time reduction from the language L_1 to the language L_2. If L_1 is recursive, then L_2 must be recursive.

(d) Every infinite regular set contains a subset that is not recursively enumerable.

(e) Every infinite recursively enumerable set contains an infinite subset that is recursive.

Solution:

(a) FALSE

(b) TRUE

(c) FALSE

(d) TRUE

(e) TRUE

Problem 3 (15 points)

Classify each of the following languages as being in one of the following classes of languages: *empty, finite, regular, context-free, recursive, recursively enumerable.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA could be empty or finite, and must always be context-free, but the smallest class that contains all such languages is that of the *regular* languages. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) The intersection of a context-free language and a regular language.

(b) The intersection of a recursive language and a regular language.

(c) The languages accepted by nondeterministic pushdown automata with a single state that accept by empty stack.

(d) The languages accepted by nondeterministic pushdown automata with two stacks.

(e) The complement of a language in NP.

Solution:

(a) Context-free

(b) Recursive

(c) Context-free

(d) Recursively enumerable

(e) Recursive

Problem 4 (15 points)

Specify which of the following problems are *decidable* and which are *undecidable*. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

(a) Given a Turing machine M, does M halt when started with an empty tape?

(b) Given a context-free language L and a regular language R, is $L \subseteq R$?

(c) Given a context-free language L and a regular language R, is $R \subseteq L$?

(d) Given a DFA, does it accept on only finitely many inputs?

(e) Given a PDA, does it accept on only finitely many inputs?

Solution:

(a) Undecidable

(b) Decidable

(c) Undecidable

(d) Decidable

(e) Decidable

Problem 5 (15 points)

A *monotone* 2-SAT formula is a 2-CNF Boolean formula $F(x_1, \ldots, x_n)$ that does not contain negated variables. For example:

$$F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_3).$$

It is clear that there always exists a truth assignment for the variables $x_1, \ldots, x_n$ satisfying the formula F—simply set each variable to TRUE.

Consider the following problem called MONOTONE 2-SAT: given a monotone 2-SAT formula F and a positive integer k, determine whether there exists a truth assignment satisfying F such that the number of variables set to TRUE is *at most* k.

Prove that the MONOTONE 2-SAT problem is NP-complete. (**Hint:** Think about the NP-complete VERTEX COVER problem.)

Solution: Recall that in a graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$, a *vertex cover* is a set $C \subseteq V$ of vertices such that for each edge $(i, j) \in E$, at least one of its endpoints is in C: $\{i, j\} \cap C \neq \emptyset$. The VERTEX COVER problem is the following: given a graph $G = (V, E)$ and a positive integer k, does G contain a vertex cover of size at most k? We know that VC is NP-hard, and establish NP-hardness of MONOTONE 2-SAT via a polynomial-time reduction from VC.

The reduction starts with a VC instance $< G, k >$ and creates an instance $< F, k >$ of MONOTONE 2-SAT, where the monotone 2-CNF formula F is defined as follows: for each vertex $i \in V$, create a Boolean variable x_i; for each edge $(i, j) \in E$, create a clause $x_i \vee x_j$. The reduction runs in linear time, but it remains to verify its correctness.

Suppose G has a vertex cover C of size at most k. Consider the truth assignment for the variables in F in which $x_i =$ TRUE if and only if $i \in C$; clearly, the number of TRUE variables is at most k. We claim that this is a satisfying truth assignment for F. To establish the claim, consider an arbitrary clause $x_i \vee x_j$ of F. Since (i, j) must be an edge of G, and hence C must contain at least one of i and j, it follows that at least one of x_i and x_j is assigned TRUE and hence the clause is satisfied.

Suppose now that there is a satisfying truth assignment for F with no more than k variables set to TRUE. Consider the set of vertices $C = \{i : x_i = \text{TRUE}\}$; clearly, $|C| \leq k$. We claim that C is a vertex cover for G. To see this, focus on any one edge $(i, j) \in E$. Since F must have a clause $x_i \vee x_j$, and that clause is satisfied, at least one of x_i and x_j is assigned TRUE and so at least one end-point of the edge (i, j) belongs to C.

Finally, MONOTONE 2-SAT is in NP because the feasibility of a candidate solution (i.e., a truth assignment) can be checked in polynomial time.