SOLUTIONS:
Automata

Comprehensive Exam: **Autumn 1994**

Automata and Formal Languages (Solutions)

Instructions: You are expected to *sketch* the main ideas in your solutions, but be very brief and avoid unnecessary detail. You are permitted to invoke any result proved in the Hopcroft-Ullman book provided you include the appropriate citation.

1. (8 points) Consider the following context-free grammar G.

$$\begin{aligned} S &\rightarrow bABaa \mid Sa \mid a \\ A &\rightarrow aB \\ B &\rightarrow baB \mid \epsilon \end{aligned}$$

Let L_A and L_B be the languages consisting of the terminal strings that can be derived from the variables A and B, respectively.

(a) [4 points] Show that L_A and L_B are regular by providing regular expressions for these two languages.

(b) [4 points] Show that $L(G)$ is regular by providing a regular expression for it.

Solution: By inspection, L_B has the regular expression $(ba)^*$, and L_A has the regular expression $a(ba)^*$.

Consider now the productions from S. We observe that the productions from A and B do not generate any sentential form containing S. Also the production $S \rightarrow bABaa$ is not recursive in S which implies that it can be used at most once in the any derivation from S. From these facts, we can conclude that the strings derived from S are of the form $(bABaa + a)a^*$. Thus, $L(G)$ has the regular expression $(ba(ba)^*(ba)^* + a)a^*$, which simplifies to $(ba(ba)^* + a)a^*$.

2. (10 points) A *monotone* 2-SAT formula is a 2-CNF boolean formula $F(x_1, \ldots, x_n)$ which does not contain negated variables. For example:

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

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.

Show that the MONOTONE 2-SAT problem is NP-hard. (**Hint:** Think about the vertex cover problem.)

Solution: In a graph $G(V, E)$ with $V = \{1, \ldots, n\}$, a vertex cover is a set of vertices $C \subseteq V$ such that for each edge $(i, j) \in E$, $\{i, j\} \cap C \neq \emptyset$. The VC 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 the NP-hardness of MONOTONE 2-SAT by 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 $G(V, E)$ 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, we show that each clause $x_i \vee x_j$ in F is satisfied. Since (i, j) must be an edge, and C must contain at least one of i and j, if follows that at least one of x_i and x_j is assigned TRUE and 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 $C = \{i \mid x_i = \text{TRUE}\}$; 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.

3. (12 points) Consider the following decision problem:

   Given a deterministic finite state automaton (DFA) M over the alphabet $\Sigma = \{0, 1\}$, does $L(M)$ contain at least 2 strings?

   Is this problem decidable? Justify your answer. (**Hint:** Think about the decision problems of deciding emptiness and finiteness of regular languages.)

   Solution: The problem is indeed decidable. The proof is similar to that for the decidability of emptiness or infiniteness of regular languages, as described in Theorem 3.7 (Hopcroft-Ullman, p. 63). We also make use of the fact that given any string x, it is possible to decide membership of x in $L(M)$.

   Let n be the number of states in the DFA M. The decision procedure enumerates all strings $x \in \Sigma^*$ of length less than n, and checks for their membership in $L(M)$; let m_1 be the number of such strings in $L(M)$. Then, the decision procedure enumerates all strings $x \in \Sigma^*$ such that $n \leq |x| < 2n$ and checks their membership in $L(M)$; let m_2 be the number of such strings in $L(M)$.

   By Theorem 3.7, if $m_2 > 0$ then $L(M)$ is infinite, and the procedure outputs YES. Assume now that $m_2 = 0$. By Theorem 3.7, or by a direct application of the Pumping Lemma, we have that if $L(M)$ contains a string of length at least $2n$ then it must contain a string of length between n and $2n - 1$, i.e., $m_2 > 0$. Since $m_2 = 0$, all strings in $L(M)$ are of length less than n implying that $|L(M)| = m_1$. Thus, the decision procedure now has to merely verify that $m_1 \geq 2$.