Suggested Solutions to Midterm Problems

(Compiled on May 3, 2000)

1. (a) Consider a set $A = \{a, b, c, d, e\}$ and a relation $R = \{(a, b), (a, c), (d, e)\}$ on A. Find the smallest equivalence relation on A that contains R.

   Solution. $R' = \{(\mathbf{a}, \mathbf{a}), (\mathbf{b}, \mathbf{b}), (\mathbf{c}, \mathbf{c}), (\mathbf{d}, \mathbf{d}), (\mathbf{e}, \mathbf{e}), (a, b), (\mathbf{b}, \mathbf{a}), (a, c), (\mathbf{c}, \mathbf{a}), (d, e), (\mathbf{e}, \mathbf{d})\}$ is the smallest equivalence relation on A that contains R. □

   (b) Suppose that R_1 and R_2 are equivalence relations on a set A. Is $R_1 \cup R_2$ an equivalence relation on A? Justify your answer.

   Solution. Suppose $R_1 = \{(a, a), (b, b), (c, c), (d, d), (e, e), (a, b), (b, a)\}$ and $R_2 = \{(a, a), (b, b), (c, c), (d, d), (e, e), (b, c), (c, b)\}$. Then, $R_1 \cup R_2 = \{(a, a), (b, b), (c, c), (d, d), (e, e), (a, b), (b, a), (b, c), (c, b)\}$ is not an equivalent relation, as $(a, b) \in R_1 \cup R_2$ and $(b, c) \in R_1 \cup R_2$ but $(a, c) \notin R_1 \cup R_2$. □

2. (a) Draw the state diagram of an NFA, with as few states as possible, that recognizes $\{w \in \{0, 1\}^* \mid w \text{ ends with } 01 \text{ or } 10\}$. (5 points)

   Solution. See the attached. □

   (b) Convert the NFA in (a) into an equivalent DFA (using the procedure discussed in class); do not attempt to optimize the number of states. (10 points)

   Solution. See the attached. □

3. (a) Draw the state diagram of a DFA that recognizes $A = \{w \in \{0, 1\}^* \mid w \text{ doesn't contain the substring } 011\}$.

   Solution. See the attached. □

   (b) Translate the DFA in (a) to an equivalent context-free grammar (using the procedure discussed in class).

   Solution.

$$
\begin{aligned}
A_1 &\rightarrow 1A_1 \mid 0A_2 \mid \varepsilon \\
A_2 &\rightarrow 0A_2 \mid 1A_3 \mid \varepsilon \\
A_3 &\rightarrow 0A_2 \mid \varepsilon
\end{aligned}
$$

□

4. Translate the DFA in Problem 3 to an equivalent regular expression (using the procedure discussed in class). Please show the intermediate automata.

   Solution. See the attached. □

5. Give a context-free grammar for generating $\{w\#x \mid w^{\mathcal{R}} \text{ is a substring of } x \text{ for } w, x \in \{0,1\}^*\}$. Please explain the intuition behind the grammar.

Solution.

$$\begin{array}{rcl} S & \rightarrow & 0S0A \mid 1S1A \mid \#A \\ A & \rightarrow & 0A \mid 1A \mid \varepsilon \end{array}$$

Without applying "$S \rightarrow \#A$", one can generate from S any intermediate string of the form wSw^RA, where $w \in \{0,1\}^*$, and nothing else. To direct the derivation toward termination, "$S \rightarrow \#A$" has to be applied, making the intermediate string become of the form $w\#Aw^RA$. The rules for A can then be applied to get arbitrary binary strings. □

6. Convert the following CFG into an equivalent CFG in Chomsky normal form (using the procedure discussed in class).

$$S \rightarrow (S) \mid SS \mid \varepsilon$$

Note: There are various definitions of Chomsky normal form. Below is the definition that we used in class.

A context-free grammar is in **Chomsky normal form** if every rule is of the form

$$\begin{array}{rcl} A & \rightarrow & BC \quad \text{or} \\ A & \rightarrow & a \end{array}$$

where a is any terminal and A, B, and C are any variables—except that B and C may not be the start variable. In addition,

$$S \rightarrow \varepsilon$$

is permitted if S is the start variable.

Solution.

(a) Add a new start symbol S_0.

$$\begin{array}{rcl} S_0 & \rightarrow & S \\ S & \rightarrow & (S) \mid SS \mid \varepsilon \end{array}$$

(b) Remove ε-rule $S \rightarrow \varepsilon$.

$$\begin{array}{rcl} S_0 & \rightarrow & S \mid \varepsilon \\ S & \rightarrow & (S) \mid SS \mid (\,) \end{array}$$

(c) Remove unit rule $S_0 \rightarrow S$.

$$\begin{array}{rcl} S_0 & \rightarrow & \varepsilon \mid (S) \mid SS \mid (\,) \\ S & \rightarrow & (S) \mid SS \mid (\,) \end{array}$$

(d) Convert all rules to the proper form.

$$
\begin{array}{lcl}
S_0 & \rightarrow & \varepsilon \mid U_1A_1 \mid SS \mid U_2U_3 \\
S & \rightarrow & U_4A_2 \mid SS \mid U_5U_6 \\
A_1 & \rightarrow & SU_7 \\
A_2 & \rightarrow & SU_8 \\
U_1 & \rightarrow & (\\
U_2 & \rightarrow & (\\
U_3 & \rightarrow &) \\
U_4 & \rightarrow & (\\
U_5 & \rightarrow & (\\
U_6 & \rightarrow &) \\
U_7 & \rightarrow &) \\
U_8 & \rightarrow &)
\end{array}
$$

No optimization was performed in the conversion. □

7. Draw the state diagram of a pushdown automaton that recognizes the set of strings over $\{a, b\}$ with twice as many a's as b's. Please explain the intuition behind the automaton.

   Solution. See the attached. □

8. (a) Prove that the class of context-free languages is closed under the regular operations: *union*, *concatenation*, and *star*. (Hint: A language is context-free if it is generated by some context-free grammar.) (10 points)

   Solution. Suppose that A_1 and A_2 are context-free languages generated by context-free grammar G_1 and G_2, respectively. Let S_1 be the start symbol of G_1 and S_2 the start symbol of G_2.

   $A_1 \cup A_2$ can be generated by "$S \rightarrow S_1 \mid S_2$" plus the rules from G_1 and G_2 (we assume that S is a new symbol not used in G_1 or G_2).

   A_1A_2 can be generated by "$S \rightarrow S_1S_2$" plus the rules from G_1 and G_2.

   A_1^* can be generated by "$S \rightarrow S_1S \mid \varepsilon$" plus the rules from G_1 and G_2. □

   (b) Prove that the class of context-free languages is not closed under either *intersection* or *complement*. (Hint: Find two languages that are context-free, but their intersection is not.) (5 points)

   Solution. $A = \{a^n b^n c^m \mid n, m \geq 0\}$ and $B = \{a^m b^n c^n \mid n, m \geq 0\}$ are context-free languages. But, $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.

   $A_1 \cap A_2 = \overline{\overline{A_1} \cup \overline{A_2}}$. We know that the class of context-free languages is closed under the union operation. If the class of context-free languages were closed under the complement operation, then it would be closed under intersection, contradicting the preceding result. □

9. Prove, using the pumping lemma, that $\{a^p \mid p \text{ is a prime number}\}$ is not context-free.

Solution. Suppose q is the pumping length. Consider a string $s = a^{q'}$, where q' is a prime number greater than or equal to q. We further suppose that s can be pumped by dividing s as $uvxyz = a^i a^j a^k a^l a^{q'-i-j-k-l}$, where $j + l > 0$ and $j + k + l \leq q \leq q'$. We can pump s up to $a^i (a^j)^m a^k (a^l)^m a^{q'-i-j-k-l}$ for any $m > 1$, obtaining strings of the form $a^{jm+lm+q'-j-l} = a^{(j+l)(m-1)+q'}$. However, for $m = q' + 1$, $a^{(j+l)(m-1)+q'} = a^{(j+l)(q'+1-1)+q'} = a^{(j+l+1)q'}$ is clearly not in the language $\{a^p \mid p \text{ is a prime number}\}$. Thus, s cannot be pumped and the language is not context-free. $\square$