Suggested Solutions to Midterm Problems

(Compiled on May 15, 2002)

1. Let R_1 and R_2 be binary relations on a set A, i.e., $R_1, R_2 \subseteq A \times A$. Prove that, if R_1 and R_2 are equivalence relations, then $R_1 \cap R_2$ (the intersection of R_1 and R_2) is also an equivalence relation on A.

   Solution. We need to show that the relation $R = R_1 \cap R_2$ is (a) reflexive, (b) symmetric, and (c) transitive.
   (a) For every $x \in A$, $(x,x) \in R_1$ (or xR_1x) and $(x,x) \in R_2$ (or xR_2x) and it follows trivially that $(x,x) \in R_1 \cap R_2 = R$ (or xRx).
   (b) For every $x, y \in A$, $(x,y) \in R$, i.e., "$(x,y) \in R_1 \cap R_2$," if and only if "$(x,y) \in R_1$ and $(x,y) \in R_2$" if and only if "$(y,x) \in R_1$ and $(y,x) \in R_2$" if and only if "$(y,x) \in R_1 \cap R_2$," i.e., $(y,x) \in R$.
   (c) Let x, y, and z be elements of A. Suppose that $(x,y) \in R$ and $(y,z) \in R$, i.e., "$(x,y) \in R_1 \cap R_2$" and "$(y,z) \in R_1 \cap R_2$." It follows that "$(x,y) \in R_1$ and $(y,z) \in R_1$ and $(x,y) \in R_2$ and $(y,z) \in R_2$," which implies that "$(x,z) \in R_1$ and $(x,z) \in R_2$" and hence "$(x,z) \in R_1 \cap R_2$," i.e., $(x,z) \in R$. Therefore, for every $x, y, z \in A$, $(x,y) \in R$ and $(y,z) \in R$ implies $(x,z) \in R$. □

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

   Solution. See the attached. □

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

   Solution. See the attached. □

3. (a) Draw the state diagram of a DFA (with as few states as possible) that recognizes $\{w \in \{0,1\}^* \mid w \text{ doesn't contain } 000 \text{ or } 111 \text{ as a substring}\}$.

   Solution. See the attached. □

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

Solution.

$$\begin{array}{lcl} A & \to & 0A_0 \mid 1A_1 \mid \varepsilon \\ A_0 & \to & 0A_{00} \mid 1A_1 \mid \varepsilon \\ A_1 & \to & 0A_0 \mid 1A_{11} \mid \varepsilon \\ A_{00} & \to & 0A_x \mid 1A_1 \mid \varepsilon \\ A_{11} & \to & 0A_0 \mid 1A_x \mid \varepsilon \\ A_x & \to & 0A_x \mid 1A_x \end{array}$$

As A_x is a "dead-end" variable, the last three rules can be optimized as follows:

$$\begin{array}{lcl} A_{00} & \to & 1A_1 \mid \varepsilon \\ A_{11} & \to & 0A_0 \mid \varepsilon \end{array}$$

□

4. Write a regular expression for the language in Problem 3.

Solution. A regular expression for a seemingly simple regular language can be very complicated and, in general, not easy to obtain by resorting only to intuition. To derive systematically a regular expression for the language in Problem 3, we first construct a GNFA from the DFA and then convert the GNFA into a two-state GNFA. Below is the resulting regular expression (note that there exist other equivalent expressions).

$$\varepsilon \cup 0 \cup 00 \cup (1 \cup 01 \cup 001)(01 \cup 001 \cup 101 \cup 1001)^*(\varepsilon \cup 1 \cup 0 \cup 00 \cup 10 \cup 100)$$

□

5. Let $A = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and if } i = 2 \text{ then } j < k\}$. Show that A satisfies the pumping lemma for regular languages. What is the (smallest) pumping length of A?

Solution. We claim that 2 is a pumping length of A, i.e., any $s \in A$ such that $|s| \geq 2$ can be pumped. We separate the proof into four cases according to the form of s:

(a) $s = b^j c^k$, where $j, k \geq 0$ and $j + k \geq 2$. Assume that $j > 0$; the case when $j = 0$ and $k > 0$ can be handled analogously. We can divide s as $\varepsilon \cdot b \cdot b^{j-1}c^k$ ($|b| > 0$ and $|\varepsilon \cdot b| \leq 2$) and pump it up or down to $\varepsilon b^i b^{j-1} c^k \in A$ for any $i \geq 0$.

(b) $s = ab^j c^k$, where $j, k \geq 0$ and $j + k \geq 1$. Assume that $j > 0$; the case when $j = 0$ and $k > 0$ can be handled analogously. We can divide s as $a \cdot b \cdot b^{j-1}c^k$ and pump it up or down to $ab^i b^{j-1} c^k \in A$ for any $i \geq 0$.

(c) $s = a^2 b^j c^k$, where $0 \leq j < k$; note that $aa \notin A$. We can divide s as $a \cdot a \cdot b^j c^k$ and pump it up or down to $a \cdot a^i \cdot b^j c^k \in A$ for any $i \geq 0$.

(d) $s = aaab^j c^k$, where $j, k \geq 0$. We can divide s as $\varepsilon \cdot a^2 \cdot ab^j c^k$ and pump it up or down to $\varepsilon (a^2)^i ab^j c^k \in A$ (the string begins either with one a or at least three a's) for any $i \geq 0$.

The pumping length cannot be smaller, as $a \in A$ ($|a| \geq 1$) cannot be pumped. The only way to divide a is as $\varepsilon \cdot a \cdot \varepsilon$, but $\varepsilon \cdot a^2 \cdot \varepsilon = a^2 \notin A$. □

6. Show that, if G is a CFG in Chomsky normal form, then any string $w \in L(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of w.

   Solution. Given a CFG G in Chomsky normal form and an arbitrary string $w \in L(G)$ of length $n \geq 1$, we observe that any parse tree T for w has the following two properties:

   - T has exactly n leaves, since no leaves may be the empty string (given that $w \neq \varepsilon$).
   - Each of the n leaves is the only child of some internal node.
   - All internal nodes, except those that are parent of a leaf, have exactly two other internal nodes as children.

   For any CFG, the number of internal nodes of a parse tree equals the number of steps in the corresponding derivation. In particular, the number of T's internal nodes equals the number of steps in the corresponding derivation of w. To count T's internal nodes, we remove all the n "single-child" leaves of T to obtain a tree T'. T' will be a binary tree with n leaves and all its internal nodes will have two children. Any tree like T' can be shown to have exactly $n - 1$ internal nodes and hence $2n - 1$ nodes in total. This implies that T has $2n - 1$ internal nodes. As T represents any parse tree for w, it follows that any derivation of w takes $2n - 1$ steps. □

7. Draw the state diagram of a pushdown automaton (PDA) that recognizes the following language: $\{w\#x \mid w^{\mathcal{R}} \text{ is a substring of } x \text{ for } w, x \in \{0,1\}^*\}$. Please explain the intuition behind the PDA.

   Solution. See the attached state diagram.

   The PDA first reads and pushes w onto the stack; as a result, w^R sits on top of the stack. After reading #, the PDA nondeterministically throws away an initial portion of x and then cancels out symbol by symbol the rest of x and w^R on the stack. If the stack becomes empty, the PDA skips the rest of x and accepts. □

8. Prove that the class of context-free languages is not closed under either *intersection* or *complement*. (Hint: the class of context-free languages is known to be closed under union.)

   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. $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free. So, the class of context-free languages is not closed under intersection.

   $A_1 \cap A_2 = \overline{\overline{A_1} \cup \overline{A_2}}$. 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. We have shown in class that $\{1^{n^2} \mid n \geq 0\}$ is not regular. Is it context-free? Prove your answer.

Solution. It is not context-free as proven below.

Assume toward contradiction that p is the pumping length for $\{1^{n^2} \mid n \geq 0\}$. Consider a string $s = 1^{p^2}$ in the language. Suppose that s can be pumped by dividing s as $uvxyz = 1^i 1^j 1^k 1^l 1^{p^2-i-j-k-l}$, where $j + l > 0$ ($|vy| \geq 0$) and $j + k + l \leq p$ ($|vxy| \leq p$). If we pump s up to $1^i(1^j)^2 1^k (1^l)^2 1^{p^2-i-j-k-l} = 1^{i+2j+k+2l+p^2-i-j-k-l} = 1^{p^2+j+l}$. As $0 < j + l \leq p$, $p^2 < p^2 + j + l \leq p^2 + p < p^2 + 2p + 1 = (p + 1)^2$ and hence $1^i(1^j)^2 1^k(1^l)^2 1^{p^2-i-j-k-l}$ is not in $\{1^{n^2} \mid n \geq 0\}$. So, s cannot be pumped, a contradiction. $\square$

10. Find a regular language A, a non-regular but context-free language B, and a non-context-free language C over $\{0, 1\}$ such that $C \subseteq B \subseteq A$.

    Solution. $A = \{0^i 1^j 0^k \mid i, j, k \geq 0\}$ is regular. $B = \{0^i 1^j 0^k \mid i, j, k \geq 0 \text{ and } i \leq j\}$ is context-free but not regular. $C = \{0^i 1^j 0^k \mid i, j, k \geq 0 \text{ and } i \leq j \leq k\}$ is not context-free. It is apparent that $C \subseteq B \subseteq A$. $\square$

Appendix

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

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

  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,

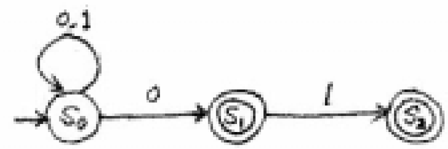
$$S \rightarrow \varepsilon$$

  is permitted if S is the start variable.

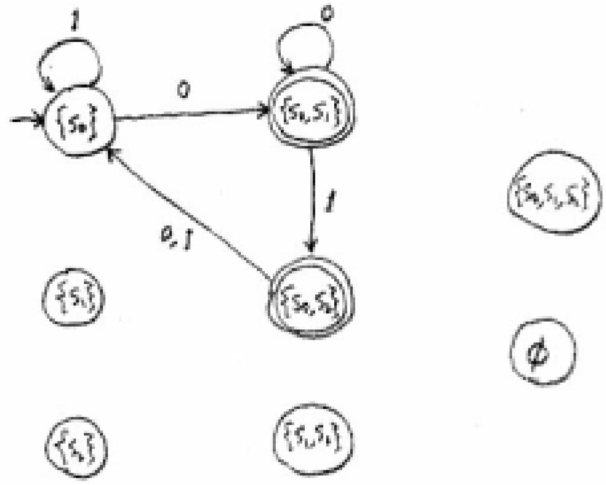
- If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided into three pieces, $s = xyz$, satisfying the conditions: (1) for each $i \geq 0$, $xy^i z \in A$, (2) $|y| > 0$, and (3) $|xy| \leq p$.

- If A is a context-free language, then there is a number p such that, if s is a string in A and $|s| \geq p$, then s may be divided into five pieces, $s = uvxyz$, satisfying the conditions: (1) for each $i \geq 0$, $uv^i xy^i z \in A$, (2) $|vy| > 0$, and (3) $|vxy| \leq p$.

2(a)

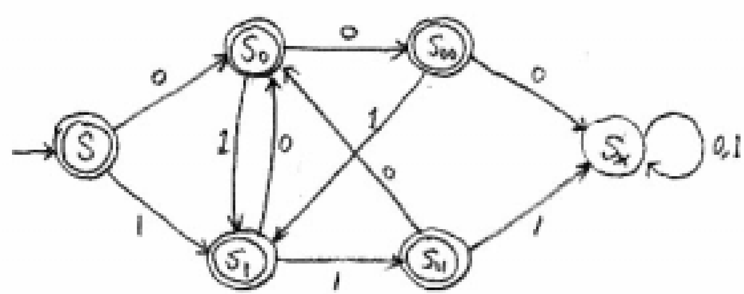


(b)



Note: transitions from unreachable states are omitted.

3(a)



7.