Problem Set #2

Due: Thursday, October 22, 2015

Problem 1 [30 pts] Give a PDA that recognizes each of the following languages. The input alphabet is $\Sigma = \{a, b, \#\}$. You may use whatever stack alphabet Γ you wish. Turn in your solutions to Blackboard as separate JFLAP files. Make sure you test your solutions! [*Hint: Parts b and c will require nondeterminism.*]

    a. $\{x\#y \mid x, y \in \{a, b\}^* \text{ and } x^{\mathcal{R}} \text{ is a substring of } y\}$. [*Hint:* $y = \{a, b\}^* x^{\mathcal{R}} \{a, b\}^*$]

    b. $\{x\#y \mid x, y \in \{a, b\}^* \text{ and } y^{\mathcal{R}} \text{ is a substring of } x\}$.

    c. $\{x_1\#x_2\# \cdots \#x_k \mid \text{each } x_i \in \{a, b\}^* \text{ and some } x_i = x_j^{\mathcal{R}}\}$.

Problem 2 [10 pts] Prove that the class of context-free languages is closed under reversal. [*Hint: Consider a CFG in CNF and use induction on the length of the strings.*]

Problem 3 [10 pts] Prove that every string $w \neq \varepsilon$ generated by a CFG in CNF of length n is derived in exactly $2n - 1$ steps. [*Hint: Use induction on the length of w.*]

Problem 4 [10 pts] Prove that the following CFG generates the language

$$\{xy \mid x, y \in \Sigma^*, \ |x| = |y|, \text{ and } x \neq y\}.$$

$$S \rightarrow AB \ \mid \ BA$$
$$A \rightarrow XAX \ \mid \ a$$
$$B \rightarrow XBX \ \mid \ b$$
$$X \rightarrow a \ \mid \ b$$

[*Hint: Consider m applications of the rule $A \rightarrow XAX$ and n applications of the rule $B \rightarrow XBX$ in the derivation of a string and note that each instance of an X gives you a single terminal. So*]

$$S \Rightarrow AB \overset{*}{\Rightarrow} X^m a X^m B \overset{*}{\Rightarrow} X^m a X^m X^n b X^n$$

where

$$X^i = \underbrace{XX \cdots X}_{i}.$$

All strings derived from the rightmost expression have length $2(m+n+1)$. Now divide such a string into two $m+n+1$ parts and show that the two parts differ in at least one position. There's a similar argument when the first step in the derivation is $S \Rightarrow BA$.]

Problem 5 [10 pts] We say that a CFG is in *forward-simple form* if all of its rules have the form $A \to tB$ or $A \to \varepsilon$ where A and B are variables (possibly the same variable), and t is a terminal. Prove that if G is a CFG in forward-simple form, then $L(G)$ is regular. [*Hint: Let $G = (V, \Sigma, R, S)$ be in FSF and build an NFA $N = (Q, \Sigma, \delta, q_0, F)$ that recognizes $L(G)$.*]

Problem 6 [10 pts] Call an alphabet *ordered* if it has an associated total order $\leqslant$.[1] For example, we can consider the alphabet $\Sigma = \{a, b\}$ an ordered alphabet by defining $a < b$ and in fact, we can turn any alphabet into an ordered alphabet by defining a total order. For strings over an ordered alphabet, define an operation on strings $\text{SORT}(w) = w'$ such that the symbols of w' are the symbols of w in sorted order. For example, $\text{SORT}(aababab) = aaaabbb$. (Note that SORT is *not* a homomorphism.)

For a language A defined over a sorted alphabet, define

$$\text{SORT}(A) = \{\text{SORT}(w) \mid w \in A\}.$$

For example, $\text{SORT}(\Sigma^*) = a^*b^*$, for $\Sigma = \{a, b\}$.

After all of that set up, here's the problem: Prove that if A is a regular language defined over the sorted alphabet $\Sigma = \{a, b\}$, then $\text{SORT}(A)$ is context-free. [*Hint: Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes A, build a CFG that generates $\text{SORT}(A)$. Your CFG will have a form similar to, but different from forward-simple form (see Problem 5).*]

Problem 7 [Extra credit: 10 pts] Give a regular language A defined over the sorted alphabet $\Sigma = \{a, b, c\}$ such that $\text{SORT}(A)$ is not context-free. Prove that $\text{SORT}(A)$ is not context-free. You may use the fact that $\{a^n b^n c^n \mid n \geqslant 0\}$ is not context-free without proof.

---

[1]See https://en.wikipedia.org/wiki/Total_order for the definition of a total order.