CSCI 510, Fall 2016, Homework # 2

SOLUTION SKETCHES

Due date: Friday, October 21, Midnight

1. Let $A/B = \{w|wx \in A \text{ for some } x \in B\}$. Show that if A is context free and B is regular then A/B is context-free.

   Solution: If A is context free and B is regular we can assume there is a PDF for A and a DFA for B. We will build a PDA for A/B by first running the A machine over the string, and, at the end, running a machine that does not consume input, but nondeterministically runs through the B machine until it gets to an accepting state.

   The devil is in the details. Let

   $$P_A = \{Q_A, \Sigma_A, \Gamma_A, \delta_A, q_{A0}, F_A\}$$

   be a PDA for A, and let

   $$D_B = \{Q_B, \Sigma_B, \delta_B, q_{B0}, F_B\}$$

   be a DFA for B. We now build a PDA for A/B,

   $$P_{A/B} = \{Q_{A/B}, \Sigma_{A/B}, \Gamma_{A/B}, \delta_{A/B}, q_{A/B0}, F_{A/B}\}$$

   as follows:

   $$\begin{aligned} Q_{A/B} &= Q_A \cup (Q_A \times Q_B) \\ \Sigma_{A/B} &= \Sigma_A \cup \Sigma_B \\ \Gamma_{A/B} &= \Gamma_A \\ q_{A/B0} &= q_{A0} \\ F_{A/B} &= \{(f_a, f_b)|f_a \in F_A \text{ and } f_b \in F_B\} \end{aligned}$$

   $\delta_{A/B}$ will consist of all transitions δ_A, together with all rules of the following types two types:

   (a) For each $q_A \in Q_A$ add the rule

   $$\delta_{A/B}(q_A, \varepsilon, \varepsilon) = ((q_A, q_{B0}), \varepsilon)$$

   This will allow us to jump, at the end of the input, from the Q_A part of $P_{A/B}$ to the $Q_A \times Q_B$ part, and get started looking for the (invisible) "B" part of the string. The input and stack are untouched.

   (b) Whenever

   $$\delta_A(q_{A1}, b, c) = (Q_{A2}, c)$$

   and

   $$\delta_B(q_{B1}, d) = q_{B2}$$

   then add the rule

   $$\delta_{A/B}((q_{A1}, q_{B1}), \epsilon, b) = ((q_{A2}, q_{B2}), c)$$

   This will allow us to simulate both the operation of the A machine and the B machine simultaneously, and have the same effect on the stack that the A machine would have had if the "B" part of the string had been on the input.

To accept, we must end up accepting the pseudo-extension to the string with the B machine, and accept the string together with the pseudo-extension with the A machine.

2. For any language A, let *suffix*$(A) = \{v | uv \in A \text{ for some string } u\}$. Show that the class of context-free languages is closed under the *suffix* operation.

   Solution:

   This solution is very similar to the previous one, except that we run a "no-input" PDA with ε transitions (similar to the ones in the A PDA) that sets up what the stack would have been had we run u (a random string), then nondeterministically jump, using ε transitions, to the full machine to finish the job. Lots of details omitted here.

3. 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. Give a proof by induction.

   Solution

   First we prove a handy lemma by induction:

   Lemma Given a derivation of w length ℓ:

$$s_0 \Rightarrow s_1 \Rightarrow s_2 \Rightarrow s_3 \Rightarrow \ldots \Rightarrow s_i \Rightarrow \ldots \Rightarrow s_\ell$$

   where s_1 is the start symbol, and $s_\ell = w$. For each $i \in 0 \ldots \ell$ let j_i be the number of times an $A \to a$ type rule is applied in the dervation up to s_i, and k_i be the number of times a $A \to BC$ type rule is applied in the derivation up to s_i. Then

$$|s_i| = k_i + 1 \tag{1}$$

   Proof of Lemma (induction)

   Base: $i = 0$. Then the derivation is just s_0, which is the start symbol with length 1, and no rules have been applied, so $k_0 = 0$, and $|s_0| = 1 = k_0 + 1$.

   Step: Assume $|s_i| = k_i + 1$ and consider the step from s_i to s_{i+1}. We have two cases:

   (a) The rule applied at step i is of the form $A \to a$. Then $|s_{i+1}| = |s_i|$ and $k_{i+1} = k_i$ and so $|s_{i+1}| = k_{i+1} + 1$

   (b) The rule applied at step i is of the form $A \to BC$. Then $|s_{i+1}| = |s_i| + 1$ and $k_{i+1} = k_i + 1$ and so $|s_{i+1}| = k_{i+1} + 1$

   ■

   Proof

   Using the same notation as in the lemma, clearly, for $i = 0 \ldots \ell$:

$$i = j_i + k_i \tag{2}$$

   Further, the last element in the derivation is all terminals. Each terminal is produced by one application of a rule of the form $A \to a$, and so $j_\ell = n = |w|$. Putting this together with our lemma gives

$$\begin{aligned} \ell &= j_\ell + k_\ell && (3)\\ &= n + k_\ell && (4)\\ &= n + |s_\ell| - 1 && (5)\\ &= n + n - 1 && (6)\\ &= 2n - 1 && (7) \end{aligned}$$

   ■