Formal Languages and Automata Theory

Second Long Test

Time: 50 minutes | 12–March–2021 | Maximum marks: 40

---

1. (a) Consider the following grammar with the start symbol S:

$$S \to abScB \mid \varepsilon$$
$$B \to bB \mid b$$

What language does this grammar generate? Is this grammar ambiguous? **(4 + 1)**

Solution $\{(ab)^n(cb^+)^n \mid n \geqslant 0\}$. Here, the different instances of b^+ contain independently many occurrences of b. Unambiguous.

(b) Prove that the language

$$L_1 = \left\{a^l b^m c^n \mid l < m \text{ and } l < n\right\}$$

is not context-free. **(5)**

Solution Assume that L_1 is context-free. Let k be a pumping-lemma constant for L_1. Consider the string $z = a^k b^{k+1} c^{k+1}$. The demon breaks z into $uvwxy$ such that $z_i = uv^iwx^iy \in L_1$ for all $i \geqslant 0$. If v or x spans across the block boundaries, then for $i = 2$, z_i is not of the form $a^*b^*c^*$. So v and x must be in individual blocks. First, suppose that v is non-empty. If v is in the block of a's, the substring x cannot be in the block of c's by the length condition $|vwx| \leqslant k$. Therefore for $i = 2$, the c's in z_i cannot be more numerous than the a's. Otherwise if v is in the block of b's or c's, then there will not be enough b's or c's in z_0. Finally, if v is empty, x must be non-empty. If x is in the block of a's, take $i = 2$. If x is in the block of b's or c's, take $i = 0$.

2. (a) Design an unambiguous CFG for the set L_2 of all non-palindromes over $\{a, b\}$. Assume that ε is *not* in the language. (Partial credit if the grammar is ambiguous.) **(6)**

Solution The following unambiguous grammar with the start symbol S generates L_2.

$$\begin{aligned}
S &\to aSa \mid bSb \mid T \\
T &\to aRb \mid bRa \\
R &\to XRX \mid X \mid \varepsilon \\
X &\to a \mid b
\end{aligned}$$

(b) Design a PDA whose language is $\sim L_2$ (the set of all palindromes over $\{a, b\}$). **(4)**

Solution You can use the CFG-to-PDA conversion on the grammar $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$, or design a PDA from the scratch. Notice that the nondeterministic switch from the first half of the input to the second half may be triggered by ε (for even-length palindromes) or by a symbol in $\{a, b\}$ (for odd-length palindromes).

3. (a) Let $P = (Q, \Sigma, \Gamma, \perp, \delta, s, F)$ be a PDA which never pops from its stack, that is, every transition of P is of the form $((p, a, A), (q, \gamma A))$, where $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $A \in \Gamma$, and $\gamma \in \Gamma^*$. Since P cannot empty its stack, it accepts by final state. Prove that $\mathscr{L}(P)$ is regular. **(5)**

Solution The idea is to remember the top of the stack in the state. Since P never pops from its stack, each transition of P uniquely identifies the next top of the stack, and therefore a finite automaton can simulate the working of P perfectly. Formally, we construct an NFA $N = (Q', \Sigma, \Delta', S', F')$ as follows (since P is non-deterministic, N would be so too). Take $Q' = Q \times \Gamma$, $S' = \{(s, \perp)\}$, and $F = \{(f, A) \mid f \in F \text{ and } A \in \Gamma\}$. For each transition $((p, a, A), (q, \gamma A))$ of P, include the transition (q, B) in $\Delta((p, A), a)$, where B is A if $\gamma = \varepsilon$, or B is the first symbol of γ if $\gamma \neq \varepsilon$. It is straightforward to establish that $\mathscr{L}(P) = \mathscr{L}(N)$.

(b) Let G be a CFG. A production $A \to \gamma$ is said to be of degree d if the number of non-terminal symbols in γ is exactly d. For example, the production $S \to aTTbcUabSc$ is of degree four (the lower-case letters are terminal symbols, and the upper-case letters are non-terminal symbols). G is said to be of degree d if the maximum degree of the productions in G is d. For example, a CFG for the language $\left\{x \in \{a,b\}^* \mid \#a(x) = 2 \times \#b(x)\right\}$ consists of the productions $S \to \varepsilon \mid aB \mid bAA$, $A \to aS \mid bAAA$, and $B \to bA \mid aBB \mid aSbS$. This grammar is of degree three (because of the production $A \to bAAA$, the other productions having degrees $\leqslant 2$). Prove that every CFL has a CFG of degree two. **(5)**

Solution It suffices to show that every production of degree $k \geqslant 3$ can be rewritten as a sequence of productions of degree two. Let $A \to \alpha_0 B_1 \alpha_1 B_2 \alpha_3 \ldots \alpha_{k-1} B_k \alpha_k$ be such a production, where B_i are non-terminal symbols, and α_j are strings in Σ^*. Introduce $k-2$ new non-terminal symbols $U_1, U_2, U_3, \ldots, U_{k-2}$ and the new productions:

$$\begin{array}{rcl} A & \to & \alpha_0 B_1 U_1 \\ U_1 & \to & \alpha_1 B_2 U_2 \\ U_2 & \to & \alpha_2 B_3 U_3 \\ & \vdots & \\ U_{k-3} & \to & \alpha_{k-3} B_{k-2} U_{k-2} \\ U_{k-2} & \to & \alpha_{k-2} B_{k-1} \alpha_{k-1} B_k \alpha_k \end{array}$$

Alternatively, note that any grammar in the Chomsky normal form is of degree (at most) two. But such grammars cannot generate ε, so you need to add the production $S \to \varepsilon$ of degree zero if ε is in the language.

4. Let Σ_1 and Σ_2 be disjoint alphabets, $\Sigma = \Sigma_1 \cup \Sigma_2$, and $L \subseteq \Sigma^*$. Denote, by L_1, the language over Σ_1 obtained by deleting all symbols of Σ_2 from the strings in L. Likewise, let L_2 denote the language over Σ_2 obtained by deleting all symbols of Σ_1 from the strings in L. For example, if $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, and $L = \left\{abab^2ab^3 \ldots ab^n \mid n \geqslant 1\right\}$, then we have $L_1 = \left\{a^n \mid n \geqslant 1\right\}$, and $L_2 = \left\{b^{n(n+1)/2} \mid n \geqslant 1\right\}$.

Prove/Disprove the statements in each of the following two parts. If you use any language that is not covered in the lectures/tutorials, it is your duty to *prove* the language to be a DCFL or not.

(a) If L is a DCFL, then both L_1 and L_2 must be DCFL. **(5)**

Solution *False*. Idea: The existence of symbol(s) from Σ_2 may help a PDA for L to take deterministic decisions, whereas a PDA for L_1 cannot leverage the hints provided by the symbol(s) from Σ_2.

Take $\Sigma_1 = \{a,b,c\}$, $\Sigma_2 = \{\$,\#\}$, and $L = \{\$a^ib^jc^k \mid i \neq j\} \cup \{\#a^ib^jc^k \mid j \neq k\}$. It is easy to construct a DPDA for L, since the first symbol fixes the inequality to verify. But we have seen that $L_1 = \{a^ib^jc^k \mid i \neq j\} \cup \{a^ib^jc^k \mid j \neq k\}$ is not a DCFL. In this example, $L_2 = \{\$,\#\}$ is a DCFL, but this does not matter.

(b) If both L_1 and L_2 are DCFL, then L must be a DCFL. **(5)**

Solution *False*. Idea: Removal of symbols from the strings in L may "simplify" the language.

Take $\Sigma_1 = \{a,b\}$, $\Sigma_2 = \{c\}$, and $L = \{a^nb^nc^n \mid n \geqslant 0\}$. We have $L_1 = \{a^nb^n \mid n \geqslant 0\}$, and $L_2 = \{c^n \mid n \geqslant 0\}$. Clearly, L_1 and L_2 are DCFL, whereas L is not even a CFL.
