CS320: Problems and Solutions for Day 9, Winter 2023

Problem 1 Let L be a language over alphabet $\Sigma = \{a, b, c, d, e\}$, defined as follows:

$$L = \{a^{m+1}b^{2n}c^{k+2}d^{3\ell}e^{j+3} \mid n = 2m, \ell = k + j\}$$

where $m, n, k, \ell, j \geq 0$.

(a) Write a complete formal definition of a context-free grammar G that generates language L. If such grammar does not exist, prove it.

Answer: The template for strings of L is:

$$a^{m+1}b^{4m}c^{k+2}d^{3k}d^{3j}e^{j+3}, \text{ where } m, k, j \geq 0$$

whence the grammar: $G = \{V, \Sigma, P, S\}$, where:
$\Sigma = \{a, b, c, d, e\}$, $V = \{S, A, B, D\}$, and P is:

$$\begin{aligned}
&S \to ABD \\
&A \to aAbbbb \mid a \\
&B \to cBddd \mid cc \\
&D \to dddDe \mid eee
\end{aligned}$$

(b) Let $\mathcal{S}$ be a class of languages over alphabet $\{a, b, c, d, e\}$, defined as follows:

> Language L is a member of $\mathcal{S}$ if and only if L is generated by some context-free grammar, but there does not exist a pushdown automaton that accepts L.

What is the cardinality of $\mathcal{S}$? Explain your answer briefly.

Answer:

$$|\mathcal{S}| = 0, \text{ since } \mathcal{S} = \emptyset$$

Class $\mathcal{S}$ is empty, since every language generated by some context-free grammar is also accepted by some pushdown automaton. In fact, this pushdown automaton is obtained by an algorithmic conversion of the original context-free grammar.

Problem 2 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$\begin{aligned}
Q &= \{q, r, s, t\} \\
\Sigma &= \{a, b\} \\
\Gamma &= \{A\} \\
F &= \{t\}
\end{aligned}$$

and the transition function δ is defined as follows:

$$\begin{aligned}
&[q, a, \lambda, r, A] \\
&[r, a, \lambda, q, \lambda] \\
&[t, b, \lambda, s, A] \\
&[s, b, \lambda, t, \lambda] \\
&[q, \lambda, \lambda, t, \lambda]
\end{aligned}$$

(Recall that M is defined so as to accept by final state and empty stack.)

Write a regular expression that defines L. If such regular expression does not exist, prove it.

Answer:

$$\boldsymbol{\lambda}$$

Problem 3 Let L_1 be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$\begin{aligned} Q &= \{q, r\} \\ \Sigma &= \{a, b, c\} \\ \Gamma &= \{A\} \\ F &= \{r\} \end{aligned}$$

and the transition function δ is defined as follows:

$$\begin{array}{l} [q, a, \lambda, q, A] \\ [q, c, \lambda, r, \lambda] \\ [r, b, A, r, \lambda] \end{array}$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L_1. If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, S\}$, where:
$\Sigma = \{a, b, c\}$, $V = \{S\}$, and the rule set P is:

$$S \to aSb \mid c$$

(b) Describe the algorithm that should be employed by a program that solves the following problem:

INPUT: An arbitrary string x over Σ.

QUESTION: Does x belong to L_1?

Explain your answer. If such algorithm does not exist, prove it.

Answer: This algorithm simulates the operation of the pushdown automaton M defined in part (a). Since M accepts when x belongs to L_1 and rejects when x does not belong to L_1, our algorithm says yes when M accepts, and says no when M rejects.

Problem 4 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$\begin{aligned} Q &= \{q, r, s, t, v\} \\ \Sigma &= \{a, b, c, d\} \\ \Gamma &= \{A, B, D\} \\ F &= \{v\} \end{aligned}$$

and the transition function δ is defined as follows:

$$\begin{array}{l} [q, a, \lambda, q, A] \\ [r, b, A, r, \lambda] \\ [s, c, \lambda, s, B] \\ [t, a, B, t, \lambda] \\ [v, d, \lambda, v, D] \\ [v, d, D, v, \lambda] \\ [q, \lambda, \lambda, r, \lambda] \\ [r, \lambda, \lambda, s, \lambda] \\ [s, \lambda, \lambda, t, \lambda] \\ [t, \lambda, \lambda, v, \lambda] \end{array}$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L. If such grammar does not exist, prove it.

Answer: The template for words accepted by M is:

$$a^n b^n c^m a^m (dd)^k$$

whence the grammar: $G = \{V, \Sigma, P, S\}$, where
$\Sigma = \{a, b, c, d\}$, $V = \{S, A, B, D\}$, and P comprises:

$$\begin{aligned}
&S \rightarrow ABD \\
&A \rightarrow aAb \mid \lambda \\
&B \rightarrow cBa \mid \lambda \\
&D \rightarrow DD \mid \lambda \mid dd
\end{aligned}$$

(b) Let $\mathcal{T}$ be a set of strings over alphabet $\{a, b, c\}$, defined as follows:

> String w is a member of $\mathcal{T}$ if and only if w is accepted by the pushdown automaton M and the length of w is greater than 6.

What is the cardinality of $\mathcal{T}$? Explain your answer briefly.

Answer:

$$|\mathcal{T}| = \aleph_0$$

Set $\mathcal{T}$ is infinite and countable. To see that $\mathcal{T}$ is infinite, observe that it contains all the strings of the infinite language L, except possibly only finitely many of them whose length does not exceed 6. (Question: Which?) To see that $\mathcal{T}$ is countable, recall that the entire set of strings $\{a, b, c\}^*$ is countable.

Problem 5 Let L be the language accepted by the pushdown automaton: $M = (Q, \Sigma, \Gamma, \delta, q, F)$, where

$$\begin{aligned}
&Q = \{q, r\} \\
&\Sigma = \{a, b, c\} \\
&\Gamma = \{A, Z\} \\
&F = \{q\}
\end{aligned}$$

and the transition function δ is defined as follows:

$$\begin{aligned}
&[q, \lambda, \lambda, r, Z] \\
&[q, \lambda, Z, q, \lambda] \\
&[r, a, Z, r, ZA] \\
&[r, c, A, r, \lambda] \\
&[r, b, Z, q, \lambda]
\end{aligned}$$

(Recall that M is defined so as to accept by final state and empty stack. Furthermore, if an arbitrary stack string, say $X_1 \ldots X_n \in \Gamma^*$, where $n \geq 2$, is pushed on the stack by an individual transition, then the left-most symbol X_1 is pushed first, while the right-most symbol X_n is pushed last.

(a) Write a regular expression that defines L. If such a regular expression does not exist, prove it.

Answer:

$$\boldsymbol{((ac)^*b)^*}$$

(b) Write a complete formal definition of a context-free grammar G that generates L. If such a grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$, $V = \{S, A\}$, and the production set P is:

$$\begin{aligned}
&S \rightarrow SS \mid \lambda \mid Ab \\
&A \rightarrow acA \mid \lambda
\end{aligned}$$

Problem 6 Let L be the language accepted by the pushdown automaton: $M = (Q, \Sigma, \Gamma, \delta, q, F)$, where $Q = \{q, r\}$, $\Sigma = \{a, b, c, d\}$, $\Gamma = \{A, B\}$, $F = \{r\}$, and the transition function δ is defined as follows:

$$\begin{aligned}
&[q, a, \lambda, q, A] \\
&[q, b, \lambda, q, B] \\
&[q, c, \lambda, q, A] \\
&[q, \lambda, \lambda, r, \lambda] \\
&[r, d, A, r, \lambda] \\
&[r, b, B, r, \lambda]
\end{aligned}$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L. If such a grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c, d\}$,
$V = \{S\}$, and P is:

$$S \to aSd \mid cSd \mid bSb \mid \lambda$$

(b) Write a complete formal definition of a context-free grammar G_1 that generates L^*. If such a grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, T)$, where $\Sigma = \{a, b, c, d\}$,
$V = \{S, T\}$, and P is:

$$T \to TT \mid \lambda \mid S$$
$$S \to aSd \mid cSd \mid bSb \mid \lambda$$

Problem 7 Let L be the language accepted by the pushdown automaton: $M = (Q, \Sigma, \Gamma, \delta, q, F)$, where $Q = \{q, r, s\}$, $\Sigma = \{a, b, d\}$, $\Gamma = \{A\}$, $F = \{s\}$, and the transition function δ is defined as follows:

$$[q, a, \lambda, q, A]$$
$$[r, d, \lambda, r, A]$$
$$[s, b, A, s, \lambda]$$
$$[q, \lambda, \lambda, r, \lambda]$$
$$[r, \lambda, \lambda, s, \lambda]$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L. If such a grammar does not exist, prove it.

Answer: Note that:

$$L = \{a^m d^n b^{m+n} \mid m, n \geq 0\}$$

whence the grammar: $G = (V, \Sigma, P, S)$, where
$\Sigma = \{a, b, d\}$, $V = \{S, A\}$, and P is:

$$S \to aSb \mid A$$
$$A \to dAb \mid \lambda$$

(b) Write a complete formal definition of a *regular* context-free grammar G_1 that generates L. If such a grammar does not exist, prove it.

Answer: The language:

$$L = \{a^m d^n b^{m+n} \mid m, n \geq 0\}$$

is not regular, and there does not exist a regular grammar to generate it.

To prove this, assume the opposite, that L is regular. Let η be the constant as in the Pumping Lemma for L. Let $m > \eta$; then the word:

$$a^m b^m$$

belongs to L, as it is obtained from the general template by setting $n = 0$.

In any "pumping" decomposition such that:

$$a^m b^m = uvx$$

we have:

$$|uv| \leq \eta < m$$

Hence, the "pumping" substring v consists entirely of a's, say $v = a^\ell$. Recall that $\ell > 0$, since the "pumping" substring cannot be empty. By the pumping, every word of the form uv^ix, $i \geq 0$, belongs to L. However, such a word is of the form:

$$w_1 = a^{m+(i-1)\ell} b^m$$

Observe that the total number of a's and d's in this word is equal to $m + (i-1)\ell$. Since $m + (i-1)\ell > m$ whenever $i > 1$, word w_1 has more a's and d's than is appropriate for its number of b's. Hence, $w_1 \notin L$, which is a contradiction.

Problem 8 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$Q = \{q, r, s\}$$
$$\Sigma = \{a, b, c\}$$
$$\Gamma = \{A, B\}$$
$$F = \{s\}$$

and the transition function δ is defined as follows:

$$[q, a, \lambda, q, \lambda]$$
$$[q, b, \lambda, q, B]$$
$$[q, c, \lambda, q, \lambda]$$
$$[q, \lambda, \lambda, r, \lambda]$$
$$[r, \lambda, B, s, \lambda]$$
$$[s, \lambda, B, s, \lambda]$$

(Recall that M is defined so as to accept by final state and empty stack.)

Write a complete formal definition of a context-free grammar that generates $\overline{L}$ (the complement of L). If such a grammar does not exist, prove it.

Answer: Observe that L contains exactly those strings over $\{a, b, c\}$ that contain at least one occurrence of the letter b. Hence, its complement is represented by the regular expression $(\boldsymbol{a} \cup \boldsymbol{c})^*$, which corresponds to the grammar $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$ is the set of terminals; $V = \{S\}$ is the set of variables; S is the start symbol, and the production set P is:

$$S \to aS \mid cS \mid \lambda$$