NTIN071 A&G: Tutorial 10 – Pushdown automata

Solve 1a-h, 2ab, 3a first (the rest is for practice).

Problem 1 (Constructing PDA)**.** Design pushdown automata for the following languages. (Acceptance can be either by final state or by empty stack, in some cases construct both.)

(a) $L = \{w \mid w \in \{0,1\}^*, |w|_1 \geq 3\}$

(b) $L = \{ww^R \mid w \in \{0,1\}^*\}$

(c) $L = \{w \in \{(,)\}^* \mid w \text{ is correctly parenthesized}\}$

(d) $L = \{w \in \{0,1\}^* \mid w = w^R\}$

(e) $L = \{a^i b^j c^k \mid i = j \text{ or } j = k\}$

(f) $L = \{a^i b^j c^k \mid i + j = k\}$

(g) $L = \{a^{2n} b^{3n} \mid n \geq 0\}$

(h) $L = \{w \in \{0,1\}^* \mid |w|_0 = |w|_1\}$

(i) $L = \{u2v \mid u, v \in \{0,1\}^* \text{ and } |u| \neq |v|\}$

(j) $L = \{w \in \{(,),[,]\}^* \mid w \text{ is correctly parenthesized}\}$

Problem 2 (Acceptance by final state vs. empty stack)**.** Convert selected PDA constructed in the previous problem from acceptance by final state to acceptance by empty stack, and vice versa. (Try both constructions.)

Problem 3 (CFG to PDA)**.** For a given context-free grammar G construct pushdown automata P_1, P_2 such that $L(G) = N(P_1) = L(P_2)$.

(a) $G = (\{S, T, X\}, \{a, b\}, \mathcal{P}, S)$

$$\begin{aligned}\mathcal{P} = \{&S \to aTXb,\\ &T \to XTS \mid \epsilon,\\ &X \to a \mid b\}\end{aligned}$$

(b) $G = (\{S, T, X\}, \{(,), *, +, , 1\}, \mathcal{P}, S)$

$$\begin{aligned}\mathcal{P} = \{&S \to S + T \mid T,\\ &T \to T * X \mid X,\\ &X \to 1 \mid (S)\}\end{aligned}$$

For a reasonably long word $w \in L(G)$ find its leftmost derivation from G and simulate the automaton P_1 on the input w.

Problem 4 (PDA to CFG)**.** Convert selected (small) pushdown automata constructed in Problem 1 to context-free grammars. For a reasonably long word w accepted by the automaton find a leftmost derivation of w from the grammar.

Bonus: Context grammars

Problem 5 (A context grammar)**.** Consider the grammar $G = (\{S, A, B, C\}, \{a, b, c\}, S, P)$, where

$$\begin{aligned}P = \{&S \to aSBC \mid aBC, B \to BBC, C \to CC, CB \to BC,\\ &aB \to ab, bB \to bb, bC \to bc, cC \to cc\}\end{aligned}$$

What language does it generate? Is G a context grammar? If not, find an equivalent context grammar.