1 Constructing PDAs ($\star$)

Construct PDAs to recognise the following languages:

(a) $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i,j,k \geq 0,\ i=j \text{ or } i=k\}$.

(b) The language $\mathcal{L}_{PB}$ of strings of properly nested parentheses `()` and brackets `[]`. For example, the following strings are in $\mathcal{L}_{PB}$:

`([]())`, `[(([()[]])]`, `[]`

but the following strings are not:

`([](])`, `[(]()[]])]`, `[(])`

(c) ($\star\star$) $\{\mathtt{a}^n\mathtt{b}^m \mid n \leq m \leq 2n\}$.

(d) $\{x\#y \mid x,y \in \{\mathtt{0},\mathtt{1}\}^* \text{ and } |x| \neq |y|\}$.

(e) $\{xy \mid x,y \in \{\mathtt{0},\mathtt{1}\}^* \text{ and } x \neq y\}$.

(f) $\{xy \mid x,y \in \{\mathtt{0},\mathtt{1}\}^+ \text{ and } x \neq y\}$.

(Constructing a PDA directly for these languages is likely to be easier than giving a CFG and then transforming it to a PDA.)

2 CFG2PDA ($\star$)

Consider the following CFG (you've seen it before).

$$\begin{aligned} E &\to E + T \mid T \\ T &\to T \times F \mid F \\ F &\to (E) \mid \mathtt{a} \end{aligned}$$

Convert it to a PDA using the algorithm described in the lectures.
Describe the computation of the PDA on the following strings.

(1.) `a+a` (2.) `((a))`

3 Ambiguous ambiguity ($\star$)

Give a context-free grammar that generates the language

$$\mathcal{L} = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i,j,k \geq 0,\ i=j \text{ or } i=k\}$$

from the first question above. Is your grammar ambiguous? If so, give an example of an ambiguous derivation; if not, give an argument for why not.

4 Deterministic PDAs ($\star$)

Construct a DPDA for the language $\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$.

5 Being complementary about CFLs ($\star\star$)

Let G be the following context-free grammar.

$$\begin{aligned} S &\rightarrow \mathtt{a}S\mathtt{b} \mid \mathtt{b}T \mid T\mathtt{a} \\ T &\rightarrow \mathtt{b}T \mid \mathtt{a}T \mid \varepsilon \end{aligned}$$

Give a simple description of the language $L(G)$ generated by G. Use this to give a CFG for the complement of $L(G)$ (i.e. the set of all strings not in $L(G)$).

6 Reflections on regular languages ($\star\star$)

For any language $\mathcal{L}$, define the language

$$\text{reflect}(\mathcal{L}) = \{ww^{\mathcal{R}} \mid w \in \mathcal{L}\}$$

which consists of all strings in $\mathcal{L}$ concatenated with their reversals. Describe a construction which, for any regular language $\mathcal{L}$, gives a PDA which recognises reflect($\mathcal{L}$).