Syntax and Semantics:
Exercise Session 7

Recall that. *A CFG is in Chomsky normal form if every production is of the following form*

$$A \to BC \qquad \text{or} \qquad A \to a$$

where A, B, C are non-terminals, $a \in \Sigma$ and B, C are not the initial non-terminal. In addition $S \to \varepsilon$ is permitted only if S is the initial non-terminal.

Recall that. *A DFA $(Q, \Sigma, \gamma, q_0, F)$ can be encoded to a PDA $(Q, \Sigma, \Gamma, \delta, q_0, F)$ with* empty stack alphabet *(i.e., $\Gamma = \emptyset$) and transition function $\delta \colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \wp(Q \times \Gamma_\varepsilon)$ defined, for arbitrary $s \in Q$ and $\sigma \in \Sigma_\varepsilon$ as*

$$\delta(q, \sigma, \varepsilon) = \{(q', \varepsilon) \mid q' \in \gamma(q, \sigma)\}\,.$$

This is an alternative way to prove that context-free languages are a superset of the regular languages (i.e., any regular language is a context-free language). It is worth to recall that the converse inclusion does not hold.

Exercise 1.
For each of the CFGs below find an equivalent CFG in Chomsky normal form. The grammar G_1 produces mathematical expressions with the alphabet $\Sigma = \{a, +, \times, (,)\}$.

$$\begin{aligned} G_1 \colon\ & E \to E + T \mid T \\ & T \to T \times F \mid F \\ & F \to (E) \mid a \end{aligned} \qquad\qquad \begin{aligned} G_2 \colon\ & R \to XRX \mid S \\ & S \to aTb \mid bTa \\ & T \to XTX \mid X \mid \varepsilon \\ & X \to a \mid b \mid \varepsilon \end{aligned}$$

Solution 1.
To convert G_1 into Chomsky normal form we first introduce a new start variable (step 1):

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

There are no ε-rules to remove (step 2), so next we remove rules of type $A \to B$ (step 3)

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

and then

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

finally

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

The next rewrites will make use of step 4, where we replace rules of type $A \to u_1, u_2 ... u_k$, where $k > 2$. First we make a new rule for $+ T$

$$\begin{aligned} &S \to EE_1 \mid T \times F \mid (E) \mid a \\ &E \to EE_1 \mid T \times F \mid (E) \mid a \\ &T \to T \times F \mid (E) \mid a \\ &F \to (E) \mid a \\ &E_1 \to + T \end{aligned}$$

And now a rule for $\times F$

$$
\begin{aligned}
S &\to EE_1 \mid TE_2 \mid (E) \mid a \\
E &\to EE_1 \mid TE_2 \mid (E) \mid a \\
T &\to TE_2 \mid (E) \mid a \\
F &\to (E) \mid a \\
E_1 &\to + T \\
E_2 &\to \times F
\end{aligned}
$$

A rule for E)

$$
\begin{aligned}
S &\to EE_1 \mid TE_2 \mid (P_1 \mid a \\
E &\to EE_1 \mid TE_2 \mid (P_1 \mid a \\
T &\to TE_2 \mid (P_1 \mid a \\
F &\to (P_1 \mid a \\
E_1 &\to + T \\
E_2 &\to \times F \\
P_1 &\to E\ )
\end{aligned}
$$

In step 5 we eliminate rules of type $A \to uB$, replacing them with $A \to UB$ and $U \to u$ (multiple rewrites shown at once here):

$$
\begin{aligned}
S &\to EE_1 \mid TE_2 \mid P_2P_1 \mid a \\
E &\to EE_1 \mid TE_2 \mid P_2P_1 \mid a \\
T &\to TE_2 \mid P_2P_1 \mid a \\
F &\to P_2P_1 \mid a \\
E_1 &\to AT \\
E_2 &\to MF \\
P_1 &\to EP_3 \\
P_2 &\to\ (\\
P_3 &\to\ ) \\
M &\to \times \\
A &\to +
\end{aligned}
$$

In what follows, the grammer G_2 is rewritten to Chomsky normal form.

$$
\begin{aligned}
&R \to XRX \mid S \\
&S \to aTb \mid bTa \\
&T \to XTX \mid X \mid \varepsilon \\
&X \to a \mid b \mid \varepsilon
\end{aligned}
$$

Insert new start variable:

$$
\begin{aligned}
&S_0 \to R \\
&R \to XRX \mid S \\
&S \to aTb \mid bTa \\
&T \to XTX \mid X \mid \varepsilon \\
&X \to a \mid b \mid \varepsilon
\end{aligned}
$$

Remove ε-rules:

$$
\begin{aligned}
&S_0 \to R \\
&R \to XRX \mid RX \mid XR \mid S \\
&S \to aTb \mid bTa \mid ab \mid ba \\
&T \to XTX \mid TX \mid XT \mid XX \mid X \\
&X \to a \mid b
\end{aligned}
$$

Remove unit rules:

$$
\begin{aligned}
&S_0 \to XRX \mid RX \mid XR \mid aTb \mid bTa \mid ab \mid ba \\
&R \to XRX \mid RX \mid XR \mid aTb \mid bTa \mid ab \mid ba \\
&S \to aTb \mid bTa \mid ab \mid ba \\
&T \to XTX \mid TX \mid XT \mid XX \mid a \mid b \\
&X \to a \mid b
\end{aligned}
$$

Split long rules

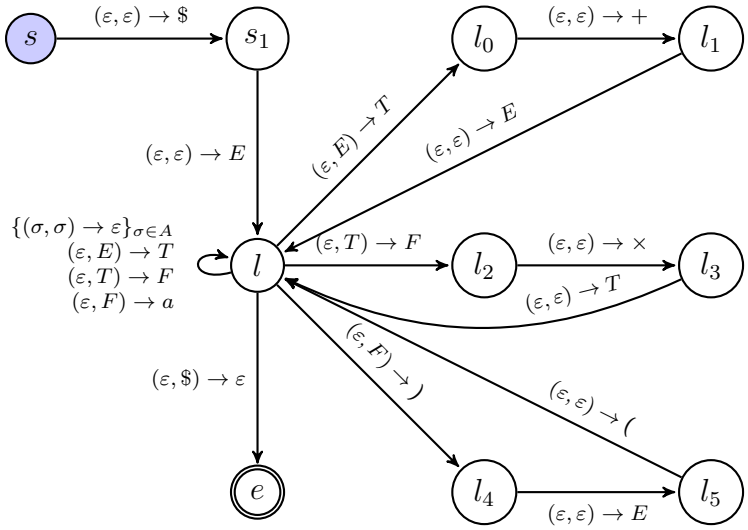
$$
\begin{aligned}
&S_0 \to XR_0 \mid RX \mid XR \mid aR_1 \mid bR_2 \mid ab \mid ba\\
&R \to XR_0 \mid RX \mid XR \mid aR_1 \mid bR_2 \mid ab \mid ba\\
&S \to aR_1 \mid bR_2 \mid ab \mid ba\\
&T \to XR_3 \mid TX \mid XT \mid XX \mid a \mid b\\
&X \to a \mid b\\
&R_0 \to RX\\
&R_1 \to Tb\\
&R_2 \to Ta\\
&R_3 \to TX
\end{aligned}
$$

Eliminate rules where terminals are not alone:

$$
\begin{aligned}
&S_0 \to XR_0 \mid RX \mid XR \mid AR_1 \mid BR_2 \mid AB \mid BA\\
&R \to XR_0 \mid RX \mid XR \mid AR_1 \mid BR_2 \mid AB \mid BA\\
&S \to AR_1 \mid BR_2 \mid AB \mid BA\\
&T \to XR_3 \mid TX \mid XT \mid XX \mid a \mid b\\
&X \to a \mid b\\
&R_0 \to RX\\
&R_1 \to TB\\
&R_2 \to TA\\
&R_3 \to TX\\
&A \to a\\
&B \to b
\end{aligned}
$$

Exercise 2.
Provide an equivalent PDA for the languages generated by the grammars G_1 and G_2 from Exercise 1

Solution 2.
A PDA that recognizes the language generated by G_1 is the following



where $A = \{+, \times, (,), a\}$. The construction follows the technique described in class. Intuitively, the stack starts with the special symbol \$ and the starting non-terminal E on top. The state l represents the core of the PDA: it removes each non terminal from the top of the stack, and replaces each non-terminal found at the top of the stack with the right hand side of a production of that non-terminal in the CFG. This is done by successive insertions of symbols in the stack (reading the right hand side of the production from left to right). Whenever the top element of the stack will be \$ the PDA will move from state l to the final state e.

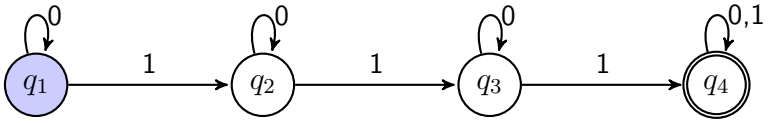
The construction of a PDA that recognizes the language generated by G_2 is left as exercise.

Exercise 3.
Construct a PDA for each of the following languages.

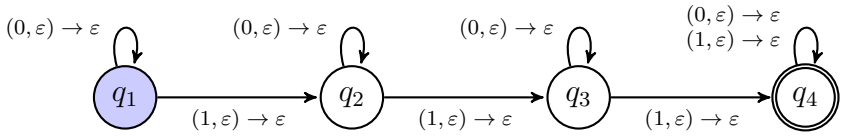
$$L_1 = \{w \in \{0,1\}^* \mid w \text{ contains at least three 1s}\}$$
$$L_2 = \{w \in \{0,1\}^* \mid w \text{ starts and ends with the same symbol}\}$$
$$L_3 = \{w \in \{0,1\}^* \mid |w| \text{ is odd and 0 is its middle symbol}\}$$

Solution 3.

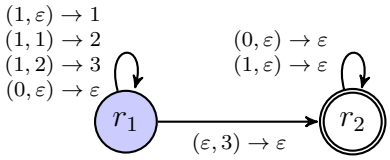
L_1) One can notice that L_1 is regular and the following DFA recognizes it



The above DFA is encoded into a PDA as follows

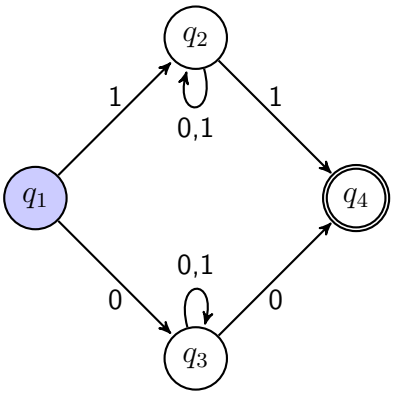


Another PDA that recognizes L_1 is the following



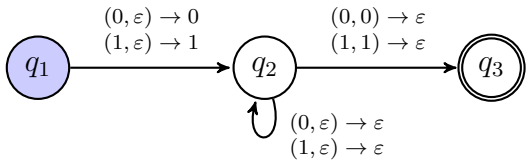
One can note that in the above PDA the stack is used to "count" the number of 1s: after consuming the first 3 ones, the PDA moves into the accepting state r_2 and stays there.

L_2) Analogously to the previous case, L_2 is regular and is recognized by the following NFA



With the technique previously described one can easily obtain a PDA that recognizes L_2.

Another PDA that recognizes L_2 is the following.

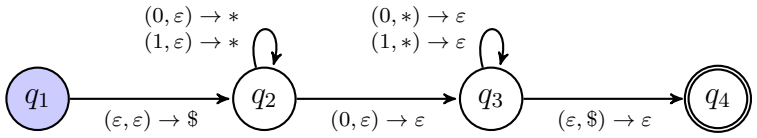


In the PDA above the stack is used to keep track of the fist symbol. The first symbol is stored in the top of the stack and used later on to check if the last symbol of the string coincides with it.

L_3) One can notice that, differently from the previous cases, L_3 is not regular. Indeed, assuming that L_3 is regular we obtain that the language

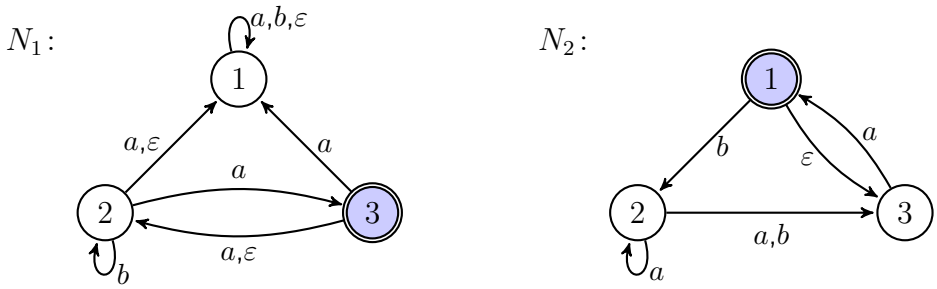
$$L_3 \cap \mathcal{L}(1^*01^*) = \{1^n01^n \mid n \in \mathbb{N}\}$$

is regular. But we have seen in a previous exercise session that it is not the case. A PDA that recognizes L_3 is the following.

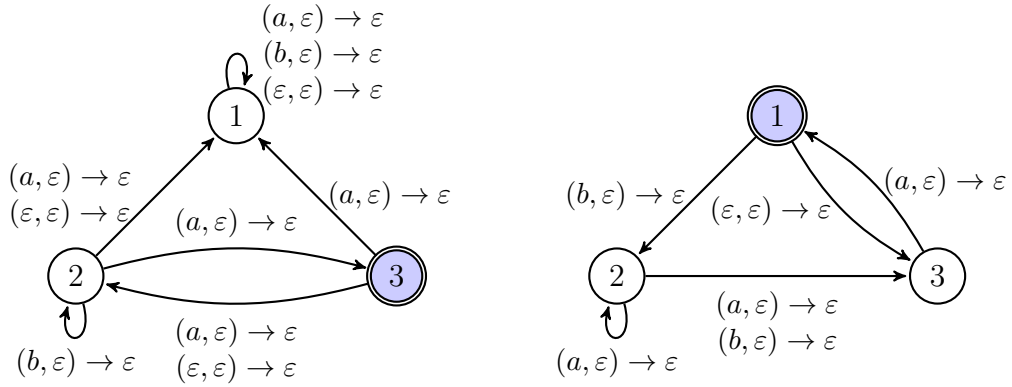


The above PDA prepares the stack by pushing the stack symbol \$ on it. Then, for each symbol that precedes the middle symbol of the string, a stack symbol $*$ is pushed on the stack. After consuming the middle symbol, the PDA removes one by one all the $*$ symbols collected in the stack until the symbol \$ is found. In this way the PDA can check that the number of symbols following the 0 in the middle is the same of those that precede it.

Exercise 4.
Construct an equivalent PDA for each of the following NFAs.



Solution 4.
Following the technique previously described the DFAs N_1 and N_2 can be respectively encoded into PDAs as follows



Exercise 5.
Give context-free grammars in Chomsky normal form for the following languages

$$L_4 = \{w \in \{a, b\}^* \mid w \text{ has more } a\text{'s than } b\text{'s}\}$$
$$L_5 = \{w\#x \in \{0, 1, \#\}^* \mid w, x \in \{0, 1\}^* \text{ and } w^R \text{ is a prefix of } x\}$$

Solution 5.

L_4) A CFG that generates L_4 is

$$S \to TaT$$
$$T \to TT \mid aTb \mid bTa \mid aT \mid Ta \mid \varepsilon$$

Notice that whenever T produces a b it will also produce an a, which guarantees that T results in at least as many a's as b's. The initial rule S ensures that there is an extra a.

The above CFG is not in Chomsky normal form. Here is an equivalent CFG in Chomsky normal form.

$$S \to TX_3 \mid TA \mid AT \mid a$$
$$T \to TT \mid AX_2 \mid BX_1 \mid AB \mid BA \mid AT \mid TA \mid a$$
$$X_1 \to TA$$
$$X_2 \to TB$$
$$X_3 \to AT$$
$$A \to a$$
$$B \to b$$

L_5) A CFG that generates L_5 is

$$S \to TX$$
$$T \to \# \mid 0T0 \mid 1T1$$
$$X \to \varepsilon \mid X\Sigma$$
$$\Sigma \to 0 \mid 1$$

The above CFG is not in Chomsky normal form. Here is an equivalent CFG in Chomsky normal form.

$$S \to TX \mid \# \mid OT_0 \mid IT_1 \qquad X \to X\Sigma \mid 0 \mid 1$$
$$T \to \# \mid OT_0 \mid IT_1 \qquad \Sigma \to 0 \mid 1$$
$$T_0 \to TO \qquad O \to 0$$
$$T_1 \to TI \qquad I \to 1$$