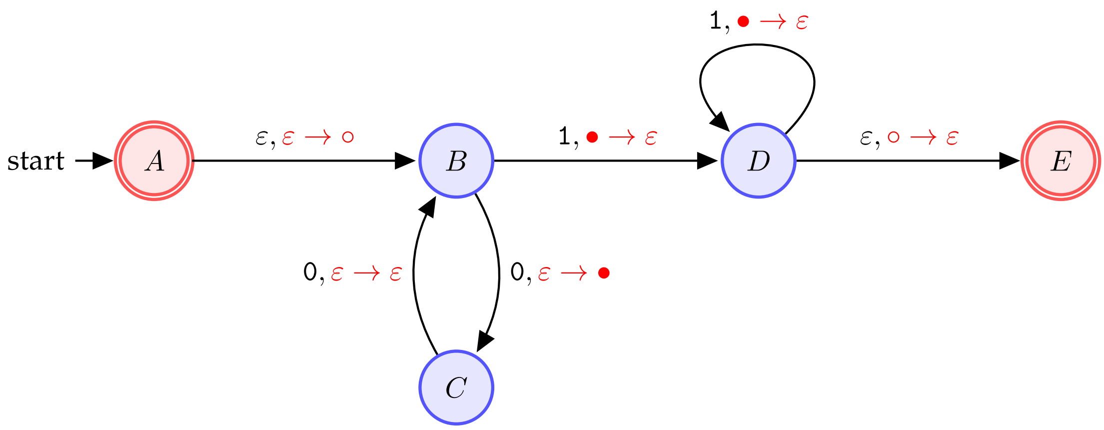
You may use JFLAP to help yourself work on these exercises.
You may wish to go through the tutorial sections: "Context-free Grammar" and "Pushdown Automata" available at https://www.jflap.org/modules/ (accessible through the left yellow navigation pane) or go through the relevant chapters in the JFLAP book https://www2.cs.duke.edu/csed/jflap/jflapbook/.

(1) Consider the following PDA




1) Simulate the following strings: (For each step record: the state, the symbol just read and the stack contents)

0001    00001    001    0011    000011

2) Use **set notation** to describe the language recognized by this PDA.

$$\{ \ 0^{\square} 1^{\square} \ \mid \ n \geq \square \ \}$$

3) Produce the formal definition for the above PDA. This should consist of:

- The set of states $Q = \{\square, \square, \square, \square, \square\}$
- The **input** alphabet $\Sigma = \{\square, \square\}$
- The **stack** alphabet $\Gamma = \{\square, \square\}$
- The start state $q_{\text{start}} = \square$
- The set of accept states $F = \{\square, \square\}$
- The transition function, $\delta \colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Q \times \Gamma_\varepsilon}$, in table form

| $\Sigma_\varepsilon \times \Gamma_\varepsilon$ : | $(0, \bullet)$ | $(0, \circ)$ | $(0, \varepsilon)$ | $(1, \bullet)$ | $(1, \circ)$ | $(1, \varepsilon)$ | $(\varepsilon, \bullet)$ | $(\varepsilon, \circ)$ | $(\varepsilon, \varepsilon)$ |
|---|---|---|---|---|---|---|---|---|---|
| $\to *A$ | | | | | | | | | $\{(B, \circ)\}$ |
| $B$ | | | $\{(C, \bullet)\}$ | $\{(D, \varepsilon)\}$ | | | | | |
| $C$ | | | $\square$ | | | | | | |
| $D$ | | | | $\{(D, \varepsilon)\}$ | | | | $\square$ | |
| $*E$ | | | | | | | | | |

The $\emptyset$ entries have been left blank to make the table easier to read.

(2) For each of the Context-Free Grammars (CFGs) given below, give answers to the accompanying questions (together with a brief justification where needed).

1) You are given the following CFG $G$ defined by the productions

$$
\begin{array}{lcll}
R & \to & XRX & \mid S \\
S & \to & \mathtt{a}T\mathtt{b} & \mid \mathtt{b}T\mathtt{a} \\
T & \to & XTX & \mid X \quad \mid \varepsilon \\
X & \to & \mathtt{a} \mid \mathtt{b} &
\end{array}
$$

This grammar generates all the strings over a and b that are not *palindromes*.

A string $w$ is a *palindrome* if $w = w^R$, where $w^R$ is formed by writing the symbols of $w$ in reverse order, e.g. if $w = 011$ then $w^R = 110$.

Answer the following questions:

1. What are the variables (non-terminals)? $V = \{\square, \square, \square, \square\}$
2. What are the terminals? $\Sigma = \{\square, \square\}$
3. What is the start variable? $\square$
4. Give three strings in $L(G)$ $\square$, $\square$, $\square$
   ($L(G)$ means: "the language of $G$")
5. Give three strings not in $L(G)$ $\square$, $\square$, $\square$
6. True or False:
   (a) $T \to \mathtt{aba}$
   (b) $T \xrightarrow{*} \mathtt{aba}$
   (c) $T \to T$
   (d) $T \xrightarrow{*} T$
   (e) $XXX \xrightarrow{*} \mathtt{aba}$
   (f) $X \xrightarrow{*} \mathtt{aba}$
   (g) $T \xrightarrow{*} XX$
   (h) $T \xrightarrow{*} XXX$
   (i) $S \xrightarrow{*} \varepsilon$

**Notation:**
$\to$: in *one* step;
$\xrightarrow{*}$: in *zero or more* steps

2)

$$
\begin{array}{lcl}
A & \to & \mathtt{bb}A\mathtt{b} \mid B \\
B & \to & \mathtt{a}B \mid \varepsilon
\end{array}
$$

Use the grammar to derive the following strings

bbab  bbb  $\mathtt{a}^6$  $\mathtt{b}^4\mathtt{a}^3\mathtt{b}^2$

3)

$$
\begin{array}{lcl}
S & \to & \mathtt{a}A\mathtt{bb} \mid \mathtt{b}B\mathtt{aa} \\
A & \to & \mathtt{a}A\mathtt{bb} \mid \varepsilon \\
B & \to & \mathtt{b}B\mathtt{aa} \mid \varepsilon
\end{array}
$$

Use the grammar to derive the following strings (where possible):

aabbbb  bbaaaa  aabb  baa

4) Let $\Sigma = \{\texttt{a}, +, \times, (, )\}$.

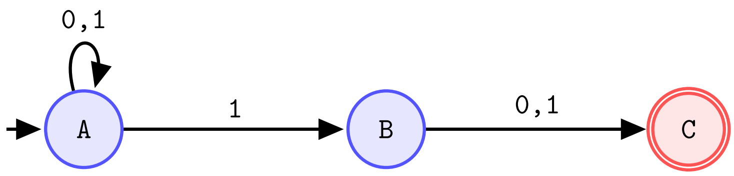
The brackets here are **symbols** in the alphabet, just like $\texttt{a}$, $+$ and $\times$.

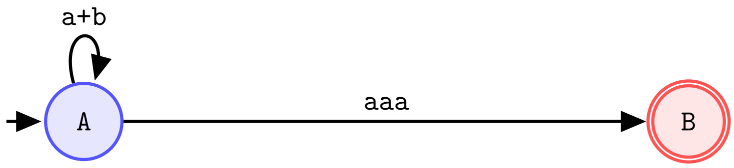
$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T \times F \mid F \\ F &\rightarrow (E) \mid \texttt{a} \end{aligned}$$

Give parse trees for each of the following strings

$$\texttt{a} \qquad \texttt{a}+\texttt{a} \qquad \texttt{a}\times\texttt{a} \qquad \texttt{a}+\texttt{a}+\texttt{a} \qquad (\texttt{a})+(\texttt{a}+\texttt{a}) \qquad ((\texttt{a}))$$

(3) Convert the following (G)NFAs into **regular grammars**.




(4) Design a PDA and a CFG for the following language over $\Sigma = \{\texttt{a}, \texttt{b}\}$

$$L = \{w \mid w = (\texttt{ab})^n \text{ or } w = \texttt{a}^{4n}\texttt{b}^{3n} \text{ for } n \geq 0\}.$$

Do this in two steps:

1) Explain the idea used, i.e. *how does the stack help you?*

2) Design a state diagram for the PDA.

3) Design a CFG.

(5) Design PDAs and CFGs for each of the following languages

1) $\{w \mid w = \texttt{b}^n\texttt{ab}^n, \quad n \geq 0\}$

2) $\{w\texttt{c}w^R \mid w \in \{\texttt{a}, \texttt{b}\}^*\}$ (so it is defined over the alphabet $\{\texttt{a}, \texttt{b}, \texttt{c}\}$)

3) $\{ww^R \mid w \in \{\texttt{a}, \texttt{b}\}^*\}$

4) The language of palindromes over $\{\texttt{a}, \texttt{b}\}$

5) The language of palindromes over $\{\texttt{a}, \texttt{b}\}$ whose length is a multiple of 3

*Hint: Consider the even and odd length cases first.*

(1) **(Ambiguity)** Sometimes a grammar can generate the same string in several different ways, with several different parse trees, and likely several different meanings. If this happens, we say that the string is derived *ambiguously* in that grammar, which is then qualified as being an **ambiguous** grammar.

Consider the CFG

$$E \to E + E \mid E \times E \mid (E) \mid a$$

Derive the string $a + a \times a$ in two different ways using parse trees, and explain their (different) meanings.

Now note that the following alternative CFG is *not* ambiguous:

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

What is the parse tree for the previous example string $(a + a \times a)$?
What is the parse tree for $(a + a) \times a$?

(2) Design CFGs generating the following languages.

1) The language of all strings over $\{\mathtt{a}, \mathtt{b}\}$ with a single symbol '$\mathtt{b}$' located *exactly in the middle* of the string.

$$\{\mathtt{b}, \mathtt{aba}, \mathtt{abb}, \mathtt{bba}, \mathtt{bbb}, \mathtt{aabaa}, \ldots\}$$

2) The language of strings over $\{\mathtt{a}, \mathtt{b}\}$ containing an equal number of $\mathtt{a}$'s and $\mathtt{b}$'s.

3) The language of strings with twice as many $\mathtt{a}$'s as $\mathtt{b}$'s.

4) $\{\mathtt{a}^i\mathtt{b}^j \mid i, j \geq 0 \text{ and } i \geq j\}$

5) $\{\mathtt{a}^i\mathtt{b}^j \mid i, j \geq 0 \text{ and } i \neq j\}$ (Complement of the language $\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$)

6) The language of strings over $\{\mathtt{a}, \mathtt{b}\}$ containing more $\mathtt{a}$'s than $\mathtt{b}$'s. (e.g. $\mathtt{abaab}$)

7) $\{w\#x \mid w, x \in \{\mathtt{0}, \mathtt{1}\}^* \text{ and } w^R \text{ is a substring of } x\}$

8) $\{x_1\#x_2\# \cdots \#x_k \mid k \geq 1, \text{ each } x_i \in \{\mathtt{a}, \mathtt{b}\}^*, \text{ and for some } i \text{ and } j, x_i = x_j^R\}$

Give informal descriptions of PDAs for the above languages. (How would you use the stack?)

(3) Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and let $B$ be the language of strings that contain at least one $\mathtt{b}$ in their second half. In other words, $B = \{uv \mid u \in \Sigma^*, v \in \Sigma^*\mathtt{b}\Sigma^* \text{ and } |v| \leq |u|\}$.

1) Give a PDA that recognizes $B$.

2) Give a CFG that generates $B$.

(4) Let

$$\begin{array}{rcl} C & = & \{x\#y \mid x, y \in \{\mathtt{0}, \mathtt{1}\}^* \text{ and } x \neq y\} \\ D & = & \{x\#y \mid x, y \in \{\mathtt{0}, \mathtt{1}\}^* \text{ and } |x| = |y| \text{ but } x \neq y\} \end{array}$$

Show that $C$ and $D$ are both CFLs by producing PDAs or CFGs for them.

(5) Give a **counter example** to show that the following construction fails to prove that the class of context-free languages (CFLs) is closed under the *star* operation.

> Let $A$ be a CFL that is generated by the CFG $G = (V, \Sigma, R, S)$.
> Add the new rule $S \to SS$ and call the resulting grammar $G'$.
> This grammar is supposed to generate $A^*$.

CFLs are actually **closed under the regular operations** (union, concatenation, and star) but this argument fails to prove closure under star. What is missing?

Intermediate

Extend your class for simulating NFAs from lab 2 to simulate PDAs.