(1) For each of the Context-Free Grammars (CFGs) given below, give answers to the accompanying questions (together with short justifications where needed)

a)

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

1. What are the variables (non-terminals)? $V = \{\boxed{R}, \boxed{S}, \boxed{T}, \boxed{X}\}$
2. What are the terminals? $\Sigma = \{\boxed{a}, \boxed{b}, \boxed{\varepsilon}\}$
3. What is the start variable? $S = \boxed{R}$
4. Give three strings in $L(G)$ $\boxed{\mathtt{ab}}$, $\boxed{\mathtt{ba}}$, $\boxed{\mathtt{aab}}$
5. Give three strings not in $L(G)$ $\boxed{\mathtt{a}}$, $\boxed{\mathtt{b}}$, $\boxed{\varepsilon}$
6. True or False:
   (a) $T \rightarrow \mathtt{aba}$
   (b) $T \xrightarrow{*} \mathtt{aba}$
   (c) $T \rightarrow T$
   (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: "$\rightarrow$" means derivable in *one* step; "$\xrightarrow{*}$" means derivable in *zero or more* steps

Solution

(a) $T \rightarrow \mathtt{aba}$: False, there is no such rule in the given set of rules.

(b) $T \xrightarrow{*} \mathtt{aba}$: True, $T \rightarrow XTX \rightarrow \mathtt{a}TX \rightarrow \mathtt{a}T\mathtt{b} \rightarrow \mathtt{a}X\mathtt{b} \rightarrow \mathtt{aba}$

(c) $T \rightarrow T$: False, there is no such rule in the given set of rules.

(d) $T \xrightarrow{*} T$: True, always possible in *zero* steps, i.e. no replacement.

(e) $XXX \xrightarrow{*} \mathtt{aba}$: True, $XXX \rightarrow \mathtt{a}XX \rightarrow \mathtt{ab}X \rightarrow \mathtt{aba}$

(f) $X \xrightarrow{*} \mathtt{aba}$: False, X can only be replaced by one terminal.

(g) $T \xrightarrow{*} XX$: True, $T \rightarrow XTX \rightarrow X\varepsilon X = XX$

(h) $T \xrightarrow{*} XXX$: True, $T \rightarrow XTX \rightarrow XXX$

(i) $S \xrightarrow{*} \varepsilon$: False, only possible route to ε is $T \rightarrow \varepsilon$, but from the starting variable there is no route to T only. ($\mathtt{a}T\mathtt{b}$ or $\mathtt{b}T\mathtt{a}$)

b)

$$\begin{aligned} A &\rightarrow \mathtt{bb}A\mathtt{b} \mid B \\ B &\rightarrow \mathtt{a}B \mid \varepsilon \end{aligned}$$

Use the grammar to derive the following strings

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

Solution

- $A \rightarrow \mathtt{bb}A\mathtt{b} \rightarrow \mathtt{bb}B\mathtt{b} \rightarrow \mathtt{bba}B\mathtt{b} \rightarrow \mathtt{bba}\varepsilon\mathtt{b} \rightarrow \mathtt{bbab}$
- $A \rightarrow \mathtt{bb}A\mathtt{b} \rightarrow \mathtt{bb}B\mathtt{b} \rightarrow \mathtt{bb}\varepsilon\mathtt{b} \rightarrow \mathtt{bbb}$
- $A \rightarrow B \rightarrow \mathtt{a}B \rightarrow \mathtt{aa}B \rightarrow \mathtt{aaa}B \rightarrow \mathtt{a}^4B \rightarrow \mathtt{a}^5B \rightarrow \mathtt{a}^6B \rightarrow \mathtt{a}^6\varepsilon = \mathtt{a}^6$
- $A \rightarrow \mathtt{bb}A\mathtt{b} \rightarrow \mathtt{bbbb}A\mathtt{bb} \rightarrow \mathtt{b}^4B\mathtt{b}^2 \rightarrow \mathtt{b}^4\mathtt{a}B\mathtt{b}^2 \rightarrow \mathtt{b}^4\mathtt{a}^2B\mathtt{b}^2 \rightarrow \mathtt{b}^4\mathtt{a}^3B\mathtt{b}^2 \rightarrow \mathtt{b}^4\mathtt{a}\varepsilon\mathtt{b}^2 = \mathtt{b}^4\mathtt{a}^3\mathtt{b}^2$

c)

$$\begin{aligned} S &\rightarrow \mathtt{a}A\mathtt{bb} \mid \mathtt{b}B\mathtt{aa} \\ A &\rightarrow \mathtt{a}A\mathtt{bb} \mid \varepsilon \\ B &\rightarrow \mathtt{b}B\mathtt{aa} \mid \varepsilon \end{aligned}$$

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

$\mathtt{aabbbb} \qquad \mathtt{bbaaaa} \qquad \mathtt{aabb} \qquad \mathtt{baa}$

Solution

- $S \rightarrow \mathtt{a}A\mathtt{bb} \rightarrow \mathtt{aa}A\mathtt{bbbb} \rightarrow \mathtt{aa}\varepsilon\mathtt{bbbb} = \mathtt{aabbbb}$
- $S \rightarrow \mathtt{b}B\mathtt{aa} \rightarrow \mathtt{bb}B\mathtt{aaaa} \rightarrow \mathtt{bb}\varepsilon\mathtt{aaaa} = \mathtt{bbaaaa}$
- aabb, not possible.
- $S \rightarrow \mathtt{b}B\mathtt{aa} \rightarrow \mathtt{b}\varepsilon\mathtt{aa} = \mathtt{baa}$

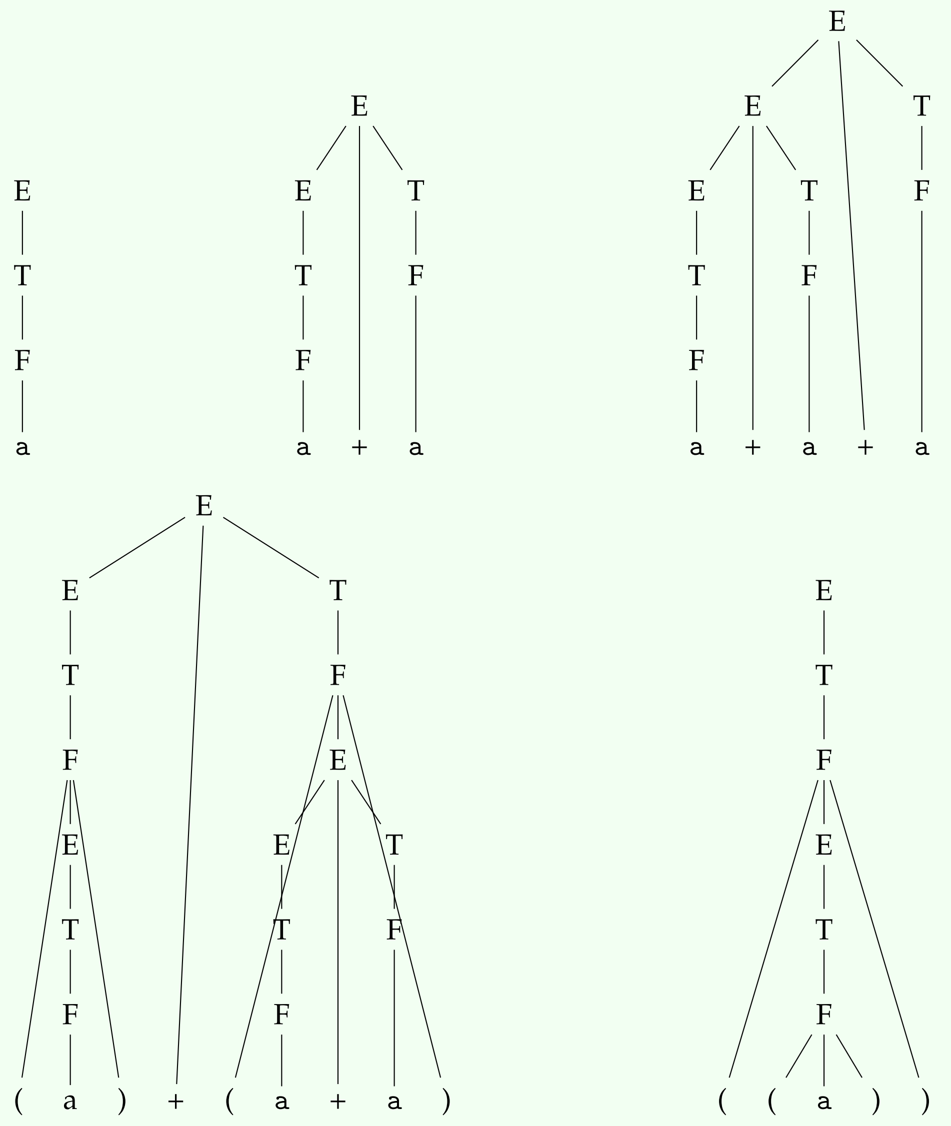
d)

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

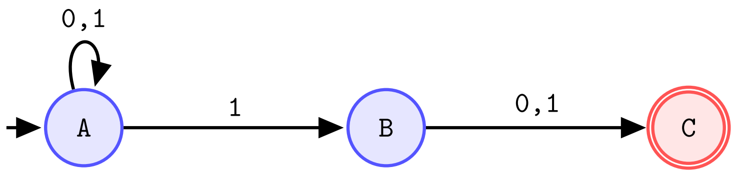
Give parse trees for each of the following strings

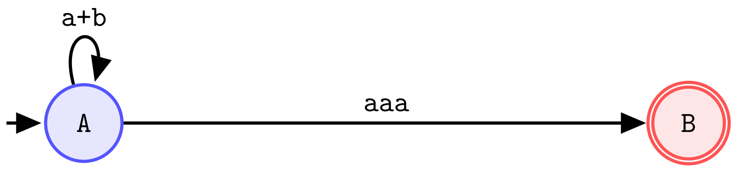
$$\texttt{a} \qquad \texttt{a}+\texttt{a} \qquad \texttt{a}+\texttt{a}+\texttt{a} \qquad (\texttt{a})+(\texttt{a}+\texttt{a}) \qquad ((\texttt{a}))$$

Solution



(2) Convert the following (G)NFAs into Regular Grammars.





Solution

First:

$$\begin{array}{rcl} A & \to & \mathtt{0}A \mid \mathtt{1}A \mid \mathtt{1}B \\ B & \to & \mathtt{0}C \mid \mathtt{1}C \\ C & \to & \varepsilon \end{array}$$

Second:

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

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

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

Do this in two steps:

a) Explain the idea used.

b) Design a state diagram for the PDA.

c) Design a CFG.

Solution

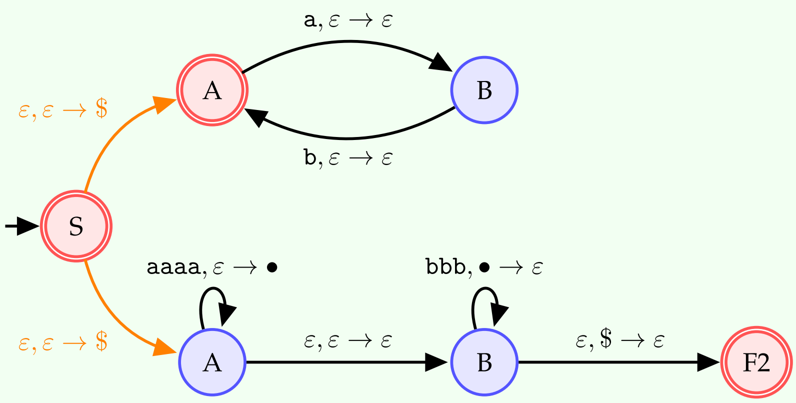
a) Idea: union of two languages

$$L = \{(\mathtt{ab})^n \mid n \geq 0\} \cup \{(\mathtt{aaaa})^n(\mathtt{bbb})^n \mid n \geq 0\}.$$

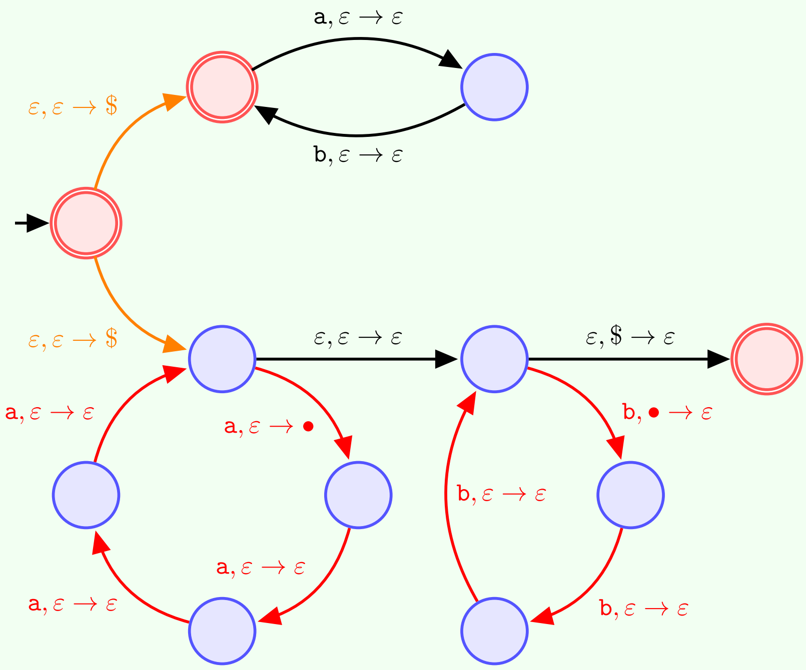
Here $\{(\mathtt{ab})^n \mid n \geq 0\} = (ab)^*$ is regular – no need to use the stack for it.

For $\{(\mathtt{aaaa})^n(\mathtt{bbb})^n \mid n \geq 0\}$: count the occurrences of the string aaaa then match it with the number of occurrences of bbb.

b) Abbreviated PDA:



Expanded PDA:



c) CFG

$$
\begin{array}{lcl}
S & \to & A \mid B \\
A & \to & \texttt{ab}A \mid \varepsilon \\
B & \to & \texttt{aaaa}B\texttt{bbb} \mid \varepsilon
\end{array}
$$

(4) **(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

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

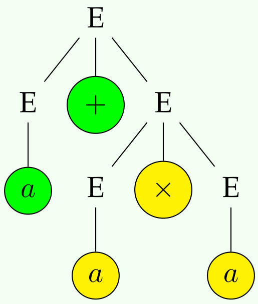
Derive the string $a + a \times a$ in two different ways using parse trees, and explain their (different) meanings.

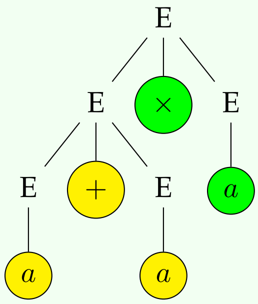
Now note that the following alternative CFG is *not* ambiguous:

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

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

Solution

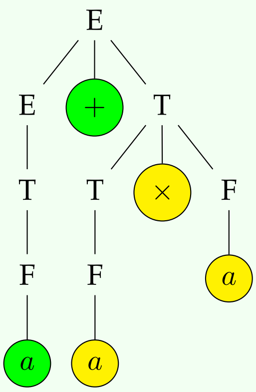




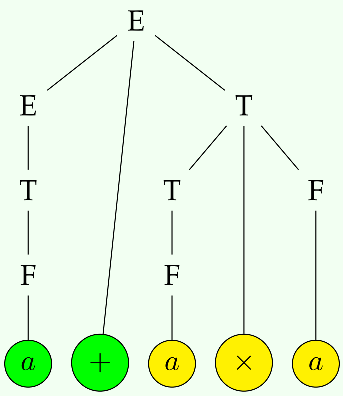
The first one: $a + (a \times a)$.

The second one: $(a + a) \times a$.

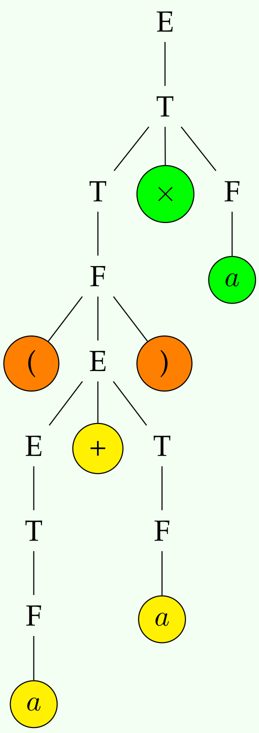
Using the second grammar to parse $a + a \times a$ gives



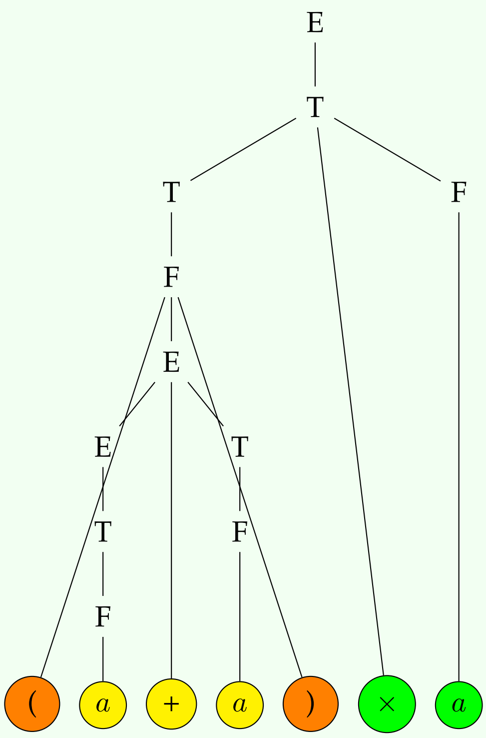
i.e.



and for $(a + a) \times a$ we get



i.e.



(5) 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 $w = \texttt{011}$ then $w^R = \texttt{110}$.

Design PDAs and CFGs for each of the following languages

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

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

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

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

e) 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.

Solution

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

$$S \to \texttt{b}S\texttt{b} \mid \texttt{a}$$

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

$$S \to \texttt{a}S\texttt{a} \mid \texttt{b}S\texttt{b} \mid \texttt{c}$$

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

$$S \to \texttt{a}S\texttt{a} \mid \texttt{b}S\texttt{b} \mid \varepsilon$$

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

$$S \to \texttt{a}S\texttt{a} \mid \texttt{b}S\texttt{b} \mid \texttt{a} \mid \texttt{b} \mid \varepsilon$$

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

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

(6) Design CFGs generating the following languages.

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

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

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

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

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

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

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

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

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

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

Solution

a) $\{\texttt{a}^i\texttt{b}^j \mid i, j \geq 0 \text{ and } i \geq j\}$: $\texttt{a}^*\texttt{a}^n\texttt{b}^n$

$$\begin{array}{lcll} S & \to & AB & \\ A & \to & \texttt{a}A & \mid \varepsilon \\ B & \to & \texttt{a}B\texttt{b} & \mid \varepsilon \end{array}$$

PDA: fill stack one token for each a, then remove one token for each b. If stack is not empty at the end then accept.

b) $\{\texttt{a}^i\texttt{b}^j \mid i, j \geq 0 \text{ and } i \neq j\} = \{\texttt{a}^i\texttt{b}^j \mid i > j\} \cup \{\texttt{a}^i\texttt{b}^j \mid i < j\}$:
$\texttt{a}^+\texttt{a}^n\texttt{b}^n$ or $\texttt{a}^n\texttt{b}^n\texttt{b}^+$

$$\begin{array}{lcll} S & \to & AC \mid CB & \\ A & \to & \texttt{a}A & \mid \texttt{a} \\ B & \to & B\texttt{b} & \mid \texttt{b} \\ C & \to & \texttt{a}B\texttt{b} & \mid \varepsilon \end{array}$$

or

$$\begin{array}{lcl} S & \to & X\texttt{b}X\texttt{a}B \mid T \mid U \\ T & \to & \texttt{a}T\texttt{b} \mid T\texttt{b} \mid \texttt{b} \\ U & \to & \texttt{a}U\texttt{b} \mid \texttt{a}U \mid \texttt{a} \\ X & \to & \texttt{a} \mid \texttt{b} \end{array}$$

PDA: non-deterministaccly create two branches at the beginnig: branch #1 for the $i > j$ case where we use the idea from the previous case; branch #2 for the $i < j$ case where we fill stack one token for each a, then remove one token for each b. If stack is empty before the end of the string then accept.

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

$$\begin{array}{lcll} S & \to & ASA & \mid \texttt{b} \\ A & \to & \texttt{a} & \mid \texttt{b} \end{array}$$

PDA: need to guess the middle of the string, so need non-deterministic transition after each symbol. State before the non-deterministic transition counts the number of prior symbols, and the one after it checks if the number of the remaining symbols matches.

d) The language of strings over $\{\texttt{a}, \texttt{b}\}$ containing more a's than b's.

$$\begin{array}{lcl} S & \to & A\texttt{a}A \\ A & \to & AA \mid \texttt{a}A\texttt{b} \mid \texttt{b}A\texttt{a} \mid \texttt{a}A \mid \varepsilon \end{array}$$

or

$$\begin{array}{rcl} S & \to & AS \mid \texttt{a}A \mid \texttt{a}S \\ A & \to & AA \mid \texttt{a}A\texttt{b} \mid \texttt{b}A\texttt{a} \mid \varepsilon \end{array}$$

This is because if a string w contains more a's that b's, then it must be of one of the following forms:

- "ax" such that x contains more a's than b's.
- "ax" such that x contains equal number of a's and b's.
- "xy" such that x contains equal number of a's and b's, and y contains more b's than a's.

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

$$S \to SS \mid \texttt{a}S\texttt{b} \mid \texttt{b}S\texttt{a} \mid \varepsilon$$

PDA: string made of sub-strings/chunks that have equal symbols: if chunk starts with a then fill stack for a's and empty for b's, and vice versa.

f) The language of strings with twice as many a's as b's.

$$S \to S\texttt{a}S\texttt{a}S\texttt{b}S \mid S\texttt{a}S\texttt{b}S\texttt{a}S \mid S\texttt{b}S\texttt{a}S\texttt{a}S \mid \varepsilon$$

PDA: string made of sub-strings/chunks that have the required property: if chunk starts with a then fill stack for a's and empty twice for b's, and vice versa.

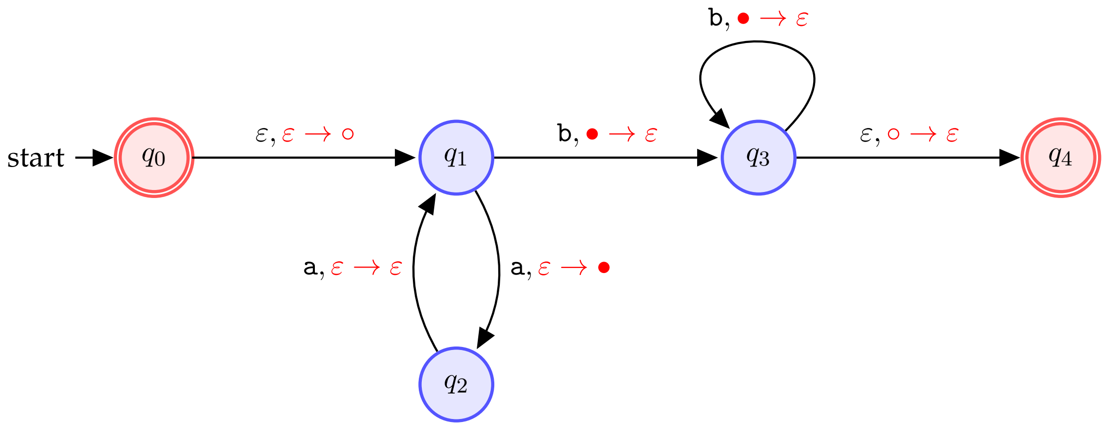
g) $\{w\#x \mid w, x \in \{\texttt{0}, \texttt{1}\}^*$ and w^R is a substring of $x\}$

$$\begin{array}{rcl} S & \to & TX \\ T & \to & 0T0 \mid 1T1 \mid \#X \\ X & \to & 0X \mid 1X \mid \varepsilon \end{array}$$

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

$$\begin{array}{rcl} S & \to & UPV \\ P & \to & \texttt{a}P\texttt{a} \mid \texttt{b}P\texttt{b} \mid T \mid \varepsilon \\ T & \to & \#MT \mid \# \\ U & \to & M\#U \mid \varepsilon \\ V & \to & \#MV \mid \varepsilon \\ M & \to & \texttt{a}M \mid \texttt{b}M \mid \varepsilon \end{array}$$

(1) Consider the following PDA



a) Produce the formal definition for the above NFA. This should consist of:

- The set of states $Q = \{\boxed{q_0}, \boxed{q_1}, \boxed{q_2}, \boxed{q_3}, \boxed{q_4}\}$
- The input alphabet $\Sigma = \{\boxed{\mathtt{a}}, \boxed{\mathtt{b}}\}$
- The stack alphabet $\Gamma = \{\boxed{\bullet}, \boxed{\circ}\}$
- The transition function, $\delta \colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Q \times \Gamma_\varepsilon}$, in table form

Σ_ε :	a			b			ε		
Γ_ε :	$\bullet$	$\circ$	ε	$\bullet$	$\circ$	ε	$\bullet$	$\circ$	ε
q_0									$\{(q_1, \circ)\}$
q_1			$\{(q_2, \bullet)\}$	$\{(q_3, \varepsilon)\}$					
q_2			$\boxed{\{(q_1, \varepsilon)\}}$						
q_3				$\{(q_3, \varepsilon)\}$				$\boxed{\{(q_4, \varepsilon)\}}$	
q_4									

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

- The start state $q_{\text{start}} = \boxed{q_0}$
- The set of accept states $F = \{\boxed{q_0}, \boxed{q_4}\}$

b) Simulate the following strings: (For each step record: the state, the symbol just read and the stack contents – you may use JFLAP to help you)

`aaab`   `aaaab`   `aab`   `aabb`   `aaaabb`

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

(Notations similar to $\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ for example).

(2) 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, $B = \{uv \mid u \in \Sigma^*, v \in \Sigma^*\mathtt{b}\Sigma^* \text{ and } |v| \leq |u|\}$.

a) Give a PDA that recognizes B.

b) Give a CFG that generates B.

(3) Let

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

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

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

> 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^*.

*Note: the class of **context-free languages** is actually **closed under the regular operations** (union, concatenation, and star) but the above argument fails to prove closure under star. What is missing?*