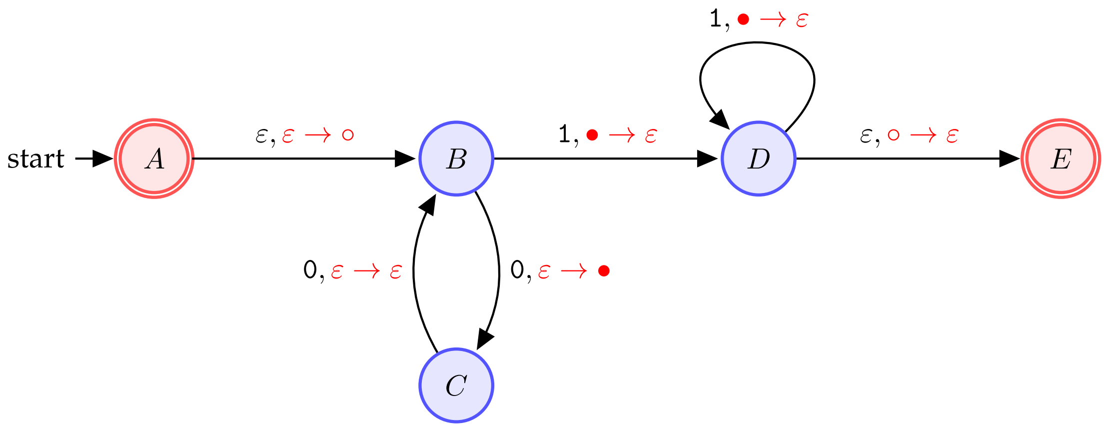
(1) Consider the following PDA



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

`0001` `00001` `001` `0011` `000011`

Solution

	States	Stack
ε	$\{B\}$	$\circ$
0	$\{C\}$	$\circ\bullet$
0	$\{B\}$	$\circ\bullet$
0	$\{C\}$	$\circ\bullet\bullet$
1	$\emptyset$	$\circ\bullet\bullet$

rejected

	States	Stack
ε	$\{B\}$	$\circ$
0	$\{C\}$	$\circ\bullet$
0	$\{B\}$	$\circ\bullet$
0	$\{C\}$	$\circ\bullet\bullet$
0	$\{B\}$	$\circ\bullet\bullet$
1	$\{D\}$	$\circ\bullet$

rejected

	States	Stack	States	Stack
ε	$\{B\}$	$\circ$		
0	$\{C\}$	$\circ\bullet$		
0	$\{B\}$	$\circ\bullet$		
1	$\{D\}$	$\circ$	$\{E\}$	(empty)

accepted

	States	Stack	States	Stack
ε	$\{B\}$	$\circ$		
0	$\{C\}$	$\circ\bullet$		
0	$\{B\}$	$\circ\bullet$		
1	$\{D\}$	$\circ$	$\{E\}$	
1	$\emptyset$	$\circ$	$\emptyset$	

rejected

	States	Stack	States	Stack
ε	$\{B\}$	$\circ$		
0	$\{C\}$	$\circ\bullet$		
0	$\{B\}$	$\circ\bullet$		
0	$\{C\}$	$\circ\bullet\bullet$		
0	$\{B\}$	$\circ\bullet\bullet$		
1	$\{D\}$	$\circ\bullet$		
1	$\{D\}$	$\circ$	$\{E\}$	(empty)

accepted

NB. I have simplified what happens at the start of the string: technically it starts in $\{A, B\}$, and we have 2 stacks, but the one at A disappears as soon as we read an actual symbol from the input string.

I have shown the non-determinism at $\{D\}$.

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

$$\{ \ 0^{\boxed{2n}} 1^{\boxed{n}} \ \mid \ n \geq \boxed{0} \ \}$$

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

- The set of states $Q = \{\boxed{A}, \boxed{B}, \boxed{C}, \boxed{D}, \boxed{E}\}$
- The **input** alphabet $\Sigma = \{\boxed{\mathtt{0}}, \boxed{\mathtt{1}}\}$
- The **stack** alphabet $\Gamma = \{\boxed{\circ}, \boxed{\bullet}\}$
- The start state $q_{\text{start}} = \boxed{A}$
- The set of accept states $F = \{\boxed{A}, \boxed{E}\}$
- The transition function, $\delta\colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Q \times \Gamma_\varepsilon}$, in table form

$\Sigma_\varepsilon \times \Gamma_\varepsilon$:	$(\mathtt{0}, \bullet)$	$(\mathtt{0}, \circ)$	$(\mathtt{0}, \varepsilon)$	$(\mathtt{1}, \bullet)$	$(\mathtt{1}, \circ)$	$(\mathtt{1}, \varepsilon)$	$(\varepsilon, \bullet)$	$(\varepsilon, \circ)$	$(\varepsilon, \varepsilon)$
$\rightarrow *A$									$\{(B, \circ)\}$
B			$\{(C, \bullet)\}$	$\{(D, \varepsilon)\}$					
C			$\boxed{\{(B, \varepsilon)\}}$						
D				$\{(D, \varepsilon)\}$				$\boxed{\{(E, \varepsilon)\}}$	
$*E$									

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

Basic

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

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

Solution

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

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

(c) $T \to T$: 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) $XXX \xrightarrow{*} \texttt{aba}$: True, $XXX \to \texttt{a}XX \to \texttt{ab}X \to \texttt{aba}$

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

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

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

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

2)

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

Use the grammar to derive the following strings

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

Solution

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

3)

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

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

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

Solution

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

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

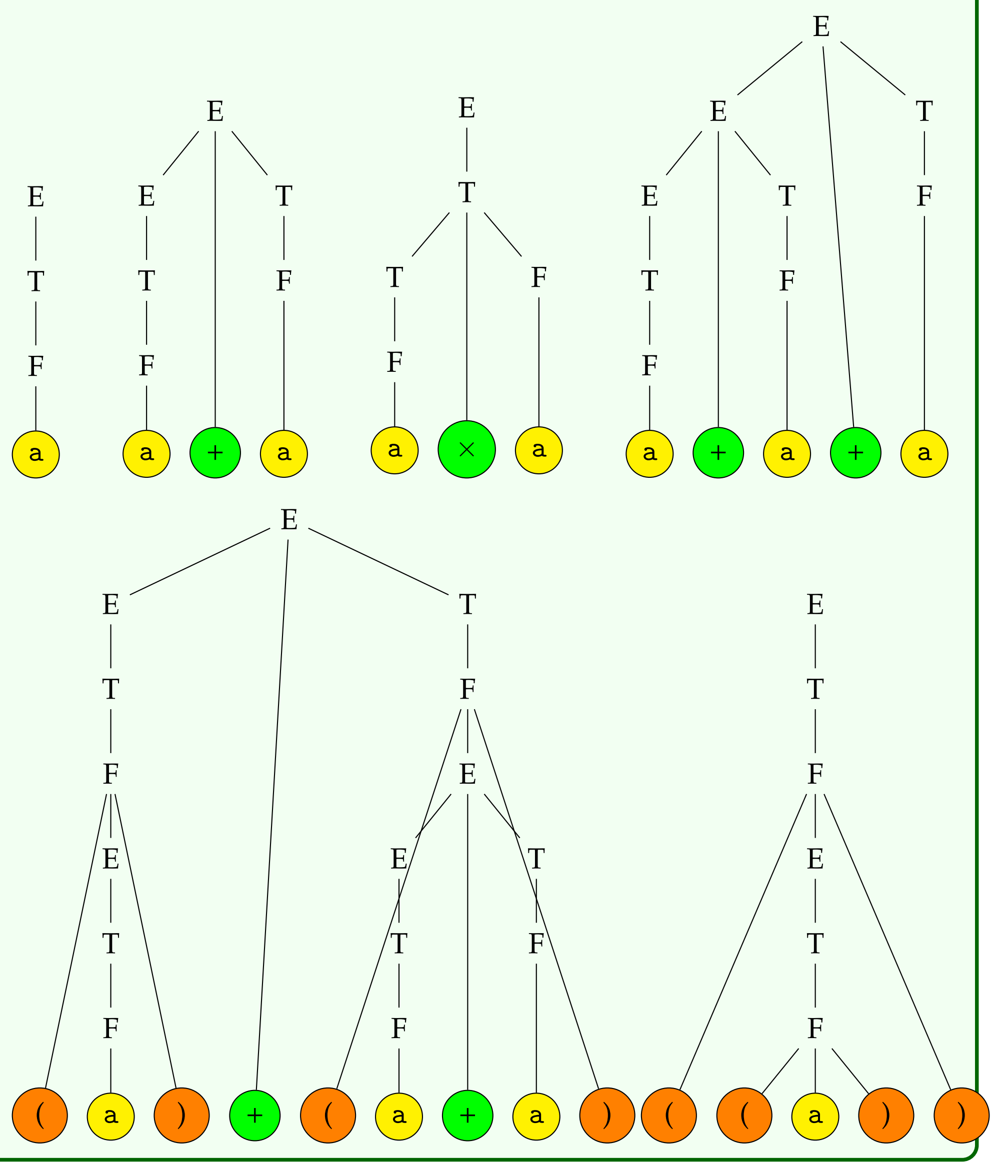
The brackets here are **symbols** in the alphabet, just like $\mathtt{a}$, $+$ and $\times$.

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

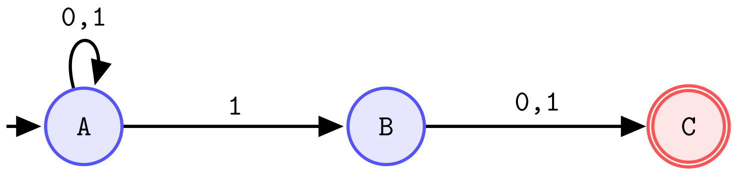
Give parse trees for each of the following strings

$\mathtt{a}$ $\quad$ $\mathtt{a}+\mathtt{a}$ $\quad$ $\mathtt{a}\times\mathtt{a}$ $\quad$ $\mathtt{a}+\mathtt{a}+\mathtt{a}$ $\quad$ $(\mathtt{a})+(\mathtt{a}+\mathtt{a})$ $\quad$ $((\mathtt{a}))$

Solution

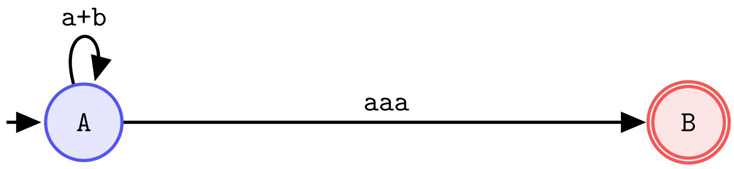


(3) Convert the following (G)NFAs into **regular grammars**.



Solution

$$\begin{array}{lcl} A & \rightarrow & 0A \mid 1A \mid 1B \\ B & \rightarrow & 0C \mid 1C \\ C & \rightarrow & \varepsilon \end{array}$$



Solution

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

(4) 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:

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.

Solution

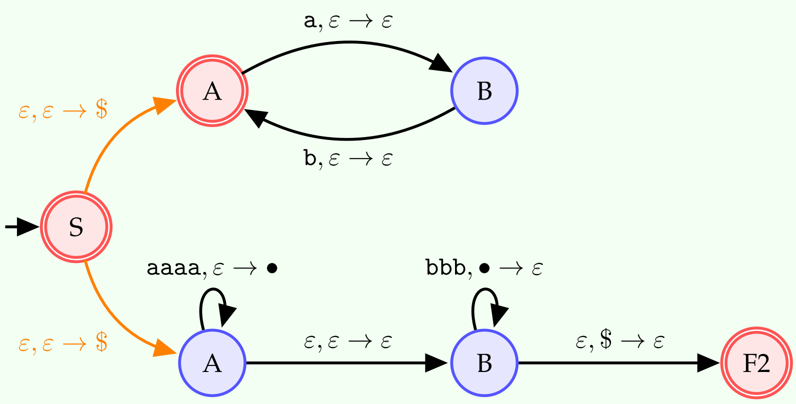
1) Idea: union of two languages

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

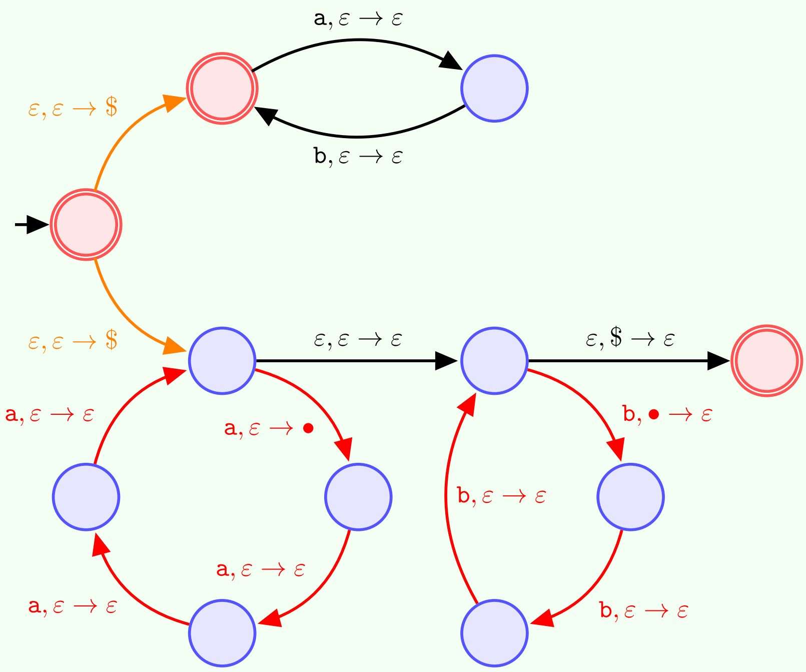
Here $\{(\mathtt{ab})^n \mid n \geq 0\} = (ab)^*$ is regular – no need to use the stack for it.

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.

2) Abbreviated PDA:



Expanded PDA:



3) CFG

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

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

1) $\{w \mid w = \mathtt{b}^n\mathtt{ab}^n, \quad n \geq 0\}$
2) $\{w\mathtt{c}w^R \mid w \in \{\mathtt{a},\mathtt{b}\}^*\}$ (so it is defined over the alphabet $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$)
3) $\{ww^R \mid w \in \{\mathtt{a},\mathtt{b}\}^*\}$
4) The language of palindromes over $\{\mathtt{a},\mathtt{b}\}$
5) The language of palindromes over $\{\mathtt{a},\mathtt{b}\}$ whose length is a multiple of 3

Hint: Consider the even and odd length cases first.

Solution

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

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

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

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

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

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

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

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

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

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

(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

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

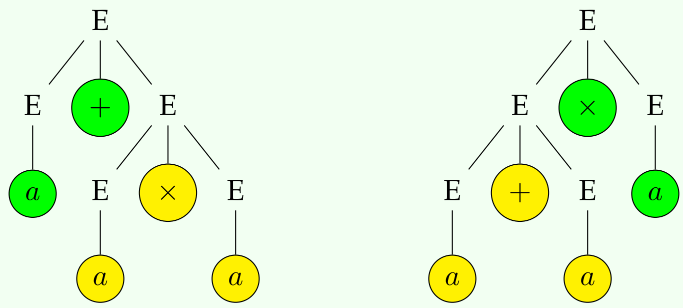
Derive the string $a + a \times a$ in two different ways using parse trees, and explain their (different) meanings.

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

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

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

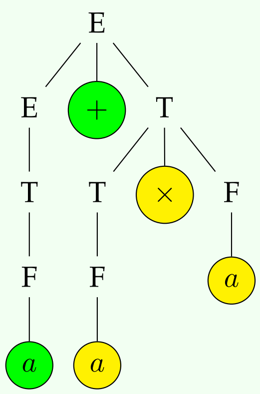
Solution



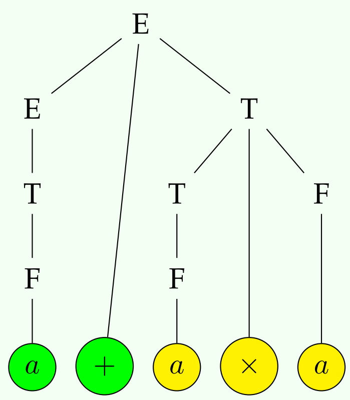
The first one: $a + (a \times a)$.

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

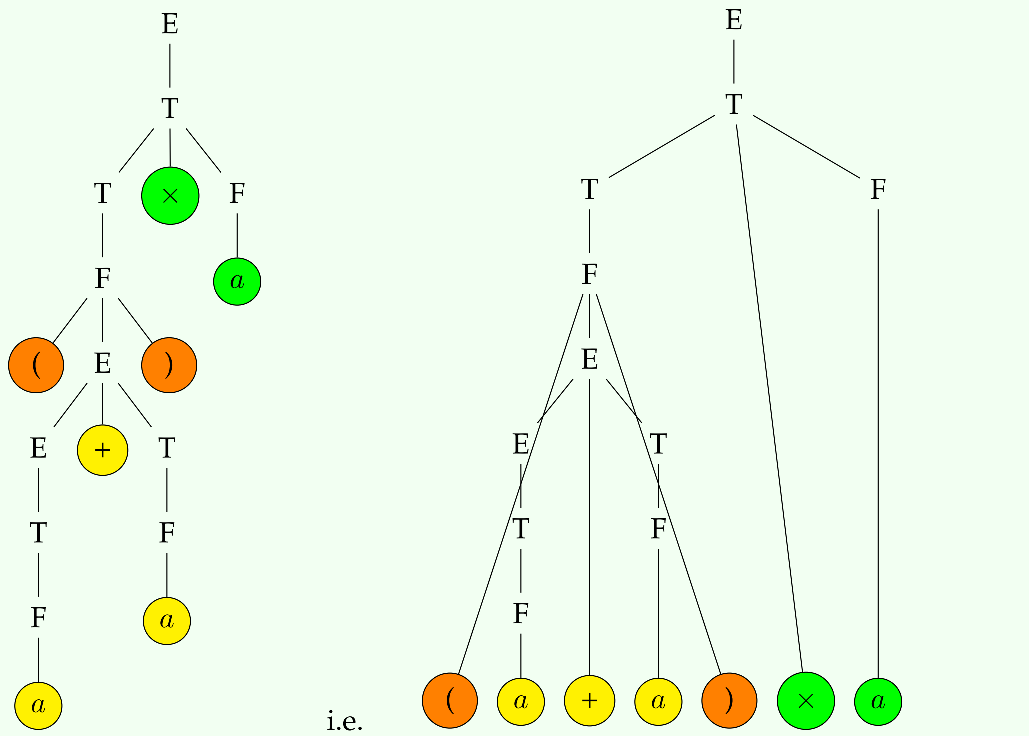
Using the second grammar to parse $a + a \times a$ gives



i.e.



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



(2) Design CFGs generating the following languages.

1) The language of all strings over $\{\mathtt{a}, \mathtt{b}\}$ with a single symbol '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 a's and b's.

3) The language of strings with twice as many a's as 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 a's than b's. (e.g. 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?)

Solution

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

$$\begin{array}{rcll} S & \to & ASA & \mid \mathtt{b} \\ A & \to & \mathtt{a} & \mid \mathtt{b} \end{array}$$

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.

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

b's.

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

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.

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

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

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.

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

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

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.

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

Using variable C for $\mathtt{a}^n\mathtt{b}^n$, A for $\mathtt{a}^+$, and B for $\mathtt{b}^+$, we get:

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

or

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

PDA: non-deterministically create two branches at the beginning: 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.

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

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

or

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

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.

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

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

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

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

(3) Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and let B be the language of strings that contain at least one 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.

Solution

PDA: We need to guess where to break the input string into uv, so we will need non-determinism. We need to compute the length of u, then ensure that v is at most as long as v and that it contains a b.

(4) Let

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

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?

Solution

$S \to SS$ produces multiple copies but does not produce the empty string. It needs to be added (if it is not present in the given language), so we need: $S \to SS \mid \varepsilon$.

Advanced

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