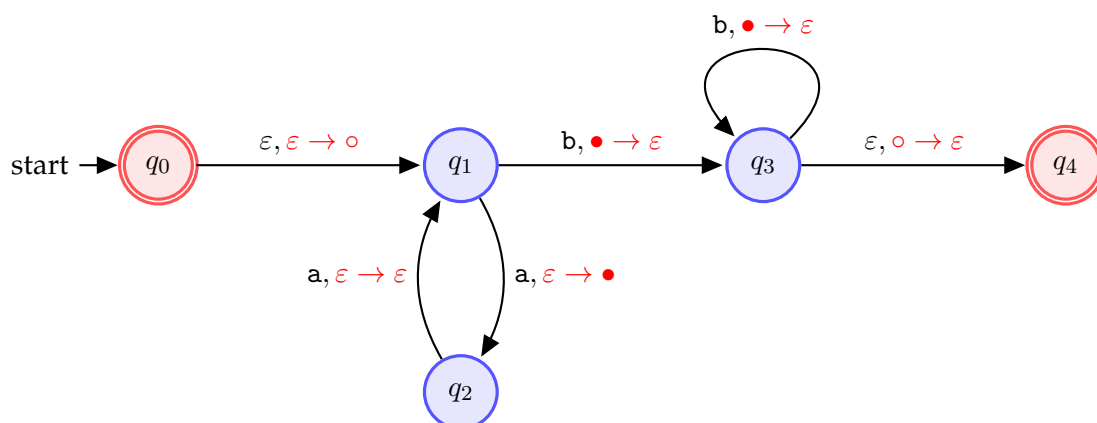


(1) Consider the following PDA



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

- The set of states $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- The input alphabet $\Sigma = \{a, b\}$
- The stack alphabet $\Gamma = \{\bullet, o\}$
- The transition function, $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow 2^{Q \times \Gamma_\epsilon}$, in table form

$\Sigma_\epsilon :$	a			b			ϵ		
$\Gamma_\epsilon :$	•	o	ϵ	•	o	ϵ	•	o	ϵ
q_0									$\{(q_1, o)\}$
q_1			$\{(q_2, \bullet)\}$	$\{(q_3, \epsilon)\}$					
q_2			$\{(q_1, \epsilon)\}$						
q_3				$\{(q_3, \epsilon)\}$				$\{(q_4, \epsilon)\}$	
q_4									

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

- The start state $q_{\text{start}} = q_0$
 - The set of accept states $F = \{q_0, q_4\}$
- 2) Simulate the following strings: (For each step record: the state, the symbol just read and the stack contents)

aaab aaaab aab aabb aaaabb

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

$$\{ a^{2n} b^n \mid n \geq 0 \}$$

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

1)

$$A \rightarrow \text{bbAb} \mid B$$

$$B \rightarrow \text{aB} \mid \varepsilon$$

Use the grammar to derive the following strings

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

Solution

- $A \rightarrow \text{bbAb} \rightarrow \text{bbBb} \rightarrow \text{bbaBb} \rightarrow \text{bba}\varepsilon\text{b} \rightarrow \text{bbab}$
- $A \rightarrow \text{bbAb} \rightarrow \text{bbBb} \rightarrow \text{bb}\varepsilon\text{b} \rightarrow \text{bbb}$
- $A \rightarrow B \rightarrow \text{aB} \rightarrow \text{aaB} \rightarrow \text{aaaB} \rightarrow \text{a}^4\text{B} \rightarrow \text{a}^5\text{B} \rightarrow \text{a}^6\text{B} \rightarrow \text{a}^6\varepsilon = \text{a}^6$
- $A \rightarrow \text{bbAb} \rightarrow \text{bbbbbAbb} \rightarrow \text{b}^4\text{Bb}^2 \rightarrow \text{b}^4\text{aBb}^2 \rightarrow \text{b}^4\text{a}^2\text{Bb}^2 \rightarrow \text{b}^4\text{a}^3\text{Bb}^2 \rightarrow \text{b}^4\text{a}^3\text{b}^2 = \text{b}^4\text{a}^3\text{b}^2$

2)

$$S \rightarrow \text{aAbb} \mid \text{bBaa}$$

$$A \rightarrow \text{aAbb} \mid \varepsilon$$

$$B \rightarrow \text{bBaa} \mid \varepsilon$$

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

aabbbb bbaaaa aabb baa

Solution

- $S \rightarrow \text{aAbb} \rightarrow \text{aaAbbbb} \rightarrow \text{aa}\varepsilon\text{bbbb} = \text{aabbbb}$
- $S \rightarrow \text{bBaa} \rightarrow \text{bbBaaaa} \rightarrow \text{bb}\varepsilon\text{aaaa} = \text{bbaaaa}$
- aabb, not possible.
- $S \rightarrow \text{bBaa} \rightarrow \text{b}\varepsilon\text{aa} = \text{baa}$

3) Here $\Sigma = \{a, +, \times, (,)\}$.

$$E \rightarrow E + T \mid T$$

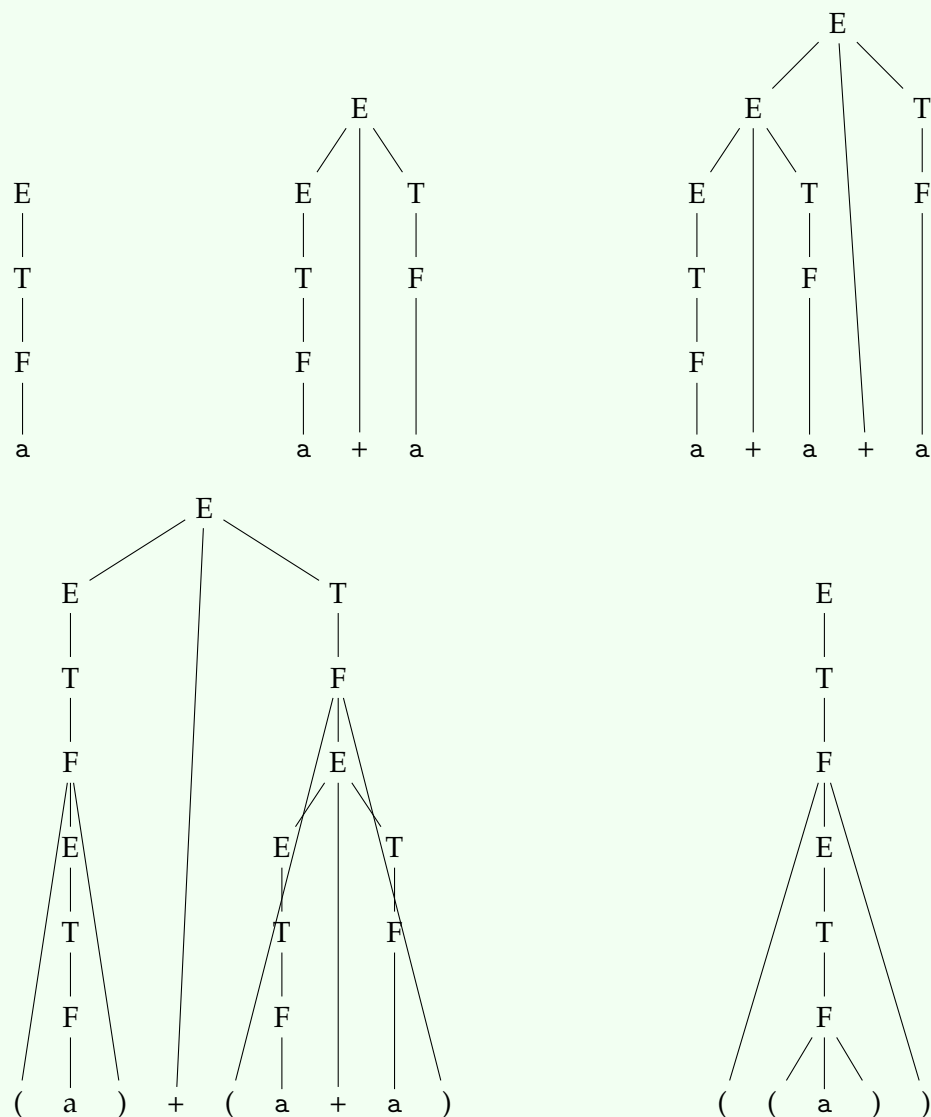
$$T \rightarrow T \times F \mid F$$

$$F \rightarrow (E) \mid a$$

Give parse trees for each of the following strings

a $a + a$ $a + a + a$ $(a) + (a + a)$ $((a))$

Solution



4) You are given the following CFG productions

$$R \rightarrow XRX \mid S$$

$$S \rightarrow aTb \mid bTa$$

$$T \rightarrow XTX \mid X \mid \varepsilon$$

$$X \rightarrow a \mid b$$

1. What are the variables (non-terminals)? $V = \{R, S, T, X\}$

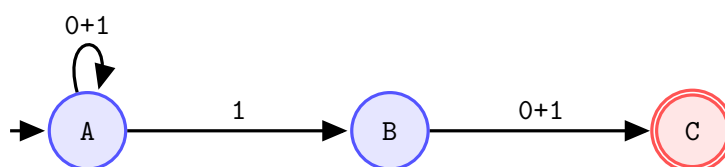
2. What are the terminals? $\Sigma = \{a, b\}$

3. What is the start variable? R
4. Give three strings in $L(G)$ ab , ba , aab
5. Give three strings not in $L(G)$ a , b , ε
6. True or False:
 - (a) $T \rightarrow aba$
 - (b) $T \xrightarrow{*} aba$
 - (c) $T \rightarrow T$
 - (d) $T \xrightarrow{*} T$
 - (e) $XXX \xrightarrow{*} aba$
 - (f) $X \xrightarrow{*} aba$
 - (g) $T \xrightarrow{*} XX$
 - (h) $T \xrightarrow{*} XXX$
 - (i) $S \xrightarrow{*} \varepsilon$

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

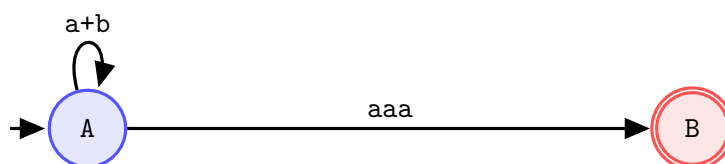
- (a) $T \rightarrow aba$: False, there is no such rule in the given set of rules.
- (b) $T \xrightarrow{*} aba$: True, $T \rightarrow XTX \rightarrow aTX \rightarrow aTb \rightarrow aXb \rightarrow aba$
- (c) $T \rightarrow T$: False, there is no such rule in the given set of rules.
- (d) $T \xrightarrow{*} T$: True, always possible in *zero* steps, i.e. no replacement.
- (e) $XXX \xrightarrow{*} aba$: True, $XXX \rightarrow aXX \rightarrow abX \rightarrow aba$
- (f) $X \xrightarrow{*} aba$: 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. (aTb or bTa)

- (3) Convert the following GNFA's into **regular grammars**.



Solution

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



Solution

$$\begin{aligned} A &\rightarrow aA \mid bA \mid aaaB \\ B &\rightarrow \varepsilon \end{aligned}$$

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

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

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

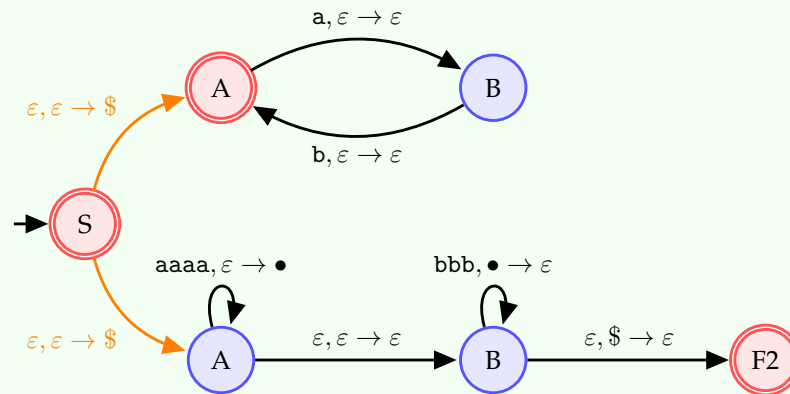
- 1) Idea: union of two languages

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

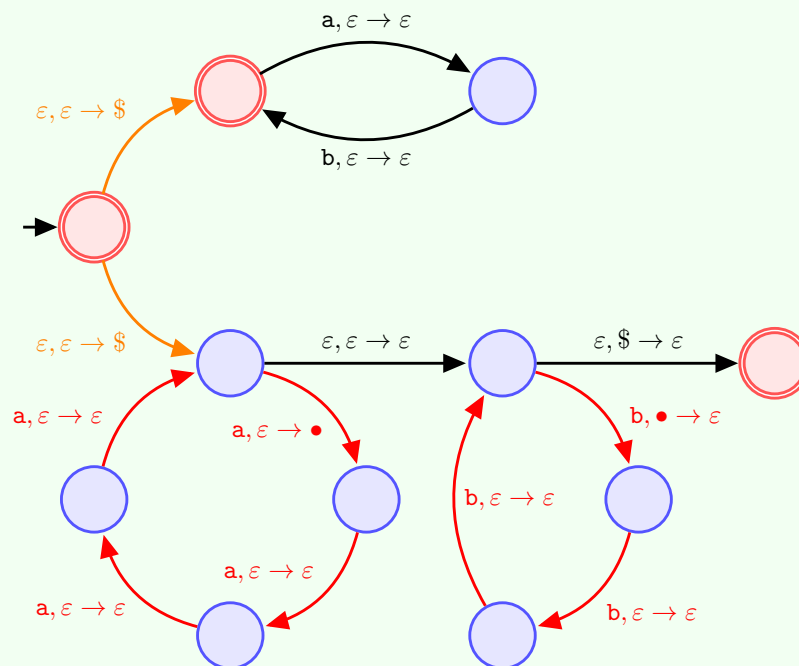
Here $\{(ab)^n \mid n \geq 0\} = (ab)^*$ is regular – no need to use the stack for it.

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

2) Abbreviated PDA:



Expanded PDA:



3) CFG

$$\begin{aligned}
 S &\rightarrow A \mid B \\
 A &\rightarrow abA \mid \varepsilon \\
 B &\rightarrow aaaaBbbb \mid \varepsilon
 \end{aligned}$$

- (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 $w = 011$ then $w^R = 110$.

Design PDAs and CFGs for each of the following languages

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

Hint: Consider the even and odd length cases first.

Solution

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

$$S \rightarrow bSb \mid a$$

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

$$S \rightarrow aSa \mid bSb \mid c$$

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

$$S \rightarrow aSa \mid bSb \mid \varepsilon$$

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

$$S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon$$

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

$$\begin{aligned} S &\rightarrow aAa \mid bAb \mid \varepsilon \\ A &\rightarrow aBa \mid bBb \mid a \mid b \\ B &\rightarrow aCa \mid bCb \mid aa \mid bb \\ C &\rightarrow S \mid \varepsilon \end{aligned}$$

- (6) **(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

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

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

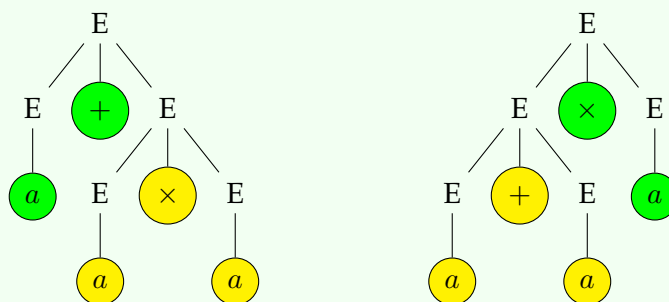
Now note that the following alternative CFG is *not* ambiguous:

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

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

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

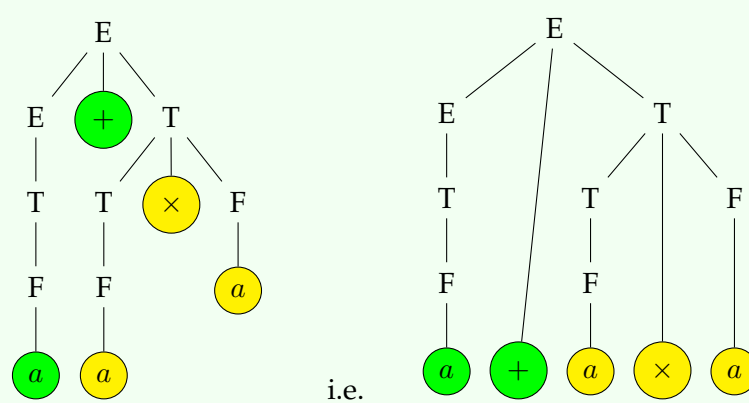
Solution



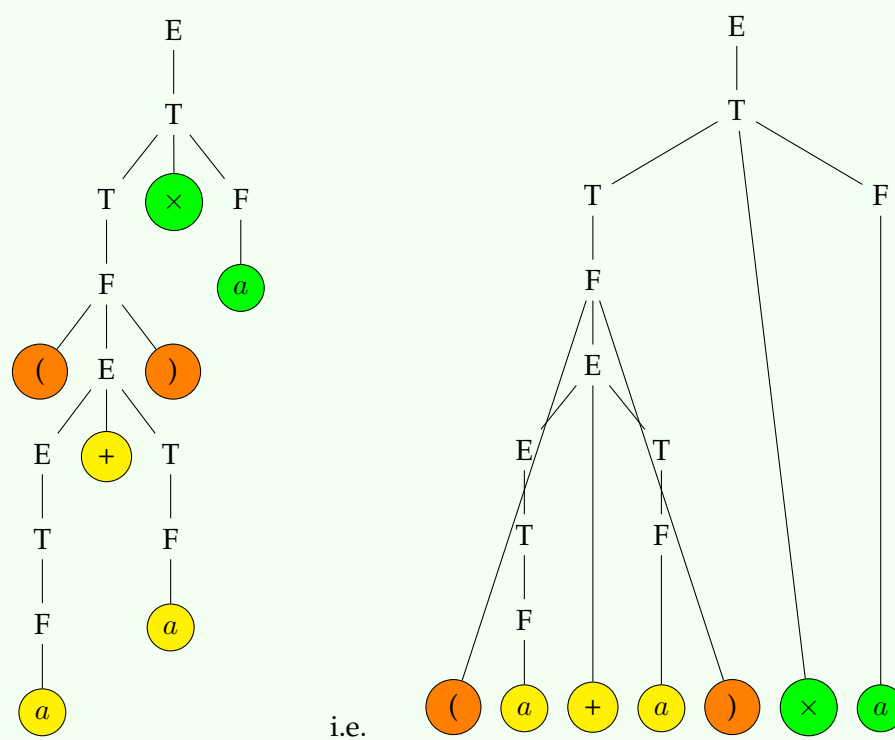
The first one: $a + (a \times a)$.

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

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



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



(1) Design CFGs generating the following languages.

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

$$\{b, aba, abb, bba, bbb, aabaa, \dots\}$$

- 2) The language of strings over $\{a, b\}$ 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) $\{a^i b^j \mid i, j \geq 0 \text{ and } i \geq j\}$

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

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

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

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

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

Solution

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

$$\begin{array}{lcl} S & \rightarrow & ASA \mid b \\ A & \rightarrow & a \mid b \end{array}$$

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 $\{a, b\}$ containing an equal number of a's and b's.

$$S \rightarrow SS \mid aSb \mid bSa \mid \varepsilon$$

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.

$$S \rightarrow SaSaSbS \mid SaSbSaS \mid SbSaSaS \mid \varepsilon$$

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

$$\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow aA \mid \varepsilon \\
B &\rightarrow aBb \mid \varepsilon
\end{aligned}$$

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

$$\begin{aligned}
S &\rightarrow AC \mid CB \\
A &\rightarrow aA \mid a \\
B &\rightarrow Bb \mid b \\
C &\rightarrow aBb \mid \varepsilon
\end{aligned}$$

or

$$\begin{aligned}
S &\rightarrow XbXaB \mid T \mid U \\
T &\rightarrow aTb \mid Tb \mid b \\
U &\rightarrow aUb \mid aU \mid a \\
X &\rightarrow a \mid b
\end{aligned}$$

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 $\{a, b\}$ containing more a's than b's.

$$\begin{aligned}
S &\rightarrow AaA \\
A &\rightarrow AA \mid aAb \mid bAa \mid aA \mid \varepsilon
\end{aligned}$$

or

$$\begin{aligned}
S &\rightarrow AS \mid aA \mid aS \\
A &\rightarrow AA \mid aAb \mid bAa \mid \varepsilon
\end{aligned}$$

This is because if a string w contains more a's than 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) $\{w\#x \mid w, x \in \{0, 1\}^* \text{ and } w^R \text{ is a substring of } x\}$

$$\begin{aligned}
S &\rightarrow TX \\
T &\rightarrow 0T0 \mid 1T1 \mid \#X \\
X &\rightarrow 0X \mid 1X \mid \varepsilon
\end{aligned}$$

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

$$\begin{aligned}
S &\rightarrow UPV \\
P &\rightarrow aPa \mid bPb \mid T \mid \varepsilon \\
T &\rightarrow \#MT \mid \# \\
U &\rightarrow M\#U \mid \varepsilon \\
V &\rightarrow \#MV \mid \varepsilon \\
M &\rightarrow aM \mid bM \mid \varepsilon
\end{aligned}$$

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

- 1) Give a PDA that recognizes B .
- 2) Give a CFG that generates B .

- (3) Let

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

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 $S \rightarrow SS$ 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?*