Assignment 3

1. In each case, what language is generated by CFG's below. Justify your claim (prove it!)

   (a) G with productions $S \to aSa|bSb|aAb|bAa,\ A \to aAa|bAb|a|b|\epsilon$

   Solution: Let's try a few derivations:
   $S \Rightarrow aSa \Rightarrow baSab \Rightarrow \cdots wSw^R$
   From here we can do $wSw^R \Rightarrow waAbw^R$, or $wSw^R \Rightarrow wbAaw^R$.
   Clearly $A \overset{*}{\Rightarrow} u$, where $u = u^R$. So all in all we get

   $$L(G) = \{wc_1uc_2w^R : w, u \in \{a, b\}^*, u = u^R, c_1, c_2 \in \{a, b\}, c_1 \neq c_2\}.$$

   (b) G with productions $S \to aS|bS|a$

   Solution: It is easily seen that $L(G) = \{wa : w \in \{a, b\}^*\}$.

   (c) $S \to SS|bS|a$

   Solution: It is not so easily seen that for this grammar G, we also have $L(G) = L$, where $L = \{wa : w \in \{a, b\}^*\}$. So let's prove it.
   Clearly $L(G) \subseteq L$ (All strings generated by G have to end in an a.)
   To see that $L \subseteq L(G)$ we show by an induction on $|w|$, that for any $w \in \{a, b\}^*$, we have $wa \in L(G)$.
   Basis: $|w| = 0$. This means that $|w| = \epsilon$. We have indeed $S \Rightarrow a$.
   Induction hypothesis: For any $w \in \{a, b\}^*$ where $|w| \leq n$, we have $wa \in L(G)$.
   Induction Step:
   Case 1: $w = av, |v| \leq n$. By the IH we have $S \overset{*}{\Rightarrow} va$. Then we can do the derivation $S \Rightarrow SS \Rightarrow aS \overset{*}{\Rightarrow} ava$.
   Case 2: $w = bv, |v| \leq n$. By the IH we have $S \overset{*}{\Rightarrow} va$. Now we can do the derivation $S \Rightarrow bS \overset{*}{\Rightarrow} bva$.

(d) G with productions $S \to SaS|b,\ \ S \to aT|bT|\epsilon,\ T \to aS|bS$.

Solution: Here again it might not be easy to see that $L(G) = \{a,b\}^*$, so we better prove it.

First we note that it is obvious that $L(G) \subseteq \{a,b\}^*$ (Any string in $L(G)$ is over $\{a,b\}$.)

We will show on an induction on $|w|$, that if $w \in \{a,b\}^*$, then $w \in L(G)$. The trick is that we need two IH's, namely $L(G) \subseteq \{a,b\}^*$, and $T \stackrel{*}{\Rightarrow} x$, where x is any string in $\{a,b\}^+$

Basis: $|w| = 0$. We have $\stackrel{*}{\Rightarrow} \epsilon$.

$|x| = 1$. We have $T \Rightarrow aS \Rightarrow a\epsilon = a$, and $T \Rightarrow bS \Rightarrow b\epsilon = b$,

Induction hypothesis: For all $w \in \{a,b\}^*$, if $|w| \leq n$, then $w \in L(G)$.

For all $x \in \{a,b\}^+$, if $|x| \leq n$, then $T \stackrel{*}{\Rightarrow} x$.

Induction Step:

Case 1: $w = av$. By IH, we have $S \stackrel{*}{\Rightarrow} v$. Then we can do the derivation $S \Rightarrow SaS \Rightarrow \epsilon aS \stackrel{*}{\Rightarrow} av$.

Case 2: $w = bv$. By IH, we have $T \stackrel{*}{\Rightarrow} v$. Now we can do the derivation $S \Rightarrow bT \stackrel{*}{\Rightarrow} bv$.

2. Find a CFG for each of the languages below.

(a) $L = \{a^n b^m : n \neq m - 1\}$

Solution:
Here $n \neq m - 1 \Leftrightarrow (n \geq m) \vee (n < m - 1)$
Hence the CFG:
$S \rightarrow A|B|\lambda$
$A \rightarrow aAb|aA|ab$
$B \rightarrow aBb|Bb|bb$

(b) $L = \{a^n b^m c^k : n = m \text{ or } m \neq k\}$

Solution:
Here $(n = m) \vee (m \neq k) \Leftrightarrow (n = m) \vee ((m < k) \vee (m > k))$
For $(n = m)$: $S \rightarrow A$ and $A \rightarrow aAb|\lambda$
For $(m > k)$: $S \rightarrow B$ and $B \rightarrow bBc|bB|b$
For $(m < k)$: $S \rightarrow C$ and $C \rightarrow bCc|Cc|c$
Therefore, $S \rightarrow A|B|C|DB|DC|EA|E$
where $D \rightarrow aA|a$ and $E \rightarrow cE|c$

(c) $L = \{w \in \{a, b\}^* : n_a(w) \neq n_b(w)\}$

Solution: $L = L_a \cup L_b$, where $L_a = \{w \in \{a, b\}^* : n_a(w) > n_b(w)\}$, and $L_b = \{w \in \{a, b\}^* : n_a(w) < n_b(w)\}$, and
L_a can be gerenated by

$$A \rightarrow a|aA|bAA|AAb|AbA$$

and L_b by

$$B \rightarrow b|bB|aBB|BBa|BaB.$$

For L we can then use $S \rightarrow A|B$.

(d) $\overline{L}$, where $L = \{w \in \{a, b\}^* : w = a^n b^n,\ n \geq 0\}$

Solution: We have
$\overline{L} = \{w \in \{a, b\}^* : w = a^n b^m, n \neq m\} \cup \{wbau : w, u \in \{a.b\}^*\}$. We then get
$S \rightarrow A|B|C$
$A \rightarrow aAb|aA|a$
$B \rightarrow aBb|Bb|b$
$C \rightarrow DbaD$
$D \rightarrow aD|bD|\lambda.$

3. In each case below, show that the grammar is ambiguous, and find an equivalent unambiguous grammar.

   (a) $S \to SS|ab|a$

   Solution:

   The grammar is ambiguous because, the string $aaba$ can be obtained by two different leftmost derivations:

   $S \Rightarrow SS \Rightarrow SSS \Rightarrow aSS \Rightarrow aabS \Rightarrow aaba$

   $S \Rightarrow SS \Rightarrow aS \Rightarrow aSS \Rightarrow aabS \Rightarrow aaba$

   An unambiguous version is: $S \to Sa|Sab|a|ab$

   (b) $S \to ABA,\ A \to aA|\epsilon,\ B \to bB|\epsilon$

   Solution:

   The grammar is ambiguous because the string a has two leftmost derivations:

   $S \Rightarrow ABA \Rightarrow aABA \Rightarrow a\epsilon BA \Rightarrow a\epsilon\epsilon A \Rightarrow a\epsilon\epsilon\epsilon = a$

   $S \Rightarrow ABA \Rightarrow \epsilon BA \Rightarrow \epsilon\epsilon A \Rightarrow \epsilon\epsilon a = a$

   An unambiguous version is:
   $S \to ABA|AB|BA|A|B|\lambda$
   $A \to aA|a$
   $B \to bB|b$

   (c) $S \to aSb|aaSb|\epsilon$

   The grammar is ambiguous because, the string $aaabb$ can be obtained by two leftmost derivations:

   $S \Rightarrow aSb \Rightarrow aaaSbb \Rightarrow aaa\epsilon bb = aaabb$

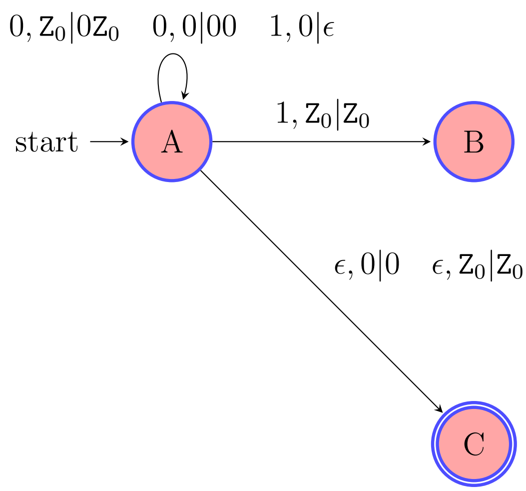
   $S \Rightarrow aaSb \Rightarrow aaaSbb \Rightarrow aaa\epsilon bb = aaabb$

   An unambiguous version is:
   $S \to A|\epsilon$
   $A \to aAb|B|ab$
   $B \to aaBb|aab$

4. Design a PDA to accept each of the following languages. You may design your PDA to accept either by final state or empty stack, whichever is more convenient.

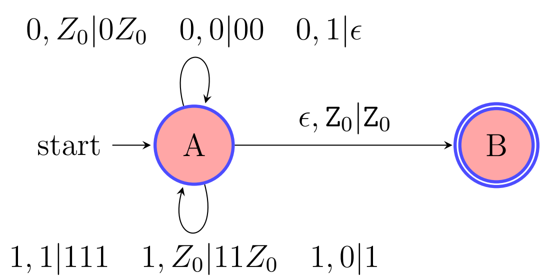
   (a) The set of strings over $\{0, 1\}$ such that no prefix has more 1's than 0's.

   Solution: The PDA is



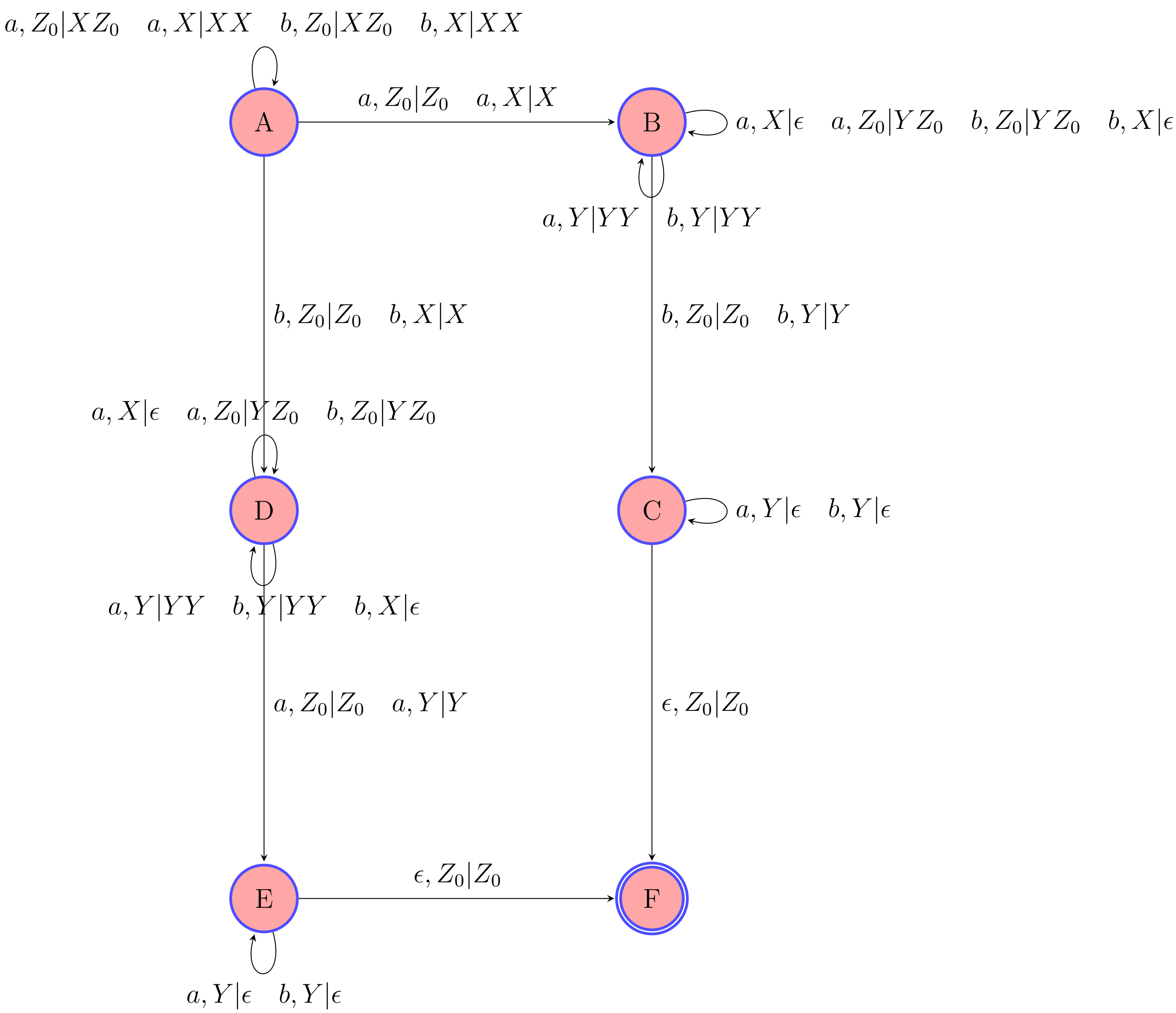
   (b) The set of strings with twice as many 0's as 1's.

   Solution: The PDA is



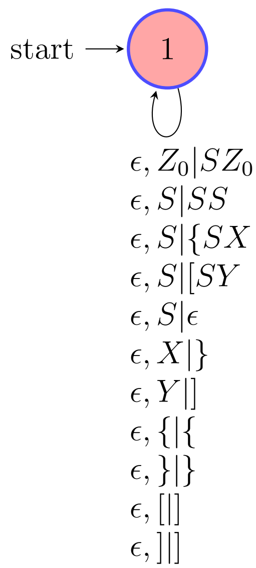
(c) The set of strings over $\{a, b\}$ that are *not* of the form ww, that is, not equal to any string repeated.

Solution: The PDA is



5. Construct a PDA corresponding to the context-free grammar
$S \to SS \mid \{SX \mid [SY \mid \epsilon$
$X \to \}$
$Y \to]$
Note that {, [,], and] are terminals.

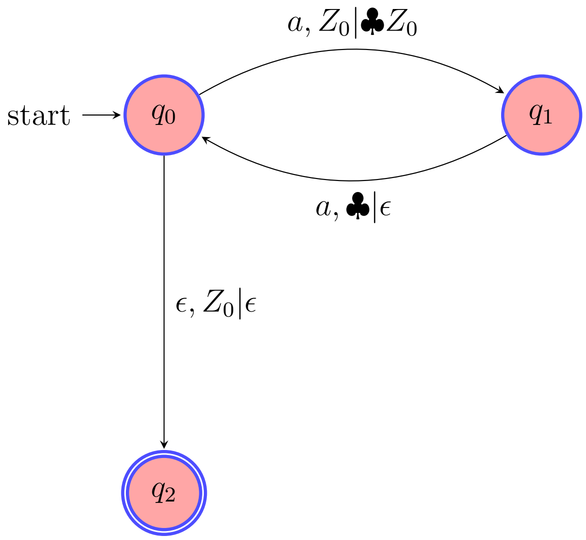
Solution:



6. Consider the PDA $P = \{\{q_0, q_1, q_2\}, \{a\}, \{\clubsuit, Z_0\}, \delta, q_0, Z_0, \{q_2\}\}$, where $\delta(q_0, a, Z_0) = \{(q_1, \clubsuit Z_0)\}$, $\delta(q_1, a, \clubsuit) = \{(q_0, \epsilon)\}$, and $\delta(q_0, \epsilon, Z_0) = \{(q_2, \epsilon)\}$.

   Construct a CFG (using the method in the text) corresponding to P.

   Solution:



$$S \to [q_0Z_0q_0] \mid [q_0Z_0q_1] \mid [q_0Z_0q_2]$$

$$[q_0Z_0q_0] \to \epsilon[q_1\clubsuit q_0][q_0Z_0q_0] \mid \epsilon[q_1\clubsuit q_1][q_1Z_0q_0] \mid \epsilon[q_1\clubsuit q_2][q_2Z_0q_0]$$

$$[q_0Z_0q_1] \to \epsilon[q_1\clubsuit q_0][q_0Z_0q_1] \mid \epsilon[q_1\clubsuit q_1][q_1Z_0q_1] \mid \epsilon[q_1\clubsuit q_2][q_2Z_0q_1]$$

$$[q_0Z_0q_2] \to \epsilon[q_1\clubsuit q_0][q_0Z_0q_2] \mid \epsilon[q_1\clubsuit q_1][q_1Z_0q_2] \mid \epsilon[q_1\clubsuit q_2][q_2Z_0q_2]$$

$$[q_0Z_0q_2] \to \epsilon$$

$$[q_1\clubsuit q_0] \to a$$

By deleting useless symbols and productions, and by renaming the variables, we get:

$$S \to T$$
$$T \to AT$$
$$T \to \epsilon$$
$$A \to a$$

7. Use the Pumping Lemma for CFL's to show that none of the following languages are context-free.

*****Notice: Answers are simplified *****

(a) $L_1 = \{ww \; : \; w \in \{a, b\}^*\}$ Solution: $z = a^n b^n a^n b^n$ is in the language. Now we show by

pumping lemma for CFG's that the language can not be generated by any CFG. It is easy to see, since $|vwx| \leq n$, it does not contain the whole z So we have the following cases

- $vwx = a^p; p \leq n$
- $vwx = a^p b^q; p + q \leq n$
- $vwx = b^q; q \leq n$
- $vwx = b^q a^p; q + p \leq n$

and there is at least an a^n and b^n which would not be affected by pumping lemma, while as a result the generated string by pumping lemma will not be in the language.

(b) $L_2 = \{a^n b^k \; : \; 0 \leq n \leq k^2\}$ Solution: In this case $z = a^{n^2} b^n$ is in the language and we

have the following cases for pumping lemma:

- $vwx = a^p; p \leq n$
- $vwx = a^p b^q; p + q \leq n$
- $vwx = b^q; q \leq n$

By pumping the first and last cases easily resulting in a string which is not in the language. But for the second case, we have $uv^{n+1}wxn + 1y = a^{n^2+np}b^{n+nq}$ implies that $n^2 + np = (n(1+q))^2 = n^2(1 + 2q + q^2) = n^2 + 2n^2q + n^2q^2$ that is, $np = 2n^2q + n^2q^2$, thus, $p = 2nq + nq^2$. Now let check what happen for $uv^2wx2y = a^{n^2+p}b^{n+q}$. That is $n^2 + p = n^2 + 2nq + nq^2 = n^2 + 2nq + q^2 = (n + q)^2$, which requires $n = 1$ however, we can not restrict n to be equal 1.

(c) $L_3 = \{a^n b^m c^k \; : \; 0 \leq n < m, \; n \leq k \leq m\}$ Solution: $z = a^n b^n c^{n+1}$ is in the language.

So we have the following cases for vwx:

- $vwx = a^p; p \leq n$
- $vwx = a^p b^q; p + q \leq n$
- $vwx = b^q; q \leq n$
- $vwx = b^q c^r; q + r \leq n$
- $vwx = c^r; r \leq n$

In first three cases uv^2wx^2y will not be in the language. In last two cases uv^0wx^0y will not be in the language.

8. Convert the following grammar into Chomsky normal form

$S \to aA|aBB$
$A \to aaA|\epsilon$
$B \to bB|bbC$
$C \to C|B$

Solution:

$$\begin{aligned} S \to&\ aA|aBB \\ A \to&\ aaA|\epsilon \\ B \to&\ bB|bbC \\ C \to&\ C|B \end{aligned}$$

Eliminate useless symbols:

$$\begin{aligned} S \to&\ aA \\ A \to&\ aaA|\epsilon \end{aligned}$$

Eliminate the ϵ-production:

$$\begin{aligned} S \to&\ aA|a \\ A \to&\ aaA|aa \end{aligned}$$

Eliminate unit product rules

$$\begin{aligned} S \to&\ aA|a \\ A \to&\ aaA|aa \end{aligned}$$

Break bodies of length more than two

$$\begin{aligned} S &\to aA|a \\ A &\to aX|aa \\ X &\to aA \end{aligned}$$

Change variables:

$$\begin{aligned} S &\to UA|U \\ A &\to UX|UU \\ X &\to UA \\ U &\to a \end{aligned}$$

9. (a) Show that the language $L = \{a^n b^n : a, b \in \{a, b\},\ n \text{ is not a multiple of } 5\}$ is context-free.

Solution: Let

$$L_1 = \{a^n b^n : n \geq 0\}$$

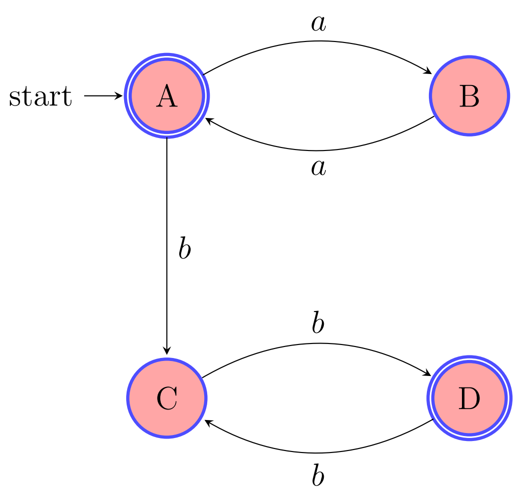
and

$$L_2 = \{w \in \{a, b\}^* : |w| \text{ is not a multiple of } 10\}.$$

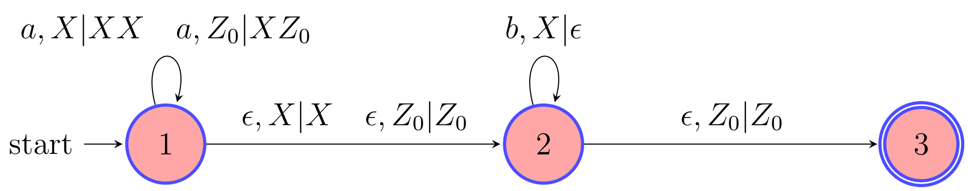
It is clear that L_1 is context-free, and L_2 is a regular language. Furthermore, we have $L = L_1 \cap L_2$, since the intersection of a context-free language and a regular language is context-free, then L is context-free.

(b) Let $L = \{a^n b^n : n \geq 0\}$, and $M = \{a^{2m} b^{2p} : m \geq 0, p \geq 0\}$. Construct a PDA for L and a DFA[1] for M. Then use the Cartesian construction to obtain a PDA for $L \cap M$.

Solution: DFA for M:



PDA for L:



Finally $L \cap M$:

state	input	stack	new state	new stack
$(A, 1)$	a	Z_0	$(B, 1)$	Z_0
$(A, 1)$	a	X	$(B, 1)$	XX
$(A, 1)$	ϵ	Z_0	$(A, 2)$	Z_0
$(A, 1)$	ϵ	X	$(A, 2)$	XX
$(A, 2)$	b	XX	$(C, 2)$	X
$(A, 2)$	ϵ	Z_0	$(A, 3)$	Z_0
$(C, 2)$	b	XX	$(D, 2)$	X
$(D, 2)$	b	XX	$(C, 2)$	X
$(D, 2)$	ϵ	Z_0	$(D, 3)$	Z_0

[1] Leave out the trap state