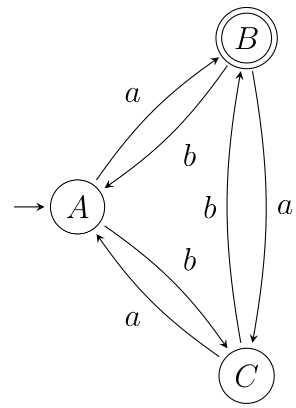
1. (*10 Points*) Give a regular grammar and DFA for the language,

$$L = \{w \mid (n_a(w) - n_b(w)) \mod 3 = 1\}$$

Solution:



$$\begin{aligned} A &\longrightarrow aB \,|\, bC \\ B &\longrightarrow aC \,|\, bA \,|\, \lambda \\ C &\longrightarrow aA \,|\, bB \end{aligned}$$

2. (*15 Points Each*) **For any two of the following three languages**, make a conjecture whether or not it is regular. Then prove your conjecture.

(a) $L = \{a^n b^p \mid |n - p| = 5\}$

Solution: By way of contradiction, assume that L is regular and let m be the positive integer guaranteed by the pumping lemma. Let $w = a^{m+5}b^m$, since $|w| = 2m + 5 > m$ we can write $w = xyz$ with $|y| \geq 1$ and $|xy| \leq m$. So $y = a^k$ for some $1 \leq k \leq m$. If we pump in one copy of y then $w_2 = a^{m+5+k}b^m$ and $|n - p| = |m + 5 + k - m| = k + 5 > 5$, hence $w_2 \notin L$, which contradicts our assumption that L was regular.

(b) $L = \{vwv \mid v, w \in \{a, b\}^* \text{ and } |v| = 3\}$

Solution: $L = L((aaa + aab + aba + abb + baa + bab + bba + bbb)(a + b)^*(aaa + aab + aba + abb + baa + bab + bba + bbb))$

(c) $L = \{a^n \mid n = k^3 \text{ for some integer } k\}$

Solution: By way of contradiction, assume that L is regular and let m be the positive integer guaranteed by the pumping lemma. Let $w = a^{m^3}$, since $|w| = m^3 > m$ we can write $w = xyz$ with $|y| \geq 1$ and $|xy| \leq m$. So $y = a^k$ for some $1 \leq k \leq m$. If we pump in one copy of y then $w_2 = a^{m^3+k}$. The value of $m^3 + k$ cannot be a perfect cube, since $m^3 + k > m^3$ the first possible perfect cube larger than m^3 is $(m + 1)^3 = m^3 + 3m^2 + 3m + 1$. For $m^3 + k = m^3 + 3m^2 + 3m + 1$, $k = 3m^2 + 3m + 1 > m$, so $w \notin L$, which contradicts our assumption that L was regular.

3. (*15 Points Each*) **For any two of the following three languages**, find context free grammars for them.

   (a) $L = \{a^n b^p c^t \mid p = 2n + 3t\}$
   Solution:

$$\begin{aligned} S &\longrightarrow AB \\ A &\longrightarrow aAbb \,|\, \lambda \\ B &\longrightarrow bbbBc \,|\, \lambda \end{aligned}$$

   (b) $L = \{a^n b^p c^t \mid t > n + p\}$
   Solution:

$$\begin{aligned} S &\longrightarrow ACc \\ A &\longrightarrow aAc \,|\, B \\ B &\longrightarrow bBc \,|\, \lambda \\ B &\longrightarrow Cc \,|\, \lambda \end{aligned}$$

   (c) $L = \{a^n b^p \mid p \le n \le 4p\}$
   Solution:

$$S \longrightarrow aSb \,|\, aaSb \,|\, aaaSb \,|\, aaaaSb \,|\, \lambda$$

4. (*10 Points*) Show that the following grammar is ambiguous.

$$\begin{aligned} S &\longrightarrow abAB \\ A &\longrightarrow bAB \,|\, b \,|\, \lambda \\ B &\longrightarrow BAa \,|\, A \,|\, a \,|\, \lambda \end{aligned}$$

Solution: There are many words that will produce two different parse trees, for example $w = ab$, $w = aba$, and $w = abb$

$$S \Rightarrow abAB \Rightarrow abB \Rightarrow ab$$

and

$$S \Rightarrow abAB \Rightarrow abB \Rightarrow abA \Rightarrow ab$$

or

$$S \Rightarrow abAB \Rightarrow abB \Rightarrow aba$$

and

$$S \Rightarrow abAB \Rightarrow abB \Rightarrow abBAa \Rightarrow abAa \Rightarrow aba$$

or

$$S \Rightarrow abAB \Rightarrow abbB \Rightarrow abb$$

and

$$S \Rightarrow abAB \Rightarrow abbABB \Rightarrow abbBB \Rightarrow abbB \Rightarrow abb$$

5. (*30 Points*) Consider the following grammar, G. In each conversion step below, follow the conversion or removal algorithm discussed in class.

$$\begin{array}{rcl} S & \longrightarrow & abAB \\ A & \longrightarrow & BC \mid b \mid \lambda \\ B & \longrightarrow & BAa \mid A \mid a \mid \lambda \\ C & \longrightarrow & Ab \end{array}$$

(a) Remove all λ-productions.
Solution: $V_N = \{A, B\}$

$$\begin{array}{rcl} S & \longrightarrow & abAB \mid abA \mid abB \mid ab \\ A & \longrightarrow & BC \mid C \mid b \\ B & \longrightarrow & BAa \mid Ba \mid Aa \mid A \mid a \\ C & \longrightarrow & Ab \mid b \end{array}$$

(b) Remove all unit-productions from your result in 5a.
Solution: The unit production dependency graph is $B \to A \to C$ so to

$$\begin{array}{rcl} S & \longrightarrow & abAB \mid abA \mid abB \mid ab \\ A & \longrightarrow & BC \mid b \\ B & \longrightarrow & BAa \mid Ba \mid Aa \mid a \\ C & \longrightarrow & Ab \mid b \end{array}$$

we add

$$\begin{array}{rcl} A & \longrightarrow & Ab \mid b \\ B & \longrightarrow & BC \mid b \\ B & \longrightarrow & Ab \mid b \end{array}$$

removing duplicates,

$$\begin{array}{rcl} S & \longrightarrow & abAB \mid abA \mid abB \mid ab \\ A & \longrightarrow & BC \mid Ab \mid b \\ B & \longrightarrow & BAa \mid Ba \mid Aa \mid Ab \mid BC \mid a \mid b \\ C & \longrightarrow & Ab \mid b \end{array}$$

(c) Remove all useless productions from your result in 5b.
Solution: All productions are useful, each variable has productions to terminal strings and the dependency graph from S reaches all other variables.

(d) Convert the grammar into Chomsky Normal Form from your result in 5c.
Solution: First convert the terminals that are needed,

$$
\begin{aligned}
S &\longrightarrow B_aB_bAB \mid B_aB_bA \mid B_aB_bB \mid B_aB_b \\
A &\longrightarrow BC \mid AB_b \mid b \\
B &\longrightarrow BAB_a \mid BB_a \mid AB_a \mid AB_b \mid BC \mid a \mid b \\
C &\longrightarrow AB_b \mid b \\
B_a &\longrightarrow a \\
B_b &\longrightarrow b
\end{aligned}
$$

Now combine variables to finish,

$$
\begin{aligned}
S &\longrightarrow D_2B \mid D_1A \mid D_1B \mid B_aB_b \\
D_1 &\longrightarrow B_aB_b \\
D_2 &\longrightarrow D_1A \\
A &\longrightarrow BC \mid AB_b \mid b \\
B &\longrightarrow D_3B_a \mid BB_a \mid AB_a \mid AB_b \mid BC \mid a \mid b \\
D_3 &\longrightarrow BA \\
C &\longrightarrow AB_b \mid b \\
B_a &\longrightarrow a \\
B_b &\longrightarrow b
\end{aligned}
$$