CS 420, Spring 2019
Homework 5 Solutions

1. Give regular expressions for the following languages: [A problem of this type should have been on Homework 3, but I forgot to do this.]

   (a) $\{w \in \{0,1\}^* | w$ contains exactly three 1's$\}$
   Soluiton: $0^*10^*10^*10^*$

   (b) $\{w \in \{0,1\}^* | w$ contains either 001 or 100 as a substring$\}$
   Solution: $(0 \cup 1)^*001(0 \cup 1)^* \cup (0 \cup 1)^*100(0 \cup 1)^*$

   (c) $\{w \in \{0,1\}^* | w$ has length at least 3 and the third symbol from the right in w is a $0\}$.
   Solution: $(0 \cup 1)^*0(0 \cup 1)(0 \cup 1)$

   (d) $\{w \in \{0,1\}^* | w$ does not contain 001 as a substring$\}$.
   [This one is tricky, but there is a short regular expression for this language.]
   Solution: $(1 \cup 01)^*0^*$

2. Let L be the language $\{w \in \{a,b\}^* | w$ contains exactly one more b than $a\}$.

   (a) Give a context-free grammar that generates L.
   Solution: We have actually given a solution to this problem in class, namely in the grammar

$$\begin{array}{rcl} S & \rightarrow & \varepsilon | aB | bA \\ A & \rightarrow & aS | bAA \\ B & \rightarrow & bS | aBB \end{array}$$

   we know that the variable B generates the strings with exactly one more b than a, so if we declare B to be the start symbol in the above grammar, we have one solution to the problem. To make this problem more interesting, I will give a different solution:

$$\begin{array}{rcl} S & \rightarrow & TbT \\ T & \rightarrow & aTb | bTa | TT | \varepsilon \end{array}$$

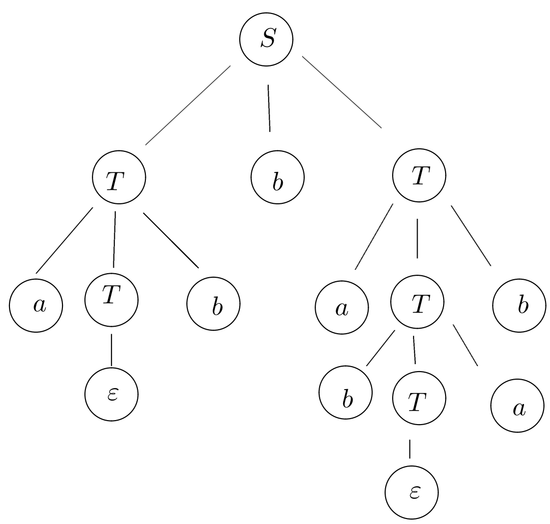
   You were not asked to explain how your grammar works, but here is an explanation for the above grammar. We know from class that the variable T generates the strings with the same number of a's and b's. Now suppose we have a string w with exactly one more b than a. Then, we have to show that w matches the rule $S \rightarrow TbT$. We think of a counter running along w where b counts as $+1$ and a counts as -1. The count at the end of w is $+1$, so we divide w up into ucv where c is the symbol read when the count first reaches $+1$. The count at the end of u is either 0 or 2, but if the count is 2 at the end of u, then the count must have been 1 somewhere in the middle of u, contradicting how we picked c, so the count is 0 at the end of u and c must be a b. Since u brings the counter from 0 to 0, u must have the same number of a's as b's, and since v brings the counter from 1 to 1, v also has the same number of a's as b's. Thus, $w = ubv$ matches the rule $S \to TbT$.

(b) Give a leftmost derivation and a parse tree in your grammar for the string $abbabab$.

Solution: A leftmost derivation in the second grammar is:

$$S \Rightarrow TbT \Rightarrow aTbbT \Rightarrow abbT \Rightarrow abbaTb \Rightarrow abbabTab \Rightarrow abbabab$$

A parse tree corresponding to this derivation is:



3. Give an unambiguous grammar for the language L of the previous problem.

(This is a difficult problem, but give it a try. As a hint, you can use three variables other than the start symbol. One variable generates strings with the same number of a's as b's, the second variable generates strings with the same number of a's as b's that have the additional property that every prefix has at least as many a's as b's, and the third variable generates all strings with the same number of a's as b's that have the additional property that every prefix has at least as many b's as a's.)

Solution:

$$\begin{array}{rcl} S & \rightarrow & WbT \\ T & \rightarrow & \varepsilon | aWbT | bVaT \\ W & \rightarrow & \varepsilon | aWbW \\ V & \rightarrow & \varepsilon | bVaV \end{array}$$

4. Let A be the language $\{a^n b^n | n \geq 0\}$ and let $B = \overline{A}$.

   Using closure of the context-free languages under union, give a context-free grammar that generates B.
   [Hint: You can express B as the union of three languages, one of which is $\overline{a^* b^*}$.]

   Solution: The language B can be expressed as $L_1 \cup L_2 \cup L_3$, where

$$L_1 = \overline{a^* b^*},$$

$$L_2 = \{a^n b^m | n > m\}\},$$

   and

$$L_3 = \{a^n b^m | n < m\}.$$

   The following grammar generates L_1:

$$\begin{array}{rcl} S_1 & \rightarrow & T_1 ba T_1 \\ T_1 & \rightarrow & aT_1 | bT_1 | \varepsilon \end{array}$$

   The language L_2 is generated by

$$S_2 \rightarrow aS_2 b | aS_2 | a$$

   and L_3 is generated by

$$S_3 \rightarrow aS_3 b | S_3 b | b$$

   Using the union construction, we get the following grammar for B

$$\begin{array}{rcl} S & \rightarrow & S_1 | S_2 | S_3 \\ S_1 & \rightarrow & T_1 ba T_1 \\ T_1 & \rightarrow & aT_1 | bT_1 | \varepsilon \\ S_2 & \rightarrow & aS_2 b | aS_2 | a \\ S_3 & \rightarrow & aS_3 b | S_3 b | b \end{array}$$

5. Let C be the language

$$\{0^n 1^m 2^p 3^q | n, m, p, q \geq 0 \text{ and } n > m \text{ and } p < q\}.$$

Using closure of the context-free languages under concatenation, give a context-free grammar for C.

Solution:

We can write $B = L_1 \circ L_2$ where

$$L_1 = \{0^n 1^m | n > m\}$$

and

$$L_2 = \{2^p 3^q | p < q\}$$

A grammar for L_1 is given by

$$S_1 \rightarrow 0S_1 1 | 0S_1 | 0$$

and a grammar for L_2 is given by

$$S_2 \rightarrow 2S_2 3 | S_2 3 | 3$$

By closure under concatenation, we obtain a grammar generating C by adding a new start symbol S and the rule $S \rightarrow S_1 S_2$.

6. Using the method from class, convert the regular expression $(a \cup \varepsilon)b^*$ into a context-free grammar.

Solution:

We have the grammars G_a, G_b, G_ε given by $S_a \rightarrow a$, $S_b \rightarrow b$ and $S_\varepsilon \rightarrow \varepsilon$, respectively, generating a, b and ε. Using the union construction, we get the grammar

$$\begin{aligned} S_{a\cup\varepsilon} &\rightarrow S_a | S_\varepsilon \\ S_a &\rightarrow a \\ S_\varepsilon &\rightarrow \varepsilon \end{aligned}$$

generating $a \cup \varepsilon$, and using the star construction, we obtain the grammar

$$\begin{aligned} S_{b^*} &\rightarrow S_{b^*} S_b | \varepsilon \\ S_b &\rightarrow b \end{aligned}$$

generating b^*, and finally, using the concatenation construction we obtain the grammar

$$
\begin{array}{rcl}
S & \rightarrow & S_{a\cup\varepsilon}S_{b^*} \\
S_{a\cup\varepsilon} & \rightarrow & S_a|S_\varepsilon \\
S_a & \rightarrow & a \\
S_\varepsilon & \rightarrow & \varepsilon \\
S_{b^*} & \rightarrow & S_{b^*}S_b|\varepsilon \\
S_b & \rightarrow & b
\end{array}
$$

generating $(a \cup \varepsilon)b^*$.

7. Give right regular grammars for the following languages:

   (a) $\{w \in \{0,1\}^* | w \text{ contains exactly three 1's}\}$
   Solution:

$$
\begin{array}{rcl}
S & \rightarrow & 0S|1T \\
T & \rightarrow & 0T|1U \\
U & \rightarrow & 0U|1W \\
W & \rightarrow & 0V|\varepsilon
\end{array}
$$

   (b) $\{w \in \{0,1\}^* | w \text{ contains either 001 or 100 as a substring}\}$
   Solution:

$$
\begin{array}{rcl}
S & \rightarrow & 0S|1S|0T_1|1U_1 \\
T_1 & \rightarrow & 0T_2 \\
T_2 & \rightarrow & 1W \\
U_1 & \rightarrow & 0U_2 \\
U_2 & \rightarrow & 0W \\
W & \rightarrow & 0W|1W|\varepsilon
\end{array}
$$

   (c) $\{w \in \{0,1\}^* || w| \geq 3 \text{ and the third symbol from the right in } w \text{ is a } 0\}$.
   Solution:

$$
\begin{array}{rcl}
S & \rightarrow & 0S|1S|0T \\
T & \rightarrow & 0U|1U \\
U & \rightarrow & 0|1
\end{array}
$$

   (d) $\{w \in \{0,1\}^* | w \text{ does not contain 001 as a substring}\}$.
   Solution:

$$
\begin{array}{rcl}
S & \rightarrow & 1S|0T|\varepsilon \\
T & \rightarrow & 1S|0U|\varepsilon \\
U & \rightarrow & 0U|\varepsilon
\end{array}
$$

8. Using the method from class, convert the DFA given in the solutions to Problem 1c on Homework 1 into a right regular grammar.

Solution:

Using the variables A_0, A_1, A_2 for the three states at the top of the diagram going left to right, and A_3 for the sink state, we get

$$\begin{array}{rcl} A_0 & \to & 1A_1|0A_3|\varepsilon \\ A_1 & \to & 0A_3|1A_2 \\ A_2 & \to & 0A_0|1A_1|\varepsilon \\ A_3 & \to & 0A_3|1A_3 \end{array}$$

9. Convert the following right regular grammar into an NFA.

$$\begin{array}{rcl} S & \to & 0S|1T|0 \\ T & \to & 0S|1U|1 \\ U & \to & 1T|0W \\ W & \to & 0W|1W|\varepsilon \end{array}$$

Solution:

We first replace the rules $S \to 0$ and $T \to 1$ to obtain:

$$\begin{array}{rcl} S & \to & 0S|1T|0Z \\ T & \to & 0S|1U|1Z \\ U & \to & 1T|0W \\ W & \to & 0W|1W|\varepsilon \\ Z & \to & \varepsilon \end{array}$$

We then obtain the following NFA: