COMP 3803 - Assignment 3 Solutions

Solutions written in LaTeX, diagrams drawn in `ipe`

Last updated on March 17, 2015

Note: S is the start variable for every given CFG, unless explicitly stated otherwise.

1. **Q:** Using the pumping lemma for regular languages, prove that the following languages with corresponding alphabets are not regular:

   A:

   - $L = \{b^4 a^M b^{2M} : M \geq 0\}$:
     Suppose that L is regular. Let p be the pumping length as given by the pumping lemma for regular languages, and $w = b^4 a^p b^{2p}$. By the pumping lemma, we can write $w = xyz$ where $|xy| \leq p$ and $|y| \geq 1$ such that $xy^i z \in L$ for every $i \geq 0$.
     Since $|xy| \leq p$, then $xy = b^4 a^k$ for some $k \leq p - 4$. If y contains a b, then xz starts with less than 4 b's, so that $xz \notin L$, contradicting the pumping lemma. If instead, y consists entirely of a's, then $xz = b^4 a^j b^{2p}$ for some $j < p$, so that $xz \notin L$ again, contradicting the pumping lemma. Thus, L cannot be regular. □
   - $L = \{a^M b^N a^{M+N} : M, N \geq 0\}$:
     Suppose that L is regular. Let p be the pumping length as given by the pumping lemma, and $w = a^p b^p a^{2p}$. By the pumping lemma, we can write $w = xyz$ where $|xy| \leq p$ and $|y| \geq 1$ such that $xy^i z \in L$ for every $i \geq 0$.
     Since $|xy| \leq p$, then xy consists entirely of a's, so $xz = a^k b^p a^{2p}$ for some $k < p$. But here, $k + p < 2p$, so $xz \notin L$, contradicting the pumping lemma. Therefore, L cannot be regular. □
   - $L = \{a^{n^3} : n \geq 0\}$:
     Suppose that L is regular. Let p be the pumping length as given by the pumping lemma, and $w = a^{p^3}$. By the pumping lemma, we can write $w = xyz$ where $|xy| \leq p$ and $|y| \geq 1$ such that $xy^i z \in L$ for every $i \geq 0$.
     Since $|xy| \leq p$, then $|y| \leq p$. Consider the string $xy^2 z$. We have:

     $$|xy^2 z| = |xyz| + |y| \leq p^3 + p < p^3 + 3p^2 + 3p + 1 = (p+1)^3$$

     Since $|y| \geq 1$, then $xy^2 z$ is strictly longer than w, but the shortest string in L longer than w is $a^{(p+1)^3}$. Since $xy^2 z$ has length strictly less than $(p+1)^3$, then $xy^2 z \notin L$, contradicting the pumping lemma, so L cannot be regular. □

2. **Q:** For the alphabet $\Sigma = \{(,)\}$, use the pumping lemma for regular languages to prove that the language consisting of matched parentheses is not regular. An example of a string in this language is "(()(()))".

   A: Let L be the given language. Suppose that L is regular. Then, let p be the pumping length as given by the pumping lemma, and $w = ({}^p){}^p \in L$. By the pumping lemma, we can write $w = xyz$, where $|xy| \leq p$ and $|y| \geq 1$, such that $xy^iz \in L$ for every $i \geq 0$.

   Since $|xy| \leq p$ and $w = ({}^p){}^p$, then xy (and thus y) must consist entirely of (. Thus, $xy^0z = xz = ({}^{p-|y|}){}^p$, and since $|y| \geq 1$, then xz is no longer a string of matched parentheses, so $xz \notin L$, contradicting the pumping lemma. Therefore, L cannot be regular. $\square$

3. **Q:** Let $\Sigma = \{0, 1\}$. Write CFGs that generate the following languages:

   A:

   - $\{W : W$ contains no more than three 1's$\}$:

   $$S \to R|R1R|R1R1R|R1R1R1R$$
   $$R \to 0R|\varepsilon$$

   - $\{W : W$ contains exactly two 1's$\}$:

   $$S \to R1R1R$$
   $$R \to 0R|\varepsilon$$

   - $\{W : W$ is of even length and starts and ends with the same symbol$\}$:

   $$S \to 0R0|1R1|\varepsilon$$
   $$R \to 00R|01R|10R|11R|\varepsilon$$

     Note: it is ambiguous whether or not ε is in the language, and either answer is accepted.

   - $\{0^n1^n : n \geq 1\} \cup \{1^{2m}0^m : m \geq 1\}$:

   $$S \to A|B$$
   $$A \to 0A1|01$$
   $$B \to 11B0|110$$

   - $\{W : W = 0X1$ or $1X0$ where X is a palindrome$\}$:

   $$S \to 0X1|1X0$$
   $$X \to 0X0|1X1|0|1|\varepsilon$$

     Note: a palindrome is a string w where $w = w^R$. This is why it is important to include the rules $X \to 0|1$, since without them, we would only recognize strings of the form ww^R, *i.e.* the set of palindromes of even length.

4. **Q:** Let $G = (V, \Sigma, R, S)$ be the context-free grammar in which $V = \{A, B, S\}$, $\Sigma = \{0, 1\}$, S is the start variable, and R consists of the rules:

$$S \to 0B|11A$$
$$A \to 0|0S|BAA$$
$$B \to 11|11S|ABB$$

A:

- By showing a sequence of productions, prove that $011011110 \in L(G)$:

$$S \Rightarrow 0B \Rightarrow 011S \Rightarrow 0110B \Rightarrow 011011S \Rightarrow 01101111A \Rightarrow 011011110$$

- Can you argue the following assertion: "Every string $W \in L(G)$ has the property that the number of 1's in W is equal to twice the number of 0's"?
Yes. More generally, we argue as well by induction (on the length of a string in the language) that any string derived from A has one more 0 than 11's, and any string derived from B has one more 11 than 0's.
For S, the least number of productions to obtain a string in the language is 2, and these two strings are 011 and 110, which both obey the desired property. Moreover, by induction $0B$ and $11A$ contain the same number of 0's as 11's, *i.e.* contain twice as many 1's as 0's.
For A, in the base case, 0 clearly has the desired property. By induction, since any string from S has the same number of 11's as 0's, then $0S$ has one more 0's than 11's. Similarly, by induction, there is $(1 + 1 - 1) = 1$ more 0's than 11's in BAA.
Finally, for B, in the base case, 11 clearly has the desired property. By induction, since any string from S has the same number of 11's as 0's, then $11S$ has one more 11's than 0's. Similarly, by induction, there is $(1 + 1 - 1) = 1$ more 11's than 0's in ABB.

□

5. **Q:** Convert the following CFGs (where $\Sigma = \{a, b\}$) to the Chomsky Normal Form:

A:

(a) $S \to SS; S \to abbS; S \to Sbba; S \to \varepsilon$

(1) Remove the start state from the right-hand side; make the new start state S_1:

$$S_1 \to S$$
$$S \to SS|abbS|Sbba|\varepsilon$$

(2) Remove ε-rules:

$$S_1 \to S|\varepsilon$$
$$S \to abb|abbS|bba|Sbba|SS$$

(3) Remove unit rules:

$$S_1 \to abb|abbS|bba|Sbba|SS|\varepsilon$$
$$S \to abb|abbS|bba|Sbba|SS$$

(4) Eliminate rules having more than 2 symbols on the right:

$$S_1 \to aA_1|aB_1|bC_1|SD_1|SS|\varepsilon$$
$$S \to aA_1|aB_1|bC_1|SD_1|SS$$
$$A_1 \to bA_2$$
$$A_2 \to b$$
$$B_1 \to bB_2$$
$$B_2 \to A_2S$$
$$C_1 \to bC_2$$
$$C_2 \to a$$
$$D_1 \to bC_2$$

(5) Eliminate rules of the form $A \to u_1u_2$ where u_1 and u_2 are not both variables:

$$S_1 \to C_2A_1|C_2B_1|A_2C_1|SD_1|SS|\varepsilon$$
$$S \to C_2A_1|C_2B_1|A_2C_1|SD_1|SS$$
$$A_1 \to A_2A_2$$
$$A_2 \to b$$
$$B_1 \to A_2B_2$$
$$B_2 \to A_2S$$
$$C_1 \to A_2C_2$$
$$C_2 \to a$$
$$D_1 \to A_2C_2$$

(b) $S \to aSbb; S \to bbSa; S \to \varepsilon$

(1) Remove the start state from the right-hand side; make the new start state S_1:

$$S_1 \to S$$
$$S \to aSbb|bbSa|\varepsilon$$

(2) Remove ε-rules:

$$S_1 \to S|\varepsilon$$
$$S \to abb|aSbb|bba|bbSa$$

(3) Remove unit rules:

$$S_1 \rightarrow abb|aSbb|bba|bbSa|\varepsilon$$
$$S \rightarrow abb|aSbb|bba|bbSa$$

(4) Eliminate rules having more than 2 symbols on the right:

$$S_1 \rightarrow aA_1|aB_1|bC_1|bD_1|\varepsilon$$
$$S \rightarrow aA_1|aB_1|bC_1|bD_1$$
$$A_1 \rightarrow bA_2$$
$$A_2 \rightarrow b$$
$$B_1 \rightarrow SB_2$$
$$B_2 \rightarrow bA_2$$
$$C_1 \rightarrow bC_2$$
$$C_2 \rightarrow a$$
$$D_1 \rightarrow bD_2$$
$$D_2 \rightarrow SC_2$$

(5) Eliminate rules of the form $A \rightarrow u_1u_2$ where u_1 and u_2 are not both variables:

$$S_1 \rightarrow C_2A_1|C_2B_1|A_2C_1|A_2D_1|\varepsilon$$
$$S \rightarrow C_2A_1|C_2B_1|A_2C_1|A_2D_1$$
$$A_1 \rightarrow A_2A_2$$
$$A_2 \rightarrow b$$
$$B_1 \rightarrow SB_2$$
$$B_2 \rightarrow A_2A_2$$
$$C_1 \rightarrow A_2C_2$$
$$C_2 \rightarrow a$$
$$D_1 \rightarrow A_2D_2$$
$$D_2 \rightarrow SC_2$$

6. **Q:** If L_1 is the language generated by the grammar in Question 5 (a), and L_2 is the language generated by the grammar in Question 5 (b), both of them *not* in the Chomsky Normal Form, create the grammars that generate:

A:

- $L_1 \cup L_2$:

$$S \rightarrow S_1|S_2$$
$$S_1 \rightarrow S_1S_1|abbS_1|S_1bba|\varepsilon$$
$$S_2 \rightarrow aS_2bb|bbS_2a|\varepsilon$$

- $L_1 \cdot L_2$:

$$S \to S_1 S_2$$
$$S_1 \to S_1 S_1 | abbS_1 | S_1 bba | \varepsilon$$
$$S_2 \to aS_2 bb | bbS_2 a | \varepsilon$$

- L_1^*:
Nothing needs to be done. $L_1^* = L_1$, since $\varepsilon \in L_1$, and the rule $S \to SS$ is already included in the grammar, so any multiple of 0 or more strings produced by S is in the language.

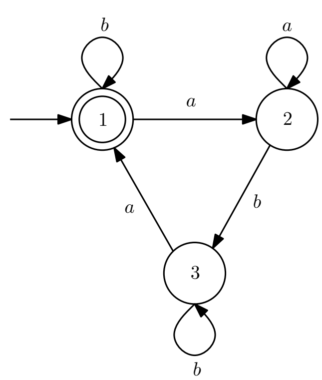
7. **Q:** Find a CFG generating $L = \{1^k 0^n 1^n 0^m 1^m : m, n, k \geq 0\}$. Give straightforward arguments to show that your answer is right. You need not formally prove that your arguments are right.

A:

$$S \to 1S | AA$$
$$A \to 0A1 | \varepsilon$$

We first observe that the rule A produces all strings of the form $0^n 1^n$ for $n \geq 0$, which can be seen by induction, whereby $A \Rightarrow \varepsilon$, and $A \Rightarrow 0A1 \Rightarrow \ldots \Rightarrow 00^{n-1}1^{n-1}1 = 0^n 1^n$. Finally, strings produced by S begin with any number of 1's, and end with AA. In this way, we see that indeed the strings produced by this grammar are of the form $1^k 0^n 1^n 0^m 1^m$ for $m, n, k \geq 0$.

8. **Q:** Write the context free grammar that generates the language accepted by the DFA of Question 8 in Assignment 2 (depicted below).



A: According to the construction in Theorem 3.3.1 of the course notes, the following grammar generates the language accepted by the given DFA:

$$R_1 \to bR_1 | aR_2 | \varepsilon$$
$$R_2 \to aR_2 | bR_3$$
$$R_3 \to bR_3 | aR_1$$

where R_1 is the start variable. □