TOC PROBLEM SET-6

SIDDHANT CHAUDHARY
BMC201953

Problem 1. Construct context-free grammar for the following languages. Justify.

$$L_1 = \{w \in \{a,b,c\}^* | |w|_a = |w|_b + |w|_c\}$$
$$L_2 = \{a^i b^j c^k | i + k = j\}$$

Solution: For L_1, the grammar is the following.

$$S \to aSR \mid RSa \mid SS \mid \epsilon$$
$$R \to b \mid c$$

Observe that L_2 is just the concatenation of the languages $\{a^n b^n \mid n \geq 0\}$ and $\{b^n c^n \mid n \geq 0\}$, and so the CFG may be constructed easily.

Problem 2. Given a language L, it's *reversal* is defined as the language $L^R = \{w^R | w \in L\}$, where w^R represents the reversal of the word w. Prove that context-free languages are closed under *reversal*.
Solution: Let L be a CFL, and we will show that L^R is a CFL as well. The idea is actually very simple. Suppose $G = (N, \Sigma, P, S)$ is a CFG for L. Make a new CFG $G' = (N, \Sigma, P', S)$, where the set of productions P' is as follows. For any production $X \to \alpha$ in P, add the production $X \to \alpha^R$ in P'. In that case, it is clear that $L(G') = L^R$, and this completes the proof.

Problem 3. It was described in the lecture how one can construct a right linear grammar for a regular language.

(1) Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, describe a construction to generate a *left-linear* grammar for the language accepted by A.
(2) Prove that a language generated by a *right-linear* (and *left-linear*) grammar is regular.

Solution: First, let A be a DFA for a regular language. Let q_0 be the initial state of A. Let G a context-free grammar as follows. The set of non-terminals of G is $Q \cup \{S\}$, where S is any symbol not in Q. Let the starting non-terminal of G be S. The set of terminals of G is simply Σ. Finally, the set of productions P is given by

$$P = \{q_0 \to \epsilon\} \cup \{q' \to qa | q \xrightarrow{a} q' \in \delta\} \cup \{S \to q | q \in F\}$$

So, in simple words, we add a production $S \to q$ for every final state $q \in F$, add the production $q_0 \to \epsilon$ and productions $q' \to qa$ for all transitions $q \xrightarrow{a} q'$ in A. Moreover, observe that G is a *left-linear* grammar. We claim that

$$L(G) = L(A)$$

Date: 25 September , 2020.

which will show that $L(A)$ is accepted by a *left-linear* grammar. First, suppose $w \in L(A)$, and let $w = a_1...a_k$. So, there is an accepting run

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} ... \xrightarrow{a_k} q_k$$

where q_k is a final state in A. So, the sequence of derivations

$$S \to q_k \to q_{k-1}a_k \to q_{k-2}a_{k-1}a_k \to ... \to q_0a_1...a_k \to a_1...a_k$$

shows that $w \in L(G)$. The fact that G accepts only those words which are in $L(A)$ is also clear, because the only way a sequence of derivations leads to a word without a non-terminal is when the last derivation is of the form $q_0w \to w$, and it is easy to see that in that case, $w \in L(A)$. So, this construction generates a *left-linear* grammar for a regular language. This completes the solution for (1).

Now, we prove (2). Let G be a *right-linear grammar*. Suppose the non-terminals of G are

$$N = \{S_1, ..., S_n\}$$

where without loss of generality the starting non-terminal is S_1. Now, we construct a Finite Automaton M as follows. Let the states of M be $N \cup \{q_{\text{end}}\} \cup Q$, where $q_{\text{end}} \notin N$ will be the *only* final state of M, and Q contains additional states which we will describe in a moment. Let the starting state of M be S_1, which is the starting non-terminal of G. Next, we describe the transitions in M. Suppose there is a production

$$S_i \to wS_j$$

for some $1 \leq i, j \leq n$, and some word $w \in \Sigma^*$ ($w = \epsilon$ is possible). In M, we make a w-path from the state S_i to S_j by adding new states if needed (and these new states will belong to Q). Next, if there is a production of the form

$$S_i \to w$$

for some word $w \in \Sigma^*$ ($w = \epsilon$ is possible), we add a w-path from S_i to q_{end} by again adding additional states if needed (which will belong to Q). Transitions in M correspond to productions in G , so it follows that $L(M) = L(G)$. Hence, it follows that $L(G)$ is a regular language, since it is acceptable by an ϵ-NFA, namely M.

Next, we consider the case where G is a *left-linear* grammar. Again, let the non-terminals in G be

$$N = \{S_1, ..., S_n\}$$

where the starting non-terminal is S_1. We construct a Finite Automaton M. Let the states of M be $N \cup \{q_{\text{init}}\} \cup Q$, where $q_{\text{init}} \notin N$ will be the *only* initial state of M, and Q contains additional states which we will now describe. Let the *only* final state of M be S_1. If there is a production

$$S_i \to S_jw$$

for some $1 \leq i, j \leq n$ and some word $w \in \Sigma^*$ ($w = \epsilon$ is possible), we make a w-path from S_j to S_i by adding new states if needed (and these new states will belong to Q). Next, if there is a production of the form

$$S_i \to w$$

for some word $w \in \Sigma^*$ ($w = \epsilon$ is possible), we add a w-path from q_{init} to S_i by again adding additional states if needed. It is clear that $L(G) = L(M)$, since transitions correspond to productions, and hence $L(G)$ is regular because it is accepted by an ϵ-NFA M. This completes the proof of (2).

Problem 4. Let the language $L_1 = \{w \in \{a,b,c\}^* | \forall u \cdot c \sqsupseteq w, |u|_a = |u|_b\}$, where $v \sqsupseteq w$ denotes that v is a prefix of w. In simple terms, L is the set of all those words whose each prefix that ends with c has equal number of a's and b's. Show that the language L_1 is context free.

Solution: First, we make some quick observations. Let $w \in L_1$. So, there are two possibilities. Either w contains the letter c, or $w \in \{a,b\}^*$. If w contains the letter c, consider the *first occurrence* of c in w. So, we can write

$$w = ucv$$

where $u \in \{a,b\}^*$ such that $|u|_a = |u|_b$ and $v \in L_1$. With this observation, we can easily make a context free grammar G for L_1. The grammar is given below.

$$S \to S_1 cS \mid \epsilon \mid S_2$$
$$S_1 \to aS_1 b \mid bS_1 a \mid S_1 S_1 \mid \epsilon$$
$$S_2 \to aS_2 \mid bS_2 \mid \epsilon$$

and here is the explanation. S is the starting non-terminal for G. From S, we can either go to ϵ (the empty word), or we can generate a word purely in $\{a,b\}^*$ (which is carried out by using the non-terminal S_2), or a word of the form ucv as discussed above. It was proven in class that the grammar

$$S_1 \to aS_1 b \mid bS_1 a \mid S_1 S_1 \mid \epsilon$$

generates all words $w \in \{a,b\}^*$ with $|w|_a = |w|_b$. This completes the construction and shows that the given grammar accepts L_1.

Problem 5. Show that the set of words w such that w is a palindrome and the number of $b's$ in w is a multiple of 4 is a CFL.

$$L = \{w | w = w^R, |w|_b \text{ is divisible by 4}\}$$

Solution: First, we make a couple of observations. Let $w \in L$. Observe that if $|w|$ is odd, then b cannot be the center letter of w (because the number of b's, being 0 mod 4, is even). So in any case, b is not the central letter of w. If w starts with an a, then

$$w = ava$$

for some palindrome v such that the number of b's in v is 0 mod 4. If w starts with a b, then

$$w = bvb$$

where v is a palindrome such that the number of b's in v is 2 mod 4. So keeping these things in mind, we construct the following grammar G for L.

$$S \to \epsilon \mid a \mid aSa \mid bS'b$$
$$S' \to aS'a \mid bSb$$

Here is the explanation. The non-terminal S' generates those palindromes in which the number of b's is 2 mod 4. The rest of the construction is clear from the discussion above.