Assignment 2
CMPUT 474

Pranav Wadhwa
1629510

February 14, 2022

1. Are the following languages context-free? Prove your answer in each case.
   (a) $\Sigma = \{0,1\}^*, L = \{xy \mid |x| = |y|, x \neq y\}$
   (b) $\Sigma = \{0,1\}^*, L = \{w \mid n_0(w) = n_1(w)$ and w includes the string '001'$\}$ where $n_0(w)$ and $n_1(w)$ denote the number of 0s and 1s in w respectively.

Solution: Yes, the language is context free. We observe in language L that it is made up of 2 strings x and y which are of same length but the strings are not equal. So the strings differ on some character i and we can make a correcponding context-free grammar around this rule.

$$S \to AB|BA$$
$$A \to XAX|0$$
$$B \to XBX|1$$
$$X \to 0|1$$

2. Let G be a CFG in Chomsky normal form, and $w \in L(G)$. How long is w if there is a derivation of w using p steps? Explain why.

Solution:

Every context-free grammar that is in Comsky normal form has rules of the form

$$A \to BC$$
$$A \to a$$

We have G a CGC in Chimsky normal form and $w \in L(G)$. w is derivated in p steps. Every derivattion in this form goes from A (non-terminal) $\to BC$ (non-terminal)(non-terminal) or A (non-terminal) $\to a$ terminal .

To derive a string of length w we will have w derivation of the first form from $A \to BC$ to a terminal $A \to a$ while expanding B as each step adds a total of $+1$ to each step. Then we will have a total of $w - 1$ derivation from $A \to C$ while expanding C to a terminal as the start symbol is already counted.

So the total number of derivations are $w + (w-1) = 2w - 1$. We are given than the total derivations are p.

So, $p = 2w - 1$ then $w = \frac{p+1}{2}$

3. Let G be a CFG. Explain an algorithm to determine whether $L(G)$ is finite. (Hint: use the pumping lemma).

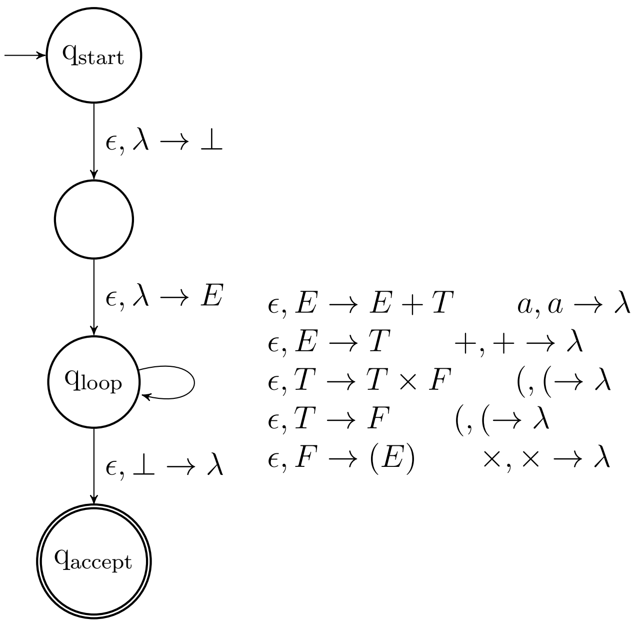
4. Convert the CFG

$$S \to E$$
$$E \to E + T | T$$
$$T \to T \times F | F$$
$$F \to (E) | \text{a}$$

to an equivalent PDA.

Solution:

A PDA of the CFC



5. Recall that the class of context-free language is closed under the regular operations, union, concatenation, and star. Prove that every regular language is context free by showing how to convert a regualar expression directly to an equivalent context-free grammar.

Solution: To convert a regular expressions to context-free grammar we will define rules on how to convert the rules of regular expressions to CFG.
Let R be a regular expression if R

1. a for some a in the alphabet Σ
   The language generated by the expression is $S \to a$.

2. ϵ
   The language generated by the expression is $S \to \epsilon$.

3. $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions.
   Let the CFG that generated the expression R_1 be S_1 and the expression R_2 be S_2. Then the union of these expressions is given by the grammar $S \to S_1|S_2$.

4. $(R_1 \circ R_2)$ where R_1 and R_2 are regular expressions.
   Let the CFG that generates the expression R_1 be S_1 and R_2 be S_2. Then the concatenation of these two expression is given by the grammar $S \to S_1S_2$.

5. (R_1^*) where R_1 is a regular expression.
   Let the CFG that generates the expression E be A. Then the star operation for the given expression is given by the grammar $S \to \epsilon|AS$.

We know that class of context-free grammar is closed under the regular operations, union, concatenation and start. The above rules 1 and 2 produce an CFC for every regular expression consisting of ϵ or a. The rest of the rules produce an equivalent CFG for the operation done on regular expression. Thus, every regular expression is context-free.

6. Consider the CFG G given by

$$S \to \mathsf{a}S\mathsf{b}S \mid \mathsf{b}S\mathsf{a}S \mid \epsilon$$

Prove that $L(G)$ is the set of strings with an equal number of as and bs.

Solution:

We want to show that every string s in $L(G)$ contains an equal number of as and bs with strong induction.

Base Case:
$|s| = 1$. If s is generated by G then the only possible string is ϵ which has $n_a = n_b = 0$
$|s| = 2$. If s is generated by G then the possible string are ab or ba where both have $n_a = n_b = 1$.
Inductive Hypothesis (IH):
G produces strings that have $n_a = n_b$ (number of as is equal to number of bs).
Inductive Step:
Assume that IH holds for all strings in G that have length n or less than n. Consider the string s of length $n+1$ and it is begin produces by rule S. We wil go over each rule and see how it is generated.
If we use rule $S \to \mathsf{a}S\mathsf{b}S$ then s consists of $\mathsf{a}w_1\mathsf{b}w_2$ where w_1 and w_2 are string derived from G and are of length less than n. From our IH we know than w_1 and w_2 have equal number of as and bs. So then length of string $|s| = |a| + |w_1| + |b| + |w_2|$ and string s adds one pair of a,b to w_1 and w_2 which makes the total number of as and bs in s to be equal.
Similarly, if we use the rule $S \to \mathsf{b}S\mathsf{a}S$ we find that it is made up of strings $\mathsf{b}w_1\mathsf{a}w_2$ where w_1 and w_2 are both derived from G are are of length less than n. Due to similar reasons given above the string s generated by this rule also have equal number of as and bs.

And lastly, if we use the rule $S \to \epsilon$ then the string s would be of length n which given by IH has equal number of as and bs.
Therefore we have proved that IH holds for string $n+1$ then $L(G)$ is the set of strings with equal number of as and bs.

7. Show the following grammar is ambiguous by drawing parse trees. Find an unambiguous grammer for this language.

$$S \to \mathsf{a}S\mathsf{b} \mid S\mathsf{b} \mid \epsilon$$

Solution:

Take this string generated from the grammer and its corresponding parse tree.

$$abb$$

Since the string has 2 parse trees therefore, the grammer is ambiguous.

8. Let P be the language of all palindromes over $\{0, 1\}$ containing equal numbers of 0s and 1s. Show that P is not context free.

Solution: We assume that P is a CFL and obtain a contradiction. Let p be the pumping length for P and select a string that is a palindrome and also contains equal numbers of 0s and 1s. So let $s = 1^p0^p1^p0^p$ be the string that is also in P. We can shorten s into $s = 1^p0^{2p}1^p$.
By the pumping lemma we may choose u, v, x, y, z such that $|vy| > 0$ and $|vxy| < p$. We can look over different cases of v and y.
case 1: v and y are in the middle and only consists of 0s. Then uxz would contain less 0s then 1s as $uxz = 1^p0^{2p-|vx|}1^p$ which in not in P.
case 2: v or y contains $m > 0$ amount of 0s and $n > 0$ amount of 1s from one side of the string but no 1s from the other size. Then the corresponding string uxz generated would have the following cases for both sides:

$$1^{p-m}0^{2p-n-|y|}1^p \quad \text{or} \quad 1^p0^{2p-n-|v|}1^{p-m}$$

Both of them don't form a palimdrome as number of 1s are not equal so $uxz \notin P$. Therefore we obtain a contradiction and P is not context free.

9. The language $L = \{ww : w \in \{a,b\}^*\}$ is not context free. However, show that the complement language, $\bar{L}$, is context free.

Solution:

Idea: We want to show that the language $\bar{L} = \{\{a,b\}^* \setminus \{ww : w \in \{a,b\}^*\}$ is context free. We can see that the language L consists of a string w concatenated to itself. The length of each string in L is an even number as $|w| + |w|$ would always be an even number. So $\bar{L}$ would contain all strings of odd length and also contain string of even length which satisfy this condition $\{xy : x, y \in \{a,b\}^*$ and $|x| = |y|$ and $x \neq y\}$.

Sol: $\bar{L} = \{w : w \in \{a,b\}^*$ and $|w|$ is odd $\} \cup \{xy : x, y \in \{a,b\}^*$ and $|x| = |y|$ and $x \neq y\}$ The context free grammar of the language is as follows:

$$
\begin{gathered}
S \rightarrow A|B|AB|BA|\epsilon \\
A \rightarrow X|XAX|XAY|YAX|YAY \\
B \rightarrow Y|YBY|YBX|XBY|XBX \\
X \rightarrow a \\
Y \rightarrow b
\end{gathered}
$$

This grammar creates odd length strings and when two string are concatenated by rule A and B they are different by 'a' and 'b' in the middle.

10. Is there a *universal* pusdown automaton? That is, there is a single pushdown automaton, say M such that given a string $s_G w$, where s_g is a string that describes a context free grammar and w is an input string, M accepts $s_G w$ if and only if G generates w? Explain your answer.