CS581 Theory of Computation: Homework #2

Due on February 1 2016 at 2:00pm

Harry H. Porter Winter 2016

Konstantin Macarenco

Problem 2.6
Give context-free grammars generating the following languages.

Problem 2.6 b
The complement of the language $\{a^n b^n | n \geq 0\}$

$$\begin{aligned} S &\rightarrow aSb \mid bY \mid Ya \\ Y &\rightarrow bY \mid aY \mid \epsilon \end{aligned}$$

Problem 2.6 d
$\{x_1 \# x_2 \# \cdots \# x_k | k \geq 1,$ each $x_i \in \{a, b\}^*,$ and for some i and $j, x_i = {x_j}^R\}$

$$\begin{aligned} S &\rightarrow ABC \\ A &\rightarrow D\#A \mid \epsilon \\ B &\rightarrow aBa \mid bBb \mid E \\ C &\rightarrow \#DC \mid \epsilon \\ D &\rightarrow bD \mid aD \mid \epsilon \\ E &\rightarrow \#DE \mid \# \end{aligned}$$

Problem 2.7
Give informal English description of PDAs for the languages in Exercise 2.6

Problem 2.7 b
The complement of the language $\{a^n b^n | n \geq 0\}$

The PDA would work as follows:

1. Push start symbol on the stack.
2. If the first symbol is a b then move to accept state, since $a^n b^n$ cannot start with b therefore this string is in compliment.
3. If the first symbol is a then push a on the stack, keep pushing a's for all consecutively following a's. When see first b after stream of a's, start consuming a's from the stack.
4. If stream is over and there is only start symbol - fail.
5. If start symbol is consumed when b is scanned, then go to accept $b > a$.
6. If followed by a, then go to accept, since string has a form of $a^i b^j a \cdots$.
7. Last option all bs are consumed, but there is still a remains on the stack $\rightarrow$ accept state since $a > b$

Problem 2.7 d
$\{x_1\#x_2\#\cdots\#x_k | k \geq 1, \text{ each } x_i \in \{a,b\}^*, \text{ and for some } i \text{ and } j, x_i = {x_j}^R\}$

A PDA that recognizes this language will nondeterministically check all possible combinations of x_i and x_j, by pushing character of x_i onto the stack, and consuming them when checking x_j. And skipping all other correct string sequences. If a pair $x_i = x_j^R$ is found, and all the rest of the input is in form of $\{x_1\#x_2\#\cdots\#x_k\}$ - Accept, fail otherwise.

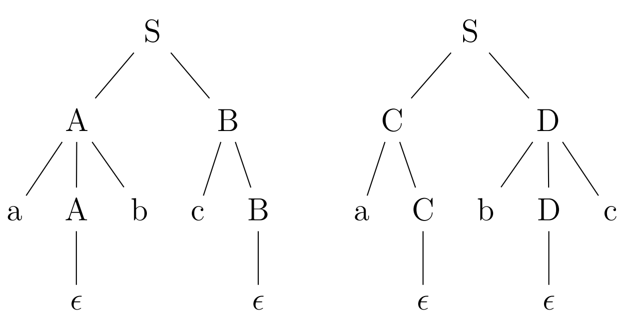
Problem 2.9
Give a context-free grammar that generates the language

$$A = \{a^i b^j c^k | i = j \text{ or } j = k \text{ where } i,j,k \geq 0\}$$

Is your grammar ambiguous? Why or why not?

$$\begin{aligned}
&S \to AB \mid CD \\
&A \to aAb \mid \epsilon \\
&B \to cB \mid \epsilon \\
&C \to aC \mid \epsilon \\
&D \to bDc \mid \epsilon
\end{aligned}$$

The language is ambiguous (inherently ambiguous). It contains all strings of format $a^y b^y c^y$, and these can be created by different derivations. For example derivation trees for string abc:



Therefore the language is ambiguous.

Problem 2.13
Let $G = (V, \sum, R, S)$ be the following grammar. $V = \{S < TU\}; \sum = \{0, \#\}$; and R is the set of rules:

$$\begin{aligned}
&\text{S} \to TT \mid U \\
&\text{T} \to 0T \mid T0 \mid \# \\
&\text{U} \to 0U00 \mid \#
\end{aligned}$$

Problem 2.13 a
Describe $L(G)$ in English.

Informally $L(G)$ is either two or more # separated by arbitrary number of 0's (zero or more) or zero or more zero followed by # and by twice as many zeros as in before #. More formally it is $\{0_1^{i_1}\#0_2^{i_2}\#0_3^{i_3}\#0_4^{i_4}\#\cdots\#0_k^{i_n} \mid \text{where } i_j \geq 0 \text{ and } k \geq 3\}$ or $\{0^n\#0^{2n}|n \geq 1\}$

Problem 2.13 b
Prove that $L(G)$ is not regular.

1. Assume that $L(G)$ is regular.
2. Consider the word $0^p\#0^{2p} \in L(G)$. By pumping lemma there exists a word $xyz \in L(G)$, and pumping length p, such that $|xy| \leq p$ and for all $i \geq 0$, $xy^iz \in L$. In case of word $= 0^p\#0^{2p}$, xy can only be in 0^p, since $|xy|$ msut be less thain p, however if $y = 0^k$ and we pump the word down to 0 then we get $0^{p-k}\#0^{2p} \notin L(G)$, which is a contradiction. Therefore $L(G)$ is not regular. $L(G)$ is no regular.

Problem 2.15
Give a counterexample to show that the following construction fails to prove that the class of context-free languages is closed under star. Let A be a CFL that is generated by th CFG $G = (V, \sum, R, S)$. Add the new rule $S \to SS$ and call the resulting grammar G'A. This grammar is supposed to generate $A*$.

This rule applied to grammar $G = (\{S\}, \{a, b\}, \{S \to aSb|\epsilon\}, S) + S \to SS$ we get grammar G' that can produce a word: $aababb \notin A^*$

Problem 2.19
Let CFG G be the following grammar.

$$S \to aSb \mid bY \mid Ya$$
$$Y \to bY \mid aY \mid \epsilon$$

Give a simple description of L(G) in English. Use that description to give a CFG for $\overline{L(G)}$, the compliment of $L(G)$.

L(G) is the language that produces all strings not in a^nb^n, i.e. compliment of of a^nb^n.

CFG of $\overline{L(G)}$:

$$S \to aSb \mid \epsilon$$

Problem 2.28

Give unambiguous CFGs for the following languages.

Problem 2.28 a
$\{w|$ in every prefix of w the number of a's is at least the number of b's $\}$

$$S \rightarrow aS \mid BS \mid \epsilon$$
$$B \rightarrow aBBb \mid \epsilon$$

Problem 2.28 b
$\{w|$ the number of a's and the number of b's in w are equal $\}$

$$S \rightarrow aA \mid bB \mid \epsilon$$
$$A \rightarrow bS \mid aAA$$
$$B \rightarrow aS \mid bBB$$

Problem 2.28 c
$\{w|$ the number of a's is at least the number of b's in $w\}$

$$S \rightarrow aA \mid bB \mid a$$
$$A \rightarrow aA \mid S$$
$$B \rightarrow aS \mid bBB \mid \epsilon$$

Problem 2.30
Use the pumping lemma to show that the following languages are not context free.

Problem 2.30 a
$L = \{0^n 1^n 0^n 1^n | n \geq 0\}$

Assume that L is regular. Let p be the pumping length, $s = 0^p 1^p 0^p 1^p$. By pumping lemma there exists $s = uv^i xy^i z \in L \; \forall i \geq 0$, where $|vxy| \leq p$ and $|vy| > 0$. There are two possible cases for v and y.

Case 1. v and y contain at most one type of symbols, then when we pump $i \rightarrow k$ we will get string with member of unequal length: $s = 0^p 1^{p+i} 0^p 1^p \notin L$ or $s = 0^{p+i} 1^p 0^p 1^p \notin L$ or $s = 0^p 1^p 0^{p+i} 1^p \notin L$ or $s = 0^p 1^p 0^p 1^{p+i} \notin L$

Case 2. v and y contain different types of symbols, then when i is pumped the resulting string will have members out of order, for example 00011100**0101010**111 $\notin L$.

s cannot be pumped without violating pumping lemma. Therefore L is not Context free.

Problem 2.30 d
$L = \{t_1\#t_2\#\cdots\#t_k | k \geq 2,\ \text{each}\ t_i \in \{a,b\}^*,\ \text{and}\ t_i = t_j\ \text{for some}\ i \neq j\}$

Assume that L is context free. Let p be the pumping length, $s = a^p b^p \# a^p b^p \in L$. By pumping lemma there exists $s = uv^ixy^iz \in L\ \forall i \geq 0$, where $|vxy| \leq p$ and $|vy| > 0$. Then

Case 1. v or y contain $\#$. Then if we pump i down to 0, $s = uv^0xy^0z = uxz$ and s no longer contains $\#$, hence $s \notin L$

Case 2. v and y on the left side of $\#$. Then when we pump up i left side becomes longer than the right side and $s \notin L$

Case 3. v and y on the right side of $\#$. Analogous to the previous case: when we pump up i right side becomes longer than the left side and $s \notin L$

s cannot be pumped without violating pumping lemma. Therefore L is no Context free.

Problem 2.31
Let L be the language of all palindromes over $\{0,1\}$ containing equal numbers of 0s and 1s. Show that B is not context free.

Assume that F is context free, then let p be the pumping length, $s = 0^p1^{2p}0^p \in L$. By pumping lemma there exists $s = uv^ixy^iz \in L$ for $\forall i \geq 0$, where $|vxy| \leq p$ and $|vy| > 0$. To prove that L is not context free we need to consider following cases

Case 1. vxy consists of only 1's. Then if we pump $i \rightarrow k$, then $s = 0^p1^{2p+k}0^p \notin L$, since number of 1's is greater than number of 0's.

Case 2. vxy consists of only 0's. Then if we pump $i \rightarrow p$, then $s = 0^{p+k}1^2p0^p \notin L$, since number of 0's is greater than number of 1's and s is no longer a palindrome.

Case 3. vxy consists of 0's and ones. Then if we pump $i = 2$, then $s = uv^2xy^2z \notin L$, since s is no longer a palindrome, i.e. character are out of order, for exaple for string 000111111000 let v be in the first 0's, and let y be in 1's then when pumping i up we get string of 0**001100110011**1111000 which is not a palindrome.

s cannot be pumped without violating the pumping lemma. Therefore L is not context free.

Problem 2.33
Show that $F = \{a^i b^j \mid i = kj$ for some positive integer $k\}$ is not context free.

Let p be the pumping length, $s = a^{q^2} b^q \in F$, where q is a prime, and $q > p$. By pumping lemma there exists $s = uv^i xy^i z \in L\ \forall i$, where $|vxy| \leq p$ and $|vy| > 0$. To prove that L is not context free we need to consider following cases:

<u>Case</u> 1. vxy consists of only a's, then if $v = a^k$ and $y = a^l$ and $l + k < q$ then $q = m + l + k$ in this case we get $t = \dfrac{(m+l+k)^2}{q}$, by pumping i down to 0 we get $wv^0xy^0z \in L$, but $t = \dfrac{(m)^2}{q}$ is not an integer $a^{m^2}b^q \notin L$, since q is prime.

<u>Case</u> 2. same logic applies to the case when vxy consists of b's onlye, then if $v = b^k$ and $y = b^l$ where $k + l < q$ by pumping i down to 0 we get $t = \dfrac{q^2}{m}$ is not an integer, and $a^{1^2}b^m \notin L$, since q is prime.

<u>Case</u> 3. vxy consists of a's and b's, then if $v = a^k$ and $y = b^l$, where $l < q$ and $k < q$, and some let $q = m + l$ and $q = n + k$, by pumping i, we get $t = \dfrac{(n+k)^2}{m+l}$, there are many possible combinations of n, m, k, l, that will make t not integer.

Therefore F is not regular.

Problem 2.35
Let G be a CFG in Chomsky normal form that contains b variables. Show that if G generates some string with a derivation having at least 2^b steps, $L(G)$ is infinite.

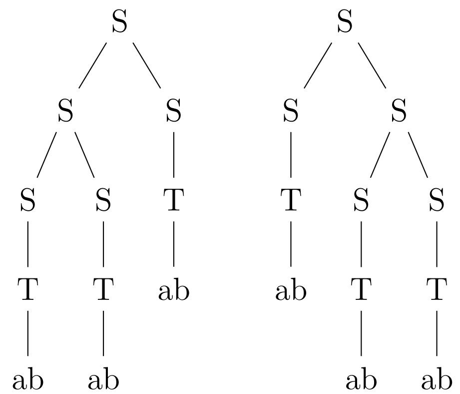
Tree of string w derived in 2^b steps will have height at least $b + 1$ (since derivation tree of height b can have at most $2^b - 1$ steps). Such a tree contains at least $b + 1$ variables, and therefore some variable is used more than once, i.e. string is derived by using recursion. Language $L(G)$ defined by recursive grammar G is infinite.

Problem 2.46
Consider the following CFG G: Describe $L(G)$ and show that G is ambiguous. Give an

$$S \to SS \mid T$$
$$T \to aTb \mid ab$$

unambiguous grammar H where $L(H) = L(G)$ and sketch a proof that H is unambiguous.

$L(G) = \{(a^n b^n)^i |$ where $i, n > 0\}$. $ababab \in L(G)$ and has two derivation trees



Therefore $L(G)$ is ambiguous.
Unambiguous grammar G' for this language is:

$$S \to TS \mid T$$
$$T \to aTb \mid ab$$

G' is unambigous, since for every string in $L(G)$ there is only one leftmost derivation.