Homework 2: Pumping Lemma, Context Free Languages

CS 212 Nature of Computation
Habib University
Homework 02 22

Fall 2023

1. Consider the grammar containing the following productions.

$$\begin{aligned} S &\to XaY \\ X &\to aX \mid \epsilon \\ Y &\to aY \mid bY \mid \epsilon \end{aligned}$$

   (a) $\boxed{\text{5 points}}$ Give a formal definition of the language generated by this grammar.

   (b) $\boxed{\text{5 points}}$ Draw the parse tree under this grammar for the string: aaaba.[1]

   (c) $\boxed{\text{5 points}}$ Argue whether this grammar is ambiguous.

2. $\boxed{\text{5 points}}$ Given an alphabet, Σ, consider the operation, $f : \Sigma^* \to \Sigma^*$, defined as follows.

$$f(w) = w_n \circ w_{n-1} \circ w_{n-2} \circ \dots w_1, \text{ where } w = w_1 \circ w_2 \circ w_3 \circ \dots \circ w_n.$$

   f is extended to apply to a given language, L, as follows.

$$f(L) = \{f(w) \mid w \in L\}.$$

   Prove or disprove the claim that the class of context-free languages is closed under f.

3. $\boxed{\text{5 points}}$ Prove or disprove the claim that the language, $L = \{x\#y \mid x, y \in \{0,1\}^*\}$, is context-free.

4. We are given the language, $L = \{0^i 1^j 2^k \mid i, j, k \geq 0, k \neq 1 \text{ or } i = j\}$.

   (a) $\boxed{\text{5 points}}$ Without using the pumping lemma, argue that L is not regular.

   (b) $\boxed{\text{5 points}}$ Show that the pumping lemma for regular languages applies to L.

   (c) $\boxed{\text{5 points}}$ What conclusion can you draw about the pumping lemma from the above observations?

[1] See this post on using the `forest` package to draw a parse tree.

Solution:

Problem 1:

(a) For the language L, our components for its grammar becomes:

- N is a non empty set of non terminal symbols; $N = \{S, X, Y\}$
- $\sum$ is a finite set of terminal symbols; $\sum = \{\varepsilon, a\}$
- P is the set of grammar rules; $P = \{S \to XaY, X \to aX \mid \varepsilon, Y \to aY \mid bY \mid \varepsilon\}$
- $S \in N$ is the start symbol; $S = S$

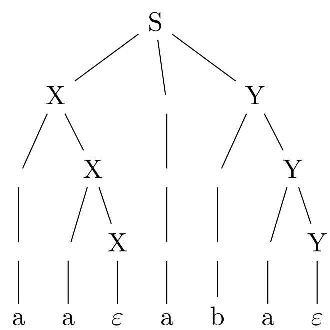
Formally the grammar for this language is,

$$G = \{\{S, X, Y\}, \{\varepsilon, a\}, \{S \to XaY, X \to aX \mid \varepsilon, Y \to aY \mid bY \mid \varepsilon\}, S\}$$

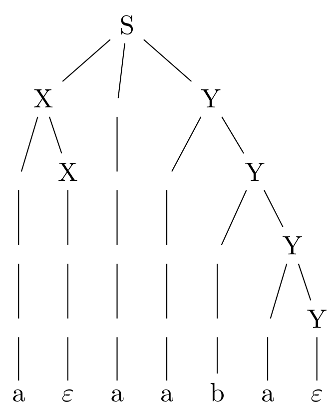
And the language can be defined as

$$L(G) = \{a^*(ba)^* \mid a, b \in \sum\}$$

(b) The parse tree for the string *aaaba* is as follows:



(c) This grammar is ambiguous because the string *aaaba* can be parsed in different ways. One is as shown in the previous part and the other is as follows:



Therefore, the grammar is ambiguous since we have found two different parse trees for the same string.

Problem 2: We are given the operation, $f : \Sigma^* \to \Sigma^*$, defined as follows:

$$f(w) = w_n \circ w_{n-1} \circ w_{n-2} \circ \ldots w_1, \text{ where } w = w_1 \circ w_2 \circ w_3 \circ \ldots \circ w_n.$$

Extending f to a language L, we have: $f(L) = \{f(w) \mid w \in L\}$

Essentially, f reverses the order of the characters in a string. Extending f to L means that we apply f to every string in L, then $f(L)$ consists of all the strings that we get by reversing each string in L.

Given that L is a context-free language over an alphabet $\sum$ and its context-free grammar G, we can assume, without loss of generality that L is in Chomsky Normal Form (CNF) - if not, then we can convert it to CNF. Then all production rules are of the form $A \to BC$, or $A \to a$, where A, B, and C are non-terminal symbols and a is a terminal symbol.

Then we construct a new grammar G' that generates the language $f(L)$ which consists of the reverse of every string in L. The non-terminal symbols of G' are the same as those of G, and the start symbol of G' is the same as that of G. The production rules of G' are as follows:

- For each rule of the form $A \to BC$ in G, we add the rule $A \to CB$ to G'
- For each rule of the form $A \to a$ in G, we add the rule $A \to a$ to G'

The reversal of order in the production rules is applied consistently throughout the grammar, therefore, this construction ensures that every string generated by G will be generated in reverse by G'.

Since we have constructed a context-free grammar for the language $f(L)$, we can conclude that $f(L)$ is context-free.

Hence proved that the class of context-free languages is closed under f.

■

Problem 3: We are given the language, $L = \{x\#y \mid x, y \in \{0,1\}^*\}$. This language consists of all the strings over the alphabet $\{0, 1, \#\}$ where # appears exactly once, and separates any sequence of 0's and 1's from any other sequence of 0's and 1's.

We can prove this language is context-free by providing a context-free grammar (CFG) that generates this language:

$S \longrightarrow X\#Y$
$X \longrightarrow 0X \mid 1X \mid \varepsilon$
$Y \longrightarrow 0Y \mid 1Y \mid \varepsilon$

Here the grammar can be formally defined as

$$G = \{\{S, X, Y\}, \{\varepsilon, 0, 1, \#\}, \{S \to X\#Y, X \to 0X \mid 1X \mid \varepsilon, Y \to 0Y \mid 1Y \mid \varepsilon\}, S\}$$

Since we can construct a CFG for this language, we can conclude that this language is context-free.

Problem 4: We are given the language, $L = \{0^i 1^j 2^k \mid i, j, k \geq 0, k \neq 1 \text{ or } i = j\}$.

(a) We can prove that L is not regular without the use of pumping lemma, by the intersection operation on L and any other language that is regular.

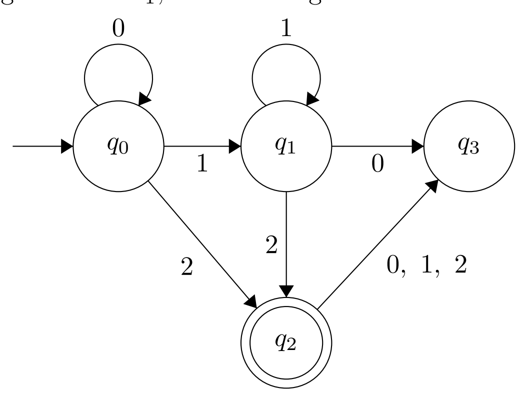
Regular languages are closed under intersection, that is,

$$L_1 \text{ is regular} \wedge L_2 \text{ is regular} \implies L_1 \cap L_2 \text{ is regular} \tag{1}$$

. Then by the contrapositive,

$$L_1 \cap L_2 \text{ is not regular} \implies L_1 \text{ is not regular} \vee L_2 \text{ is not regular} \tag{2}$$

Let us assume that the languages $L = \{0^i 1^j 2^k \mid i, j, k \geq 0, k \neq 1 \text{ or } i = j\}$ and $L_1 = \{0^* 1^* 2\}$ are regular. For L_1, the following finite automata can be constructed:



Then by the closure of regular languages under intersection, we know that $L \cap L_1$ will also be regular. Then $L \cap L_1 = L_2 = \{0^n 1^n 2 \mid n \geq 0\}$(we take 0 and 1 both to the power of n, since $i = j$, therefore, we have equal number of 0s and 1s). Suppose that

L_2 is also regular, then the pumping lemma should hold. Consider a string $s = 0^p 1^p 2$. Since $s \in L_2$, and $|s| \geq p$, the pumping lemma guarantees that s can be split into three pieces; $s = xyz$ such that:

1. for each $i \geq 0$, $xy^i z \in L_2$.
2. $|y| > 0$
3. $|xy| \leq p$

Condition 3 of the pumping lemma guarantees that y can only consist of 0s;

- y cannot be a 1, as that would imply $x = 0^p$, then $|xy| > p$, furthermore, then the first part of the string would more 1's than the second part, which is a contradiction
- By the same argument as above, y cannot be a combination of a 1's followed by a 2 either as that would imply $x = 0^p$, then $|xy| > p$

Then $xy = 0^p$, $y = 0^m$, and $x = 0^{p-m}$. Pumping y into the string:

$$xyyz = 0^p 0^m 1^p 2$$

$$xy^2 z = 0^{p+m} 1^p 2$$

This shows that there will inevitably be more 0s than 1s since $p + m > p$, then $xy^2 z \notin L_2$. Hence we arrive at a contradiction as the pumping lemme does not hold.

Then L_2 is not regular. We know that L_1 is regular, therefore, by (2), L is not regular. Hence proved.

$\blacksquare$