CS101 Homework Assignment 5

Tadius Frank

October 8, 2021

1. **Academic Honesty Declaration**

   In the process of finishing this homework:

   (a) I had conversations about the contents and solutions of this assignment with the following people: Melat Feseha

   (b) I consulted the following resources, such as books, articles, webpages:
   - https://wch.github.io/latexsheet/
   - https://en.wikibooks.org/wiki/LaTeX/Basics

   (c) I did not look at the answers of any other students.

   (d) I did not provide my answers to other students.

2. **Writing Component**

   (a) **Pumping Lemma**

   Show that the following language is not regular: $L = \{ww : w \ \epsilon \ \{a, b\}^* \ \}$.

   Solution:

   i. Pick a p
      $p = n$(any natural that doesn't yield an empty string)
   ii. Pick a String l in L
      $l = a^p b^p a^p b^p$
   iii. Break down w with the Pumping Lemma
      By P.L. $w = xyz$ s.t. $|xy| \leq p$ s.t. $xy^i z$ for all i.
   iv. Prove the break down is incorrect
      Then,
      $x = a^i$
      $y = a^j$
      where $i + j \leq p$
   v. Pick an i.
      $k = 2$,
      $xy^2 z = a^{p+j} b^p a^p b^p$,
      $a^{p+j} b^p a^p b^p \notin L$.

   Explanation: $p + j$ will not make the length of xy less than or equal to p. As a result, $a^{p+j} b^p$ will produce a string where the first half of the string l is not equivalent to the other half of the string l, so this string will not be in Language L. Thus, making L not a regular language with this contradiction.

(b) **Pumping Lemma**

Prove that $\{a^n ba^m ba^{m+n} : n, m \geq 1\}$ is not regular.

Solution:

i. Pick a p
   $p = n$(any natural that doesn't yield an empty string)

ii. Pick a String w in L
   $w = a^p ba^p ba^{2p}$

iii. Break down w with the Pumping Lemma
   By P.L. $w = xyz$ s.t. $|xy| \leq p$ s.t. $xy^i z$ for all i.

iv. Prove the break down is incorrect
   Then,
   $x = a^i$
   $y = a^j$
   where $i + j \leq p$

v. Pick an i.
   $k = 2$,
   $xy^2 z = a^{p+j} ba^p ba^{2p}$,
   $a^{p+j} ba^p ba^{2p} \notin L$.

Explanation: $p + j$ will not make the length of xy less than or equal to p. As a result, $a^{p+j} ba^p ba^{2p}$ will produce a string w with a distribution of a's that does not abide by ratio of a's in $a^n ba^m ba^{m+n}$ (the set notation), so w will not be in Language L. Thus, making L not a regular language with this contradiction.

3. **Regular Languages**

Are the following statements true or false? For each, if true, prove it. Otherwise, give a counter-example. For simplicity, we assume the alphabet = a, b, c (though any alphabet would do). Be careful, the answers can be tricky!

(1) Every subset of a regular language is regular.

Solution:

FALSE. The language $L = a^*b^*c^*$, is regular because it can be represented by a regular expression, as provided above. Let language $L_1 = a^nb^nc^n; n > 0$; in addition, L is subset of L. We can prove this subset is not regular by the Pumping Lemma:

(a) Pick a p
$p = n$(any natural that doesn't yield an empty string)

(b) Pick a String w in L
$w = a^pb^pc^p$

(c) Break down w with the Pumping Lemma
By P.L. $w = xyz$ s.t. $|xy| \leq p$ s.t. xy^iz for all i.

(d) Prove the break down is incorrect
Then,
$x = a^i$
$y = a^j$
where $i + j \leq p$

(e) Pick an i.
$k = 2$,
$xy^2z = a^{p+j}b^pc^p$,
$a^{p+j}b^pc^p \notin L$.

Explanation: $p + j$ will not make the length of xy less than or equal to p. As a result, $a^{p+j}b^pc^p$ will produce a string w that will not have equal amounts of a's, b's, and c's as required by the regular expression of L which is $a^nb^nc^n$. Thus, making L not a regular language with this contradiction.

(2) If L is regular, so is $L = \{xy \ : \ x\epsilon L \ and \ y\epsilon L\}$.

Solution:

TRUE. The statement that $y \notin L$ is equivalent to the statement that $y \ \epsilon \ \bar{L}$ It is given that L is regular and we know that under complement regular languages are closed, so $\bar{L}$ is a regular language. Thus, L' is a concatenation of two regular languages, so L' is regular since regular language are closed under concatenation.

(c) If L is a regular language, then so is $L = \{w \ : \ w\epsilon L \ and \ w^{rev}\epsilon L\}$.

Solution:
TRUE. The statement that $w^R \epsilon L$is equivalent to the statement $w\epsilon L^R$. As a result, if w must be in both L and L^R, that is same as saying that $L = L \cap L$. It is given that L is regular, and is since regular languages are closed under reversal L^R is regular. We know that regular languages are closed under intersection. So, the intersection of L and L^R which is $L^{'}$ must be regular.

4. **Context-free Language**
Let $L = w \ \ \{a, b\}^*| \ w \ = \ w^{rev}$, the language of palindromes.

(a) Design a CFG generating L, explaining informally why it is correct.

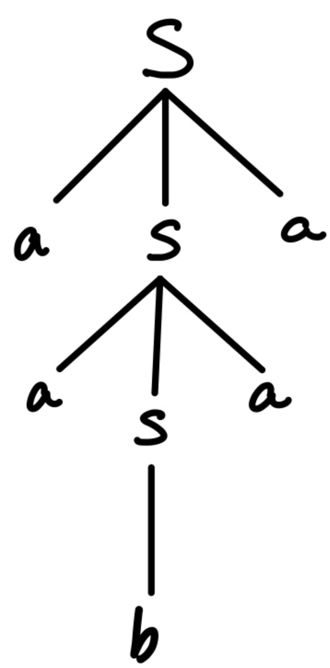
Solution:

$S \longrightarrow a|b|\epsilon|aSa|bSb$

The base cases a, b handles palindromes of odd lengths. The recursive cases aSa and bSb handle the cases of even length palindromes or more specifically we know that in other for a string to be a palindrome it must be wrapped at the head and tail by the same string character and another palindrome S(a non terminal state) can be in the middle. The base case epsilon ϵ handles empty strings produced by processing even length palindromes.

(b) Show the parse tree for $aabaa$ in your grammar for L.

Solution:



(c) Design a CFG generating the complement of L, explaining informally why it is correct.

Solution:

$S \longrightarrow a|b|aSa|bSb$

$T \longrightarrow a|b|\epsilon|aTb|bTa$

S handles the case that the given string is a palindrome.The base cases a, b handles palindromes of odd lengths. The recursive cases aSa and bSb handle the cases of even length palindromes or more specifically we know that in other for a string to be a palindrome it must be wrapped at the head and tail by the same string character and another palindrome S(a non terminal state) can be in the middle.
T handles the case that the given string is not a palindrome.The base cases a, b handles non-palindromes of odd lengths. The recursive cases aSa and bSb handle the cases of even length palindromes or more specifically we know that in other for a string to be a non-palindrome it must be wrapped at the head and tail by different string characters and another non-palindrome in the middle. The base case epsilon ϵ handles empty strings produced by processing even length non-palindromes.