CS 181 - Homework 8

Kuruvilla Cherian, Jahan
UID: 104436427
Section 1B

November 26, 2017

Problem 1.54

a

F is not regular because we can see that the non-regular language $\{ab^nc^n : n \geq 0\}$ is the same as $F \cap ab^*c^*$. Since regular languages are closed under intersection, but it is so for a non-regular language, then we know that F is non-regular.

b

We see that F satisfies the pumping lemma if we consider pumping length of 2 such that some $s \in F$ considering four cases for s:

1. $s = b^*c^*, x = \varepsilon$, in which case we select y as the first symbol of s and z as the rest.
2. $s = ab^*c^*, x = \varepsilon$, in which case we select y as the first symbol of s and z as the rest.
3. $s = aab^*c^*, x = \varepsilon$, in which case we select y as the first two symbols of s and z as the rest.
4. $s = aaa^*b^*c^*, x = \varepsilon$, in which case we select y as the first symbol of s and z as the rest.

In each case F satisfies the pumping lemma for $xy^iz : i \geq 0$ thus satisfying the pumping lemma.

c

The pumping lemma is in no way violated in part b since we only state that the pumping lemma holds for regular languages. At no point do we make a claim that non-regular languages fail to hold the three conditions of the pumping lemma.

Problem 1.63

a

The key idea here is to use the pumping lemma to be able to split the language A into two constituents that are all the odd and even amounts that union together to form A. This can be stated as

$$A_1 = xy^{2k}z : k \geq 0$$

$$A_2 = A - A_1$$

We see that A_1 is clearly infinite in nature as it is non-regular. We also see that A_2 is also infinite since $xy^{2k+1}z \subseteq A_2$, and thus have shown that A can be split into two disjoint infinite subsets given A is infinite and regular.

Problem 1.64

a

If A is non-empty then the N must recognize it by having at least one accept state $q \in F$ that can be reached from the start state q_0. Let us now consider a string s that is accepted by N while traversing the shortest path $q_0, q_1, ..., q$ such that $|s| = n$ and the length of the shortest path being $n + 1$. The states in the sequence must be pairwise distinct, because otherwise we would have a shorter path by removing repeated states. Since there are k states in N, and there are n distinct states, then we must have that $n \leq k$ which thereby implies that A contains a string of length at most k.

b

Let us consider the language where A contains all the strings of size at most k. The complement of A would be all strings greater than size k, which would make the statement fall through as it would contain strings of size greater than k. We also see that this language is regular and non-empty as we can construct an NFA with loops that can blow up in size greater than k.

c

To complement an NFA we can employ the algorithm to first convert the NFA to its corresponding DFA using the Power-Set which would create a DFA D of maximum size 2^k. Complementing this DFA then becomes the equivalent of complementing the language recognized by D, which is just a matter of reversing the accept and reject states with no change in the number of states. Thus D will still have at most 2^k states. Since there is no way to grow to a higher number of states, then the complement of A should still have at most 2^k states.

Problem 2.9

A CFG that generates the language A can be stated as the following tuple:

$$G = (V, \Sigma, R, S)$$

$$V = (S, E_{ab}, E_{bc}, C, A$$

$$\Sigma = a, b, c$$

With the following rules for G:

$$\begin{aligned}
&S \rightarrow E_{ab}C \mid AE_{bc} \\
&E_{ab} \rightarrow aE_{ab}b \mid \varepsilon \\
&Ebc \rightarrow bE_{bc}c \mid \varepsilon \\
&C \rightarrow Cc \mid \varepsilon \\
&A \rightarrow Aa \mid \varepsilon
\end{aligned}$$

We see this grammar is ambiguous because we can generate ε in multiple ways. Using the start terminal, we can use the first production to generate ε with E_{ab} and C or we can use the second production to generate ε with A and E_{bc}.

Problem 2.24

The key idea here is to split the grammar into three disjoint grammars G_1, G_2, G_3 such that $E = G_1 \cup G_2 \cup G_3$. We define these grammars as follows:

$$G_1 = \{a^i b^j : i > j\}$$

$$G_2 = \{a^i b^j : i < j \text{ } and \text{ } 2i > j\}$$

$$G_3 = \{a^i b^j : 2i < j\}$$

1. G_1:

$$\begin{aligned} S &\to aSb \mid aA \\ A &\to aA \mid \varepsilon \end{aligned}$$

2. G_2:

$$\begin{aligned} S &\to aSb \mid aBb \\ B &\to aBbb \mid aEbb \\ E &\to \varepsilon \end{aligned}$$

3. G_3:

$$\begin{aligned} S &\to aSbb \mid Bb \\ B &\to Bb \mid \varepsilon \end{aligned}$$

Thus by generating a CFG for each of the grammars and seeing that context-free grammars are closed under union, we show that E is CFL.

Problem 2.31

Assume B is context-free with a pumping length p. Choose $s = 0^p 1^{2p} 0^p \equiv uvxyz \in B$. Consider the following cases:

1. Both v and y contain only 0's or only 1's. Then by pumping up we get an imbalance in the number of zeros or one's such that $uv^2xy^2z \notin B$.
2. v contains only 0's and y only contains 1's (or vice-versa). Then by pumping up we no longer have a palindrome such that $uv^2xy^2z \notin B$.
3. If v and y contain both 0's and 1's then we violate the third condition of the pumping lemma, namely $|v| + |y| \leq p$.
4. Either v or y contains both 0's and 1's in which case pumping up would no longer make it a palindrome such that $uv^2xy^2z \notin B$.

Thus we see that s cannot be pumped in any consideration and thus we reach a contradiction, and thus B is no context-free.