Finite Automata Homework 5

Due: Thursday, Oct 15
Alexander Powell

1. (a) Let $B = \{1^k y | y \in \{0,1\}^*\}$ and y contains at least k $1s$ for $k \geq 1$. Prove B is regular.

   Solution:

   Proof. To prove B is regular we simply need to create a regular expression that generates all words in B. We know that all words need to start with a 1 and they need to have at least one more 1 somewhere in them. The regex is shown below:

   $$(1)(0)^*(1)(0 \cup 1)^*$$

   □

   (b) Let $C = \{1^k y | y \in \{0,1\}^*\}$ and y contains at most k $1s$ for $k \geq 1$. Prove C is not regular.

   Proof. To prove C is not regular we can use the pumping lemma for regular languages. Begin by assuming that C is indeed a regular language, then the pumping lemma applies. We can choose our s to be $s = 1^P 0 1^p$, $s = xy^i z$, $|xy| \leq P$, and $|y| \neq 0$ where P is our pumping length. From this, we can see the string xy in s is composed only of $1s$ since its length is less than P. Therefore, we can write

   $$s = 1^k 1^m 1^{P-k-m} 0 1^P,$$

   where $x = 1^k$, $y = 1^m$, and $z = 1^{P-k-m} 0 1^P$.

   By taking $i = 0$ so $xy^0 z = 1^k 1^{P-k-m} 0 1^P = 1^{P-k} 0 1^P$. Here we have the number of $1s$ in the second part of the string is larger than the number of $1s$ in the first part, so $xy^0 z$ is not an element of C. This is a contradiction to our assumption, and therefore C is not a regular language.

   □

2. Give the context free grammar for the complement of the language

   $$A = \{a^n b^n | n \geq 0\}$$

   Solution: The given language contains all words with some number of as followed by an equal number of bs. Therefore, the complement can be described as the language of all strings where the number of as and bs differ as well as all strings not in the pattern of $a^i b^j$. This can be written as:

   $$\bar{A} = \{a^i b^j | i > j\} \cup \{a^i b^j | i < j\} \cup (a \cup b)^* b (a \cup b)^* a (a \cup b)^*$$

   Then the union of these languages can be represented with the grammar:

   $$S \to S_1 | S_2 | S_3$$

   The language $\{a^i b^j | i > j\}$ can be represented with:

   $$S_1 \to aS_1 b | aS_1 | a$$

   and the language $\{a^i b^j | i < j\}$ can be represented with:

   $$S_2 \to aS_2 b | S_2 b | b$$

The regular expression is represented with

$$S_3 \to S_4 b S_4 a S_4$$

$$S_4 \to a S_4 | b S_4 | \varepsilon$$

Finally, we can write the context free grammar as a whole with:

$$S \to S_1 | S_2 | S_3$$

$$S_1 \to a S_1 b | a S_1 | a$$

$$S_2 \to a S_2 b | S_2 b | b$$

$$S_3 \to S_4 b S_4 a S_4$$

$$S_4 \to a S_4 | b S_4 | \varepsilon$$

3. **Solution:**

   (a) The language of this grammar can be defined as the union of two sets. All strings generated from T can be represented as the language

   $$S_1 = \{0^i \# 0^j | i, j \geq 0\},$$

   and all strings generated from U can be represented as the language

   $$S_2 = \{0^i \# 0^{2i} | i \geq 0\}.$$

   The language can be written as $S_1 S_1 \cup S_2$.

   (b) To prove $L(G)$ is not regular we will use closure properties. Begin by assuming that $L(G)$ is a regular language. Take the homomorphism H_1 where $H_1(0) = 0$, $H_1(\#) = \#$, and $H_1(1) = 00$. Also let $M_1 = H_1^{-1}(L(G))$ and $M_2 = M_1 \cap (0^* \# 1^*)$. Then we can write $M_2 = \{0^i \# 1^i | i \geq 0\}$.

   Now, let's take another homomorphism H_2 where $H_2(0) = 0$, $H_2(1) = 1$, and $H_2(\#) = \varepsilon$ and $M_3 = H_2(M_2) = \{0^n 1^n | n \geq 0\}$.

   Since M_1, M_2, and M_3 are obtained through $L(G)$, using closure properties that preserve regularity, and we see that M_3 is clearly a non-regular language, we have a contradiction to our original assumption. Therefore $L(G)$ is not regular.

4. Give the context free grammar for the language

$$L = \{a^i b^j c^k | i + j \neq k\}$$

Solution: It's easiest to begin by creating a grammar that generates the complement of the language (ie. when $i + j = k$). To start, we can write the two production rules

$$C \to aCc | B$$

$$B \to bBc | \varepsilon$$

Now, to change this grammar to generate words in the language where $i+j \neq k$ we can add the following production rules to ensure cases where $i+j<k$

$$S_1 \rightarrow Lc$$

$$L \rightarrow Lc|C$$

Now, we need to create production rules when $i+j>k$, shown below.

$$S_2 \rightarrow R_b b|R_a a$$

$$R_a \rightarrow R_a a|R_b|C$$

$$R_b \rightarrow R_b b|B$$

By putting all the rules together to form one context free grammar, we get:

$$S \rightarrow S_1|S_2$$

$$S_1 \rightarrow Lc$$

$$L \rightarrow Lc|C$$

$$S_2 \rightarrow R_b b|R_a a$$

$$R_a \rightarrow R_a a|R_b|C$$

$$R_b \rightarrow R_b b|B$$

$$B \rightarrow bBc|\varepsilon$$

$$C \rightarrow aCc|B$$