CSE 105: Homework Set 4

Joshua Wheeler

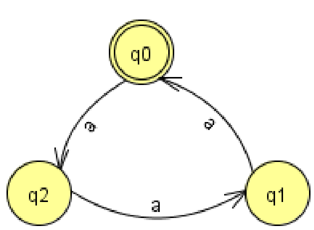
April 29, 2014

1. Minimum Pumping Length

   Part 1C:
   This regular expression has a minimum pumping length of 1. To see this, let x = ϵ, y = 0, z = ϵ. This string is in the language and can be pumped. There is no other string that can be pumped that has a pumping length less than this string. The minimum length cannot be 0 since the empty string is in the language and cannot be pumped.

   Part 1H:
   The minimum pumping length is 3. To see this let x = 1, y = 0, z = 0. This is the minimum string that can be pumped. The minimum length cannot be 2 because 10 is not in the language. 100 is in the language and can be pumped.

2. Proving Regularity
   Part 2A:
   This language is regular and can be described by the language which recognizes a string with a number of a's that has a cardinality which is divisible by 3. The following DFA describes the language (assume q0 is the initial state):



   Part 2B:
   This language is not regular. Assume for a contradiction that the language

is regular. Let $S = a^{p!} = a^p a^{(p-1)} a^{(p-2)} ... a^2 a^1 = a^{(p+(p-1)+(p-2)+...+2+1)}$ By the pumping lemma, $xy = a^p$ since $|xy| \leq p$ So we must chop up the string where $x = a^r, y = a^s, z = a^t$ where $r + s = p, t = (p-1)(p-2)...(2)(1)$. So $r + s + t > p$ as required. When we pump this string, we can see that it is not in the language. For example, $xyyz = a^r a^{2s} a^t = a^p a^s a^t = a^p a^s a^{(p-1)(p-2)...(2)(1)}$ which is not necessarily in the set of outputs for $a^{p!}$.

3. Proving Regularity
Part 3A:
This statement is false. One counter is example is if we define a language

$$M = \{w \mid \text{ w accepts } \Sigma^*\}$$

where $\Sigma = 0, 1$ and let

$$L = \{0^n 1^n \mid n \geq 0\}$$

It is clear that $L \subseteq M$. Section 1.4 in the book proved that L is non-regular. We also know that M is regular and can recognize any possible finite string of 1's and 0's. Therefore we have shown that

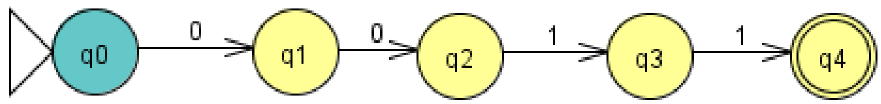
$$L \subseteq M \text{ and L is not regular } \quad \not\Longrightarrow \quad \text{M is not regular.}$$

Part 3B:
This statement is false. One counter example is if we define the languages

$$B = \{0^n 1^n \mid n \geq 0\}$$

$$A = \{w \mid \text{w accepts } 0011\} = \{0011\}$$

It is clear that $A \subseteq B$ since A is just one of the possible strings in B, namely when n=2. Section 1.4 of the book proved that B is not regular. We can also see that A is regular and can be recognized by the following NFA:



Therefore we have shown that

$$A \subseteq B \text{ and B is not regular } \quad \not\Rightarrow \quad \text{A is not regular.}$$

4. Context-Free Grammars
   Part 4A: L(G) is any string of variables representing a pair of parenthesis ”()” which may be either nested within each other and/or concatenated together. An example of nesting is either ”(())” or ”((()))”. An example of concatenation would be ”()()” or ”()()()”. An example with both nesting and concatenation would be ”(())()”.

   Part 4B: L(G) may be described as

   $$L(G) = \{[(^n)^n]^* \mid n \geq 0\}$$

   Note that the [ ] symbols are in place to show the precedence of the * operator since the ( ) symbols are being used as our variables.

   Suppose for a contradiction that this language is regular. Then choose a string $S = xyz = (^p)^p$ where by the pumping lemma $x = (^r, y = (^s, z =)^p$ where $r+s = p$. We can see that this language cannot be pumped since $xyyz = (^r(^s(^s)^p$. This string yields more ”(” than there are ”)”.