HW7: Pump It Up

Solution Key

Include your full name*, CS login, and the problem number(s) on each piece of paper you hand in, and please staple your pages together before handing in.*

While collaboration is encouraged in this class, please remember not to take away notes from collaboration sessions other than your scheduled lab section.

Problem 1

Consider the language

$$L = \{0^m 1^n : m \neq n\}$$

Prove that L is not regular.

Suppose L is regular. Then L^c is regular because regular languages are closed under complement. Additionally $0^*1^* \cap L^c$ is regular because 0^*1^* is regular and regular languages are closed under intersection. However,

$$0^*1^* \cap L^c = \{0^m 1^n : m = n\} = 0^n 1^n$$

Because $0^n 1^n$ is not regular, L cannot be regular.

Problem 2

*A string x is a **prefix** of string y if there exists a string z such that $xz = y$. A string x is a **proper prefix** of y if it is also true that $x \neq y$. Define the operation* NotPrefix *on a language A as follows:*

$$\text{NotPrefix}(A) = \{w \in A \mid w \textit{ is not the proper prefix of any string in } A\}$$

Show that the class of regular languages is closed under NotPrefix*. That is to say, if A is a regular language, so is* NotPrefix(A)*.*

Let $M = (Q, \Sigma, \delta, q_0, F)$ be an *NFA* recognizing A, where A is some regular language. We construct $M' = (Q, \Sigma, \delta, q_0, F')$ recognizing NotPrefix(A), where $F' = \{q \mid q \in F$ and there is no path of length ≥ 1 from q to an accept state $\}$. $w \in$ NotPrefix(A) iff $w \in A$ and there is no string x such that $wx \in A$. If M recognizes A, this is exactly what M' will recognize, because a string will lead to an accepting state in M' iff that string was accepted by M and there was no way for any string wx to be accepted by M.

The following questions are lab problems. *Please remember to prepare a solution for your assigned problem before going to your lab section.*

Lab Problem 1

Show that the following two languages are not regular:

a. $L_a = \{a^{pq} : p \text{ and } q \text{ are both prime and } p \neq q\}$

b. $L_b = \{w : w \text{ is a palindrome}\}$

Suppose L_a is regular. Applying the pumping lemma for regular languages, every string of length greater than the pumping length m can be broken up into three substrings rst, where $|rs| < m$ and $|s| > 0$, such that $rs^n t \in L_a$ for any $n \geq 0$. Choose $n = (|r| + |t|)(2 + |s|)$. Then the string $rs^n t$ is of length:

$$\begin{aligned}
&(|r| + |t|) + |s|(|r| + |t|)(2 + |s|) \\
=&(|r| + |t|)(1 + |s|(2 + |s|) \\
=&(|r| + |t|)(1 + 2|s| + |s|^2) \\
=&(|r| + |t|)(1 + |s|)(1 + |s|)
\end{aligned}$$

Since $|s| > 0$, this number is not the product of exactly two prime numbers, unless $|r| + |t| = 1$. If that were the case, then we could instead choose $n = 0$, and so have the string $rt = a$ of length 1. But the exponent is not the product of two different prime numbers. In either case, the pumped string is not in L_a, which contradicts the pumping lemma. Therefore, L_a is not regular.

Suppose L_b is regular. Once again, we can apply the pumping lemma. Specifically, let us apply the pumping lemma to the word $w = 0^p 1 0^p$, where p is the pumping length. w is clearly a palindrome, so it must be in L_b. But the pumping lemma states that $w = rst$, where $rs^n t$ is also in L_b, and $|rs| \leq p$. Therefore the pumping string s must contain only 0s, and the only 1 in w must be in t. Therefore, for any $n > 1$, $rs^n t$ will contain more 0s before the 1 than after it, so it cannot be a palindrome (or in L_b). This contradicts the pumping lemma. Therefore, L_b is not regular.

Lab Problem 2

Sometimes you see people who have the whole "rock star look" – leather jacket, dark sunglasses, metal belt, wild hair. Then you find out that they've never played a single bass groove, and wouldn't know a snare drum from a hi-hat. They're not rockers at all, they just happen to dress that way. Well, the same thing happens with languages.

Consider the language $F = \{a^i b^j c^k : i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$.

a. *Show that F is not regular.*

b. *Show that F looks like a regular language when the pumping lemma is applied. In other words, give a pumping length p and demonstrate that, for all strings $w \in F$, we can write $w = xyz$ with $|xy| \leq p$ and $|y| \geq 1$ such that $xy^iz \in F$ for all $i \geq 0$.*

c. *Explain why parts (a) and (b) do not contradict the pumping lemma.*

a. F is not regular, because the nonregular language $\{ab^nc^n | n \geq 0\}$ is the same as $F \cap ab^*c^*$, and the regular languages are closed under intersection.

   To prove that $L = \{ab^nc^n\}$ is not regular, we use the pumping lemma: Let p be the pumping length; consider $ab^pc^p \in L$. If L were regular, we could break it up into xyz such that $|y| > 0$, $|xy| \leq p$, and $xy^iz \in L \forall i \geq 0$.

   There are three cases: If $x = \varepsilon$, then y contains an a, so pumping y creates more than one a, so $xyyz \notin L$. If $x = a$, then y consists of all b's, and so pumping it creates unequal number of b's and c's, so $xyyz \notin L$. Finally, if $x = ab^*$, then y again consists of all b's, and so pumping it creates unequal numbers of b's and c's, so $xyyz \notin L$. Thus, ab^pc^p cannot be pumped so L is not regular.

b. Language F satisfies the conditions of the pumping lemma using pumping length 2. If $s \in F$ is of length 2 or more we show that it can be pumped by considering four cases, depending on the number of a's that s contains.

   (a) If s is of the form b^*c^*, let x = ϵ, y be the first symbol of s, and let z be the rest of s.

(b) If s is of the form ab^*c^*, let x = ϵ, y be the first symbol of s, and let z be the rest of s.

(c) If s is of the form aab^*c^*, let x= ϵ, y be the first two symbols of s, and let z be the rest of s.

(d) If s is of the form $aaaa^*b^*c^*$, let x= ϵ, y be the first symbol of s, and let z be the rest of s.

In each case, the strings xy^iz are members of F for every $i \geq 0$. Hence F satisfies the conditions fo the pumping lemma.

c. The pumping lemma is not violated because it states only that regular languages satisfy the three conditions and it doesn't state that nonregular languages fail to satisfy the three conditions.

Lab Problem 3

Let L_A and L_B be regular languages over the same alphabet Σ. Prove or disprove that

$$L_{AB} = \{a_1b_1a_2b_2\ldots a_ib_i \mid a = a_1a_2\ldots a_i \in L_A,\, b = b_1b_2\ldots b_i \in L_B\}$$

is a regular language.

Note: *It may be the case that there do not exist $a \in L_A$ and $b \in L_B$ such that $|a| = |b|$. In this case, $L_{AB} = \emptyset$.*

Let A and B be machines accepting L_A and L_B respectively. Let Q_A and Q_B be the states of A and B and F_A and F_B be the accept states of A and B. Create a machine with states $Q = Q_A \times Q_B \times \{\text{“}A\text{”}, \text{“}B\text{”}\}$, accept states $F = F_A \times F_B \times \{\text{“}A\text{”}\}$, start state $q_0 = (q_{0a}, q_{0b}, \text{“}A\text{”})$ and transition function δ such that

$$\delta((q_a, q_b, \text{“}A\text{”}), \Sigma) = (\delta_A(q_a, \Sigma), q_b, \text{“}B\text{”})$$
$$\delta((q_a, q_b, \text{“}B\text{”}), \Sigma) = (q_a, \delta_B(q_b, \Sigma), \text{“}A\text{”}).$$

This machine accepts the interleaved language, and hence it is regular.