CSE 105 Homework 2

Due January 29

- You may (and are encouraged to) discuss the problems with other students, but what you hand in must be your own work.
- If you do not understand a problem or find a mistake, please email William at wmatthews@cs.ucsd.edu.
- In addition to these problems, we recommend doing Sipser problems 1.4, 1.5, 1.7, 1.11, 1.16, 1.31, 1.32, 1.40 for more practice. However, you should not hand in your solutions to these problems.
- One of the main ways that languages are proved to be not regular is by using the pumping lemma. You are strongly encouraged to wait to do such problems until after the lectures on the pumping lemma.
- Typed homework solutions will receive 5% extra credit.

1 Regular Expressions

For this problem you will need the file data.txt, which contains an excerpt of the CSE class schedule from Spring 2008. The file can be found on the class webpage.

The UNIX program egrep uses regular expressions for text processing. Given a file and a regular expression, egrep will output every line of the file that matches the regular expression. You should read the man page for egrep (http://compute.cnr.berkeley.edu/cgi-bin/man-cgi?egrep) to become familiar with its notation, but the important things are that . matches any alphabet symbol, operator * is as expected, the union operation is |, and parentheses are used for grouping. Lastly, egrep matches the regular expression anywhere in the line, so it essentially places .* before and after whatever expression you give it.

So, using our class schedule data, if we wanted to find all lectures and discussion sections, we would type egrep '^(LE|DI)$' data.txt and we would get 86 lines of output. If we just wanted to know the number of lines that match, we could use the -c option (add -c after egrep but before the quoted regular expression).

For each of the questions below, you should include in your write-up the egrep regular expression you used as well as the number of lines outputted. Do not include the actual lines in your write-up! Also, you should only use union, concatenation, and star operations. You do not need any additional justification of the correctness of your regular expressions.

a) How many lectures were cancelled?

b) How many discussion sections are on either Monday or Tuesday?

c) How many lectures begin at exactly 2 PM?

d) How many finals are in Warren Lecture Hall?

e) How many classes are for a variable number of units?
1.1 Solution

a) `egrep -c 'LE.*Cancelled' data.txt` returns 6.

b) `egrep -c 'DI.*(M|Th)' data.txt` returns 10.

c) `egrep -c 'LE.*2:00p - (2|3)' data.txt` returns 9.

d) `egrep -c 'FI.*WLH' data.txt` returns 9.

e) `egrep -c '\(.*(-|/).* units\)' data.txt` returns 17.

2 More Regular Expressions

For each of the following NFAs or DFAs, give an equivalent regular expression. In each case Σ = {0, 1}. You do not need to justify your regular expressions.

![Image of NFAs or DFAs]

2.1 Solution

a) Let $a = ((1(00)^*1)\cup(0(11)^*0))^*$. The full regular expression is $a(0a(1\cup0)\cup01a)^*$ or $((1(00)^*1)\cup(0(11)^*0))^*1\cup1((1(00)^*1)\cup(0(11)^*0))^*0(1(00)^*1)\cup01((1(00)^*1)\cup(0(11)^*0))^*$. The idea here is that the DFA accepts any string with an odd number 0's and an odd number of 1's.

b) $0^*110^*(110^*)^*$

c) $00 \cup 110 \cup 11100$

3 Non-Regular Languages

Let Σ = {x}. For each of the following languages, prove whether the language is regular or not:

$L_1 = \{x^{3k} \mid k \in \{0, 1, 2, 3, \ldots\}\}$

$L_2 = \{x^k \mid k \in \{0, 1, 2, 3, \ldots\}\}$

$L_3 = \{x^k \mid k \in \{0, 1, 2, 3, \ldots\}\}$

3.1 Solution

a) $L_1$ is equivalent to the regular expression $(xxx)^*$ and therefore is regular. If the $xxx$ is repeated $k$ times, you get $x^{3k}$, and the regular expression give this for all possible values of $k$. 
b) $L_2$ is not regular. We will prove this by contradiction: Assume that $L_2$ is regular, then by the Pumping Lemma for regular languages it must have some pumping length $p'$. Let $p$ be the larger of $p'$ and 2. Consider the string $s = x^{p'}$. By the Pumping Lemma, this can be broken up into parts $X = x^a$, $Y = x^b$, and $Z = x^{p'-a-b}$ where $b > 0$ and $a + b \leq p' \leq p$. Again by the Pumping Lemma, $XYYZ = x^{a+2b+(p'-a-b)}$ must be in $L_2$. However, $p^3 < a + 2b + (p^3 - a - b) = p^3 + b \leq p^3 + p < (p+1)^3 = p^3 + 3p^2 + 3p + 1$. Thus $XYYZ$ is not in $L_2$ so we get a contradiction and conclude that $L_2$ cannot be regular.

c) $L_3$ is not regular. We will prove this by contradiction: Assume that $L_3$ is regular, then by the Pumping Lemma for regular languages it must have some pumping length $p'$. Let $p$ be the larger of $p'$ and 3. Consider the string $s = x^{p'}$. By the Pumping Lemma, this can be broken up into parts $X = x^a$, $Y = x^b$, and $Z = x^{p'-a-b}$ where $b > 0$ and $a + b \leq p' \leq p$. Again by the Pumping Lemma, $XYYZ = x^{a+2b+(p'-a-b)}$ must be in $L_3$. However, $p! \leq a + 2b + (p! - a - b) = p! + b \leq p! + p < 2p! < (p+1)p! = (p + 1)!$. Thus $XYYZ$ is not in $L_3$ so we get a contradiction and conclude that $L_3$ cannot be regular.

4 More Non-Regular Languages

Let $\Sigma = \{1, 0\}$. Consider the following two languages:

$L_1 = \{a_1a_2\ldots a_kb_1b_2\ldots b_k \mid a_i, b_i \in \Sigma \text{ and when viewed as binary numbers, } |a_1a_2\ldots a_kb_1b_2\ldots b_k| = 1\}$

$L_2 = \{a_1a_2\ldots a_kb_kb_{k-1}\ldots b_1 \mid a_i, b_i \in \Sigma \text{ and when viewed as binary numbers, } |a_1a_2\ldots a_kb_kb_{k-1}\ldots b_1| = 1\}$

For example, $10000111 \in L_1$ since $|10001011| = 1$, but $10001010 \notin L_1$ since $|10001010| = 2$. Similarly, $10001110 \in L_2$ since $|10001110| = 1$, but $10001011 \notin L_2$ since $|10001011| = 2$.

Prove that neither $L_1$ nor $L_2$ are regular.

4.1 Solution

We will use an identical proof for both $L_1$ and $L_2$. Let $L$ refer to either one of $L_1$ or $L_2$.

We will prove that $L$ is not regular by contradiction. Assume that $L$ is regular. Then by the Pumping Lemma for Regular Languages, $L$ has a pumping length $p$. Consider the string $s = 0^p1^p0^p \in L$. $s$ can be split up into 3 parts as follows: $x = 0^a$, $y = 0^b$, and $z = 0^{p-a-b}1^p0^p$, where $b > 0$ and $a + b \leq p$. By the Pumping Lemma, $xy = 0^a0^{p-a-b}1^p0^p$ must be in $L$. If $b$ is odd, then the length of the string $0^a0^{p-a-b}1^p0^p$ is odd so it cannot be in $L$. For the remainder of the argument we will assume that $b$ is even and $b \geq 2$. The total length of the string when we remove the $y$ part is $2p + 2 - b$, so when we split this string up into the $a$ half and the $b$ half, each side has length $p + 1 - b/2 \leq p$. The last $p + 1$ symbols of $s$ are all 0, and none of these symbols can be in $y$. Thus, the last $p + 1$ symbols of $xz$ are also all 0. Therefore, $a_{p+1-b/2} = 0$ so the value of the $a$ half of the string must be at least 2. The value of the $b$ half of the string is 0 since every symbol is 0. Thus, the difference is greater than 1, so the string $xz$ is not in $L$. This is a contradiction to the Pumping Lemma, so $L$ cannot be regular.

5 Properties of Languages

Prove or disprove each of the following claims:

a) For any languages $L$ and $M$, if $L \subseteq M$ and $L$ is not regular then $M$ is not regular.

b) For any languages $A$ and $B$, if $A \subseteq B$ and $B$ is not regular then $A$ is not regular.

c) For any language $C$, if $C$ is not regular then $C \cup \{\epsilon\}$ is not regular.
5.1 Solution

a) Let $L$ be an arbitrary non regular language over an alphabet $\Sigma$. Let $M = \Sigma^*$. $L \subseteq M$, yet $M$ is regular since it was defined by a regular expression.

b) Let $B$ be an arbitrary non regular language. Let $A = \emptyset$. $A \subseteq B$, yet $A$ is regular since it was defined by a regular expression.

c) We will prove this statement by proving the contrapositive. Namely, if $C \cup \{\epsilon\}$ is regular then $C$ must be regular: If $\epsilon \in C$ then $C \cup \{\epsilon\} = C$ so $C$ is regular. Otherwise, $C = (C \cup \{\epsilon\}) \cap \{\overline{\epsilon}\}$. Regular languages are closed under complement and intersection and we are assuming that $(C \cup \{\epsilon\})$ is regular, so $C$ must be regular.