**COMP2270/6270 – Theory of Computation**
**Sixth week**
**School of Electrical Engineering & Computing**
**The University of Newcastle**

**Exercise 1)** Define the function *twice*($L$) = $\{w : \exists x \in L$ ($x$ can be written as $c_1c_2 \ldots c_n$, for some $n \geq 1$, where each $c_i \in \Sigma_L$, and $w = c_1c_1c_2c_2 \ldots c_nc_n)\}$.

a) Let $L = (1 \cup 0)^*1$. Write a regular expression for *twice*($L$).

$(11 \cup 00)^*11$.

b) Are the regular languages closed under *twice*? Prove your answer.

Yes, by construction. If $L$ is regular, then there is some DFSM $M$ that accepts it. We build an FSM $M'$ that accepts *twice*($L$):

1. Initially, let $M'$ be $M$.
2. Modify $M'$ as follows: For every transition in $M$ from some state $p$ to some state $q$ with label $c$, do:
   - 2.1. Remove the transition from $M'$.
   - 2.2. Create a new state $p'$.
   - 2.3. Add to $M'$ two transitions: $((p, c), p')$ and $((p', c), q)$.
3. Make the start state of $M'$ be the same as the start state of $M$.
4. Make every accepting state in $M$ also be an accepting state in $M'$.

**Exercise 2)** For each of the following claims, state whether it is *True* or *False*. Prove your answer.:

a) The union of an infinite number of regular languages must be regular.

False. Let $L = \cup$ ({ε}, {ab}, {aabb}, {aaabbb}, …) Each of these languages is finite and thus regular. But the infinite union of them is $\{a^nb^n, n \geq 0\}$, which is not regular.

b) The union of an infinite number of regular languages is never regular.

Nothing says the languages that are being unioned have to be different. So, Let $L = \cup$ (a*, a*, a*, …), which is a*, which is regular.

c) If $L_1$ and $L_2$ are regular languages and $L_1 \subseteq L \subseteq L_2$, then $L$ must be regular.

False. Let $L_1 = \varnothing$. Let $L_2 = \{a \cup b\}^*$. Let $L = \{a^nb^n : n \geq 0\}$, which is not regular.

d) The intersection of two nonregular languages must not be regular.

False. Let $L_1 = \{a^p : p \text{ is prime}\}$, which is not regular. Let $L_2 = \{b^p : p \text{ is prime}\}$, which is also not regular. $L_1 \cap L_2 = \varnothing$, which is regular.

e) The intersection of an infinite number of regular languages must be regular.

False. Let $x_1, x_2, x_3, \ldots$be the sequence 0, 1, 4, 6, 8, 9, … of nonprime, nonnegative integers. Let $a^{x_i}$ be a string of $x_i$ a's. Let $L_i$ be the language a* - $\{a^{x_i}\}$.

Now consider $L$ = the infinite intersection of the sequence of languages $L_1, L_2, \ldots$ Note that $L = \{a^p$, where $p$ is prime$\}$. We have proved that $L$ is not regular.

f) If $L$ is a language that is not regular, then $L^*$ is not regular.

False.
Let $L$ = Prime$_a$ = {$a^n$ : $n$ is prime}. $L$ is not regular.
$L^* = \{\varepsilon\} \cup \{a^n : 1 < n\}$. $L^*$ is regular.

g) If $L^*$ is regular, then $L$ is regular.

False.
Let $L$ = Prime$_a$ = {$a^n$ : $n$ is prime}. $L$ is not regular.
$L^* = \{\varepsilon\} \cup \{a^n : 1 < n\}$. $L^*$ is regular.

h) Every subset of a regular language is regular.

False.
Let $L$ = $a^*$, which is regular.
Let $L' = a^p$, where $p$ is prime. $L'$ is not regular, but it is a subset of $L$.

**Exercise 3)** (Chapter 8, Exercise 2 of Ref[1]) For each of the following languages $L$, state whether $L$ is regular or not and prove your answer:

a) $\{w \in \{a, b, c\}^* :$ in each prefix $x$ of $w$, $\#_a(x) = \#_b(x) = \#_c(x))\}$.

Regular. $L = \{\varepsilon\}$.

b) $\{w \in \{a, b, c\}^* : \exists$ some prefix $x$ of $w$ $(\#_a(x) = \#_b(x) = \#_c(x))\}$.

Regular. $L = \Sigma^*$, since every string has $\varepsilon$ as a prefix.

c) $\{w \in \{a, b, c\}^* : \exists$ some prefix $x$ of $w$ $(x \neq \varepsilon$ and $\#_a(x) = \#_b(x) = \#_c(x))\}$.

Not regular, which we prove by pumping. Let $w = a^k b^k c^k$.

**Exercise 4)** (Chapter 8, Exercise 2 of Ref[1]) Define the following two languages:

$L_a = \{w \in \{a, b\}^* :$ in each prefix $x$ of $w$, $\#_a(x) \geq \#_b(x)\}$.
$L_b = \{w \in \{a, b\}^* :$ in each prefix $x$ of $w$, $\#_b(x) \geq \#_a(x)\}$.

a) Let $L_1 = L_a \cap L_b$. Is $L_1$ regular? Prove your answer.

Regular. $L_1 = \{\varepsilon\}$.

b) Let $L_2 = L_a \cup L_b$. Is $L_2$ regular? Prove your answer.

Not regular. First, we observe that $L_2 = \{\varepsilon\} \cup \{w$: the first character of $w$ is an a and $w \in L_a\}$
$\cup \{w$: the first character of $w$ is a b and $w \in L_b\}$

We can show that $L_2$ is not regular by pumping. Let $w = a^{2k}b^{2k}$. $y$ must be $a^p$ for some $0 < p \leq k$. Pump out. The resulting string $w' = a^{2k-p}b^{2k}$. Note that $w'$ is a prefix of itself. But it is not in $L_2$ because it is not ε, nor is it in $L_a$ (because it has more b's than a's) or in $L_b$ (because it has starts with a).

**Exercise 5)** (Chapter 8, Exercise 4 of Ref[1]) For each of the following languages $L$, state whether $L$ is regular or not and prove your answer:

a) $\{uww^Rv : u, v, w \in \{\mathtt{a}, \mathtt{b}\}^+\}$.

Regular. Every string in $L$ has at least 4 characters. Let $w$ have length 1. Then $ww^R$ is simply two identical characters next to each other. So $L$ consists of exactly those strings of at least four characters such that there's a repeated character that is not either the first or last. Any such string can be rewritten as $u$ (all the characters up to the first repeated character) $w$ (the first repeated character) $w^R$ (the second repeated character) $v$ (all the rest of the characters). So $L = (\mathtt{a} \cup \mathtt{b})^+ (\mathtt{aa} \cup \mathtt{bb}) (\mathtt{a} \cup \mathtt{b})^+$.

b) $\{xyzy^Rx : x, y, z \in \{\mathtt{a}, \mathtt{b}\}^+\}$.

Not regular, which we show by pumping. Let $w = \mathtt{a}^k\mathtt{babaa}^k\mathtt{b}$. Note that $w$ is in $L$ because, using the letters from the language definition, $x = \mathtt{a}^k\mathtt{b}$, $y = \mathtt{a}$, and $z = \mathtt{b}$. Then $y$ (from the Pumping Theorem) must occur in the first a region. It is $\mathtt{a}^p$ for some nonzero $p$. Set $q$ to 2 (i.e., pump in once). The resulting string is $\mathtt{a}^{k+p}\mathtt{babaa}^k\mathtt{b}$. This string cannot be in $L$. Since its initial $x$ (from the language definition) region starts with a, there must be a final $x$ region that starts with a. Since the final $x$ region ends with a b, the initial $x$ region must also end with a b. So, thinking about the beginning of the string, the shortest $x$ region is $\mathtt{a}^{k+p}\mathtt{b}$. But there is no such region at the end of the string unless $p$ is 1. But even in that case, we can't call the final $\mathtt{aa}^k\mathtt{b}$ string $x$ because that would leave only the middle substring ab to be carved up into $yzy^R$. But since both $y$ and $z$ must be nonempty, $yzy^R$ must have at least three characters. So the resulting string cannot be carved up into $xyzy^Rx$ and so is not in $L$.

**Exercise 6)** (Chapter 9, Exercise 1 of Ref[1]) Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. For the languages that are defined by each of the following grammars, do each of the following:

*i*. List five strings that are in $L$.

*ii*. List five strings that are not in $L$.

*iii*. Describe $L$ concisely. You can use regular expressions, expressions using variables (e.g., $\mathtt{a}^n\mathtt{b}^n$, or set theoretic expressions (e.g., $\{x: \ldots\}$)

*iv*. Indicate whether or not $L$ is regular. Prove your answer.

a) $S \rightarrow \mathtt{a}S \mid S\mathtt{b} \mid \varepsilon$

*i*. ε, a, b, aaabbbb, ab

*ii*. ba, bbaa, bbbbba, ababab, aba

*iii*. $L = \mathtt{a}^*\mathtt{b}^*$.

*iv*. $L$ is regular because we can write a regular expression for it.

b) $S \rightarrow \mathtt{a}S\mathtt{a} \mid \mathtt{b}S\mathtt{b} \mid \mathtt{a} \mid \mathtt{b}$

*i*. a, b, aaa, bbabb, aaaabaaaa

*ii*. ε, ab, bbbbbbba, bb, bbbaaa

*iii*. $L$ is the set of odd length palindromes, i.e., $L = \{w = x (\mathtt{a} \cup \mathtt{b}) x^R$, where $x \in \{\mathtt{a,b}\}^*\}$.

*iv*. $L$ is not regular. Easy to prove with pumping. Let $w = \mathtt{a}^k\mathtt{baba}^k$. $y$ must be in the initial a region. Pump in and there will no longer be a palindrome.

**Exercise 7)** (Chapter 9, Exercise 4 of Ref[1]) Consider the following context free grammar $G$:

$S \rightarrow \mathrm{a}S\mathrm{a}$
$S \rightarrow T$
$S \rightarrow \varepsilon$
$T \rightarrow \mathrm{b}T$
$T \rightarrow \mathrm{c}T$
$T \rightarrow \varepsilon$

One of these rules is redundant and could be removed without altering $L(G)$. Which one?

$S \rightarrow \varepsilon$

**Exercise 8)** (Chapter 9, Exercise 6 of Ref[1]) Show a context-free grammar for each of the following languages $L$:

a) BalDelim = {$w$ : where $w$ is a string of delimeters: (, ), [, ], {, }, that are properly balanced}.

$S \rightarrow (\ S\ )\ |\ [\ S\ ]\ |\ \{\ S\ \}\ |\ S\ S\ |\ \varepsilon$

b) $\{\mathrm{a}^i\mathrm{b}^j : 2i \neq 3j + 1\}$.

We can begin by analyzing $L$, as shown in the following table:

| # of a's | Allowed # of b's |
|---|---|
| 0 | any |
| 1 | any |
| 2 | any except 1 |
| 3 | any |
| 4 | any |
| 5 | any except 3 |
| 6 | any |
| 7 | any |
| 8 | any except 5 |

$S \rightarrow \mathrm{aaa}S\mathrm{bb}$
$S \rightarrow \mathrm{aaa}X$ /* extra a's
$S \rightarrow T$ /* terminate
$X \rightarrow A\ |\ A\ \mathrm{b}$ /* arbitrarily more a's
$T \rightarrow A\ |\ B\ |\ \mathrm{a}\ B\ |\ \mathrm{aabb}\ B$ /* note that if we add two more a's we cannot add just a single b.
$A \rightarrow \mathrm{a}\ A\ |\ \varepsilon$
$B \rightarrow \mathrm{b}\ B\ |\ \varepsilon$

c) $\{w \in \{\mathrm{a}, \mathrm{b}\}^* : \#_{\mathrm{a}}(w) = 2\ \#_{\mathrm{b}}(w)\}$.

$S \rightarrow S\mathrm{a}S\mathrm{a}S\mathrm{b}S$
$S \rightarrow S\mathrm{a}S\mathrm{b}S\mathrm{a}S$
$S \rightarrow S\mathrm{b}S\mathrm{a}S\mathrm{a}S$
$S \rightarrow \varepsilon$

## Extra from book:

1) (Chapter 8, Exercise 1 of Ref[1]) For each of the following languages $L$, state whether or not $L$ is regular. Prove your answer:

a) $\{\mathtt{a}^i\mathtt{b}^j : 0 \le i < j < 2000\}$.

Regular. Finite.

b) $\{w \in \{\mathtt{Y}, \mathtt{N}\}^* : w$ contains at least two Y's and at most two N's$\}$.

Regular. $L$ can be accepted by an FSM that keeps track of the count (up to 2) of Y's and N's.

c) $\{w = xy : x, y \in \{\mathtt{a}, \mathtt{b}\}^*$ and $|x| = |y|$ and $\#_{\mathtt{a}}(x) \ge \#_{\mathtt{a}}(y)\}$.

Not regular, which we'll show by pumping. Let $w = \mathtt{a}^k\mathtt{bba}^k$. $y$ must occur in the first a region and be equal to $\mathtt{a}^p$ for some nonzero $p$. Let $q = 0$. If $p$ is odd, then the resulting string is not in $L$ because all strings in $L$ have even length. If $p$ is even it is at least 2. So both b's are now in the first half of the string. That means that the number of a's in the second half is greater than the number in the first half. So resulting string, $\mathtt{a}^{k-p}\mathtt{bba}^k$, is not in $L$.

d) $\{w = xyzy^{\mathrm{R}}x : x, y, z \in \{\mathtt{a}, \mathtt{b}\}^*\}$.

Regular. Note that $L = (\mathtt{a} \cup \mathtt{b})^*$. Why? Take any string $s$ in $(\mathtt{a} \cup \mathtt{b})^*$. Let $x$ and $y$ be $\varepsilon$. Then $s = z$. So the string can be written in the required form. Moral: Don't jump too fast when you see the nonregular "triggers", like $ww$ or $ww^{\mathrm{R}}$. The entire context matters.

e) $\{w = xyzy : x, y, z \in \{0, 1\}^+\}$.

Regular. The key to why this is so is to observe that we can let $y$ be just a single character. Then the rest of $w$ can generated by $x$ and $z$. So any string $w$ in $\{0, 1\}^+$ is in $L$ iff:

- the last letter of $w$ occurs in at least one other place in the string,
- that place is not the next to the last character,
- nor is it the first character, and
- $w$ contains least 4 letters.

Either the last character is 0 or 1. So:

$$L = ((0 \cup 1)^+\ 0\ (0 \cup 1)^+\ 0) \cup ((0 \cup 1)^+\ 1\ (0 \cup 1)^+\ 1).$$

f) $\{w \in \{0, 1\}^* : \#_0(w) \ne \#_1(w)\}$.

Not regular. This one is quite hard to prove by pumping. Since so many strings are in $L$, it's hard to show how to pump and get a string that is guaranteed not to be in $L$. Generally, with problems like this, you want to turn them into problems involving more restrictive languages to which it is easier to apply pumping. So: if $L$ were regular, then the complement of $L$, $L'$ would also be regular.

$$L' = \{w \in \{0, 1\}^* : \#_0(w) = \#_1(w)\}.$$

It is easy to show, using pumping, that $L'$ is not regular: Let $w = 0^k1^k$. $y$ must occur in the initial string of 0's, since $|xy| \leq k$. So $y = 0^i$ for some $i \geq 1$. Let $q$ of the pumping theorem equal 2 (i.e., we will pump in one extra copy of $y$). We now have a string that has more 0's than 1's and is thus not in $L'$. Thus $L'$ is not regular. So neither is $L$. Another way to prove that $L'$ isn't regular is to observe that, if it were, $L'' = L' \cap 0^*1^*$ would also have to be regular. But $L''$ is $0^n1^n$, which we already know is not regular.

g) $\{w \in \{a, b\}^* : w = w^R\}$.

Not regular, which we show by pumping. Let $w = a^kb^kb^ka^k$. So $y$ must be $a^p$ for some nonzero $p$. Pump in once. Reading $w$ forward there are more a's before any b's than there are when $w$ is read in reverse. So the resulting string is not in $L$.

h) $\{w \in \{a, b\}^* : \exists x \in \{a, b\}^+ (w = x\, x^R\, x)\}$.

Not regular, which we show by pumping: Let $w = a^kbba^ka^kb$. $y$ must occur in the initial string of a's, since $|xy| \leq k$. So $y = a^i$ for some $i \geq 1$. Let $q$ of the pumping theorem equal 2 (i.e., we will pump in one extra copy of $y$). That generates the string $a^{k+i}bba^ka^kb$. If this string is in $L$, then we must be able to divide it into thirds so that it is of the form $x\, x^R\, x$. Since its total length is $3k + 3 + i$, one third of that (which must be the length of $x$) is $k + 1 + i/3$. If $i$ is not a multiple of 3, then we cannot carve it up into three equal parts. If $i$ is a multiple of 3, we can carve it up. But then the right boundary of $x$ will shift two characters to the left for every three a's in $y$. So, if $i$ is just 3, the boundary will shift so that $x$ no longer contains any b's If $i$ is more than 3, the boundary will shift even farther away from the first b. But there are b's in the string. Thus the resulting string cannot be in $L$. Thus $L$ is not regular.

i) $\{w \in \{a, b\}^*$ : the number of occurrences of the substring ab equals the number of occurrences of the substring ba$\}$.

Regular. The idea is that it's never possible for the two counts to be off by more than 1. For example, as soon as there's an ab, there can be nothing but b's without producing the first ba. Then the two counts are equal and will stay equal until the next b. Then they're off by 1 until the next a, when they're equal again. $L = a^* \cup a^+b^+a^+(b^+a^+)^* \cup b^* \cup b^+a^+b^+(a^+b^+)^*$.

j) $\{w \in \{a, b\}^* : w$ contains exactly two more b's than a's$\}$.

Not regular, which we'll show by pumping. Let $w = a^kb^{k+2}$. $y$ must equal $a^p$ for some $p > 0$. Set $q$ to 0 (i.e., pump out once). The number of a's changes, but the number of b's does not. So there are no longer exactly 2 more b's than a's.

k) $\{w \in \{a, b\}^* : w = xyz, |x| = |y| = |z|$, and $z = x$ with every a replaced by b and every b replaced by a$\}$. Example: abbbabbaa $\in L$, with $x$ = abb, $y$ = bab, and $z$ = baa.

Not regular, which we'll show by pumping. Let $w = a^ka^kb^k$. This string is in $L$ since $x = a^k$, $y = a^k$, and $z = b^k$. $y$ (from the pumping theorem) $= a^p$ for some nonzero $p$. Let $q = 2$ (i.e., we pump in once). If $p$ is not divisible by 3, then the resulting string is not in $L$ because it cannot be divided into three equal length segments. If $p = 3i$ for integer $i$, then, when we divide the resulting string into three segments of equal length, each segment gets longer by $i$ characters. The first segment is still all a's, so the last segment must remain all b's. But it doesn't. It grows by absorbing a's from the second segment. Thus $z$ no longer = $x$ with every a replaced by b and every b replaced by a. So the resulting string is not in $L$.

l) $\{w: w \in \{$a – z$\}^*$ and the letters of $w$ appear in reverse alphabetical order$\}$. For example, spoonfeed $\in L$.

Regular. $L$ can be recognized by a straightforward 26-state FSM.

m) $\{w: w \in \{$a - z$\}^*$ every letter in $w$ appears at least twice$\}$. For example, unprosperousness $\in L$.

Regular. $L$ can be recognized by an FSM with $26^3$ states. The states count the occurrences of each letter. For each of the 26 letters, there are three values (0, 1, at least 2).

n) $\{w : w$ is the decimal encoding of a natural number in which the digits appear in a non-decreasing order without leading zeros$\}$.

Regular. $L$ can be recognized by an FSM with 10 states that checks that the digits appear in the correct order. Or it can be described by the regular expression: 0*1*2*3*4*5*6*7*8*9*.

o) $\{w$ of the form: <$integer_1$>+<$integer_2$>=<$integer_3$>, where each of the substrings <$integer_1$>, <$integer_2$>, and <$integer_3$> is an element of $\{$0 - 9$\}^*$ and $integer_3$ is the sum of $integer_1$ and $integer_2\}$. For example, 124+5=129 $\in L$.

Not regular, which we can prove by pumping. Let $w$ = $1^k+2^k=3^k$. $y$ must be $1^p$ for some nonzero $p$. Pump in once. The resulting arithmetic statement is false.

p) $L_0^*$, where $L_0 = \{$b$a^i b^j a^k$, $j \geq 0$, $0 \leq i \leq k\}$.

Regular. Both $i$ and $j$ can be 0. So $L = (b^+a^*)^*$.

q) $\{w : w$ is the encoding (in the scheme we describe next) of a date that occurs in a year that is a prime number$\}$. A date will be encoded as a string of the form *mm*/*dd*/*yyyy*, where each *m*, *d*, and *y* is drawn from {0-9}.

Regular. Finite, since there is only a finite number of values for each of *mm*, *dd*, and *yyyy*.

r) $\{w \in \{1\}^* : w$ is, for some $n \geq 1$, the unary encoding of $10^n\}$. (So $L = \{1111111111, 1^{100}, 1^{1000}, \ldots\}$.)

Not regular, which we can prove by pumping. Let $w = 1^t$, where $t$ is the smallest integer that is a power of ten and is greater than $k$. $y$ must be $1^p$ for some nonzero $p$. Clearly, $p$ can be at most $t$. Let $q = 2$ (i.e., pump in once). The length of the resulting string $s$ is at most $2t$. But the next power of 10 is $10t$. Thus $s$ cannot be in $L$.

REFERENCES

[1] Elaine Rich, Automata Computatibility and Complexity: Theory and Applications, Pearson, Prentice Hall, 2008.