# COMP 3803 — Solutions Assignment 2

**Question 1:** Write your name and student number.

**Solution:** Cristiano Ronaldo, CR7

**Question 2:**

- Consider the language $A$ consisting of all binary strings that end with an even, and non-zero, number of 0s. Give a regular expression that describes the language $A$. As always, justify your answer.
- What is the language described by the following regular expression:

$$(0 \cup 1)^*(00)^*00.$$

  As always, justify your answer.

**Solution:** We start with the first part. Each string in the language is obtained as follows:

- Write the empty string or an arbitrary string that ends with 1, i.e., $\epsilon \cup (0 \cup 1)^*1$.
- Write an even (and non-zero) number of 0's, i.e., $(00)^*00$.

This gives the regular expression

$$(\epsilon \cup (0 \cup 1)^*1)\,(00)^*00.$$

For the second part, consider the regular expression

$$(0 \cup 1)^*(00)^*00.$$

If a string is in the language described by this regular expression, then the string ends with 00. The claim is that every string that ends with 00 is in the language described by this regular expression. To prove this, let $w$ be an arbitrary bitstring that ends with 00. We can write $w = v\epsilon 00$, where $v$ is a (possibly empty) bitstring.

- The string $v$ is in the language described by $(0 \cup 1)^*$.
- The string $\epsilon$ is in the language described by $(00)^*$.
- The string 00 is in the language described by 00.
- Thus, the string $w = v\epsilon 00$ is in the language described by

$$(0 \cup 1)^*(00)^*00.$$

**Question 3:** Give regular expressions describing the following two languages. In both cases, the alphabet is $\{a, b\}$. Justify your answers.

- $\{w :$ the number of $a$'s in $w$ is a multiple of three$\}$.
- $\{w : w$ does not contain $aaa\}$.

**Solution:** We start with

$$\{w : \text{ the number of } a\text{'s in } w \text{ is a multiple of three}\}.$$

Each string in this language is obtained by

- writing zero or more $b$'s, i.e., $b^*$
- or repeating the following zero or more times:
  - write a string having exactly three $a$'s, i.e., $b^*ab^*ab^*ab^*$.

This gives the regular expression

$$b^* \cup (b^*ab^*ab^*ab^*)^* .$$

Next, we do

$$\{w : w \text{ does not contain } aaa\}.$$

Each string in this language is obtained as follows:

- Start with zero or more $b$'s, i.e., $b^*$.
- Repeat zero or more times:
  - $a$ followed by at least one $b$ or $aa$ followed by at least one $b$, i.e., $abb^* \cup aabb^*$.
- End with $\epsilon$ or $a$ or $aa$, i.e., $\epsilon \cup a \cup aa$.

This gives the regular expression

$$b^* \left(abb^* \cup aabb^*\right)^* \left(\epsilon \cup a \cup aa\right).$$

If you want, you can add $b^*$ at the end of this regular expression; this will also give you a correct regular expression.

**Question 4:** Use the construction given in class to convert the regular expression

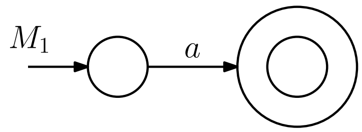
$$(a \cup b)^* aa(a \cup b)^*$$

to an NFA. Do not simplify your NFA; just apply the construction rules "without thinking".

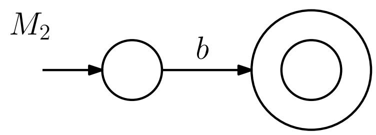
**Solution:** We first consider how the regular expression is "built":

- Take the regular expressions $a$ and $b$, and combine them into the regular expression $a \cup b$.
- Take the regular expression $a \cup b$, and turn it into the regular expression $(a \cup b)^*$.
- Take the regular expressions $a$ and $a$, and combine them into the regular expression $aa$.
- Take the regular expressions $(a \cup b)^*$ and $aa$, and combine them into the regular expression $(a \cup b)^* aa$.
- Take the regular expressions $(a \cup b)^* aa$ and $(a \cup b)^*$, and combine them into the regular expression $(a \cup b)^* aa(a \cup b)^*$.

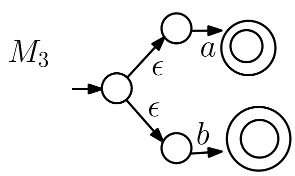
First, we construct an NFA $M_1$ that accepts the language described by the regular expression $a$:



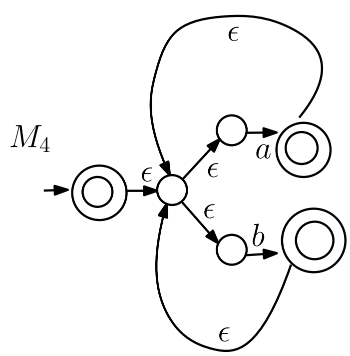

Next, we construct an NFA $M_2$ that accepts the language described by the regular expression $b$:



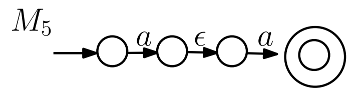

Next, we apply the union construction to $M_1$ and $M_2$. This gives an NFA $M_3$ that accepts the language described by the regular expression $a \cup b$:



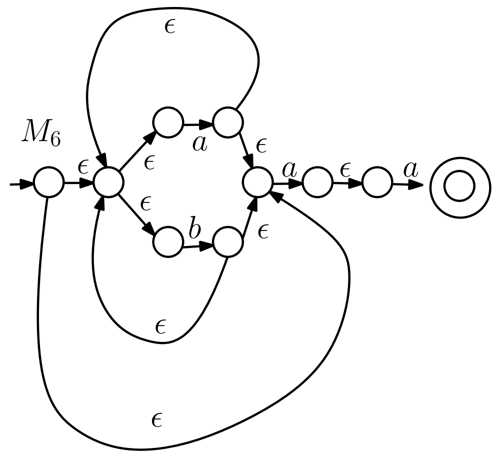

Next, we apply the star construction to $M_3$. This gives an NFA $M_4$ that accepts the language described by the regular expression $(a \cup b)^*$:



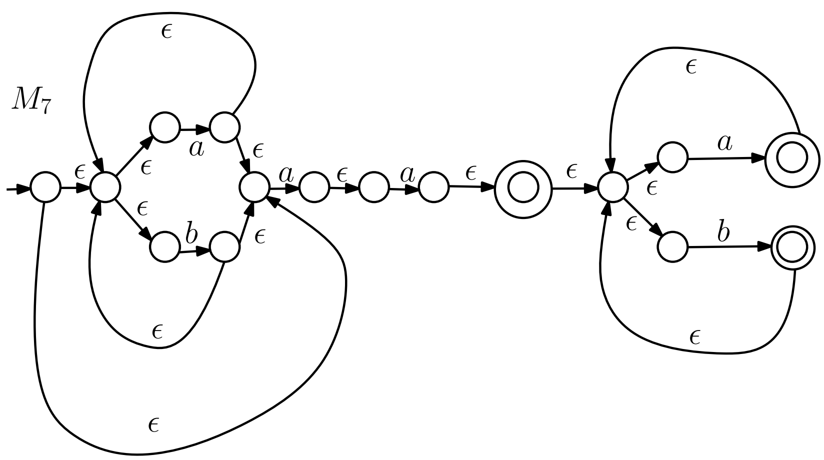
Next, we apply the concatenate construction to $M_1$ and $M_1$. This gives an NFA $M_5$ that accepts the language described by the regular expression $aa$:



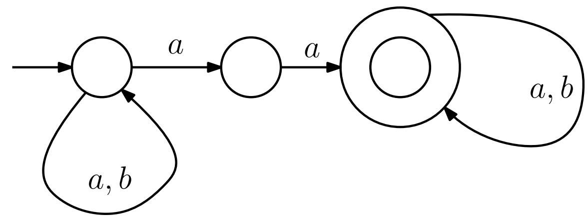

Next, we apply the concatenate construction to $M_4$ and $M_5$. This gives an NFA $M_6$ that accepts the language described by the regular expression $(a \cup b)^*aa$:



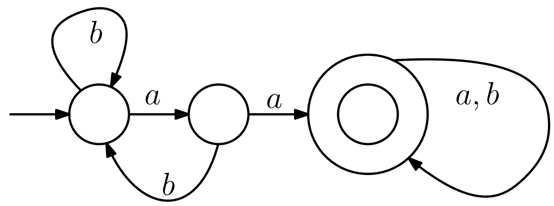

Finally, we apply the concatenate construction to $M_6$ and $M_4$. This gives an NFA $M_7$ that accepts the language described by the regular expression $(a \cup b)^*aa(a \cup b)^*$:



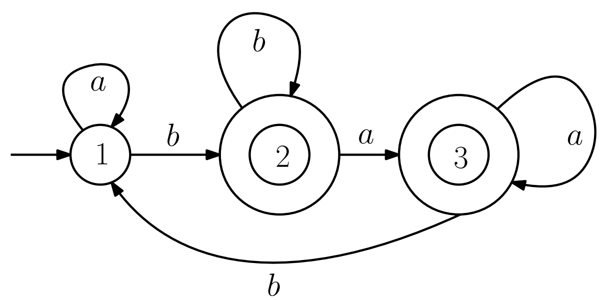

This was painful, eh! We just applied the algorithm without thinking and obtained the complicated NFA $M_7$. Of course, there is a much smaller NFA that accepts the language described by the regular expression $(a \cup b)^*aa(a \cup b)^*$:




There is even a DFA with three states that accepts this language:




**Question 5:** Use the construction given in class to convert the following DFA to a regular expression.




**Solution:** For each state $i = 1, 2, 3$, we define $L_i$ to be the set of all strings $w$ in $\{a, b\}^*$ such that the path in the state diagram that starts in state $i$ and corresponds to $w$ ends in one of the two accept states. We obtain the following three equations:

$$L_1 = aL_1 \cup bL_2 \quad (1)$$
$$L_2 = \epsilon \cup aL_3 \cup bL_2 \quad (2)$$
$$L_3 = \epsilon \cup aL_3 \cup bL_1 \quad (3)$$

Since 1 is the start state, we need a regular expression for $L_1$.

We use the following tool to solve these equations:

$$\text{If } L = BL \cup C \text{ and } \epsilon \notin B, \text{ then } L = B^*C. \quad (4)$$

We rewrite (3) as

$$L_3 = aL_3 \cup (\epsilon \cup bL_1),$$

which is of the form (4) with $L = L_3$, $B = a$, and $C = \epsilon \cup bL_1$. Since $\epsilon$ is not in the language described by $B$, we obtain

$$L_3 = a^* (\epsilon \cup bL_1) = a^* \cup a^* bL_1,$$

which we substitute into (2):

$$\begin{aligned} L_2 &= \epsilon \cup a (a^* \cup a^* bL_1) \cup bL_2 \\ &= bL_2 \cup (\epsilon \cup aa^* \cup aa^* bL_1). \end{aligned}$$

This is of the form (4) with $L = L_2$, $B = b$, and

$$C = \epsilon \cup aa^* \cup aa^* bL_1.$$

Since $\epsilon$ is not in the language described by $B$, we obtain

$$\begin{aligned} L_2 &= b^* (\epsilon \cup aa^* \cup aa^* bL_1) \\ &= b^* \cup b^* aa^* \cup b^* aa^* bL_1, \end{aligned}$$

which we substitute into (1):

$$\begin{aligned} L_1 &= aL_1 \cup b (b^* \cup b^* aa^* \cup b^* aa^* bL_1) \\ &= (a \cup bb^* aa^* b) L_1 \cup (bb^* \cup bb^* aa^*). \end{aligned}$$

This is of the form (4) with $L = L_1$,

$$B = a \cup bb^* aa^* b$$

and

$$C = bb^* \cup bb^* aa^*.$$

Since $\epsilon$ is not in the language described by $B$, we obtain

$$L_1 = (a \cup bb^* aa^* b)^* (bb^* \cup bb^* aa^*).$$

**Question 6:** Let $A$ be a regular language with alphabet $\{a, b\}$, and let

$$B = \{uv : u \in A, v \in \{a, b\}^*, |v| = 2\},$$

where $|v|$ denotes the length of the string $v$. Prove that $B$ is a regular language. Your proof must use the fact that a language is regular if and only if there exists a regular expression that describes the language.

**Solution:** Let

$$C = \{v : v \in \{a, b\}^*, |v| = 2\}.$$

Then $B = AC$, i.e., $B$ is the concatenation of $A$ and $C$.

Since $A$ is regular, there is a regular expression $R$ that describes $A$. The following regular expression

$$R' = aa \cup ab \cup ba \cup bb$$

describes the language $C$. Since $R$ and $R'$ are regular expressions, $RR'$ is also a regular expression, and it describes the language $AC$, which is equal to $B$. Thus, there is a regular expression that describes the language $B$. Therefore, $B$ is a regular language.

**Question 7:** Prove that the following languages are not regular.

1. $\{a^n b a^m b a^{n+m} : n \geq 0, m \geq 0\}$.
2. $\{w \in \{a,b\}^* : w \text{ is not a palindrome}\}$.

   Remark: A string $w = w_1 w_2 \cdots w_n$ is a palindrome, if $w_1 w_2 \cdots w_n = w_n \cdots w_2 w_1$. For example, each of $abba$, $\epsilon$, and $b$ is a palindrome.
3. $\{ucu : u \in \{a,b\}^*\}$. (The alphabet is $\{a,b,c\}$.)
4. $\{aba^2ba^3b \cdots a^nb : n \geq 0\}$.

**Solution:** First, we do

$$A = \{a^n b a^m b a^{n+m} : n \geq 0, m \geq 0\}.$$

Assume that the language $A$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^p baba^{p+1}$. Then $s \in A$ and $|s| = 2p + 4 \geq p$. Hence, by the Pumping Lemma, we can write $s = xyz$, where

1. $y \neq \epsilon$,
2. $|xy| \leq p$, and
3. $xy^iz \in A$, for all $i \geq 0$.

Since $|xy| \leq p$, the string $y$ only contains $a$'s from the leftmost $a$-block in $s$. Since $y \neq \epsilon$, the string $y$ contains at least one $a$.

Consider the string $xy^2z = xyyz$. Let $\alpha$, $\beta$, and $\gamma$ be the number of $a$'s in the leftmost $a$-block, middle $a$-block, and rightmost $a$-block in $xyyz$. Then $\alpha \geq p+1$, $\beta = 1$, and $\gamma = p+1$. Since $\alpha + \beta \neq \gamma$, the string $xyyz$ is not in the language $A$. This is a contradiction, because, by the Pumping Lemma, this string is an element of $A$. So we have a contradiction, and we can conclude that $A$ is not regular.

Next, we do

$$B = \{w \in \{a,b\}^* : w \text{ is not a palindrome}\}.$$

We will give two proofs.

For the first proof, assume that the language $B$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let

$$s = a^{p!} b a^{2 \cdot p!},$$

where $2 \cdot p!$ is interpreted as $2(p!)$.

Since $s$ is not a palindrome, this string is in $B$. Note that $|s| = 1 + 3 \cdot p!$, which is at least $p$. Hence, by the Pumping Lemma, we can write $s = xyz$, where

1. $y \neq \epsilon$,
2. $|xy| \leq p$, and
3. $xy^i z \in A$, for all $i \geq 0$.

Since $|xy| \leq p \leq p!$, the string $y$ only contains $a$'s from the leftmost $a$-block in $s$. Since $y \neq \epsilon$, the string $y$ contains at least one $a$.

Let $k$ be the length of the string $y$. Note that $k$ can be any value in $\{1, 2, \ldots, p\}$. Our goal is to find a value for $i$ such that the string $xy^i z$ is a palindrome and, therefore, not in $B$. This will be a contradiction, because, by the Pumping Lemma, the string $xy^i z$ is an element of $B$.

For any $i \geq 0$,

$$xy^i z = a^{p! + (i-1)k} b a^{2 \cdot p!}.$$

This string is a palindrome if and only if

$$p! + (i-1)k = 2 \cdot p!,$$

i.e.,

$$i = 1 + \frac{p!}{k}.$$

Since $1 \leq k \leq p$, this value for $i$ is an integer. Thus, we can take this $i$ and obtain a pumped string that is a palindrome.

For the second proof, we again assume that the language $B$ is regular. Then the complement

$$\overline{B} = \{w \in \{a, b\}^* : w \text{ is a palindrome}\}$$

is also regular. (Going to the complement will make the proof easier!)

Let $p \geq 1$ be the pumping length for $\overline{B}$, as given by the Pumping Lemma. Let $s = a^p b a^p$. Then $s \in \overline{B}$ and $|s| = 2p + 1 \geq p$. Hence, by the Pumping Lemma, we can write $s = xyz$, where

1. $y \neq \epsilon$,
2. $|xy| \leq p$, and
3. $xy^i z \in \overline{B}$, for all $i \geq 0$.

Since $|xy| \leq p$, the string $y$ only contains $a$'s from the leftmost $a$-block in $s$. Since $y \neq \epsilon$, the string $y$ contains at least one $a$.

Consider the string $xy^2z = xyyz$. This string starts with at least $p+1$ many $a$'s, followed by one $b$, and ends with $p$ many $a$'s. Hence, the string $xyyz$ is not a palindrome and, therefore, not in the language $\overline{B}$. This is a contradiction, because, by the Pumping Lemma, this string is an element of $\overline{B}$. So we have a contradiction, and we can conclude that $B$ is not regular.

Next, we do

$$C = \{ucu : u \in \{a,b\}^*\}.$$

As we will see, the proof is basically the same as the one for palindromes.

Assume that the language $C$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^p c a^p$. Then $s \in C$ and $|s| = 2p + 1 \geq p$. Hence, by the Pumping Lemma, we can write $s = xyz$, where

1. $y \neq \epsilon$,
2. $|xy| \leq p$, and
3. $xy^iz \in C$, for all $i \geq 0$.

Since $|xy| \leq p$, the string $y$ only contains $a$'s from the leftmost $a$-block in $s$. Since $y \neq \epsilon$, the string $y$ contains at least one $a$.

Consider the string $xy^2z = xyyz$. This string starts with at least $p+1$ many $a$'s, followed by one $c$, and ends with $p$ many $a$'s. Hence, the string $xyyz$ is not in the language $C$. This is a contradiction, because, by the Pumping Lemma, this string is an element of $C$. So we have a contradiction, and we can conclude that $C$ is not regular.

Finally, we do

$$D = \{aba^2ba^3b \cdots a^nb : n \geq 0\}.$$

Assume that the language $D$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let

$$s = aba^2ba^3b \cdots a^pba^{p+1}b.$$

Then $s \in \overline{B}$. The number of $b$'s in $s$ is equal to $p + 1$. Therefore, $|s| \geq p + 1 \geq p$. Hence, by the Pumping Lemma, we can write $s = xyz$, where

1. $y \neq \epsilon$,
2. $|xy| \leq p$, and
3. $xy^iz \in D$, for all $i \geq 0$.

Since $|xy| \leq p$, the string $y$ does not overlap the rightmost $a^{p+1}b$-block in $s$.

Consider the string $xy^2z = xyyz$. This string is longer than $s$ (because $y \neq \epsilon$) and it ends with $a^{p+1}b$. It is sort of obvious that $xyyz$ is not in $D$. A short argument is as follows:

- If a string is in $D$ and ends with $a^{p+1}b$, then the length of this string is equal to

$$(p+1) + (1+2+3+\cdots+(p+1)) = (p+1) + \frac{(p+1)(p+2)}{2},$$

which is the same as the length of $s$.

The string $xyyz$ ends with $a^{p+1}b$, but its length is more than $|s|$. Therefore, $xyyz$ is not in the language $D$. This is a contradiction, because, by the Pumping Lemma, this string is an element of $D$. So we have a contradiction, and we can conclude that $D$ is not regular.