COMP 3803 — Fall 2025 — Solutions Problem Set 2

Question 1: Consider the language A consisting of all strings over the alphabet $\{a, b\}$ that do not contain bb as a substring. Give a regular expression that describes the language A. As always, justify your answer.

First Solution: Each string in the language A is of the following form:

- It starts with zero or more a's.
  - This is described by the regular expression a^*.
- Then we see zero or more times a string described by baa^*. This guarantees that every b that is not the rightmost symbol is followed by at least one a.
  - This is described by the regular expression $(baa^*)^*$.
- The string may end with b.
  - This is described by the regular expression $\varepsilon \cup b$.

By concatenating these, we get the regular expression

$$a^* (baa^*)^* (\varepsilon \cup b) .$$

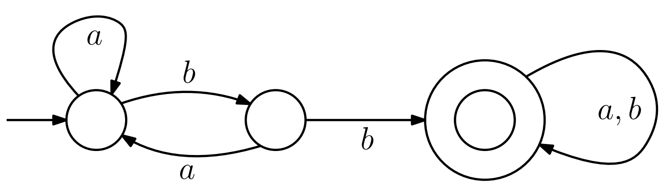
Second Solution: You can check that the following regular expression is also correct:

$$(a \cup ba)^* (\varepsilon \cup b) .$$

Using Prof. Swift's result in Question 2, we can write this as

$$(a^* (ba)^*)^* (\varepsilon \cup b) .$$

Third Solution: Here is a DFA whose language is the set of all strings in $\{a, b\}^*$ that contain bb:



If we "flip" the states, then we get a DFA whose language is the set of all strings in $\{a, b\}^*$ that do not contain bb. Now we can use the technique seen in class to convert this to a regular expression.

Question 2: Let R_1 and R_2 be two arbitrary regular expressions over the same alphabet. Professor Taylor Swift claims that the regular expressions

$$(R_1 \cup R_2)^*$$

and

$$(R_1^* R_2^*)^*$$

describe the same language. If Professor Swift's claim correct? As always, justify your answer.

Solution: As all Swifties know, Taylor is always right. Giving a formal proof is vey painful: You would have to use induction on R_1 and then, inside, induction on R_2. Those of you who come to class know that it is enough to give a proof in English.

Left is contained in right: We show that the language described by $(R_1 \cup R_2)^*$ is contained in the language described by $(R_1^* R_2^*)^*$.

Take an arbitrary string in the language described by $(R_1 \cup R_2)^*$. There is an integer $k \geq 0$, such that this string is described by "k times, do R_1 or R_2". Think of this as "do $A_1, A_2, \ldots, A_k$", where each A_i is either R_1 or R_2. We show that this is contained in $(R_1^* R_2^*)^*$:

- For $i = 1, 2, \ldots, k$:
  - if $A_i =$ "do R_1": We do R_1 once and R_2 zero times. This is contained in $R_1^* R_2^*$.
  - if $A_i =$ "do R_2": We do R_1 zero times and R_2 once. This is contained in $R_1^* R_2^*$.
- The entire for-loop is contained in $(R_1^* R_2^*)^*$.

Right is contained in left: Now we show that the language described by $(R_1^* R_2^*)^*$ is contained in the language described by $(R_1 \cup R_2)^*$.

Take an arbitrary string in the language described by $(R_1^* R_2^*)^*$. There is an integer $k \geq 0$, such that this string is described by "k times, do $R_1^* R_2^*$". Think of this as "do $A_1, A_2, \ldots, A_k$", where each A_i is $R_1^* R_2^*$. We show that this is contained in $(R_1 \cup R_2)^*$:

- For $i = 1, 2, \ldots, k$:
  - Since A_i is "do $R_1^* R_2^*$", there are integers $m_i \geq 0$ and $n_i \geq 0$, such that A_i is "m_i times, do R_1, followed by n_i times, do R_2". This is contained in "$m_i + n_i$ times, do $R_1 \cup R_2$". Thus, A_i is contained in $(R_1 \cup R_2)^*$.
- The entire for-loop is contained in $(R_1 \cup R_2)^*$.

Question 3: In this question, the alphabet is $\{0, 1\}$. Let A be the language consisting of all bitstrings that are the binary representation of an integer at least equal to 40. (Assume that the leftmost bit in the binary representation of a positive integer is 1. For example, the integer 41 in binary is 101001 and not 0101001.) Give a regular expression that describes the language A. As always, justify your answer.

Solution: The solution will be based on the following observations:

- Every integer at least equal to 40 has at least six bits in its binary representation.
- Every integer that has at least seven bits in its binary representation is at least equal to $2^6 = 64$ and, therefore, at least equal to 40.
- Every integer at least equal to 40 that has exactly six bits in its binary representation is
  - either $11 * * * *$, where each $*$ is 0 or 1,
  - or $101 * * *$, where each $*$ is 0 or 1.

This leads to the regular expression

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

$$\cup$$

$$11(0 \cup 1)(0 \cup 1)(0 \cup 1)(0 \cup 1)$$

$$\cup$$

$$101(0 \cup 1)(0 \cup 1)(0 \cup 1)$$

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

$$(a \cup bb)^* (ba^* \cup \varepsilon)$$

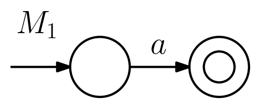
to an NFA. Do not simplify your NFA; just apply the construction rules "without thinking".

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

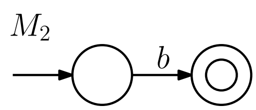
- Take the regular expression a.
- Take the regular expression b.
- Take the regular expressions b and b, and turn them into the regular expression bb.
- Take the regular expressions a and bb, and turn them into the regular expression $a \cup bb$.
- Take the regular expression $a \cup bb$, and turn it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

- Take the regular expressions b and a^*, and turn them into the regular expression ba^*.
- Take the regular expression ε.
- Take the regular expressions ba^* and ε, and turn them into the regular expression $ba^* \cup \varepsilon$.
- Take the regular expressions $(a \cup bb)^*$ and $ba^* \cup \varepsilon$, and turn them into the regular expression $(a \cup bb)^*(ba^* \cup \varepsilon)$.

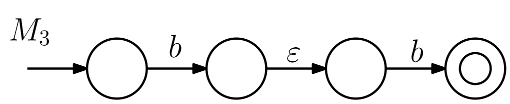
First, we construct an NFA M_1 that accepts the language described by the regular expression a:



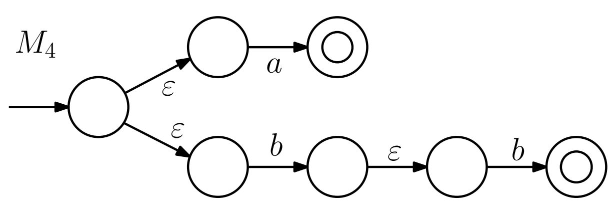
Next, we construct an NFA M_2 that accepts the language described by the regular expression b:



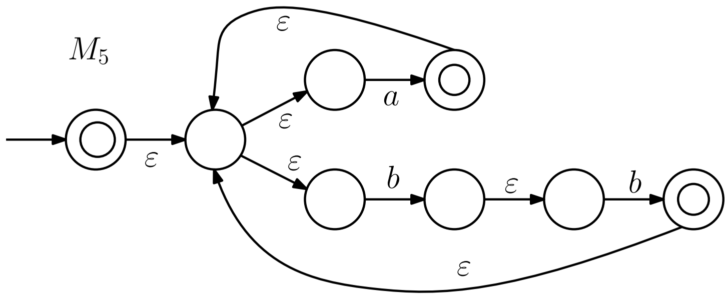
Next, we apply the concatenate construction to M_2 and M_2. This gives an NFA M_3 that accepts the language described by the regular expression bb:



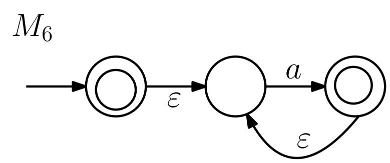
Next, we apply the union construction to M_1 and M_3. This gives an NFA M_4 that accepts the language described by the regular expression $a \cup bb$:



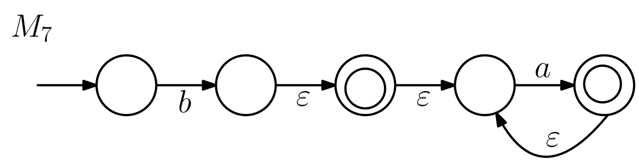
Next, we apply the star construction to M_4. This gives an NFA M_5 that accepts the language described by the regular expression $(a \cup bb)^*$:



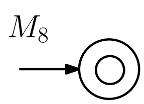
Next, we apply the star construction to M_1. This gives an NFA M_6 that accepts the language described by the regular expression a^*:



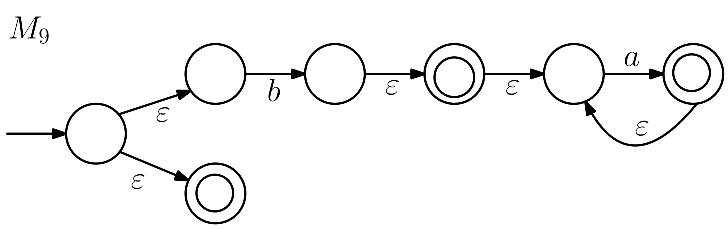
Next, we apply the concatenate construction to M_2 and M_6. This gives an NFA M_7 that accepts the language described by the regular expression ba^*:



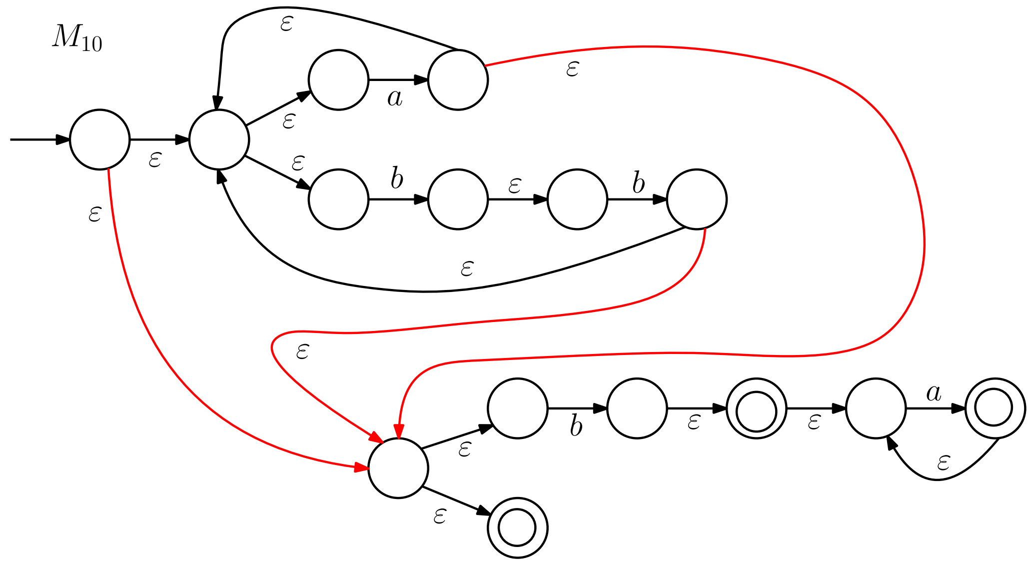
Next, we construct an NFA M_8 that accepts the language described by the regular expression ε:



Next, we apply the union construction to M_7 and M_8. This gives an NFA M_9 that accepts the language described by the regular expression $ba^* \cup \varepsilon$:

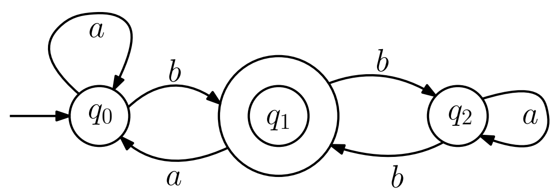


Finally, we apply the concatenate construction to M_5 and M_9. This gives an NFA M_{10} that accepts the language described by the regular expression $(a \cup bb)^*(ba^* \cup \varepsilon)$:



This was fun eh!

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



Solution: For each state q_i, $i = 0, 1, 2$, 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 q_i and corresponds to w ends in the accept state q_1. We obtain the following set of equations:

$$L_0 = aL_0 \cup bL_1 \tag{1}$$
$$L_1 = \varepsilon \cup aL_0 \cup bL_2 \tag{2}$$
$$L_2 = aL_2 \cup bL_1 \tag{3}$$

Since q_0 is the start state, we need a regular expression for L_0.

We use the following tool to solve these equations:

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

We solve the equations (1), (2), and (3), in the following way: Equation (3) is in the form of (4). This gives

$$L_2 = a^* bL_1.$$

By substituting this into (2), we obtain

$$L_1 = \varepsilon \cup aL_0 \cup ba^*bL_1,$$

which we rewrite as

$$L_1 = ba^*bL_1 \cup (\varepsilon \cup aL_0).$$

This equation is in the form of (4). This gives

$$\begin{aligned} L_1 &= (ba^*b)^* (\varepsilon \cup aL_0) \\ &= (ba^*b)^* \cup (ba^*b)^* aL_0. \end{aligned}$$

By substituting this into (1), we obtain

$$\begin{aligned} L_0 &= aL_0 \cup b (ba^*b)^* \cup b (ba^*b)^* aL_0 \\ &= (a \cup b (ba^*b)^* a) L_0 \cup b (ba^*b)^*. \end{aligned}$$

This equation is in the form of (4). This gives

$$L_0 = (a \cup b (ba^*b)^* a)^* b (ba^*b)^*.$$