Question 1: Write your name and student number.

Solution: Karim Benzema, 9

Question 2: Consider the language $A$ consisting of all strings over the alphabet $\{a, b\}$ that contain both $aba$ and $bab$ as substrings. Give a regular expression that describes the language $A$. As always, justify your answer.

Solution: Each string in the language $A$ is of one of the following four types:

- It contains the string $abab$.
  These start with a string in $\{a, b\}^*$, followed by $abab$, and end with a string in $\{a, b\}^*$.

- It contains the string $baba$.
  These start with a string in $\{a, b\}^*$, followed by $baba$, and end with a string in $\{a, b\}^*$.

- $aba$ is to the left of $bab$ and they are not overlapping.
  These start with a string in $\{a, b\}^*$, followed by $aba$, then a string in $\{a, b\}^*$, then $bab$, and end with a string in $\{a, b\}^*$.

- $bab$ is to the left of $aba$ and they are not overlapping.
  These start with a string in $\{a, b\}^*$, followed by $bab$, then a string in $\{a, b\}^*$, then $aba$, and end with a string in $\{a, b\}^*$.

This gives the regular expression

$$(a\cup b)^*abab(a\cup b)^*\cup(a\cup b)^*baba(a\cup b)^*\cup(a\cup b)^*aba(a\cup b)^*bab(a\cup b)^*\cup(a\cup b)^*bab(a\cup b)^*aba(a\cup b)^*$$

Question 3: Let $A$ be the language over the alphabet $\{a, b\}$ that is described by the regular expression $aa$. Give a regular expression that describes the complement $\overline{A}$ of $A$. As always, justify your answer.

Solution: The language described by the regular expression $aa$ is $A = \{aa\}$. We need a regular expression for its complement, i.e., all strings that are not equal to $aa$.

Each string in $\overline{A}$ is of one of the following three types:

- Any string of length at most one.
- It has length exactly two and is not equal to $aa$.
- It has length at least three.

$$(a\cup b)^*aa(a\cup b)^*\cup(a\cup b)^*baba(a\cup b)^*\cup(a\cup b)^*aba(a\cup b)^*bab(a\cup b)^*\cup(a\cup b)^*bab(a\cup b)^*aba(a\cup b)^*$$
This gives the regular expression
\[\varepsilon \cup a \cup b \cup ab \cup ba \cup bb \cup (a \cup b)(a \cup b)(a \cup b)^*\]

**Question 4:** In this question, the alphabet is \{a, b\}. A block in a string is a maximal substring all of whose symbols are the same. For example, the string aaabbaa has three blocks: aaa, bb, and aa.

Let \( A \) be the language of all strings \( w \) such that every block in \( w \) has length two or three. The empty string is in \( A \), as is the string aaabbaa.

Give a regular expression that describes the language \( A \). As always, justify your answer.

**Solution:** Throughout the solution, a block always refers to a block of length two or three.

Each string in the language \( A \) is of one of the following two types:

- The string starts with an \( a \)-block, and the \( a \)-blocks and \( b \)-blocks alternate. The total number of blocks can be even or odd.
- The string starts with a \( b \)-block, and the \( b \)-blocks and \( a \)-blocks alternate. The total number of blocks can be even or odd.

This gives the regular expression
\[((aa \cup aaa)(bb \cup bbb))^*(\varepsilon \cup aa \cup aaaa) \cup ((bb \cup bbb)(aa \cup aaaa))^*(\varepsilon \cup bb \cup bbbb)\]

Note that \( \varepsilon \) is in the language described by this regular expression.

**Question 5:** Use the construction given in class to convert the regular expression
\[a \cup ba^*\]
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 expression \( a \), and turn it into the regular expression \( a^* \).
- Take the regular expressions \( b \) and \( a^* \), and combine them into the regular expression \( ba^* \).
- Take the regular expressions \( a \) and \( ba^* \), and combine them into the regular expression \( a \cup ba^* \).
First, we construct an NFA $M_1$ that accepts the language described by the regular expression $a$:

$$M_1\quad \xrightarrow{a}\quad$$

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

$$M_2\quad \xrightarrow{b}\quad$$

Next, we apply the star construction to $M_1$. This gives an NFA $M_3$ that accepts the language described by the regular expression $a^*$:

$$M_3\quad \xrightarrow{\varepsilon}\quad \xrightarrow{a}\quad \xrightarrow{\varepsilon}\quad$$

Next, we apply the concatenate construction to $M_2$ and $M_3$. This gives an NFA $M_4$ that accepts the language described by the regular expression $ba^*$:

$$M_4\quad \xrightarrow{b}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{a}\quad \xrightarrow{\varepsilon}\quad$$

Finally, we apply the union construction to $M_1$ and $M_4$. This gives an NFA $M_5$ that accepts the language described by the regular expression $a \cup ba^*$:

$$M_5\quad \xrightarrow{a}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{b}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{\varepsilon}\quad \xrightarrow{a}\quad \xrightarrow{\varepsilon}\quad$$

**Question 6:** 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 the accept state. We obtain the following set of equations:

$$
L_1 = aL_1 \cup bL_2 \\
L_2 = aL_3 \cup bL_2 \\
L_3 = \epsilon \cup aL_1 \cup bL_2
$$

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

We use the following tool to solve these equations:

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

We solve the equations (1), (2), and (3), in the following way: By substituting (3) into (2), we obtain

$$
L_2 = a(\epsilon \cup aL_1 \cup bL_2) \cup bL_2,
$$

which we rewrite as

$$
L_2 = (ab \cup b)L_2 \cup (a \cup aaL_1). \quad (5)
$$

This equation is in the form of (4), with $L = L_2$, $B = (ab \cup b)$, and $C = (a \cup aaL_1)$. Since $\epsilon$ is not in the language described by $B$, we can apply (4) to (5), and we obtain

$$
L_2 = (ab \cup b)^*(a \cup aaL_1),
$$

which we rewrite as

$$
L_2 = (ab \cup b)^*a \cup (ab \cup b)^*aaL_1. \quad (6)
$$

By substituting (6) into (1), we obtain

$$
L_1 = aL_1 \cup b((ab \cup b)^*a \cup (ab \cup b)^*aaL_1),
$$

which we rewrite as

$$
L_1 = (a \cup b(ab \cup b)^*aa) L_1 \cup b(ab \cup b)^*a. \quad (7)
$$

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

$$
B = (a \cup b(ab \cup b)^*aa)
$$
and $C = b(ab \cup b)^*a$. Since $\epsilon$ is not in the language described by $B$, we can apply (4) to (7), and we obtain

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

**Question 7:** Use the Pumping Lemma to prove that the following languages are not regular. In all cases, the alphabet is $\{a, b\}$.

1. $\{a^n u : n \geq 0, u \in \{a, b\}^*, |u| \leq n\}$.
2. $\{a^m b^n : m \geq 0, n \geq 0, n \text{ is a multiple of } m\}$.
3. $\{a^m b^n : m \geq 0, n \geq 0, m \text{ is a multiple of } n\}$.

   (a) Professor Justin Bieber claims that this can be proven by taking the string $s = a^p b^p$, where $p$ is the Pumping length. Show that Professor Bieber is (again!) wrong.

   (b) Now give a correct proof.
4. $\{uv : u \in \{a, b\}^*, v \in \{a, b\}^*, u = v^R\}$.

Note: If $v = v_1 v_2 \cdots v_n$ is a string, then $v^R = v_n v_{n-1} \cdots v_1$ is the reverse of $v$.

**Solution:** First, we do

$$A = \{a^n u : n \geq 0, u \in \{a, b\}^*, |u| \leq n\}.$$ 

Assume the language $A$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^p b^p$. We can write $s = a^p u$, where $u = b^p$. Since $|u| \leq p$, the string $s$ is in $A$. Also, $|s| = 2p \geq p$. Hence, by the Pumping Lemma, we can write $s = xyz$, where

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

Since $|xy| \leq p$, the string $y$ contains only $a$s. Since $y \neq \epsilon$, the string $y$ contains at least one $a$. Let $k = |y|$, so that $y = a^k$. Note that $1 \leq k \leq p$. By the Pumping Lemma, the string

$$s' = xy^0 z = a^{p-k} b^p$$

is in $A$. Hence, there is a $j$ such that $s' = a^j u$, where $u \in \{a, b\}^*$ and $|u| \leq j$.

However, since $j \leq k - p$, we must have $|u| \geq p > p - k \geq j$. Thus, $s'$ is not in $A$. So we have a contradiction, and we can conclude that $A$ is not regular.
Next we do
\[ B = \{ a^m b^n : m \geq 0, n \geq 0, \text{ } n \text{ is a multiple of } m \}. \]
Assume the language \( B \) is regular. Let \( p \geq 1 \) be the pumping length, as given by the
Pumping Lemma. Let \( s = a^p b^p \). Since \( p \) is a multiple of \( p \), the string \( s \) is in \( B \). Also,
\( |s| = 2p \geq p \). Hence, by the Pumping Lemma, we can write \( s = xyz \), where

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

Since \( y \neq \epsilon \) and \( |xy| \leq p \), the string \( y \) has the form \( y = a^k \), for some integer \( k \) with \( 1 \leq k \leq p \).

Consider the string
\[ s' = xy^2 z = xy yz = a^{p+k} b^p. \]
By the Pumping Lemma, \( s' \) is in \( B \). However, since \( p \) is not a multiple of \( p+k \), the string
\( s' \) is not in \( B \). So we have a contradiction, and we can conclude that \( B \) is not regular.

Next we do
\[ C = \{ a^m b^n : m \geq 0, n \geq 0, \text{ } m \text{ is a multiple of } n \}. \]
We first show why Professor Bieber should fail COMP 3803: The string \( s = a^p b^p \) is in \( C \)
because \( p \) is a multiple of \( p \). Also \( |s| = 2p \geq p \). Hence, by the Pumping Lemma, we can write \( s = xyz \), where

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

Since \( y \neq \epsilon \) and \( |xy| \leq p \), the string \( y \) has the form \( y = a^k \), for some integer \( k \) with \( 1 \leq k \leq p \).

\textbf{It may happen that } \( k = p \), i.e., \( x = \epsilon \) and \( y = a^p \). In this case, for every \( i \geq 0 \),
\[ xy^i z = a^{ip} b^p. \]
Since \( ip \) is a multiple of \( p \), all these strings \( xy^i z \) are in \( C \). In other words, we do not get a
contradiction.

Here are two correct proofs.

\textbf{First proof:} Assume the language \( C \) is regular. The reverse language \( C^R \), obtained by
reversing all strings in \( C \) is equal to
\[ C^R = \{ b^n a^m : m \geq 0, n \geq 0, \text{ } m \text{ is a multiple of } n \}. \]
We have seen in class that the reverse of a regular language is also regular. Thus, \( C^R \) is
regular. But, by swapping \( a \) and \( b \), \( C^R \) is equal to \( B \), the language in part 2. Thus, \( B \) is
regular. This is a contradiction. We conclude that \( C \) is not regular.
Second proof: Assume the language $C$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^{p+1}b^{p+1}$. Since $p + 1$ is a multiple of $p + 1$, the string $s$ is in $B$. Also, $|s| = 2p + 2 \geq p$. 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 $y \neq \epsilon$ and $|xy| \leq p$, the string $y$ has the form $y = a^k$, for some integer $k$ with $1 \leq k \leq p$. Consider the string

$$s' = xy^2z = xyyz = a^{p+1+k}b^{p+1}.$$ 

By the Pumping Lemma, $s'$ is in $C$. Thus, $p + 1 + k$ is a multiple of $p + 1$. However,

$$p + 1 < p + 1 + k \leq p + 1 + p < 2(p + 1),$$

i.e., $p + 1 + k$ is strictly between two consecutive multiples of $p + 1$. It follows that $p + 1 + k$ is not a multiple of $p + 1$. So we have a contradiction, and we can conclude that $C$ is not regular.

Finally, we do

$$D = \{ uv : u \in \{a, b\}^*, v \in \{a, b\}^*, u = v^R \}.$$ 

Assume the language $D$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^pbpa^p$. Since the first half is the reverse of the second half, the string $s$ is in $D$. Also, $|s| = 2p + 2 \geq p$. 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 $y$ has the form $y = a^k$, for some integer $k$ with $1 \leq k \leq p$. Consider the string

$$s' = xy^2z = xyyz = a^{p+k}bba^p.$$ 

By the Pumping Lemma, $s'$ is in $D$. Note that the reverse of any string in $D$ is also in $D$. It is clear, however, that the reverse of $s'$ is not in $D$. This is a contradiction. So we have a contradiction, and we can conclude that $D$ is not regular.

Question 8: Consider the language

$$A = \{a^{n^2} : n \geq 0\} \cup \{a^{2n+1} : n \geq 0\};$$

note that the alphabet is $\{a\}$.

In this question, you will prove that $A$ is not regular, but the concatenation $AA$ is regular.
1. Prove that $A$ is not a regular language.

   **Hint:** IGNORE THIS HINT!!!!!!! What is $A \cap \{a^{2n+1} : n \geq 0\}$?

2. Prove that $AA = \{a^n : n \geq 0\}$.

3. Prove that $AA$ is a regular language.

**Solution:** SINCE THE HINT IS A PoS, THIS QUESTION WILL NOT BE MARKED.

We start with part 1. Assume the language $A$ is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^{(2p)^2}$, i.e., the string consisting of $(2p)^2 = 4p^2$ many $a$’s. Then $s$ is a string in $A$ and $|s| = 4p^2 \geq p$. 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 $y \neq \epsilon$ and $|xy| \leq p$, the string $y$ has the form $y = a^k$, for some integer $k$ with $1 \leq k \leq p$. Consider the string $s' = xy^3z = xyyyz = a^{(2p)^2+2k}$.

By the Pumping Lemma, $s'$ is in $A$. We now argue that $s'$ is not in $A$. First, the length of $s'$ is even. Second $(2p)^2 < (2p)^2 + 2k \leq (2p)^2 + 2p < (2p + 1)^2$ and, thus, the length of $s'$ is not a square (because it is strictly between two consecutive squares). So we have a contradiction, and we can conclude that $A$ is not regular.

For part 2., it is clear that $AA \subseteq \{a^n : n \geq 0\}$. We show that $\{a^n : n \geq 0\} \subseteq AA$:

- Since $\epsilon \in A$, the string $a^0 = \epsilon \epsilon$ is in $AA$.
- Let $n \geq 0$. Since both $\epsilon$ and $a^{2n+1}$ are in $A$, the string $a^{2n+1} = \epsilon a^{2n+1}$ is in $AA$.
- Let $n \geq 2$ be an even integer. Then $n = 1 + (n - 1)$, which is a sum of two odd integers. Since both $a$ and $a^{n-1}$ are in $A$, the string $a^n = aa^{n-1}$ is in $AA$.

For part 3., we have just shown that $AA = \{a^n : n \geq 0\}$. Since the regular expression $a^*$ describes $AA$, this language is regular.