COMP 3803 — Fall 2025 — Solutions Problem Set 3

Question 1: Is the language

$$L = \{a^k b^\ell a^m : k \geq 0, \ell \geq 0, m \geq 0, k + \ell + m \geq 2025\}$$

regular? As always, justify your answer.

Solution: The language L is regular. To prove this, we are going to write L as the union of finitely many languages, each of which is regular.

- For each triple (x, y, z) of non-negative integers for which $x + y + z = 2025$, consider the regular expression

$$R(x, y, z) = \underbrace{a \cdots a}_{x} a^* \underbrace{b \cdots b}_{y} b^* \underbrace{a \cdots a}_{z} a^*.$$

  This regular expression describes a regular language. Each string in this language belongs to L.

- In COMP 2804, you have learned that the number of such triples (x, y, z) is equal to $\binom{2027}{2}$; in particular, this number is finite.

It remains to show that each string in L is in the language of some regular expression $R(x, y, z)$.

Let $k \geq 0$, $\ell \geq 0$, and $m \geq 0$ be such that $k + \ell + m \geq 2025$. Consider the string $a^k b^\ell a^m$. Let x, y, and z be integers such that

$$0 \leq x \leq k, 0 \leq y \leq \ell, 0 \leq z \leq m, x + y + z = 2025.$$

Then the string $a^k b^\ell a^m$ is in the language described by the regular expression $R(x, y, z)$.

Question 2: For any string $w \in \{a, b\}^*$, we denote the number of a's in w by $N_a(w)$, and we denote the number of b's in w by $N_b(w)$. Consider the language

$$A = \{w \in \{a, b\}^* : N_a(w) = N_b(w)\}.$$

Assume that we are going to use the Pumping Lemma to prove that A is not regular. As always, we assume that A is regular. The Pumping Lemma gives us a pumping length p.

1. Explain in a few sentences why we may assume that p is even.
2. Given that p is even, can we choose the string $s = a^{p/2} b^{p/2}$ to obtain a contradiction?

Solution: For the first part, the Pumping Lemma gives us an integer p such that every string in A having length at least p can be "pumped". This implies that every string in A having length at least $p+1$ can also be "pumped". If p happens to be odd, then we can take $p+1$ to be the pumping length.

For the second part, we assume that p is even. Consider the string $s = a^{p/2}b^{p/2}$. Then $s \in A$ and $|s| = p$. 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 $i \geq 0$.

It may happen that $y = a^kb^k$ for some k with $1 \leq k \leq p/2$. In this case, for every $i \geq 0$, the string xy^iz contains as many a's as b's. Thus, each such string is in A. In other words, this string s is the wrong string.

Question 3: Use the Pumping Lemma to prove that the following languages are not regular.

1. $\{a^kb^\ell a^m : k \geq 0, \ell \geq 0, m \geq 0, \text{ and } k = \ell \text{ or } \ell \neq m\}$. The alphabet is $\{a, b\}$.
2. $\{a^mb^n : m \geq 0, n \geq 0, \ m+n \text{ is a prime number}\}$. The alphabet is $\{a, b\}$.
3. $\{a^m(ab)^n(ca)^{2m} : m \geq 1, n \geq 1\}$. The alphabet is $\{a, b, c\}$.
4. $\{a^ib^mc^n : i \geq 1, m \geq 1, n \geq m+1\}$. The alphabet is $\{a, b, c\}$.

Solution: First, we do

$$A = \{a^kb^\ell a^m : k \geq 0, \ell \geq 0, m \geq 0, \text{ and } k = \ell \text{ or } \ell \neq m\}.$$

Assume the language A is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let $s = a^pb^pa^p$. Then $s \in A$ and $|s| = 3p \geq p$. 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 contains only a's from the left block of a's in 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

$$s' = xy^2z = a^{p+k}b^pa^p$$

is in A. However, s' is not in A, because the left a-block is longer than the b-block, and the b-block has the same length as the right a-block. Thus, we have a contradiction, and we can conclude that A is not regular.

Next we do

$$B = \{a^m b^n : m \geq 0, n \geq 0,\ m + n \text{ is a prime number}\}.$$

Since the value of n can be zero, we can copy the proof done in class: Take the string $s = a^m b^0 = a^m$, where m is a prime number with $m \geq p$ and p is the pumping length .

Here is a different proof (it is basically the same). Assume the language B is regular. Let $p \geq 1$ be the pumping length, as given by the Pumping Lemma. Let n be a prime number such that $n \geq p$. Let $s = a^p b^{n-p}$. Then $s \in B$ and $|s| = n \geq p$. 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 $|xy| \leq p$, 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, for each $i \geq 0$, the string

$$xy^i z = a^{p+(i-1)k} b^{n-p}$$

is in B. Therefore, for each $i \geq 0$,

$$p + (i-1)k + n - p = (i-1)k + n$$

is a prime number. However, for $i = n + 1$ we have

$$(i-1)k + n = nk + n = n(k+1),$$

which is not a prime number because $n \geq 2$ and $k + 1 \geq 2$. Thus, we have a contradiction, and we can conclude that B is not regular.

Next we do

$$C = \{a^m (ab)^n (ca)^{2m} : m \geq 1, n \geq 1\}.$$

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

$$s = a^p (ab)(ca)^{2p}.$$

Then $s \in C$ and $|s| = 3p + 2 \geq p$. 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 contains only a's from the a^p-block in 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

$$s' = xyyz = a^{p+k}(ab)(ca)^{2p}$$

is in C. However, s' is not in C, because the length of the a^{p+k}-block is not half of the length of the $(ca)^{2p}$ block. Thus, we have a contradiction, and we can conclude that C is not regular.

Finally we do

$$D = \{a^ib^mc^n : i \geq 1, m \geq 1, n \geq m+1\}.$$

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

$$s = ab^pc^{p+1}.$$

Then $s \in D$ and $|s| = 2p + 2 \geq p$. 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 $i \geq 0$.

Observe that y is contained in the ab^p part of s. We distinguish two cases.

Case 1: y does not contain the symbol a.

In this case, y only contains b's. Consider the string $s' = xyyz$. By the Pumping Lemma, $s' \in D$. However, s' is not in D, because the number of b's in s' is at least $p+1$, whereas the number of c's in s' is equal to $p+1$.

Case 1: y contains the symbol a (and, thus, $x = \varepsilon$).

Consider the string $s' = xy^0z = xz$. By the Pumping Lemma, $s' \in D$. However, s' does not contain any a and, therefore, s' is not in D.

In both cases, we have a contradiction. Therefore, D is not regular.

Question 4: Even though Justin Bieber is enjoying COMP 3803, he is confused about the Pumping Lemma. Here is Justin's confusion:

1. Justin knows that any finite language is regular. Can you prove this in a few sentences?
2. Let A be a finite regular language. By the Pumping Lemma, there is an integer pumping length $p \geq 1$. Take an arbitrary string s in A, with $|s| \geq p$. By the Pumping Lemma, we can write $s = xyz$, such that for every integer $i \geq 0$, the string xy^iz is in A. Since there are infinitely many integers $i \geq 0$, this seems to imply that A

contains infinitely many strings. Therefore, the Pumping Lemma is not valid for finite languages.

Is Justin's reasoning correct?

Solution: First we show that any finite language is regular. Let A be a finite language and let m be the number of strings in A. Number the strings in A as $s_1, s_2, \ldots, s_m$. For each i, define the language $A_i = \{s_i\}$. Then

$$A = \{s_1, s_2, \ldots, s_m\} = \bigcup_{i=1}^{m} A_i.$$

For each i, the language A_i is regular, because s_i is a regular expression that describes A_i. For example, if $s_i = abba$, then $abba$ is a regular expression. We conclude that A is the union of m many regular languages. Therefore, A is regular.

For the second part, as always, Justin's reasoning is not correct: Since the language A is regular, the length of a longest string, say ℓ, is well-defined. Let M be an arbitrary DFA that accepts A. Remember from the proof of the Pumping Lemma, that we can take the pumping length p to be equal to the number of states of M.

We will show below that $p \geq \ell + 1$. Assume that this is true. Then the Pumping Lemma says that any string in A, whose length is at least p (and thus, at least $\ell + 1$) can be pumped. Since the subset of all strings in A whose lengths are at least $\ell + 1$ is empty, this becomes

$$\forall s \in \emptyset : \; s \text{ can be pumped},$$

which is a true statement! Thus, the Pumping Lemma is valid for finite languages.

It remain to prove that $p \geq \ell + 1$. Assume that $p \leq \ell$. Let s be a longest string in A. Then $|s| = \ell \geq p$. By the Pumping Lemma, we can write $s = xyz$, such that for every integer $i \geq 0$, the string xy^iz is in A. Since $y \neq \varepsilon$, and since there are infinitely many integers $i \geq 0$, this implies that A contains infinitely many strings. This is a contradiction.

Question 5: Let A be an arbitrary regular language, let M be a DFA that accepts A, and let p be the number of states of M.

Prove that A is non-empty if and only if there is a string s in A whose length is strictly less than p.
Hint: There is a reason why the letter p is used to denote the number of states of M.

Solution: If there is a string in A whose length is strictly less than p, then A is obviously non-empty.

For the converse, we assume that A is non-empty. Let s be a shortest string in A. We will show by contradiction that $|s| < p$.

Assume that $|s| \geq p$. Since A is regular, the Pumping Lemma gives us a pumping length. From the proof of the Pumping Lemma, we know that we can take the pumping length to be the number of states in M, which is p.

Since $s \in A$ and $|s| \geq p$, the Pumping Lemma tells us that we can write $s = xyz$, where $y \neq \varepsilon$, $|xy| \leq p$, and for all $i \geq 0$, the string xy^iz is in A.

Thus, for $i = 0$, the string $s' = xy^0z = xz$ is in A. Since $|y| \geq 1$, the string s' is strictly shorter than s. Thus, s is not a shortest string in A. This is a contradiction.

Question 6: Consider the context-free grammar $G = (V, \Sigma, R, S)$, where the set of variables is $V = \{S, A, B\}$, the set of terminals is $\Sigma = \{a, b\}$, the start variable is S, and the rules are as follows:

$$\begin{array}{lcl} S & \rightarrow & abB \\ A & \rightarrow & \varepsilon \mid aaBb \\ B & \rightarrow & bbAa \end{array}$$

Prove that the language $L(G)$ that is generated by G is equal to

$$L(G) = \{ab(bbaa)^n bba(ba)^n : n \geq 0\}.$$

(Remember: To prove that two sets X and Y are equal, you have to prove that $X \subseteq Y$ and $Y \subseteq X$.)

Solution: We write

$$L = \{ab(bbaa)^n bba(ba)^n : n \geq 0\},$$

so that we have to prove that $L = L(G)$.

First we prove that $L \subseteq L(G)$. If we start with the variable B and apply the rule $B \rightarrow bbAa$ followed by the rule $A \rightarrow aaBb$, then we see that

$$B \Rightarrow bbAa \Rightarrow (bbaa)B(ba).$$

If we repeat this n times, then we see that

$$B \overset{*}{\Rightarrow} (bbaa)^n B(ba)^n.$$

It follows that, for each integer $n \geq 0$,

$$\begin{array}{lcl} S & \Rightarrow & abB \\ & \overset{*}{\Rightarrow} & ab(bbaa)^n B(ba)^n \\ & \Rightarrow & ab(bbaa)^n bbAa(ba)^n \\ & \Rightarrow & ab(bbaa)^n bb\varepsilon a(ba)^n = ab(bbaa)^n bba(ba)^n. \end{array}$$

This proves that each string in L can be derived from the start variable S. In other words, this proves that $L \subseteq L(G)$.

It remains to show that $L(G) \subseteq L$, i.e., no other strings can be derived from the start variable S.

To derive a string in $L(G)$, we must start with the start variable S. At the start, the only rule that can be applied is $S \rightarrow abB$, after which the only rule that can be applied is $B \rightarrow bbAA$. At this moment, we apply either the rule $A \rightarrow \varepsilon$ or the rule $A \rightarrow aaBb$. In the first case, we are done. In the second case, we can only apply the rule $B \to bbAA$, after which we either apply $A \to \varepsilon$ or $A \to aaBb$. From this, it follows that the only derivations in the grammar G are the ones given in the proof of the fact that $L \subseteq L(G)$.

Question 7: Give context-free grammars that generate the following languages. For each case, justify your answer.
(7.1) $\{a^{n+3}b^n : n \geq 0\}$. The set of terminals is equal to $\{a, b\}$.
(7.2) $\{a^n b^m : n \geq 0, m \geq 0, 2n \leq m \leq 3n\}$. The set of terminals is equal to $\{a, b\}$.
(7.3) $\{a^m b^n c^n : m \geq 0, n \geq 0\}$. The set of terminals is equal to $\{a, b, c\}$.

Solution:

We start with

$$L_1 = \{a^{n+3}b^n : n \geq 0\}.$$

We can write L_1 as

$$L_1 = aaaL_1',$$

where any string in L_1' is either

- empty or
- starts with a, followed by a string in L_1', and ends with b.

This leads to the context-free grammar $G = (V, \Sigma, R, S)$, where $V = \{S, A\}$, $\Sigma = \{a, b\}$, and R consists of the rules

$$\begin{aligned} S &\to aaaA \\ A &\to \varepsilon \mid aAb \end{aligned}$$

Observe that from A, we can derive all strings of the form $a^n b^n$ for some $n \geq 0$. From S, we can derive all strings that start with aaa and are followed by any string that can be derived from the variable A. Therefore, from S, we can derive all strings in L_1 (and nothing else).

Next we do

$$L_2 = \{a^n b^m : n \geq 0, m \geq 0, 2n \leq m \leq 3n\}.$$

Any string in L_2 is either

- empty or
- is a non-empty string in which all the a's are to the left of all the b's, and for each a, there are two or three b's.

This leads to the context-free grammar $G = (V, \Sigma, R, S)$, where $V = \{S\}$, $\Sigma = \{a, b\}$, and R consists of the rules

$$S \to \varepsilon \mid aSbb \mid aSbbb$$

It is clear that for each string in $L(G)$, all a's are to the left of all b's, and the number of b's is at least twice and at most three times the number of a's.

It remains to argue that every string in L_2 is in $L(G)$. Let $u = a^n b^m$ be an arbitrary string in L_2, where $n \geq 0$, $m \geq 0$, and $2n \leq m \leq 3n$. The string u is derived from the start variable S in the following way:

- Start with S, and apply the rule $S \rightarrow aSbbb$ exactly $m-2n$ times (note that $m-2n \geq 0$). This gives
$$S \stackrel{*}{\Rightarrow} a^{m-2n} S b^{3(m-2n)}.$$
- Now apply the rule $S \rightarrow aSbb$ exactly $3n - m$ times (note that $3n - m \geq 0$). This gives
$$S \stackrel{*}{\Rightarrow} a^{m-2n} \left(a^{3n-m} S b^{2(3n-m)} \right) b^{3(m-2n)}.$$
- Finally, apply the rule $S \rightarrow \varepsilon$. This gives
$$S \stackrel{*}{\Rightarrow} a^{m-2n} \left(a^{3n-m} \, \varepsilon \, b^{2(3n-m)} \right) b^{3(m-2n)} = a^n b^m = u.$$

Finally, we do
$$L_3 = \{a^m b^n c^n : m \geq 0, n \geq 0\}.$$
Any string in L_3

- starts with zero or more a's, followed by a string of the form $b^n c^n$, for some $n \geq 0$.

This leads to the context-free grammar $G = (V, \Sigma, R, S)$, where $V = \{S, X\}$, $\Sigma = \{a, b, c\}$, and R consists of the rules
$$\begin{array}{lcl} S & \rightarrow & AX \\ A & \rightarrow & \varepsilon \mid aA \\ X & \rightarrow & \varepsilon \mid bXc \end{array}$$
Observe that from A, we can derive all strings of the form a^m for some $m \geq 0$. From X, we can derive all strings of the form $b^n c^n$, for some $n \geq 0$. Therefore, from S, we can derive all strings in L_3 (and nothing else).