- Start as early as possible, and contact the instructor if you get stuck.
- See the course outline for details about the course's marking policy and rules on collaboration.
- Submit your completed solutions to **Crowdmark**.

1. Regular Expressions

[6] Let $\Sigma = \{0, 1\}$. Give a **rigourous** proof for the equality of languages

$$L((00^*1)^*1) = L(1 + 0(0 + 10)^*11).$$

Let $L_1 = (00^*1)^*1$ and $L_2 = 1 + 0(0 + 10)^*11$ for conveinence. Notice,

$$\begin{aligned} 1 + 0(0 + 10)^*11 &\equiv (\epsilon + 0(0 + 10)^*1)1 \\ &\equiv (0(0 + 10)^*1)^*1 \\ &\equiv (00^*1)^*1 + (0(10)^*1)^*1 \end{aligned} \tag{1}$$

To prove that $L_1 \subseteq L_2$ is trivial since $L_2 = (00^*1)^*1 + (0(10)^*1)^*1$ and the regular expression defining $L_1$ is the first term of the union in $L_2$.

Proving the containment $L_2 \subseteq L_1$ we let $w \in L2$ be arbitrary and note that $w$ can be split into substrings $x_1x_2 \dots x_n$, $n > 0$. Each substring $x_i$, $i \leq n$, will take either the form $(00^*1)^*1$ or $(0(10)^*1)^*1$. Since we have already covered the case where $x_i = (00^*1)^*1$ in the first containment we consider when $x_i = (0(10)^*1)^*1$. To show every $x_i \in L_2$ is also in $L_1$ we will use structural induction on the string $u = (0(10)^*1)^*1$
We begin by describing the structure of $u$. From the basis $u$ has a minimum length of 1 and any increase to it's length can be attributed to one of the Kleene closures. We describe the lengthening of $u$ with the following equation:

$$u = (0(10)^i1)^j1$$

We can see from the above equation that there are 3 cases to consider.

**Case $i$ increasing but $j$ remains the same** affixing $j = 1$ then $u = 011, 01011, 0101011, ...$ and takes the general form $0(01)^*11$

**Case $j$ increasing but i remains the same** affixing $ito0$ then $u = 011, 01011, 0101011, ...$ and takes the gernal form $(01)^*1$

**Case $i$ and $j$ both increase** starting j and i at 1 and increasing gives u = 01011, 0101010101011, ...

In all cases the string $u$ starts with a 0 has a repeating segment of (10) which occurs zero or more times and ends with 11. When $i$ increased by one it results in exactly one copy of 10 being appended to the string. When $j$ is increaed it represents $i * j$ appensions of 10 to the string. Since $i$ and $j$ are unbounded, affixing $j$ to 1 will have no effect since any increase of $j$ can be expressed by $i * j$ increases to $i$. We will therefore induct on $i$ to prove that all strings formed by u appear in $L_1$
**Basis:**

$i = j = 0$ As $u$ does not accept the empty string the minimum string it accepts can be seen by choosing $i = j = 0$ , taking 0 occurances of each. Then, $(0(10)^i1)^j1 = (0(10)^01)^01 = 1$. Similarly for $L_1$ we can see $L_1 = (00^01)^01 = 1$.

$i = 0, j = 1$ For $u$ we have $u = (0(10)^01)^11 = 011$ Showing for $L_1$ we take the case where $L_1 = (00^01)^11 = 011$ and $u \in L_1$.

$i = j = 1$ Looking at $u = (0(10)^i1)^j1$ we define another base case where $i = j = 1$ since in our induction we will be affixing j at 1. Then $u = (0(10)^11)^11 = 01011$, and looking to $L_1$ we see that $L_1 = (00^01)^21 = 010111$

**Induction:** Assume $u \in L_1$ where, $u = (0(10)^i1)^j1$ and $j = 1$ for all $i > 1$ Then we will show true for $i + 1$

$$\begin{aligned} u &= (0(10)^{i+1}1)^11 \\ u &= 0(10)^{1+i}11 \\ &= 010(10)^i11 \\ &= 01(0(10)^i1)1 \end{aligned} \tag{2}$$

Now we consider $L_1$, we let $L_1 = (00^m1)^n1$ from the basis when $m = 0$ and $n = 1$ $L_1 = 011$. If we continue to let $m = 0$ and consider an arbitrary $n$ we have $L_1 = (00^01)^n1 = (01)^n1$ then as $n$ increases we can see $L_1 = (01)^{n+1}1 = 01(01)^n1$ from the structure of $L_1$ we see that with every increase of $n$ a 01 is appended at the beginning of the string. Likewise, for $u$ any increase in $i$ will appead a 01 to the original string. As we have already shown in the basis that $u \in L_1$ when $i, j \geq 0$, then from our inductive hypothesis we assumed true for $i > 1$ and finally, through the structure of $u$ showed any increase of $i$ to append 01 to $u$ and since we were able to show the same is true for any incremental increases in the length $L_1$ where any increase to $n$ results in a 01 being appended to the string we can be confident that $u \subseteq L_1$ Thus $L_2 \subseteq L_1$. Since we have already shown that $L_1 \subseteq L_2$ We conclude that,

$$L_1 = L_1$$

2. Closure Properties of Regular Languages

Let $\Sigma$ be an alphabet. Consider the following unary operation on languages over $\Sigma$:

$$Pref(L) = \{w \mid \exists x \in \Sigma^* \text{ such that } wx \in L\}.$$

In this question you will supply the key content of two of the recursive cases for a proof by structural induction on $L$ that

if $L$ is regular, then it follows that $Pref(L)$ is also regular.

In such a proof, we would be constructing new languages starting from regular languages $L_1$ and $L_2$, such that $Pref(L_1)$ and $Pref(L_2)$ are regular by our induction hypothesis. (I encourage you to write down the rest of this proof in your spare time after you finish working on this assignment!)

[6] (a) Let $L_1$ and $L_2$ be any languages over $\Sigma$, with $L_2 \neq \emptyset$. Give a **rigourous** proof of the equality of languages:

$$Pref(L_1L_2) = Pref(L_1) \cup (L_1 Pref(L_2)).$$

[**Remark:** We lose no generality in the proof of regularity by assuming $L_2 \neq \emptyset$ here; if $L_2 = \emptyset$, then $Pref(L_1L_2) = \emptyset$, which is regular by definition.] To prove $Pref(L)$ is regular given that $L$ is regular we define the following base cases:

$$Pref(\emptyset) = \emptyset$$
$$Pref(\epsilon) = \{\epsilon\}$$
$$Pref(a) = \{a\}$$

Let $L_1$ and $L_2$ be regular languages. Then by definition $L_1L_2$ is also regular since regular languages are closed under concatenation. We note the definition of $Pref(L)$ above and letting $L = L_1L_2$ we see that for all $w \in Pref(L_1L_2)$ there exists an $x$ such that $wx \in L_1L_2$ To work toward a proof of regularity of $Pref(L)$ we define the recursive case of concatenation and show $Pref(L_1L_2) = Pref(L_1) \cup (L_1 Pref(L_2))$ by showing both containments.
Since $L_1L_2$ is a concatenation of strings we define arbitrary strings $u$ and $v$ such that $uv = L_1L_2$ where $u \in L_1$ and $v \in L_2$ then $wx$ can be rewritten as $tuv$ where $t = w$ and $uv = x$ noting that $t$ is the prefix of $u$. This gives us two possibilities.
**Case $t = \epsilon$:** If $t$ is the empty string then we have $\epsilon(uv)$ Since $t, u, v$ are arbitrary, we can allow $L_1 = \epsilon$ then $uv \in L_2$ and $u$ is a prefix of $L_2$

$$Pref(L_1L_2) \subseteq L_1 Pref(L_2)$$

**Case $t \neq \epsilon$:** If $t$ is not the empty string then $t$ is a prefix of $u$ and since $u \in L_1$ we have, $tuv \in Pref(L_1)$, or

$$Pref(L_1L_2) \subseteq Pref(L_1)$$

Combining both cases we have,

$$Pref(L_1L_2) \subseteq Pref(L_1) \cup (L_1 Pref(L_2))$$

To rigorously prove the equality we must also show the containment in the other direction where,

$$Pref(L_1) \cup (L_1 Pref(L_2)) \subseteq Pref(L_1 L_2)$$

We let $w \in (Pref(L_1) \cup L_1 Pref(L_2))$ be arbitrary.
**Case** $w \in Pref(L_1)$**:** If $w \in Pref(L_1)$ then there exists a string $m \in \Sigma^*$ such that $wm \in L_1$. Since $m$ was chosen to be any string in $\Sigma^*$ and regular languages are closed under concatenation we can easily let $m = L_2$ and as we have set it up, $wm \in L_1 L_2$ and therefore,

$$w \in Pref(L_1 L_2)$$

**Case** $w \in L_1 Pref(L_2)$**:** Let $w = tuv$ such that $t \in L_1$ and $uv \in L_2$ then we have $tuv \in L_1 L_2$. Since $L_1$ is regular and $\{\epsilon\}$ is regular, we consider the case where $t = \epsilon$ then $tuv = \epsilon(uv) = uv$. Since we defined $uv \in L_2$ $u$ is a prefix of $uv$ We see that $tPref(uv) \equiv L_1 Pref(L_2)$ and therefore,

$$w \in L_1 Pref(L_2)$$

Combining both cases we have,

$$Pref(L_1) \cup (L_1 Pref(L_2)) \subseteq Pref(L_1 L_2)$$

Since we have already shown,

$$Pref(L_1 L_2) \subseteq Pref(L_1) \cup (L_1 Pref(L_2))$$

We conclude through a rigorous proof that,

$$Pref(L_1 L_2) = Pref(L_1) \cup Pref(L_2)$$

[6] (b) Let $L_1$ be any language over $\Sigma$. Give a **rigourous** proof of the equality of languages:

$$Pref(L_1^*) = L_1^* \cup (L_1^* Pref(L_1)) .$$

3. Non-regular languages
   Prove that each of the following languages over $\Sigma = \{0, 1\}$ is not regular.

[4] (a) Let $f(i)$ be the $i$-th entry in the Fibonacci sequence ($f(1) = 1, f(2) = 1, f(i) = f(i-1) + f(i-2)$ when $i > 2$). Then the language is

$$L_a = \left\{1^{f(i)} \mid i \geq 1\right\}.$$

[4] (b) $L_b = \{0^i 1^j \mid i \neq j\}$.

To prove that $L_b$ is not regular using the pumping lemma directly may be impossible. However, by utilizing the closure properties of regular languages and assuming $L_b$ is regular we can use the arithmatic of regular expressions to arrive at a more provable language.

We begin by describing the language $L_b$ which is the language consisting of strings with a certain amound of 0's followed by an **unequal** number of 1's. If we were to use the pumping lemma directly on $L_b$ and break $w \in L_b$ into substrings $xyz$ then "pump" $y$ $k \geq 1$ times we have $w = xy^k z$ but we have no way of knowing if $y$ may be a string such as 01 where-by adding any additional copies of $y$ will add an equal number of 1s and 0s thereby keeping the inequality of $i$ and $j$ and never reaching a contradiction.

We consider the complement of $L_b$, $\overline{L_b}$. As we have assumed $L_b$ to be regular so then must be it's complement since the class of regular languages are closed under complement. Because $L_b$ contains an **unequal** number of 0s and 1s it follows that it's complement will contain an **equal** number of 0s and 1s. However, the complement of $L_b$ will also contain undesired strings where all the 0s are not contained to the first half of the string with the 1s contained on the second. To rectify this we can take the intersection with the regular language defined by the expression $E = 0^*1^*$. Since $E$ is regular by Kleene's Theorem and regular languages are closed under intersection; given our previous assumption that $L_b$ is regular than so must $\overline{L_b} \cap E$.

We describe $\overline{L_b} \cap E = \overline{L_b} \cap (0^*1^*) = L = \{0^n 1^n | n \geq 1\}$ and show it is not regular.
$L$ is in the form $L = \{0^i 1^i | i \geq 0\}$. We will prove through the Pumping Lemma for regular languages that $L$ is not regular.

**Proof:**

- Let $n > 0$ be arbitrary.
- There exists a string $x \in L$ where $|x| \geq n$ and $x = 0^n 1^n$
- The pumping lemma states any for a regular language any large string can be decomposed where $x = uvw$ such that $|uv| \leq n$ and $v \neq \epsilon$. To prove $L$ is non-regular we must show all such decompositions cannot be pumped.
- Notice that for any decomposition of $x = uvw$ $uv = 0^k$ for some $0 < k \leq n$ since the first n characters of $x$ must be 0 by definition and $|uv| \leq n$.
- To argue $uv^*w \not\subset L$, we must show some $i \geq 0$ such that $uv^i w \notin L$.
- Let $i = 0$. We defined $v$ above to be all 0's, so $uv^0 w$ has fewer 0s than 1s. Therefore $uv^0 w \notin L$.
- Thus we have found a string garunteed to be in our language, which we assumed regular, by the pumping lemma and shown that the garunteed string is not present in our language which is a contradiction.
- Therefore $L_b$ is not regular.

4. A Non-Regular Language In Which All Long Words Can Be Pumped
Let $\Sigma = \{a, b, c\}$.

[3] (a) Prove that $L = \{ab^j c^j \mid j \geq 0\}$ is not regular.
Assume $L$ is regular.

We define a regular langage $M$ by the regular expression $b^*c^*$ and note that by Kleene's Theorem $M$ is regular by definition as it is described by a regular expression. Becuase $M$ is regular and we assume $L$ regular we know by the closure properties of regular languages that $L \cap M$ must be regular as well since regular languages are closed under intersection. We define a new regular language $N$ as follows,

$$N = L \cap M = \{b^n c^n | n \geq 0\}$$

By our original assumption of $L$s regularity we know $N$ must be regular. Now, if we are able to show that $N$ is not regular it will follow that $L$ will be deemed non-regular as a contradiction to our original asumption.

**Proof:**

- Let $n > 0$ be arbitrary.
- There exists a string $x \in N$ where $|x| \geq n$ and $x = b^n c^n$
- The pumping lemma states any for a regular language any large string can be decomposed where $x = uvw$ such that $|uv| \leq n$ and $v \neq \epsilon$. To prove $N$ is non-regular we must show all such decompositions cannot be pumped.
- Notice that for any decomposition of $x = uvw$ $uv = b^k$ for some $0 < k \leq n$ since the first n characters of $x$ must be $b$ by definition and $|uv| \leq n$.
- To argue $uv^*w \not\subset N$, we must show some $i \geq 0$ such that $uv^iw \notin N$.
- Let $i = 0$. We defined $v$ above to be all $b$'s, so $uv^0w$ has fewer $b$s than $c$s. Therefore $uv^0w \notin N$.
- Thus we have found a string garunteed to be in our language, which we assumed regular, by the pumping lemma and shown that the garunteed string is not present in our language which is a contradiction.
- Therefore $N$ is not regular and it follows that $L$ is not regularas well.

[3] (b) Prove that $F = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$ is not regular.

- Assume $F$ is regular.
- Let $M = ab^*c^*$ which is regular by Kleene's Theorem.
- Let $N = b^*c^*$ which is also regular by theorem.
- The class of regular languages is closed under intersection so $F \cap M$ must be regular
- Note $F \cap M = \{a^1 b^j c^k | i, j, k \geq 0 \text{ and } i = 1 \rightarrow j = k\}$
- We see that in $F \cap M$ $a^i$ is always $a^1$ and since $i = 1$ then $j = k$ by definition.
- Becuase $F \cap M$ is regular so too must be $(F \cap M) \cap N$ by the property of closure under intersection.
- Then $(F \cap M) \cap N = \{b^j c^k | j = k\}$
- We Define $L = (F \cap M) \cap N = \{b^n c^n | n \geq 0\}$ for convienece.

Now, if we are able to show that $L$ is not regular than by or original assumption of $F$ being regular we will have reached a contradiction and can say with confidence that $F$ does not decribe a regular language.

**Proof:**

- Let $n > 0$ be arbitrary.
- There exists a string $x \in L$ where $|x| \geq n$ and $x = b^n c^n$
- The pumping lemma states any for a regular language any large string can be decomposed where $x = uvw$ such that $|uv| \leq n$ and $v \neq \epsilon$. To prove $L$ is non-regular we must show all such decompositions cannot be pumped.
- Notice that for any decomposition of $x = uvw$ $uv = b^k$ for some $0 < k \leq n$ since the first n characters of $x$ must be $b$ by definition and $|uv| \leq n$.
- To argue $uv^*w \not\subset N$, we must show some $i \geq 0$ such that $uv^i w \notin L$.
- Let $i = 0$. We defined $v$ above to be all $b$'s, so $uv^0 w$ has fewer $b$s than $c$s. Therefore $uv^0 w \notin L$.
- Thus we have found a string garunteed to be in our language, which we assumed regular, by the pumping lemma and shown that the garunteed string is not present in our language which is a contradiction.
- Therefore $L$ is not regular and it follows that $F$ is not regular as well.

[4] (c) Exhibit with proof a choice of a positive integer $n$, such that, for any $z \in F$ with $|z| \geq n$, we may write $z = uvw$ where

- $|uv| \leq n$,
- $|v| \geq 1$ and
- $uv^i w \in F$, for all $i \geq 0$.

Choose $n = 3$ as our value for $n$.

- Let $z = uwv = bc$ where,

$$u = \epsilon$$
$$v = b$$
$$w = cc = c^2$$

- From the definition of $F = \{a^i b^j c^k | i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$ we can see that $z = a^0 b^1 c^2$ and $z \in F$. Furthermore, since $z$ is in the form where $a^i = a^0$ we see that $i \neq 1$ thus, $j$ and $k$ are not bound.
- In our decomposition of $z = uvw$ we see that $|uv| \leq 1 \leq n$, $|v| \geq 1$ and $uvw \in F$.
- To show that for some arbitrary $k > 0$ that $uv^k w \in F$ we can define $uvw$ in terms of $abc$. In our example $uvw = a^0 b^1 c^2$ then $uv^k w = a^0 b^k c^2$.
- As we noted the number of $b$ symbols in the string is not bound by the predicate defining $F$ to equal the number of $c$ symbols since $a^i = a^0$ and $i \neq 1$.
- Therefore any pumping of $uv^k w$ will result in a string of the form $a^0 b^k c^2$ for some $k \geq 0$ and will be accepted by the language defined by $F$.

[2] (d) Explain briefly why the results of parts 4b and 4c do **not** contradict the Pumping Lemma for regular languages.

Part **4B** proves $F = \{a^i b^j c^k | i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$ is not regular but part **4C** gives an example of a decomposition of $F$ under the pumping lemma in which the $F$ can be pumped to generate an infinate number of strings that will be accepted by $F$. This may seem like a contradiction since the pumping lemma gives us a way to reason about the non-regularity of languages. However, this is a pit fall of the pumping lemma, where we must show that all decompositions cannot be pumped to ascertain a languages non-regularity.

The pumping lemma only states that long words in regular languages can be pumped. This does not say that if all long words in a language can be pumped that the language is regular. Some non-regular languages, like the palendrome example from class, can be pumped.

5. More Closure Properties of Regular Languages
Let $\Sigma = \{0, 1\}$ be the alphabet for all languages in this problem. State whether each sentence given below is true or false. If you think the statement is true, then prove it. If you think the statement is false, then provide a counterexample and explain why your counterexample is correct.

[3] (a) If $L_1$ is regular, $L_2$ is non-regular and $L_1 \cap L_2$ is regular, then $L_1 \cup L_2$ is non-regular.
**True**:

- There are only two cases where the intersection of a regular language with a non-regular language produces a regular language Let $L_1$ be an arbitrary regular language, and $L_2$ be an arbitrary non-regular language. Then, $L_1 \cap L_2$ is only regular when,
  - $L_1 = \emptyset$: In this case $\emptyset \cap L_2 = \emptyset$ which is regular. Furthermore $\emptyset \cup L_2 = L_2$ which is non-regular. The statement holds
  - $L_1 = \epsilon$: Similar to the first case $\epsilon \cap L_2 = \emptyset$ but $\epsilon \cup L_2 = L_2$. Again the statement holds true.
- Since we have explored both cases where $L_1 \cap L_2$ produces a regular langage and in both these cases their union produces a non-regular language we can be confident the statement above is true.

[3] (b) If $L_1$ is regular, $L_2$ is non-regular and $L_1 \cap L_2$ is non-regular, then $L_1 \cup L_2$ is non-regular.

**FALSE**:

- Let $L_1 = L(a^*b^*)$ which is regular as it is defined by a regular expression.
- Let $L_2 = \{a^i b^i | i \geq 0\}$ This is the cannonical example of a non-regular language from this assignment, class and the textbook.
- $L_1 \cap L_2 = L_2$ and $L_2$ is not regular.
- But $L_1 \cup L_2 = a^*b^*$ which is regular.