Student ID: 805167986

Collaborators: Sashwath Sundher, David Tran, Dylan Gunn

CS181 Winter 2021 – Problem Set 2

Due Monday, February 1, 11:59 pm

- Please write your student ID **and the names of anyone you collaborated with** in the spaces provided and attach this sheet to the front of your solutions. **Please do not include your name anywhere since the homework will be blind graded.**
- An extra credit of **5%** will be granted to solutions written using LaTeX. Here is one place where you can create LaTeX documents for free: `https://www.overleaf.com/`. The link also has tutorials to get you started. There are several other editors you can use. We have also posted a short LaTeX tutorial on CCLE under References.
- If you are writing solutions by hand, please write your answers in a neat and readable hand-writing.
- Always explain your answers. When a proof is requested, you should provide a rigorous proof.
- If you don't know the answer, write "I don't know" along with a clear explanation of what you tried. For example: "I couldn't figure this out. I think the following is a start, that is correct, but I couldn't figure out what to do next. [[Write down a start to the answer that you are sure makes sense.]] Also, I had the following vague idea, but I couldn't figure out how to make it work. [[Write down vague ideas.]]" At least 20% will be given for such an answer. Note that if you write things that do not make any sense, no points will be given.
- The homework is expected to take anywhere between 10 to 16 hours. You are advised to start early.
- Submit your homework online on Gradescope. The Gradescope code is 5V7GW5.
- Homework points will be scaled according to the number of homework assignments. All assignments will be weighted equally. As per the syllabus, your homework assignments will together comprise 25% of your final grade.

> Note: *All questions in the problem sets are challenging; you should not expect to know how to answer any question before trying to come up with innovative ideas and insights to tackle the question. If you want to do some practice problems before trying the questions on the problem set, we suggest trying problems 1.17 and 1.23 from the book. Do not turn in solutions to problems from the book.*

1. **(20 points)** Let L_1 and L_2 be languages and define

$$\text{shuffle}(L_1, L_2) = \{x_1\, y_1\, x_2\, y_2\, \ldots\, x_n\, y_n \mid x_1\, \ldots\, x_n \in L_1, y_1\, \ldots\, y_n \in L_2\}.$$

For example, if $L_1 = \{1, 23, 45, 678\}$*, and* $L_2 = \{a, b, cd\}$ *then*
$\text{shuffle}(L_1, L_2) = \{1a, 1b, 2c3d, 4c5d\}$.
Hint: For (a) recall closure properties of regular languages.

   (a) **(10 points)** Show that if the language L_1 is not regular and L_2 is any language then the languages $\text{shuffle}(L_1, L_2)$ and $\text{shuffle}(L_1, \overline{L_2})$ cannot both be regular.

Given L_1 is not regular and L_2 is any language.
Suppose $\text{shuffle}(L_1, L_2)$ and $\text{shuffle}(L_1, \overline{L_2})$ are regular.
Let Σ be the alphabet of L_2.

$$\overline{L_2} = \Sigma^* \setminus L_2$$

From lecture we proved that the union of two regular languages is still regular.
Let $L_{shuffle} = \text{shuffle}(L_1, L_2) \cup \text{shuffle}(L_1, \overline{L_2})$

$$L_{shuffle} = \left\{x_1\, y_1\, x_2\, y_2\, \ldots\, x_n\, y_n \mid x_1\, \ldots\, x_n \in L_1, y_1\, \ldots\, y_n \in \left(L_2 \cup \overline{L_2}\right)\right\}$$

$$L_2 \cup \overline{L_2} = \Sigma^*$$

$$L_{shuffle} = \text{shuffle}(L_1, \Sigma^*)$$

From homework 1 we know L_{alt} of any regular language L is also regular.
Where L_{alt} is defined as

$$L_{alt} = \{x \mid \exists y \in L \text{ such that } x_1 x_2 x_3 \ldots = y_1 y_3 y_5 \ldots\}$$

$$L_{shuffle_{alt}} = L_1$$

This leads to our contradiction because proved that $L_{shuffle_{alt}}$ is regular while L_1 is not regular.
$\therefore$ $\text{shuffle}(L_1, L_2)$ and $\text{shuffle}(L_1, \overline{L_2})$ cannot both be regular.

$\square$

(b) **(10 points)** Show that if L_1 and L_2 are regular languages then shuffle(L_1, L_2) is regular.

Given L_1 and L_2 are regular languages.
We want to prove that shuffle(L_1, L_2) is regular.
That means there are two DFAs M_1 and M_2 which solve L_1 and L_2.

$$M_1 = \{Q_1, q_{0_1}, \Sigma_1, \delta_1, F_1\} \text{ and } M_2 = \{Q_2, q_{0_2}, \Sigma_2, \delta_2, F_2\}$$

If shuffle(L_1, L_2) is regular then there exists a DFA $M_{shuffle}$ which solves it.

$$\text{Let } M_{shuffle} = \{Q', q_0', \Sigma', \delta', F'\}$$

Let each state of $M_{shuffle}$ be a tuple with 3 elements.

i. The first element is the state in Q_1 we are in.
ii. The second element is the state in Q_2 we are in.
iii. The third element is 1 when we are about to transition in M_1 and 2 when we are about to transition in M_2.

$$Q' = Q_1 \times Q_2 \times \{1, 2\}$$

The starting state is a tuple of the first state in M_1 and M_2 and we are about to transition in M_1

$$q_0' = (q_{0_1}, q_{0_2}, 1)$$

The alphabet of $M_{shuffle}$ is the union of the alphabet of M_1 and M_2.

$$\Sigma' = \Sigma_1 \cup \Sigma_2$$

The transition function maps in the following way

$$\delta' : (Q_1 \times Q_2 \times \{1, 2\}) \times \Sigma' \rightarrow Q_1 \times Q_2 \times \{1, 2\}$$

The transition function will transition in M_1 when the third element of the tuple is 1 then in M_2 when it is 2.

$$\delta_1'(x, \sigma \in \Sigma') = (a, q_2 \in Q_2, 2) \text{ s.t. } x = (q_1, q_2, 1) \text{ and } a = \delta_1(q_1, \sigma)$$

$$\delta_2'(y, \sigma \in \Sigma') = (q_1 \in Q_1, b, 1) \text{ s.t. } y = (q_1, q_2, 2) \text{ and } b = \delta_2(q_2, \sigma)$$
$$\delta' = \delta_1' \cup \delta_2'$$

The accepting states are where we are at an accepting state of M_1 and M_2 and the third element of our tuple is 0.

$$F' = F_1 \times F_2 \times \{1\}$$

This DFA $M_{shuffle}$ will solve for any string $s \in L_{shuffle}$ where s is comprised of two equal length strings $s_1 \in L_1$ and $s_2 \in L_2$ which are shuffled.
We were able to define a DFA $M_{shuffle}$ which solves the language of shuffle(L_1, L_2).
$\therefore$ shuffle(L_1, L_2) is regular.

$\square$

---

2. (**40 points**) In this problem we investigate the limits of the Pumping Lemma as it was stated in class and look for an alternative that remedies one of these shortcomings.

   (a) **(10 points)** Let L_1 be the language

$$L_1 = \{a^i b^p \mid i \geq 0 \text{ and } p \text{ is a prime}\}.$$

   Prove that the language $L_2 = b^* \cup L_1$ satisfies the conditions of the Pumping Lemma. I.e. show that there exists a $q \in \mathbb{N}$ such that for every word $w \in L_2$ with $|w| \geq q$ we can write $w = xyz$ such that $|xy| \leq q$, $|y| > 0$, and for every $i \geq 0$, $xy^i z \in L_2$.

We want to prove that $L_2 = b^* \cup L_1$ satisfies the conditions of the Pumping Lemma.
Let $q = 5$.
Given a word $w \in L_2$ where $|w| \geq q$.
We can split $w = xyz$ s.t.

$$|xy| \leq q \text{ s.t. } |y| > 0$$

We are going to have two cases where $w \in b^*$ or $w \in L_1$.
First where $w \in b^*$

$$x = bb, y = bb, z = b^{|w|-4}$$

Here $|xy| \leq q$ and $|y| > 0$.
This satisfies the last condition of the Pumping Lemma.

$$\forall i \geq 0 : b^2 b^{2i} b^{|w|-4} \in b^* \subset L_2$$

Second where $w \in L_1$

$$x = \varepsilon, y = a, z = a^{|w|-p-1} b^p$$

Here $|xy| \leq q$ and $|y| > 0$.
This satisfies the last condition of the Pumping Lemma.

$$\forall i \geq 0 : a^i a^{|w|-p-1} \in L_1 \subset L_2$$

Because we found a $q \in \mathbb{N}$ where any $w \in L_2$ s.t. $|w| \geq q$ can be split into $w = xyz$ and satisfy all conditions of the Pumping Lemma, we can say L_2 satisfies the Pumping Lemma.

□

(b) **(20 points)** Prove the following generalization of the Pumping Lemma:
Let L be a regular language. There exists a $q \in \mathbb{N}$ such that for every $w \in L$ and every partition of w into $w = xyz$ with $|y| \geq q$ there are strings a, b, c such that $y = abc$, $|b| > 0$, and for all $i \geq 0$, $xab^icz \in L$.

Given L is a regular language.
Let the DFA $M = \{Q, q_0, \Sigma, \delta, F\}$ solve the language L.
Let q be the number of states in M.

$$q = |Q|$$

Let there be a string $w \in L$ where $w = xyz$ and $|w| \geq q$
Let $|y| \geq q$
Because M accepts w it will have to process the sub string

$$y = y_1 y_2 y_3 \ldots y_{|y|} \text{ of } w = xyz.$$

If $b_1, b_2, b_3, \ldots, b_{|y|+1}$

is the path our machine M take when processing y then

$$|y| + 1 > q$$

and by the **pigeonhole principle** there exists i and j where

$$i \neq j \text{ and } b_i = b_j$$

which means there is a loop.
We can break apart y into 3 sub strings a, b, c.

$$a = y_1 \ldots y_i$$

$$b = y_{i+1} \ldots y_j$$

$$c = y_{j+1} \ldots y_{|y|}$$

The sub string b will the part of y which has the loop.
Because $j \neq i + 1$, $|b| > 0$ and there cannot be an empty loop.
This gives us

$$\forall i \geq 0 : xab^icz \in L$$

Because you cannot skip the loop in b or go through it more than once we have shown that there is a $q \in \mathbb{N}$ s.t. $\forall$ words and partitions of $w \in L$, $w = xyz$ satisfies the generalization of the Pumping Lemma.

□

(c) **(10 points)** Prove that the language L_2 is not regular.

Let L_2 be regular.
Let $w = ab^p \in L_2$.
Using the generalization of the Pumping Lemma we can say that there exists a

$$q \in \mathbb{N}$$

where p is the first prime greater than or equal to q.

If $w = xyz$ we choose the partition of w below because the generalized Pumping Lemma must account for every partition xyz of w.

$$x = a,\ y = b^q,\ z = b^{p-q}$$

Using the generalized Pumping Lemma there exists strings

$$y_1, y_2, y_3 \in \Sigma^*$$

s.t. $y = y_1 y_2 y_3$ and $|y_2| > 0$

We define y_1, y_2, y_3 to be

$$y_1 = b^j,\ y_2 = b^k \text{ s.t. } k > 0,\ y_3 = b^{q-j-k}$$

$$\forall i \geq 0 : x\ y_1 y_2^i y_3\ z = a\ b^j b^{ki} b^{q-j-k}\ b^{p-q} = ab^{p+k(i-1)} \in L_2$$

Let $i = p + 1$

$$ab^{p+k(p+1-1)} = ab^{p+kp} = ab^{p(1+k)} \in L_2$$

This is a contradiction because $p(1+k)$ is not prime because it is the product of p and $1+k$.
$\therefore L_2$ is not regular.

$\square$

---

3. (**40 points**) For a language L over alphabet Σ, we define

$$L_{\frac{1}{3}-\frac{1}{3}} = \{xz \in \Sigma^* \mid \exists y \in \Sigma^* \text{ with } |x| = |y| = |z| \text{ such that } xyz \in L\}.$$

For example, if $L = \{a, to, cat, math, solve, theory\}$, *then* $L_{\frac{1}{3}-\frac{1}{3}} = \{ct, thry\}$.
Prove that if L is regular, then $L_{\frac{1}{3}-\frac{1}{3}}$ need not be regular.
Hint: Consider the language 0^*21^* *and recall closure properties of regular languages*

Looking at the language $L = 0^*21^*$ we can say it is trivially regular because it is the concatenation of 3 regular languages $0^*, 2, 1^*$ which is part of the closure properties we saw during lecture.

For our language $L = 0^*21^*$ we have 3 ways to partition it into x,y,z s.t. $xyz \in L$ and xy is accepted by $L_{\frac{1}{3}-\frac{1}{3}}$.

Let $a = i + j + 1$ and $i, j \geq 0$

(a)

$$x = 0^i21^j,\ y = 1^a,\ z = 1^a$$

$$L_1 = \{0^i21^j\ 1^a\}$$

$$\text{Where } L_1 \subset L_{\frac{1}{3}-\frac{1}{3}}.$$

(b)

$$x = 0^a,\ y = 0^i21^j,\ z = 1^a$$

$$L_2 = \{0^a\ 1^a\} = \{0^b1^b \mid b \geq 1\}$$

$$\text{Where } L_2 \subset L_{\frac{1}{3}-\frac{1}{3}}.$$

(c)

$$x = 0^a,\ y = 0^a,\ z = 0^i21^j$$

$$L_3 = \{0^a\ 0^i21^j\}$$

$$\text{Where } L_3 \subset L_{\frac{1}{3}-\frac{1}{3}}.$$

$$L_{\frac{1}{3}-\frac{1}{3}} = L_1 \cup L_2 \cup L_3$$

From the closure properties if we intersect two regular languages it should output a regular language, but here we intersect $L_{\frac{1}{3}-\frac{1}{3}}$ with a regular language $\{0,1\}^*$ is equal to L_2 which is not regular. From class we proved that $L_2 = \{0^b1^b \mid b \geq 1\}$ is not regular as we have seen in class.

$$L_{\frac{1}{3}-\frac{1}{3}} \cap \{0,1\}^* = L_2 = \{0^b1^b \mid b \geq 1\}$$

From intersection closure we can see that $L_{\frac{1}{3}-\frac{1}{3}}$ does not have to be regular even if L is regular. □
