# Problem Set 3

Due Date ........ September 20, 2022
Name ........ **Alex Ojemann**
Student ID ........ **109722375**
Collaborators ........ **None**

## Contents


**1 Instructions** **1**

**2 Honor Code (Make Sure to Virtually Sign)** **3**

**3 Standard 8: Pumping Lemma & Closure Properties** **4**
Problem 1 . . . . 4
3.1 Pumping Lemma . . . . 4
3.2 Closure Properties . . . . 5

**4 Standard 10: Classification of Regular Languages** **6**
Problem 2 . . . . 6

**5 Standard 9: DFA Minimization** **8**
Problem 3 . . . . 8
5.1 Part (a) . . . . 8
5.2 Part (b) . . . . 9
5.3 Part (c) . . . . 10

**6 Standards 11 and 2/3: Novel Construction + Proof** **11**
Problem 4 . . . . 11
6.1 Standard 11: Novel Construction . . . . 11
6.2 Standards 2/3: Proofs . . . . 12


## 1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. (See this short intro to LaTeX plus other resources on Canvas.)
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.

- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

## 2 Honor Code (Make Sure to Virtually Sign)

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

*(I agree to the above, Alex Ojemann).* $\square$

# 3 Standard 8: Pumping Lemma & Closure Properties

**Problem 1.** Let BRACKET be the language of strings of balanced brackets. For example, [[[][]][]] is in BRACKET, but the string ][[] is not in BRACKET.
Do the following.

## 3.1 Pumping Lemma

(a) Prove that BRACKET is not regular using the Pumping Lemma. Spell out each step, especially your choice of $x, y, z$ (which depend on $k$) and $i$ (which may depend on $k$ and $u, v, w$). [**Note:** You must use and are responsible for knowing the version of the Pumping Lemma in Kozen, which is quite different than the version found in different texts and online resources.]

*Proof.* Assume Bracket is regular.

k given

Let $x =, y = [^k, z = ]^k$

$y = uvw$ and $|v| > 0$,so v must contain at least one [

Take i = 0 :

$xuv^iwz = xuwz = [^{k-|v|}]^k$

$|v| > 0$, so $[^{k-|v|}]^k$ isn't in BRACKET.

Therefore, BRACKET is not regular by the pumping lemma. □

### 3.2 Closure Properties

(b) Prove that BRACKET is not regular using closure properties. Here you can use any closure property of regular languages proven in the book / discussed in class, such as concatenation, union, intersection, asterate, reverse, image or inverse image of homomorphism, etc. I recommend spelling out the logic. For example, to justify why the set $D$ at the bottom of page 73 is nonregular, you could write:

> Suppose the given language $D$ were regular. Then $A = D \cap L(a^*b^*)$ would be regular. But $D \cap L(a^*b^*) = \{a^n b^n | n \geq 0\}$, so $A$ is not regular, a contradiction. Thus, $D$ must be non-regular.

We would also accept a more terse version:

> $D \cap L(a^*b^*) = \{a^n b^n | n \geq 0\}$, which is not regular, so neither is $D$.

But tread carefully with the terse version, since it's easy to get the logic backwards!

*Proof.* Let $A = [^*]^*$

A is regular because it's of the form $a^*b^*$.

Suppose BRACKET is regular. Therefore, $A\cap$ BRACKET should be regular as well.

However, $A\cap$ BRACKET $= [^n]^n | n \geq 0$, which is not regular because it's of the form $a^n b^n$.

Therefore, BRACKET is not regular. □

# 4 Standard 10: Classification of Regular Languages

**Problem 2.** For each of the following languages, decide whether the language is regular. For the justification, when showing a language is regular you can demonstrate using the usual constructions (DFA,NFA,regexp) and do not need to prove that your construction works. You could also use closure properties and known regular languages. (Here "known" means anything that we proved in class or is proved in the book.) To justify the claim that a language is non-regular, you can use either the pumping lemma or closure properties.

(a) $L = \{a^n b^{2m} : m, n \geq 0\}$

*Proof.* $L = a^*(bb)^*$ □

(b) $L = \{a^{p-1} : p \text{ is prime }\}$. [**Note:** A similar problem ME.36 is solved on page 363 of Kozen.]

*Proof.* Assume L is regular.

k given

Let $x = \epsilon, y = a^k, z = \epsilon$

With $y = uvw$, let $u = a^l, v = a^m, w = a^n (l + m + n = k)$.

$xuv^iwz = a^l a^{mi} a^n = a^{l+mi+n} = a^{k+m(i-1)} = a^{p-1+m(i-1)} = a^{(p+m(i-1))-1}$

Take i = p + 1:

$(p + m(i - 1)) = (p + m(p + 1 - 1)) = p + mp = p(m + 1)$

p(m + 1) is not prime, so L is not regular. □

(c) $L = \{a^n b^{n+3434} : n \geq 0\}$

*Proof.* Assume L is regular

k given

$x = a^k, y = b^{k+3434}, z = \epsilon$

$y = uvw, |v| > 0, v = b^{|v|}$

take i = 2:

$xuv^iwz = xuvvwz = a^k b^{k+3434+|v|}$

There are now more than 3434 more bs than as so this string is not in the language so this language is non-regular by the pumping lemma □

(d) $L = \{a^n b^m : n \geq m \text{ and } m \leq 3434\}$.

*Proof.* Let $x \cup y = b*, x = b^p$ where $p \leq 3434$ and $y = b^q$ where $q \geq 3434$

Since $x \cup y$ is regular, x must also be regular by the union closure property.

Let $v \cup w = a*, v = a^r$ where $r \leq m$ (and $m \leq 3434$ as defined) and $w = a^s$ where $s \geq m$

Since $v \cup w$ is regular, w must also be regular by the union closure property.

L can be expressed as wx, which is regular by the concatenation closure property. □

(e) $L = \{a^n b^m : n \geq m \text{ and } m \geq 3434\}$.

*Proof.* Assume L is regular

k given

$x = a^{3434}, y = a^k, z = b^k b^{3434}$

$y = uvw, |v| > 0, v = a^{|v|}$

take i = 0:

$xuv^iwz = xuwz = a^{3434}a^{k-|v|}b^kb^{3434}$

n is now less than m so this string is not in the language so this language is non-regular by the pumping lemma □

# 5 Standard 9: DFA Minimization

**Problem 3.** Consider the following DFA.

| | a | b |
|---|---|---|
| $\rightarrow 1F$ | 3 | 5 |
| 2F | 8 | 7 |
| 3 | 7 | 2 |
| 4 | 6 | 2 |
| 5 | 1 | 8 |
| 6 | 2 | 3 |
| 7 | 1 | 4 |
| 8 | 5 | 1 |

Do the following.

## 5.1 Part (a)

(a) Determine which states are accessible and which are not.

*Answer.*

- **Accessible:** 1, 3, 5, 7, 2, 8, 4, 6
- **Not Accessible:** None

□

## 5.2 Part (b)

(b) List the equivalence classes of the collapsing relation $\approx$ defined in class and in Lecture 13 of Kozen:

$$p \approx q \iff \forall x \in \Sigma^* \, (\hat{\delta}(p, x) \in F \iff \hat{\delta}(q, x) \in F);$$

Please list the classes like "$1 \approx 2 \approx 3$; $4 \approx 5$; 6; $7 \approx 8$." Furthermore, please clearly justify why states are inequivalent.

*Answer.* $1 \approx 2$; $3 \approx 4 \approx 8$; $5 \approx 6 \approx 7$.

1 and 2 are equivalent because they're both final states and transition to one of 3, 4, or 8 with a and one of 5, 6, or 7 with b. 3, 4, and 8 are equivalent because they transition to one of 5, 6, or 7 with a and one of 1 or 2 with b. 5, 6, and 7 are equivalent because they transition to one of 1 or 2 with a and one of 3, 4, or 8 with b. □

## 5.3 Part (c)

(c) Give the automaton obtained by collapsing equivalent states and removing inaccessible states.

*Answer.*

| | a | b |
|---|---|---|
| 12F | 38 | 57 |
| 348 | 567 | 12 |
| 567 | 12 | 348 |

□

# 6 Standards 11 and 2/3: Novel Construction + Proof

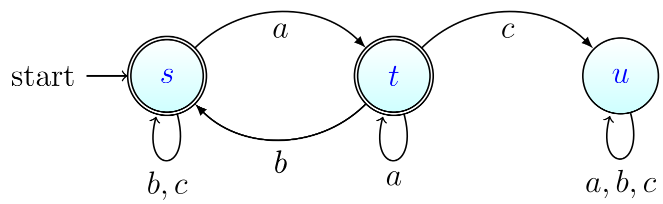
**Problem 4.** Let $x$ be a string, and fix $a, b, c \in \Sigma$. Define $\text{ABC}(x)$ to replace each instance of $ac$ in $x$ with $abc$. Now define:

$$\text{ABC}(L) := \{\text{ABC}(x) : x \in L\}.$$

## 6.1 Standard 11: Novel Construction

(a) Suppose that $L$ is regular. Give a construction (e.g., regexp, DFA/NFA) to justify that $\text{ABC}(L)$ is regular. You need not prove that your construction works to get credit for Standard 11.

*Answer.*




□

### 6.2 Standards 2/3: Proofs

(b) Carefully prove that your construction from part (a) works. That is, again suppose $L$ is regular, and let $K$ be the language accepted by your construction. Carefully prove that $K = \text{ABC}(L)$.

*Proof.* ABC(L) should take any string in L and turn all instances of 'ac' into 'abc', so the result must have no instances of 'ac'. If there is an instance of 'ac', the dfa will transition from state s to t to u and be forever stuck in u which is a non accepting state. Any other string in the language will be accepted. $\square$