HW3CSE105F24: Homework assignment 3 solution

CSE105F24

Due: October 22nd at 5pm, via Gradescope

In this assignment,

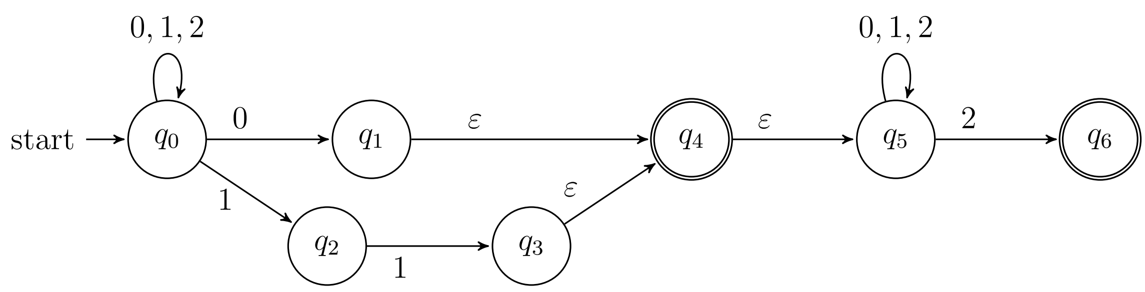
You will demonstrate the richness of the class of regular languages, as well as its boundaries.

Resources: To review the topics for this assignment, see the class material from Week 3. We will post frequently asked questions and our answers to them in a pinned Piazza post.

Reading and extra practice problems: Sipser Chapter 1. Chapter 1 exercises 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 1.14, 1.15, 1.16, 1.17, 1.19, 1.20, 1.21, 1.22. Chapter 1 problem 1.51.

Assigned questions

1. **Using general constructions** (16 points): Consider the NFA N over $\{0, 1, 2\}$ with state diagram



(a) (*Graded for completeness*) [1] Give two examples of strings of length greater than 2 that are accepted by N and two examples of strings of length greater than 2 that are rejected by N.

[1]This means you will get full credit so long as your submission demonstrates honest effort to answer the question. You will not be penalized for incorrect answers. To demonstrate your honest effort in answering the question, we expect you to include your attempt to answer *each* part of the question. If you get stuck with your attempt, you can still demonstrate your effort by explaining where you got stuck and what you did to try to get unstuck.

For each example string, list at least one of the computations of N on this string and label whether this computation witnesses that the string is accepted by N.

Solution:

String 000 is accepted by N. An example computation starts from q_0, reads 0 and stays in q_0, reads another 0 and stays in q_0, reads the last 0 and transit to q_1, and then spontaneously transit to q_4 and the computation ends. This computation witnesses that the string is accepted by N since it fully processes the input string and ends in an accept state.

String 0012 is accepted by N. An example computation starts from q_0, reads 0 and transit to q_1, spontaneously move to q_4, then spontaneously move to q_5, reads another 0 and stays in q_5, reads 1 and stays in q_5, reads 2 and transit to q_6 and the computation ends. This computation witnesses that the string is accepted by N since it fully processes the input string and ends in an accept state.

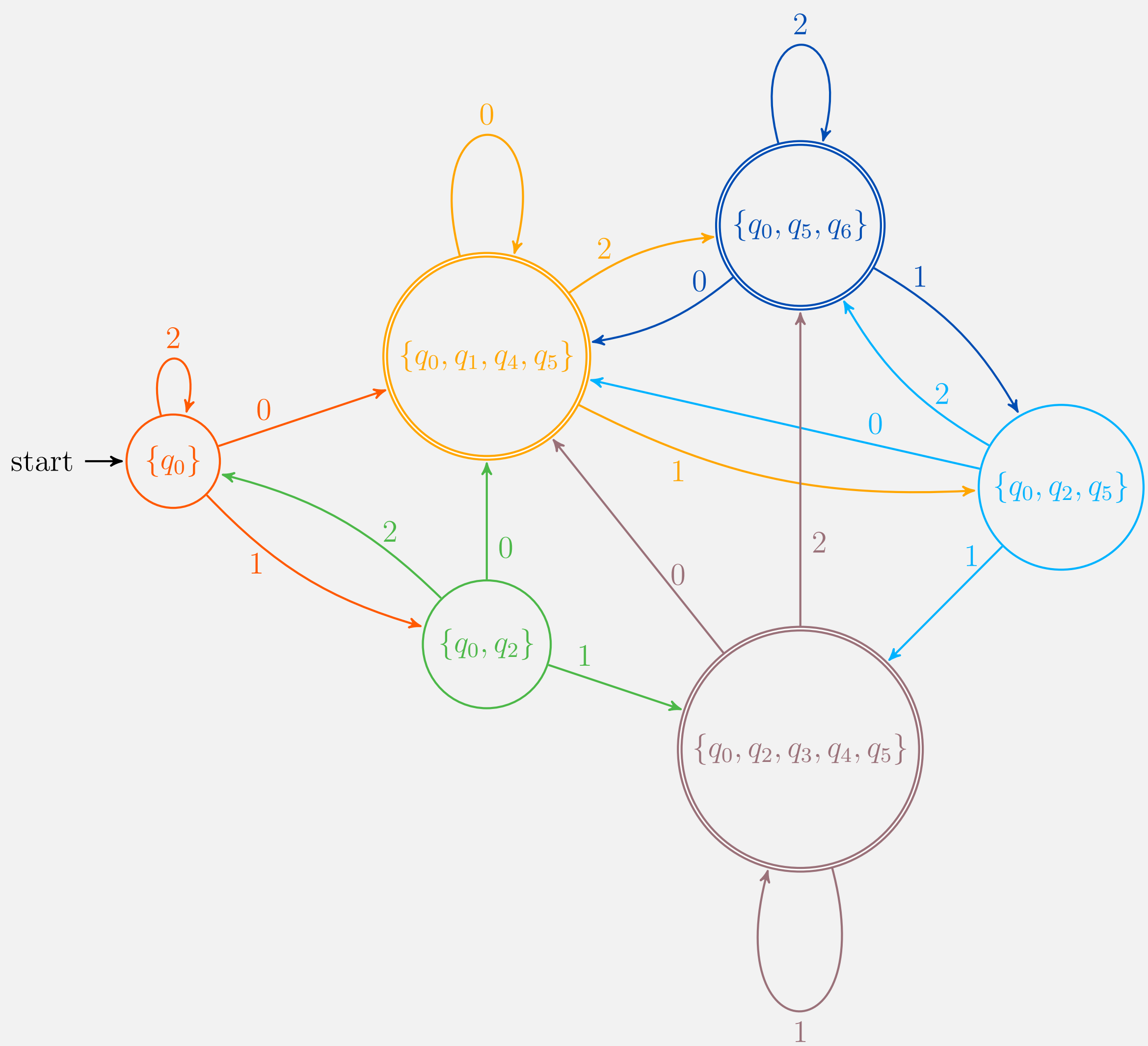
String 222 is rejected by N. An example computation starts from q_0, reads 2 and stays in q_0, reads another 2 and stays in q_0, reads the last 2 and stays in q_0, and the computation ends. This computation does not witnesses that the string is accepted by N since it fully processes the input string but does not end in an accept state. Note that all other computations of this machine on 222 also do not witness that the string is accepted, which is why the string is rejected by N.

String 101 is rejected by N. An example computation starts from q_0, reads 1 and transit to q_2. Then the computation stuck since there is not outgoing arrow labeled 0 from q_2. This computation does not witnesses that the string is accepted by N since it does not fully process the input string. Note that all other computations of this machine on 222 also do not witness that the string is accepted, which is why the string is rejected by N.

(b) (*Graded for correctness*) [2] Use the "macro-state" construction from Theorem 1.39 and class to create the DFA M recognizing the same language as N. You only need to include states that are reachable from the start state. For full credit, submit (1) a state diagram that is deterministic (there should be arrows labelled 0, 1, and 2 coming out of each state) and where each state is labelled by a subset of the states in N; and (2) for one of your example strings that is accepted by N, give the computation of M on this string as a seqeuence of states visited; and (3) for one of your example strings that is rejected by N, give the computation of M on this string as a seqeuence of states visited.

[2] This means your solution will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

Solution:



Computation on 000 (accepted):
$\{q_0\} \xrightarrow{0} \{q_0, q_1, q_4, q_5\} \xrightarrow{0} \{q_0, q_1, q_4, q_5\} \xrightarrow{0} \{q_0, q_1, q_4, q_5\}$

Computation on 101 (rejected):
$\{q_0\} \xrightarrow{1} \{q_0, q_2\} \xrightarrow{0} \{q_0, q_1, q_4, q_5\} \xrightarrow{1} \{q_0, q_2, q_5\}$

(c) (*Graded for completeness*) Give a mathematical description either using set builder notation or a regular expression for $L(N)$ and for $L(M)$.

Solution: Let $\Sigma = \{0, 1, 2\}$. The regular expression $\Sigma^*(0 \cup 11)(\varepsilon \cup \Sigma^* 2)$ can describe both $L(N)$ and $L(M)$ (note that $L(N) = L(M)$).

2. **Multiple representations** (12 points):

(a) Consider the language $A_1 = \{uw \mid u \text{ and } w \text{ are strings over } \{0,1\} \text{ and have the same length}\}$ and the following argument.

> "Proof" that A_1 is not regular using the Pumping Lemma: Let p be an arbitrary positive integer. We will show that p is not a pumping length for A_1.
> Choose s to be the string 1^p0^p, which is in A_1 because we can choose $u = 1^p$ and $w = 0^p$ which each have length p. Since s is in A_1 and has length greater than or equal to p, if p were to be a pumping length for A_1, s ought to be pump'able. That is, there should be a way of dividing s into parts x, y, z where $s = xyz$, $|y| > 0$, $|xy| \leq p$, and for each $i \geq 0$, $xy^iz \in A_1$. Suppose x, y, z are such that $s = xyz$, $|y| > 0$ and $|xy| \leq p$. Since the first p letters of s are all 1 and $|xy| \leq p$, we know that x and y are made up of all 1s. If we let $i = 2$, we get a string xy^iz that is not in A_1 because repeating y twice adds 1s to u but not to w, and strings in A_1 are required to have u and w be the same length. Thus, s is not pumpable (even though it should have been if p were to be a pumping length) and so p is not a pumping length for A_1. Since p was arbitrary, we have demonstrated that A_1 has no pumping length. By the Pumping Lemma, this implies that A_1 is nonregular.

i. (*Graded for completeness*) Find the (first and/or most significant) logical error in the "proof" above and describe why it's wrong.

> **Solution:** The part that "If we let $i = 2$, we get a string xy^iz that is not in A_1" is incorrect because the new string can still have even length, in which case we can reassign u and w to be the new first half and second half so we can see that the new string is still in A_1. For example, when we choose integer $p = 2$, then for any string s in A_1 of length at least 2, s can be divided into three pieces $s = xyz$ where we let $x = \varepsilon$ and let y be the first two characters of s which satisfies $|xy| \leq p$. In this case, for each $i \geq 0$, $xy^iz \in A_1$ since xyz is even length and y is also even length, which makes xy^iz also even length.

ii. (*Graded for completeness*) Prove that the set A_1 is actually regular (by finding a regular expression that describes it or a DFA/NFA that recognizes it, and justifying why) **or** fix the proof so that it is logically sound.

> **Solution:** Based on the definition of A_1, we can see that A_1 is the set of strings over $\{0,1\}$ with even lengths. Thus the regular expression $((0 \cup 1)(0 \cup 1))^*$ can describe A_1, therefore A_1 is regular.

(b) Consider the language $A_2 = \{u1w \mid u \text{ and } w \text{ are strings over } \{0,1\} \text{ and have the same length}\}$ and the following argument.

> "Proof" that A_2 is not regular using the Pumping Lemma: Let p be an arbitrary positive integer. We will show that p is not a pumping length for A_2.

Choose s to be the string $1^{p+1}0^p$, which is in A_2 because we can choose $u = 1^p$ and $w = 0^p$ which each have length p. Since s is in A_2 and has length greater than or equal to p, if p were to be a pumping length for A_2, s ought to be pump'able. That is, there should be a way of dividing s into parts x, y, z where $s = xyz$, $|y| > 0$, $|xy| \leq p$, and for each $i \geq 0$, $xy^iz \in A_2$. When $x = \varepsilon$ and $y = 1^{p+1}$ and $z = 0^p$, we have satisfied that $s = xyz$, $|y| > 0$ (because p is positive) and $|xy| \leq p$. If we let $i = 0$, we get the string $xy^iz = 0^p$ that is not in A_2 because its middle symbol is a 0, not a 1. Thus, s is not pumpable (even though it should have been if p were to be a pumping length) and so p is not a pumping length for A_2. Since p was arbitrary, we have demonstrated that A_2 has no pumping length. By the Pumping Lemma, this implies that A_2 is nonregular.

i. (*Graded for completeness*) Find the (first and/or most significant) logical error in the "proof" above and describe why it's wrong.

> **Solution:** The part that "When $x = \varepsilon$ and $y = 1^{p+1}$ and $z = 0^p$, we have satisfied that $s = xyz$, $|y| > 0$ (because p is positive) and $|xy| \leq p$" is incorrect. Firstly, here by the choices, $|xy| = p + 1 > p$ so it does not satisfy $|xy| \leq p$. But the more significant problem is that we want to prove that there is no way to split $s = xyz$ such that $|y| > 0$, $|xy| \leq p$, and for each $i \geq 0$, $xy^iz \in A_2$, therefore we should not only prove that one particular choice of x, y, z does not work. Instead, we should prove that for all possible splits $s = xyz$ s.t. $|y| > 0$, $|xy| \leq p$, there exists some $i \geq 0$, where $xy^iz \notin A_2$.

ii. (*Graded for completeness*) Prove that the set A_2 is actually regular (by finding a regular expression that describes it or a DFA/NFA that recognizes it, and justifying why) **or** fix the proof so that it is logically sound.

> **Solution:** A_2 is non-regular. We can fix the proof:
>
> Let p be an arbitrary positive integer. We will show that p is not a pumping length for A_2. Choose s to be the string $1^{p+1}0^p$, which is in A_2 because we can choose $u = 1^p$ and $w = 0^p$ which each have length p. Since s is in A_2 and has length greater than or equal to p, if p were to be a pumping length for A_2, s ought to be pump'able. That is, there should be a way of dividing s into parts x, y, z where $s = xyz$, $|y| > 0$, $|xy| \leq p$, and for each $i \geq 0$, $xy^iz \in A_2$. **Since the first p letters of s are all 1 and $|xy| \leq p$, we know that x and y are made up of all 1s. If we let $i = 0$, we get the string xy^iz that is not in A_2 because given $|y| > 0$, either the new string becomes even length which do not have a middle symbol, or the new string has odd length but the middle symbol is a 0, not a 1.** Thus, s is not pumpable (even though it should have been if p were to be a pumping length) and so p is not a pumping length for A_2. Since p was arbitrary, we have demonstrated that A_2 has no pumping length. By

the Pumping Lemma, this implies that A_2 is nonregular.

3. **Pumping** (10 points):

(a) (*Graded for correctness*) Give an example of a language over the alphabet $\{a, b\}$ that has cardinality 5 and for which 4 is a pumping length and 3 is not a pumping length. Is this language regular? A complete solution will give (1) a clear and precise description of the language, (2) a justification for why 4 is a pumping length, (3) a justification for why 3 is not a pumping length, (4) a correct and justified answer to whether the language is regular.

Solution: Consider the language $L = \{aaa, aab, aba, abb, baa\}$. This language has exactly five distinct elements.

We will prove that 4 is a pumping length of L: To prove this, we want to show that for each string $s \in L$ and $|s| \geq 4$, there are strings x, y, z such that $s = xyz$ satisfies for each $i \geq 0$, $xy^iz \in L$; $|y| > 0$; and $|xy| \leq 4$. All strings in L have length 3. Thus, there is no string in L with length at least 4. So the statement above is vacuously true. Therefore, 4 is a pumping length of L.

We will now prove that 3 is not a pumping length of L: To prove this, we want to find a counterexample string $s \in L, |s| \geq 3$ such that for any x, y, z where $s = xyz$, $|y| > 0$, and $|xy| \leq 3$, there exists $i \geq 0$ such that $xy^iz \notin L$. Take $s = aaa \in L$ as a counterexample which is a string of length 3. Consider any x, y, z where $s = xyz$, $|y| > 0$, and $|xy| \leq 3$. Let $i = 0$ so $xy^iz = xz$. Since $|y| > 0$, we have $|xy^iz| = |xz| < 3$. However, all strings in L have length 3, therefore the string that we get when $i = 0$, xz, is not in L. Thus we proved that 3 is not a pumping length of L.

The language L is regular, since it can be described by the regular expression $aaa \cup aab \cup aba \cup abb \cup baa$. (This also follows from the observation that every finite language is regular.)

(b) (*Graded for completeness*) In class and in the reading so far, we've seen the following examples of nonregular sets:

$\{0^n1^n \mid n \geq 0\}$	$\{0^n1^m \mid 0 \leq m \leq n\}$	$\{0^n1^m0^n \mid n, m \geq 0\}$
$\{0^n1^n \mid n \geq 2\}$	$\{0^i1^{2i} \mid 0 \leq i\}$	$\{w \in \{0,1\}^* \mid w = w^R\}$
$\{0^n1^m \mid 0 \leq n \leq m\}$	$\{0^i1^{i+1} \mid 0 \leq i\}$	$\{ww^R \mid w \in \{0,1\}^*\}$

Modify one of these sets in some way and use the Pumping Lemma to prove that the resulting set is still nonregular.

Solution: Consider the language $A = \{0^n 1^{3n} \mid n \geq 0\}$. We will prove that this language is nonregular using the Pumping Lemma.

Let p be an arbitrary positive integer. We will show that p is not a pumping length for A. Choose s to be the string $0^p 1^{3p}$. Since s is in A and has length greater than or equal to p, if p were to be a pumping length for A, s ought to be pump'able. That is, there should be a way of dividing s into parts x, y, z where $s = xyz$, $|y| > 0$, $|xy| \leq p$, and for each $i \geq 0$, $xy^i z \in A$. Since the first p letters of s are all 0 and $|xy| \leq p$, we know that x and y are made up of all 0s. Denote $x = 0^m, y = 0^n, z = 0^{p-m-n}1^{3p}$, where $m \geq 0, n > 0$. If we let $i = 0$, we get the string $xy^i z = xz = 0^{p-n}1^{3p}$ that is not in A because given $|y| > 0, n > 0$, we have $3p > 3(p-n)$. Thus, s is not pumpable (even though it should have been if p were to be a pumping length) and so p is not a pumping length for A. Since p was arbitrary, we have demonstrated that A has no pumping length. By the Pumping Lemma, this implies that A is nonregular.

4. **Regular and nonregular languages** (12 points): In Week 2's review quiz, we saw the definition that a set X is said to be **closed under an operation** if, for any elements in X, applying to them gives an element in X. For example, the set of integers is closed under multiplication because if we take any two integers, their product is also an integer .

Prove or disprove each closure claim statement below about the class of regular languages and the class of nonregular languages. Your arguments may refer to theorems proved in the textbook and class, and if they do, should include specific page numbers and references (i.e. write out the claim that was proved in the book and/or class).

Recall the definitions we have:

For language L over the alphabet $\Sigma_1 = \{0, 1\}$, we have the associated sets of strings

$$SUBSTRING(L) = \{w \in \Sigma_1^* \mid \text{there exist } a, b \in \Sigma_1^* \text{ such that } awb \in L\}$$

and

$$EXTEND(L) = \{w \in \Sigma_1^* \mid w = uv \text{ for some strings } u \in L \text{ and } v \in \Sigma_1^*\}$$

(a) (*Graded for completeness*) The set of regular languages over $\{0, 1\}$ is closed under the $SUBSTRING$ operation.

Solution: The statement is correct. Consider any regular language A over $\Sigma_1 = \{0, 1\}$. A can be recognized by some NFA $N = (Q, \Sigma_1, \delta, q_0, F)$ (week 3 note page 6: "A language is regular when there is an NFA that recognizes it"). We can build a new NFA $N_{new} = (Q \cup \{q_{new}\}, \Sigma_1, \delta_{new}, q_{new}, F_{new})$ where $q_{new} \notin Q$ and $F_{new} = \{q \in$

$Q \mid \exists w \in \Sigma_1^*(\delta^*((q,w)) \in F)\}$, and

$$\delta_{new}((q,a)) = \begin{cases} \delta((q,a)) & \text{if } q \in Q,\, a \in \Sigma_\varepsilon \\ \{q' \in Q \mid \exists w \in \Sigma_1^*(\delta^*((q_0,w)) = q')\} & \text{if } q = q_{new},\, a = \varepsilon \\ \emptyset & \text{if } q = q_{new},\, a \in \Sigma_1 \end{cases}$$

There are spontaneous moves from the new start state to any state reachable in the original NFA from the original start state. The new set of accepting states is the collection of states in the original NFA from which an accepting state (in the original NFA) is reachable. With these definitions, N_{new} accepts all strings that are each a substring of some string in A, and rejects all other strings. Therefore, N_{new} recognizes $SUBSTRING(A)$, which means that $SUBSTRING(A)$ is also regular (week 3 note page 6: "A language is regular when there is an NFA that recognizes it"). Therefore, the set of regular languages over $\{0,1\}$ is closed under the $SUBSTRING$ operation.

(b) (*Graded for completeness*) The set of nonregular languages over $\{0,1\}$ is closed under the $SUBSTRING$ operation.

Solution: The statement is incorrect. Consider a counterexample language $L = \{0^n1^n \mid n \geq 0\}$ which is a nonregular language (week 3 note page 12). We get that $SUBSTRING(L) = \{0^m1^n \mid m \geq 0, n \geq 0\}$ because by definition, $SUBSTRING(L)$ consists of all strings that are each a substring of some string in L. Thus $SUBSTRING(L)$ can be described by regular expression 0^*1^*, which means that $SUBSTRING(L)$ is a regular language (Sipser, Theorem 1.54: "A language is regular if and only if some regular expression describes it"). Therefore, the set of nonregular languages over $\{0,1\}$ is not closed under the $SUBSTRING$ operation.

(c) (*Graded for correctness*) The set of regular languages over $\{0,1\}$ is closed under the $EXTEND$ operation.

Solution: The statement is correct. Each regular language A can be recognized by some NFA N (week 3 note page 6: "A language is regular when there is an NFA that recognizes it"), i.e. $A = L(N)$. We also proved that for any NFA N we can construct an NFA N_{new} such that $L(N_{new}) = EXTEND(L(N))$ (HW2 3(b)). Therefore, for each regular language A, there exists an NFA that can recognize $EXTEND(A)$, which means that $EXTEND(A)$ is also a regular language (week 3 note page 6: "A language is regular when there is an NFA that recognizes it"). Thus the set of regular languages over $\{0,1\}$ is closed under the $EXTEND$ operation.

(d) (*Graded for correctness*) The set of nonregular languages over $\{0,1\}$ is closed under the $EXTEND$ operation.

Solution: The statement is incorrect. Consider a counterexample language $L = \{0^n 1^n \mid n \geq 0\}$ which is a nonregular language (week 3 note page 12). We get that $EXTEND(L) = \{0,1\}^*$ since by definition, $EXTEND(L)$ consists of all strings that are each the result of taking a string in L and concatenating it with a string over $\{0,1\}$. Since $\varepsilon \in L$ (setting $n = 0$), we can get any string over $\{0,1\}$ in $EXTEND(L)$ (as the result of concatenating ε with this string). Thus $EXTEND(L)$ can be described by regular expression Σ_1^*, which means that $EXTEND(L)$ is a regular language (Sipser, Theorem 1.54: "A language is regular if and only if some regular expression describes it"). Therefore, the set of nonregular languages over $\{0,1\}$ is not closed under the $EXTEND$ operation.