HW3 : Nonregular Languages and Pushdown Automata

CSE105Sp22

In this assignment,

You practiced distinguishing between regular and nonregular languages using both closure arguments and the pumping lemma. You also practiced with the definition of pushdown automata.

Reading and extra practice problems: Sipser Section 1.4, 2.2. Chapter 1 exercises 1.29, 1.30. Chapter 1 problems 1.49, 1.50, 1.51. Chapter 2 exercises 2.5, 2.7.

Assigned questions

1. (*Graded for fair effort completeness*[1]) Do the following for each of the following attempted "proofs" that a set is nonregular:

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

   ii Either prove that the set 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.

   (a) The language $X_1 = \{uw \mid u \text{ and } w \text{ are strings over } \{0,1\} \text{ and have the same length}\}$.

   > "Proof" that X_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 X_1.
   > Choose s to be the string 1^p0^p, which is in X_1 because we can choose $u = 1^p$ and $w = 0^p$ which each have length p. Since s is in X_1 and has length greater than or equal to p, if p were to be a pumping length for X_1, s ought to be pumpable. 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 $i \geq 0$, $xy^iz \in X_1$. 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

[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 ask that you 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.

$i = 2$, we get a string xy^iz that is not in X_1 because repeating y twice adds 1s to u but not to w, and strings in X_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 X_1. Since p was arbitrary, we have demonstrated that X_1 has no pumping length. By the Pumping Lemma, this implies that X_1 is nonregular.

Solution i The error comes when the string xy^iz is claimed to not be in X_1 because the argument considers a specific u and w but not allowing these parts of a string to change to correspond to different strings in X_1.

Solution ii This set is regular because it is the set of even length strings and can be described by the regular expression

$$((0 \cup 1)(0 \cup 1))^*$$

(b) The language $X_2 = \{u0w \mid u \text{ and } w \text{ are strings over } \{0,1\} \text{ and have the same length}\}$.

"Proof" that X_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 X_2.

Choose s to be the string 1^p0^{p+1}, which is in X_2 because we can choose $u = 1^p$ and $w = 0^p$ which each have length p. Since s is in X_2 and has length greater than or equal to p, if p were to be a pumping length for X_2, s ought to be pumpable. 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 $i \geq 0$, $xy^iz \in X_2$. When $x = \varepsilon$ and $y = 1^p$ and $z = 0^{p+1}$, we have satisfied that $s = xyz$, $|y| > 0$ (because p is positive) and $|xy| \leq p$. If we let $i = 2$, we get the string $xy^iz = 1^{2p}0^{p+1}$ that is not in X_2 because its middle symbol is a 1, not a 0. 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 X_2. Since p was arbitrary, we have demonstrated that X_2 has no pumping length. By the Pumping Lemma, this implies that X_2 is nonregular.

Solution i The error comes when the string x is chosen to be ε and y is chosen to be 1^p. To prove that a string is not pumpable, we must prove that *all* valid choices of x and y lead to a counterexample i value.

Solution ii A valid proof that X_2 is nonregular is the following: Let p be an arbitrary positive integer. We will show that p is not a pumping length for X_2.

Choose s to be the string 1^p01^p, which is in X_2 because we can choose $u = 1^p = w$. Since s is in X_2 and has length $2p+1$ which is greater than or equal to p, if p were to be a pumping length for X_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 $i \geq 0$, $xy^iz \in L$. Let x, y, z be arbitrary strings with $s = xyz$, $|xy| \leq p$, and $|y| > 0$. By definition of s, this means that x and y are both strings of all 1s, so there are integers k and m (with $m > 0$) such that

$$x = 1^k \qquad y = 1^m \qquad z = 1^{p-k-m}01^p$$

Consider $i = 0$ and the string $xy^iz = xy^0z = xz = 1^k1^{p-k-m}01^p = 1^{p-m}01^p$. This string is not in X_2 because it has only one zero so, were it in X_2, that 0 would need to be the middle symbol but it's not because $p - m < p$ (since $m > 0$). 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 X_2. Since p was arbitrary, we have demonstrated that X_2 has no pumping length. By the Pumping Lemma, this implies that X_2 is nonregular.

2. (*Graded for correctness*[2]) Give an example of a language over the alphabet $\{a, b, c\}$ that has cardinality 2 and for which 4 is a pumping length and 3 is not a pumping length. A complete solution will give a clear and precise description of the language, a justification for why 4 is a pumping length, and a justification for why 3 is not a pumping length.

Solution: Recall the definition from Week 4 Monday's notes: A positive integer p is a **pumping length** of a language L over Σ means that, for each string $s \in \Sigma^*$, if $|s| \geq p$ and $s \in L$, then there are strings x, y, z such that

$$s = xyz$$

and

$$|y| > 0, \qquad \text{for each } i \geq 0,\ xy^iz \in L, \qquad \text{and} \qquad |xy| \leq p.$$

Let's consider the language $\{aca, b\}$ which has two distinct elements (hence cardinality 2) and where the elements of this language are strings over $\{a, b, c\}$. We will prove that 4 is a pumping length of this language and 3 is not a pumping length of this language.

- To prove that 4 is a pumping length for $\{aca, b\}$, we need to show that each string in the language of length greater than or equal to 4 is pumpable. Since the strings in our language have lengths 3 and 1, there are no strings in our language of length greater than or equal to 4 so the universal claim about all such strings is vacuously true. Thus, 4 is a pumping length.
- To prove that 3 is not a pumping length for this language, we need to find a string in the language and of length greater than or equal to 3 that is not pumpable. Consider $s = aca$. We will show that it is not pumpable. To do so: consider an arbitrary choice of strings x, y, z satisfying the constraints that $s = xyz$, $|xy| \leq 3$ and $|y| > 0$. We need to show that there is some nonnegative integer i such that $xy^iz \notin \{aca, b\}$. Consider $i = 4$. Then $|xy^iz| = |x| + 4|y| + |z| \geq |x| + 4 + |z|$ because $|y| \geq 1$. Since $|x| \geq 0, |z| \geq 0$, we have that $|xy^iz| \geq 4 > 3$. This string cannot be in $\{aca, b\}$ because its length is greater than the max length of strings in the set. We have therefore proved that aca is not pumpable and hence that 3 is not a pumping length for $\{aca, b\}$.

[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.

3. (*Graded for fair effort completeness*) Prove or disprove each of the following statements. (In other words, decide whether each statement is true or false and justify your decision.) Fix Σ an arbitrary (but unknown) alphabet.

   (a) If a language L over Σ is nonregular then its complement $\overline{L}$ is regular.

   Solution: False. The class of regular languages is closed under complementation. This means that the complement of a regular language is also regular. Applying the contrapositive, since the complement of the complement of a language is the original language, we get that the complement of a nonregular language is also nonregular.

   (b) Each nonregular language over Σ is infinite.

   Solution: True. This is the contrapositive version of the fact we proved in class that every finite language is regular.

   (c) For each $w \in \Sigma^*$, there is a regular language L_w such that $w \in L_w$.

   Solution: True. The language $\{w\}$ is regular (because it's finite) and has w as an element.

   (d) For each $w \in \Sigma^*$, there is a nonregular language L_w such that $w \in L_w$.

   Solution: True. Let L_{nr} be some nonregular language over Σ. Consider the language $L_{new} = L_{nr} \cup \{w\}$. If $w \in L_{nr}$ then $L_{new} = L_{nr}$ is nonregular and has w as an element. Otherwise, $w \notin L_{nr}$. Assume towards a contradiction that L_{new} is regular. In this case, the set $L_{nr} = L_{new} \cap (\Sigma^* \setminus \{w\})$. Moreover, it can be expressed as the intersection of two regular sets, so it is regular (by our work in class). This contradicts the choice of L_{nr} as a nonregular set. Thus, L_{new} must be nonregular, and has w as an element.

   (e) If a language over Σ is recognized by a PDA then it is nonregular.

   Solution: False. Every regular language is also recognized by a PDA.

4. (*Graded for correctness*) In the first week's homework, we saw the definitions of two functions on the set of languages over $\{0, 1\}$: for L a set of strings over the alphabet $\{0, 1\}$, we can define the following associated sets

$$LZ(L) = \{0^k w \mid w \in L, k \in \mathbb{Z}, k \geq 0\}$$

$$TZ(L) = \{w 0^k \mid w \in L, k \in \mathbb{Z}, k \geq 0\}$$

This week we'll just focus on $LZ(L)$. In class and in the reading so far, we've seen the following examples of nonregular languages:

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

Use (some of) the sets above, along with any regular sets you would like, to prove or disprove the statement: "The class of nonregular languages is closed under the function LZ."

A complete solution will include a precise description of whether the statement is true or false, referring back to the definition of closure, the definition of the function LZ, and the definition of nonregularity. You may use any claims we proved in class or that are proved in the textbook reading, so long as you reference them clearly in your argument by referring to a specific page in the notes, timestamp of a video, or page in the book.

Bonus; not for credit: extend this homework problem for $TZ(L)$ as well.

Solution: This statement is false, which we will prove by giving a specific nonregular set whose image under the function LZ is not nonregular (hence regular).

Consider the set $\{0^{n^2} \mid 0 \leq n\}$. This set is similar to the one listed in the question statement

$$\{1^{n^2} \mid 0 \leq n\}$$

, except using $0s$ instead of 1s. Following the proof in Example 1.76 of the book (page 82), we can use the Pumping Lemma to show that $\{0^{n^2} \mid 0 \leq n\}$ is nonregular (basically, because the lengths of successive elements of the set get big too quickly; and precisely, by copying the proof in the book and substituting the character 0 for 1).

To show that $LZ(\{0^{n^2} \mid 0 \leq n\})$ is regular, we apply the definition of the function LZ:

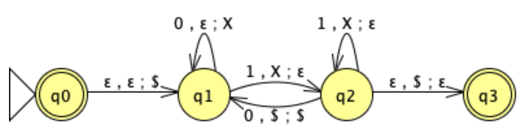
$$LZ(\{0^{n^2} \mid 0 \leq n\}) = \{0^k w \mid w = 0^{n^2}, k \in \mathbb{Z}, k \geq 0, n \geq 0\} = \{0^k 0^{n^2} \mid k \in \mathbb{Z}, k \geq 0, n \geq 0\}$$

Since any nonnegative integer can be expressed as $k + n^2$ for some $k \geq 0$ and $n \geq 0$ (e.g. by choosing $n = 0$), we get that

$$LZ(\{0^{n^2} \mid 0 \leq n\}) = \{0^k \mid k \geq 0\} = L(0^*)$$

a regular language, as necessary for proving that the class of nonregular languages is not closed under the function LZ.

5. Consider the PDA with input alphabet $\Sigma = \{0, 1\}$ and stack alphabet $\Gamma = \{\$, X\}$ and the following state diagram



(a) (*Graded for correctness*) Specify an example string w_1 over Σ that is accepted by this PDA, or explain why there is no such example. A complete solution will include either (1) a precise and clear description of your example string and a precise and clear description of the accepting computation of the PDA on this string (potentially using diagrams like those we used in class when tracing PDA computations) or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

Solution: Consider the example string 01001. An accepting computation of the PDA on this string is as follows:

- The computation starts at $q0$ and moves to $q1$ without consuming any input characters, while pushing a \$ to the stack. At this point, the rest of the string to be read is 01001 and the stack is \$.
- Reading a 0, the computation stays at $q1$ and pushes an X to the stack. At this point, the rest of the string to be read is 1001 and the stack is (top)X\$.
- Reading a 1, the computation moves to $q2$ and pops an X from the stack. At this point, the rest of the string to be read is 001 and the stack is (top)\$.
- Reading a 0, the computation moves back to $q1$ and pops then pushes a \$ from/to the stack. At this point, the rest of the string to be read is 01 and the stack is (top)\$.
- Reading a 0, the computation stays at $q1$ and pushes an X to the stack. At this point, the rest of the string to be read is 1 and the stack is (top)X\$.
- Reading a 1, the computation moves to $q2$ and pops an X from the stack. At this point, there are no more characters of the string to read and the stack is (top)\$.
- The computation moves to $q3$ while popping the \$ from the stack.

Since the computation processes the whole input and ends in $q3$, an accepting state, this computation witnesses that the PDA accepts 01001.

(b) (*Graded for correctness*) Specify an example string w_2 over Σ that is **not** accepted by this PDA, or explain why there is no such example. A complete solution will include either (1) a precise and clear description of your example string and a precise and clear description of all possible computations of the PDA on this string (potentially using diagrams like those we used in class when tracing PDA computations) to show that none of them are accepting or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

Solution: Consider the example string 1. Any computation of the PDA on this string must start at $q0$. At this point, the computation either gets stuck (in which case it's not an accepting computation) or moves to $q1$ without consuming any input characters, while pushing a \$ to the stack. There are only two outgoing arrows from $q1$, one that requires reading a 0 (which the string 1 doesn't have) and the other which requires popping an X from the stack (which we don't have, since the stack only has \$). Thus, any computation of the PDA on the string 1 gets stuck and the PDA does not accept this string.

(c) (*Graded for completeness*) Is the language recognized by this PDA regular or nonregular? You might find it useful to first write out this language in set notation.

Solution: The language recognized by this PDA is

$$\{\varepsilon\} \cup \{0^{i_1}1^{i_1}0^{i_2+1}1^{i_2}\cdots 0^{i_k+1}1^{i_k} \mid k > 1, i_k > 0\}$$

This language is nonregular, which we can prove using the Pumping Lemma: considering an arbitrary positive integer p and showing it is not a pumping length for this language with the counterexample string 0^p1^p.

(d) (*Graded for completeness*) Modify the set of accept states of this state diagram to get a different PDA (with the same set of states, input alphabet, stack alphabet, start state, and transition function) that recognizes an **infinite regular language**, if possible. A complete solution will include either (1) the diagram of this new PDA and an explanation of why the language it recognizes is both infinite and regular, or (2) a sufficiently general and correct argument for why there is no way to choose the set of accept states to satisfy this requirement.

Solution: This is impossible. If we choose the set of accept states to be empty or $\{q0\}$ then the language recognized by the resulting PDA will be finite (either $\emptyset$ or $\{\varepsilon\}$). Informally, adding any of the other states to the set of accept states will result in a PDA that detects some amount of balance between $0s$ and $1s$ in the string so its language is nonregular. Notice that PDAs whose set of accept states equals the set of all states do not necessarily recognize the set of all strings because the computations of such PDAs on some strings can get stuck and therefore not be accepting (even though their last state is in the set of accept states).