COMS W3261: Computer Science Theory, Fall 2024
Yihan Shen, Hellen Zhao

Handout 4B: Solutions to Practice Problems

1. $L_1 = \{0^n 1^m | n \neq m\}$. This can be proved non-regular by all three methods, although in this case the pumping lemma is the most involved, while the other ways are simpler.

   Proof by Pumping Lemma:
   First, we show an INCORRECT way of applying the pumping lemma (a failed attempt). Given p is the pumping length guaranteed by the Pumping Lemma, suppose we choose the simple string $w = 0^p 1^{p+1}$.

   According to the Pumping Lemma, we can write w as $w = xyz$, where:

   - $xy^i z \in L_1$ for all $i \geq 0$,
   - $|xy| \leq p$,
   - $|y| > 0$ (i.e., y is non-empty and can only consist of 0s, since the first p characters of w are all 0s).

   By our choice of w, we know that $y = 0^k$ for some $k > 0$. When we pump substring y i times, this introduces $i - 1$ extra copies of y (because the original word had one copy of y already). Thus, the string becomes $xy^i z = 0^{p+k(i-1)} 1^{p+1}$. To conclude that that L_1 is not regular, we need to show that there exists i such that the resulting string is not in the language, i.e., $p + k(i - 1) = p + 1$. If we knew that $k = 1$, we could have chosen $i = 2$. However, *we cannot assume* $k = 1$ – we only know $1 \leq k \leq p$. So if for example $k = 2$, we get that $p + k(i - 1) \neq p + 1$ (no matter what i we choose), so the string is still in the form of $0^n 1^m$ where $n \neq m$, meaning it still is in the language.

   Thus, pumping this choice of string does not lead to a contradiction with the Pumping Lemma, so let's try a different choice of string.

   Instead, let us choose the string $w = 0^p 1^{p+p!}$. This string is in L_1 because $p \neq p + p!$.

   Now, we apply the Pumping Lemma. For any way to write $w = xyz$, where $|xy| \leq p$ and $|y| > 0$, it must hold that $y = 0^k$ for some $k > 0$. The goal is to pump y to achieve the same number of 0s and 1s. If we pump y i times and consider $w' = xy^i z = 0^{p+(i-1)k} 1^{p+p!}$, we must choose i such that:

   $$(i - 1)k + p = p! + p$$

   This works because $p!$ is divisible by any $k > 0$, which ensures that by selecting the appropriate i (namely $i = p!/k + 1$), we can make the number of 0s and 1s equal, i.e., the resulting string is no longer in L_1. Thus, we conclude that L_1 is not regular. □

Proof by Closure Properties

Assume towards contradiction that the language $L_1 = \{0^m 1^n \mid m \neq n\}$ is regular, then, we know that its complement, L_1^c, is also regular, since regular languages are closed under complement. We consider the following two languages: $L_{eq} = \{0^n 1^n \mid n \geq 0\}$. This is the language of strings with equal numbers of 0s and 1s, which we know from class is not regular. On the other hand, $L_{all} = \{0^m 1^n \mid m, n \geq 0\}$, which contains all strings consisting of any number of 0s followed by any number of 1s, is regular (again, we showed this in class).

We can express L_{eq} as

$$L_{eq} = L_1^c \cap L_{all},$$

Since regular languages are closed under intersection, L_{eq} would also be regular, but we know that it is not, so this leads to a contradiction. Therefore, our assumption is incorrect and L_1 cannot be regular. □

Proof by Myhill-Nerode Theorem

We just need to show that there are infinite equivalence classes under $\sim_{L_1}$. We claim that the strings 0^q, $q \geq 0$, are all distinguishable from each other. Let $q \neq r$, and we just need to show that $0^q \not\sim_{L_1} 0^r$. Take $z = 1^q$ as a distinguishing extension. Note that $0^q 1^q \notin L_1$ but $0^r 1^q \in L_1$. Thus, $0^p \not\sim_{L_1} 0^r$ for all $p, r \geq 0$, so there are infinitely many equivalence classes for $\sim_{L_1}$. Thus, by Myhill-Nerode, L_1 is not regular. □

2. $L_2 = \{ww | w \in \{a, b\}^*\}$

Proof by Pumping Lemma:

Assume for the sake of contradiction that L_2 is regular, and let p be the pumping length.

We choose the string $w = a^p b a^p b$, which is in the language. According to the pumping lemma, w can be split into 3 pieces, $w = xyz$, such that $|xy| \leq p$, $|y| > 0$, and for all $i \geq 0$, $xy^i z \in L_2$. Since the first p characters of w are a, it must be that $y = a^k$ where $0 \leq k \leq p$ (by parts 1 and 2 of the pumping lemma).

Let $i = 3$. The new string is $w' = xy^i z = a^{p+2k} b a^p b$, which is not in the language. We have arrived at a contradiction, and thus L_2 is not regular. □

Proof by Myhill-Nerode Theorem:

Consider two nonequal strings $w \neq s$ in $\{a, b\}^*$. Let $z = w$, so that $wz = ww \in L$ and $sw \notin L$. We have identified a distinguishing extension for w and s, which implies that w and s are pairwise distinguishable by L_2.

Since w and s were arbitrarily chosen, any pair of nonequal strings must be pairwise distinguishable. So each string must form its own equivalence class. Because there are infinitely many strings, there must be infinitely many equivalence classes. By the Myhill-Nerode theorem, this means L_2 is not regular. □

3. $L_3 = \{1^{2^n} | n \geq 0\}$

Proof by Pumping Lemma:
Assume for the sake of contradiction that L_3 is regular. Let p be the pumping length.

Consider $w = 1^{2^p}$. $w \in L_3$ and its length is $|w| = 2^p > p$ (this can be proven by induction for any $p > 0$). According to the pumping lemma, w can be split into 3 pieces, $w = xyz$, such that $|xy| \leq p$, $|y| > 0$, and for all $i \geq 0$, $xy^iz \in L_3$. Then the pieces must be of the following form: $x = 1^a$, $y = 1^b$, and $z = 1^c$, where $a+b+c = 2^p$, $a+b \leq p$, and $b > 0$.

Let $i = 2$. The new string is $w' = xy^iz = 1^a1^{2b}1^c$. Therefore, $|w'| = 2^p + b > 2^p$ because $b > 0$, and we know $b \leq p < 2^p$, since $|xy| = a + b \leq p$. Adding 2^p to the second inequality, we get $|w'| = 2^p + b < 2^p + 2^p = 2^{p+1}$. Since $2^p < |w'| < 2^{p+1}$, the length of w' cannot be a power of two, so w' is not in the language. This is a contradiction, so L_3 is not regular. $\square$

Proof by Myhill-Nerode Theorem:
It would suffice to show that any two strings in L_3 have a distinguishing extension, as there are infinitely many such strings, so this would imply there are an infinite number of equivalence classes.

Let $x = 1^{2^m}$ and $y = 1^{2^n}$ for any nonnegative integers $m < n$. The string $z = 1^{2^m}$ is a distinguishing extension. We can see $xz = 1^{2^m+2^m} = 1^{2^{m+1}} \in L_3$ whereas $yz = 1^{2^n+2^m}$. But $m < n$ implies that $2^m < 2^n$ and $2^m + 2^n < 2^n + 2^n = 2^{n+1}$. Because $2^n < |yz| = 2^m + 2^n < 2^{n+1}$, yz is not in the language. Since $xz \in L_3$ and $yz \notin L_3$, they are pairwise distinguishable by L_3.

This is true for all pairs of strings in L_3, so there must be infinitely many equivalence classes under $\sim_{L_3}$. By the Myhill-Nerode theorem then, L_3 is not regular. $\square$

4. For the alphabet $\{a, b, c\}$, define

$$L_4 = \{c^n w | n \geq 0, w \in \{a, b\}^*, \text{and if } n \text{ is odd then } w = w^R\}$$

This is an example of a non-regular language that satisfies the pumping lemma - so we cannot use the pumping lemma to prove non-regularity (there won't be a contradiction). Below we elaborate on this, and then show two alternative solutions using the other two methods we saw.

Why Pumping Lemma won't work to get contradiction
First, we will show that this language satisfies the pumping lemma, even though it is Not a regular language. We need to show that there exists a pumping number p such that for any string $w \in L_4$, $|w| \geq p$, we can never derive a contradiction with the pumping lemma. Set the pumping number to $p = 2$. We need to show that for any string $c^n w \in L_4$ with $|c^n w| \geq 2$, that there exists a parsing $c^n w = xyz$ with $|xy| \leq 2$, $|y| \geq 1$ such that for any $i \geq 0$, $xy^iz \in L_4$.

Case 1: n is odd. Then, w is a palindrome. Take the parsing $x = \epsilon$, $y = c$, $z = w$. Clearly $|xy| = 1 \leq 2$ and $|y| = 1 \geq 1$. Note that for any $i \geq 0$, $xy^iz = c^ic^{n-1}w = c^{n+i-1}w$. If $n+i-1$ is even we trivially have $xy^iz \in L_4$, and if $n+i-1$ is odd then we have $xy^iz \in L_4$ as w is a palindrome.

Case 2: n is even. Set $x = \epsilon$, $y =$ first two characters of c^nw (these exist since $|c^nw| \geq 2$), $z =$ the remainder of the string. If $n = 0$, then $x, y, z \in \{a, b\}^*$ and so $xy^iz \in \{a, b\}^*$ for any $i \geq 0$. Thus, since xy^iz begins with no c's, it begins with an even number of c's and so $xy^iz \in L_4$. If $n \neq 0$, then $n \geq 2$, and so $y = c^2$. Thus, for any $i \geq 0$, $xy^iz = c^{n+2(i-1)}w$ where $w \in \{a, b\}^*$. But note that since n is even, $n+2(i-1)$ is always even, and so $xy^iz \in L_4$.

In either case, we have a parsing $c^nw = xyz$ such that for any $i \geq 0$, $xy^iz \in L_4$. Therefore, the pumping lemma holds for L_4. □

Note that the pumping lemma holding for L_4 tells us nothing about whether or not L_4 is regular. Next we prove it's not regular using two alterantive methods.

Proof by Myhill-Nerode Theorum

We need to show that there are infinite equivalence classes under $\sim_{L_4}$. We claim that the strings ca^n, $n \geq 0$, are all distinguishable from each other. Let $n \neq m$, and we need to show that $ca^n \not\sim_{L_4} ca^m$. Take $z = ba^n$ as a distinguishing extension. Note that $ca^nba^n \in L_4$ since a^nba^n is a palindrome. However, $ca^mba^n \notin L_4$ since c appears an odd number of times, and a^mba^n is not a palindrome as $m \neq n$. Thus, $ca^n \not\sim_{L_4} ca^m$ for all $n \neq m; n, m \geq 0$, and so there are infinitely many equivalence classes for $\sim_{L_4}$. Thus, by Myhill-Nerode, L_4 is not regular. □

Proof by Closure Properties

Alternatively, one can prove this using a combination of closure properties and the pumping lemma. Supposed that L_4 is regular. Let $L_5 = \{cw \mid w \in \{a, b\}^*\}$. It is not hard to show that L_5 is regular (we leave this as an exercise). Thus, since regular languages are closed under intersection, we get that $L_r = L_4 \cap L_5$ is also regular.

However, we can see that $L_r = L_4 \cap L_5 = \{cw \mid w \in \{a, b\}^*, w = w^R\}$.

Now, we can use the pumping lemma to prove that L_r cannot be regular (we leave this as an exercise – it is very similar to how we proved in class that the langauage of palindromes is not regular).

We proved that if L_4 were regular, then L_r would be regular, but we know that L_r is not regular, so we have a contradiction, and conclude that L_4 is not regular. □