Lecture 10: Proving non-regularity using Myhill-Nerode Thm and Pumping Lemma

18 February 2010

In this lecture, we will see how to prove that a language is **not** regular.

We will see two methods for showing that a language is not regular. The Myhill-Nerode theorem and the "pumping lemma" show that certain key "seed" languages are not regular. From these seed languages, we can show that many similar languages are also not regular, using closure properties.

1 Proving non-regularity via the Myhill-Nerode Theorem

Recall that the Myhill-Nerode theorem says, among other things, that if L is a language that has an infinite number of suffix languages, then L is not regular. Intuitively, we need a state of a DFA for every suffix language of L, and hence need an infinite number of states that no DFA can accommodate.

This gives a way of proving a language is not regular. Let $L \subseteq \Sigma^*$. Then L is not regular if there are an infinite number of suffix languages, say $\{[\![L/x_1]\!], [\![L/x_2]\!], \ldots\}$ that are all distinct from each other (i.e. $[\![L/x_i]\!] \neq [\![L/x_j]\!]$, for any $i \neq j$).

In other words, in order to prove L is not regular, we need to exhibit an infinite set of strings $S = \{x_1, x_2, \ldots\}$ such that for each $x, y \in S$, if $x \neq y$, then $[\![L/x]\!] \neq [\![L/y]\!]$. But when is $[\![L/x]\!] \neq [\![L/y]\!]$ hold? Clearly, this holds if and only if there exists a $z \in \Sigma^*$ that is in one suffix language but not in the other, i.e. $\exists z \in \Sigma^*$ such that $z \in [\![L/x]\!]$ *and* $z \notin [\![L/y]\!]$, or $z \notin [\![L/x]\!]$ *and* $z \in [\![L/y]\!]$. Restating this, we must show that $\exists z \in \Sigma^*$ such that ($xz \in L$ *and* $yz \notin L$), or ($xz \notin L$ *and* $yz \in L$).

This leads us to the following definition of when two strings x and y are *distiguishable* with respect to L: they are distinguishable if we can find a z such that xz and yz have a different membership status in L.

Definition 1.1 Two strings $x, y \in \Sigma^*$ are ***distinguishable*** with respect to $L \subseteq \Sigma^*$, if there exists a word $z \in \Sigma^*$, such that *precisely one* of the strings xw and yw is in L (and the other is not).

Hence, continuing our above discussion, in order to prove L is non-regular, it is sufficient to show that there is an infinite set $S \subseteq \Sigma^*$ such that for every $x, y \in S$ with $x \neq y$, x and y are distinguishable with respect to L, i.e. $\exists z \in \Sigma^*$ such that ($xz \in L$ *and* $yz \notin L$), or ($xz \notin L$ *and* $yz \in L$).

We hence have

Theorem 1.2 *Let $L \subseteq \Sigma^*$. If there is an infinite set $S \subseteq \Sigma^*$ such that for every $x, y \in S$ with $x \neq y$, x and y are distinguishable with respect to L, then L is not a regular language.*

The above theorem gives us the first technique for proving non-regularity, where to establish that a language is not regular, we exhibit an infinite set S and show that elements of S are pairwise distinguishable.

1.1 Examples of Proving Non-regularity using MNT

1.1.1 Example

Lemma 1.3 *The language*

$$L = \left\{ \mathtt{1}^k y \;\middle|\; y \in \{\mathtt{0},\mathtt{1}\}^*, \text{ and } y \text{ contains at most } k \text{ ones} \right\}$$

is not regular.

Proof: Let $S = \{\mathtt{1}^i \mid i \in \mathbb{N}\}$. Clearly S is infinite. Let $x = \mathtt{1}^i$ and $y = \mathtt{1}^j$ be two different elements in S. Then $i \neq j$. Assume, without loss of generality, that $j > i$. Now choose $z = \mathtt{0}\mathtt{1}^j$. Then $xz = \mathtt{1}^i\mathtt{0}\mathtt{1}^j \notin L$ but $y\mathtt{0}\mathtt{1}^j = \mathtt{1}^j\mathtt{0}\mathtt{1}^j \in L$. Hence x and y are distinguishable with respect to L, for any $x, y \in S$ with $x \neq y$. We conclude, by Theorem 1.2, that L is not regular. ■

1.1.2 Example: ww is not regular

Claim 1.4 *For $\Sigma = \{\mathtt{0},\mathtt{1}\}$, the language $L = \left\{ ww \;\middle|\; w \in \Sigma^* \right\}$ is not regular.*

Proof: Let $S = \{\mathtt{0}^i \mid i \in \mathbb{N}\}$. Of course, S is infinite. Let $x, y \in S$ with $x \neq y$. Let $x = \mathtt{0}^i$ and $y = \mathtt{0}^j$, with $i \neq j$. Now choose $z = 10^i1$. Then

$$xz = \mathtt{0}^i \underbrace{\mathtt{1}\mathtt{0}^i\mathtt{1}}_{z} \in L \quad \text{but} \quad yz = \mathtt{0}^j \underbrace{\mathtt{1}\mathtt{0}^i\mathtt{1}}_{z} \notin L$$

Which means that x and y are distinguishable with respect to L. Hence we have that L is not regular (by Theorem 1.2). ■

2 The Pumping Lemma

2.1 Proof by repetition of states

We next prove a language non-regular by a slightly different argument.

Claim. *The language $L = \left\{ \mathtt{a}^n\mathtt{b}^n \;\middle|\; n \geq 0 \right\}$ is not regular.*

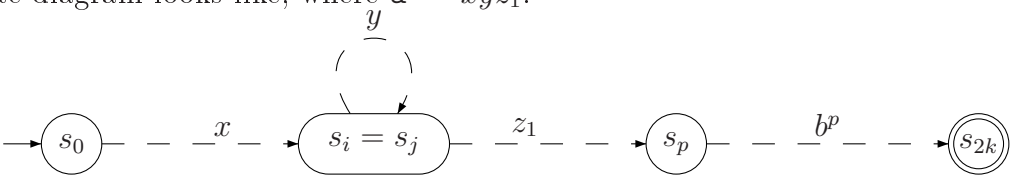
Proof: Suppose that L were regular. Then L is accepted by some DFA

$$M = (Q, \Sigma, \delta, q_0, F).$$

Suppose that M has p states.

Consider the string $\mathsf{a}^p\mathsf{b}^p$. It is accepted using a sequence of states $s_0 s_1 \ldots s_{2p}$. Right after we read the last a, the machine is in state s_p.

In the sub-sequence $s_0 s_1 \ldots s_p$, there are $p+1$ states. Since L has only p distinct states, this means that two states in the sequence are the same (by the pigeonhole principle). Let us call the pair of repeated states q_i and q_j, $i < j$. This means that the path through M's state diagram looks like, where $\mathsf{a}^p = xyz_1$.



But this DFA will accept all strings of the form $xy^j z_1 \mathsf{b}^p$, for $j \geq 0$. Indeed, for $j = 0$, this is just the string $xz_1\mathsf{b}^p$, which this DFA accepts, but it is not in the language, since it has less as than bs. That is, if $|y| = m$, the DFA accepts all strings of the form $\mathsf{a}^{p-m+jm}\mathsf{b}^m$, for any $j \geq 0$. For any value of j other than 1, such strings are not in L.

So our DFA M accepts some strings that are not in L. This is a contradiction, because L was supposed to accept L. Therefore, we must have been wrong in our assumption that L was regular. ■

2.2 The pumping lemma

The pumping lemma generalizes the above argument into a standard template, which we can prove once and then quickly apply to many languages.

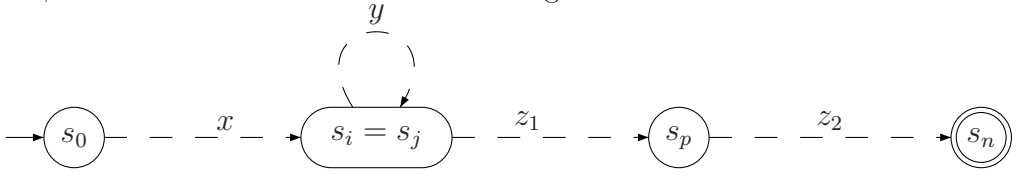
Theorem 2.1 (Pumping Lemma.) *Let L be a regular language. Then there exists an integer p (the "pumping length") such that for any string $w \in L$ with $|w| \geq p$, w can be written as xyz with the following properties:*

- $|xy| \leq p$.
- $|y| \geq 1$ *(i.e. y is not the empty string).*
- $xy^k z \in L$ *for every $k \geq 0$.*

Proof: The proof is written out in full detail in Sipser, here we just outline it.

Let M be a DFA accepting L, and let p be the number of states of M. Let $w = c_1 c_2 \ldots c_n$ be a string of length $n \geq p$, and let the accepting state sequence (i.e., trace) for w be $s_0 s_1 \ldots s_n$.

There must be a repeat within the sequence from s_0 to s_p, since M has only p states, and as such, the situation looks like the following.



So if we set $z = z_1 z_2$, we now have x, y, and z satisfying the conditions of the lemma.

- $|xy| \leq p$ because repeat is within first $p+1$ states

- $|y| \geq 1$ because i and j are distinct
- $xy^k z \in L$ for every $k \geq 0$ because a loop in the state diagram can be repeated as many or as few times as you want.

  Formally, for any k, the word $xy^i z$ goes through the following sequence of states:

$$s_0 \xrightarrow{\mathtt{x}} \overbrace{s_i \xrightarrow{\mathtt{y}} s_i \xrightarrow{\mathtt{y}} \cdots \xrightarrow{\mathtt{y}}}^{k \text{ times}} s_i = s_j \xrightarrow{\mathtt{z}} s_n,$$

  and s_n is an accepting state. Namely, M accepts $xy^k z$, and as such $xy^k z \in L$.

This completes the proof of the theorem. ■

Notice that we do not know exactly where the repeat occurs, so we have very little control over the length of x and z_1.

2.3 Using the PL to show non-regularity

If L is regular, then it satisfies the pumping lemma (PL). Therefore, intuitively, if L does not satisfy the pumping lemma, L cannot be regular.

2.3.1 Restating the Pumping Lemma via the contrapositive

We want to restate the pumping lemma in the contrapositive. Now, it is **not** true that if L satisfies the conditions of the PM, then L must be regular. Reminder from CS 173: contrapositive of if-then statement is equivalent, converse is not.

What does it mean to **not** satisfy the Pumping Lemma? Write out PL compactly:

$$\text{L is regular.} \implies \left(\exists p \; \forall w \in L \quad |w| \geq p \Rightarrow \left(\exists x, y, z \text{ s.t. } \begin{array}{l} w = xyz, \\ |xy| \leq p, \\ |y| \geq 1, \end{array} \text{ and } \forall i \quad xy^i z \in L. \right) \right).$$

Now, we know that if A implies B, then $\overline{B}$ implies $\overline{A}$ (contraposition), as such the Pumping Lemma, can be restated as

$$\overline{\left(\exists p \; \forall w \in L \quad |w| \geq p \Rightarrow \left(\exists x, y, z \; \begin{array}{l} w = xyz, \\ |xy| \leq p, \\ |y| \geq 1, \end{array} \text{ and } \forall i \quad xy^i z \in L. \right) \right)} \implies \overline{L \text{ is regular.}}$$

Now, the logical statement $A \Rightarrow B$ is equivalent to $\overline{A} \vee B = \overline{A \wedge \overline{B}}$. As such $\overline{A \Rightarrow B} = A \wedge \overline{B}$. In addition, negation flips quantifies, as such, the above is equivalent to

$$\left(\forall p \; \exists w \in L \quad |w| \geq p \text{ and } \overline{\left(\exists x, y, z \; \begin{array}{l} w = xyz, \\ |xy| \leq p, \\ |y| \geq 1, \end{array} \text{ and } \forall i \quad xy^i z \in L. \right)} \right) \implies \begin{array}{c} \text{L is} \\ \text{not regular.} \end{array}$$

Since, $\overline{A \wedge \overline{B}} = A \Rightarrow B$ we have that $\overline{A \wedge B} = \big(A \Rightarrow \overline{B}\big)$. Thus, we have

$$\left(\forall p\ \exists w \in L \quad |w| \geq p \text{ and } \left(\forall x,y,z \begin{array}{l} w = xyz, \\ |xy| \leq p, \\ |y| \geq 1, \end{array} \implies \overline{\forall i \quad xy^i z \in L.}\right)\right) \implies \begin{array}{c}\text{L is} \\ \text{not regular.}\end{array}$$

Which is equivalent to

$$\left(\forall p\ \exists w \in L \quad |w| \geq p \text{ and } \left(\forall x,y,z \begin{array}{l} w = xyz, \\ |xy| \leq p, \\ |y| \geq 1, \end{array} \implies \exists i \quad xy^i z \notin L.\right)\right) \implies \begin{array}{c}\text{L is} \\ \text{not regular.}\end{array}$$

The translation into words is the contrapositive of the Pumping Lemma (stated in Theorem 2.2 below).

2.3.2 The contrapositive of the Pumping Lemma

Theorem 2.2 (Pumping Lemma restated.) *Consider a language L. If for any integer $p \geq 0$ there exists a word $w \in L$, such that $|w| \geq p$, and for any breakup of w into three strings x, y, z, such that:*

- $w = xyz$,
- $|xy| \leq p$,
- $|y| \geq 1$,

implies that there exists an i such that $xy^i z \notin L$, then the language L is not regular.

2.3.3 Proving that a language is not regular

Let us assume that we want to show that a language L is *not* regular.

Such a proof is done by contradiction. To prove L is not regular, we assume it is regular. This gives us a specific (but unknown) pumping length p. We then show that L satisfies the rest of the contrapositive version of the pumping lemma, so it can not be regular.

So the proof outline looks like:

- Suppose L is regular. Let p be its pumping length.
- Consider $w =$ [formula for a specific class of strings]
- By the Pumping Lemma, we know there exist x, y, z such that $w = xyz$, $|xy| \leq p$, and $|y| \geq 1$.
- Consider $i =$ [some specific value, almost always 0 or 2]
- $xy^i z$ is not in L. [explain why it can't be]

Notice that our adversary picks p. We get to pick w whose length depends on p. But then our adversary gets to pick the specific division of w into x, y, and z.

2.4 Examples

2.4.1 The language $L = \mathtt{a}^n\mathtt{b}^n$ is not regular

Claim 2.3 *The language $L = \mathtt{a}^n\mathtt{b}^n$ is not regular.*

Proof: For any $p \geq 0$, consider the word $w = \mathtt{a}^p\mathtt{b}^p$, and consider any breakup of w into three parts, such that $w = xyz$ $|y| \geq 1$, and $|xy| \leq p$. Clearly, xy is a prefix of w made out of only $\mathtt{a}$s. As such, the word $xyyz$ has more $\mathtt{a}$s in it than $\mathtt{b}$s, and as such, it is not in L.

But then, by the Pumping Lemma (Theorem 2.2), L is not regular. ■

2.4.2 The language $\{ww\}$ is not regular

Claim 2.4 *The language $L = \left\{ ww \;\middle|\; w \in \Sigma^* \right\}$ is not regular.*

Proof: For any $p \geq 0$, consider the word $w = \mathtt{0}^p\mathtt{1}\mathtt{0}^p\mathtt{1}$, and consider any breakup of w into three parts, such that $w = xyz$ $|y| \geq 1$, and $|xy| \leq p$. Clearly, xy is a prefix of w made out of only $\mathtt{0}$s. As such, the word $xyyz$ has more $\mathtt{0}$s in its first part than the second part. As such, $xyyz$ is not in L.

But then, by the Pumping Lemma (Theorem 2.2), L is not regular. ■

Consider the word w used in the above claim:

- It is concrete, made of specific characters, no variables left in it.
- These strings are a subset of L, chosen to exemplify what is not regular about L.
- Its length depends on p.
- The $\mathtt{1}$ in the middle serves as a barrier to separate the two groups of $\mathtt{0}$'s. (Think about why the proof would fail if it was not there.)
- The $\mathtt{1}$ at the end of w does not matter to the proof, but we nee it so that $w \in L$.

2.5 A note on finite languages

A language L is ***finite*** if has a bounded number of words in it. Clearly, a finite language is regular (since you can always write a finite regular expression that matches all the words in the language).

It is natural to ask why we can not apply the pumping lemma Theorem 2.1 to L? The reason is because we can always choose the threshold p to be larger than the length of the longest word in L. Now, there is no word in L with length larger than p in L. As such, the claim of the Pumping Lemma holds trivially for a finite language, but no word can be pumped - and as such L stays finite. So the pumping lemma makes sense even for finite languages!

3 Non-regularity via closure properties

If we know certain seed languages are not regular, then we can use closure properties to show other languages are not regular.

We remind the reader that *homomorphism* is a mapping $h : \Sigma_1 \to \Sigma_2^*$ (namely, every letter of Σ_1 is mapped to a string over Σ_2). We showed that if a language L over Σ_1 is regular, then the language $h(L)$ is regular. We referred to this property as *closure of regular languages under homomorphism.*

We know that the language $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ is not regular (by MNT or pumping lemma arguments). Now let us show this:

Claim 3.1 *The language* $L' = \{\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\}$ *is not regular.*

Proof: Assume for the sake of contradiction that L' is regular. Let h be the homomorphism that maps `0` to `a` and `1` to `b`. Then $h(L')$ must be regular (closure under homomorphism). But $h(L')$ is the language

$$L = \left\{ \mathtt{a}^n\mathtt{b}^n \;\middle|\; n \geq 0 \right\}, \tag{1}$$

which is not regular. A contradiction. As such, L' is not regular. ■

We remind the reader that regular languages are also closed under intersection.

Claim 3.2 *The language* $L_2 = \left\{ w \in \{\mathtt{a},\mathtt{b}\}^* \;\middle|\; w \text{ has an equal \# of } \mathtt{a}\text{'s and } \mathtt{b}\text{'s} \right\}$ *is not regular.*

Proof: Suppose L_2 were regular. Consider $L_2 \cap \mathtt{a}^*\mathtt{b}^*$. This must be regular because L_2 and $\mathtt{a}^*\mathtt{b}^*$ are both regular and regular languages are closed under intersection. But $L_2 \cap \mathtt{a}^*\mathtt{b}^*$ is just the language $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$, which we know is not regular. The contradiction proves that L_2 is not regular. ■

Claim 3.3 *The language* $L_3 = \left\{ \mathtt{a}^n\mathtt{b}^n \;\middle|\; n \geq 1 \right\}$ *is not regular.*

Proof: Assume for the sake of contradiction that L_3 is regular. Consider $L_3 \cup \{\epsilon\}$. This must be regular because L_3 and $\{\epsilon\}$ are both regular and regular languages are closed under union. But $L_3 \cup \{\epsilon\}$ is just $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$, which is not regular. This contradiction shows that L_3 is not regular. ■

3.1 Being careful in using closure arguments

Most closure properties must be applied in the correct direction: We show (or assume) that all inputs to the operation are regular, therefore the output of the operation must be regular.

For example, consider (again) the language $L_B = \{0^n1^n \mid n \geq 0\}$, which is not regular.

Since L_B is not regular, $\overline{L_B}$ is also not regular. If $\overline{L_B}$ were regular, then L_B would also have to be regular because regular languages are closed under set complement. However, many similar lines of reasoning do not work for other closure properties.

For example, L_B and $\overline{L_B}$ are both non-regular, but their union is regular. Similarly, suppose that L_k is the set of all strings of length $\leq k$. Then $L_B \cap L_k$ is regular, even though L_B is not regular.

If you are not absolutely sure of what you are doing, always use closure properties in the forward direction. That is, establish that L and L' are regular, then conclude that L OP L' must be regular.

Also, be sure to apply only closure properties that we know to be true. In particular, regular languages are **not** closed under the subset and superset relations. Indeed, consider $L_1 = \{001, 00\}$, which is regular. But L_1 is a subset of L_B, which is not regular. Similarly, $L_2 = (0+1)^*$ is regular. And it is a superset of L (from Eq. (1) in the proof of Claim 3.1)). But you can not deduce that L is therefore regular. We know it is not.

So regular languages can be subsets of non-regular ones and vice versa.