## Definition

A language that cannot be defined by a regular expression is called a **nonregular** language.

By Kleene’s Theorem, a nonregular language can also not be accepted by any Finite Automaton (DFA or NFA) or by any Transition Graph.

## Example

\[ L = \{\lambda \text{ } ab \text{ } aabbb \text{ } aaaaabbb \text{ } \ldots\} \]

or alternatively defined as:

\[ L = \{a^n b^n\} \]
The Pumping Lemma

**Lemma**

Let $L$ be any regular language that has infinitely many words. Then there exists some three strings $x$, $y$, and $z$ (where $y$ is **not** the null string) such that all strings of the form

$$xy^n z$$

for $n = 1, 2, 3, \ldots$ are words in $L$.

**Proof (start...)**

If $L$ is a regular language, then there is an FA that accepts exactly the words in $L$ and no more. This FA will have a finite number of states but infinitely many words. This means there is some cycle.

Let $w$ be some word in $L$ that has more letters in it than there are states in the machine. When this word generates a path through the machine, we **must** revisit a state that it has been to before.
Let us break up the word $w$ into three parts:

1. Let $x$ be all the letters of $w$ starting at the beginning that lead up to the first state that is revisited. $x$ may be the null string.

2. Let $y$ denote the substring of $w$ that travels around the “circuit” which loops. $y$ cannot be the null string.

3. Let $z$ be the rest of the letters in $w$ that starts after $y$. This $z$ could be null. The path for $z$ could also possibly loop around the $y$-circuit (it’s arbitrary).

Clearly, from this definition given above,

$$w = xyz$$

and $w$ is accepted by this machine.
Continuing the Proof of the Pumping Lemma (3/3)

Q1: What is the path through this machine of the input string $xyz$?

Q2: What is the path through this machine of the input string $xyyz$?

Note: All languages $L$ must be of the form $w = xy^n z$ for this to be “accepted”. If they were not of this form, then the FA would not have such a trace.
Example

\[ w = \text{bbbababa} \]

\[ w = \text{b baa baba} \]

\[ x \quad y \quad z \]

What would happen when \( w = xyz = \text{b bba bba baba} \)?
Show $L$ is Non-regular with the Pumping Lemma

Suppose for a moment that we never talked about $L = \{a^n b^n\}$

The pumping lemma states there must be strings $x, y, \text{ and } z$ such that all words of the form $x y^n z$ are in $L$. Is this possible?

$$aaa \ldots aaaabbbb \ldots bbb$$

- If $y$ is made entirely of $a$’s then when we pump to $x y y z$, the word will have more $a$’s than $b$’s.
- If $y$ is made entirely of $b$’s then when we pump to $x y y z$, the word will have more $b$’s than $a$’s.
- $y$ must be made up of some number of $a$’s followed by some number of $b$’s. This means $x y y z$ would have two copies of the substring $a b$. Our original language prohibits this. Therefore, $x y y z$ cannot be a word in $L$. And $L$ is not regular.
Another Example of Showing $L$ is Non-regular

Once we have shown $\{a^n b^n\}$ is non-regular, we can show that the language EQUAL (all words with the same total number of $a$’s and $b$’s) is also non-regular.

- The language $\{a^n b^n\}$ is the intersection of all words defined by the regular expression $a^* b^*$ and the language EQUAL.

$$\{a^n b^n\} = a^* b^* \cap \text{EQUAL}$$

- If EQUAL were a regular language, then $\{a^n b^n\}$ would be the intersection of two regular language (as discussed in Chapter 9). Additionally, it would need to be regular itself (which it is not).
- Therefore, EQUAL cannot be regular since $\{a^n b^n\}$ is non-regular.
Yet Another Non-regular Language

Consider the language \( L = a^nba^n = \{b \ aba \ aabaa \ aaabaaa \ldots\} \). If this language were regular, then we know the Pumping Lemma would have to hold true.

- \( xyz \) and \( xyyz \) would both need to be in \( L \)
- **Observation 1**: If the \( y \) string contained the \( b \), then \( xyyz \) would contain two \( b \)'s. This is not possible – \( xyyz \) is not part of \( L \)
- **Observation 2**: If the \( y \) string contained all \( a \)'s then the \( b \) in the middle is either on the \( x \) or \( z \) side. In either case, \( xyyz \) would increase the number of \( a \)'s either before or after the \( b \)
- **Conclusion 1**: \( xyyz \) does not have \( b \) in the middle and is not of the form \( a^nba^n \)
- **Conclusion 2**: \( L \) cannot be pumped and is therefore not regular
Additional Examples (on Chalkboard)

1. $a^n b^n a b^{n+1}$
2. PALINDROME
3. PRIME = \{ $a^n$ where $p$ is a prime $\}$

Plus a Stronger Theorem

Let $L$ be an infinite language accepted by a finite automaton with $N$ states. Then for all words $w$ in $L$ that have more than $N$ letters, there are strings $x$, $y$, and $z$, where $y$ is not null and $\text{length}(x) + \text{length}(y)$ does not exceed $N$ such that

$$w = xyz$$

and all strings of the form

$$xy^n z \text{ (for } n = 1 \ 2 \ 3 \ \ldots)$$

are in $L$ \qed
Limitations of the pumping lemma

The pumping lemma is *negative* in its application. It can only be used to show that certain languages are not regular.

- Let’s consider some FA – each state (final or non-final) can be thought of as creating a society of a certain class of strings.
- If there exists a string formed by some path leading to a state, it is part of that state’s society.
- If string $x$ and string $y$ are in the same society, then for all other strings $z$, either $xz$ and $yz$ are both accepted or rejected

Theorem (The Myhill-Nerode Theorem)

*Given a language $L$, we shall say two string $x$ and $y$ are in the same class if for all possible strings $z$, $xz$ and $yz$ are both in $L$ or both are not*

1. **The language $L$ divides the set of all strings into separate classes**
2. **If $L$ is regular, the number of classes $L$ creates is finite.**
3. **If the number of classes $L$ creates is finite, then $L$ is regular**
Proof by contradiction – Part 1.

• Split classes in an intentionally bad way: Suppose any two students at college are in the same class if they have taken a course together.

• A and B may have taken history together, B and C may have taken geography together, but A and C never took a class together. Then A, B, and C are not all in the same class.

• If AZ and BZ are always in L and BZ (or not) and CZ are always in L (or not), then A, B, and C must all be in the same class.

• If S is in a class with X and S is also in a class with Y, then by reasoning above X and Y must be in the same class.

• Therefore, S cannot be in two different classes. No string is in two different classes and every string must be in some class.

• Therefore, every string is in exactly one class
Proving the Myhill-Nerode Theorem

Proof of Part 2.

• If $L$ is regular, then there is some FA that accepts $L$.
• Its finite number of states create a finite division of all strings into a finite number of societies.
• The problem is that two different states may define societies that are actually the same class

• Society “class” of $q_1$ and $q_2$: any word in them when followed by a string $z$ will be accepted IFF $z$ contains an $a$
• Since the societies are in the same class, and there are finitely many societies, there must be a finite number of classes.
Let the finitely many classes be $C_1, C_2, \ldots C_n$ where $C_1$ is the class containing $\lambda$. We will transform these classes into an FA by showing how to draw the edges between (and assign start and final states)

1. The start state must be $C_1$ because of $\lambda$

2. If a class contains one word of $L$ then $w \in L \ \forall w \in C$. Let $s \in L, w \in L \ | \ w \in C_k$. When $z = \lambda$, $w\lambda \in L \land s\lambda \in L$ (or not).
Label all states that are subsets of $L$ as final states.

3. Repeat the following for all classes $C_m$:
If $x \in C_m \land y \in C_m$, then $\forall z \ (xz \in L \land yz \in L)$.
Let $C_a = xa \forall x \in C_m$. Draw an $a$-edge from $C_m$ to $C_a$.
Let $C_b = xb \forall x \in C_m$. Draw an $b$-edge from $C_m$ to $C_b$.

4. Once outgoing edges are drawn for all classes, we have an FA
Examples using Myhill-Nerode Theorem

All words that end in $a$

- $C_1$ – all strings that end in $a$ (final)
- $C_2$ – all strings that don’t end in $a$ (start)

All words that contain a double $a$

- $C_1$ – strings without $aa$ that end in $a$
- $C_2$ – strings without $aa$ that end in $b$ or $\lambda$
- $C_3$ – strings with $aa$
Examples using Myhill-Nerode Theorem

Showing languages are regular

- EVEN-EVEN
- two or more $b$’s
- start and end with the same letter

Showing languages are non-regular

- $a^n b^n$
- $a^n ba^n$
- EQUAL
- PALINDROME
Examples using Myhill-Nerode Theorem

Showing languages are regular

• EVEN-EVEN
• two or more $b$’s
• start and end with the same letter

Showing languages are non-regular

• $a^n b^n$ We only need to observe that $a$, $aa$, $aaa$, . . . are all in different classes because there’s exactly $b^m$ that will match $a^m$
• $a^n ba^n$
• EQUAL
• PALINDROME
Examples using Myhill-Nerode Theorem

Showing languages are regular

- EVEN-EVEN
- two or more $b$’s
- start and end with the same letter

Showing languages are non-regular

- $a^n b^n$ We only need to observe that $a, aa, aaa, \ldots$ are all in different classes because there’s exactly $b^m$ that will match $a^m$
- $a^n ba^n$ The strings $ab, aab, aaab, \ldots$ are all in different classes because we need a matching $ba^m$ for each class
- EQUAL
- PALINDROME
Examples using Myhill-Nerode Theorem

Showing languages are regular

• EVEN-EVEN
• two or more $b$’s
• start and end with the same letter

Showing languages are non-regular

• $a^n b^n$ We only need to observe that $a$, $aa$, $aaa$, … are all in different classes because there’s exactly $b^m$ that will match $a^m$
• $a^n ba^n$ The strings $ab$, $aab$, $aaab$, … are all in different classes because we need a matching $ba^m$ for each class
• EQUAL Because for each of the strings $a$, $aa$, $aaa$, $aaaa$, … some $z = b^m$ will be alone in EQUAL
• PALINDROME
Examples using Myhill-Nerode Theorem

Showing languages are regular

- EVEN-EVEN
- two or more $b$’s
- start and end with the same letter

Showing languages are non-regular

- $a^n b^n$ We only need to observe that $a$, $aa$, $aaa$, . . . are all in different classes because there’s exactly $b^m$ that will match $a^m$
- $a^n b a^n$ The strings $ab$, $aab$, $aaab$, . . . are all in different classes because we need a matching $ba^m$ for each class
- EQUAL Because for each of the strings $a$, $aa$, $aaa$, $aaaa$, . . . some $z = b^m$ will be alone in EQUAL
- PALINDROME $ab$, $aab$, $aaab$, . . . are all in different classes. For each, one value of $z = a^m$ will create a PALINDROME when added but to no other
## Bonus: Prefixes

### Definition

If $R$ and $Q$ are languages, then the language “the prefixes of $Q$ in $R$,” denoted by the symbolism $\text{Pref}(Q \text{ in } R)$ is the set of all strings of letters that can be concatenated to the front of some word in $Q$ to produce some word in $R$

$$\text{Pref} (Q \text{ in } R) = \text{all strings } p \text{ such that } q \in Q, w \in R \mid pq = w$$

### Theorem

*If $R$ is regular and $Q$ is any language whatsoever, then the language $P = \text{Pref}(Q \text{ in } R)$ is regular*
Homework 6b

1. Use the pumping lemma, show each are non-regular
   - i. $a^n b^{n+1}$
   - ii. $a^n b^n a^n$

2. Using Myhill-Nerode theorem, show each are non-regular
   - i. EVEN-PALINDROME (all PALINDROMES with even length)
   - ii. SQUARE ($a^{n^2} \mid n \geq 1$)

3. Let us define PARENTHESES to be the set of all algebraic expressions where everything but parentheses have been deleted.
   - e.g. \{\lambda, ((), (), ((())), ()(()), (()()), ()()(), \ldots\}

   - 1. Show its non-regular using Myhill-Nerode
   - 2. Show the pumping lemma can’t prove that it’s non-regular
   - 3. If we convert ( to a and ) to b, show that PARENTHESES becomes a subset of EQUAL in which each word has the property that when read from left-to-right, there are never more b’s than a’s