The Pumping Lemma
For Regular Languages

Proof that Computer Scientists should not be allowed to come up with names for anything. Ever.
A Quick Perspective

How do we prove that a Language is Regular?

How do we prove that a Language is NOT Regular?
A Quick Perspective

How do we prove that a Language is Regular?

How do we prove that a Language is NOT Regular?
Examples of Nonregular Languages

- $B = \{0^n1^n \mid n \geq 0\}$
- $C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$
- $D = \{w \mid w \text{ has an equal number of occurrences of 01 and 10 as substrings}\}.$

OOPS!

$D$ is Regular!!!! Intuition may be wrong.

How do we keep from getting fooled?
The Pumping Lemma

We need a tool to prove that a language is NOT Regular.

Intuition: There is a property that ALL Regular Languages have. If a Language can be shown to NOT have this property, then that Language is NOT Regular.

Question: What is the longest string that can be generated by this FSM (stopping when a loop is formed in the $\delta$ transitions)?

Answer: 3 characters

(3 transitions: $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_2$)
Important Ideas/Definitions

Pumping
Repeating a section of the string an arbitrary number of times (≥0), with the resulting string remaining in the language.

Pumping Length
“All string in the language can be pumped if they are at least as long as a certain special value, called the pumping length.”

Intuitive Examples:
101
⇒ 1(01)*
000001
⇒ (0)*00001
110100
⇒ 1(1)*0100
Pumping Lemma (for Regular Languages)

If $A$ is a Regular Language, then there is a number $p$ (the pumping length) where if $s$ is any string in $A$ of length at least $p$, then $s$ may be divided into 3 pieces, $s = xyz$, satisfying the following conditions:

a. For each $i \geq 0$, $xy^iz \in A$,

b. $|y| > 0$, and

c. $|xy| \leq p$. 
Pumping Lemma (RL) Proof

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA recognizing $A$ and $p$ be the number of states of $M$.

Let $s = s_1 s_2 \ldots s_n$ be a string in $A$ with length $n$, where $n \geq p$.

Let $r_1, \ldots, r_{n+1}$ be the sequence of states $M$ enters when processing $s$. $r_{i+1} = \delta(r_i, s_i)$ for $1 \leq i \leq n$. The sequence has length $n + 1$, which is at least $p + 1$.

Among the first $p + 1$ elements in the sequence, two must be the same state, via the pigeonhole principle. The first is called $r_j$, and the second is $r_l$.

Because $r_j$ occurs among the first $p + 1$ places in a sequence starting at $r_1$, we have $l \leq p + 1$.

Now let $x = s_1 \ldots s_{j-1}$, $y = s_j \ldots s_{l-1}$, and $z = s_l \ldots s_n$.

As $x$ takes $M$ from $r_1$ to $r_j$, $y$ takes $M$ from $r_j$ to $r_l$, and $z$ takes $M$ from $r_l$ to $r_{n+1}$, which is an accept state. Because $r_j = r_l$, $M$ must accept $xy^iz$ for $i \geq 0$.

We know $j \neq l$, so $|y| > 0$; and $l \leq p + 1$, so $|xy| \leq p$.

Thus, we have satisfied all conditions of the pumping lemma. $\square$
Example applications of the Pumping Lemma (RL)

\[ B = \{ \theta^n1^n \mid n \geq 0 \} \]

Is this Language a Regular Language?
- If Regular, build a FSM
- If Nonregular, prove using the Pumping Lemma

Proof by Contradiction:
Assume \( B \) is Regular, then Pumping Lemma must hold.

\( p \) is the pumping length given by the PL.

Choose \( s \) to be \( \theta^p1^p \).

Because \( s \in B \) and \( |s| \geq p \), PL guarantees \( s \) can be split into 3 pieces, \( s = xyz \), where for any \( i \geq 0 \), \( xy^iz \in B \). Consider 3 cases:
1. \( y \) is only \( \theta \)s. \( xyyz \) has more \( \theta \)s than \( 1 \)s, thus a contradiction via condition 1 of PL.
2. \( y \) is only \( 1 \)s. Also a contradiction.
3. \( y \) is both \( \theta \)s and \( 1 \)s. \( xyyz \) may have same number of \( 1 \)s and \( \theta \)s, but will be out of order, with some \( 1 \)s before \( \theta \)s, also a contradiction.

Contradiction is unavoidable, thus \( B \) is not Regular.

(Could simplify with condition 3 of PL).
Example applications of the Pumping Lemma (RL)

\( C = \{ w \mid w \text{ has an equal number of } 0\text{s and } 1\text{s} \} \)

Is this Language a Regular Language?
- If Regular, build a FSM
- If Nonregular, prove with Pumping Lemma

Proof by Contradiction:
Assume \( C \) is Regular, then Pumping Lemma must hold.

\( p \) is the pumping length given by the PL.

Choose \( s \) to be \( \theta^p 1^p \).

Because \( s \in C \) and \( |s| \geq p \), PL guarantees \( s \) can be split into 3 pieces, \( s = xyz \), where for any \( i \geq 0 \), \( xy^i z \in C \).

\( \text{Condition 3 (} |xy| \leq p \text{) keeps us from setting } x \text{ and } z \text{ to } \varepsilon, \text{ because then } |y| \text{ would be greater than } p. \)

Because \( |xy| \leq p \), \( y \) must be only \( \theta \text{s} \). \( xyyz \) is not in the language because then the string would contain more \( \theta \text{s} \) than \( 1\text{s} \). This is a contradiction.

Contradiction is unavoidable, thus \( C \) is not Regular.

\( \boxed{\ } \)

Note: \( (01)^p \) is a bad choice, because it can be pumped.
Example applications of the Pumping Lemma (RL)

\[ F = \{ww \mid w \in \{0, 1\}^*\} \]

Is this Language a Regular Language?
- If Regular, build a FSM
- If Nonregular, prove with Pumping Lemma

Proof by Contradiction:
Assume \( F \) is Regular, then Pumping Lemma must hold.

\( p \) is the pumping length given by the PL.

Choose \( s \) to be \( 0^p10^p1 \). (Note: \( 0^p0^p \) is a bad choice.)

Because \( s \in F \) and \( |s| \geq p \), PL guarantees \( s \) can be split into 3 pieces, \( s = xyz \), where for any \( i \geq 0 \), \( xy^iz \in F \).

Because \( |xy| \leq p \), \( y \) must be only \( 0s \). \( xy^iz \) is not in the language because the string will no longer be in the form \( ww \).

Contradiction is unavoidable, thus \( F \) is not Regular. ∎

(Notice the importance of Condition 3 of PL, which keeps \( x \) and \( z \) from both being \( \varepsilon \).)
General Form of a Proof by Contradiction with the PL

Assume that a Language $L$ is Regular, then Pumping Lemma must hold.

Define $p$ to be the pumping length given by the Pumping Lemma.

Choose $s$ (often in terms of $p$).

Because $s \in L$ and $|s| \geq p$, PL guarantees that $s$ can be split into 3 pieces, $s = xyz$, where for any $i \geq 0$, $xy^iz \in L$.

For all possible values of $y$ (given the conditions of the Pumping Lemma), show that pumping $xy^iz$ is not in the Language.

Contradiction is shown for all cases, proving that $L$ is not a Regular Language.

Notes:
- Choosing a $s$ that can be pumped proves nothing.
- Sometimes finding an appropriate $s$ is the hard part.