The Limits of Regular Languages

Lecture 8

26 September 2019
Assignments:

Assignment 2 is out

Soon: (Mini) Assignment 3

Previously:

Introduced regular languages and three equivalent models of computation for recognizing them (DFAs, NFAs, REs)

Today:

What languages aren’t regular? How do we prove it?
Or, Beyond the Valley of the Regular Languages
THEOREM. The following are all equivalent:

$L$ is a regular language.

There is a DFA $D$ such that $L(D) = L$.

There is an NFA $N$ such that $L(N) = L$.

There is a regular expression $R$ such that $L(R) = L$. 
PROOF.

\[
\text{Regular Expression: } \left((a\cup ba^*)\cup ca^*\right)^* \cup ab^*(cub)^*
\]

- NFA,
- NFA-\(\epsilon\)
- Subset construction
- DFA
- Thompson’s algorithm
- State elimination
Why does this matter?

DFAs correspond to computers with finite memory.

The equivalence of DFAs and NFAs tells us that given finite memory, nondeterminism doesn’t increase computational power.

(Though it might save on memory.)

The equivalence of DFAs and regular expressions tells us that all problems solvable by finite computers can be assembled out of smaller building blocks.
Is every language regular?
To prove a language is regular, we can construct a DFA, NFA, or RE to recognize it.

Or directly use known closure properties.
To prove a language is not regular, we need to show that it’s not possible to construct a DFA, NFA, or regular expression for it.

This kind of argument is challenging – how can we show that we wouldn’t be able to devise a finite automaton for it if we tried harder?
Proof strategy: *Infinite states*
A simple language

Let $\Sigma = \{a, b\}$ and consider this language:

$$L = \{a^i b^i \mid i \in \mathbb{N}_0\}$$

$L$ is the language of all strings of $i$ $a$s followed by $i$ $b$s:

$$\{\varepsilon, ab, aabb, aaabbb, \ldots\}$$

Is this language regular?
How many states are needed to recognize \{a^i b^i\}?
This language is not regular!

Intuitive explanation:

Imagine a finite automaton to accept this language. When any DFA for L is run on any two of the strings $\varepsilon$, $a$, $aa$, $aaa$, $aaaa$, etc., the DFA must end in different states.

Suppose $a^n$ and $a^m$ end up in the same state, where $m \neq n$.

Then $a^n b^n$ and $a^m b^n$ will end up in the same state.

The DFA will either accept a string not in the language or reject a string in the language, which it shouldn’t be able to do.

We can’t place all these strings into different states; there are only finitely many states!
Formal proof

Consider the language \( \{a^ib^i \mid i \geq 0\} \) and a DFA to recognize it.

For any \( i \), let \( q_i \) be the state entered after processing \( a^i \), i.e.,
\[ \hat{\delta}(q_0, a^i) = q_i \]

Consider any \( i \) and \( j \) such that \( i \neq j \).

\[ \hat{\delta}(q_0, a^ib^i) \neq \hat{\delta}(q_0, a^ib^i) \] since the former is accepting and the latter is rejecting.

\[ \hat{\delta}(q_0, a^ib^i) = \hat{\delta}(\hat{\delta}(q_0, a^i), b^i) = \hat{\delta}(a_i, b^i), \] by definition of \( \hat{\delta} \) and definition of \( a_i \), respectively.

\[ \hat{\delta}(q_0, a^ib^i) = \hat{\delta}(\hat{\delta}(q_0, a^i), b^i) = \hat{\delta}(a_j, b^i), \] by the same reasoning.
Since inputs $a^i$ and $a^j$ lead to different states on the same input, the states must be different: $q_i \neq q_j$.

Since $i$ and $j$ were arbitrary, and since there are an infinite number of ways to pick them, there must be an infinite number of states.

Thus there is no DFA to recognize this language, and the language is non-regular.
The intuition for this proof is helpful to think about. However, actually writing one of these proofs becomes difficult for more complicated languages.

We can take the idea of the number of states required for a DFA to recognize a language and develop a powerful proof framework.
Proof strategy: The Pumping Lemma
An important observation
Visiting multiple states

Let $D$ be a DFA with $n$ states.

Any string $w$ accepted by $D$ that has length at least $n$ must visit some state twice within its first $n$ characters.

Number of states visited is equal to $n + 1$.

By the Pigeonhole Principle, some state is duplicated.

The substring of $w$ between those revisited states can be removed, duplicated, tripled, etc. without changing the fact that $D$ accepts $w$. 
Informally

Let $L$ be a regular language.

If we have a string $w \in L$ that is “sufficiently long”, then we can split the string into three pieces and “pump” the middle:

We can write $w = xyz$ such that $xy^0z$, $xy^1z$, $xy^2z$, … are all in $L$. 
The Pumping Lemma for Regular Languages

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x$, $y$, and $z$ such that

$$w = xyz$$

$$|xy| \leq n$$

$$y \neq \varepsilon$$

$$xy^iz \in L \text{ for all } i \geq 0$$
The Pumping Lemma for Regular Languages

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x$, $y$, and $z$ such that $w = xyz$ with $|xy| \leq n$, $y \neq \varepsilon$, and $xy^iz \in L$ for all $i \geq 0$. The "pumping length" is $n$. 
The Pumping Lemma for Regular Languages

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x$, $y$, and $z$ such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and $xy^iz \in L$ for all $i \geq 0$. Strings longer than the pumping length must have a special property.
The Pumping Lemma for Regular Languages

For every regular language L,

there exists a positive integer n such that

for every string w ∈ L with |w| ≥ n,

there exist strings x, y, and z such that

w = xyz

|xy| ≤ n

y ≠ ε

xy^iz ∈ L for all i ≥ 0
The Pumping Lemma for Regular Languages

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x, y, z$ such that

$$w = xyz$$

$|xy| \leq n$

$y \neq \epsilon$

$xy^iz \in L$ for all $i \geq 0$

w can be broken into three pieces
The Pumping Lemma for Regular Languages

For every regular language \( L \), there exists a positive integer \( n \) such that for every string \( w \in L \) with \( |w| \geq n \), there exist strings \( x, y, \) and \( z \) such that

\[
w = xyz
\]

\[
|xy| \leq n
\]

\[
y \neq \varepsilon
\]

\[
xy^iz \in L \text{ for all } i \geq 0
\]
The Pumping Lemma for Regular Languages

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x, y, z$ such that

$$w = xyz$$

$$|xy| \leq n$$

$$y \neq \varepsilon$$

$xy^iz \in L$ for all $i \geq 0$
The Pumping Lemma for Regular Languages

For every regular language \( L \),

there exists a positive integer \( n \) such that

for every string \( w \in L \) with \( |w| \geq n \),

there exist strings \( x, y, \) and \( z \) such that

\[
    w = xyz
\]

\[
    |xy| \leq n
\]

\[
    y \neq \varepsilon
\]

\[
    xy^iz \in L \text{ for all } i \geq 0
\]

\( w \) can be broken into three pieces

where the first two pieces are no longer than the pumping length

the middle part isn’t empty

and the middle piece can be replicated zero or more times
Rationale for requirements in the Pumping Lemma

\( y \neq \varepsilon \)

Because \( y \) labels the loop, it has to consist of at least one symbol.

\( |xy| \leq n \)

Because \( xy \) is what you get when you take the loop once.

\( xy^iz \in L \) for all \( i \geq 0 \)

Because \( y \) can be *pumped* zero or more times.
The Pumping Lemma gets its name because the repeated string is “pumped”.

Note that because of the nature of FAs, we cannot control the number of times it is pumped.

So, a regular language with strings of length \( \geq n \) is always infinite!

The Pumping Lemma is only interesting for infinite languages.

But it works for finite languages, which are always regular.

For finite languages, \( n \) is larger than the longest string, so nothing can be pumped.
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$.

Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped".

1 0 0 1 0
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

\[\text{1 0 0 1 0} \quad \text{1 0 0 1 0}\]
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* | w$ contains $00$ as a substring$\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped".
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w$ contains $00$ as a substring$\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

1 0 0 1 0
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

1 0 0 1 1 0
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

1 0 0 1 1 1 0
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped".
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

The first piece is just the empty string! This is perfectly fine.
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

\[
\begin{array}{c}
1 \\
0 \\
0 \\
\end{array}
\]
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

\[
\begin{array}{c c c}
1 & 1 & 0 \\
\end{array}
\]
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w$ contains $00$ as a substring$\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”. 

1 1 1 0 0
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

1 1 0 0 0 0 1
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

\[\begin{array}{ccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}\]
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w$ contains $00$ as a substring$\}$.

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”. 

1 1 0 0
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.

```
1 1 0 0 0 0 1
```
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w$ contains $00$ as a substring$\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped”.
Let $\Sigma = \{0, 1\}$ and
$L = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$
Let \( \Sigma = \{0, 1\} \) and 
\[ L = \{ \varepsilon, 0, 1, 00, 01, 10, 11 \} \]

Any string of length three or greater can be split into three pieces, the second of which can be “pumped”.
Let $\Sigma = \{0, 1\}$ and $L = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$

Any string of length three or greater can be split into three pieces, the second of which can be “pumped”.

The Pumping Lemma holds for finite languages because the pumping length can be longer than the longest string!
Testing equality

The equality problem is defined as follows: Given two strings, $x$ and $y$, decide if $x = y$.

Let $\Sigma = \{0, 1, ?\}$. We can encode the equality problem as a string of the form $x?y$.

“Is 001 equal to 110?” would be encoded as 001?110

“Is 11 equal to 11?” would be encoded as 11?11

“Is 110 equal to 110?” would be encoded as 110?110

Let $EUAL = \{w?w \mid w \in \{0, 1\}^*\}$

QUESTION. Is $EUAL$ a regular language?
The Pumping Lemma for Regular Languages

For every regular language \( L \), there exists a positive integer \( n \) such that for every string \( w \in L \) with \( |w| \geq n \), there exist strings \( x, y, \) and \( z \) such that

\[
w = xyz
\]

\[
|xy| \leq n
\]

\[
y \neq \varepsilon
\]

\[
xyz^i \in L \text{ for all } i \geq 0
\]
Using the Pumping Lemma

\[ \text{EQUAL} = \{w?w \mid w \in \{0, 1\}^*\} \]
Using the Pumping Lemma

\[ \text{EQUAL} = \{w?w \mid w \in \{0, 1\}^* \} \]
Using the Pumping Lemma

\[ \text{EQUAL} = \{w\overline{w} \mid w \in \{0, 1\}^*\} \]
Using the Pumping Lemma

\[ EQUAL = \{w?w \mid w \in \{0, 1\}^*\} \]
Using the Pumping Lemma

\[ \text{EQUAL} = \{ w?w \mid w \in \{0, 1\}^* \} \]
Using the Pumping Lemma

$$EQUAL = \{w?w \mid w \in \{0, 1\}^*\}$$
Using the Pumping Lemma

\[ \text{EQUAL} = \{w \text{?} w \mid w \in \{0, 1\}^*\} \]
Using the Pumping Lemma

\[ \text{EQUAL} = \{w?w \mid w \in \{0, 1\}^*\} \]
Using the Pumping Lemma

\[ \text{EQUAL} = \{w?w \mid w \in \{0, 1\}^*\} \]
Using the Pumping Lemma

$$EQUAL = \{ w?w \mid w \in \{0, 1\}^* \}$$
Using the Pumping Lemma

$EQUAL = \{w?w \mid w \in \{0, 1\}^*\}$
Using the Pumping Lemma

\[ EQUAL = \{w?w \mid w \in \{0, 1\}^*\} \]
What’s going on?

The Pumping Lemma says that for “sufficiently long” strings, we should be able to pump some part of the string.

We can’t pump any part containing the ? because we can’t duplicate it or remove it.

We can’t pump just one part of the string because then the strings on opposite sides of the ? wouldn’t match.

Can we formally show that EQUAL is not regular?
THEOREM. $EQUAL$ is not regular.

PROOF. By contradiction; assume that $EQUAL$ is regular.
**THEOREM.** EQUAL is not regular.

**PROOF.** By contradiction; assume that EQUAL is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma.

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x, y, z$ such that

- $w = xyz$
- $|xy| \leq n$
- $y \neq \varepsilon$
- $xy^i z \in L$ for all $i \geq 0$
THEOREM. **EQUAL** is not regular.

PROOF. By contradiction; assume that **EQUAL** is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma.

For every regular language \( L \),

there exists a positive integer \( n \) such that

for every string \( w \in L \) with \( |w| \geq n \),

there exist strings \( x, y, \) and \( z \) such that

\[
\begin{align*}
    w &= xyz \\
    |xy| &\leq n \\
    y &\neq \varepsilon \\
    xy^i z \in L \text{ for all } i \geq 0
\end{align*}
\]
THEOREM. EQUAL is not regular.

PROOF. By contradiction; assume that EQUAL is regular. Let n be the pumping length guaranteed by the Pumping Lemma.
THEOREM. EQUAL is not regular.

PROOF. By contradiction; assume that EQUAL is regular. Let n be the pumping length guaranteed by the Pumping Lemma.

The hardest part of most Pumping Lemma proofs is choosing a string that we should be able to pump but cannot.
THEOREM. \textit{EQUAL} is not regular.

PROOF. By contradiction; assume that \textit{EQUAL} is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma. Let \( w = 0^n?0^n \).
For every regular language $L$, there exists a positive integer $n$ such that

for every string $w \in L$ with $|w| \geq n$,

there exist strings $x$, $y$, and $z$ such that

$w = xyz$

$|xy| \leq n$

$y \neq \varepsilon$

$xy^iz \in L$ for all $i \geq 0$

**THEOREM.** $\textsc{Equal}$ is not regular.

**PROOF.** By contradiction; assume that $\textsc{Equal}$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n?0^n$. 
THEOREM. \textit{EQUAL} is not regular.

PROOF. By contradiction; assume that \textit{EQUAL} is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n 1^n$. Then $w \in \text{EQUAL}$ and $|w| = 2n + 1 \geq n$. 

\begin{center}
\begin{tabular}{l}
For every regular language $L$, \\
there exists a positive integer $n$ such that \\
\textit{for every string} $w \in L \text{ with } |w| \geq n$, \\
there exist strings $x, y, \text{ and } z$ such that \\
$w = xyz$ \\
$|xy| \leq n$ \\
y $\neq \varepsilon$ \\
$xy^iz \in L \text{ for all } i \geq 0$
\end{tabular}
\end{center}
THEOREM. $EQUAL$ is not regular.

PROOF. By contradiction; assume that $EQUAL$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. 

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x$, $y$, and $z$ such that $w = xyz$ $|xy| \leq n$ $y \neq \varepsilon$ $xy^iz \in L$ for all $i \geq 0$
For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$,

there exist strings $x$, $y$, and $z$ such that

\[
    w = xyz \\
    |xy| \leq n \\
    y \neq \epsilon \\
    xy^iz \in L \text{ for all } i \geq 0
\]

**THEOREM.** $EQUAL$ is not regular.

**PROOF.** By contradiction; assume that $EQUAL$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n10^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. 
For every regular language \( L \), there exists a positive integer \( n \) such that for every string \( w \in L \) with \( |w| \geq n \), there exist strings \( x, y, \) and \( z \) such that
\[
\begin{align*}
 w &= xyz \\
 |xy| &\leq n \\
 y &\neq \epsilon \\
 xyz^i &\in L \text{ for all } i \geq 0
\end{align*}
\]

**THEOREM.** \( EQUAL \) is not regular.

**PROOF.** By contradiction; assume that \( EQUAL \) is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma. Let \( w = 0^n?0^n \). Then \( w \in EQUAL \) and \( |w| = 2n + 1 \geq n \). Thus by the Pumping Lemma, we can write \( w = xyz \) such that \( y \neq \epsilon \) and for any \( i \in \mathbb{N} \), \( xy^iz \in EQUAL \).
THEOREM. $EQUAL$ is not regular.

PROOF. By contradiction; assume that $EQUAL$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the Pumping Lemma, we can write $w = xyz$ such that $|xy| \leq n$ and $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. 

For every regular language $L$,

there exists a positive integer $n$ such that

for every string $w \in L$ with $|w| \geq n$,

there exist strings $x$, $y$, and $z$ such that

$w = xyz$

$|xy| \leq n$

$y \neq \varepsilon$

$xy^iz \in L$ for all $i \geq 0$
For every regular language \( L \),

there exists a positive integer \( n \) such that

for every string \( w \in L \) with \( |w| \geq n \),

there exist strings \( x, y, \) and \( z \) such that

\[
\begin{align*}
  w &= xyz \\
  |xy| &\leq n \\
  y &\neq \varepsilon \\
  xy^i z &\in L \text{ for all } i \geq 0
\end{align*}
\]

**THEOREM.** \( \text{EQUAL} \) is not regular.

**PROOF.** By contradiction; assume that \( \text{EQUAL} \) is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma. Let \( w = 0^n1^0^n \). Then \( w \in \text{EQUAL} \) and \( |w| = 2n + 1 \geq n \). Thus by the Pumping Lemma, we can write \( w = xyz \) such that \( |xy| \leq n \) and \( y \neq \varepsilon \) and for any \( i \in \mathbb{N} \), \( xy^i z \in \text{EQUAL} \).

**At this point, we have some string that we should be able to split into pieces and pump. The rest of the proof shows that no matter what choice we made, the middle can’t be pumped.**
THEOREM. $EQUAL$ is not regular.

PROOF. By contradiction; assume that $EQUAL$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the Pumping Lemma, we can write $w = xyz$ such that $|xy| \leq n$ and $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. The string $y$ must consist only of $0$s before the $?$ or it would violate that $|xy| \leq n$. 

For every regular language $L$,

there exists a positive integer $n$ such that

for every string $w \in L$ with $|w| \geq n$,

there exist strings $x, y, z$ such that

$w = xyz$

$|xy| \leq n$

$y \neq \varepsilon$

$xy^iz \in L$ for all $i \geq 0$
THEOREM. $EQUAL$ is not regular.

PROOF. By contradiction; assume that $EQUAL$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the Pumping Lemma, we can write $w = xyz$ such that $|xy| \leq n$ and $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. The string $y$ must consist only of $0$s before the $?$ or it would violate that $|xy| \leq n$. Therefore, $xy^0z = 0^m?0^n$, where $m < n$, and is not in $L$. 

For every regular language $L$, there exists a positive integer $n$ such that for every string $w \in L$ with $|w| \geq n$, there exist strings $x$, $y$, and $z$ such that $w = xyz$ $|xy| \leq n$ $y \neq \varepsilon$ $xy^iz \in L$ for all $i \geq 0$
For every regular language $L$,
there exists a positive integer $n$ such that
for every string $w \in L$ with $|w| \geq n$,
there exist strings $x$, $y$, and $z$ such that

$$w = xyz$$

$$|xy| \leq n$$

$$y \neq \varepsilon$$

$$xy^iz \in L$$ for all $i \geq 0$

THEOREM. $EQUAL$ is not regular.

PROOF. By contradiction; assume that $EQUAL$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the Pumping Lemma, we can write $w = xyz$ such that $|xy| \leq n$ and $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. The string $y$ must consist only of $0$s before the $?$ or it would violate that $|xy| \leq n$. Therefore, $xy^0z = 0^m?0^n$, where $m < n$, and is not in $L$. This contradicts the Pumping Lemma, so our assumption was wrong. Thus $EQUAL$ is not regular. □
Non-regular languages

The Pumping Lemma describes a property common to all regular languages.

Any language $L$ that does not have this property cannot be regular.

What other languages can we find that are not regular?
A canonical non-regular language

Recall the language $L = \{a^i b^i \mid i \in \mathbb{N}_0\}$.

$L = \{\varepsilon, \text{ab}, \text{aabb}, \text{aaabbb}, \text{aaaabbbb}, \ldots\}$

$L$ is a classic example of a non-regular language.

Intuitively, since we can only have finitely many states in a DFA, we can’t “remember” an arbitrary number of $a$s.

How could we use the Pumping Lemma to prove $L$ is non-regular?
The Pumping Lemma ““game””

You can think of a Pumping Lemma proof as a game between you and an adversary.

You win by finding a contradiction of the Pumping Lemma for the given language.

The adversary wins if they can make a choice for which the Pumping Lemma succeeds.

The game goes as follows:

The adversary chooses a pumping length $n$.

You choose a string $w$ with $|w| \geq n$ and $w \in L$.

The adversary break it into $x$, $y$, and $z$ such that $|xy| \leq n$ and $y \neq \varepsilon$.

You choose an $i$ such that $xy^iz \not\in L$. (If you can’t, you lose!)
Gameplay Magazine described the rules as “punishingly intricate”.
The Pumping Lemma Game

Adversary

You
The Pumping Lemma Game

**Adversary**

1. Maliciously choose pumping length $n$

**You**
The Pumping Lemma Game

**Adversary**

1. Maliciously choose pumping length $n$

**You**

2. Cleverly choose a string $w \in L$, $|w| \geq n$
The Pumping Lemma Game

**Adversary**

1. Maliciously choose pumping length $n$

3. Maliciously split $w = xyz$ with $y \neq \epsilon$ and $|xy| \leq n$

**You**

2. Cleverly choose a string $w \in L$, $|w| \geq n$
The Pumping Lemma Game

**Adversary**

1. Maliciously choose pumping length \( n \)

3. Maliciously split \( w = xyz \) with \( y \neq \varepsilon \) and \( |xy| \leq n \)

**You**

2. Cleverly choose a string \( w \in L, |w| \geq n \)

4. Cleverly choose \( i \) such that \( xy^iz \notin L \)
The Pumping Lemma Game

**Adversary**

1. Maliciously choose pumping length $n$

3. Maliciously split $w = xyz$ with $y \neq \varepsilon$ and $|xy| \leq n$

5. I’ll get you next time, Gadget!

Next time!

**You**

2. Cleverly choose a string $w \in L$, $|w| \geq n$

4. Cleverly choose $i$ such that $xy^iz \not\in L$
THEOREM. $L = \{a^ib^i \mid i \in \mathbb{N}_0\}$ is not regular.
THEOREM. $L = \{a^ib^i \mid i \in \mathbb{N}_0\}$ is not regular.

PROOF. By contradiction; assume $L$ is regular.
THEOREM. \( L = \{a^ib^i \mid i \in \mathbb{N}_0\} \) is not regular.

PROOF. By contradiction; assume \( L \) is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma.
THEOREM. $L = \{a^i b^i \mid i \in \mathbb{N}_0\}$ is not regular.

PROOF. By contradiction; assume $L$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Consider the string $w = a^n b^n$. 
THEOREM. \( L = \{a^ib^i \mid i \in \mathbb{N}_0\} \) is not regular.

PROOF. By contradiction; assume \( L \) is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma. Consider the string \( w = a^n b^n \). Then \( |w| = 2n \geq n \) and \( w \in L \), so we can break this string into \( w = xyz \), where \( |xy| \leq n \) and \( y \neq \varepsilon \), and for any \( i \in \mathbb{N}_0 \), the string \( xy^iz \in L \).
THEOREM. $L = \{a^ib^i \mid i \in \mathbb{N}_0\}$ is not regular.

PROOF. By contradiction; assume $L$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Consider the string $w = a^n b^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can break this string into $w = xyz$, where $|xy| \leq n$ and $y \neq \varepsilon$, and for any $i \in \mathbb{N}_0$, the string $xy^iz \in L$. Because $|xy| \leq n$ and $|y| > 0$, the string $y$ has to consist only of $a$s.
THEOREM. $L = \{a^i b^i \mid i \in \mathbb{N}_0\}$ is not regular.

PROOF. By contradiction; assume $L$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Consider the string $w = a^n b^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can break this string into $w = xyz$, where $|xy| \leq n$ and $y \neq \varepsilon$, and for any $i \in \mathbb{N}_0$, the string $xy^i z \in L$. Because $|xy| \leq n$ and $|y| > 0$, the string $y$ has to consist only of $a$s. So, no matter what segment of the string $xy$ covers, pumping $y$ adds to the number of $a$s, hence there are more $a$s than $b$s.
THEOREM. $L = \{a^ib^i \mid i \in \mathbb{N}_0\}$ is not regular.

PROOF. By contradiction; assume $L$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Consider the string $w = a^n b^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can break this string into $w = xyz$, where $|xy| \leq n$ and $y \neq \varepsilon$, and for any $i \in \mathbb{N}_0$, the string $xy^iz \in L$. Because $|xy| \leq n$ and $|y| > 0$, the string $y$ has to consist only of $a$s.

So, no matter what segment of the string $xy$ covers, pumping $y$ adds to the number of $a$s, hence there are more $a$s than $b$s.

There is no way to segment $w$ into $xyz$ that can’t be pumped to produce a string that isn’t in the language.
THEOREM. $L = \{a^ib^i \mid i \in \mathbb{N}_0\}$ is not regular.

PROOF. By contradiction; assume $L$ is regular. Let $n$ be the pumping length guaranteed by the Pumping Lemma. Consider the string $w = a^nb^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can break this string into $w = xyz$, where $|xy| \leq n$ and $y \neq \varepsilon$, and for any $i \in \mathbb{N}_0$, the string $xy^iz \in L$. Because $|xy| \leq n$ and $|y| > 0$, the string $y$ has to consist only of $a$s.

So, no matter what segment of the string $xy$ covers, pumping $y$ adds to the number of $a$s, hence there are more $a$s than $b$s.

There is no way to segment $w$ into $xyz$ that can’t be pumped to produce a string that isn’t in the language.

Contradiction! Therefore, $L$ is not regular. ■
Critical point

It’s necessary to show there is no segmentation of the chosen string that won’t lead to a contradiction.

This means considering every possible mapping of xy onto the first $n$ symbols in the chosen string.

We chose our string to make this easy, since every possible segmentation consists of as only.

Pumping therefore disrupts the equivalence of the number of as and bs.
Critical point

We only need to show that there’s one string in the language for which the Pumping Lemma doesn’t work.

For some strings in $L$, it may work perfectly well!
The Pumping Lemma mascot, the Pumping Llama
by Kimberly Do
Example: $L = \{ww^R\}$ is not regular

We use $w^R$ to denote $w$ reversed. Let $\Sigma = \{a, b\}$.

Whatever $n$ is, we can always choose a $w$ as follows:

Because of this choice and the requirement that $|xy| \leq n$, the opponent is forced to choose a $y$ that consists entirely of $a$s.

In Step 4, we use $i = 2$; the string $xy^2z$ has more $a$s on the left than on the right, so it cannot be of form $ww^R$.

Therefore $L$ is not regular!
Example: $L = \{ww^R\}$ is not regular

We use $w^R$ to denote $w$ reversed. Let $\Sigma = \{a, b\}$.

Whatever $n$ is, we can always choose a $w$ as follows:

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
\[a\ldots ab\ldots bb\ldots ba\ldots a|\]
\[x \quad y \quad z\]
```

Because of this choice and the requirement that $|xy| \leq n$, the opponent is forced to choose a $y$ that consists entirely of $a$s.

In Step 4, we use $i = 2$; the string $xy^2z$ has more $a$s on the left than on the right, so it cannot be of form $ww^R$.

Therefore $L$ is not regular!
Example

Consider the alphabet $\Sigma = \{0, 1\}$ and the language

$$BALANCE = \{w \mid w \text{ has an equal number of } 1\text{s and } 0\text{s}\}$$

E.g.,

$$01 \in BALANCE$$
$$110010 \in BALANCE$$
$$11011 \notin BALANCE$$

Is $BALANCE$ a regular language?
An incorrect proof

**Theorem:** BALANCE is regular.

**Proof:** We show that BALANCE satisfies the condition of the Pumping Lemma. Let $n = 2$ and consider any string $w \in \text{BALANCE}$ such that $|w| \geq 2$. Then we can write $w = xyz$ such that $x = z = \varepsilon$ and $y = w$, so $y \neq \varepsilon$. Then for any natural number $i$, $xy^iz = w^i$, which has the same number of 0s and 1s. Since BALANCE passes the conditions of the Pumping Lemma, BALANCE is regular.
For every regular language $L$, there exists an integer $n$ such that for every string $w \in L$ of length $n$ or more, there exist strings $x$, $y$, and $z$ such that

\[ w = xyz \]
\[ |xy| \leq n \]
\[ y \neq \varepsilon \]
\[ xy^iz \in L \text{ for all } i \geq 0 \]
Caution with the Pumping Lemma

The Pumping Lemma describes a necessary condition of regular languages.

If $L$ is regular, $L$ passes the conditions of the Pumping Lemma.

The Pumping Lemma is not a sufficient condition to be a regular language.

If $L$ is not regular, it still might pass the conditions of the Pumping Lemma!
Example: $L = \{w \mid w$ has an equal number of $1$s and $0$s$\}$ is not regular

Given $n$, we choose the string $(01)^n$.

We need to show splitting this string into $xyz$ where $xy^i z$ is in $L$ is impossible…

*But it is possible!*

If $x = \varepsilon$, $y = 01$, and $z = (01)^{n-1}$, $xy^i z$ is in $L$ for every value of $i$

*Are we out of luck?*
When using the Pumping Lemma:

*If your string does not succeed, try another!*
Let’s try $1^n0^n$.

Again, we need to show splitting this string into $xyz$ where $xy^iz$ is in $L$ is impossible…
Let’s try $1^n0^n$.

Again, we need to show splitting this string into $xyz$ where $xy^iz$ is in $L$ is impossible…

*But it is possible!*

If $x$ and $z$ are $\varepsilon$ and $y$ is $1^n0^n$, then $xy^iz$ always has an equal number of 0s and 1s
Let’s try $1^n0^n$.

Again, we need to show splitting this string into $xyz$ where $xy^iz$ is in $L$ is impossible…

**But it is possible!**

If $x$ and $z$ are $\varepsilon$ and $y$ is $1^n0^n$, then $xy^iz$ always has an equal number of $0$s and $1$s

Are we still in trouble?
Not this time…

The Pumping Lemma says that our string has to be divided so that $|xy| \leq n$ and $|y| > 0$

If $|xy| \leq n$, then $y$ must consist only of $1$s, so $x y y z \not\in L$.

Contradiction! We win!
Example:
$L = \{ww \mid w \in \Sigma^*\}$ is not regular

We choose the string $a^nba^n$, where $n$ is the number of states in the FA.

We now show that there is no decomposition of this string into $xyz$ where for any $i \geq 0$, $xy^iz$ is in $L$.

Since $|xy| \leq n$, it’s easy to show that the Pumping Lemma won’t hold for $L$ because $y$ must consist only of $a$s, so $xyyz$ is not in $L$. 
In the previous example, as before, the choice of string is critical.

Had we chosen $a^n a^n$ (which is a member of $L$) instead of $a^n b a^n b$, it wouldn’t work because it can be pumped and still satisfy the Pumping Lemma.

**Moral:** *Choose your strings wisely.*
Example:

\[ L = \{0^i1^j \mid i > j\} \] is not regular
Example:
\[ L = \{0^i1^j | i > j\} \text{ is not regular} \]

Given \( n \), choose \( s = 0^{n+1}1^n \)
Example:
$L = \{0^i1^j \mid i > j\}$ is not regular

Given $n$, choose $s = 0^{n+1}1^n$

If $L$ is regular, we can split $s$ into $xyz$ where for any $i \geq 0$, $xy^iz$ is in $L$, $|xy| \leq n$, and $|y| > 0$
Example:
$L = \{0^i1^j \mid i > j\}$ is not regular

Given $n$, choose $s = 0^{n+1}1^n$

If $L$ is regular, we can split $s$ into $xyz$ where for any $i \geq 0$, $xy^iz$ is in $L$, $|xy| \leq n$, and $|y| > 0$

Because $|xy| \leq n$, $y$ consists only of 0s
Example:
$L = \{0^i1^j \mid i > j\}$ is not regular

Given $n$, choose $s = 0^{n+1}1^n$

If $L$ is regular, we can split $s$ into $xyz$ where for any $i \geq 0$, $xy^iz$ is in $L$, $|xy| \leq n$, and $|y| > 0$

Because $|xy| \leq n$, $y$ consists only of 0s

Is $x\underline{yy}z$ in $L$?
The Pumping Lemma states that $xy^iz$ is in $L$ even when $i = 0$

So, consider the string $xy^0z$

Removing string $y$ decreases the number of 0s in s
s has only one more 0 than 1
Therefore, $xz$ cannot have more 0s than 1s, and is not a member of $L$

Contradiction!

*This strategy is called “pumping down”*
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular
Example:

$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$
Example:
$L = \{ a^i \mid i \text{ is prime} \}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$
If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $x y^i z$ is in $L$ for all $i \geq 0$
Example:

$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.

If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.
Example:

$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.

If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.

The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$. If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.

The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

$$\text{length}(xy^{k+1}z) = \text{length}(xyz^ky)$$
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.
If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.
Assume such a decomposition exists.
The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

\[
\text{length}(xy^{k+1}z) = \text{length}(xyzy^k)
\]
\[
= \text{length}(xyz) + \text{length}(y^k)
\]
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.
If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.
Assume such a decomposition exists.
The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

$$\text{length}(xy^{k+1}z) = \text{length}(xyzy^k)$$
$$= \text{length}(xyz) + \text{length}(y^k)$$
$$= k + k(\text{length}(y))$$
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.

If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.

The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

$$\text{length}(xy^{k+1}z) = \text{length}(xyzy^k)$$
$$= \text{length}(xyz) + \text{length}(y^k)$$
$$= k + k(\text{length}(y))$$
$$= k(1 + \text{length}(y))$$
Example:

$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$. If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.

The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

$$\text{length}(xy^{k+1}z) = \text{length}(xyzy^k) = \text{length}(xyz) + \text{length}(y^k) = k + k(\text{length}(y)) = k(1 + \text{length}(y))$$

The length of $xy^{k+1}z$ is therefore not prime, since it is the product of two numbers other than 1. So $xy^{k+1}z$ is not in $L$. 
Example:

$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.

If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.

The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

$$\text{length}(xy^{k+1}z) = \text{length}(xyz^k)$$
$$= \text{length}(xyz) + \text{length}(y^k)$$
$$= k + k(\text{length}(y))$$
$$= k (1 + \text{length}(y))$$

The length of $xy^{k+1}z$ is therefore not prime, since it is the product of two numbers other than 1. So $xy^{k+1}z$ is not in $L$.

**Contradiction!**
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$.

If $L$ is regular, PL implies that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$.

Assume such a decomposition exists.

The length of $w = xy^{k+1}z$ must be a prime if $w$ is in $L$. But

\[
\text{length}(xy^{k+1}z) = \text{length}(xyzy^k) \\
= \text{length}(xyz) + \text{length}(y^k) \\
= k + k(\text{length}(y)) \\
= k(1 + \text{length}(y))
\]

The length of $xy^{k+1}z$ is therefore not prime, since it is the product of two numbers other than 1. So $xy^{k+1}z$ is not in $L$.

Contradiction!
Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma constant, and let $k$ be a prime greater than $n$.

If $L$ is regular, the Pumping Lemma states that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$. Assume such a decomposition exists.

Then if $w$ is in $L$, the length of $w = xy^{k+1}z$ must be a prime. However,

$$
\text{length}(xy^{k+1}z) = \text{length}(xyzy^k) \\
= \text{length}(xyz) + \text{length}(y^k) \\
= k + k(\text{length}(y)) \\
= k \(1 + \text{length}(y))
$$

The length of $xy^{k+1}z$ is therefore not prime, since it is the product of two numbers other than 1. So $xy^{k+1}z$ is not in $L$.

Contradiction!
Remember
Remember

You only need to find one string for which the Pumping Lemma does not hold to prove a language is not regular.
Remember

You only need to find one string for which the Pumping Lemma does not hold to prove a language is not regular.

But you must show that for any decomposition of that string into $xyz$ the Pumping Lemma holds.

This sometimes means considering several different cases.
The Pumping Lemma mascot, the Pumping Llama

by Kimberly Do
Acknowledgments

This lecture incorporates material from:

Nancy Ide
Keith Schwarz
Michael Sipser