The Limits of Regular Languages

20 February 2020
Assignments 1 & 2
  Graded

Assignment 3
  Corrections due now

Exam 1
  Looming
What exactly is a finite state machine?
Computers as finite automata

Computing devices have internal workings that can be in one of finitely many possible configurations.

Each state in a DFA corresponds to some possible configuration of the internal workings.

If your computer has 8 GB of RAM and 200 GB of hard drive space, that’s a total of 208 GB of memory, which is 1,786,706,395,136 bits. There are “only” $2^{1,391,569,403,904}$ possible configurations of the memory in this computer.

You could, in principle, build a DFA representing this computer, where there’s one symbol per type of input the computer can receive.
A powerful intuition

*Regular languages correspond to problems that can be solved with finite memory.*

At each point in time, we only need to store one of finitely many pieces of information.

Non-regular languages correspond to problems that cannot be solved with finite memory.

Since every computer ever built has finite memory, non-regular languages correspond to problems that cannot be solved by physical computers!
To prove a language is not regular, we can:

Argue that a finite automaton would require an infinite number of states.

Write proofs by contradiction using a result called the **Pumping Lemma** and/or **closure properties**.
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’s at least $n$ characters long must visit some state twice within the 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

\[
w = xyz
\]

\[
|xy| \leq n
\]

\[
y \neq \varepsilon
\]

\[
xy^iz \in L \text{ for all } i \in \mathbb{N}_0
\]
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

\[ w = xyz \]

\[ |xy| \leq n \]

\[ y \neq \varepsilon \]

\[ xy^iz \in L \text{ for all } i \in \mathbb{N}_0 \]
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 $x, y, \text{ and } z$ such that

$$w = xyz$$

$$|xy| \leq n$$

$$y \neq \epsilon$$

$$xy^iz \in L \text{ for all } i \in \mathbb{N}_0$$

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 \( w \in L \) with \( |w| \geq n \),

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

\[
w = xyz
\]

\[
|xy| \leq n
\]

\[
y \neq \varepsilon
\]

\[
x y^i z \in L \text{ for all } i \in \mathbb{N}_0
\]
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 $x, y, \text{ and } z$ such that $w = xyz$.

$|xy| \leq n$

$y \neq \varepsilon$

$xy^iz \in L$ for all $i \in \mathbb{N}_0$
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

$$w = xyz$$

$$|xy| \leq n$$

$$y \neq \varepsilon$$

$$xy^iz \in L \text{ for all } i \in \mathbb{N}_0$$

w can be broken into three pieces where the first two pieces occur at the start of the string
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

$$w = xyz$$

$|xy| \leq n$

$y \neq \varepsilon$

$xy^iz \in L$ for all $i \in \mathbb{N}_0$

$w$ can be broken into three pieces

where the first two pieces occur at the start of the string

the middle part isn’t empty
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

$$w = xyz$$

$$|xy| \leq n$$

$$y \neq \varepsilon$$

$$xy^iz \in L \text{ for all } i \in \mathbb{N}_0$$

$w$ can be broken into three pieces

where the first two pieces occur at the start of the string

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.

$xy^i z \in L$ for all $i \in \mathbb{N}_0$

Because $y$ can be *pumped* zero or more times.
A (sad) pump
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!
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

Any string of length three 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 three 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 $L = \{w \in \Sigma^* \mid w$ contains $00$ as a substring$\}$

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 $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

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

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

Any string of length three 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 three 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 $L = \{w \in \Sigma^* \mid w$ contains $00$ as a substring$\}$

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 $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

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

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

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

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 $L = \{w \in \Sigma^* | w \text{ contains } 00 \text{ as a substring}\}$

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

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

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

$11000001$
Let $\Sigma = \{0, 1\}$ and $L = \{w \in \Sigma^* \mid w \text{ contains } 00 \text{ as a substring}\}$

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

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

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

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

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

```
1 1 0 0 0 0 1 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 three or greater can be split into three pieces, the second of which can be “pumped”.

\[
\begin{array}{ccccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1
\end{array}
\]
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!
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 $EQUAL = \{w?w \mid w \in \{0, 1\}^*\}$

Is $EQUAL$ 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

\[ w = xyz \]

where the first two pieces occur at the start of the string

\[ |xy| \leq n \]

the middle part isn’t empty

\[ y \neq \varepsilon \]

and the middle piece can be replicated zero or more times

\[ xy^iz \in L \text{ for all } i \in \mathbb{N}_0 \]
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

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

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

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

\[ \text{EQUAL} = \{ w?w \mid w \in \{0, 1\}^* \} \]
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.

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 $i \in \mathbb{N}_0$
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.

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^iz &\in L \text{ for all } i \in \mathbb{N}_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.

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
      \( w = xyz \)
      \( |xy| \leq n \)
      \( y \neq \varepsilon \)
      \( xy^i z \in L \) for all \( i \in \mathbb{N}_0 \)
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

\[ w = xyz \]
\[ |xy| \leq n \]
\[ y \neq \epsilon \]
\[ xy^iz \in L \text{ for all } i \in \mathbb{N}_0 \]
THEOREM. *EQUAL* is not regular.

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

\[
\text{For every regular language } L, \\
\quad \text{there exists a positive integer } n \text{ such that} \\
\quad \text{for every string } w \in L \text{ with } |w| \geq n, \\
\quad \text{there exist strings } x, y, \text{ and } z \text{ such that} \\
\quad w = xyz \\
\quad |xy| \leq n \\
\quad y \neq \varepsilon \\
\quad xy^iz \in L \text{ for all } i \in \mathbb{N}_0
\]

*The hardest part of most Pumping Lemma proofs is choosing a string that we should be able to pump but cannot.*
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 $w = 0^n?0^n$.

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 $i \in \mathbb{N}_0$.
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 \( w = 0^n?0^n \).

\begin{center}
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

\( w = xyz \)

\(|xy| \leq n\)

\( y \neq \epsilon \)

\( xy^i z \in L \) for all \( i \in \mathbb{N}_0 \)
\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 $w = 0^n \cdot 0^n$. 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 $x, y, z$ such that

- $w = xyz$
- $|xy| \leq n$
- $y \neq \varepsilon$
- $xy^iz \in L$ for all $i \in \mathbb{N}_0$
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 $w \in \textit{EQUAL}$ and $|w| = 2n + 1 \geq n$.

\begin{center}
\begin{tabular}{l}
\textbf{For every} regular language $L$, \\
\quad \textbf{there exists} a positive integer $n$ such that \\
\quad \textbf{for every} string $w \in L$ with $|w| \geq n$, \\
\quad \textbf{there exist} strings $x$, $y$, and $z$ such that \\
\quad $w = xyz$ \\
\quad $|xy| \leq n$ \\
\quad $y \neq \varepsilon$ \\
\quad $xy^iz \in L$ for all $i \in \mathbb{N}_0$
\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 $w = 0^n1^n$. 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 $i \in \mathbb{N}_0$
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 \( w = 0^n?0^n \).

Then \( w \in \textsc{Equal} \) 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 \( i \in \mathbb{N}_0 \), \( xy^iz \in \textsc{Equal} \).

\[
\begin{align*}
\text{For every regular language } L, \\
\text{there exists a positive integer } n \text{ such that} \\
\text{for every string } w \in L \text{ with } |w| \geq n, \\
\text{there exist strings } x, y, \text{ and } z \text{ such that} \\
w = xyz \\
|xy| \leq n \\
y \neq \varepsilon \\
xyiz \in L \text{ for all } i \geq 0
\end{align*}
\]
**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 \( w = 0^n?0^n \). 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 \( i \in \mathbb{N}_0 \), \( xy^iz \in \text{EQUAL} \).
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 $w = 0^n?0^n$. Then $w \in \textit{EQUAL}$ and $|w| = 2n + 1 \geq n$. Thus by the Pumping Lemma, we can write $w = xyz$ such that $|xy| \leq n$ and $y \neq \epsilon$ and for any $i \in \mathbb{N}_0$, $xy^iz \in \textit{EQUAL}$.

\textbf{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.}

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 $i \in \mathbb{N}_0$
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 $w \in \textit{EQUAL}$ 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 $i \in \mathbb{N}_0$, $xy^iz \in \textit{EQUAL}$. The string $y$ must consist only of 0s 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 $i \in \mathbb{N}_0$
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 \( w = 0^n?0^n \).

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 \( i \in \mathbb{N}_0 \), \( xy^iz \in \text{EQUAL} \). 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 \( \text{EQUAL} \).

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 \( i \in \mathbb{N}_0 \)
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 $i \in \mathbb{N}_0$, $xy^iz \in EQUAL$. The string $y$ must consist only of 0s before the ? or it would violate that $|xy| \leq n$. Therefore, $xy^0z = 0^m?0^n$, where $m < n$, and is not in EQUAL. This contradicts the Pumping Lemma, so our assumption was wrong. Thus EQUAL is not regular. ■

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

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

$L$ 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 $xy^iz \notin L$. (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 $y \neq \varepsilon$ 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 $n$

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

**You**

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

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

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

PROOF. By contradiction; assume \( L \) is regular. Let \( n \) be the pumping length guaranteed by the Pumping Lemma.
THEOREM. $L = \{a^n b^n \mid n \in \mathbb{N}_0\}$ 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. $L = \{a^n b^n \mid n \in \mathbb{N}_0\}$ 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 $xy^i z \in L$. 
THEOREM. $L = \{a^n b^n \mid n \in \mathbb{N}_0\}$ 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.
THEOREM. $L = \{a^n b^n \mid n \in \mathbb{N}_0\}$ 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. $L = \{a^n b^n \mid n \in \mathbb{N}_0\}$ 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.

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. \( L = \{a^n b^n \mid n \in \mathbb{N}_0\} \) 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.

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

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 \( w \in BALANCE \) 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 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

\[ w = xyz \]

\[ |xy| \leq n \]

\[ y \neq \varepsilon \]

\[ xyz^i \in L \text{ for all } i \geq 0 \]
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: \( L = \{w \mid w \text{ has an equal number of } 1\text{s and } 0\text{s}\} \) is not regular

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

We need to show splitting this string into \( xyz \) where \( xy^iz \) is in \( L \) is impossible...

But it is possible!

If \( x = \varepsilon, y = 01, \) and \( z = (01)^{n-1}, \) \( xy^iz \) 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 $0$s and $1$s.
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

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 1s, so $x\text{yyz} \not\in L$.

Contradiction! We win!
Remember

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

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

This sometimes means considering several different cases.
Broken Pumping Lemma proofs
Broken Pumping Lemma proofs

The following slides show several attempts to use the Pumping Lemma to show that the language consisting of strings that are palindromes over \( \Sigma = \{a, b\}^* \) is not regular.

Each proof has a flaw that makes it inadequate or incorrect.
Broken proof 1

Suppose that the set of palindromes were regular. Let $n$ be the value from the Pumping Lemma. Consider the string $w = 00011000$. $w$ is clearly a palindrome. By the Pumping Lemma, there must exist strings $x$, $y$, and $z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x$, $y$, and $z$ such that $w = xyz$, $|xy| \leq n$, and $|y| \geq 1$. Because $|xy| \leq n$, $xy$ is entirely contained in the $0^n$ at the start of $w$. So, $x$ and $y$ consist entirely of zeros.

Now, consider $xy^0z$. By the Pumping Lemma, $xy^0z$ must be in the language. But $xy^0z$ can’t be a palindrome. This means that the set of palindromes doesn’t satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.
Suppose that the set of palindromes were regular. Let $n$ be the value from the Pumping Lemma. Consider the string $w = 00011000$. $w$ is clearly a palindrome. By the Pumping Lemma, there must exist strings $x$, $y$, and $z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x$, $y$, and $z$ that satisfy the constraints. Let $|y| > 1$. Because $|xy| \leq n$, $xy$ is entirely contained in the start of $w$. So, $x$ and $y$ consist entirely of zeros.

Now, consider $xy^0z$. By the Pumping Lemma, $xy^0z$ must be in the language. But $xy^0z$ can't be a palindrome. This means that the set of palindromes doesn't satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.

This proof has a big mistake. It's using a fixed string $w$, whose length doesn't depend on $n$. There's no way to flesh out the later bits of the proof if you have done this. The string $w$ has to get longer as $n$ gets bigger.
Broken proof 2

Suppose that the set of palindromes were regular. Let $n$ be the value from the Pumping Lemma. Consider a string $w$ in $L$, with $|w| \geq n$. By the Pumping Lemma, there must exist strings $x, y, \text{and } z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x, y, \text{and } z$ such that $w = xyz$, $|xy| \leq n$, and $|y| \geq 1$. Because $|xy| \leq n$, $y$ is contained in the first $n$ characters of $w$.

Now, consider $xy^0z$. By the Pumping Lemma, $xy^0z$ must be in the language. But $xy^0z$ can't be a palindrome. This means that the set of palindromes doesn't satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.
Suppose that the set of palindromes were regular. Let $n$ be the value from the Pumping Lemma. Consider a string $w$ in $L$, with $|w| \geq n$. By the Pumping Lemma, there must exist strings $x$, $y$, and $z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x$, $y$, and $z$ such that $w = xyz$, $|xy| \leq n$, and $|y| \geq 1$. Because $|xy| \leq n$, $y$ is contained in the first $n$ characters of $w$.

Now, consider $xy^0z$. By the Pumping Lemma, $xy^0z$ must be in the language. But $xy^0z$ can't be a palindrome. This means that the set of palindromes doesn't satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.

**Broken proof 2**

Wow – this proof never assigned a value to $w$! There's no way you can write a Pumping Lemma proof without a specific value for $w$. 
Broken proof 3

Suppose that the set of palindromes were regular. Let $n$ be the value from the Pumping Lemma. Consider the string $w = (01)^n(10)^n$. $w$ is clearly a palindrome and $|w| \geq n$. By the Pumping Lemma, there must exist strings $x, y, \text{ and } z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x, y, \text{ and } z$ such that $w = xyz$, $|xy| \leq n$, and $|y| \geq 1$. Because $|xy| \leq n$, $xy$ is entirely contained in the $(01)^n$ at the start of $w$. So $y$ must be $(01)^i$ for some integer $i \geq 1$.

Now, consider $xy^0z$. $xy^0z = (01)^{n-i}(10)^n$. By the Pumping Lemma, $xy^0z$ is supposed to be in the language. But the number of $01$ and $10$ pairs don’t match (since $i \geq 1$), so $xy^0z$ can’t be a palindrome.

This means that the set of palindromes doesn’t satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.
Suppose that the set of palindromes were regular. Let \( n \) be the value from the Pumping Lemma. Consider the string \( w = (01)^n (10)^n \). 

\( w \) is clearly a palindrome and \( |w| \geq n \). By the Pumping Lemma, there must exist strings \( x, y, \) and \( z \) satisfying the four constraints of the Pumping Lemma.

So, pick any \( x, y, \) and \( z \) such that \( w = xyz \), \( |xy| \leq n \), and \( |y| \geq 1 \). Because \( |xy| \leq n \), \( xy \) is entirely contained in the \((01)^n\) at the start of \( w \). So \( y \) must be \((01)^i\) for some integer \( i \geq 1 \).

Now, consider \( xy^0z \). \( xy^0z = (01)^{n-i}(10)^n \). By the Pumping Lemma, \( xy^0z \) is supposed to be in the language. But the number of \( 01 \) and \( 10 \) pairs don’t match (since \( i \geq 1 \)), so \( xy^0z \) can’t be a palindrome.

This means that the set of palindromes doesn’t satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.

**Your adversary gets to pick how \( w \) is divided into \( x, y, \) and \( z \), so \( y \) might not actually be a neat set of \( 01 \) pairs; it might start with a \( 1 \) or end with a \( 0 \). The middle bit of this proof would need to be patched up. It probably can be, since \( w \) is so much longer than \( n \), but it’s not going to be easy.**

**A better way to fix this proof would be to find a better choice for \( w! \)**
Broken proof 4

Suppose that the set of palindromes were regular. Let \( n \) be the value from the Pumping Lemma. Consider the string \( w = 0^n110^n \). \( w \) is clearly a palindrome and \( |w| \geq n \). By the Pumping Lemma, there must exist strings \( x, y, \) and \( z \) satisfying the four constraints of the Pumping Lemma.

So, pick any \( x, y, \) and \( z \) such that \( w = xyz, |xy| \leq n, \) and \( |y| \geq 1 \). Because \( |xy| \leq n \), \( xy \) is entirely contained in the \( 0^n \) at the start of \( w \). So \( x \) and \( y \) consist entirely of 0s.

Now, consider \( xy^0z \). By the Pumping Lemma, \( xy^0z \) must be in the language. But \( xy^0z \) can’t be a palindrome. This means that the language of palindromes doesn’t satisfy the Pumping Lemma and, thus, cannot be regular.
Suppose that the set of palindromes were regular. Let $n$ be the value from the Pumping Lemma. Consider the string $w = 0^n110^n$. $w$ is clearly a palindrome and $|w| \geq n$. By the Pumping Lemma, there must exist strings $x$, $y$, and $z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x$, $y$, and $z$ such that $w = xyz$, $|xy| \leq n$, and $|y| \geq 1$. Because $|xy| \leq n$, $xy$ is entirely contained in the $0^n$ at the start of $w$. So $x$ and $y$ consist entirely of $0$s.

Now, consider $xy^0z$. By the Pumping Lemma, $xy^0z$ must be in the language. But $xy^0z$ can’t be a palindrome. This means that the language of palindromes doesn’t satisfy the Pumping Lemma and, thus, cannot be regular.

This proof doesn’t contain enough detail about why $xy^0z$ isn’t a palindrome.
Suppose that the set of palindromes were regular. Let $n$ be the value from
the Pumping Lemma. Consider the string $w = 0^n110^n$. $w$ is clearly a
palindrome, and $|w| \geq n$. By the Pumping Lemma, there must exist strings $x, y, \text{ and } z$ satisfying the four constraints of the Pumping Lemma.

So, pick any $x, y, \text{ and } z$ such that $w = xyz$, $|xy| \leq n$, and $|y| \geq 1$. Because $|xy| \leq n$, $xy$ is entirely contained in the $0^n$ at the start of $w$. So $x$ and $y$ consist entirely of zeros, i.e., $x = 0^i$ and $y = 0^j$. Then $z$ must look like $0^k110^n$, where $i + j + k = n$.

Now, consider $xy^0z$. By the Pumping Lemma, $xy^0z$ must be in the language. But $xy^0z = 0^i0^k110^n$. This is just $0^{i+k}110^n$. Since $|y| \geq 1$, we know that $j \geq 1$. So $i + k < n$. This means that $xz$ is not a palindrome, because the numbers of zeros on the two ends don’t match.

This means that the set of palindromes doesn’t satisfy the Pumping Lemma and, thus, the set of palindromes cannot be regular.
The Pumping Lemma mascot, the Pumping Llama
by Kimberly Do
Proof strategy: 
*Closure properties*
Recall: Closure properties

Certain operations on regular languages are guaranteed to produce regular languages.

These *closure properties* can be used to prove a language is regular – or that it’s non-regular.
We’ve seen the regular languages are closed under the \textit{regular operations}, i.e., those used to construct regular expressions:

\begin{itemize}
  \item \textbf{Union:} \(L_1 \cup L_2\)
  \item \textbf{Concatenation:} \(L_1 L_2\)
  \item \textbf{Star-closure:} \(L_1^*\)
\end{itemize}

And that they’re also closed under these set operations:

\begin{itemize}
  \item \textbf{Complementation:} \(\overline{L_1}\)
  \item \textbf{Intersection:} \(L_1 \cap L_2\)
  \item \textbf{Difference:} \(L_1 - L_2\)
\end{itemize}
CLAIM: If $L$ is regular, then $L^R$ is also regular.

PROOF SKETCH: Suppose $L$ is a regular language. We can therefore construct an NFA with a single final state that accepts $L$. (Use $\varepsilon$-transitions!)

We can then make the start state of this NFA the final state, make the final state the start state, and reverse the direction of all arcs in the NFA.

The modified NFA accepts a string $w^R$ if and only if the original NFA accepts $w$.

Therefore the modified NFA accepts $L^R$. 
To show that a language $L$ is non-regular using closure properties, we do a *proof by contradiction*:

Assume $L$ is regular.

Combine $L$ with known regular languages using operations the regular languages are closed under.

If you produce a known non-regular language, then the assumption was wrong and $L$ is non-regular.
CLAIM. $L = \{ w \mid w \text{ in } \{a,b\}^* \mid n_a(w) = n_b(w) \}$ is non-regular.

PROOF SKETCH. Assume $L$ is regular. We know $a^*b^*$ is a regular language because we can write it as a regular expression. Because the regular languages are closed under intersection, $L \cap a^*b^* = \{ a^n b^n \mid n \geq 0 \}$ must be regular. However, $\{ a^n b^n \mid n \geq 0 \}$ is easily proved non-regular using the Pumping Lemma.

Therefore $L$ must be non-regular.
What the closure theorem for union does not say

Closure theorem for union says: If $L_1$ and $L_2$ are regular, then $L = L_1 \cup L_2$ is regular.

What happens if (for example) $L$ is regular? Does that mean that $L_1$ and $L_2$ are also?
What the closure theorem for union does not say

Closure theorem for union says: If $L_1$ and $L_2$ are regular, then $L = L_1 \cup L_2$ is regular.

What happens if (for example) $L$ is regular? Does that mean that $L_1$ and $L_2$ are also? 

Maybe.
Example

We know $a^+$ is regular.

Consider two cases for $L_1$ and $L_2$:

\[ a^+ = \{a^n \mid n > 0 \text{ and } n \text{ is prime}\} \cup \{a^n \mid n > 0 \text{ and } n \text{ is not prime}\} \]
\[ a^+ = L_1 \cup L_2 \]
Example

We know $a^+$ is regular.

Consider two cases for $L_1$ and $L_2$:

\[ a^+ = \{a^n \mid n > 0 \text{ and } n \text{ is prime}\} \cup \{a^n \mid n > 0 \text{ and } n \text{ is not prime}\} \]

\[ a^+ = L_1 \cup L_2 \]

*Neither $L_1$ nor $L_2$ is regular!* 

\[ a^+ = \{a^n \mid n > 0 \text{ and } n \text{ is even}\} \cup \{a^n \mid n > 0 \text{ and } n \text{ is odd}\} \]

\[ a^+ = L_1 \cup L_2 \]
Example

We know $a^+$ is regular.

Consider two cases for $L_1$ and $L_2$:

$$a^+ = \{a^n \mid n > 0 \text{ and } n \text{ is prime}\} \cup \{a^n \mid n > 0 \text{ and } n \text{ is not prime}\}$$

$$a^+ = L_1 \cup L_2$$

*Neither $L_1$ nor $L_2$ is regular!*

$$a^+ = \{a^n \mid n > 0 \text{ and } n \text{ is even}\} \cup \{a^n \mid n > 0 \text{ and } n \text{ is odd}\}$$

$$a^+ = L_1 \cup L_2$$

*Both $L_1$ and $L_2$ are regular!*
Where we are

We’ve ended up where we are by trying to answer the question “what problems can you solve with a computer?”

We defined a computer to be a DFA, which means that the problems we can solve are precisely the regular languages.

We’ve discovered several equivalent ways to think about regular languages (DFAs, NFAs, and regular expressions).

We now have a powerful intuition for these languages: DFAs are finite-memory computers, and regular languages correspond to the problems solvable with finite memory.
Next

What does computation look like with unbounded memory?

What problems can you solve with unbounded-memory computers?
Acknowledgments

This lecture incorporates material from:

Nancy Ide
Keith Schwarz
Michael Sipser