Chapter Eleven: Non-Regular Languages
We have now encountered regular languages in several different places. They are the languages that can be recognized by a DFA. They are the languages that can be recognized by an NFA. They are the languages that can be denoted by a regular expression. They are the languages that can be generated by a right-linear grammar. You might begin to wonder: are there any languages that are not regular?

In this chapter, we will see that there are. There is a proof tool that is often used to prove languages non-regular. It is called the pumping lemma, and it describes an important property that all regular languages have. If you can show that a given language does not have this property, you can conclude that it is not a regular language.
Outline

• 11.1 The Language \( \{a^n b^n\} \)
• 11.2 The Languages \( \{xx^R\} \)
• 11.3 Pumping
• 11.4 Pumping-Lemma Proofs
• 11.5 Strategies
• 11.6 Pumping And Finite Languages
The Language $\{a^n b^n\}$

- Any number of $a$s followed by the same number of $b$s
- Easy to give a grammar for this language:

  \[
  S \rightarrow aSb \mid \varepsilon
  \]

- All derivations of a fully terminal string use the first production $n=0$ or more times, then the last production once: $a^n b^n$

- Is it a regular language? For example, is there an NFA for it?
Trying To Build An NFA

• We'll try working up to it
• The subset \( \{ a^n b^n \mid n \leq 0 \} \):

• The subset \( \{ a^n b^n \mid n \leq 1 \} \):
The Subset \( \{ a^n b^n \mid n \leq 2 \} \)
The Subset $\{a^n b^n \mid n \leq 3\}$
A Futile Effort

• For each larger value of \( n \) we added two more states
• We're using the states to count the \( a \)s, then to check that the same number of \( b \)s follow
• That's not going to be a successful pattern on which to build an NFA for all of \( \{a^n b^n\} \)
  – NFA needs a fixed, finite number of states
  – No fixed, finite number will be enough to count the unbounded \( n \) in \( \{a^n b^n\} \)
• This is not a proof that no NFA can be constructed
• But it does contain the germ of an idea for a proof…

*Formal Language*, chapter 11, slide 8
Theorem 11.1

The language \( \{a^n b^n\} \) is not regular.

- Let \( M = (Q, \{a,b\}, \delta, q_0, F) \) be any DFA over the alphabet \( \{a,b\} \); we'll show that \( L(M) \neq \{a^n b^n\} \)
- Given as for input, \( M \) visits a sequence of states:
  - \( \delta^*(q_0,\varepsilon) \), then \( \delta^*(q_0,a) \), then \( \delta^*(q_0,aa) \), and so on
- Since \( Q \) is finite, \( M \) eventually revisits one:
  - \( \exists i \) and \( j \) with \( i < j \) such that \( \delta^*(q_0,a^i) = \delta^*(q_0,a^j) \)
- Append \( b^j \), and we see that \( \delta^*(q_0,a^i b^j) = \delta^*(q_0,a^i b^j) \)
- So \( M \) either accepts both \( a^i b^j \) and \( a^j b^i \), or rejects both
- \( \{a^n b^n\} \) contains \( a^i b^j \) but not \( a^j b^i \), so \( L(M) \neq \{a^n b^n\} \)
- So no DFA has \( L(M) = \{a^n b^n\} \): \( \{a^n b^n\} \) is not regular

*Formal Language, chapter 11, slide 9*
A Word About That Proof

• Nothing was assumed about the DFA $M$, except its alphabet $\{a,b\}$
• In spite of that, we were able to infer quite a lot about its behavior
• The basic insight: with a sufficiently long string we can force any DFA to repeat a state
• That's the basis of a wide variety of non-regularity proofs
Outline

• 11.1 The Language \( \{a^n b^n\} \)
• 11.2 The Languages \( \{xx^R\} \)
• 11.3 Pumping
• 11.4 Pumping-Lemma Proofs
• 11.5 Strategies
• 11.6 Pumping And Finite Languages
The Languages \(\{xx^R\}\)

- The notation \(x^R\) means the string \(x\), reversed
- \(\{xx^R\}\) is the set of strings that can be formed by taking any string in \(\Sigma^*\), and appending the same string, reversed
- For \(\Sigma = \{a,b\}\), \(\{xx^R\}\) includes the strings \(\epsilon, aa, bb, abba, baab, aaaa, bbbb\), and so on
- Another way of saying it: \(\{xx^R\}\) is the set of even-length palindromes
A Grammar For \( \{xx^R \mid x \in \{a,b\}^*\} \)

\[ S \rightarrow aSa \mid bSb \mid \varepsilon \]

- A derivation for \( abba \):
  - \( S \Rightarrow aSa \Rightarrow abSba \Rightarrow abba \)
- A derivation for \( abaaba \):
  - \( S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow abaaba \)
- Every time you use one of the first two productions, you add a symbol to the end of the first half, and the same symbol to the start of the second half
- So the second half is always the reverse of the first half: \( L(G) = \{xx^R \mid x \in \{a,b\}^*\} \)
- But is this language regular?
Intuition

• After seeing the first example, you may already have the feeling this can't be regular
  – A finite state machine would have to use states to keep track of $x$, then check that it is followed by a matching $x^R$
  – But there is no bound on the length of $x$, so no fixed, finite number of states will suffice

• The formal proof is very similar to the one we used for $\{a^n b^n\}$...
Theorem 11.2

The language \( \{xx^R\} \) is not regular for any alphabet with at least two symbols.

- Let \( M = (Q, \Sigma, \delta, q_0, F) \) be any DFA with \(|\Sigma| \geq 2\); we'll show that \( L(M) \neq \{xx^R\} \)
- \( \Sigma \) has at least two symbols; call two of these \( a \) and \( b \)
- Given as for input, \( M \) visits a sequence of states:
  - \( \delta^*(q_0,\varepsilon) \), then \( \delta^*(q_0,a) \), then \( \delta^*(q_0,aa) \), and so on
- Since \( Q \) is finite, \( M \) eventually revisits one:
  - \( \exists i \) and \( j \) with \( i < j \) such that \( \delta^*(q_0,a^i) = \delta^*(q_0,a^j) \)
- Append \( bbai^j \), and we see that \( \delta^*(q_0,a^i bbai^j) = \delta^*(q_0,a^j bbai^j) \)
- So \( M \) either accepts both \( bbai^i \) and \( bbai^j \), or rejects both
- \( \{xx^R\} \) contains \( bbai^i \) but not \( bbai^j \), so \( L(M) \neq \{xx^R\} \)
- So no DFA has \( L(M) = \{xx^R\} \): \( \{xx^R\} \) is not regular

*Formal Language, chapter 11, slide 15*
Outline

• 11.1 The Language \( \{a^n b^n\} \)
• 11.2 The Languages \( \{xx^R\} \)
• 11.3 Pumping
• 11.4 Pumping-Lemma Proofs
• 11.5 Strategies
• 11.6 Pumping And Finite Languages
Review

• We've shown two languages non-regular: \( \{a^n b^n\} \) and \( \{xx^R\} \)

• In both cases, the key idea was to choose a string long enough to make any given DFA repeat a state

• For both those proofs we just used strings of \( a \)s, and showed that \( \exists \ i \) and \( j \) with \( i < j \) such that \( \delta^*(q_0,a^i) = \delta^*(q_0,a^j) \)
Multiple Repetitions

• When you've found a state that repeats once, you can make it repeat again and again
• For example, our $\delta^*(q_0, a^i) = \delta^*(q_0, a^j)$:
  – Let $r$ be the state in question: $r = \delta^*(q_0, a^i)$
  – After $j-i$ more as it repeats: $r = \delta^*(q_0, a^{i+(j-i)})$
  – That little substring $a^{(j-i)}$ takes it from state $r$ back to state $r$

\[
\begin{align*}
  r &= \delta^*(q_0, a^i) \\
  &= \delta^*(q_0, a^{i+(j-i)}) \\
  &= \delta^*(q_0, a^{i+2(j-i)}) \\
  &= \delta^*(q_0, a^{i+3(j-i)})
\end{align*}
\]
Pumping

• We say that the substring $a^{(j-i)}$ can be *pumped* any number of times, and the DFA always ends up in the same state

• All regular languages have an important property involving pumping

• Any sufficiently long string in a regular language must contain a pumpable substring

• Formally, the pumping lemma…
Lemma 11.3: The Pumping Lemma for Regular Languages

For all regular languages $L$ there exists some integer $k$ such that for all $xyz \in L$ with $|y| \geq k$, there exist $uvw = y$ with $|v| > 0$, such that for all $i \geq 0$, $xuv^i wz \in L$.

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be any DFA with $L(M) = L$
- Choose $k = |Q|$
- Consider any $x$, $y$, and $z$ with $xyz \in L$ and $|y| \geq k$
- Let $r$ be a state that repeats during the $y$ part of $xyz$
  - We know such a state exists because we have $|y| \geq |Q|$...

In state $r$ here  
And again here
Lemma 11.3: The Pumping Lemma for Regular Languages

For all regular languages $L$ there exists some integer $k$ such that for all $xyz \in L$ with $|y| \geq k$, there exist $uvw = y$ with $|v| > 0$, such that for all $i \geq 0$, $xuv^iwz \in L$.

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be any DFA with $L(M) = L$
- Choose $k = |Q|
- Consider any $x, y, z$ with $xyz \in L$ and $|y| \geq k$
- Let $r$ be a state that repeats during the $y$ part of $xyz$
- Choose $uvw = y$ so that $\delta^*(q_0, xu) = \delta^*(q_0, xuv) = r$
- Now $v$ is pumpable: for all $i \geq 0$, $\delta^*(q_0, xuv^i) = r$...

In state $r$ here

And again here
Lemma 11.3: The Pumping Lemma for Regular Languages

For all regular languages \( L \) there exists some integer \( k \) such that for all \( xyz \in L \) with \( |y| \geq k \), there exist \( uvw = y \) with \( |v| > 0 \), such that for all \( i \geq 0 \), \( xuv^i wz \in L \).

- Let \( M = (Q, \Sigma, \delta, q_0, F) \) be any DFA with \( L(M) = L \)
- Choose \( k = |Q| \)
- Consider any \( x, y, \) and \( z \) with \( xyz \in L \) and \( |y| \geq k \)
- Let \( r \) be a state that repeats during the \( y \) part of \( xyz \)
- Choose \( uvw = y \) so that \( \delta^*(q_0, xu) = \delta^*(q_0, xuv) = r \)
- Now \( v \) is pumpable: for all \( i \geq 0 \), \( \delta^*(q_0, xuv^i) = r \)
- Then for all \( i \geq 0 \), \( \delta^*(q_0, xuv^i wz) = \delta^*(q_0, xuvwz) = \delta^*(q_0, xyz) \in F \)
- Therefore, for all \( i \geq 0 \), \( xuv^i wz \in L \)
Pumping Lemma Structure

For all regular languages \( L \) there exists some integer \( k \) such that for all \( xyz \in L \) with \( |y| \geq k \), there exist \( uvw = y \) with \( |v| > 0 \), such that for all \( i \geq 0 \), \( xuv^i wz \in L \).

- Notice the alternating "for all" and "there exist" clauses:
  1. \( \forall L \) ...
  2. \( \exists k \) ...
  3. \( \forall xyz \) ...
  4. \( \exists uvw \) ...
  5. \( \forall i \) ...
- Our proof showed how to construct the \( \exists \) parts
- But that isn't part of the lemma: it's a black box
- The lemma says only that \( k \) and \( uvw \) exist
Outline

• 11.1 The Language \(a^n b^n\)
• 11.2 The Languages \(xx^R\)
• 11.3 Pumping
• 11.4 Pumping-Lemma Proofs
• 11.5 Strategies
• 11.6 Pumping And Finite Languages
Pumping-Lemma Proofs

• The pumping lemma is very useful for proving that languages are not regular
• For example, \{a^nb^n\}…
\{a^n b^n\} Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that \( L = \{a^n b^n\} \) is regular, so the pumping lemma holds for \( L \). Let \( k \) be as given by the pumping lemma.

2. Choose \( x, y, \) and \( z \) as follows:
   \[ x = a^k \]
   \[ y = b^k \]
   \[ z = \epsilon \]
   Now \( xyz = a^k b^k \in L \) and \( |y| \geq k \) as required.

3. Let \( u, v, \) and \( w \) be as given by the pumping lemma, so that \( uvw = y, |v| > 0 \), and for all \( i \geq 0, xuv^iwz \in L \).

4. Choose \( i = 2 \). Since \( v \) contains at least one \( b \) and nothing but \( bs \), \( uv^2w \) has more \( bs \) than \( uvw \). So \( xuv^2wz \) has more \( bs \) than \( as \), and so \( xuv^2wz \notin L \).

5. By contradiction, \( L = \{a^n b^n\} \) is not regular.
The Game

- The alternating ∀ and ∃ clauses of the pumping lemma make these proofs a kind of game.
- The ∃ parts \((k\text{ and }uvw)\) are the pumping lemma's moves: these values exist, but are not ours to choose.
- The ∀ parts \((L, xyz, \text{ and } i)\) are our moves: the lemma holds for all proper values, so we have free choice.
- We make our moves strategically, to force a contradiction.
- No matter what the pumping lemma does with its moves, we want to end up with some \(xuv^iwz \notin L\).
The Pattern

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $L = \{a^n b^n\}$ is regular, so the pumping lemma holds for $L$. Let $k$ be as given by the pumping lemma.

2. Here, you chose $xyz$ and show that they meet the requirements, $xyz \in L$ and $|y| \geq k$. Choose them so that pumping in the $y$ part will lead to a contradiction, a string $\not\in L$.

3. Let $u$, $v$, and $w$ be as given by the pumping lemma, so that $uvw = y$, $|v| > 0$, and for all $i \geq 0$, $xuv^i wz \in L$.

4. Here, you choose $i$, the number of times to pump, and show that you have a contradiction: $xuv^i wz \not\in L$.

5. By contradiction, $L = \{a^n b^n\}$ is not regular.
\{xx^R\} Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $L = \{xx^R\}$ is regular, so the pumping lemma holds for $L$. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   \[
x = a^kbb
   
y = a^k
   \]
   \[
z = \varepsilon
   \]
   Now $xyz = a^kbb_a^k \in L$ and $|y| \geq k$ as required.

3. Let $u$, $v$, and $w$ be as given by the pumping lemma, so that $uvw = y$, $|v| > 0$, and for all $i \geq 0$, $xuv^iwz \in L$.

4. Choose $i = 2$. Since $v$ contains at least one $a$ and nothing but $a$s, $uvw^2w$ has more $a$s than $uvw$. So $xuv^2wz$ has more $a$s after the $bs$ than before them, and thus $xuv^2wz \notin L$.

5. By contradiction, $L = \{xx^R\}$ is not regular.
Outline

• 11.1 The Language \( \{a^n b^n\} \)
• 11.2 The Languages \( \{xx^R\} \)
• 11.3 Pumping
• 11.4 Pumping-Lemma Proofs
• 11.5 Strategies
• 11.6 Pumping And Finite Languages
Proof Strategy

• It all comes down to those four delicate choices: xyz and $i$
• Usually, there are a number of choices that successfully lead to a contradiction
• And, of course many others that fail
• For example: let $A = \{a^n b^i a^n \mid n \geq 0, j \geq 1\}$
• We'll try a pumping-lemma proof that $A$ is not regular
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n | n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x, y, \text{ and } z$ as follows:
   
   \[ x = a a a \]
   
   \[ y = b \]
   
   \[ z = a a a \]

?
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:

   $x = aaa$
   $y = b$
   $z = aaa$

   **Bad choice.** The pumping lemma requires $|y| \geq k$. It never applies to fixed-size examples. Since $k$ is not known in advance, $y$ must be some string that is constructed using $k$, such as $a^k$. 

*Formal Language*, chapter 11, slide 33
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   
   $x = \varepsilon$
   
   $y = a^k$
   
   $z = a^k$

   ?
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x, y,$ and $z$ as follows:
   - $x = \epsilon$
   - $y = a^k$
   - $z = a^k$

   Bad choice. The pumping lemma lemma only applies if the string $xyz \in A$. That is not the case here.
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   
   $x = a^n$
   $y = b$
   $z = a^n$

?
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that \( A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\} \) is regular. Let \( k \) be as given by the pumping lemma.

2. Choose \( x, y, \) and \( z \) as follows:
   \[ x = a^n \]
   \[ y = b \]
   \[ z = a^n \]
   This is ill-formed, since the value of \( n \) is not defined. At this point the only integer variable that is defined is \( k \).
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   \[
   \begin{align*}
   x &= a^k \\
   y &= b^{k+2} \\
   z &= a^k
   \end{align*}
   \]
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   
   \begin{align*}
   x &= a^k \\
   y &= b^{k+2} \\
   z &= a^k
   \end{align*}

   This meets the requirements $xyz \in A$ and $|y| \geq k$, but it is a bad choice because it won't lead to a contradiction. Pumping within the string $y$ will change the number of $b$s in the middle, but the resulting string can still be in $A$. 

---

Formal Language, chapter 11, slide 39
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   \[
   x = a^k \\
   y = bba^k \\
   z = \varepsilon
   \]

?
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that \( A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\} \) is regular. Let \( k \) be as given by the pumping lemma.

2. Choose \( x, y, \) and \( z \) as follows:
   \[
   \begin{align*}
   x &= a^k \\
   y &= bba^k \\
   z &= \varepsilon
   \end{align*}
   \]

   This meets the requirements \( xyz \in A \) and \( |y| \geq k \), but it is a bad choice because it won't lead to a contradiction. The pumping lemma can choose any \( uvw = y \) with \( |v| > 0 \). If it chooses \( u=b, v=b, \) and \( w = a^k \), there will be no contradiction, since for all \( i \geq 0 \), \( xuv^i wz \in A \).
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^nb^ja^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x, y, \text{ and } z$ as follows:
   
   $x = a^kb$
   
   $y = a^k$
   
   $z = \epsilon$

?
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   \begin{align*}
   x &= a^k b \\
   y &= a^k \\
   z &= \varepsilon
   \end{align*}

   Good choice. It meets the requirements $xyz \in A$ and $|y| \geq k$, and it will lead to a contradiction because pumping anywhere in the $y$ part will change the number of $a$s after the $b$, without changing the number before the $b$. 

*Formal Language*, chapter 11, slide 43
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   
   $$x = \varepsilon$$
   $$y = a^k$$
   $$z = b a^k$$

   ?
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that \( A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\} \) is regular. Let \( k \) be as given by the pumping lemma.

2. Choose \( x, y, \) and \( z \) as follows:
   \[
   x = \varepsilon \\
   y = a^k \\
   z = b a^k
   \]

An equally good choice.
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that \( A = \{a^n b^j a^n \mid n \geq 0, j \geq 1 \} \) is regular. Let \( k \) be as given by the pumping lemma.

2. Choose \( x, y, \) and \( z \) as follows:
   \[
   x = \epsilon \\
y = a^k \\
z = b a^k
   \]
   Now \( xyz = a^k b a^k \in A \) and \( |y| \geq k \) as required.

3. Let \( u, v, \) and \( w \) be as given by the pumping lemma, so that \( uvw = y, |v| > 0, \) and for all \( i \geq 0, xuv^i wz \in A. \)

1. Choose \( i = 1 \)
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^n b^j a^n \mid n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x, y,$ and $z$ as follows:
   
   $x = \varepsilon$

   $y = a^k$

   $z = ba^k$

   Now $xyz = a^k ba^k \in A$ and $|y| \geq k$ as required.

3. Let $u, v,$ and $w$ be as given by the pumping lemma, so that $uvw = y, |v| > 0,$ and for all $i \geq 0,$ $xuv^i w z \in A.$

1. Choose $i = 1$

   Bad choice -- the only bad choice for $i$ in this case! When $i = 1,$ $xuv^1 w z \in A,$ so there is no contradiction.
A Is Not Regular

1. Proof is by contradiction using the pumping lemma for regular languages. Assume that $A = \{a^nb^j|n \geq 0, j \geq 1\}$ is regular. Let $k$ be as given by the pumping lemma.

2. Choose $x$, $y$, and $z$ as follows:
   
   \begin{align*}
   x &= \varepsilon \\
   y &= a^k \\
   z &= ba^k
   \end{align*}

   Now $xyz = a^kba^k \in A$ and $|y| \geq k$ as required.

3. Let $u$, $v$, and $w$ be as given by the pumping lemma, so that $uvw = y$, $|v| > 0$, and for all $i \geq 0$, $xuv^iwz \in A$.

4. Choose $i = 2$. Since $v$ contains at least one $a$ and nothing but $a$s, $uv^2w$ has more as than $uvw$. So $xuv^2wz$ has more as before the $b$ than after it, and thus $xuv^2wz \notin A$.

5. By contradiction, $A$ is not regular.
Outline

• 11.1 The Language $\{a^n b^n\}$
• 11.2 The Languages $\{xx^R\}$
• 11.3 Pumping
• 11.4 Pumping-Lemma Proofs
• 11.5 Strategies
• 11.6 Pumping And Finite Languages
What About Finite Languages?

For all regular languages $L$ there exists some integer $k$ such that for all $xyz \in L$ with $|y| \geq k$, there exist $uvw = y$ with $|v| > 0$, such that for all $i \geq 0$, $xuv^iwz \in L$.

- The pumping lemma applies in a trivial way to any finite language $L$
- Choose $k$ greater than the length of the longest string in $L$
- Then it is clearly true that "for all $xyz \in L$ with $|y| \geq k$, …" since there are no strings in $L$ with $|y| \geq k$
- It is vacuously true
- In fact, all finite languages are regular…
Theorem 11.6

All finite languages are regular.

• Let $A$ be any finite language of $n$ strings: $A = \{x_1, \ldots, x_n\}$
• There is a regular expression that denotes this language: $A = L(x_1 + \ldots + x_n)$
• Or, in case $n = 0$, $A = L(\emptyset)$
• Since $A$ is denoted by a regular expression, $A$ is a regular language