More Properties of Regular Languages
We have proven

Regular languages are closed under:

Union
Concatenation
Star operation
Reverse
Namely, for regular languages $L_1$ and $L_2$:

- **Union**: $L_1 \cup L_2$
- **Concatenation**: $L_1L_2$
- **Star operation**: $L_1^*$
- **Reverse**: $L_1^R$

Regular Languages
We will prove

Regular languages are closed under:

Complement

Intersection
Namely, for regular languages $L_1$ and $L_2$:

\begin{align*}
\text{Complement} & \quad \overline{L_1} \\
\text{Intersection} & \quad L_1 \cap L_2
\end{align*}

\text{Regular Languages}
Complement

Theorem: For regular language $L$, the complement $\overline{L}$ is regular.

Proof: Take DFA that accepts $L$ and make
- nonfinal states final
- final states nonfinal
Resulting DFA accepts $\overline{L}$.
**Example:**

$L = L(a \ast b)$

```
q_0 ----> b ----> q_1 ----> a,b ----> q_2
```

$\overline{L} = L(a \ast + a \ast b(a + b)(a + b)^*)$

```
q_0 ----> b ----> q_1 ----> a,b ----> q_2
```
Intersection

Theorem: For regular languages \( L_1 \) and \( L_2 \) and the intersection \( L_1 \cap L_2 \) is regular.

Proof: Apply DeMorgan's Law:

\[
L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}
\]
\[ L_1, \ L_2 \quad \text{regular} \]

\[ \overline{L_1}, \ \overline{L_2} \quad \text{regular} \]

\[ \overline{L_1 \cup L_2} \quad \text{regular} \]

\[ \overline{L_1} \cup \overline{L_2} \quad \text{regular} \]

\[ \overline{L_1 \cup L_2} \quad \text{regular} \]

\[ L_1 \cap L_2 \quad \text{regular} \]
Standard Representations of Regular Languages
Standard Representations of Regular Languages

Regular Languages

DFAs

NFAs

Regular Grammars

Regular Expressions
When we say: We are given a Regular Language $L$

We mean: Language $L$ is in a standard representation
Elementary Questions about Regular Languages
Membership Question

**Question:** Given regular language $L$ and string $w$, how can we check if $w \in L$?

**Answer:** Take the DFA that accepts $L$ and check if $w$ is accepted.
DFA

\[ w \in L \]

DFA

\[ w \notin L \]
Question: Given regular language \( L \) how can we check if \( L \) is empty: \( (L = \emptyset) \)?

Answer: Take the DFA that accepts \( L \) Check if there is a path from the initial state to a final state
\[ L \neq \emptyset \]

\[ L = \emptyset \]
Question: Given regular language $L$, how can we check if $L$ is finite?

Answer: Take the DFA that accepts $L$. Check if there is a walk with cycle from the initial state to a final state.
DFA

$L$ is infinite

DFA

$L$ is finite
Question: Given regular languages $L_1$ and $L_2$ how can we check if $L_1 = L_2$?

Answer: Find if $(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset$
\[(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset\]

\[L_1 \cap \overline{L_2} = \emptyset \quad \text{and} \quad \overline{L_1} \cap L_2 = \emptyset\]

\[L_1 \subseteq L_2 \quad \text{and} \quad L_2 \subseteq L_1\]

\[L_1 = L_2\]
\[(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) \neq \emptyset\]

- \(L_1 \cap \overline{L_2} \neq \emptyset\)
- \(L_1 \not\subset L_2\)

or

- \(\overline{L_1} \cap L_2 \neq \emptyset\)
- \(L_2 \not\subset L_1\)

\(L_1 \neq L_2\)
Non-regular languages
Non-regular languages

\{a^n b^n : n \geq 0\}
\{w w^R : w \in \{a, b\}^*\}

Regular languages

\[ a^* b \]
\[ b^* c + a \]
\[ b + c (a + b)^* \]

 etc...
How can we prove that a language $L$ is not regular?

Prove that there is no DFA that accepts $L$.

**Problem:** this is not easy to prove.

**Solution:** the Pumping Lemma !!!
The Pigeonhole Principle
4 pigeons

3 pigeonholes
A pigeonhole must contain at least two pigeons
$n$ pigeons

$\cdots$

$m$ pigeonholes

$n > m$
The Pigeonhole Principle

$n$ pigeons

$m$ pigeonholes

$n > m$

There is a pigeonhole with at least 2 pigeons
The Pigeonhole Principle

and

DFAs
DFA with 4 states

$q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{b} q_4 \xrightarrow{b}$
In walks of strings:  

- $a$  
- $aa$  
- $aab$

There is no state that is repeated.
In walks of strings: $aabb$, a state is repeated $bbaa$, $abbb$, $abbbabbbabbb...$
If the walk of string $w$ has length $|w| \geq 4$

then a state is repeated
Pigeonhole principle for any DFA:

If in a walk of a string $w$ transitions $\geq$ states of DFA then a state is repeated.
In general:

A string $w$ has length $\geq$ number of states

A state $q$ must be repeated in the walk $w$
The Pumping Lemma
Take an infinite regular language $L$

DFA that accepts $L$

$m$ states
Take string \( w \) with \( w \in L \)

There is a walk with label \( w \):

walk \( W \)
If string $w$ has length $|w| \geq m$ number of states
then, from the pigeonhole principle:

a state $q$ is repeated in the walk $w$

walk $w$
Write \[ w = x \ y \ z \]
Observations:  

$\text{length } |x y| \leq m$ \hspace{1cm} \text{number of states} 

$\text{length } |y| \geq 1$
Observation: The string $x z$ is accepted
Observation: The string $x y y z$ is accepted
Observation: The string $x y y y y z$ is accepted
In General: The string $x\, y^i\, z$ is accepted for $i = 0, 1, 2, \ldots$
In other words, we described:

The Pumping Lemma !!!
The Pumping Lemma:

• Given a infinite regular language $L$

• there exists an integer $m$

• for any string $w \in L$ with length $|w| \geq m$

• we can write $w = x y z$

• with $|x y| \leq m$ and $|y| \geq 1$

• such that: $x y^i z \in L$ for $i = 0, 1, 2, ...$
Applications
of
the Pumping Lemma
Theorem: The language \( L = \{ a^n b^n : n \geq 0 \} \) is not regular.

Proof: Use the Pumping Lemma.
\[ L = \{a^n b^n : n \geq 0\} \]

Assume for contradiction that \( L \) is a regular language.

Since \( L \) is infinite, we can apply the Pumping Lemma.
\[ L = \{a^n b^n : n \geq 0\} \]

Let \( m \) be the integer in the **Pumping Lemma**

Pick a string \( w \) such that: \( w \in L \) \( \quad \text{length } |w| \geq m \)

Example: \( \text{pick } w = a^m b^m \)
Write: \( a^m b^m = x y z \)

From the Pumping Lemma
it must be that: length \( |x y| \leq m, \quad |y| \geq 1 \)

Therefore: \( a^m b^m = a \ldots a a \ldots a a \ldots a b \ldots b \)

\( y = a^k, \quad k \geq 1 \)
We have: \[ x \ y \ z = a^m b^m \quad y = a^k, \quad k \geq 1 \]

From the Pumping Lemma: \[ x \ y^i z \in L \]
\[ i = 0, 1, 2, \ldots \]

Thus: \[ x \ y^2 z \in L \]
\[ x \ y^2 z = x \ y \ y \ z = a^{m+k} b^m \in L \]
Therefore: \[ a^{m+k} b^m \in L \]

\[ L = \{a^n b^n : n \geq 0\} \]

\[ a^{m+k} b^m \notin L \]

**CONTRADICTION!!**
Therefore: Our assumption that $L$ is a regular language is not true

**Conclusion:** $L$ is not a regular language
Non-regular languages \( \{a^n b^n : n \geq 0\} \)

Regular languages

\[ a^* b \]
\[ b^* c + a \]
\[ b + c(a + b)^* \]
\[ etc... \]
More Applications of the Pumping Lemma
The Pumping Lemma:

- Given a infinite regular language $L$
- there exists an integer $m$
- for any string $w \in L$ with length $|w| \geq m$
- we can write $w = x y z$
- with $|x y| \leq m$ and $|y| \geq 1$
- such that: $x y^i z \in L$ for $i = 0, 1, 2, ...$
Non-regular languages \[ L = \{ww^R : w \in \Sigma^* \} \]

Regular languages
Theorem: The language

\[ L = \{ww^R : w \in \Sigma^*\} \quad \Sigma = \{a, b\} \]

is not regular

Proof: Use the Pumping Lemma
\[ L = \{ w w^R : w \in \Sigma^* \} \]

Assume for contradiction that \( L \) is a regular language.

Since \( L \) is infinite, we can apply the Pumping Lemma.
$$L = \{ww^R : w \in \Sigma^*\}$$

Let $m$ be the integer in the Pumping Lemma

Pick a string $w$ such that: $w \in L$ and $|w| \geq m$

pick $w = a^m b^m b^m a^m$
Write \( a^m b^m b^m a^m = x \ y \ z \)

From the **Pumping Lemma** it must be that length \(|x\ y| \leq m, \ |y| \geq 1\)

\[
\begin{align*}
a^m b^m b^m a^m &= \underbrace{\underbrace{a \ldots a}_m a \ldots ab \ldots bb \ldots ba \ldots a}_{m} \\
x &\quad \underbrace{y}_{m} \quad \underbrace{z}_{m}
\end{align*}
\]

\(y = a^k, \ k \geq 1\)
We have: \[ x \ y \ z = a^m b^m b^m a^m \]

\[ y = a^k, \quad k \geq 1 \]

From the Pumping Lemma: \[ x \ y^i z \in L \]

\[ i = 0, 1, 2, \ldots \]

Thus: \[ x \ y^2 z \in L \]

\[ x \ y^2 z = x \ y \ y \ z = a^{m+k} b^m b^m a^m \in L \]
Therefore: \[ a^{m+k} b^m b^m a^m \in L \]

**BUT:** \[ L = \{ww^R : w \in \Sigma^*\} \]

\[ a^{m+k} b^m b^m a^m \notin L \]

**CONTRADICTION!!!**
Therefore: Our assumption that $L$
is a regular language is not true

**Conclusion:** $L$ is not a regular language
Non-regular languages

\[ L = \{a^n b^l c^{n+l} : n, l \geq 0\} \]

Regular languages
Theorem: The language 

\[ L = \{a^n b^l c^{n+l} : n, l \geq 0\} \]

is not regular

Proof: Use the Pumping Lemma
\[ L = \{a^n b^l c^{n+l} : n, l \geq 0\} \]

Assume for contradiction that \( L \) is a regular language.

Since \( L \) is infinite, we can apply the Pumping Lemma.
\[ L = \{ a^n b^l c^{n+l} : n, l \geq 0 \} \]

Let \( m \) be the integer in the Pumping Lemma.

Pick a string \( w \) such that: \( w \in L \) and \( \text{length } |w| \geq m \).

pick \( w = a^m b^m c^{2m} \)
Write \( a^m b^m c^{2m} = x y z \)

From the **Pumping Lemma** it must be that length \( |x y| \leq m, \quad |y| \geq 1 \)

\[
a^m b^m c^{2m} = a_{x} a_{y} a_{y} a_{b} b_{c} b_{c} c_{c} c_{c} \ldots c_{z}
\]

\( y = a^k, \quad k \geq 1 \)
We have: \[ x \quad y \quad z = a^m b^m c^{2m} \]

\[ y = a^k, \quad k \geq 1 \]

From the Pumping Lemma: \[ x \; y^i \; z \in L \]

\[ i = 0, 1, 2, ... \]

Thus: \[ x \; y^0 \; z \in L \]

\[ x \; y^0 \; z = x \; z = a^{m-k} b^m c^{2m} \in L \]
Therefore: \( a^{m-k} b^m c^{2m} \in L \)

\textbf{BUT:} \[ L = \{a^n b^l c^{n+l} : n, l \geq 0\} \]

\[ a^{m-k} b^m c^{2m} \not\in L \]

\textbf{CONTRACTION!!!}
Therefore: Our assumption that $L$ is a regular language is not true.

**Conclusion:** $L$ is not a regular language.
Non-regular languages

\[ L = \{a^n!: n \geq 0\} \]

Regular languages
Theorem: The language $L = \{a^{n!} : n \geq 0\}$ is not regular.

$n! = 1 \cdot 2 \cdot \cdots (n-1) \cdot n$

Proof: Use the Pumping Lemma.
\[ L = \{ a^n! : n \geq 0 \} \]

Assume for contradiction that \( L \) is a regular language.

Since \( L \) is infinite, we can apply the Pumping Lemma.
\[ L = \{a^n! : n \geq 0\} \]

Let \( m \) be the integer in the Pumping Lemma

Pick a string \( w \) such that: \( w \in L \), length \( |w| \geq m \)

pick \( w = a^m! \)
Write \( a^{m!} = x \ y \ z \)

From the **Pumping Lemma**

it must be that length \( |x \ y| \leq m, \ |y| \geq 1 \)

\[
a^{m!} = \underbrace{a \ldots a}_{m} \underbrace{a \ldots a}_{m!-m} \underbrace{a \ldots a}_{x} \underbrace{a \ldots a}_{y} \underbrace{a \ldots a}_{z}
\]

\( y = a^k, \ 1 \leq k \leq m \)
We have: \[ x \ y \ z = a^m! \]

\[ y = a^k, \quad 1 \leq k \leq m \]

From the Pumping Lemma:

\[ x \ y^i \ z \in L \]

\[ i = 0, 1, 2, \ldots \]

Thus:

\[ x \ y^2 \ z \in L \]

\[ x \ y^2 \ z = x \ y \ y \ z = a^{m! + k} \in L \]
Therefore: \[ a^{m!+k} \in L \quad 1 \leq k \leq m \]

And since: \[ L = \{a^n!: \ n \geq 0\} \]

There is \( p \): \[ m!+k = p! \quad 1 \leq k \leq m \]
However: \[ m! + k \leq m! + m \quad \text{for} \quad m > 1 \]

\[ \leq m! + m! \]

\[ < m!m + m! \]

\[ = m!(m + 1) \]

\[ = (m + 1)! \]

\[ m! + k < (m + 1)! \]

\[ m! + k \neq p! \quad \text{for any} \quad p \]
Therefore: \[ a^{m!+k} \in L \]

**BUT:** \[ L = \{ a^n : n \geq 0 \} \] and \[ 1 \leq k \leq m \]

\[ a^{m!+k} \notin L \]

**CONTRADICTION!!!**
Therefore: Our assumption that $L$ is a regular language is not true

**Conclusion:** $L$ is not a regular language