The language of a DFA

\[ M = \left( Q, \Sigma, q_0, F, \delta \right) \]

- We write \( q' \xrightarrow{x} q'' \) to denote that if \( M \) is in state \( q' \) and reads the string \( x \in \Sigma^* \) then it will end up in state \( q'' \).

- \( M \) accepts a string \( x \in \Sigma^* \) if \( q_0 \xrightarrow{x} q \) where \( q \in F \).

- The language recognized by \( M \) is the language

\[ L(M) = \{ x \in \Sigma^* \mid M \text{ accepts } x \} \]

over the alphabet \( \Sigma \).
This is a DFA formally specified by:

\[ M = (\{0, 1, 2, 3\}, \{a, b\}, 0, \{3\}, \delta), \]

where the transition function \( \delta \) is defined by:

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Regular Languages

**Definition:** A language $L \subseteq \Sigma^*$ is **regular** if there is a DFA $M$ such that $L = L(M)$.

**Examples:** The following languages are regular:

\[
\begin{align*}
L_1 &= \{xaaa y \mid x, y \in \{a, b\}^*\} \subseteq \{a, b\}^*, \\
L_2 &= \{x1102 \mid x \in \{0, 1, \ldots, 9\}^*\} \subseteq \{0, 1, \ldots, 9\}^*, \\
L_3 &= \{x \in \{0, 1\}^* \mid x \text{ contains an even number of 0s}\} \subseteq \{0, 1\}^*.
\end{align*}
\]
Which of the following languages are regular?

\[ L_4 = \{ abc x \mid x \in \{ a, b, c \}^* \} \subseteq \{ a, b, c \}^*, \]
\[ L_5 = \{ \varepsilon \} \subseteq \{ a, b \}^*, \]
\[ L_6 = \{ 0^n1^n \mid n \in \mathbb{N}_0 \} \subseteq \{ 0, 1 \}^*. \]
Non-regular Languages

To prove that a language is regular, we just have to produce a DFA accepting the language, but . . .

. . . how can we prove that a language is NOT regular?
The Pumping Lemma

Let

\[ M = (Q, \Sigma, q_0, F, \delta) \]

be a DFA with \( k \) states, and let \( x \in L(M) \) be a string that it accepts.

If \( |x| \geq k \) then there exist three strings \( u, v, w \in \Sigma^* \) such that the four properties below all hold:

(i) \( uvw = x \),

(ii) \( |v| \geq 1 \),

(iii) \( |uv| \leq k \),

(iv) for every \( i \in \mathbb{N}_0 \): \( uv^i w \in L(M) \).
Proof of the Pumping Lemma

Suppose \( q_0 \xrightarrow{x} q_F \) where \( q_F \in F \).

\(|x| \geq k, |Q| = k \implies \text{There is a state } q \in Q \text{ that occurs twice in the first } k + 1 \text{ steps of the computation.} \)

Thus we can write

\[
q_0 \xrightarrow{u} q \xrightarrow{v} q \xrightarrow{w} q_F
\]

for suitable \( u, v, w \in \Sigma^* \) satisfying (i)–(iii).

But then we also have

\[
\begin{align*}
q_0 & \xrightarrow{u} q \xrightarrow{v} q \xrightarrow{w} q_F \\
q_0 & \xrightarrow{u} q \xrightarrow{v} q \xrightarrow{v} q \xrightarrow{w} q_F \\
q_0 & \xrightarrow{u} q \xrightarrow{v} q \xrightarrow{v} q \xrightarrow{v} q \xrightarrow{w} q_F \\
\vdots
\end{align*}
\]

In other words, for every \( i \geq 0 \) we have \( q_0 \xrightarrow{u} q \xrightarrow{v^i} q \xrightarrow{w} q_F \). Thus \( M \) accepts \( uv^i w \).
Application of the Pumping Lemma I

**Theorem:** The language $L = \{0^n1^n \mid n \in \mathbb{N}_0\} \subseteq \{0, 1\}^*$ is not regular.

**Proof:** Suppose for contradiction that $L = L(M)$ for an automaton $M$ with $k$ states.

1. Pumping Lemma applied to $x = 0^k1^k \in L$ yields $u, v, w \in \{0, 1\}^*$ satisfying (i)–(iv).

2. (iii) $\implies u,v$ only consist of 0s.

3. Thus (ii) $\implies uv^2w \not\in L$.

4. But (iv) $\implies uv^2w \in L(M)$ — contradiction!
Applications of the Pumping Lemma II

Theorem: The language

\[ L = \{ x \in \{(, )\}^* \mid \text{the parenthesis in } x \text{ are well balanced} \} \]

is not regular.

Proof: Suppose for contradiction that \( L = L(M) \) for an automaton \( M \) with \( k \) states.

1. Pumping Lemma applied to \( x = (^k)^k \in L \) yields \( u, v, w \in \{(, )\}^* \) satisfying (i)–(iv).

2. (iii) \( \implies \) \( u, v \) only consist of ‘(‘s.

3. Thus (ii) \( \implies \) \( uv^2w \notin L \).

4. But (iv) \( \implies \) \( uv^2w \in L(M) \) — contradiction!
Applications of the Pumping Lemma III

**Theorem:** The language $\text{JAVA} \subseteq \text{ASCII}^*$ consisting of all syntactically correct JAVA programs is not regular.

**Hint for the proof:**
Consider the following family of syntactically correct JAVA programs:

- class A { public static void main(String[] args) { int A = 1; } }
- class A { public static void main(String[] args) {{ int A = 1; }} }
- class A { public static void main(String[] args) {{{ int A = 1; }}} }
- class A { public static void main(String[] args) {{{{ int A = 1; }}}} }
  
  .
  .
  .
How to show that a language is not regular

• Begin by **assuming that the language is regular**. The idea is to use the Pumping Lemma to reach a contradiction from this assumption.

• Because $L$ is assumed to be regular, there must be some DFA $M$ that recognises it. Write $k$ for the number of states in $M$.

• Choose some string $x$ in $L$ with $|x| \geq k$.

• Apply the Pumping Lemma to $x$. The Pumping Lemma breaks $x$ up into suitable $u$, $v$ and $w$ satisfying properties (i)–(iv).

• Choose $i \in \mathbb{N}_0$ so that $uv^i w \notin L$ according to the original definition of $L$.

• This contradicts property (iv) of the Pumping Lemma, which guarantees that $uv^i w \in L(M)$, since $L = L(M)$.

• Having reached the sought **contradiction**, conclude that the initial assumption (that $L$ is regular) is flawed.
Applications of the Pumping Lemma IV

**Theorem:** The language $L = \{a^p \mid p \text{ is a prime number}\} \subseteq \{a\}^*$ is not regular.

**Proof:** Suppose for contradiction that $L = L(M)$ for an automaton $M$ with $k$ states.

1. Pumping Lemma applied to $x = a^p$, where $p$ is a prime number bigger than $k$, yields $u, v, w \in \{a\}^*$ satisfying (i)–(iv).

2. Let $x' = uv^{p+1}w$. Then (iv) $\implies x' \in L(M)$.

3. Let $l = |u|$ and $m = |v|$, so that $|w| = p - m - l$ with $m \geq 1$ and $l + m \leq k$. Now $|x'| = l + m(p + 1) + (p - m - l) = (m + 1)p$, which is not a prime number.

4. Thus $x' \not\in L$ — contradiction!