Finite Automata and Regular Languages (part I)

Prof. Dan A. Simovici

UMB
Informally, a deterministic finite automaton consists of:

- an **input tape** divided into cells;
- a **control device** equipped with a **reading head** that scans the input tape one cell at a time.

Each cell of the input tape contains a symbol $a \in A$, where $A$ is an alphabet, called the **input alphabet**. The tape can accommodate words of arbitrary finite length. Thus, although the tape is thought of as being infinitely long, only a **finite initial segment** of it contains input symbols.
Main Components of a Finite Automaton

Control device

Read head

\[ a_{i_0} \quad a_{i_1} \quad a_{i_2} \quad a_{i_3} \quad a_{i_4} \quad \ldots \]

input tape
How a finite automaton works

- A dfa works discretely. Consider a clock that advances in discrete units; at any time on the clock, the automaton is resting in one of its states.
- Between two successive clock times, the automaton consumes its next available input and goes into a new state (which may happen to be the same state it was in at the previous time).
- The time scale of the automaton is the set $\mathbb{N}$ of natural numbers.
Definition

A deterministic finite automaton (DFA) is a quintuple

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

where \( A \) and \( Q \) are two finite, disjoint sets called the input alphabet of \( M \), and the set of states of \( M \), respectively, \( \delta : Q \times A \rightarrow Q \) is the transition function, \( q_0 \) is the initial state of \( M \), and \( F \subseteq Q \) is the set of final states of \( M \).
Example

Let $\mathcal{M} = (\{a, b\}, \{q_0, q_1, q_2, q_3\}, \delta, q_0, \{q_3\})$ be the dfa defined by the following table:

<table>
<thead>
<tr>
<th>Input</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$q_0$</td>
</tr>
<tr>
<td>$a$</td>
<td>$q_1$</td>
</tr>
<tr>
<td>$b$</td>
<td>$q_2$</td>
</tr>
</tbody>
</table>

The entry that corresponds to the input line labeled $i$ and the state column labeled $q$ gives the value of $\delta(q, i)$. 
The graph of the deterministic finite automaton $\mathcal{M} = (A, Q, \delta, q_0, F)$ is the graph $\mathcal{G}(\mathcal{M})$ whose set of vertices is the set of states $Q$.

The set of edges of $\mathcal{G}(\mathcal{M})$ consists of all pairs $(q, q')$ such that there is a transition from $q$ to $q'$; an edge $(q, q')$ is labeled by the symbol $a$ if $\delta(q, a) = q'$.

The initial state $q_0$ is denoted by an incoming arrow with no source, and the final states are circled.
Example
The graph of the previous DFA is:
The Work of a dfa

- the symbols of a word $x = a_{i_0} \cdots a_{i_{n-1}}$ are read by the automaton one at a time;
- to compute the state reached by the dfa after the application of $x$, the function $\delta$ must be extended from single symbols to a function $\delta^*$ defined for words.
Extending the Transition Function

Starting from a function $\delta : Q \times A \rightarrow Q$ we define the function $\delta^* : Q \times A^* \rightarrow Q$ by:

$$
\begin{align*}
\delta^*(q, \lambda) & = q \\
\delta^*(q, xa) & = \delta(\delta^*(q, x), a),
\end{align*}
$$

for every $x \in A^*$ and $a \in A$.

Note that for single character words, e.g., $y = a$, where $a \in A$, $\delta^*(q, y) = \delta(q, a)$. This follows from by setting $x = \lambda$ and noticing that $y = \lambda a$. Thus,

$$
\delta^*(q, a) = \delta(q, a) \text{ for all } q \in Q \text{ and } a \in A,
$$

justifying our observation that $\delta^*$ extends $\delta$. 

Theorem

Let $\delta : Q \times A \rightarrow Q$ be a function, and let $\delta^*$ be its extension to $Q \times A^*$. Then

$$\delta^*(q, xy) = \delta^*(\delta^*(q, x), y)$$

for every $x, y \in A^*$.

Proof.

The argument is by induction on $|y|$. The basis step, $|y| = 0$, is immediate since the equality of the theorem amounts to

$$\delta^*(q, x \lambda) = \delta^*(\delta^*(q, x), \lambda) = \delta^*(q, x).$$
Proof (cont’d)

For the induction step, suppose that the equality holds for words of length less or equal to $n$, and let $y$ be a word of length $n + 1$, $y = za$, where $z \in A^*$ and $a \in A$. We have

$$
\delta^*(q, xy) = \delta^*(q, xza)
$$

$$
= \delta(\delta^*(q, xz), a) \text{ (since } \delta^* \text{ extends } \delta) 
$$

$$
= \delta(\delta^*(\delta^*(q, x), z), a) \text{ (ind. hyp.)} 
$$

$$
= \delta^*(\delta^*(q, x), za) \text{ (since } \delta^* \text{ extends } \delta) 
$$

$$
= \delta^*(\delta^*(q, x), y). 
$$
Definition

The language accepted by the dfa $\mathcal{M} = (A, Q, \delta, q_0, F)$ is the set

$$L(\mathcal{M}) = \{ x \in A^* \mid \delta^*(q_0, x) \in F \}.$$ 

A language $L \subseteq A^*$ is regular if it is accepted by some finite automaton $\mathcal{M}$ whose input alphabet is $A$. 
Example
Let $\mathcal{M} = (A, Q, \delta, q_0, F)$ be the dfa whose graph is given below, where $A = \{a, b\}$ and $Q = \{q_0, q_1, q_2\}$.
The language accepted by \( \mathcal{M} \) consists of all words over \( A \) that contain at least two consecutive \( b \) symbols; in other words, \( L(\mathcal{M}) = A^* bbA^* \).
if $x \in L(M)$, then $x$ contains two consecutive $b$ symbols since $q_2$ cannot be reached otherwise from $q_0$ using the symbols of $x$;

conversely, suppose that $x$ contains two consecutive $b$ symbols; we can decompose $x = ubbv$, where $bb$ is the leftmost occurrence of $bb$ in $x$.

The definition of $M$ implies that $\delta^*(q_0, u) = q_0$, $\delta^*(q_0, bb) = q_2$ and $\delta^*(q_2, v) = q_2$. Thus, $\delta^*(q_0, x) = q_2$, and this implies $x \in L(M)$. We conclude that $L(M) = A^* bb A^*$. 
Counting Numbers

The DFA with $n$ states shown below accepts only inputs whose length is 0 (mod $n$), that is, an integral multiple of $n$. 
Example

The DFA given below accepts those words in \( \{a, b\}^\ast \) that have \( 0(\text{mod } n) \) \( a \)'s, regardless of how many \( b \)'s are in the input.
Example

Next, we present a dfa that accepts words over the alphabet \{0, 1\} only when their binary equivalents are multiples of a fixed integer, say \(m \in \mathbb{N}\).

Let \(B = \{0, 1\}\). A word \(x \in B^*\) can be regarded as a binary number as follows. Define the function \(f : B^* \rightarrow \mathbb{N}\) by

\[
\begin{align*}
f(\lambda) &= 0 \\
f(xb) &= \begin{cases} 
2f(x) + 0 & \text{if } b = 0 \\
2f(x) + 1 & \text{if } b = 1, 
\end{cases}
\end{align*}
\]

for every \(x \in B^*\) and \(b \in B\). Note that \(f(x)\) is the value represented by \(x\) regarded as a binary number.
Let \( m \in \mathbb{N} \) be a number such that \( m > 1 \). Note that for every \( x \in B^* \), there exists a number \( k \), \( 0 \leq k \leq m - 1 \), such that \( f(x) \equiv k(\text{mod } m) \). Of course, if \( f(x) \equiv 0(\text{mod } m) \), then \( f(x) \) is a multiple of \( m \), so \( x \) will be accepted by the automaton that we intend to define.

We design an automaton \( M_m \) that accepts the set of words \( x \) such that \( f(x) \) is a multiple of a fixed number \( m \). The states of \( M_m \) are defined such that \( \delta^*(q_0, x) = q_h \) if and only if \( f(x) \equiv h(\text{mod } m) \). In other words, if \( M_m \) reaches the state \( q_h \) after reading the symbols of \( x \), then \( f(x) \) is congruent to \( h \) modulo \( m \). Therefore, after reading the symbol \( b \), \( M \) enters the state \( q_\ell \), where \( 2h + b \equiv \ell(\text{mod } m) \). This allows us to define the transition function by \( \delta(q_h, b) = q_\ell \).
The dfa $M_3 = (B, \{q_0, q_1, q_2\}, \delta, q_0, \{q_0\})$ that recognizes the set of multiples of 3 is defined by the table:

<table>
<thead>
<tr>
<th>Input</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$q_0$</td>
</tr>
<tr>
<td>0</td>
<td>$q_0$</td>
</tr>
<tr>
<td>1</td>
<td>$q_1$</td>
</tr>
</tbody>
</table>

Therefore, the language $L = \{x \in B^* \mid f(x) \equiv 0(\text{mod } 3)\}$ is regular.
Example

Let $A = \{a, b, \ldots, z, 0, \ldots, 9\}$. The automaton

$$
\mathcal{M} = \{A, \{q_0, q_1, q_2\}, \delta, q_0, \{q_1\}\}
$$

accepts those words in $A^*$ that begin with a letter and contain a sequence of letters and digits. In other words, $L(\mathcal{M}) = \{a, \ldots, z\}A^*$
The finiteness of the set of states $Q$ of a dfa $M = (A, Q, \delta, q_0, F)$ is essential for the definition of regular languages. If this assumption is dropped we obtain a weaker type of device.

Definition

A deterministic automaton (da) is a quintuple

$$M = (A, Q, \delta, q_0, F),$$

where $A$ is an alphabet, called the input alphabet; $Q$ is a set that is disjoint from $A$, called the set of states, $\delta : Q \times A \rightarrow Q$ is the transition function of the da, $q_0$ is the initial state, and $F \subseteq Q$ is the set of final states.

The transition function $\delta$ can be extended to $Q \times A^*$ in exactly the same way as for the deterministic finite automata. Again, we denote this extension by $\delta^*$. 
The role of the finiteness of the set of states of a dfa is highlighted by the next theorem.

**Theorem**

*For every language* $L \subseteq A^*$, *there is a deterministic automaton* $\mathcal{M} = (A, Q, \delta, q_0, F)$ *such that* $L = L(\mathcal{M})$.

**Proof.**

Consider the da $\mathcal{M} = (A, Q, \delta, q_\lambda, \{q_u \mid u \in L\})$, where $Q = \{q_x \mid x \in A^*\}$ and $\delta(q_x, a) = q_{xa}$ for every $x \in A^*$ and $a \in A$. It is easy to verify that $\delta^*(q_x, y) = q_{xy}$ for every $x, y \in A^*$. Therefore, $L(\mathcal{M}) = \{y \in A^* \mid \delta^*(q_\lambda, y) = q_y \text{ and } y \in L\} = L$, which means that $L$ is the language accepted by $\mathcal{M}$. 

□
Definition

Let $\mathcal{M} = (A, Q, \delta, q_0, F)$ be an automaton. The set of accessible states is the set

$$\text{acc}(\mathcal{M}) = \{q \in Q \mid \delta^*(q_0, x) = q \text{ for some } x \in A^*\}.$$  

The automaton $\mathcal{M}$ is accessible if $\text{acc}(\mathcal{M}) = Q$. 
Only the set of accessible states plays a role in defining the language accepted by the automaton.

- If $\delta'$ is the restriction of $\delta$ to $\text{acc}(M) \times A$, then the automata $M$ and $M' = (A, \text{acc}(M), \delta', q_0, F \cap \text{acc}(M))$ accept the same language.
- If $x \in L(M)$, then $\delta^*(q_0, x) \in F$ and $\delta^*(q_0, y) \in \text{acc}(M)$ for every prefix $y$ of $x$ (including $x$). Therefore, $(\delta')^*(q_0, x) = \delta^*(q_0, x) \in F$, so $x \in L(M')$.
- It is immediate that $x \in L(M')$ implies $x \in L(M)$, so $L(M) = L(M')$.

$M'$ is denoted by $\text{ACC}(M)$ and we refer to it as the accessible component of $M$. 
Example

Consider an automaton $\mathcal{M} = (\{a\}, Q, \delta, q_0, F)$ having a one-symbol input alphabet. We have $\text{acc}(\mathcal{M}) = \{\delta(q_0, a^n) \mid n \in \mathbb{N}\}$. Therefore, the subgraph of the accessible states in the graph of $\mathcal{M}$ consists of a path attached to a circuit, as shown:
Theorem

Let \( \mathcal{M} = (A, Q, \delta, q_0, F) \) be an accessible automaton. For every state \( q \in Q \) there is a word \( x \in A^* \) such that \( |x| < |Q| \) and \( \delta^*(q_0, x) = q \).

Proof.

Since \( \mathcal{M} \) is an accessible automaton, for every state \( q \in Q \) there is a word \( y \) such that \( \delta^*(q_0, y) = q \). Let \( x \) be a word of minimal length that allows \( \mathcal{M} \) to reach the state \( q \). We claim that \( |x| < |Q| \). Let \( x = a_{i_0} \cdots a_{i_p} \), and let \( q_1, \ldots, q_{p+1} \) be the sequence of states reached while processing \( x \), i.e.,

\[
q_1 = \delta(q_0, a_{i_0}) = \delta(q_1, a_{i_1}) = \cdots = \delta(q_p, a_{i_p}) = q_{p+1}
\]

that is, the sequence of states assumed by \( \mathcal{M} \) when the symbols of \( x \) are applied starting from the state \( q_0 \).
If $p + 1 \geq |Q|$, then the sequence $(q_0, q_1, \ldots, q_{p+1})$ must contain two equal states because its length exceeds the number of elements of $Q$. If, say, $q_c = q_d$, we can write $x = uvw$, where $\delta^*(q_0, u) = q_c$, $\delta^*(q_c, v) = q_d$, $\delta^*(q_d, w) = q_{p+1}$ and $|v| > 0$. Since $q_d = q_c$, we have $\delta^*(q_0, uw) = q_{p+1} = q$, and this contradicts the minimality of $x$. Therefore, $|x| < |Q|$.
Computing The Accessible States

**Input:** A dfa $\mathcal{M} = (A, Q, \delta, q_0, F)$.

**Output:** The set $\text{acc}(\mathcal{M})$.

**Method:** Define the sequence $Q_0, Q_1, \ldots, Q_n, \ldots$ by

$Q_0 = \{q_0\}$ and $Q_{i+1} = Q_i \cup \{s = \delta(q, a) \mid q \in Q_i \text{ and } a \in A\}$.

$\text{acc}(\mathcal{M}) = Q_k$, where $k$ is the least number such that $Q_k = Q_{k+1}$. 
Proof of Correctness

Since $Q_0, \ldots, Q_i, \ldots$ is an increasing sequence and all sets $Q_i$ are subsets of the finite set $Q$, there is a number $k$ such that $Q_0 \subset Q_1 \subset \cdots \subset Q_k = Q_{k+1} = \cdots$.

We claim that

$$Q_i = \{ q \in Q \mid \delta^* (q_0, x) = q, \text{ for some } x \in A^*, |x| \leq i \},$$

for every $i \in \mathbb{N}$. The argument is by induction on $i$ and is omitted. Thus, every state in $Q_k$ belongs to $\text{acc} (M)$.

Conversely, if $q \in \text{acc} (M)$, then there is a word $x$ such that $|x| < |Q|$ and $\delta^* (q_0, x) = q$. Therefore, $q \in Q_{|x|} \subseteq Q_k$. We conclude that $\text{acc} (M) = Q_k$. 
Let $\mathcal{M} = (\{a, b\}, \{q_i \mid 0 \leq i \leq 7\}, \delta, q_0, \{q_5, q_6\})$ be the dfa whose graph is shown:
Thus, \( \text{ACC}(M) \) is the dfa \( M' = (\{a, b\}, \{q_0, q_1, q_2, q_4, q_5\}, \delta', q_0, \{q_5\}) \) whose graph is given next.
Deterministic Finite Automata

![Deterministic Finite Automata Diagram]

- **States:** $q_0, q_1, q_2, q_3, q_4, q_5$
- **Alphabet:** $a, b$
- **Transition Rules:**
  - $q_0 \xrightarrow{a} q_1$
  - $q_1 \xrightarrow{b} q_2$
  - $q_2 \xrightarrow{a} q_3$
  - $q_2 \xrightarrow{b} q_4$
  - $q_3 \xrightarrow{a} q_4$
  - $q_4 \xrightarrow{b} q_5$
  - $q_5 \xrightarrow{a} q_3$

**Final States:** $q_3, q_4, q_5$