Finite Automata

Alexander Meduna, Lukáš Vrábel, and Petr Zemek

Brno University of Technology, Faculty of Information Technology

Božetěchova 1/2, 612 00 Brno, CZ

http://www.fit.vutbr.cz/∼{meduna,ivrabel,izemek}

Supported by the FRVŠ MŠMT FR271/2012/G1 grant, 2012.

Contents

Definition

A *finite automaton* is a quintuple

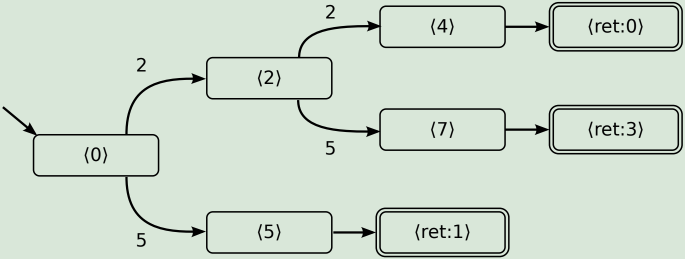
$$M = (Q, \Sigma, R, s, F)$$

where

- Q is a finite set of *states*,
- Σ is a finite set of *input symbols*,
- $R \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ is a finite relation, called the set of *rules*,
- $s \in Q$ is the *start state*, and
- $F \subseteq Q$ is a set of *final states*.

Example

Consider the following model of a coke-vending machine:



Finite Automaton (Example 2/2)

Example

Let $M = (Q, \Sigma, R, \langle 0 \rangle, F)$ be a finite automaton, where

$$
\begin{array}{rcl}
Q & = & \{\langle 0\rangle, \langle 2\rangle, \langle 4\rangle, \langle 5\rangle, \langle 7\rangle, \langle \text{ret:0}\rangle, \langle \text{ret:1}\rangle, \langle \text{ret:3}\rangle\} \\
\Sigma & = & \{2, 5\} \\
R & = & \{\langle 0\rangle 2 \rightarrow \langle 2\rangle, \langle 0\rangle 5 \rightarrow \langle 5\rangle, \langle 2\rangle 2 \rightarrow \langle 4\rangle, \langle 2\rangle 5 \rightarrow \langle 7\rangle, \\
 & & \ \langle 4\rangle \varepsilon \rightarrow \langle \text{ret:0}\rangle, \langle 7\rangle \varepsilon \rightarrow \langle \text{ret:3}\rangle, \langle 5\rangle \varepsilon \rightarrow \langle \text{ret:1}\rangle\} \\
F & = & \{\langle \text{ret:0}\rangle, \langle \text{ret:1}\rangle, \langle \text{ret:3}\rangle\}
\end{array}
$$

Definition

Let $M = Q, \Sigma, R, s, F)$ be a finite automaton. M is a *deterministic finite automaton* if R is a function from $Q \times \Sigma$ to Q.

Deterministic Finite Automaton

Definition

Let $M = Q, \Sigma, R, s, F)$ be a finite automaton. M is a *deterministic finite automaton* if R is a function from $Q \times \Sigma$ to Q.

Example

Consider the following finite automaton:

$$M = (Q, \Sigma, R, s, F)$$

where $Q = \{s, f\}$, $\Sigma = \{a, b, c\}$, $R = \{sa \rightarrow s, sb \rightarrow f, fc \rightarrow f\}$, and $F = \{f\}$. M is a deterministic finite automaton.

Configuration

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. A *configuration* of M is any pair from $Q \times \Sigma^*$.

Configuration

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. A *configuration* of M is any pair from $Q \times \Sigma^*$.

Example

Consider the finite automaton from the previous example:

$$M = \left(Q, \Sigma, R, s, F\right)$$

where $Q = \{s, f\}$, $\Sigma = \{a, b, c\}$, $R = \{sa \rightarrow s, sb \rightarrow f, fc \rightarrow f\}$, and $F = \{f\}$. Then, examples of configurations are (s, abc), $(p, ccbd)$, and (f, ε). Notice that the unread part of the input string can be empty, like in the case of (f, ε).

Direct Move Relation

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. The *direct move relation* over $Q \times \Sigma^*$, symbolically denoted by $\vdash_M$, is defined as follows: $(p, ax) \vdash_M (q, x)$ in M if and only if $pa \rightarrow q \in R$ and $x \in \Sigma^*$.

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. The *direct move relation* over $Q \times \Sigma^*$, symbolically denoted by $\vdash_M$, is defined as follows: $(p, ax) \vdash_M (q, x)$ in M if and only if $pa \rightarrow q \in R$ and $x \in \Sigma^*$.

Example

Consider the finite automaton from the previous example:

$$M = (Q, \Sigma, R, s, F)$$

where $Q = \{s, f\}$, $\Sigma = \{a, b, c\}$, $R = \{sa \rightarrow s, sb \rightarrow f, fc \rightarrow f\}$, and $F = \{f\}$. Then, for example, $(s, abc) \vdash_M (s, bc)$ by using the rule $sa \rightarrow s$, $(s, bc) \vdash_M (f, c)$ by the rule $sb \rightarrow f$, and $(f, c) \vdash_M (f, \varepsilon)$ by using the rule $fc \rightarrow f$.

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. If

$$c_1 \vdash_M c_2 \vdash_M \cdots \vdash_M c_n$$

where c_i is a configuration of M for $1 \leq i \leq n$ and $n \geq 1$, then we write $c_1 \vdash_M^* c_n$ (a *computation*).

Computation

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. If

$$c_1 \vdash_M c_2 \vdash_M \cdots \vdash_M c_n$$

where c_i is a configuration of M for $1 \leq i \leq n$ and $n \geq 1$, then we write $c_1 \vdash_M^* c_n$ (a *computation*).

Example

Consider the finite automaton M from the previous example. Then, for example,

$$(s, abc) \vdash_M^* (f, \varepsilon)$$

by using the rules $sa \to s$, $sb \to f$, and $fc \to f$, but also

$$(s, abc) \vdash_M^* (s, abc)$$

by using no rules.

Accepted Language

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. The *accepted language* by M is denoted by $L(M)$ and defined as

$$L(M) = \left\{ w : w \in \Sigma^*, (s, w) \vdash_M^* (f, \varepsilon) \text{ with } f \in F \right\}$$

Accepted Language

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a finite automaton. The *accepted language* by M is denoted by $L(M)$ and defined as

$$L(M) = \{w : w \in \Sigma^*, (s, w) \vdash_M^* (f, \varepsilon) \text{ with } f \in F\}$$

Example

Consider the finite automaton M from the previous example. By inspecting its set of rules and the way they can be used during a computation, we immediately see that the accepted language is composed of strings which start with an arbitrary number of as, followed by a single occurrence of b, and ended by an arbitrary number of cs. By using the standard notion involving operations over languages, we may state that the language accepted by M is

$$L(M) = \{a\}^*\{b\}\{c\}^*$$

References

A. Meduna, L. Vrábel, and P. Zemek.
Mathematical foundations of formal language theory, 2012.
http://www.fit.vutbr.cz/~izemek/frvs2012.

Discussion