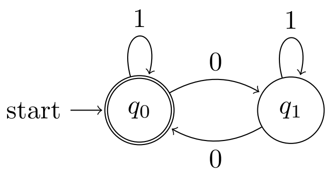
Deterministic Finite Automata

Jan C. Bierowiec

The language $\{w \in \{0,1\}^* \mid \text{the number of 0s in } w \text{ is even}\}$.

1 State Diagram

Below is the state diagram of the DFA that accepts strings with an even number of 0s:



2 Formal Description

The DFA can be described as a 5-tuple:

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

where:

$$\begin{array}{ll} Q = \{q_0, q_1\} & \text{the set of states} \\ \Sigma = \{0, 1\} & \text{the input alphabet} \\ q_0 & \text{the start state} \\ F = \{q_0\} & \text{the set of accepting states} \\ \delta & \text{the transition function:} \end{array}$$

δ	0	1
q_0	q_1	q_0
q_1	q_0	q_1

How the DFA Works

The DFA starts in state q_0. Every time a 0 is read, it transitions between q_0 and q_1, thereby toggling between even and odd counts of 0s. Reading a 1 causes the automaton to stay in the current state. If the machine ends in q_0, the input is accepted (even number of 0s).

Example Strings and Traces

Accepted Strings ε, `1`, `11`, `0011`, `1010`
Rejected Strings: `0`, `10`, `011`, `000`

Trace for 1010:

$$q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_0 \quad \text{Accepted}$$

3 Context-Free Grammar (CFG)

The CFG that generates the same language (even number of 0s) is defined as:

$$G = (V, \Sigma, R, S)$$

where:

$$\begin{array}{ll} V = \{S, A\} & \text{the set of variables} \\ \Sigma = \{0, 1\} & \text{the set of terminal symbols} \\ S & \text{the start symbol} \\ R & \text{the set of production rules} \end{array}$$

Production rules:

$$\begin{aligned} S &\to 1S \mid 0A \mid \varepsilon \\ A &\to 1A \mid 0S \end{aligned}$$

This CFG generates all strings over $\{0, 1\}^*$ with an even number of 0s.

4 Regular Expression

Since the language is regular, it can also be described by the following regular expression:

$$(1^*(01^*01^*)^*)^*$$

This expression generates all strings with pairs of 0s, possibly separated by 1s.