Theoretical Computer Science

Roger Küng

roger.kueng@epfl.ch

February 23, 2014

1 Lecutre 1

Amongst other subject we will stduy the following:

- What can (or can't) be computed?
- What can be computed with limited memory?
- what is the power of randomness?

1.1 Simple door problem

Both means there is an object in front and the back of the door, **Front** and **Back** mean that there is either a persion in front or the back of the door and **Neither** means there is no object detected by the door

Current State	Neither	Front	Back	Both
Open	Closed	Open	Open	Open
Closed	Closed	Open	Closed	Closed

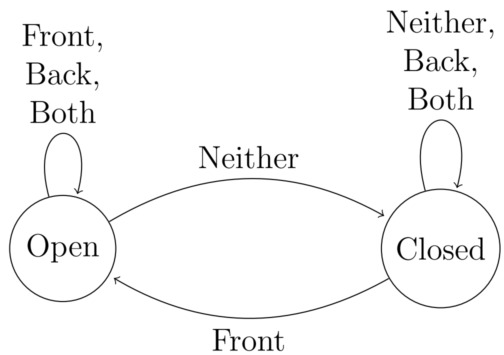

Figure 1: Door Machine

In General we can define systems in the following way:

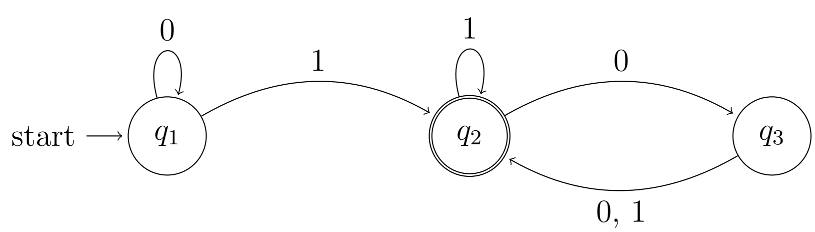

Figure 2: More general Machine

Where the ⊚ represents an accepting state.
For a given input we end up in a specific state: $001 \rightarrow q_2$

The Machine will accept and input if the endstate will be in an **accepting state**

- M accepts $0|0|1$
- M accepts $1|0|1|1$
- M rejects $0|1|0$

In general we can define the language for this machine as follows:

$$L(M) := \{\omega \in \{0,1\}^* | M \text{ accepts } \omega\}$$

The next question which arises is what strings this machine would accept. It appears that every string which ends with a 1 would be accepted by the machine, also everything ending with an **even** number of 0 with at least a 1.

$$L(M_1) = \left\{ \omega \in \{0,1\}^* \;\middle|\; \begin{array}{l} \omega = x1 \text{ for some } x \in \{0,1\} \vee \\ \omega \text{ contains a 1 and ends with an even number of 0s} \end{array} \right\}$$

1.2 DFA

Def: A deterministic Finite Automation (DFA)

$$\begin{aligned} M = \; & < Q, \Sigma, \delta, q_0, F > \\ \Sigma \Rightarrow \; & \text{ alphabet of } M(\text{ usually } \Sigma\{0,1\} \\ Q \Rightarrow \; & \text{set of states} \\ \delta \Rightarrow \; & \text{transition function of } M \\ & \delta : Q \times \Sigma \to Q \\ & \delta(q_1, 0) = q_2 \\ q_0 \Rightarrow \; & \text{starting state} \\ F \Rightarrow \; & \text{set of accepting states} \end{aligned}$$

There is also a specific symbol ϵ to indicate the empty string, this should not be confused with the empty set: $L(M_1) = \epsilon \neq L(M_2) = \varnothing$.

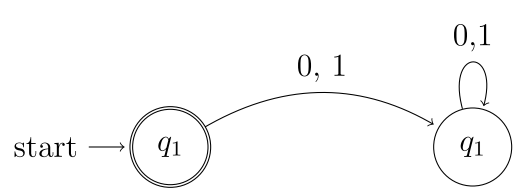

Figure 3: Machine which only accepts empty string

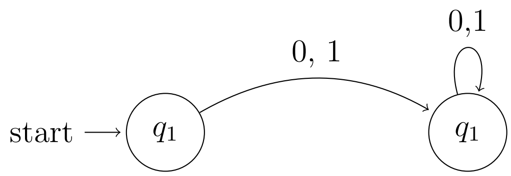

Figure 4: Empty language → no way to end in an accepting state

$$
\begin{aligned}
M_1 = &\ < Q_1, \Sigma_1, \delta_1, q_0^1, F_1 > \\
Q_1 = &\ \{q_1, q_2, q_3\} \\
\Sigma_1 = &\ \{0, 1\} \\
q_0^1 = &\ q_1 \\
F_1 = &\ \{q_2\}
\end{aligned}
$$

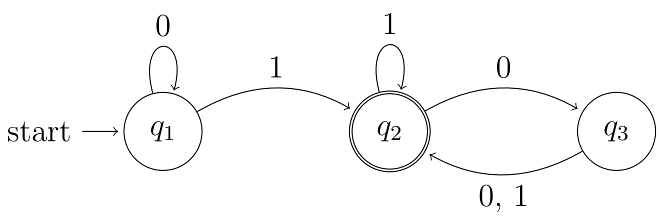

Figure 5: State Machine

δ_1	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Figure 6: State table

The best way to find out what the language a machine can process is to try out a few string to be able to find a pattern.

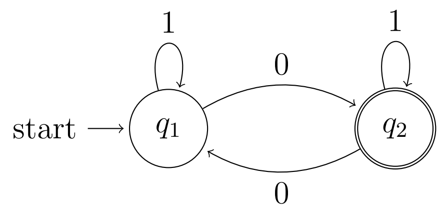

Figure 7: M3

$$M_3 = \begin{cases} 0|0|0 & \textit{Accepted} \\ 0|1|1|0|0 & \textit{Accepted} \\ 0|0|0|0 & \textit{Rejected} \end{cases}$$

We can observe that all 1's are irrelevant, therefore we should focus on the 0's. Therefore the language for the machine can be described by:

$$L(M_3) = \{\omega \in \{0,1\}^* \mid \#\text{ of 0 is odd}\}$$

A complementary Machine can be built by:

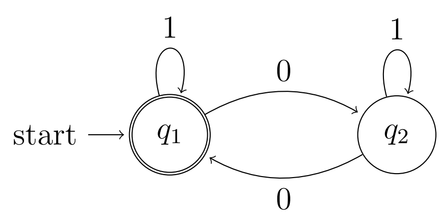

Figure 8: M4

$$L(M_4) = \{\omega \in \{0,1\}^* \mid \#\text{ of 0 is even}\}$$
$$L(M_3) = \overline{L(M_4)}$$

Lemma 1 *For any DFA:*

$$
\begin{aligned}
M &= < Q, \Sigma, \delta, q_0, F > \\
\overline{M} &= < Q, \Sigma, \delta, q_0, \overline{F} > \\
\overline{L(M)} &= L(\overline{M})
\end{aligned}
$$

1.3 Notation:

$$
\begin{aligned}
&\hat{\delta} : Q \times \Sigma^* \Rightarrow Q \\
&\underset{q \in Q}{\forall} \hat{\delta}(q, \epsilon) = q \\
&\underset{b \in \Sigma}{\forall} \hat{\delta}(q, b\omega) = \hat{\delta}(\underbrace{\delta(q, b)}_{\text{recursion}}, \omega)
\end{aligned}
$$

$$M \text{ accepts } \omega \iff \hat{\delta}(q_0, \omega) \in F$$

Definition 1

$$\textit{Language } L \subseteq \{0,1\}^* \textit{is } \underline{\textit{regular}} \iff \exists \textit{ DFA } M \textit{ s.t. } L(M) = L$$

Corollary 1

$$\forall L \subseteq \{0,1\}^* \textit{ if } L \textit{ is regular then so is} \overline{L}$$

Definition 2 *Given a machine M we define $L(M)$ to be the language recognised by M the set of all strings accepted by M*