2.6 Automata with Output

A Moore automaton is a DFA in which an output symbol is assigned to each state. Accordingly, a Moore automaton A is given through a 6-tuple $A = (Q, \Sigma, \Delta, \delta, \sigma, q_0)$, where

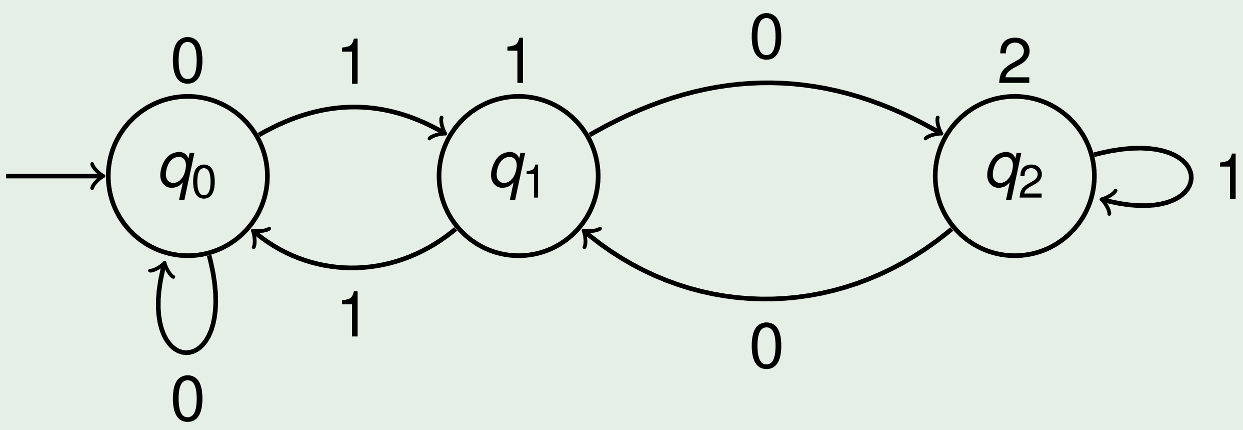
- Q is a finite set of (internal) states,
- Σ is a finite input alphabet,
- Δ is a finite output alphabet,
- $q_0 \in Q$ is the initial state,
- $\delta : Q \times \Sigma \to Q$ is the transition function, and
- $\sigma : Q \to \Delta$ is the output function.

For $u = a_1 a_2 \ldots a_n$, let $q_i := \delta(q_0, a_1 a_2 \ldots a_i)$, $i = 1, 2, \ldots, n$, that is, on input u, A visits the sequence of states $q_0, q_1, q_2, \ldots, q_n$. During this computation A generates the output

$$\sigma(q_0)\sigma(q_1)\sigma(q_2)\ldots\sigma(q_n) \in \Delta^{n+1}.$$

Example:

Let A be the Moore automaton that is given by the following graph, where the output symbols are written as external markings to the various states:



Example (cont.):

Claim:

If $u = b_1 b_2 \dots b_n$, then $\sigma(u) = r_0 r_1 \dots r_n$, where $r_i \equiv \sum_{j=1}^{i} b_j \cdot 2^{i-j} \bmod 3$.

If $u = \mathrm{bin}(m)$, that is, $m = \sum_{j=1}^{n} b_j \cdot 2^{n-j}$, then the last symbol r_n of the output $\sigma(u)$ is just the remainder of $m \bmod 3$.

Proof.

For $i = 0, 1, 2$, $\sigma(q_i) = i$. Hence, it suffices to prove the following:

$(*)$ For all $u = b_1 b_2 \dots b_n$, $\delta(q_0, u) = q_i$, where $i \equiv \sum_{j=1}^{n} b_j \cdot 2^{n-j} \bmod 3$.

If $n = 0$, then $u = \varepsilon$, und $\delta(q_0, u) = q_0$.

If $n = 1$, then $u = b_1 \in \{0, 1\}$. Hence, $\delta(q_0, u) = \begin{cases} q_0, & \text{for } b_1 = 0, \\ q_1, & \text{for } b_1 = 1. \end{cases}$

Example (cont.):

Proof (cont.)

Assume that the statement $(*)$ has been verified for some $n \geq 1$, and let $u = b_1 b_2 \ldots b_n b_{n+1}$.

Then $\delta(q_0, u) = \delta(\delta(q_0, b_1 b_2 \ldots b_n), b_{n+1})$.

For $i = n$, the statement holds by the induction hypothesis.

For index $n + 1$, the following can be checked by case analysis:

$$\delta(\delta(q_0, b_1 b_2 \ldots b_n), b_{n+1}) = q_i, \text{ where } i \equiv \sum_{j=1}^{n+1} b_j \cdot 2^{n+1-j} \bmod 3.$$

This completes the proof. □

A Mealey automaton is a DFA that outputs a symbol during each transition. Accordingly, a Mealey automaton A is specified by a 6-tuple $A = (Q, \Sigma, \Delta, \delta, \sigma, q_0)$, where $Q, \Sigma, \Delta, \delta$, and q_0 are defined as for a Moore automaton, while $\sigma : Q \times \Sigma \to \Delta$ is the output function.

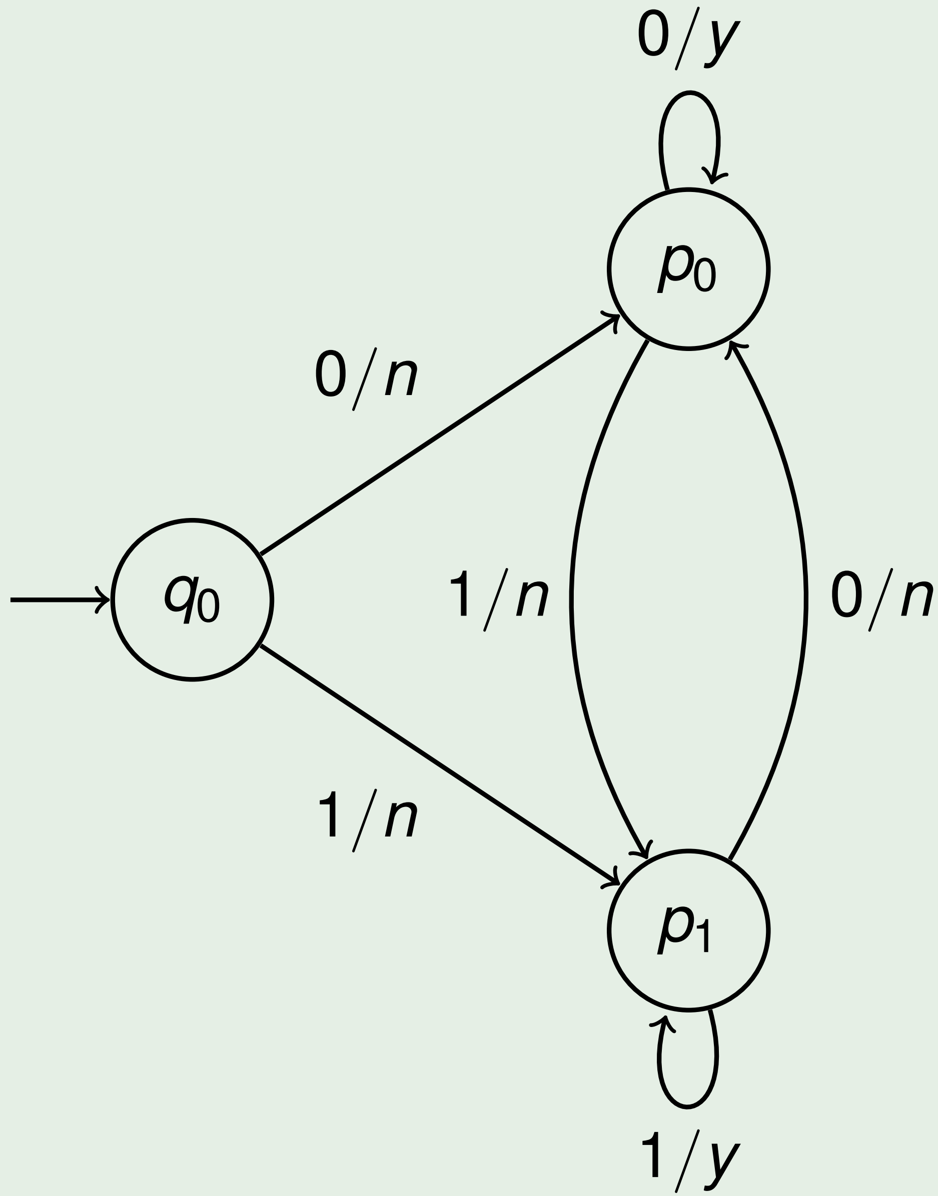
The output function σ can be extended to a function $\sigma : Q \times \Sigma^* \to \Delta^*$:

$$\begin{array}{lcll} \sigma(q, \varepsilon) & := & \varepsilon & \text{for all } q \in Q, \\ \sigma(q, ua) & := & \sigma(q, u) \cdot \sigma(\delta(q, u), a) & \text{for all } q \in Q, u \in \Sigma^*, a \in \Sigma. \end{array}$$

Example:

Let A be the Mealey automaton given through the following graph:

Example (cont.):



Example (cont.):

Let $L = \{0,1\}^* \cdot \{00,11\}$. On input $u = b_1 b_2 \dots b_m \in \{0,1\}^m$, A produces output $v = c_1 c_2 \dots c_m \in \{y,n\}^m$, where

$$c_i = \begin{cases} y, & \text{if } b_1 b_2 \dots b_i \in L, \\ n, & \text{if } b_1 b_2 \dots b_i \notin L. \end{cases}$$

In state p_0 or p_1, A “stores” the latest input symbol.

Let $A = (Q, \Sigma, \Delta, \delta, \sigma, q_0)$ be a Moore automaton and let $B = (Q', \Sigma, \Delta, \delta', \sigma', q_0')$ be a Mealey automaton. For $u \in \Sigma^*$, let $F_A(u) \in \Delta^*$ and $F_B(u) \in \Delta^*$ be the output words that are generated by A and B on input u. Then $|F_A(u)| = |u| + 1$ and $|F_B(u)| = |u|$.

The automata A und B are called equivalent, if

$$F_A(u) = \sigma(q_0) \cdot F_B(u)$$

for all $u \in \Sigma^*$.

Theorem 2.26

(a) *For each Moore automaton, there exists an equivalent Mealey automaton.*

(b) *For each Mealey automaton, there exists an equivalent Moore Automaton.*

Proof.

As an exercise! □

In fact, the equivalent automata can be constructed effectively!

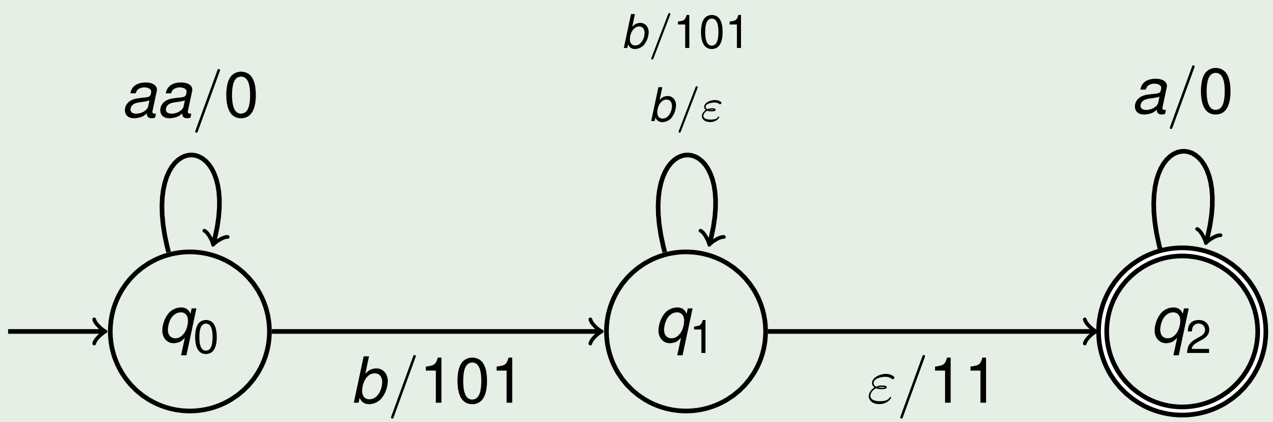
A finite-state transducer (FST) T is given through a 6-tuple

$$T = (Q, \Sigma, \Delta, \delta, q_0, F),$$

where Q, Σ, Δ, and q_0 are defined as for a Mealey automaton,
$F \subseteq Q$ is a set of final states,
and $\delta : D \to 2^{Q\times\Delta^*}$ is the transition and output function.
Here D is a finite subset of $Q \times \Sigma^*$, and
δ associates a finite subset of $Q \times \Delta^*$ to each pair $(q, u) \in D$.

Example:

Let T be the following finite-state transducer:



For $u \in \Sigma^*$, $v \in \Delta^*$ is a possible output of T, if u admits a factorisation of the form $u = u_1 u_2 \cdots u_n$ such that there are states $q_1, q_2, \ldots, q_n \in Q$ and transitions

$$\delta(q_0, u_1) \ni (q_1, v_1), \delta(q_1, u_2) \ni (q_2, v_2), \ldots, \delta(q_{n-1}, u_n) \ni (q_n, v_n)$$

such that $q_n \in F$ and $v = v_1 v_2 \cdots v_n$.
By $T(u) \subseteq \Delta^*$ we denote the set of all possible outputs of T for input u.
In this way T induces a (partial) mapping $T : \Sigma^* \to 2^{\Delta^*}$.

A mapping $\varphi : \Sigma^* \to 2^{\Delta^*}$ is called a finite transduction, if there exists a finite-state transducer T such that $T(w) = \varphi(w)$ for all $w \in \Sigma^*$.

Let $T = (Q, \Sigma, \Delta, \delta, q_0, F)$ be an FST.
For a language $L \subseteq \Sigma^*$,

$$T(L) := \bigcup_{w \in L} T(w)$$

is the image of L w.r.t. T.

For a language $L \subseteq \Delta^*$,

$$T^{-1}(L) := \{ u \in \Sigma^* \mid T(u) \cap L \neq \emptyset \}$$

is the preimage of L w.r.t. T.

By $R_T \subseteq \Sigma^* \times \Delta^*$ we denote the relation

$$R_T := \{ (u, v) \mid v \in T(u) \}.$$

Relations of this form are called rational relations.

Example (cont.):

$$\begin{array}{lcl} T(aabb) & = & \{010111, 010110111\}, \\ T(bbba) & = & \{101110, 101101110, 101101101110\}, \\ T(\varepsilon) & = & \emptyset, \\ T(aaab) & = & \emptyset, \\ T(\{b, ba\}) & = & \{10111, 101110\}, \\ T^{-1}(\{10111, 101110\}) & = & \{ b^{n+1}, b^{n+1}a \mid n \geq 0 \}. \end{array}$$

Theorem 2.27 (Nivat 1968)

Let Σ and Δ be finite alphabets, and let $R \subseteq \Sigma^* \times \Delta^*$.
The relation R is a rational relation if and only if
there exist a finite alphabet Γ, a regular language $L \subseteq \Gamma^*$, and
morphisms $g : \Gamma^* \to \Sigma^*$ and $h : \Gamma^* \to \Delta^*$ such that
$R = \{ (g(w), h(w)) \mid w \in L \}$.

Proof.

“$\Rightarrow$”: Let R be a rational relation, that is,

$$R = R_T = \{ (u, v) \mid u \in \Sigma^* \text{ and } v \in T(u) \}$$

for some FST $T = (Q, \Sigma, \Delta, \delta, q_0, F)$.
We must show that $R = \{ (g(w), h(w)) \mid w \in L \}$ for some $L \in \mathrm{REG}(\Gamma)$
and morphisms $g : \Gamma^* \to \Sigma^*$ and $h : \Gamma^* \to \Delta^*$.

Proof of Theorem 2.27 (cont.)

Let $\Gamma := \{ [q, u, v, p] \mid (p, v) \in \delta(q, u) \}$.

From the definition of T it follows that Γ is a finite set, the elements of which we interpret as letters.

We define morphisms $g : \Gamma^* \to \Sigma^*$ and $h : \Gamma^* \to \Delta^*$ through

$$g([q, u, v, p]) := u \text{ and } h([q, u, v, p]) := v.$$

Finally, let $L \subseteq \Gamma^*$ be defined as follows:
$[q_1, u_1, v_1, p_1][q_2, u_2, v_2, p_2] \cdots [q_n, u_n, v_n, p_n] \in L$ iff

1. $q_1 = q_0$,
2. $p_n \in F$, and
3. for all $i = 1, \ldots, n-1$, $p_i = q_{i+1}$.

One can easily define a DFA for this language, that is, $L \in \mathrm{REG}(\Gamma)$.

Proof of Theorem 2.27 (cont.)

The word

$$[q_1, u_1, v_1, p_1][q_2, u_2, v_2, p_2] \cdots [q_n, u_n, v_n, p_n] \in L$$

describes an accepting computation of T for input $u := u_1 u_2 \cdots u_n$ producing output $v := v_1 v_2 \cdots v_n$.

Thus, if $q_0 \notin F$, then

$$R = R_T = \{ (u, v) \mid v \in T(u) \} = \{ (g(w), h(w)) \mid w \in L \}.$$

If $q_0 \in F$, then we consider the language $L' := L \cup \{\varepsilon\}$, since then the empty computation is an accepting computation of T for input ε that produces the output ε.

Proof of Theorem 2.27 (cont.)

"$\Leftarrow$": Now let $R = \{ (g(w), h(w)) \mid w \in L \}$ for a regular language $L \subseteq \Gamma^*$. Then there is a DFA $A = (Q, \Gamma, \delta, q_0, F)$ such that $L(A) = L$. Let T be the FST $T = (Q, \Sigma, \Delta, \delta', q_0, F)$ that is obtained from A by replacing each transition $q \xrightarrow{c} q'$ of A by $q \xrightarrow{g(c)/h(c)} q'$. Then

$$R = \{ (g(w), h(w)) \mid w \in L \} = \{ (u, v) \mid v \in T(u) \}. \qquad \square$$

Corollary 2.28

If T is a finite transduction, then so is T^{-1}.

Corollary 2.29

The class REG *is closed under finite transductions and inverse finite transductions, that is, if $T \subseteq \Sigma^* \times \Delta^*$ is a finite transduction, then the following implications hold:*

1. *If $L \in \text{REG}(\Sigma)$, then $T(L) \in \text{REG}(\Delta)$.*
2. *If $L \in \text{REG}(\Delta)$, then $T^{-1}(L) \in \text{REG}(\Sigma)$.*

Proof.

By Corollary 2.28 it suffices to prove (1).
Let $T \subseteq \Sigma^* \times \Delta^*$ be a finite transduction, and let $L_1 \in \text{REG}(\Sigma)$. We must prove that $T(L_1) \in \text{REG}(\Delta)$ ist.

By Theorem 2.27, there are an alphabet Γ, a language $L \in \text{REG}(\Gamma)$, and morphisms $g : \Gamma^* \to \Sigma^*$ and $h : \Gamma^* \to \Delta^*$ such that

$$T = \{ (g(w), h(w)) \mid w \in L \}.$$

Proof of Corollary 2.29 (cont.)

We obtain the following sequence of equivalent statements:

$$\begin{aligned} T(L_1) &= \{ v \in \Delta^* \mid \exists u \in L_1 : v \in T(u) \} \\ &= \{ h(w) \mid w \in L \text{ and } \exists u \in L_1 : g(w) = u \} \\ &= \{ h(w) \mid w \in L \text{ and } g(w) \in L_1 \} \\ &= \{ h(w) \mid w \in L \cap g^{-1}(L_1) \} \\ &= h(L \cap g^{-1}(L_1)). \end{aligned}$$

By Theorem 2.13, $g^{-1}(L_1) \in \mathrm{REG}(\Gamma)$,
by Theorem 2.8, REG is closed under intersection,
which yields $L \cap g^{-1}(L_1) \in \mathrm{REG}(\Gamma)$,
and by Remark 2.4(c), REG is closed under morphisms, that is,
$T(L_1) = h(L \cap g^{-1}(L_1))$ is a regular language. □