The use of matrix theory leads to considerable insights into the behaviour of recognizable sets. However the interested reader is referred to the literature for further details (e.g. Cohn [1975]).

5.8 Exercises

5.1 Let $x \in \Sigma^*$. Prove that $\{x\}$ is recognizable.

5.2 Let $\mathfrak{M} = (\mathcal{M}, i, T)$, $\mathcal{M} = (Q, \Sigma, F)$ and $q^{-1} \circ T = q_1^{-1} \circ T \Rightarrow q = q_1$ for all $q, q_1 \in Q$. Prove that if $A = |\mathfrak{M}|$ then $\mathfrak{M} \cong \mathfrak{M}_A$.

5.3 Let $\mathfrak{M} = (\mathcal{M}, i, T)$, $\mathcal{M} = (Q, \Sigma, F)$. Define a relation E on Q by

$$qEq_1 \Leftrightarrow q^{-1} \circ T = q_1^{-1} \circ T \quad \text{for } q, q_1 \in Q.$$

Consider $\mathcal{M}/E = (Q/E, \Sigma, \bar{F})$ where $\bar{F}$ is defined by

$$[q]\bar{F}_\sigma = [qF_\sigma] \quad \text{for } [q] \in Q/E,\ \sigma \in \Sigma.$$

Show that $\mathcal{M}/E$ is well-defined and if

$$\mathfrak{M}/E = (\mathcal{M}/E, [i], \{[t]/t \in T\})$$

then

$$|\mathfrak{M}/E| = |\mathfrak{M}|.$$

5.4 With $\mathfrak{M}$ and $\mathcal{M}$ as defined in 5.3 and $n \geq 0$ consider the relation E_n on Q given by

$$qE_nq_1 \Leftrightarrow \{q\alpha \in T \Leftrightarrow q_1\alpha \in T \text{ for all } \alpha \in \Sigma^* \text{ with } |\alpha| \leq n\}.$$

Prove that $E = \bigcap_{n \geq 0} E_n$.

5.5 Complete the proof of theorem 5.3.3.

5.6 Prove theorem 5.3.5.

5.7 Prove theorem 5.3.6.

5.8 If $A, B, C \subseteq \Sigma^*$ prove that

$$(A \cdot B)^{-1} \cdot C = B^{-1}(A^{-1} \cdot C)$$
$$(A^{-1} \cdot B) \cdot C \subseteq A^{-1} \cdot (B \cdot C).$$

5.9 If $A \subseteq \Sigma^*$ is recognizable then so is $A_P = A \backslash A\Sigma^+$.

5.10 *If* π is an admissible partition on $\mathcal{M}$ what does

$$\mathfrak{M}/\pi = (\mathcal{M}/\pi, [i], [T])$$

recognize?

6

Sequential machines and functions

Mealy machines were briefly introduced in chapter 2 to provide a motivational basis for the discussion of products of state machines and transformation semigroups. In this chapter, Mealy machines and their associated functions will be examined in their own right and some of the results from earlier chapters will be applied to them.

6.1 Mealy machines again

Recall that a Mealy machine, as defined in section 2.5, is a quintuple $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ where Q, Σ and Θ are finite sets, and

$$F : Q \times \Sigma \to Q,$$
$$G : Q \times \Sigma \to \Theta$$

are functions.

Thus $\mathcal{M} = (Q, \Sigma, F)$ is a complete state machine. It is now reasonable to extend our concept slightly by including the possibilities that either $F : Q \times \Sigma \to Q$ or $G : Q \times \Sigma \to \Theta$ are partial functions rather than functions. Thus $\mathcal{M} = (Q, \Sigma, F)$ may not be complete.

A *Mealy machine* is now understood to be a quintuple $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ where Q, Σ, Θ are finite sets (Σ and Θ being non-empty) and $F : Q \times \Sigma \to Q$, $G : Q \times \Sigma \to \Theta$ partial functions. If F and G are both functions we say that $\hat{\mathcal{M}}$ is a *complete* Mealy machine.

It is now inappropriate to describe Mealy machines by directed graphs, we will have to use tables.

Example 6.1

Let

$$Q = \{q_1, q_2, q_3\} \qquad \Sigma = \{0, 1\} \qquad \Theta = \{a, b\}.$$

We define $F:Q\times\Sigma\rightarrow Q$ and $G:Q\times\Sigma\rightarrow\Theta$ by the table

$\hat{\mathcal{M}}$		q_1	q_2	q_3
F	0	q_1	$\varnothing$	q_3
	1	q_2	q_3	$\varnothing$
G	0	a	a	a
	1	$\varnothing$	b	$\varnothing$

Thus

$$q_1F_0=q_1,\quad q_2F_0 \text{ is undefined,}$$

$$q_1G_0=a,\quad q_1G_1 \text{ is undefined etc.}$$

(Some authors use a dash – instead of the symbol $\varnothing$ in such tables.)

If $\hat{\mathcal{M}}=(Q,\Sigma,\Theta,F,G)$ satisfies the property that

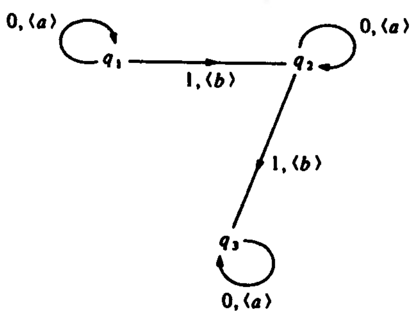
$$qF_\sigma=\varnothing\Leftrightarrow qG_\sigma=\varnothing$$

we can use the directed graph method of describing $\hat{\mathcal{M}}$. Such Mealy machines are called *normal.*

$\hat{\mathcal{M}}$ is called *state complete* if F is a function and *output complete* if G is a function.

Example 6.2

$Q=\{q_1,q_2,q_3\}$, $\Sigma=\{0,1\}$, $\Theta=\{a,b\}$.



represents the normal Mealy machine:

$\hat{\mathcal{M}}$		q_1	q_2	q_3
F	0	q_1	q_2	q_3
	1	q_2	q_3	$\varnothing$
G	0	a	a	a
	1	b	b	$\varnothing$

The output of a Mealy machine clearly depends on the set of states that are traversed in the process of the operation of the machine. We have seen that for complete machines, if $\sigma_1\sigma_2\ldots\sigma_k\in\Sigma$ and $q\in Q$ then the output word of Θ^* obtained when $\sigma_1\sigma_2\ldots\sigma_k$ is the input word and q is the initial state is given by

$$\theta_1\theta_2\ldots\theta_k$$

where

$$\theta_1=qG_{\sigma_1}$$

$$\theta_2=qF_{\sigma_1}G_{\sigma_2}$$

$$\vdots$$

$$\theta_k=qF_{\sigma_1\ldots\sigma_{k-1}}G_{\sigma_k}.$$

The state q defines a function $f_q:\Sigma^*\rightarrow\Theta^*$ described by

$$f_q(\sigma_1\sigma_2\ldots\sigma_k)=\theta_1\theta_2\ldots\theta_k,\quad \text{for } \sigma_1\sigma_2\ldots\sigma_k\in\Sigma^+.$$

It is clear that f_q satisfies the following properties

$$f_q(\sigma)=qG_\sigma$$

$$f_q(x\sigma)=(f_q(x))(qF_xG_\sigma)=f_q(x)f_{qF_x}(\sigma),\quad \text{for } \sigma\in\Sigma,\ x\in\Sigma^+,$$

and these are enough to define f_q.

We will also ask that f_q satisfies the property $f_q(\Lambda)=\Lambda$.

Proposition 6.1.1

If $\hat{\mathcal{M}}=(Q,\Sigma,\Theta,F,G)$ is a complete Mealy machine and $q\in Q$ then for $x,y\in\Sigma^*$

$$f_q(xy)=f_q(x)f_{qF_x}(y).$$

Proof We proceed by induction on the length of y. If $y=\sigma\in\Sigma$ then

$$f_q(x\sigma)=f_q(x)qF_xG_\sigma$$

$$=f_q(x)f_{qF_x}(\sigma).$$

We assume now that $f_q(xy)=f_q(x)f_{qF_x}(y)$ holds for all words $y\in\Sigma^*$ of length less than t, and states $q\in Q$.

Let $y=\sigma_1\ldots\sigma_{t-1}\sigma_t$ and put $z=\sigma_1\ldots\sigma_{t-1}$, then

$$f_q(xz)=f_q(x)f_{qF_x}(z).$$

Now

$$\begin{aligned} f_q(xy) &= f_q(xz\sigma_t) \\ &= f_q(xz)qF_{xz}G_{\sigma_t} \\ &= f_q(x)f_{qF_x}(z)qF_{xz}G_{\sigma_t} \\ &= f_q(x)f_{qF_x}(z)qF_xF_zG_{\sigma_t} \\ &= f_q(x)f_{qF_x}(z\sigma_t) \\ &= f_q(x)f_{qF_x}(y). \end{aligned}$$

For $x = \Lambda$ or $y = \Lambda$ the result is immediate. $\square$

Next we turn to the case where $\hat{\mathcal{M}}$ is not complete. For $q \in Q$ we can define a partial function $f_q : \Sigma^* \to \Theta^*$ by

$$f_q(\sigma_1\sigma_2 \ldots \sigma_k) = \theta_1\theta_2 \ldots \theta_k \quad \text{for } \sigma_1\sigma_2 \ldots \sigma_k \in \Sigma^*$$

if

$$qF_{\sigma_1}, qG_{\sigma_1}, qF_{\sigma_1\sigma_2}, qF_{\sigma_1}G_{\sigma_2}, \ldots, qF_{\sigma_1 \ldots \sigma_k}, qF_{\sigma_1 \ldots \sigma_{k-1}}G_{\sigma_k}$$

are all defined and $f_q(\Lambda) = \Lambda$.

However if $qF_{\sigma_1}, qF_{\sigma_1\sigma_2}, \ldots, qF_{\sigma_1\sigma_2 \ldots \sigma_{t-1}}$ are defined but $qF_{\sigma_1\sigma_2 \ldots \sigma_t} = \varnothing$ we note that the machine stops completely and no more output symbols can be printed.

The case where $qF_{\sigma_1}, \ldots, qF_{\sigma_1 \ldots \sigma_{t-1}}$ are all defined but $qG_{\sigma_1}, \ldots, qF_{\sigma_1 \ldots \sigma_{t-1}}G_{\sigma_t}$ are not all defined can be dealt with as follows. If $qF_{\sigma_1 \ldots \sigma_{l-1}}G_{\sigma_l} = \varnothing$ we regard the output as a blank space on the output tape. This will be denoted by $\square$. Consequently the output word could take the form

$$\theta_1 \ldots \theta_l \square \theta_{l+1} \ldots \theta_k.$$

(A slightly different interpretation will be used in sections 6.4 and 6.5.) We should not confuse $\square$ with Λ. A simple example will illustrate some of these points.

Example 6.3
$\hat{\mathcal{M}}$ is defined by the table:

$\hat{\mathcal{M}}$		q_1	q_2
F	0	$\varnothing$	q_1
	1	q_2	q_2
G	0	a	$\varnothing$
	1	b	b

so that

$$q_1F_0 = \varnothing, q_1F_1G_0 = \varnothing \text{ etc.}$$

Now $f_{q_1}(0) = a$, $f_{q_1}(10) = b\square$, $f_{q_1}(101) = b \square b$, $f_{q_1}(100) = b \square a$, and $f_{q_1}(01) = \varnothing$ since $q_1F_0 = \varnothing$. But what is $f_{q_1}(1001)$? Either $f_{q_1}(1001) = \varnothing$ since the machine stops before all the input word has been fed in, that is after 100 in this case, or we put $f_{q_1}(1001) = b \square a$, that is $f_{q_1}(1001) = f_{q_1}(100)$ where 100 is the smallest initial segment of 1001 for which the machine produces a complete output.

To avoid these problems we will only consider $f_q(x)$ to be defined if

$$qF_{\sigma_1}, qF_{\sigma_1\sigma_2}, \ldots, qF_{\sigma_1 \ldots \sigma_{l-1}}$$

are all defined where $x = \sigma_1\sigma_2 \ldots \sigma_l \in \Sigma^*$. We describe this by saying that x is *applicable* to q. This will guarantee that the image $f_q(x)$ is of the same length as the length of x whenever x is applicable to q. Notice that the length of $\square$ is 1, whereas the length of Λ is 0 and clearly in our example $f_{q_2}(\Lambda) = \Lambda$ (no tape goes in and no tape comes out!) whereas $f_{q_2}(0) = \square$ (a blank tape comes out of length 1).

If xy is applicable to q then the conclusion of 6.1.1, namely $f_q(xy) = f_q(x)f_{qF_x}(y)$ is valid. Clearly if $x \in \Sigma^*$ is not applicable to q then neither is xy for any $y \in \Sigma^*$.

We can, to a certain extent, overcome some of the difficulties concerned with the applicability of inputs by moving to the completion of the Mealy machine.

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a Mealy machine such that the state machine $\mathcal{M} = (Q, \Sigma, F)$ is incomplete. Let $\mathcal{M}^c = (Q \cup \{z\}, \Sigma, F')$ be the completion of $\mathcal{M}$ and define $G' : (Q \cup \{z\}) \times \Sigma \to \Theta$ by

$$\begin{aligned} G'(q, \sigma) &= G(q, \sigma) && \text{for } q \in Q, \sigma \in \Sigma, \\ G'(z, \sigma) &= \varnothing && \text{for } \sigma \in \Sigma. \end{aligned}$$

Then $\hat{\mathcal{M}}^c = (Q \cup \{z\}, \Sigma, \Theta, F', G')$ is called the *state completion* of $\hat{\mathcal{M}}$.

We now notice that every $x \in \Sigma^+$ is applicable to any state $q' \in Q \cup \{z\}$ and so $f'_{q'}$ is defined as a function although from Σ^* to $(\Theta \cup \{\square\})^*$.

If $\sigma_1\sigma_2 \ldots \sigma_k$ is not applicable to q in the original machine $\hat{\mathcal{M}}$ but $\sigma_1\sigma_2 \ldots \sigma_{k-1}$ is applicable to q, the output obtained by applying $\sigma_1\sigma_2 \ldots \sigma_k$ to q in the state completion is given by $f_q(\sigma_1 \ldots \sigma_{k-1})\square$.

Note that $f'_q(x) = f_q(x)$ if x is applicable to q.

In general if $\hat{\mathcal{M}}$ is an arbitrary Mealy machine and q is a state of $\hat{\mathcal{M}}$ then q defines a partial function $f_q : \Sigma^* \to (\Theta \cup \{\square\})^*$ where $f_q(x)$ is defined only when x is applicable to q with $x \in \Sigma^*$.

Put $\Theta_1 = \Theta \cup \{\square\}$. Let $x, y \in \Theta_1^*$, say

$$x = \alpha_1\alpha_2 \ldots \alpha_k,\ y = \alpha'_1\alpha'_2 \ldots \alpha'_k \quad \text{where } \alpha_i, \alpha'_i \in \Theta_1.$$

We say that x *covers* y if, for each $1 \le i \le k$ we have either $\alpha_i = \alpha'_i$ or $\alpha'_i = \square$. This is written as $x \# y$. We say that x and y are *compatible*, written $x \| y$ if, for each $1 \le i \le k$, we have either $\alpha_i = \alpha'_i$ or $\alpha_i = \square$ or $\alpha'_i = \square$.

Clearly $x \# y$ implies $x \| y$.

Thus for $\Theta = \{0, 1\}$ we have $0\square110 \# \square\square1\square0$ and $0\square110 \| \square\square1\square0$, also $\square\square1\square0 \| 0110\square$. Notice however that $0\square110 \| 0110\square$ is false and so compatibility is not a transitive relation. The relation 'covers' is not even symmetric.

Turning from compatibility amongst words to a related concept for states we proceed to the following definition. Let $q, q_1 \in Q$, we say that q and q_1 are *compatible* (or output compatible) if

$$f_q(x) \| f_{q_1}(x) \quad \text{for all } x \in \Sigma^* \text{ and } x \text{ applicable to } q \text{ and } q_1,$$

and write $q \| q_1$. If two states are compatible and the machine is started in either of these states then the output words will not be 'noticeably different', they may not be identical but where they do differ one word will have a blank space at that position.

One basic aim is to construct a Mealy machine with a state set of minimal size that will behave in the same way as a given Mealy machine. This involves looking at the partial functions f_q for each state q in the original machine.

6.2 Minimizing Mealy machines

We first consider a complete Mealy machine $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$. Define a relation $\sim$ on Q by

$$q \sim q_1 \Leftrightarrow f_q = f_{q_1}, \quad \text{where } q, q_1 \in Q.$$

A machine $\hat{\mathcal{M}}/\!\sim\ = (Q/\!\sim, \Sigma, \Theta, F', G')$ can now be constructed by defining

$$\left.\begin{aligned}[q]F'_\sigma &= [qF_\sigma]\\ [q]G'_\sigma &= qG_\sigma\end{aligned}\right\} \text{for } q \in Q, \sigma \in \Sigma,$$

where $[q]$ denotes the $\sim$-class containing q. This definition is meaningful since $\sim$ is an equivalence relation and if $q \sim q_1$ and $\sigma \in \Sigma$, $x \in \Sigma^*$ then

$$f_q(\sigma x) = f_q(\sigma) f_{qF_\sigma}(x) \quad \text{by 6.1.1}$$

and

$$f_{q_1}(\sigma x) = f_{q_1}(\sigma) \cdot f_{q_1F_\sigma}(x) = f_q(\sigma) \cdot f_{q_1F_\sigma}(x)$$

so that

$$f_{qF_\sigma} = f_{q_1F_\sigma} \quad \text{and thus } qF_\sigma \sim q_1F_\sigma.$$

Furthermore

$$qG_\sigma = f_q(\sigma) = f_{q_1}(\sigma) = q_1G_\sigma \quad \text{for } \sigma \in \Sigma.$$

We will see that $f'_{[q]}: \Sigma^* \to \Theta^*$, the function defined by $\hat{\mathcal{M}}/\!\sim$ in state $[q]$ equals $f_q: \Sigma^* \to \Theta^*$.

We call $\hat{\mathcal{M}}/\!\sim$ the *minimal Mealy machine* of $\hat{\mathcal{M}}$. The reason for this name is to be found in 6.2.2.

Theorem 6.2.1

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a complete Mealy machine. If $\hat{\mathcal{M}}/\!\sim$ is the minimal Mealy machine of $\hat{\mathcal{M}}$ then

(i) the surjective function $\psi: Q \to Q/\!\sim$ defined by

$\psi(q) = [q]$ satisfies the conditions

$$\psi(q)F'_\sigma = \psi(qF_\sigma)$$

$$\psi(q)G'_\sigma = qG_\sigma \quad \text{for } q \in Q, \sigma \in \Sigma$$

(ii) for each $q \in Q$, $f'_{[q]} = f_q$.

Proof (i) For $q \in Q$, $\sigma \in \Sigma$ the definition of $\hat{\mathcal{M}}/\!\sim$ yields

$$\psi(q)F'_\sigma = [q]F'_\sigma = [qF_\sigma] = \psi(qF_\sigma)$$

and

$$\psi(q)G'_\sigma = [q]G'_\sigma = qG_\sigma.$$

(ii) Let $\sigma \in \Sigma$, then

$$f'_{[q]}(\sigma) = [q]G'_\sigma = qG_\sigma = f_q(\sigma).$$

Assume that for words $x \in \Sigma^*$ of length less than n we have $f'_{[q]}(x) = f_q(x)$ and let $y \in \Sigma^*$ be of length n, so that $y = x\sigma$ for some $x \in \Sigma^*$ and $\sigma \in \Sigma$, $q \in Q$, then $f'_{[q]}(y) = f'_{[q]}(x)[q]F'_xG'_\sigma = f_q(x)\psi(qF_x)G'_\sigma = f_q(x)qF_xG_\sigma = f_q(x\sigma) = f_q(y)$. Hence the result follows by induction.

(We note that $\psi(q)F'_\sigma = \psi(qF_\sigma)$ can easily be extended by induction to $\psi(q)F'_x = \psi(qF_x)$ where $x \in \Sigma^*$.) $\square$

Corollary 6.2.2

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a complete Mealy machine and $\hat{\mathcal{M}}/\!\sim\ = (Q/\!\sim, \Sigma, \Theta, F', G')$ the minimal Mealy machine of $\hat{\mathcal{M}}$. Suppose that $\hat{\mathcal{M}}_1 = (Q_1, \Sigma, \Theta, F_1, G_1)$ is a complete Mealy machine and $\phi: Q \to Q_1$ is a surjective function satisfying

(i) $\phi(q)(F_1)_\sigma = \phi(qF_\sigma)$ for all $q \in Q, \sigma \in \Sigma$

(ii) $(f_1)_{\phi(q)} = f_q$ for all $q \in Q$

then a surjective function $\xi : Q_1 \to Q/\sim$ exists such that

(i)′ $\xi(q_1)F'_\sigma = \xi(q_1(F_1)_\sigma)$ for all $q_1 \in Q_1, \sigma \in \Sigma$
(ii)′ $f'_{\xi(q_1)} = (f_1)_{q_1}$ for all $q_1 \in Q_1$.

Proof Exercise 6.2. □

The method of actually calculating the minimal Mealy machine depends on finding the relation $\sim$. This can be done by a series of approximations to the relation. For each positive integer i define a relation $\sim_i$ on Q by

$$q \sim_i q' \Leftrightarrow f_q(x) = f_{q'}(x) \quad \text{for all } x \in \Sigma^+ \text{ of length less than or equal to } i.$$

Clearly $q \sim q' \Leftrightarrow q \sim_i q'$ for all $i > 0$.

Proposition 6.2.3
For $i > 1$, $q \sim_i q'$ if and only if

$$q \sim_1 q' \text{ and } qF_\sigma \sim_{i-1} q'F_\sigma \quad \text{for all } \sigma \in \Sigma.$$

Proof Suppose $q \sim_1 q'$ and $qF_\sigma \sim_{i-1} q'F_\sigma$ and let $x \in \Sigma^+$ be of length i. Then $x = \sigma y$ for some $y \in \Sigma^+$ of length $i-1$ and $\sigma \in \Sigma$. Now

$$\begin{aligned} f_q(x) &= f_q(\sigma y) \\ &= f_q(\sigma) \cdot f_{qF_\sigma}(y) \\ &= f_{q'}(\sigma) \cdot f_{q'F_\sigma}(y) \\ &= f_{q'}(\sigma y) \\ &= f_{q'}(x). \end{aligned}$$

The converse is now obvious. □

Each equivalence relation $\sim_i$ defines a partition π_i of the set Q and it is clear that

$$\pi_1 \supseteq \pi_2 \supseteq \ldots$$

Suppose that H_1 is a π_1-block and $q, q' \in H_1$; if $\sigma \in \Sigma$ is such that qF_σ and $q'F_\sigma$ belong to different π_1-blocks then proposition 6.2.3 tells us that $q \sim_2 q'$ cannot hold. More generally if q and q' belong to the same π_i-block but qF_σ and $q'F_\sigma$ belong to different π_i-blocks then q and q' cannot belong to the same π_{i+1}-block. In the language of state machines this means that if π_i is an admissible partition then $\pi_{i+1} = \pi_i$.

Proposition 6.2.4
For $i \geq 1$, $\pi_{i+1} = \pi_i$ if and only if π_i is an admissible partition on $\mathcal{M} = (Q, \Sigma, F)$.

Proof Suppose that $\pi_{i+1} = \pi_i$ and $q, q' \in Q$ are such that $q \sim_i q'$. Then for $\sigma \in \Sigma$, $qF_\sigma \sim_i q'F_\sigma$ since $q \sim_{i+1} q'$.
Conversely suppose that π_i is admissible and let $q, q' \in Q$ with $q \sim_i q'$ but $q \not\sim_{i+1} q'$. By 6.2.3 either $q \not\sim_1 q'$ or some $\sigma \in \Sigma$ exists such that $qF_\sigma \not\sim_i q'F_\sigma$, but this is impossible. □

We are now in a position to calculate the minimal Mealy machine since we can easily establish that $\pi_i = \pi_{i+1} \Rightarrow \pi_i = \pi_{i+k}$ for $k \geq 0$ and hence the relations $\sim_i$ and $\sim$ coincide.

Example 6.4
Let $\Sigma = \Theta = \{0, 1\}$ and $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be given by

	$\hat{\mathcal{M}}$	q_1	q_2	q_3	q_4	q_5
F	0	q_2	q_4	q_2	q_1	q_5
	1	q_5	q_5	q_5	q_3	q_4
G	0	0	1	0	0	1
	1	1	1	1	1	1

Then

$$\pi_1 = \{\{q_1, q_3, q_4\}, \{q_2, q_5\}\},$$
$$\pi_2 = \{\{q_1, q_3\}, \{q_4\}, \{q_2\}, \{q_5\}\}$$

which is admissible and hence $\pi_2 = \pi_3$ etc. The minimal machine is thus:

	$\hat{\mathcal{M}}/\sim$	$[q_1]$	$[q_2]$	$[q_4]$	$[q_5]$
F'	0	$[q_2]$	$[q_4]$	$[q_1]$	$[q_5]$
	1	$[q_2]$	$[q_2]$	$[q_4]$	$[q_4]$
G'	0	0	1	0	1
	1	1	1	1	1

Turning to the incomplete case we will first examine the problem of minimizing a state complete Mealy machine. If $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ is state complete, the relation $\|$ on Q may not be an equivalence relation,

since transitivity may fail. We can, however, still define a sequence of relations on Q as follows:
for each positive integer i and $q, q' \in Q$ define

$$q \parallel_i q' \Leftrightarrow f_q(x) \parallel f_{q'}(x)$$

for all $x \in \Sigma^*$ of length less than or equal to i. Then

$$q \parallel q' \Leftrightarrow q \parallel_i q' \quad \text{for all } i > 0.$$

Proposition 6.2.5
For $i > 1$, $q \parallel_i q'$ if and only if $q \parallel_1 q'$ and $qF_\sigma \parallel_{i-1} q'F_\sigma$ for all $\sigma \in \Sigma$.

Proof Suppose that $q \parallel_1 q'$ and $qF_\sigma \parallel_{i-1} q'F_\sigma$ and let $x \in \Sigma^*$ be of length i. Then $x = \sigma y$ for some $y \in \Sigma^*$ of length $i - 1$ and $\sigma \in \Sigma$. Now

$$f_q(x) = f_q(\sigma y) = f_q(\sigma) f_{qF_\sigma}(y)$$
$$f_{q'}(x) = f_{q'}(\sigma y) = f_{q'}(\sigma) f_{q'F_\sigma}(y).$$

We have

$$f_q(\sigma) \parallel f_{q'}(\sigma) \text{ and } f_{qF_\sigma}(y) \parallel f_{q'F_\sigma}(y)$$

and clearly this means $f_q(x) \parallel f_{q'}(x)$, that is $q \parallel_i q'$. The converse is easily checked. □

For each relation $\parallel_{i+1}$ on Q, we examine the relation $\parallel_i$ on Q and see what the state maps F_σ $(\sigma \in \Sigma)$ do to the pairs of states (q, q') satisfying $q \parallel_i q'$. If $qF_\sigma \parallel_i q'F_\sigma$ is false for some $\sigma \in \Sigma$ then $q \parallel_{i+1} q'$ is false. As before we eventually must reach a position where the relations $\parallel_n$ and $\parallel_{n+1}$ are identical. Then $\parallel_n$ equals the relation $\parallel$.

For each $q \in Q$, define

$$A(q) = \{q' \mid q \parallel q'\}.$$

Clearly $q \in A(q)$. The collection $\mathscr{K}$ of *distinct* $A(q)$ $(q \in Q)$ forms a set of subsets of Q but not generally a partition, i.e. we could have $A(q) \cap A(q') \neq \varnothing$ and $A(q) \neq A(q')$, we could also have $A(q) \subsetneqq A(q')$, $q, q' \in Q$. It is clear that if $q \parallel q'$ then $qF_\sigma \parallel q'F_\sigma$ for $\sigma \in \Sigma$, and so for all $q' \in A(q)$ we have $q'F_\sigma \in A(qF_\sigma)$ and thus $A(q)F_\sigma \subseteq A(qF_\sigma)$ for $q \in Q$, $\sigma \in \Sigma$.

The subsets $A(q) \in \mathscr{K}$ may have the following unfortunate property; namely that if $q', q'' \in A(q)$ then $q' \parallel q''$ is false. We now search for an *admissible* subset system $\pi = \{H_i\}_{i \in I}$ of Q satisfying the following conditions:

(i) Given $i \in I$, there exists $q \in Q$ such that

$$H_i \subseteq A(q).$$

(ii) If $q', q'' \in H_i$ then

$$q' \parallel q''.$$

It is always possible to find such an admissible subset system for any machine $\hat{\mathscr{M}}$, since 1_Q clearly satisfies the conditions. We call such an admissible subset system a *compatible subset system*. In general it may not be a partition of Q.

If $\pi = \{H_i\}_{i \in I}$ is a compatible subset system then a Mealy machine $\hat{\mathscr{M}}/\pi$ can be defined as follows:

$$\hat{\mathscr{M}}/\pi = (\{H_i\}_{i \in I}, \Sigma, \Theta, F^\pi, G^\pi)$$

where

$$H_iF^\pi_\sigma = H_j \quad \text{where } j \in I \text{ is chosen so that } H_iF_\sigma \subseteq H_j,$$

$$H_iG^\pi_\sigma = \begin{cases} qG_\sigma & \text{if a } q \in H_i \text{ exists such that } qG_\sigma \neq \varnothing \\ \varnothing & \text{otherwise.} \end{cases}$$

Since π is an admissible subset system, rather than a partition, in general there may be many possibilities for the definition of F^π and we will assume that a particular choice has been made (see chapter 4 for a similar definition) and then F^π is well-defined. Since π is compatible it is clear that G^π is also well-defined.

Now let $q \in Q$, there then exists an $i \in I$ such that $q \in H_i$; we now establish a connection between the sequential function f_q defined with respect to $\hat{\mathscr{M}}$ and the sequential function $f^\pi_{H_i}$ defined with respect to $\hat{\mathscr{M}}/\pi$.

Theorem 6.2.6
Let $\hat{\mathscr{M}} = (Q, \Sigma, \Theta, F, G)$ be a state complete Mealy machine and $\pi = \{H_i\}_{i \in I}$ a compatible subset system on Q. Let $q \in Q$ then $q \in H_i$ for some $i \in I$ and if $f_{H_i} : \Sigma^* \rightarrow (\Theta \cup \square)^*$ is the sequential function of $\hat{\mathscr{M}}/\pi$ in state H_i then for each $x \in \Sigma^*$,

$$f^\pi_{H_i}(x) \# f_q(x).$$

Proof Let $\sigma \in \Sigma$, then $f_q(\sigma) = qG_\sigma$. If $q'G_\sigma = \varnothing$ for all $q' \in H_i$ then $qG_\sigma = H_iG^\pi_\sigma = \varnothing$. If $q'G_\sigma \neq \varnothing$ for some $q' \in H_i$ then $H_iG^\pi_\sigma = q'G_\sigma$. Since $q \parallel q'$ either $qG_\sigma = q'G_\sigma$ or $qG_\sigma = \varnothing$. In all cases $H_iG^\pi_\sigma \# qG_\sigma$ so $f^\pi_{H_i}(\sigma) \# f_q(\sigma)$. Now suppose that $f^\pi_{H_i}(x) \# f_q(x)$ for all $x \in \Sigma^*$ of length less than n and let $y \in \Sigma^*$ be of length n. Writing $y = x\sigma$ for $x \in \Sigma^*$, $\sigma \in \Sigma$ we see that

$$f_q(y) = f_q(x) \cdot qF_xG_\sigma$$

and

$$f^\pi_{H_i}(y) = f^\pi_{H_i}(x) \cdot H_iF^\pi_xG^\pi_\sigma.$$

By the inductive assumption $f^{\pi}_{H_l}(x) \# f_q(x)$. Let $H_iF^{\pi}_x = H_l$, where $H_iF_x \subseteq H_l$, then $qF_x \in H_l$. Now $H_lG^{\pi}_{\sigma} = q'G_{\sigma}$ where $q' \in H_l$ and since $q' \| qF_x$ we see that $H_lG^{\pi}_{\sigma} \# qF_xG_{\sigma}$ and so $f^{\pi}_{H_i}(y) \# f_q(y)$. The result follows by induction. □

In many ways the Mealy machine $\hat{\mathcal{M}}/\pi$ performs similar tasks to the original machine $\hat{\mathcal{M}}$, but it may not be the smallest such machine. The size of $\hat{\mathcal{M}}/\pi$ equals the number of subsets in the compatible subset system $\pi = \{H_i\}_{i \in I}$ and we would naturally ask for this to be as small as possible. A compatible subset system π is called *maximal* if no non-trivial compatible subset system τ exists such that $\pi < \tau$. We regard $\{Q\}$ as a trivial compatible subset system.

The Mealy machine $\hat{\mathcal{M}}/\pi$, where π is a maximal compatible subset system, will be called a *minimal cover for* $\hat{\mathcal{M}}$. There is no unique minimal cover in general for a Mealy machine $\hat{\mathcal{M}}$, and in fact different minimal covers for a particular machine $\hat{\mathcal{M}}$ can have rather different properties. The task of constructing the minimal covers will not be discussed in any detail here; it amounts to the calculation of the maximal compatible subset systems and this in general is done using *ad hoc* methods.

Example 6.5
Consider the machine $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$, where $\Sigma = \Theta = \{0, 1\}$, given by:

$\hat{\mathcal{M}}$		q_1	q_2	q_3	q_4	q_5
F	0	q_2	q_1	q_2	q_1	q_5
	1	q_5	q_5	q_5	q_3	q_4
G	0	0	∅	∅	0	1
	1	1	1	∅	1	∅

To calculate the relation $\|$ on Q we proceed as follows. First we describe the relation $\|_1$ by writing $(i, j)_1$ to denote $q_i \|_1 q_j$ and recall that the relation is symmetric. Thus

$$(1, 2)_1, (1, 3)_1, (1, 4)_1, (2, 3)_1, (2, 4)_1, (2, 5)_1, (3, 4)_1, (3, 5)_1.$$

To determine the relation $\|_2$ we examine $(q_iF_{\sigma}, q_jF_{\sigma})$ for each pair $(i, j) \in \|_1$ and if $q_iF_{\sigma} = q_k, q_jF_{\sigma} = q_l$ and $(k, l) \notin \|_1$ then by 6.2.4 we know that $(i, j) \notin \|_2$. This leads to

$$(1, 2)_2, (1, 3)_2, (1, 4)_2, (2, 3)_2, (2, 4)_2, (3, 4)_2$$

and then

$$(1, 2)_3, (1, 3)_3, (2, 3)_3$$

and

$$(1, 2)_4, (1, 3)_4, (2, 3)_4.$$

Therefore $\|$ is the same as $\|_3$ and the set $\mathscr{X} = \{\{q_1, q_2, q_3\}, \{q_4\}, \{q_5\}\}$ which is, in this case, a compatible subset system, and a partition. A minimal cover is thus given by:

$\hat{\mathcal{M}}/\pi$		H_1	H_2	H_3
F^{π}	0	H_1	H_1	H_3
	1	H_3	H_1	H_2
G^{π}	0	0	0	1
	1	1	1	∅

where $\pi = \{H_1, H_2, H_3\}$ and $H_1 = \{q_1, q_2, q_3\}$, $H_2 = \{q_4\}$, $H_3 = \{q_5\}$.

Example 6.6
Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be defined by

$\hat{\mathcal{M}}$		q_1	q_2	q_3	q_4	q_5
F	0	q_1	q_1	q_5	q_3	q_1
	1	q_3	q_1	q_1	q_5	q_3
G	0	0	∅	0	0	0
	1	0	1	∅	1	∅

where $\Sigma = \Theta = \{0, 1\}$. Then $\|_1$ is given by

$$(1, 3)_1, (1, 5)_1, (2, 3)_1, (2, 4)_1, (2, 5)_1, (3, 4)_1, (3, 5)_1, (4, 5)_1$$

and we also obtain

$$(1, 3)_2, (1, 5)_2, (2, 3)_2, (2, 4)_2, (2, 5)_2, (3, 4)_2, (3, 5)_2, (4, 5)_2.$$

Thus

$$\mathscr{X} = \{\{q_1, q_3, q_5\}, \{q_2, q_3, q_4, q_5\}, Q\}.$$

If $H_1 = \{q_1, q_3, q_5\}$ and $H_2 = \{q_2, q_4\}$ then $\pi = \{H_1, H_2\}$ is an admissible subset system which is also compatible. Then $\hat{\mathcal{M}}/\pi$ is given by

$\hat{\mathcal{M}}/\pi$		H_1	H_2
F^π	0	H_1	H_1
	1	H_1	H_1
G^π	0	0	0
	1	0	1

and this is a minimal cover for $\hat{\mathcal{M}}$. Here again π was a partition of Q even though $\mathscr{X}$ was not.

Example 6.7

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be given by

$\hat{\mathcal{M}}$		q_1	q_2	q_3	q_4	q_5
F	a	q_1	q_3	q_2	q_5	q_4
	b	q_1	q_1	q_3	q_4	q_4
	c	q_1	q_3	q_2	q_5	q_4
G	a	0	1	$\varnothing$	0	$\varnothing$
	b	$\varnothing$	0	1	1	0
	c	1	$\varnothing$	0	$\varnothing$	1

where $\Sigma = \{a, b, c\}$ and $\Theta = \{0, 1\}$. Then $\|_1$ is given by $(1, 4)_1$, $(1, 5)_1$, $(2, 5)_1$, $(3, 4)_1$ which is also $\|$.

$$\mathscr{X} = \{\{q_1, q_4, q_5\}, \{q_2, q_5\}, \{q_3, q_4\}, \{q_1, q_3, q_4\}, \{q_1, q_2, q_5\}\}.$$

Let $H_1 = \{q_1, q_4\}$, $H_2 = \{q_2, q_5\}$, $H_3 = \{q_3, q_4\}$, $H_4 = \{q_1, q_5\}$, then $\pi = \{H_1, H_2, H_3, H_4\}$ is an admissible and compatible subset system which is not a partition. A machine, $\hat{\mathcal{M}}/\pi$, which is a minimal cover for $\hat{\mathcal{M}}$ is given by

$\hat{\mathcal{M}}/\pi$		H_1	H_2	H_3	H_4
F^π	a	H_2	H_3	H_2	H_1
	b	H_1	H_1	H_3	H_1
	c	H_2	H_3	H_2	H_1
G^π	a	0	1	$\varnothing$	0
	b	$\varnothing$	0	1	0
	c	1	$\varnothing$	0	1

More general covers can be introduced as follows. First let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a Mealy machine. If $\hat{\mathcal{M}}' = (Q', \Sigma, \Theta, F', G')$ is another state complete Mealy machine and $\phi : Q \to Q'$ is a function then we say that ϕ is a *covering* of $\hat{\mathcal{M}}$ by $\hat{\mathcal{M}}'$ if, for each $q \in Q$,

$$f'_{\phi(q)}(x) \# f_q(x) \quad \text{for all } x \in \Sigma^*,$$

where f_q and $f'_{\phi(q)}$ are the partial functions associated with $\hat{\mathcal{M}}$ in state q and $\hat{\mathcal{M}}'$ in state $\phi(q)$ respectively. We write

$$\hat{\mathcal{M}} \le \hat{\mathcal{M}}'.$$

This means that machine $\hat{\mathcal{M}}'$ will do all that $\hat{\mathcal{M}}$ can do, and possibly more. In the case where $\hat{\mathcal{M}}$ is state complete and $\hat{\mathcal{M}}' = \hat{\mathcal{M}}/\pi$ for some compatible subset system π then

$$\hat{\mathcal{M}} \le \hat{\mathcal{M}}/\pi.$$

It is now necessary to extend our concepts of compatibility to Mealy machines that may not be state complete.

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a general Mealy machine and let $q, q_1 \in Q$. We say that q and q_1 are *compatible* if, whenever $x \in \Sigma^*$ is applicable to both q and q_1, then

$$f_q(x) \| f_{q_1}(x).$$

As before we may define the relations $\|_i$ on Q for each positive integer i. The subsets

$$A(q) = \{q' \mid q \| q'\}$$

may be formed for each $q \in Q$ and also the collection $\mathscr{X}$ of the distinct $A(q)$. Using our new compatibility definition we can now look for admissible subset systems $\pi = \{H_i\}_{i \in I}$ of Q satisfying

(i) for each $i \in I$ there exists a $q \in Q$ such that $H_i \subseteq A(q)$,

(ii) if $q', q'' \in H_i$ then $q' \| q''$.

We call π a *compatible subset system* as before. Define a Mealy machine $\hat{\mathcal{M}}/\pi = (\{H_i\}_{i \in I}, \Sigma, \Theta, F^\pi, G^\pi)$ as follows:

$$H_iF^\pi_\sigma = \begin{cases} H_j & \text{if } \exists\, j \in I \text{ such that } \varnothing \ne H_iF_\sigma \subseteq H_j \\ \varnothing & \text{otherwise} \end{cases}$$

$$H_iG^\pi_\sigma = \begin{cases} qG_\sigma & \text{if a } q \in H_i \text{ exists satisfying } qG_\sigma \ne \varnothing \\ \varnothing & \text{otherwise.} \end{cases}$$

As before we make a choice for the definition of F^π. The compatibility of π ensures that G^π is well-defined.

Theorem 6.2.7

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a Mealy machine which is not state complete and suppose that $\hat{\mathcal{M}}^c = (Q \cup \{z\}, \Sigma, \Theta, F', G')$ is the state completion of $\hat{\mathcal{M}}$. If $\pi = \{H_i\}_{i\in I}$ is a compatible subset system for $\hat{\mathcal{M}}$ then

$$\pi^c = \{H_i \cup \{z\}\}_{i\in I}$$

is a compatible subset system for $\hat{\mathcal{M}}^c$. Conversely, let $\tau = \{K_j\}_{j\in J}$ be a compatible subset system for $\hat{\mathcal{M}}^c$ then

$$\tau^* = \{K_j\backslash\{z\}\}_{j\in J}$$

is a compatible subset system for $\hat{\mathcal{M}}$.

Proof Let $\pi = \{H_i\}_{i\in I}$ be a compatible subset system for $\hat{\mathcal{M}}$. Clearly $\pi^c = \{H_i \cup \{z\}\}_{i\in I}$ is an admissible subset system for $\hat{\mathcal{M}}^c$ for if $\sigma \in \Sigma$ then

$$H_iF_\sigma \subseteq H_j \quad \text{for some } j \in I$$

and

$$(H_i \cup \{z\})F'_\sigma \subseteq H_j \cup \{zF_\sigma\} = H_j \cup \{z\} \in \pi^c.$$

Now let $H_i \in A(q)$. Since $z \| q$ in $\hat{\mathcal{M}}^c$ we see that $H_i \cup \{z\} \subseteq A^c(q)$ where $A^c(q)$ denotes the set of states of $\hat{\mathcal{M}}^c$ compatible with q.

Finally for each $q', q'' \in H_i$ we have $q' \| q''$ in $\hat{\mathcal{M}}$. Clearly $q' \| z$ and $q'' \| z$ in $\hat{\mathcal{M}}^c$ and so π^c is a compatible subset system.

Now suppose that $\tau = \{K_j\}_{j\in J}$ is a compatible subset system for $\hat{\mathcal{M}}^c$. First note that the non-empty subsets of the form $K_j\backslash\{z\}$ $(j \in J)$ form an admissible subset system for $\hat{\mathcal{M}}$, since

$$(K_j\backslash\{z\})F_\sigma = K_jF_\sigma\backslash\{z\} \subseteq K_l\backslash\{z\}$$

for some $l \in J$, where $\sigma \in \Sigma$.

Now let $K_j \subseteq A^c(q)$ for some $q \in Q \cup \{z\}$. We may assume that $q \neq z$ since $A^c(z) = Q \cup \{z\}$. Let $q' \in K_j\backslash\{z\}$, then $q' \| q$ in $\hat{\mathcal{M}}^c$, where $q \in Q$. Suppose that $x \in \Sigma^*$ is such that x.is applicable to both q' and q in $\hat{\mathcal{M}}$, then $f_{q'}(x)$ and $f_q(x)$ exist (in $\hat{\mathcal{M}}$) and since $f_{q'}(x) \| f_q(x)$ in $\hat{\mathcal{M}}^c$ we have $f_{q'}(x) \| f_q(x)$ in $\hat{\mathcal{M}}$. Hence $q' \| q$ in $\hat{\mathcal{M}}$.

Finally let $q', q'' \in K_j\backslash\{z\}$. By a similar argument we see that $q' \| q''$ in $\hat{\mathcal{M}}^c$ and thus $q' \| q''$ in $\hat{\mathcal{M}}$. Therefore τ^* is a compatible subset system for $\hat{\mathcal{M}}$. □

As before a compatible subset system π is called *maximal* if no non-trivial compatible subset system τ exists such that $\pi < \tau$.

Theorem 6.2.8

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a Mealy machine which is not state complete and $\hat{\mathcal{M}}^c = (Q \cup \{z\}, \Sigma, \Theta, F', G')$ its state completion. If $\pi = \{H_i\}_{i\in I}$ is a maximal compatible subset system for $\hat{\mathcal{M}}$ then $\pi^c = \{H_i \cup \{z\}\}_{i\in I}$ is a maximal compatible subset system for $\hat{\mathcal{M}}^c$.

Conversely let $\tau = \{K_j\}_{j\in J}$ be a maximal compatible subset system for $\hat{\mathcal{M}}^c$, then

$$\tau^* = \{K_j\backslash\{z\}\}_{j\in J}$$

is a maximal compatible subset system for $\hat{\mathcal{M}}$.

Proof Assume first that $\pi = \{H_i\}_{i\in I}$ is maximal and let $\pi^c < \tau$ where τ is a compatible subset system of $\hat{\mathcal{M}}^c$. Consider the subset system

$$\tau^* = \{K_j\backslash\{z\}\}_{j\in J}$$

where the system $\tau = \{K_j\}_{j\in J}$. From the previous result we see that τ^* is a compatible subset system for $\hat{\mathcal{M}}$ and clearly $\pi \leq \tau^*$, if $\pi = \tau^*$ then we must have $\pi^c = (\tau^*)^c$, but $\pi^c < \tau$ implies that $z \in K_j$ for all $j \in J$ and so $(\tau^*)^c = \tau$. Thus we obtain a contradiction and so $\pi < \tau^*$. This means that $\tau^* = \{Q\}$ since π is maximal. Then $\tau = (\tau^*)^c = \{Q \cup \{z\}\}$ and so π^c is maximal in $\hat{\mathcal{M}}^c$.

Now let $\tau = \{K_j\}_{j\in J}$ be maximal in $\hat{\mathcal{M}}^c$ and suppose that $\tau^* < \rho$ where ρ is a compatible subset system for $\hat{\mathcal{M}}$. Then $(\tau^*)^c \leq \rho^c$ and clearly

$$\tau^* = ((\tau^*)^c)^* \leq (\rho^c)^* = \rho$$

which implies that $(\tau^*)^c < \rho^c$. If $\rho = \{L_t\}_{t\in T}$ then for each $j \in J$, $K_j\backslash\{z\} \subseteq L_t$ for some $t \in T$ and so $K_j \subseteq L_t \cup \{z\} \in \rho^c$ for each $j \in J$, even when $K_j = \{z\}$. Therefore $\tau < \rho^c$ and the maximality of τ forces $\rho^c = \{Q \cup \{z\}\}$ and thus $\rho = \{Q\}$. □

These two results enable us to obtain minimal covering machines for incomplete Mealy machines directly from the covering machines of their state completions.

Let $\hat{\mathcal{M}}$ be a Mealy machine. If $\hat{\mathcal{M}}$ is not state complete, consider the state completion $\hat{\mathcal{M}}^c$ and construct a maximal compatible subset system π for $\hat{\mathcal{M}}^c$. Then the compatible subset system π^* for $\hat{\mathcal{M}}$ is maximal and any Mealy machine of the form $\hat{\mathcal{M}}/\pi^*$ will be a minimal cover for $\hat{\mathcal{M}}$.

The justification for this terminology is obtained if we generalize our notion of Mealy machine covering to include incomplete Mealy machines.

For any arbitrary Mealy machines $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ and $\hat{\mathcal{M}}' = (Q', \Sigma, \Theta, F', G')$ we say that $\hat{\mathcal{M}}'$ *covers* $\hat{\mathcal{M}}$, written $\hat{\mathcal{M}}' \geq \hat{\mathcal{M}}$, if there exists a function $\phi : Q \to Q'$ such that for each $q \in Q$

$$f'_{\phi(q)}(x) \# f_q(x)$$

for all $x \in \Sigma^*$ applicable to q.

Theorem 6.2.9

Let $\hat{\mathcal{M}}$ be a Mealy machine and π a compatible subset system for $\hat{\mathcal{M}}$. Then

$$\hat{\mathcal{M}} \leq \hat{\mathcal{M}}/\pi.$$

Proof Let $x \in \Sigma^*$ be applicable to the state $q \in Q$, then, if $x = \sigma_1 \ldots \sigma_k$, all of $qF_{\sigma_1}, \ldots, qF_{\sigma_1 \ldots \sigma_{k-1}}$ are defined. If $q \in H_i$ for $i \in I$ we have

$$qF_{\sigma_1} \in H_iF_{\sigma_1}, \ldots, qF_{\sigma_1 \ldots \sigma_{k-1}} \in H_iF_{\sigma_1 \ldots \sigma_{k-1}}$$

and so x is applicable in $\hat{\mathcal{M}}/\pi$ to H_i.

Putting $\phi : Q \to \{H_i\}_{i \in I}$ to be any function satisfying $q \in \phi(q)$, $q \in Q$ we see that a similar proof to 6.2.6 will yield

$$f^{\pi}_{\phi(q)}(x) \# f_q(x). \qquad \square$$

Example 6.8

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be given by

	$\hat{\mathcal{M}}$	q_1	q_2	q_3	q_4	q_5
F	a	q_1	∅	q_2	q_5	q_4
	b	q_1	∅	q_3	q_4	∅
	c	q_1	q_3	q_2	q_5	q_4
G	a	0	1	∅	0	∅
	b	∅	0	1	1	0
	c	1	∅	0	∅	1

where $\Sigma = \{a, b, c\}$, $\Theta = \{0, 1\}$. (This is an incomplete version of example 6.7.)

Then $\hat{\mathcal{M}}^c$ is given by

	$\hat{\mathcal{M}}^c$	q_1	q_2	q_3	q_4	q_5	z
F	a	q_1	z	q_2	q_5	q_4	z
	b	q_1	z	q_3	q_4	z	z
	c	q_1	q_3	q_2	q_5	q_4	z
G	a	0	1	∅	0	∅	∅
	b	∅	0	1	1	0	∅
	c	1	∅	0	∅	1	∅

We calculate the relation $\|$ for $\hat{\mathcal{M}}^c$.

Now $\|_1$ is given by

$$(1, 4)_1, (1, 5)_1, (2, 5)_1, (3, 4)_1, (1, z)_1, (2, z)_1, (3, z)_1, (4, z)_1, (5, z)_1.$$

The relation $\|$ is given by

$$(1, 4)_2, (1, 5)_2, (1, z)_2, (2, 5)_2, (2, z)_2, (3, 4)_2, (3, z)_2, (4, z)_2, (5, z)_2.$$

So

$$\mathscr{X} = \{\{q_1, q_4, z\}, \{q_2, q_5, z\}, \{q_3, q_4, z\}, \{q_1, q_3, q_4, z\},$$
$$\{q_1, q_2, q_5, z\}, \{q_1, q_2, q_3, q_4, q_5, z\}\}.$$

Let $H_1 = \{q_1, q_4, z\}$, $H_2 = \{q_2, q_5, z\}$, $H_3 = \{q_3, q_4, z\}$, $H_4 = \{q_1, q_5, z\}$, then $\pi = \{H_1, H_2, H_3, H_4\}$ is a compatible subset system for $\hat{\mathcal{M}}^c$. It is a maximal compatible subset system for $\hat{\mathcal{M}}^c$ and so $\pi^* = \{H_1\backslash\{z\}, H_2\backslash\{z\}, H_3\backslash\{z\}, H_4\backslash\{z\}\}$ is a maximal compatible subset system for $\hat{\mathcal{M}}$.

Now $\hat{\mathcal{M}}/\pi^*$ could take the form, for example

	$\hat{\mathcal{M}}/\pi^*$	H_1^*	H_2^*	H_3^*	H_4^*
F^*	a	H_2^*	H_3^*	H_2^*	H_1^*
	b	H_1^*	∅	H_3^*	H_1^*
	c	H_2^*	H_3^*	H_2^*	H_1^*
G^*	a	0	1	∅	0
	b	∅	0	1	0
	c	1	∅	0	1

where $H_1^* = \{q_1, q_4\}$, $H_2^* = \{q_2, q_5\}$, $H_3^* = \{q_3, q_4\}$, $H_4^* = \{q_1, q_5\}$.

This is 'almost isomorphic' to the Mealy machine $\hat{\mathcal{M}}/\pi$ obtained in example 6.7, but this should come as no surprise since the machine $\hat{\mathcal{M}}$ in 6.7 clearly covers the machine considered here and so we would expect some close connection between their minimal covers.

We close with the remark that our approach to the minimization of a Mealy machine actually makes use of the fact that the machine may not be completely defined. The entries $\varnothing$ in the tables specifying the machine's output are sometimes called 'don't care' entries since their value is of no consequence. We can take advantage of this freedom to generate much smaller covering machines than if we were to complete the output function in a similar way to the completion of the state function. For this reason we have chosen a rather general form of the concept of machine covering.

6.3 Two sorts of covering

The purpose of this section is to examine the relationship between the covering of one Mealy machine by another and the connections between their state machines. To examine this problem in general it is necessary to extend the definition of Mealy machine covering to include the case where the input and output alphabets do not coincide.

Let $\hat{\mathcal{M}}=(Q,\Sigma,\Theta,F,G)$ and $\hat{\mathcal{M}}'=(Q,'\Sigma',\Theta',F',G')$ be Mealy machines, not necessarily state complete.

Let $\xi:\Sigma\to\Sigma'$, $\rho:\Theta\to\Theta'$ be functions and suppose that a function $\psi:Q\to Q'$ exists such that for each $q\in Q$ we have

$$f'_{\psi(q)}(\xi(x))\,\#\,\rho(f_q(x))$$

for all $x\in\Sigma^*$ applicable to q and such that $\xi(x)$ is applicable to $\psi(q)$. (The functions ξ and ρ are of course assumed to have been extended to the free monoids Σ^* and Θ^* respectively.)

As usual we will write $\hat{\mathcal{M}}\leq\hat{\mathcal{M}}'$. If $\mathcal{M}=(Q,\Sigma,F)$ we will call $\mathcal{M}$ the state machine of $\hat{\mathcal{M}}$.

Theorem 6.3.1

Let $\hat{\mathcal{M}}=(Q,\Sigma,\Theta,F,G)$ be a Mealy machine and suppose that $\mathcal{M}'=(Q',\Sigma',F')$ is a state machine satisfying $\mathcal{M}\leq\mathcal{M}'$, then there exists a Mealy machine $\hat{\mathcal{M}}'=(Q',\Sigma',\Theta',F',G')$ such that for each $q\in Q$ and $x\in\Sigma^*$ applicable to q

$$f_q(x)=f'_{\psi(q)}(\xi(x)),\quad \text{for some function } \psi:Q\to Q'.$$

Proof We are given a function $\xi:\Sigma\to\Sigma'$ and a surjective partial function $\eta:Q'\to Q$ such that $\eta(q')F_x\subseteq\eta(q'F'_{\xi(x)})$ for each $q'\in Q'$ and $x\in\Sigma^*$.

Put $\Theta'=\Theta$ and define $G':Q'\times\Sigma'\to\Theta$ by

$$G'(q',\sigma')=\begin{cases}G(\eta(q'),\sigma) & \text{if } \sigma'=\xi(\sigma) \text{ for some } \sigma\in\Sigma \text{ and } \eta(q')\neq\varnothing,\\ \varnothing & \text{otherwise.}\end{cases}$$

Now let $\psi:Q\to Q'$ be a function satisfying the condition $\eta\circ\psi=1_Q$; such a function must exist since η is surjective. We show that ψ defines a covering of Mealy machines $\hat{\mathcal{M}}\leq\hat{\mathcal{M}}'$ where $\hat{\mathcal{M}}'=(Q',\Sigma',\Theta,F',G')$. Choose any $q\in Q$. Let $x\in\Sigma^*$ be applicable to q in $\hat{\mathcal{M}}$ and suppose that $x=\sigma_1\ldots\sigma_k$. Then

$$qF_{\sigma_1},\ldots,qF_{\sigma_1\ldots\sigma_{k-1}}$$

are all defined. Since $q=\eta(\psi(q))$ we see that

$$\begin{aligned}qF_{\sigma_1}&=\eta(\psi(q))F_{\sigma_1}\subseteq\eta(\psi(q)F'_{\xi(\sigma_1)})\\ &\vdots\\ qF_{\sigma_1\ldots\sigma_{k-1}}&=\eta(\psi(q))F_{\sigma_1\ldots\sigma_{k-1}}\subseteq\eta(\psi(q)F'_{\xi(\sigma_1\ldots\sigma_{k-1})})\end{aligned}$$

since η is a state machine covering. Thus $\xi(x)$ is applicable to $\psi(q)$ in $\hat{\mathcal{M}}'$. Now for $\sigma\in\Sigma$,

$$f_q(\sigma)=qG_\sigma=\eta(\psi(q))G_\sigma=\psi(q)G'_{\xi(\sigma)}=f'_{\psi(q)}(\xi(\sigma)).$$

Assume that $f_q(x)=f'_{\psi(q)}(\xi(x))$ for all words $x\in\Sigma^*$ of length less than n which are applicable to q.

Now let $y=x\sigma$ where y is of length n and y is applicable to q. Then

$$\begin{aligned}f_q(y)&=f_q(x)\cdot qF_xG_\sigma\\ &=f'_{\psi(q)}\xi(x)\cdot qF_xG_\sigma.\end{aligned}$$

Now

$$\begin{aligned}qF_xG_\sigma&=\eta(\psi(q))F_xG_\sigma\\ &\subseteq\eta(\psi(q)F'_{\xi(x)})G_\sigma\\ &=\psi(q)F'_{\xi(x)}G'_{\xi(\sigma)}\quad\text{by the definition of } G'.\end{aligned}$$

Since

$$qF_x\neq\varnothing$$

we have

$$qF_xF_\sigma=\psi(q)F'_{\xi(x)}G'_{\xi(\sigma)}$$

and so

$$f_q(y)=f'_{\psi(q)}(\xi(y)).\qquad\square$$

Corollary 6.3.2

In the situation of 6.3.1 we have

$$\hat{\mathcal{M}}\leq\hat{\mathcal{M}}'.$$

One conclusion that we may draw from this result is that whereas the concept of covering of Mealy machines developed in section 6.2 and above is suitable for the problem of minimizing incomplete Mealy machines, when we come to examine the relationship of Mealy machine covering with state machine covering it is too general. Our aim in this chapter is to apply the results of chapters 3 and 4 on state machines to the theory of Mealy machines, and to achieve this we will introduce a special form of Mealy machine covering more suitable for this task. When the machines are complete there is no difference in the two concepts.

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ and $\hat{\mathcal{M}}' = (Q', \Sigma', \Theta', F', G')$ be Mealy machines. Suppose that $\xi : \Sigma \to \Sigma'$, $\rho : \Theta \to \Theta'$ are functions, and a function $\psi : Q \to Q'$ exists such that for each $q \in Q$ and $x \in \Sigma^*$, x is applicable to q if and only if $\xi(x)$ is applicable to $\psi(q)$ and

$$\rho(f_q(x)) = f'_{\psi(q)}(\xi(x)).$$

We say that $\hat{\mathcal{M}}'$ *strongly covers* $\hat{\mathcal{M}}$, or that ψ is a strongly covering function, and write

$$\hat{\mathcal{M}} \ll \hat{\mathcal{M}}'.$$

Clearly $\hat{\mathcal{M}} \ll \hat{\mathcal{M}}'$ implies $\hat{\mathcal{M}} \leq \hat{\mathcal{M}}'$.

To make progress in the other direction we need the following concept. A Mealy machine $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ is called *reduced* if given distinct states q, q_1 then there exists $x \in \Sigma^*$ such that $f_q(x) \neq f_{q_1}(x)$, with x applicable to both q and q_1.

Theorem 6.3.3

(Ginzburg [1968]) Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a reduced Mealy machine and suppose that $\hat{\mathcal{M}} \ll \hat{\mathcal{M}}'$ where $\hat{\mathcal{M}}' = (Q', \Sigma, \Theta, F', G')$. Then $\mathcal{M} \leq \mathcal{M}'$ as state machines.

Proof Let $\psi : Q \to Q'$ be given such that, for $q \in Q$,

$$f_q(x) = f'_{\psi(q)}(x)$$

for all $x \in \Sigma^*$ applicable to q.

We first note that ψ is a one-one function, for if q, $q_1 \in Q$ and $\psi(q) = \psi(q_1)$ then $f'_{\psi(q)}(x) = f_q(x) = f_{q_1}(x)$ for all $x \in \Sigma^*$ applicable to q and q_1 and so $q = q_1$ since $\hat{\mathcal{M}}$ is reduced.

We wish to construct a surjective partial function $\eta : Q' \to Q$ such that

$$\eta(q')F_x \subseteq \eta(q'F'_x) \quad \text{for all } q' \in Q', x \in \Sigma^*.$$

First note that there exists a unique function $\chi : \psi(Q) \to Q$ defined by

$$\chi(\psi(q)) = q \quad \text{for all } \psi(q) \in \psi(Q).$$

This is well-defined since ψ is one-one. Thus χ defines a surjective partial function from Q' to Q, but it does not necessarily satisfy the requirement for it to be a state machine covering since

$$\psi(Q)F'_x \subseteq \psi(Q)$$

may not hold for all $x \in \Sigma^*$.

For $x \in \Sigma^*$ define a partial function

$$\psi_x : Q \to Q'$$

by

$$\psi_x(q) = (\psi(q))F'_x \quad \text{for } q \in Q.$$

Now choose a partial function $\alpha_x : Q' \to Q$ such that

$$\mathfrak{D}(\alpha_x) = \psi(Q)F'_x$$

and

$$\psi_x\alpha_x(q') = q' \quad \text{for all } q' \in \psi(Q)F'_x$$

thus

$$(\psi(\alpha_x(q'))F'_x = q' \quad \text{for all } q' \in \psi(Q)F'_x.$$

Define a partial function

$$\eta_x : Q' \to Q$$

by

$$\eta_x(q') = (\alpha_x(q'))F_x \quad \text{for all } q' \in \psi(Q)F'_x.$$

Notice that $\psi_\Lambda = \psi$ and $\eta_\Lambda(\psi(q)) = \alpha_\Lambda(\psi(q))$ for all $\psi(q) \in \psi(Q)$, and since ψ is one-one and $\psi\alpha_\Lambda(\psi(q)) = \psi(q)$ we have

$$\alpha_\Lambda(\psi(q)) = q.$$

Consider now the relation

$$\eta = \bigcup_{x \in \Sigma^*} \eta_x : Q' \to Q.$$

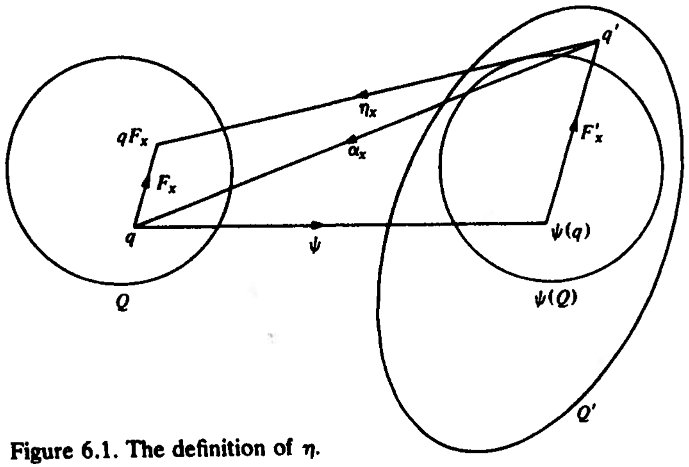


Figure 6.1. The definition of η.

Then $\mathfrak{D}(\eta)=\bigcup_{x\in\Sigma^*}\psi(Q)F'_x$. We establish that η is a partial surjective function. Since $\eta_\Lambda\subseteq\eta$ it is clear that η is surjective. See figure 6.1. We must now show that η is a partial function.

First let $q\in Q$, and $\alpha,\beta\in\Sigma^*$ with $\alpha\beta$ applicable to q. Then $f_q(\alpha\beta)=f'_{\psi(q)}(\alpha\beta)$, which by 6.1.1 gives

$$f_q(\alpha)\cdot f_{qF_\alpha}(\beta)=f'_{\psi(q)}(\alpha)\cdot f'_{\psi(q)F'_\alpha}(\beta)$$

and so

$$f_{qF_\alpha}(\beta)=f'_{\psi(q)F'_\alpha}(\beta)$$

since $f_q(\alpha)=f'_{\psi(q)}(\alpha)$.

Now

$$f_{qF_\alpha}(\beta)=f'_{\psi(qF_\alpha)}(\beta)$$

since β is applicable to qF_α and thus

$$f'_{\psi(qF_\alpha)}(\beta)=f'_{\psi(q)F'_\alpha}(\beta)=f_{qF_\alpha}(\beta).$$

Now let $z\in\Sigma^+$ be applicable to $\eta_x(q')$ where $q'\in\mathfrak{D}(\eta_x)$.

Then

$$\begin{aligned}f_{\eta_x(q')}(z)&=f_{\alpha_x(q')F_x}(z)\\&=f'_{\psi(\alpha_x(q')F_x)}(z)\\&=f'_{\psi(\alpha_x(q'))F'_x}(z)\\&=f'_{\psi_x(\alpha_x(q'))}(z)\\&=f'_{q'}(z).\end{aligned}$$

Thus

$$f_{\eta(q')}(z)=\bigcup_{x\in\Sigma^*}\{f_{\eta_x(q')}(z)\}=f'_{q'}(z)$$

for $q'\in\mathfrak{D}(\eta)$ and all $z\in\Sigma^+$ applicable to $\eta(q')$.

If we now assume that $q_1,q_2\in Q$ are such that $q_1,q_2\in\eta(q')$ for some $q'\in\mathfrak{D}(\eta)$ and $q_1\neq q_2$, then there exists a $z\in\Sigma^*$ applicable to q_1 and q_2 such that $f_{q_1}(z)\neq f_{q_2}(z)$. Then

$$f_{q'}(z)\in\{f'_{q'}(z)\}\quad\text{and}\quad f_{q_2}(z)\in\{f'_{q'}(z)\}$$

which implies that $f_{q_1}(z)=f_{q_2}(z)$ since $\{f'_{q'}(z)\}$ is a singleton element of Θ^*. This contradicts the assumption that $q_1\neq q_2$. Consequently $\eta:Q'\to Q$ is a partial surjective function.

We now show that for $q'\in Q'$, $t\in\Sigma^*$

$$\eta(q')F_t\subseteq\eta(q'F'_t).$$

If $\eta(q')F_t\neq\varnothing$ then there exists $z\in\Sigma^*$ such that $f_{\eta(q')F_t}(z)\neq\varnothing$ since $\hat{\mathcal{M}}$ is reduced.

Now

$$f_{\eta(q')}(tz)=f'_{q'}(tz)$$

and so

$$f_{\eta(q')}(t)f_{\eta(q')F_t}(z)=f'_{q'}(t)f'_{q'F'_t}(z)$$

which implies

$$f_{\eta(q')F_t}(z)=f'_{q'F'_t}(z)\neq\varnothing.$$

Hence $q'F'_t\neq\varnothing$.

For $x\in\Sigma^*$ applicable to $q\in Q$ we have

$$f_q(x)=f'_{\psi(q)}(x)$$

and for $x\in\Sigma^*$ applicable to qF_x we have

$$f_{qF_x}(z)=f'_{\psi(qF_x)}(z)$$

and

$$f_{\eta_x(\psi(qF_x))}(z)=f'_{\psi(qF_x)}(z).$$

But

$$f_qF_x(z)=f'_{\psi(qF_x)}(z)=f'_{\psi(q)F'_x}(z)$$

and thus

$$f_{\eta_x(\psi(q)F'_x)}(z)=f'_{\psi(q)F'_x}(z)=f_{qF_x}(z).$$

Hence $\eta_x(\psi(q)F'_x)=qF_x$ for all $x\in\Sigma^*$ applicable to q.

Finally let $\eta(q')$ be defined, then

$$q'=\psi(q)F'_x\in\psi(Q)F'_x\quad\text{for some } x\in\Sigma^*,\ q\in Q$$

and

$$\begin{aligned}\eta(q')F_t\subseteq\eta_x(\psi(q)F'_x)F_t&=qF_xF_t=qF_{xt}\\&=\eta_{xt}(\psi(q)F'_{xt})=\eta(q'F'_t)\end{aligned}$$

as required. □

We can now piece together some of our earlier results. Let $\hat{\mathcal{M}}$ be a Mealy machine and suppose that $\mathcal{M}$ is the state machine of $\hat{\mathcal{M}}$. From chapters 3 and 4 we can obtain a decomposition

$$\mathcal{M}\leq\mathcal{A}_1\circ\mathcal{A}_2\circ\ldots\circ\mathcal{A}_n$$

and then by 6.3.1 the state machine $\mathcal{A}_1\circ\mathcal{A}_2\circ\ldots\circ\mathcal{A}_n$ can be provided with outputs to turn it into a Mealy machine that covers the original machine. It follows that in general a Mealy machine can be replaced by a minimal covering machine which, in turn, can then be replaced by a series of machines connected up in series and parallel which have, as underlying state machines, group machines and reset machines. This is a very significant result.

6.4 Sequential functions

For this and the next section we will use a slightly different interpretation of the behaviour of an incomplete Mealy machine. We will only consider normal Mealy machines, and the difference between their operation here and in the previous sections is concerned with the appearance of blanks on the output tape.

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a normal Mealy machine and let $q \in Q$, $x \in \Sigma^*$. If $x = \sigma_1\sigma_2 \ldots \sigma_k$ and $qF_{\sigma_1}, qF_{\sigma_1\sigma_2}, \ldots, qF_{\sigma_1\sigma_2\ldots\sigma_k}$ are all defined then the output word $f_q(x)$ is completely defined and is an element of Θ^*. We define a partial function $\bar{f}_q : \Sigma^* \to \Theta^*$ by

$$\bar{f}_q(x) = \begin{cases} f_q(x) & \text{if } qF_{\sigma_1}, qF_{\sigma_1\sigma_2}, \ldots, qF_{\sigma_1\sigma_2\ldots\sigma_k} \text{ are all defined} \\ & \text{where } x = \sigma_1\sigma_2 \ldots \sigma_k. \\ \varnothing & \text{otherwise} \end{cases}$$

This adaptation of the function f_q satisfies several properties. Clearly blanks cannot occur in $\bar{f}_q(x)$ for any $x \in \Sigma^*$. Another point of interest is that for $\bar{f}_q(x) \neq \varnothing$ the machine must stop in a defined state, i.e. qF_x must exist. Thus $\bar{f}_q(x) \neq \varnothing$ if and only if x is applicable to q and $qF_x \neq \varnothing$. In general $\bar{f}_q : \Sigma^* \to \Theta^*$ is a partial function according to this interpretation, and will be a function if $\hat{\mathcal{M}}$ is complete, in this case $\bar{f}_q = f_q$.

We can now state some simple consequences of this interpretation which are really analogues of some earlier results, namely 6.1.1, 6.2.1, 6.2.2 and 6.2.3.

Proposition 6.4.1

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a normal Mealy machine and $q \in Q$, then for $x, y \in \Sigma^*$

$$\bar{f}_q(xy) = \bar{f}_q(x) \cdot \bar{f}_{qF_x}(y).$$

Proof We need only note that if $\bar{f}_q(xy) = \varnothing$ then either $\bar{f}_q(x) = \varnothing$ or $\bar{f}_{qF_x}(y) = \varnothing$ which means that $\bar{f}_q(x) \cdot \bar{f}_{qF_x}(y) = \varnothing$. □

Theorem 6.4.2

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a normal Mealy machine. The relation defined on Q by

$$q \sim q_1 \Leftrightarrow \bar{f}_q = \bar{f}_{q_1}$$

is an equivalence relation. If $\hat{\mathcal{M}}/\sim = (Q/\sim, \Sigma, \Theta, F', G')$ is defined by

$$[q]F'_\sigma = [qF_\sigma]$$
$$[q]G'_\sigma = [qG_\sigma]$$

for $q \in Q$, $\sigma \in \Sigma$ then

(i) the function $\phi : Q \to Q/\sim$ defined by $\phi(q) = [q]$, $q \in Q$, satisfies

$$\left.\begin{array}{l} \phi(q)F'_\sigma = [qF_\sigma] \\ \phi(q)G'_\sigma = qG_\sigma \end{array}\right\} \text{ for } q \in Q, \sigma \in \Sigma;$$

(ii) $\bar{f}'_{[q]} = \bar{f}_q$ for each $q \in Q$.

We call $\hat{\mathcal{M}}/\sim$ the *minimal machine of* $\hat{\mathcal{M}}$. (An analogue of 6.2.4 also holds here.) The machine $\hat{\mathcal{M}}/\sim$ has the property that if $[q], [q_1] \in Q/\sim$ there exists a word $x \in \Sigma^*$ applicable to both $[q]$ and $[q_1]$ such that

$$\bar{f}'_{[q]}(x) \neq \bar{f}'_{[q_1]}(x).$$

This property will be described by saying that $\hat{\mathcal{M}}/\sim$ is *sequentially reduced*, or *s-reduced* for short.

Let $f : \Sigma^* \to \Theta^*$ be a partial function. We call f a *sequential partial function* if there exists a normal Mealy machine $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ and a state $q \in Q$ such that $f(x) = \bar{f}_q(x)$ for all $x \in \Sigma^*$.

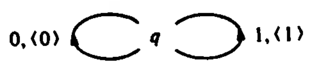
Naturally the machine $\hat{\mathcal{M}}$ may not be unique and one of our aims is to find a minimal Mealy machine satisfying the required conditions. This can be set into the more general problem of minimizing an arbitrary Mealy machine, and the minimization procedure will yield a machine with as few states as possible.

Example 6.9

Let $\Sigma = \Theta = \{0, 1\}$. The function $f : \Sigma^* \to \Theta^*$ is defined by

$$f(x) = x \quad \text{for all } x \in \Sigma^*.$$

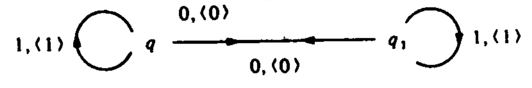
If we construct the Mealy machine:



then

$$f_q = f.$$

The Mealy machine:



also satisfies the property

$$f_q = f.$$

Furthermore

$$f_{q_1} = f.$$

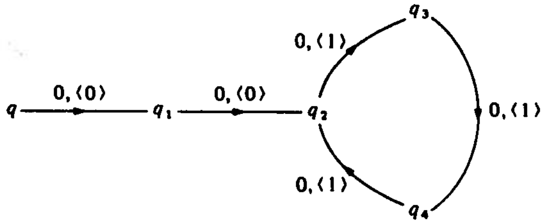
This second machine is in some sense less efficient than the first, it has more states but can do nothing more than the first machine. Both machines are complete.

Example 6.10

Let $\Sigma = \{0\}$, $\Theta = \{0, 1\}$. Define $f : \Sigma^* \to \Theta^*$ by

$$f(\Lambda) = \Lambda,\ f(0) = 0,\ f(00) = 00,\ f(0^{n+2}) = 001^n\ (n > 0).$$

To see that f is sequential we construct the Mealy machine:



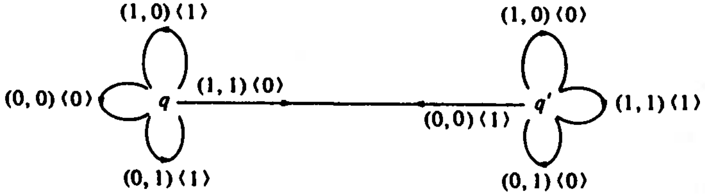
and note that $f_q = f$. This machine is also complete.

Example 6.11

Let $\Sigma = \{0, 1\}$ and suppose that $x, y \in \Sigma^*$ then x and y represent binary numbers, and we will define a machine that adds them together and gives the result as a binary number. Recall that if $x = \sigma_1\sigma_2 \ldots \sigma_k$ then x can represent a positive integer by using the expansion

$$x = \sigma_1 + \sigma_2 \cdot 2 + \sigma_3 \cdot 2^2 + \ldots \sigma_k \cdot 2^{k-1}.$$

We have written this expansion out in the reverse order to what is normal; this is caused by our convention that the tapes enter machines so that the left-most symbol is the first one read. When adding two numbers normally we look first at the right-most symbols, so these are the symbols that we must input first. Thus $2 \equiv 01$, $3 \equiv 11$, $4 \equiv 001$, $5 \equiv 101$, $6 \equiv 011$, $7 \equiv 111$, $8 \equiv 0001$ etc. Define the machine $\hat{\mathscr{M}} = (Q, \Sigma, \Theta, F, G)$, where $\Sigma = \{0, 1\} \times \{0, 1\}$, $\Theta = \{0, 1\}$,



Now let x, y be binary numbers, we first ensure that they are of the same length by adding a succession of 0s to the *right* hand side of the shortest word until they are of equal length. Now we have two words $\sigma_1 \ldots \sigma_k$ and $\sigma'_1 \ldots \sigma'_k$ representing the binary numbers x and y. If we add them together the binary representation of $x + y$ is either of length k or $k + 1$. Since our sequential machine cannot convert two words of length k into a word of length $k + 1$ we must make sure that our original inputs are of length $k + 1$ by adding a further 0 to the right of each word $\sigma_1 \ldots \sigma_k$ and $\sigma'_1 \ldots \sigma'_k$. Now we input the word $(\sigma_1, \sigma'_1)(\sigma_2, \sigma'_2) \ldots (\sigma_k, \sigma'_k)(0, 0) \in \Sigma^*$ into the machine in state q. The resulting output

$$f_q((\sigma_1, \sigma'_1) \ldots (\sigma_k, \sigma'_k)(0, 0))$$

will represent the sum $x + y$ (in our reverse binary representation).

For example $2 \equiv 01$ and so the input $(0, 0)(1, 1)(0, 0)$ will result in the sum $2 + 2$ which can be read off from the machine diagram as $001 \equiv 4$. Similarly $5 + 8$ is obtained with the input $(1, 0)(0, 0)(1, 0)(0, 1)(0, 0)$ which gives $10110 \equiv 13$, and so on. The sequential function f_q is thus a *binary adder.* The final input $(0, 0)$, which must be incorporated in any input word, is called a *carry* and ensures that the final state is q and that no part of the binary sum has been 'left at' q'.

We now consider a sequential partial function $f : \Sigma^* \to \Theta^*$. Thus a normal Mealy machine $\hat{\mathscr{M}} = (Q, \Sigma, \Theta, F, G)$ exists such that $f = \bar{f}_q$ for some $q \in Q$. Because of 6.4.2 we can replace $\hat{\mathscr{M}}$ by the minimal machine $\hat{\mathscr{M}}/\sim$ and then $f = \bar{f}'_{[q]}$. This means that given a sequential partial function we can find an s-reduced machine to represent the partial function. To ensure that the machine is the most efficient possible we remove all states that cannot be reached from the initial state.

Let $f : \Sigma^* \to \Theta^*$ be a sequential partial function and $\hat{\mathscr{M}} = (Q, \Sigma, \Theta, F, G)$ the minimal, s-reduced Mealy machine such that $f = \bar{f}_q$ for some $q \in Q$.

Form the set $Q_f = \{qF_x \mid x \in \Sigma^*\}$ and define the Mealy machine

$$\hat{\mathscr{M}}_f = (Q_f, \Sigma, \Theta, F_1, G_1)$$

where

$$q'(F_1)_\sigma = q'F_\sigma$$

and

$$q'(G_1)_\sigma = q'G_\sigma$$

for $\sigma \in \Sigma$, $q' \in Q_f$.

The pair $(\hat{\mathscr{M}}_f, q)$ is called the *minimal sequential machine* for f. Our terminology implies that $(\hat{\mathcal{M}}_f, q)$ is unique, but to be more precise it is only unique up to isomorphism, where this is defined next.

Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ and $\hat{\mathcal{M}}' = (Q', \Sigma, \Theta, F', G')$ be complete Mealy machines. A function $\psi : Q \to Q'$ is called a *Mealy machine homomorphism* if

$$\psi(qF_\sigma) = \psi(q)F'_\sigma$$
$$qG_\sigma = \psi(q)G'_\sigma$$

for $q \in Q$, $\sigma \in \Sigma$.

If ψ is a bijective function then ψ is called an *isomorphism.*

Theorem 6.4.3

Let $f : \Sigma^* \to \Theta^*$ be a sequential partial function. Suppose that s-reduced Mealy machines

$$\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G) \quad \text{and} \quad \hat{\mathcal{M}}' = (Q', \Sigma, \Theta, F', G')$$

exist such that $f = \bar{f}_q$ for some $q \in Q$ and $f = \bar{f}'_{q'}$ for some $q' \in Q'$. If $(\hat{\mathcal{M}}_f, q)$ and $(\hat{\mathcal{M}}_{f'}, q')$ are minimal sequential machines of f then there exists an isomorphism

$$\psi : \hat{\mathcal{M}}_f \to \hat{\mathcal{M}}'_{f'} \quad \text{such that } \psi(q) = q'.$$

Proof Let $x, z \in \Sigma^*$. Since $\bar{f}_q = \bar{f}'_{q'}$ we have $\bar{f}_q(xz) = \bar{f}'_{q'}(xz)$ and so

$$\bar{f}_q(x) \cdot \bar{f}_{qF_x}(z) = \bar{f}'_{q'}(x) \cdot \bar{f}'_{q'F'_x}(z)$$

which implies that $\bar{f}_{qF_x}(z) = \bar{f}'_{q'F'_x}(z)$ and so

$$\bar{f}_{qF_x} = \bar{f}_{q'F'_x}.$$

Define $\psi : \hat{\mathcal{M}}_f \to \hat{\mathcal{M}}'_f$ by

$$\psi(qF_x) = q'F'_x \quad (x \in \Sigma^*).$$

This is well-defined for if $qF_x = qF_y$ $(x, y \in \Sigma^*)$ then

$$\bar{f}_{qF_x} = \bar{f}_{qF_y} \quad \text{and also} \quad \bar{f}_{qF_x} = \bar{f}'_{q'F'_x},$$

$\bar{f}_{qF_y} = \bar{f}'_{q'F'_y}$ which implies that $\bar{f}'_{q'F'_x} = \bar{f}'_{q'F'_y}$. By the s-reduced nature of $\hat{\mathcal{M}}'$ we must then have $q'F'_x = q'F'_y$. A similar argument yields the fact that ψ is one-one. It is clearly onto and for $x \in \Sigma^*$

$$\psi(qF_x) = \psi(q)F'_x \quad \text{since } q' = \psi(q).$$

Given $\sigma \in \Sigma$ we have

$$\psi(qF_xF_\sigma) = \psi(qF_x)F'_\sigma$$

and

$$qF_xG_\sigma = \bar{f}_{qF_x}(\sigma) = \bar{f}'_{q'F'_x}(\sigma) = (qF_x)G'_\sigma.$$

Hence ψ is an isomorphism. □

Theorem 6.4.4

Let $\psi : \hat{\mathcal{M}} \to \hat{\mathcal{M}}'$ be a Mealy machine homomorphism and suppose that $q \in Q$, the state set of $\hat{\mathcal{M}}$. Then

$$\bar{f}_q = \bar{f}'_{\psi(q)}.$$

Proof This is left as an exercise. □

Given a partial sequential function $f : \Sigma^* \to \Theta^*$ we can now consider associating with it a minimal machine $(\hat{\mathcal{M}}_f, q)$ in an essentially unique way, any other minimal machine will be isomorphic by 6.4.3. This justifies calling $(\hat{\mathcal{M}}_f, q)$ *the minimal machine of* f. By the construction of $\hat{\mathcal{M}}_f$ it is accessible in the sense that any state of $\hat{\mathcal{M}}_f$ occurs as an image of q under a suitable input. This ensures that there are no 'redundant' states in $\hat{\mathcal{M}}_f$.

Let $f : \Sigma^* \to \Theta^*$ and $g : \Theta^* \to \Gamma^*$ be partial sequential functions and suppose that $(\hat{\mathcal{M}}_f, q)$, $(\hat{\mathcal{M}}'_g, q')$ are the minimal machines of f and g. Writing

$$\hat{\mathcal{M}}_f = (Q, \Sigma, \Theta, F, G) \quad \text{and} \quad \hat{\mathcal{M}}'_g = (Q', \Theta, \Gamma, F', G')$$

we can now form the Mealy machine

$$\hat{\mathcal{M}}'_g \omega \hat{\mathcal{M}}_f = (Q' \times Q, \Sigma, \Gamma, F^\omega, G^\omega)$$

where

$$(q'_1, q_1)F^\omega_\sigma = (q'_1F'_{\omega(q,\sigma)}, q_1F_\sigma)$$

and

$$(q'_1, q_1)G^\omega_\sigma = q'_1G'_{\omega(q_1,\sigma)} \quad \text{for } q_1 \in Q_1, q'_1 \in Q'_1, \sigma \in \Sigma,$$

and where

$$\omega : Q \times \Sigma \to \Theta$$

is defined by

$$\omega(q, \sigma) = G(q, \sigma) \quad \text{for } q \in Q, \sigma \in \Sigma.$$

Consider the partial sequential function $h : \Sigma^* \to \Gamma^*$ defined by this machine in state (q', q). Let $\sigma \in \Sigma$, then

$$\begin{aligned} h(\sigma) &= q'G'_{\omega(q,\sigma)} \\ &= q'G'_{qG_\sigma} \\ &= q'G'_{f(\sigma)} \\ &= g(f(\sigma)). \end{aligned}$$

For $x \in \Sigma^*, \sigma \in \Sigma$

$$\begin{aligned} h(x\sigma) &= h(x) \cdot (q', q)F_x^{\omega}G_\sigma^{\omega} \\ &= h(x) \cdot (q'F'_{\omega^+(q,x)}, qF_x)G_\sigma^{\omega} \quad \text{using the notation of 2.6} \\ &= h(x) \cdot (q'F'_{\omega^+(q,x)})G'_{\omega(qF_x,\sigma)} \\ &= h(x) \cdot q'F'_{\omega^+(q,x)}G'_{qF_xG_\sigma} \\ &= h(x) \cdot q'F'_{f(x)}G'_{qF_xG_\sigma} \\ &= g(f(x)qG_xG_\sigma) \\ &= g(f(x\sigma)) \end{aligned}$$

providing that we can establish the identity

$$\omega^+(q_1, x) = \bar{f}_{q_1}(x) \quad \text{for } q_1 \in Q_1, x \in \Sigma^*.$$

Now for $\sigma \in \Sigma$, $\omega^+(q, \sigma) = \omega(q, \sigma) = qG_\sigma = f(\sigma)$. Let us consider $\alpha \in \Sigma^*$ and

$$\begin{aligned} \omega^+(q_1, \sigma\alpha) &= \omega(q_1, \sigma)\omega^+(q_1F_\sigma, \alpha) \\ &= \bar{f}_{q_1}(\sigma) \cdot \bar{f}_{q_1F_\sigma}(\alpha) \\ &= \bar{f}_{q_1}(\sigma\alpha) \end{aligned}$$

by the usual inductive process. Hence $\omega^+(q_1, x) = \bar{f}_{q_1}(x)$ as required. Consequently the machine $\hat{\mathcal{M}}'_g\omega\hat{\mathcal{M}}_f$ in state (q', q) defines the partial sequential function $g \circ f : \Sigma^* \to \Gamma^*$.

Thus the composition of two partial sequential functions is again a partial sequential function.

The other 'products' defined between Mealy machines give rise to natural operations on the corresponding partial sequential functions. For example, given Mealy machines $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ and $\hat{\mathcal{M}}' = (Q', \Sigma', \Theta', F', G')$ we can define the product

$$\hat{\mathcal{M}} \times \hat{\mathcal{M}}' = (Q \times Q', \Sigma \times \Sigma', \Theta \times \Theta', F \times F', G \times G')$$

which will then define a partial sequential function

$$\overline{(f \times f')}_{(q,q')} : (\Sigma \times \Sigma')^* \to (\Theta \times \Theta')^*$$

by

$$\overline{(f \times f')}_{(q,q')}(x, x') = (\bar{f}_q(x), \bar{f}'_{q'}(x'))$$

for $x \in \Sigma^*, x' \in (\Sigma')^*, q \in Q, q' \in Q'$.

6.5 Decompositions of sequential functions

In this final section we will apply some of our earlier results to problems associated with sequential functions.

Let $f : \Sigma^* \to \Theta^*$ be a partial sequential function and suppose then $(\hat{\mathcal{M}}_f, q)$ is the minimal machine of f. If $\hat{\mathcal{M}}_f = (Q_f, \Sigma, \Theta, F, G)$, then (Q_f, Σ, F) is a state machine, and so we may construct the transformation semigroup $\mathbf{TS}(Q_f, \Sigma, F)$ and we call this the *syntactic transformation semigroup* of f. It will be convenient to write this as

$$\mathcal{A}_f = (Q_f, S_f).$$

Suppose that $\mathcal{A}_f$ has a decomposition of the form

$$\mathcal{A}_f \leq \mathcal{B}_1 \circ \ldots \circ \mathcal{B}_n,$$

what can we say about the sequential function f?

Theorem 6.5.1

(Eilenberg [1976]) Let $f : \Sigma^* \to \Theta^*$ be a partial sequential function and let $\mathcal{A}_f \leq \mathcal{B}_1 \circ \mathcal{B}_2$. Then there exist partial sequential functions $g_1 : \Sigma^* \to \Gamma^*$, $g_2 : \Gamma^* \to \Theta^*$ such that

$$f \subseteq g_1 \circ g_2$$

and

$$\mathcal{A}_{g_1} \leq \mathcal{B}_1, \quad \mathcal{A}_{g_2} \leq \mathcal{B}_2.$$

Proof Consider the syntactic transformation semigroup $\mathcal{A}_f = (Q_f, S_f)$ of the minimal machine $(\hat{\mathcal{M}}_f, q)$. For each $x \in \Sigma^+$ we have $F_x \in S_f$. If $\mathcal{B}_1 = (P_1, T_1)$ and $\mathcal{B}_2 = (P_2, T_2)$ then $\mathcal{A} \leq \mathcal{B}_1 \circ \mathcal{B}_2$ implies that a partial function $\psi : P_1 \times P_2 \to Q_f$ exists such that for each $s \in S_f$ there exists $t_2^s \in T_2$, $h^s : P_2 \to T_1$ such that

$$\psi(p_1, p_2)s \subseteq \psi(p_1h^s(p_2), p_2t_2^s)$$

for $(p_1, p_2) \in P_1 \times P_2$. Let $(i_1, i_2) \in P_1 \times P_2$ such that $\psi(i_1, i_2) = q$.

Put $\Gamma = P_2 \times \Sigma \times T_1$ and define

$$\hat{\mathcal{M}}_2 = (P_2, \Sigma, \Gamma, F^2, G^2)$$

by

$$\begin{aligned} p_2F_\sigma^2 &= p_2t_2^s \quad \text{where } F_\sigma = s \in S_f \\ p_2G_\sigma^2 &= (p_2, \sigma, h^s(p_2)). \end{aligned}$$

Then consider the partial sequential function

$$g_2 : \Sigma^* \to \Gamma^*$$

defined by $\hat{\mathcal{M}}_2$ in state i_2.

Now put $\hat{\mathcal{M}}_1 = (P_1, \Gamma, \Theta, F^1, G^1)$ where

$$p_1F_\gamma^1 = p_1t_1 \quad \text{if } \gamma = (p_2, \sigma, t_1)$$

and

$$p_1G^1_\gamma = \begin{cases} \psi(p_1, p_2)G_\sigma & \text{if } \psi(p_1, p_2) \neq \varnothing \neq p_1t_1 \\ \varnothing & \text{if } p_1t_1 = \varnothing \\ \text{arbitrary} & \text{otherwise.} \end{cases}$$

Now $g_1 : \Gamma^* \to \Theta^*$ is defined by $\hat{\mathscr{M}}_1$ in state i_1. The partial function $g_1 \circ g_2 : \Sigma^* \to \Theta^*$ is then defined by $\hat{\mathscr{M}}_1 \omega \hat{\mathscr{M}}_2$ in state (i_1, i_2).

For $p_1 \in P_1$, $p_2 \in P_2$, $\sigma \in \Sigma$ there exists $s = F_\sigma \in S_f$ and then

$$\begin{aligned}(p_1, p_2)\bar{G}_\sigma &= p_1G^1_{G^2(p_2,\sigma)} \\ &= p_1G^1_\gamma \quad \text{where } \gamma = (p_2, \sigma, h^s(p_2)) \\ &= \psi(p_1, p_2)G_\sigma\end{aligned}$$

whenever $\psi(p_1, p_2) \neq \varnothing \neq p_1h^s(p_2)$. (Here $\bar{G}$ is the output function from $\hat{\mathscr{M}}_1 \omega \hat{\mathscr{M}}_2$.) If $p_1h^s(p_2) = \varnothing$ then $(p_1, p_2)(h^s, t^s) = \varnothing$ and so $\psi(p_1, p_2)s = \varnothing$. Consequently $f \subseteq g_1 \circ g_2$ as required. □

Theorem 6.5.2

Let $f : \Sigma^* \to \Theta^*$ be a partial sequential function and let

$$\mathscr{A}_f \leq \mathscr{B}_1 \times \mathscr{B}_2,$$

then there exist partial sequential functions

$$g_1 : \Sigma^* \to \Gamma_1^*$$
$$g_2 : \Sigma^* \to \Gamma_2^*$$

and a function $\beta : \Gamma_1 \times \Gamma_2 \to \Theta$ such that

$$f(x) = (g_1 \wedge g_2)(\beta(x)),$$

where

$$(g_1 \wedge g_2)(x) = (g_1(x), g_2(x))$$

for $x \in \Sigma^*$ and

$$\mathscr{A}_{g_1} \leq \mathscr{B}_1, \quad \mathscr{A}_{g_2} \leq \mathscr{B}_2.$$

Proof This construction follows a similar argument to the previous proof. However we use the Mealy machine construction $\hat{\mathscr{M}}_1 \wedge \hat{\mathscr{M}}_2$, that is, the restricted direct product. Here the alphabets Γ_1 and Γ_2 are defined to be $P_1 \times \Sigma$ and $P_2 \times \Sigma$ respectively.

$$\beta : \Gamma_1 \times \Gamma_2 \to \Theta$$

is defined by

$$\beta(p_1, \sigma, p_2, \sigma') = \begin{cases} (\psi(p_1, p_2), \sigma)G_\sigma & \text{if } \sigma = \sigma' \text{ and } \psi(p_1, p_2) \neq \varnothing \\ \text{arbitrary} & \text{otherwise,} \end{cases}$$

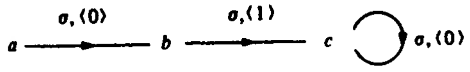
where $\psi : P_1 \times P_2 \to Q$ is the covering partial function. The details are left to the reader; they will be found in Eilenberg [1976].

Example 6.12

Let $\Sigma = \{\sigma\}$, $\Theta = \{0, 1\}$ and $f : \Sigma^* \to \Theta^*$ be defined by

$$f(\sigma) = 0,\ f(\sigma^2) = 01,\ f(\sigma^n) = 010^{n-2}$$

for $n \geq 3$. Then f is a sequential function defined by the complete Mealy machine started in state a:



This has state machine $\mathscr{C}_{(1,2)}$ which, by the holonomy decomposition theorem, has a covering

$$\mathscr{C}_{(1,2)} \geq \bar{\mathbf{2}} \circ \mathscr{C}.$$

The covering $\psi : P_1 \times P_2 \to Q$ (using the notation of theorem 6.5.1) is given by

$$a = \psi((b, a)) = \psi((c, a))$$
$$b = \psi((b, \{b, c\}))$$
$$c = \psi((c, \{b, c\}))$$

where $P_1 = \{b, c\}$, $P_2 = \{\{a\}, \{b, c\}\}$. If $T_1 = \{t_1, t_1'\}$, $P_1t_1 = \{c\}$, $P_1t_1' = \{b\}$, $T_2 = \{t_2\}$ then $h^\sigma : P_2 \to T_1$ is defined by $h^\sigma(\{a\}) = t_1'$, $h^\sigma(\{b, c\}) = t_1$ and $g_2 : \Sigma^* \to \Gamma^*$ is defined by

$$g_2(\sigma) = (a, \sigma, t_1'), \quad g_2(\sigma^2) = (a, \sigma, t_1') \cdot (\{b, c\}, \sigma, t_1)$$

and generally

$$g_2(\sigma^n) = (a, \sigma, t_1')(\{b, c\}, \sigma, t_1) \ldots (\{b, c\}, \sigma, t_1)$$

for $n > 1$.

Now

$$\begin{aligned}g_1(g_2(\sigma)) &= g_1(a, \sigma, t_1') \\ &= \psi((b, a))G_\sigma = aG_a = 0 \\ g_1(g_2(\sigma^2)) &= 0bF'_{(a,\sigma,t_1)}G'_\gamma \quad \text{where } \gamma = (\{b, c\}, \sigma, t_1) \\ &= 0bt_1G'_\gamma \\ &= 0\psi(bt_1, \{b, c\})G_\sigma \\ &= 0bG_\sigma \\ &= 01\end{aligned}$$

$$\begin{aligned} g_1(g_2(\sigma^3)) &= 01(bF'_\gamma G'_\gamma) \\ &= 01(cG'_\gamma) \\ &= 01(cG_\sigma) \\ &= 010. \end{aligned}$$

Continuing we see that

$$g_1(g_2(\sigma^n)) = 010^{n-2} \quad \text{for } n \geq 3.$$

6.6 Conclusion

We have seen how the concept of a Mealy machine can be used to model a variety of discrete situations. Using the results of this chapter we can analyse the underlying state machine and transformation semigroup by means of our results from chapters 3 and 4. To recover information about the original situation we can apply the results of this chapter to give facts about Mealy machine coverings, or if the model is concerned with sequential functions we can decompose them. By choosing a suitable decomposition theory we can then highlight various properties of our original model and this may well throw light on the situation that we are modelling.

The subject discussed here is undergoing much rapid development and it is likely that over the next few years many new and useful results will appear. For those interested in reading further I would strongly recommend that the two masterful volumes by S. Eilenberg be studied.

6.7 Exercises

6.1 Let $\hat{\mathcal{M}} = (Q, \Sigma, \Theta, F, G)$ be a Mealy machine and $i \in Q$ a given initial state. Consider the partial sequential function $\bar{f}_i : \Sigma^* \to \Theta^*$. Let $\mathfrak{M} = (\mathcal{M}, i, Q)$ be a recognizer defined by $\mathcal{M} = (Q, \Gamma, H)$ where $\Gamma = \Sigma \times \Theta$, $H : Q \times \Sigma \to Q$ is given by

$$(q_1(\sigma, \theta))H = qF_\sigma \quad \text{if and only if } qG_\sigma = \theta,\ q \in Q,\ \sigma \in \Sigma,\ \theta \in \Theta.$$

Prove that $|\mathfrak{M}| = \{(\alpha, \beta) \mid \alpha \in \Sigma^*, \beta \in \Theta^*, \bar{f}_i(\alpha) = \beta\}$.
This shows that $\bar{f}_i$ is a *rational* function, i.e. one whose graph is a rational subset of $\Sigma^* \times \Theta^*$.

6.2 Prove 6.2.2.

6.3 In the notation of 6.2.8 prove the following:

$$(\pi^c)^* = \pi$$
$$(\tau^*)^c \leq \tau.$$

6.4 Examine 6.3.3 in the case where $\psi : Q \to Q'$ satisfies

$$f_q(x) \subseteq f'_{\psi(q)}(x) \quad \text{for all } x \in \Sigma^*.$$

6.5 Prove that if $\psi : \hat{\mathcal{M}} \to \hat{\mathcal{M}}'$ is an isomorphism and $\psi(q) = q'$, then $f_q = f'_{q'}$.

6.6 Prove 6.5.2.

6.7 Minimize (if possible) the following machines (where $\Sigma = \Theta = \{0, 1\}$):

(i)

	$\hat{\mathcal{M}}_1$	a	b	c	d	e	f
F	0	a	b	c	e	b	a
	1	c	d	c	b	d	f
G	0	0	1	0	1	0	1
	1	1	1	1	1	0	1

(ii)

	$\hat{\mathcal{M}}_2$	a	b	c	d	e	f	g
F	0	a	d	b	f	c	a	c
	1	c	d	e	d	a	b	d
G	0	$\varnothing$	0	$\varnothing$	0	1	$\varnothing$	1
	1	1	$\varnothing$	1	0	$\varnothing$	0	1

(iii)

	$\hat{\mathcal{M}}_3$	a	b	c	d	e	f	g
F	0	a	d	b	c	c	a	c
	1	$\varnothing$	$\varnothing$	e	a	a	b	d
G	0	$\varnothing$	0	0	$\varnothing$	0	$\varnothing$	1
	1	1	0	0	1	$\varnothing$	1	1

6.8 Describe the sequential partial functions f_a, $\bar{f}_a$ for the machine

	$\hat{\mathcal{M}}$	a	b	c
F	0	b	c	a
	1	a	$\varnothing$	$\varnothing$
G	0	1	1	1
	1	0	$\varnothing$	$\varnothing$

6.9 Describe the sequential function $f_{(\mathrm{off},\mathrm{off})}$ of example 3.2.7.

6.10 Describe the sequential function f_{off} of example 3.2.9. Find a minimal machine for this function.

6.11 Complete the details of theorem 6.5.2.

Appendix

The following program evaluates the semigroup of a state machine with up to five states and nine inputs. The semigroup elements are listed and a semigroup multiplication table constructed. The states are described by numbers 1, 2, 3, 4, 5 and the inputs by letters *A*, *B*, *C*, *D*, *E*, ... The next state function is described as an *n*-tuple ($n \leq 5$). Implementation is on an Apple or ITT 2020 microcomputer running Apple Pascal with a printer. The program was written by Dr A. W. Wickstead, Department of Pure Mathematics, Queen's University, Belfast.

```
PROGRAM SEMIGROUPS;
CONST BLANK='               ';(*15 BLANKS*)
SEPARATOR='------------------------------------------------------------------';
   (* '-' 66 TIMES*)
      MAXWORD=15;
      STACKSIZE=50;
TYPE WORD=RECORD VAL:PACKED ARRAY[1..5] OF 1..5;
                 STR:STRING[MAXWORD]
                 END;
      VALUES=PACKED ARRAY[1..5] OF 1..5;
VAR PRINT:TEXT;
     SGFILE:FILE OF WORD;
     DOMPOINT,I,J:INTEGER;
     OPTION,NUM:CHAR;
     FILENAME:STRING;

PROCEDURE CONTINUE;
BEGIN
WRITELN;
WRITE('PRESS RETURN TO CONTINUE ');
READLN;
END;

PROCEDURE GENERATE;
VAR PRINTCNT,WORDSIZE,WHICH,WORDNUM,FUNCTNUM,DOMSIZE,FUNCTSIZE:INTEGER;
     NEWONE:WORD;
     STACKEND:ARRAY[0..1] OF INTEGER;
     STACK:ARRAY[0..1,1..STACKSIZE] OF WORD;
     STARTER:ARRAY[1..9] OF WORD;
     USED:PACKED ARRAY [1..5,1..5,1..5,1..5,1..5] OF BOOLEAN;
     TEMP:STRING[1];

PROCEDURE CHECKPAGE;
VAR I:INTEGER;
BEGIN
IF PRINTCNT>59 THEN
```

```
  BEGIN
  FOR I:=1 TO 66-PRINTCNT DO WRITELN(PRINT);
  PRINTCNT:=0;
  END
END;

PROCEDURE WORDMESS(I:INTEGER);
BEGIN
WRITELN(PRINT);
WRITELN(PRINT,'WORD SIZE: ',WORDSIZE,' NUMBER OF NEW WORD(S): ',STACKEND[I]);
WRITELN(PRINT);
PRINTCNT:=PRINTCNT+3;
CHECKPAGE;
END;

PROCEDURE OUT;
BEGIN
WRITE(PRINT,'(');
FOR DOMPOINT:=1 TO DOMSIZE DO
  BEGIN
  WRITE(PRINT,NEWONE.VAL[DOMPOINT]);
  IF DOMPOINT<DOMSIZE THEN WRITE(PRINT,',') ELSE WRITE(PRINT,')  ');
  END;
WRITELN(PRINT,NEWONE.STR);
PRINTCNT:=PRINTCNT+1;
CHECKPAGE;
(*$I-*)
SGFILE^:=NEWONE;
PUT(SGFILE);
(*$I+*)
END;

PROCEDURE SETUP;
BEGIN
WRITELN('FILENAME FOR OUTPUT? (<RETURN> FOR');
WRITE('NONE ');
READLN(FILENAME);
IF (FILENAME<>'') THEN IF (FILENAME[LENGTH(FILENAME)]<>':')  THEN
      REWRITE(SGFILE,FILENAME) ELSE WRITELN('NO OUTPUT FILE OPENED');
FOR DOMPOINT:=1 TO 5 DO NEWONE.VAL[DOMPOINT]:=DOMPOINT;
NUM:=CHR(0);
PRINTCNT:=0;
TEMP:='?';
WORDSIZE:=1;
WRITE('SIZE OF DOMAIN (1..5)? ');
REPEAT UNITREAD(2,NUM,1) UNTIL NUM IN ['1'..'5'];
     (*THIS USE OF UNITREAD READS SINGLE CHARACTER NUM FROM KEYBOARD WITHOUT
                   PRINTING IT ON THE VDU*)
WRITELN(NUM);
DOMSIZE:=ORD(NUM)-48;
WRITE('NUMBER OF FUNCTIONS (1..9)? ');
REPEAT UNITREAD(2,NUM,1) UNTIL NUM IN ['1'..'9'];
WRITELN(NUM);
FUNCTSIZE:=ORD(NUM)-48;
FOR FUNCTNUM:=1 TO FUNCTSIZE DO
  BEGIN
  FOR DOMPOINT:=1 TO DOMSIZE DO
    BEGIN
    WRITE('VALUE OF FUNCTION ',FUNCTNUM,' AT ',DOMPOINT,' ? ');
    REPEAT UNITREAD(2,NUM,1) UNTIL NUM IN ['1'..CHR(48+DOMSIZE)];
    WRITELN(NUM);
    NEWONE.VAL[DOMPOINT]:=ORD(NUM)-48;
    TEMP[1]:=CHR(64+FUNCTNUM);
    NEWONE.STR:=TEMP;
    END;
  STACK[0,FUNCTNUM]:=NEWONE;
  STARTER[FUNCTNUM]:=NEWONE;
  OUT;
  END;
STACKEND[0]:=FUNCTSIZE;
WRITELN(PRINT);
WRITELN(PRINT,'ORIGINAL ',FUNCTSIZE,' FUNCTION(S)');
WRITELN(PRINT);
PRINTCNT:=PRINTCNT+3;
CHECKPAGE;
WHICH:=1;
FILLCHAR(USED,SIZEOF(USED),CHR(0));
```

```
FOR FUNCTNUM:=1 TO FUNCTSIZE DO USED[STARTER[FUNCTNUM].VAL[1],
    STARTER[FUNCTNUM].VAL[2],STARTER[FUNCTNUM].VAL[3],STARTER[FUNCTNUM].VAL[4],
    STARTER[FUNCTNUM].VAL[5]]:=TRUE;
END;

PROCEDURE TOOHARD;
BEGIN
WRITELN('SEMIGROUP IS TOO BIG FOR THIS PROGRAM');
(*$I-*)
CLOSE(SGFILE,PURGE);
(*$I+*)
CONTINUE;
EXIT(GENERATE)
END;

PROCEDURE NEWLEVEL;
BEGIN
STACKEND[WHICH]:=0;
FOR WORDNUM:=1 TO STACKEND[1-WHICH] DO (*FOR EACH NEW WORD AT LAST LEVEL*)
  BEGIN
  FOR FUNCTNUM:=1 TO FUNCTSIZE DO (*FOR EACH ORIGINAL FUNCTION*)
    BEGIN
    FOR DOMPOINT:=1 TO DOMSIZE DO (*FOR EACH POINT OF DOMAIN*)
      BEGIN
      NEWONE.VAL[DOMPOINT]:=STARTER[FUNCTNUM].VAL[STACK[1-WHICH,WORDNUM].
          VAL[DOMPOINT]]; (*VALUE OF ((ORIGINAL FUNCTION))*WORD) AT DOMPOINT*)
      END;
    NEWONE.STR:=CONCAT(STACK[1-WHICH,WORDNUM].STR,STARTER[FUNCTNUM].STR);
    IF NOT USED[NEWONE.VAL[1],NEWONE.VAL[2],NEWONE.VAL[3],NEWONE.VAL[4],
                                                    NEWONE.VAL[5]] THEN

      BEGIN
      IF WORDSIZE>MAXWORD THEN TOOHARD;
      USED[NEWONE.VAL[1],NEWONE.VAL[2],NEWONE.VAL[3],NEWONE.VAL[4],
                                                    NEWONE.VAL[5]]:=TRUE;

      STACKEND[WHICH]:=STACKEND[WHICH]+1;
      IF STACKEND[WHICH]>STACKSIZE THEN TOOHARD;
      STACK[WHICH,STACKEND[WHICH]]:=NEWONE;
      OUT;
      END;
     END
   END;
 WORDSIZE:=WORDSIZE+1;
 IF STACKEND[WHICH]<>0 THEN WORDMESS(WHICH);
 WHICH:=1-WHICH;
 END;

 BEGIN(*GENERATE*)
 PAGE(OUTPUT);
 GOTOXY(10,6);
 WRITELN('SEMIGROUP GENERATION');
 GOTOXY(0,10);
 WRITELN('YOU MAY SPECIFY UP TO 9 FUNCTIONS ON');
 WRITELN('A SET OF UP TO 5 ELEMENTS. THE PROGRAM');
 WRITELN('WILL LIST THE ELEMENTS IN THE ');
 WRITELN('SEMIGROUP THAT THEY GENERATE, AND A');
 WRITELN('DESCRIPTION OF EACH IN TERMS OF THE');
 WRITELN('ORIGINAL FUNCTIONS.');
 WRITELN;
 SETUP;
 REPEAT NEWLEVEL UNTIL STACKEND[1-WHICH]=0;
 (*$I-*)
 CLOSE(SGFILE,LOCK);
 (*$I+*)
 FOR I:=1 TO 66-PRINTCNT DO WRITELN(PRINT);
 END;

 PROCEDURE MULTIPLY;
 VAR PRODUCT:VALUES;
     STACK:ARRAY[0..255] OF VALUES;
     CODE:PACKED ARRAY [1..5,1..5,1..5,1..5,1..5] OF 0..255;
     PAGESWD,PAGESHT,SIZE,I,J:INTEGER;
     LIST:ARRAY[0..255] OF STRING[MAXWORD];


 PROCEDURE SETUP;
 BEGIN
 WRITE('FILENAME FOR INPUT? ');
 REPEAT READLN(FILENAME) UNTIL FILENAME<>'';
```

```
RESET(SGFILE,FILENAME);
FILLCHAR(CODE,SIZEOF(CODE),CHR(0));
SIZE:=0;
REPEAT
  BEGIN
  STACK[SIZE]:=SGFILE^.VAL;
  LIST[SIZE]:=CONCAT(SGFILE^.STR,COPY(BLANK,1,15-LENGTH(SGFILE^.STR)));
  CODE[SGFILE^.VAL[1],SGFILE^.VAL[2],SGFILE^.VAL[3],SGFILE^.VAL[4],
                SGFILE^.VAL[5]]:=SIZE;
  (*$I-*)
  GET(SGFILE);
  (*$I+*)
  SIZE:=SIZE+1;
  END
UNTIL EOF(SGFILE) OR (SIZE>255);
IF NOT EOF(SGFILE) THEN
  BEGIN
  WRITELN('SEMIGROUP IS TOO BIG TO COMPUTE TABLE');
  WRITELN('ROUTINE ABORTING');
  CONTINUE;
  CLOSE(SGFILE,LOCK);
  EXIT(MULTIPLY);
  END;
CLOSE(SGFILE,LOCK);
PAGESHT:=(SIZE-1) DIV 60;
PAGESWD:=(SIZE-1) DIV 7;
END;

PROCEDURE PRINTPAGE(I,J:INTEGER);
VAR K,L,X,Y:INTEGER;
BEGIN
WRITE(PRINT,BLANK,CHR(124));
FOR K:=0 TO 6 DO
  BEGIN
  IF 7*J+K<SIZE THEN WRITE(PRINT,' ',LIST[7*J+K]);
  END;
WRITELN(PRINT);
WRITELN(PRINT,SEPARATOR,SEPARATOR);
FOR L:=0 TO 59 DO
  BEGIN
  X:=60*I+L;
  IF X<SIZE THEN
    BEGIN
    WRITE(PRINT,LIST[X],CHR(124));
    FOR K:=0 TO 6 DO
      BEGIN
      Y:=7*J+K;
      IF Y<SIZE THEN
        BEGIN
        FOR DOMPOINT:=1 TO 5 DO
          BEGIN
          PRODUCT[DOMPOINT]:=STACK[Y,STACK[X,DOMPOINT]];
          END;
        WRITE(PRINT,' ',LIST[CODE[PRODUCT[1],PRODUCT[2],PRODUCT[3],
                                                        PRODUCT[4],PRODUCT[5]]]);
        END;
      END;
    END;
  WRITELN(PRINT);
  END;
FOR L:=0 TO 3 DO WRITELN(PRINT);
END;

BEGIN(*MULTIPLY*)
PAGE(OUTPUT);
GOTOXY(7,6);
WRITELN('PRINT MULTIPLICATION TABLE');
GOTOXY(0,10);
WRITELN('THIS ROUTINE WILL PRINT THE TABLE');
WRITELN('OF A FUNCTION SEMIGROUP THAT HAS BEEN');
WRITELN('PRODUCED BY THE GENERATION OPTION');
WRITELN;
SETUP;
FOR I:=0 TO PAGESHT DO
  BEGIN
  FOR J:=0 TO PAGESWD DO PRINTPAGE(I,J)
  END;
END;
```

```
BEGIN(*PROGRAM*)
OPTION:=CHR(0);
REWRITE(PRINT,'REMOUT:');
REPEAT
  BEGIN
  PAGE(OUTPUT);
  GOTOXY(10,6);
  WRITELN('FUNCTION SEMIGROUPS');
  GOTOXY(9,8);
  WRITELN('(C) 1981 A.W. WICKSTEAD');
  GOTOXY(0,12);
  WRITELN('OPTIONS: G]ENERATE SEMIGROUP');
  WRITELN;
  WRITELN('         M]ULTIPLICATION TABLE');
  WRITELN;
  WRITELN('         Q]UIT');
  REPEAT UNITREAD(2,OPTION,1) UNTIL OPTION IN ['G','M','Q'];
  CASE OPTION OF
    'G':GENERATE;
    'M':MULTIPLY;
    'Q':PAGE(OUTPUT)
    END(*CASES*)
  END;
UNTIL OPTION='Q';
CLOSE(PRINT);
END.
```

References

Arbib, M. A. [1964] *Brains, machines and mathematics.* McGraw-Hill, New York

Arbib, M. A. [1969] 'Memory limitations of S-R models'. *Psychol. Review* **76** 507–10

Arbib, M. A. [1969A] *Theories of abstract automata.* Prentice-Hall, Englewood Cliffs, N.J.

Chittenden, B. [1978] 'Specification of software as finite state automata' in *Proceedings of the International Symposium on Mini and Micro Computers.* I.E.E.E. New York pp. 105–11

Cohn, P. M. [1975] 'Algebra and Language theory'. *Bull. L.M.S.* **7** 1–29

Dilger, E. [1976] 'On permutation-reset automata'. *Information and Control* **30** 86–95

Eilenberg, S. [1974] *Automata, languages and machines, Vol. A.* Academic Press, New York

Eilenberg, S. [1976] *Automata, languages and machines, Vol. B.* Academic Press, New York.

Fiksel, J. R. and Bower, G. H. [1976] 'Question answering by a semantic network of parallel automata'. *J. Math. Psychol.* **13** 1–45

Ginzburg, A. [1968] *Algebraic theory of automata.* Academic Press, New York

Keville, T. J. [1978] 'The decomposition of transformation semigroups'. M.Sc. thesis, Queen's University, Belfast

Kieras, D. E. [1976] 'Finite automata and S-R models'. *J. Math. Psychol.* **13** 127–47

Krohn, K., Langer, R. and Rhodes, J. L. [1967] 'Algebraic principles for the analysis of a biochemical system'. *J. Comp. and System Sci.* **1** 119–36

Krohn, K. and Rhodes, J. L. [1965] 'Algebraic theory of machines, I. Prime decomposition theorem for finite semigroups and machines'. *Trans. Amer. Math. Soc.* **116** 450–64

Minsky, M. [1967] *Computation, finite and infinite machines.* Prentice-Hall, Englewood Cliffs, N.J.

Nelson, R. J. [1975] 'Behaviorism, finite automata and stimulus response theory'. *Theory and Decision* 6 249–67

Roedding, W. [1975] *Netzwerke endlicher Automaten als Modelle wirtshaftlicher und sozialer systeme.* Reports of the Austrian Society for Cybernetic Studies

Rosen, R. [1972] *Foundations of mathematical biology Vol. 2.* Academic Press, New York

Shibata, Y. [1972] 'On the structure of an automaton and automorphisms'. *Systems-Computers-Controls* 3 10–15

Suppes, P. [1969] 'Stimulus response theory of finite automata'. *J. Math. Psychol.* 6 327–55

Index of notation

		page
$\mathscr{C}_{(p,r)}$		35
$\mathscr{G}$	transformation group of G	35
$\mathscr{A}^c$	completion of $\mathscr{A}$	36
$(\alpha, \beta): \mathcal{M} \to \mathcal{M}'$	state machine homomorphism	36
$\mathcal{M} \cong \mathcal{M}'$	isomorphic state machines	37
$(f, g): \mathscr{A} \to \mathscr{A}'$	transformation semigroup homomorphism	37
$\mathscr{A} \cong \mathscr{A}'$	isomorphic transformation semigroups	38
$\mathcal{M}/\pi$	quotient state machine	40
$\mathscr{A}/(\pi)$	quotient transformation semigroup	40
$\mathcal{M} \leq \mathcal{M}'$	covering (state machines)	43
$\mathscr{A} \leq \mathscr{B}$	covering (transformation semigroups)	45
$S \mid T$	S divides T	46
$\hat{\mathcal{M}}$	Mealy machine	47
$\hat{\mathcal{M}} \wedge \hat{\mathcal{M}}'$	restricted direct product	49
$\hat{\mathcal{M}} \times \hat{\mathcal{M}}'$	(full) direct product	49
$\hat{\mathcal{M}} * \hat{\mathcal{M}}'$	general direct product	50
$\hat{\mathcal{M}} \omega \hat{\mathcal{M}}'$	cascade product	51
$\hat{\mathcal{M}} \circ \hat{\mathcal{M}}'$	wreath product	52
$\mathcal{M} \wedge \mathcal{M}'$	restricted direct product	52
$\mathcal{M} \times \mathcal{M}'$	(full) direct product	52
$\mathcal{M} \,\omega\, \mathcal{M}'$	cascade product	52
$\mathcal{M} \circ \mathcal{M}'$	wreath product	52
$\mathscr{A} \wedge \mathscr{A}'$	restricted direct product	53
$\mathscr{A} \times \mathscr{A}'$	(full) direct product	54
$\prod^{r} \mathscr{A}$	$\mathscr{A} \times \mathscr{A} \times \ldots \times \mathscr{A}$ (r times)	54
$\mathscr{A} \circ \mathscr{A}'$	wreath product	55
$\mathscr{A}^r$	$\mathscr{A} \circ \mathscr{A} \circ \ldots \circ \mathscr{A}$ (r times)	57
$\mathcal{M}\vert_P$	restriction of $\mathcal{M}$ to P	61
$\mathscr{A}\vert_P$	restriction of $\mathscr{A}$ to P	61
$[X]_\pi$	π-block containing X	80
$\max(\pi)$	maximum size of a π-block	82
$H^G = \bigcap_{g \in G} g^{-1} H g$		91
$\mathbf{Aut}_S(Q)$	set of automorphisms of Q	98
$\mathbf{C}(\mathscr{A})$	complexity of $\mathscr{A}$	106
$\mathbf{D}(\mathcal{M})$	dimension of $\mathcal{M}$	112
$\mathscr{A} \lhd_\alpha \mathscr{B}$	relational covering	116
$\mathbf{I}(\mathscr{A})$	skeleton	119
$h(A)$	height of A	121
$\mathbf{M}(Q)$	maximal image space of Q	122
$\mathbf{M}(A)$	maximal image space of A	123
$\mathbf{G}(A)$		123
$\mathscr{H}(A)$	holonomy transformation group of A	124
$\mathbf{H}(A)$	holonomy group of A	124
$\pi^n > \pi^{n-1} > \ldots$	derived sequence	126
$\mathscr{H}_i^{\vee}(\mathscr{A})$		126
$\mathscr{H}_i^{\times}(\mathscr{A})$		130
$\mathfrak{M}$	recognizer	145
$\vert\mathfrak{M}\vert$	behaviour of $\mathfrak{M}$	146
$\mathfrak{M}^c$	completion of $\mathfrak{M}$	148
$\mathfrak{M}^a$	accessible part of $\mathfrak{M}$	149
$\mathfrak{M}^b$	coaccessible part of $\mathfrak{M}$	149
$q * \alpha$		150
$q * \alpha^{-1}$		151
$a \cdot b^{-1}$		151

		page
$a^{-1} \cdot b$		151
$q^{-1} \circ R$		151
$\mathcal{M}_A$		152
$\mathfrak{M}_A$	minimal recognizer	152
$\sim_{\mathcal{M}}$		159
$\approx_A$	Myhill congruence	160
$\mathrm{Rat}(\Sigma)$	set of rational sets	162
$\mathrm{Reg}(\Sigma)$	set of regular sets	163
A		170
A_P	prefix part of A	170
$A^{(n)}$	set of words of A of length n	173
f_q		179
□	blank	180
$\hat{\mathcal{M}}^c$	state completion of $\hat{\mathcal{M}}$	181
$*$	covers (words)	182
$\Vert$	compatible (words)	182
$\Vert$	compatible (states)	182
$\hat{\mathcal{M}}/\sim$	minimal machine of $\hat{\mathcal{M}}$	183
$\sim_i$		184
$\Vert_i$		186
$A(q)$		186
$\mathscr{X}$		186
$\hat{\mathcal{M}}/\pi$		187
$\hat{\mathcal{M}} \leq \hat{\mathcal{M}}'$	Mealy machine covering	191
π^c		192
τ^*		192
$\hat{\mathcal{M}} \ll \hat{\mathcal{M}}'$	strong covering	198
$\bar{f}_q$		202
$\hat{\mathcal{M}}_f$		205
$(\hat{\mathcal{M}}_{f,q})$	minimal sequential machine for f	205
$\mathscr{A}_f$		209

Index