10

Transformation of Clock Rates of Sequential Machines

10.1. GENERAL CONSIDERATIONS REGARDING CLOCK RATE TRANSFORMATION. DEFINITION OF REPRESENTATION AND REPRODUCTION

In discussing various practical embodiments of finite automata and sequential machines in Chapter 5, we have singled out a design method whereby an s-machine with a desired clock rate is created on the basis of the equilibrium states of another s-machine, operating at a much faster rate. We shall now return to this problem, and shall analyze it in more general terms. First, however, we shall recall some concepts and definitions of Chapter 5.

Assume we have an s-machine S, to which we feed (at discrete moments $0, 1, \ldots p$) a sequence $\rho^0\ \rho^1 \ldots \rho^p$. We thus obtain the tape of Table 10.1:

Table 10.1

Discrete moment	0	1	2	...	p	...
ρ	ρ^0	ρ^1	ρ^2	...	ρ^p	...
λ	λ^0	λ^1	λ^2	...	λ^p	...

Now we select some sequence of discrete moments, for example, moments 0, 1, 4, and so on, which lie on a continuous scale such that

$$t_0 < t_1 < t_2 < \ldots < t_s. \tag{10.1}$$

We then extract from the tape of Table 10.1 the columns corresponding to this sequence. We thus get Table 10.2:

Table 10.2

t	t_0	t_1	...	t_s	...
ρ	ρ^{t_0}	ρ^{t_1}	...	ρ^{t_s}	...
λ	λ^{t_0}	λ^{t_1}	...	λ^{t_s}	...

We now introduce another clock rate to match our selected sequence of discrete moments, assuming that moment 0 of that sequence occurs at time t_0, moment 1—at time t_1, and so on. We then rewrite Table 10.2 in terms of this new clock rate, and obtain Table 10.3:

Table 10.3

Discrete moment	0	1	...	s	...
ρ	ρ^{t_0}	ρ^{t_1}	...	ρ^{t_s}	...
λ	λ^{t_0}	λ^{t_1}	...	λ^{t_s}	...

The tape of Table 10.3 may be regarded as produced by some new machine G. In fact, if the given tape of machine S (Table 10.1) and, therefore, the tape obtained from it by clock rate transformation (Table 10.3) are both finite, then there must exist an s-machine G producing that last tape (see Section 8.2).

To illustrate this concept, imagine an s-machine whose tape is flash-illuminated at times $t_0, t_1, \ldots, t_s$, corresponding to the sequence of discrete moments of our second clock rate. Machine S will then appear to us to be processing the sequence $\rho^{t_0}, \rho^{t_1}, \ldots, \rho^{t_s}$ into the sequence $\lambda^{t_0}, \lambda^{t_1}, \ldots, \lambda^{t_s}$ in accordance with Table 10.3, whereas in reality it is operating in accordance with Table 10.1, processing the sequence $\rho^0, \rho^1, \ldots, \rho^P$ into the sequence $\lambda^0, \lambda^1, \ldots, \lambda^P$.

Let us now assume that the sequence of times $t_0, t_1, \ldots$, at which the flashes illuminate tape S, is so fortuitously chosen that whatever the input sequence processed by S and whatever its initial state $\varkappa_S^0$, we shall always perceive a sequence of input-output pairs that could be attributed to some s-machine G, which starts up from some state $\varkappa_G^0$ (whereby $\varkappa_G^0$ may vary with each input sequence). If that is the case, we have a *clock rate transformation*—machine S, operating at a rate which, by convention, we shall call *fast*, serves as a basis for another machine G operating at clock rate which we shall call slow.* We shall also say that the fast machine S *represents* the slow machine G.

These concepts are quite broad, but have a drawback. The point is that the initial state $\varkappa_G^0$ of G is governed not only by the initial state $\varkappa_S^0$ of S, but also by the input sequence $\rho(t)$. This means that at different $\rho(t)$, there will be different $\varkappa_S^0$ for the same $\varkappa_G^0$. Thus to find the appropriate state $\varkappa_G^0$ of G one must not only know beforehand the state $\varkappa_S^0$ of S, but also the input sequence which will be fed into S. This is not unlike the situation encountered in Chapter 9 in connection with the definition of equivalence of s-machines. There the problem was solved by narrowing the concepts of equivalence in such a way that the choice of the initial state did not require an *a priori* knowledge of the input sequence. However, the present authors' attempt to similarly narrow the definitions of representation and transformation of clock rate was unsuccessful. This is because a rigid adherence to a scheme whereby any state $\varkappa_S^0$ of S would always correspond to the same $\varkappa_G^0$ of G, regardless of the input sequence, would have prevented us from investigating several important practical cases of clock rate transformation (we shall return to this question at a later stage and shall then clarify this statement by an example). We shall, therefore, resort to other definitions which are narrower than those above and do not require an *a priori* knowledge of the entire input sequence in order to determine the initial state of the represented machine.

The algorithm for selecting the appropriate time sequence $t_0, t_1, t_2, \ldots$ synchronizing S and G, will be called the ***rule of clock rate transformation.*** We shall define it by saying that *the fast machine S represents the slow machine G if for any initial state $\varkappa_S^0$ of S and any input sequence $\rho^0\rho^1\rho^2 \ldots$ there exists at least one initial state $\varkappa_G^0$ of G such that G, starting from this state and processing a sequence $\rho^{t_0}\rho^{t_1}\rho^{t_2} \ldots$, will generate a tape coinciding with the image obtained by viewing the tape of S at times $t_0, t_1, t_2, \ldots$.*

Given this definition of representation, *state $\varkappa_G^0$ of G is determined by the state $\varkappa_S^0$ of S and the first term of the input sequence to S.*

Note now that the fast machine S, which admits any arbitrary input sequence, usually represents a machine G which can admit inputs only from a restricted set L_G. This means that an image of

*It is convenient, but not necessary, to imagine that the fast machine does indeed operate at a faster clock rate than the slow machine. In general, however, S and G are totally unrelated. Our further discussion shall deal with the general case, in which S and G may operate at any desired clock rates.

the tape of S, obtained by viewing it at $t_0, t_1, t_2, \ldots$, may represent only one of the several possible variants of operation of G. We shall encounter a case of this kind in Section 10.2, where set L_G will consist of sequences containing only one symbol. In general, representation is not a unique relationship, because at any specific clock rate transformation, a given machine S may represent several different machines $G_1, G_2, G_3, \ldots$. This conclusion also holds for the case where there is no restriction on the set of input sequences of G, that is, when $L_G = E$.

By analogy with relative equivalence (see Chapter 9), we can also define relative representation. The definition of relative representation differs from that of representation in general only in that *the fast machine S may not admit arbitrary input sequences but only those belonging to set L_S of sequences allowed in S*. We shall say that in this case *machine S represents machine G in terms of set L_S*.

When $L_S \subseteq E$, then L_G may coincide with L_S, be narrower or broader, intersect with it, etc. In particular, when $L_S = E$, L_G may be restricted, and, conversely, it can happen that $L_S \subset E$ and $L_G = E$.

It is quite obvious that the mode of representation by S of any machine G is closely related to the time sequence $t_0, t_1, t_2, \ldots$ at which the tape of S is viewed. In the general case, this time sequence may be such that S does not represent any sequential machine.

The choice of the (viewing) times $t_0, t_1, t_2, \ldots$ may depend on the input sequence $\rho(t)$, the output sequence $\lambda(t)$, the sequence $\varkappa(t)$ of the states of machine S, as well as the time t.

The "clock," which is a machine that signals the advent of the "slow" discrete (viewing) moments $t_0, t_1, t_2, \ldots$, must allow the input of time t and the symbols $\rho(t)$, $\lambda(t)$, and $\varkappa(t)$ [or some of these symbols], all of which are related to the operation of the fast machine S. The "clock" must be able to perform an algorithm* which processes a given sequence of symbols of the s-machine into the sequence $t_0, t_1, t_2, \ldots$.

We shall assume that the clock itself is a finite automaton with an output converter which operates at the same fast clock rate as the s-machine S. The alphabet of this automaton is obtained by combining all or part of alphabets $\{\rho\}$, $\{\varkappa\}$, and $\{\lambda\}$, depending on the factors determining the sequence $t_0, t_1, t_2, \ldots$. The process of producing a synchronizing signal indicating the advent of a discrete moment such as $t_0, t_1, \ldots$, can then be regarded as a representation of an event at the input of this clock automaton.

*That is, it is a Turing machine (see Chapter 13).

Having defined representation (or relative representation), we are faced with the following problems:

1. Given a machine S, a set L_S, and a clock (that is, an automaton A with a converter Φ), find at least one machine G which can be represented by S in terms of L_S, as well as its set of allowed input sequences L_G.

2. Given a machine S and a machine G, find whether there exists a clock rate transformation such that S will represent G, and if so, determine it (construct automaton A and converter Φ of the clock).

A similar problem also arises with respect to relative representation (here, the set L_G must also be determined).

3. Given a machine S, a set L_S, and the clock rate transformation, construct a minimal machine $G_{\min}$, represented by S in terms of L_S, and find its set of allowed input sequences $L_{G_{\min}}$.

No general solutions to these problems exist as of now, and it is possible that some of them will prove to be algorithmically unsolvable.

In conclusion of our discussion of representation and clock rate transformation, let us note that these concepts could be broadened by permitting the use of converters $\rho^* = \Phi_1(\rho)$ and $\lambda^* = \Phi_2(\lambda)$, in accordance with Fig. 10.1. In this scheme, the input-output pairs occurring at $t_0, t_1, t_2, \ldots$ are not (ρ, λ), but (ρ^*, λ^*). However, we do not need this broader definition for our discussion.

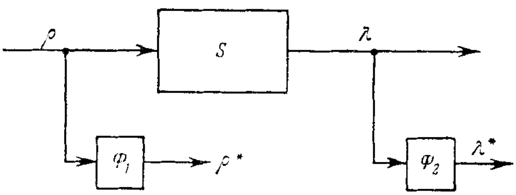


Fig. 10.1.

Assume now that we are given a specific clock rate transformation, a slow machine G and a fast machine S, and the set L_G of G (since G and L_G are given, we can determine the output of G at any input from L_G and at any initial state $\varkappa_G^0$). Now let us define *relative reproduction*. We shall say that S *reproduces* G *in terms of set* L_G *if for any* $\varkappa_G^0$ *of* G *and at any input sequence* $\rho^*(t) = \rho^0\rho^1\rho^2 \ldots$, *from* L_G *there exist at least one initial state* $\varkappa_S^0$ *of* S, *determined by* $\varkappa_G^0$ *and by the first term* ρ^0 *of* $\rho^*(t)$, *as well as at least one sequence* $\tilde{\rho}(t)$ *such that the tape of* S, *which operates under these conditions and is viewed only during the discrete moments of the "slow" time sequence* $t_0, t_1, t_2, \ldots$, *coincides with the tape of* G. When $L_G = E$, relative reproduction and (simple) reproduction become identical.

To avoid confusion, we must stress that representation and reproduction (both relative and nonrelative) are two entirely different and even opposing concepts. Thus representation requires coincidence between each tape of the fast machine S, when viewed at $t_0, t_1, t_2, \ldots,$ and one of the tapes of the slow machine G; on the other hand, reproduction requires that for each tape of G there be a tape of the fast machine S such that when it is viewed at $t_0, t_1, t_2, \ldots,$ it will coincide with the given tape of G. Representation does not imply reproduction, because S may represent G but not reproduce it, and vice versa. Again, reproduction is not unique: for any given specific clock rate transformation there may exist many different fast machines $S_1, S_2, S_3, \ldots,$ each of which will reproduce a given slow machine G.

The set L_S allowed in a fast machine S reproducing a given machine G in terms of L_G,* is also not unique;** and what is more, in many cases L_S may contain symbols which do not appear in L_G.

Indeed, according to the definition of reproduction, for each sequence $\rho^*(t) \in L_G$ and state $\varkappa_G^0$ there will be at least one corresponding sequence $\tilde{\rho}(t)$ allowed as an input to S. However, in the general case, for each $\rho^*(t)$ and $\varkappa_G^0$ there may be not one, but many (possibly even an infinite number) of different sequences $\tilde{\rho}(t)$. These sequences $\tilde{\rho}(t)$, corresponding to all the $\rho^*(t)$ at all possible $\varkappa_G^0$, may form many (possibly even an infinite number) of different sets L_S^1, L_S^2, $L_S^3, \ldots,$ each of the sets L_S^i containing at least one sequence $\tilde{\rho}(t)$ corresponding to any given $\rho^*(t) \in L_G$ and $\varkappa_G^0$.

Each of the sets L_S^i may be considered as a set of inputs allowed in the fast machine S which reproduces G in terms of L_G. Which of these sets is selected depends on additional, practical considerations—sometimes it is convenient to use $L_S = \bigcup\limits_i L_S^i$, and on other occasions set E (which is always usable) is selected as the set of inputs allowed in S.

The concept of reproduction gives rise to the same problems as representation [(1) given a clock and one of the machines, find the other machine; (2) given the two machines, find the clock; and (3) the minimization problem].

In conclusion let us point out that the definition of reproduction entails the same restriction as that of representation: that is, the state $\varkappa_S^0$ of machine S is determined by the state $\varkappa_G^0$ of machine G and only the first term of input $\rho^*(t)$. If $\varkappa_S^0$ were related to $\varkappa_G^0$ and the entire input sequence, we should obtain a broader, but also a

*L_G may coincide with E, that is, the reproduction may be nonrelative.

**This nonuniqueness is not encountered in the case of representation where, given L_S, a specific clock rate transformation, and machine S, the set L_G is uniquely determined.

more inconvenient definition of reproduction [we would have to know, *a priori*, the entire input sequence $\rho^*(t)$].

We shall now clarify representation and reproduction by two simple examples.

10.2. EXAMPLES OF REPRESENTATION AND REPRODUCTION

a. Flip-Flop

Our first example involves the flip-flop of Chapter 5.

Table 10.4

$x \backslash \rho$	$\rho_1 = 0$	$\rho_2 = 1$
x_1	x_3	—
x_2	—	x_1
x_3	—	x_4
x_4	x_2	—

Table 10.5

$x \backslash \rho$	$\rho_1 = 0$	$\rho_2 = 1$
x_1	x_3	x_4
x_2	x_3	x_1
x_3	x_2	x_4
x_4	x_2	x_1

Consider a P-Pr automaton with a basic Table 10.4. It accepts inputs from set R, containing all sequences in which no two successive symbols are identical. The blanks in the table indicate that the automaton is never in these internal states, so that the corresponding squares may be filled in any desired fashion, for instance, as in Table 10.5. This automaton is diagrammed in Fig. 10.2. We shall observe it only at times $t_0, t_1, t_2 \ldots$, when the state of the input changes from $\rho_1 = 0$ to $\rho_2 = 1$, and shall find which machine G is represented by this automaton in terms of R.

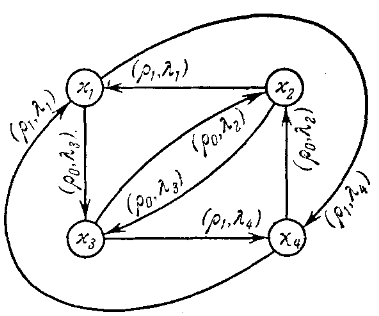


Fig. 10.2.

First of all, let us determine the set L_G of G. Since we observe this automaton only when $\rho = \rho_2 = 1$, L_G will contain only unit sequences. In other words, G will be an autonomous s-machine.

An analysis of operation of the automaton of Fig. 10.2 shows that the s-machine G (or, to be precise, one of the many possible machines G), which the automaton represents has the state diagram of Fig. 10.3, that is, the slow machine G is a flip-flop (see Section 5.2). The relationship between the various states automaton S and flip-flop G is given in Table 10.6.

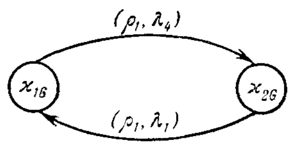


Fig. 10.3.

Now, if the blank squares of Table 10.4 were filled differently, for example, as in Table 10.7, then the relationship between the states of S and G would be represented by Table 10.8.

Table 10.6

S	G	Type of input
x_1	x_{1G}	independent of input sequence
x_2	x_{1G}	if the input sequence begins with $\rho_1 = 0$
x_2	x_{2G}	if the input sequence begins with $\rho_2 = 1$
x_3	x_{2G}	if the input sequence begins with $\rho_1 = 0$
x_3	x_{1G}	if the input sequence begins with $\rho_2 = 1$
x_4	x_{2G}	independent of input sequence

Table 10.7

$x \backslash \rho$	$\rho_1 = 0$	$\rho_2 = 1$
x_1	x_3	x_4
x_2	x_2	x_1
x_3	x_3	x_4
x_4	x_2	x_1

Table 10.8

S	G	Type of input
x_1	x_{1G}	independent of input sequence
x_2	x_{2G}	independent of input sequence
x_3	x_{1G}	independent of input sequence
x_4	x_{2G}	independent of input sequence

In this example the clock is a finite automaton which represents the event "ρ_2 occurs after ρ_1." Note that S not only *represents* G with respect to R, but also *reproduces* it in terms of set L_G of unit length input sequences. To achieve reproduction the set L_S of sequences allowed in the automaton can coincide either with E (which contains all sequences of 0 and 1), or with set R, or with any set containing R.

b. Delay Line

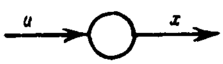
Our second example involves reproduction of a slow s-machine G by a fast machine S built from fast delay elements. Assume we require an s-machine G in which the interval between discrete moments (clock rate) is τ seconds, and where the set L_G coincides with the set E of all allowed input sequences. The input alphabet of G contains r differing characters ρ.

There is no problem in synthesizing a machine G from n-ary delay elements operating in given clock rate τ. However, we have only "fast" n-ary delay elements operating at rate q times faster than τ, that is, at intervals of $\frac{\tau}{q}$, where $q \geqslant 2$. We shall now use these elements to synthesize a fast machine S, and we shall find a clock rate transformation such that S will reproduce G.

The equation for the fast delay element (Fig. 10.4) is

$$x\left(t+\frac{\tau}{q}\right)=u(t), \tag{10.2}$$

while that for the required slow element is



$$x(t+\tau)=u(t). \tag{10.3}$$

Fig. 10.4.

Now, a chain of q fast delay elements (Fig. 10.5) is described by a system of recurrence relations

$$\left.\begin{aligned} x\left(t+\frac{\tau}{q}\right) &= x_1(t), \\ x_1\left(t+\frac{\tau}{q}\right) &= x_2(t), \\ \cdots&\cdots\cdots \\ x_{q-1}\left(t+\frac{\tau}{q}\right) &= u(t). \end{aligned}\right\} \tag{10.4}$$

Eliminating all x_i except x, we get for the entire chain

$$x(t+\tau)=u(t). \tag{10.5}$$

Equation (10.5) coincides with Eq. (10.3) for the slow delay element; therefore, a chain of q fast elements is equivalent to one slow element.

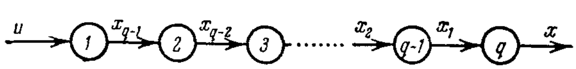


Fig. 10.5.

Bearing this in mind, we construct the s-machine S in the following manner. Assuming for a moment that we have at our disposal the slow delay elements of Eq. (10.3), we construct from these elements and from instantaneous logical converters a machine G, using one of the methods of Chapter 5. Now, we replace each slow element of G by a chain of q fast delays. The resulting s-machine S will operate at a fast clock rate, that of the fast delay elements. But if S is observed only at $t_0, t_1, t_2, \ldots,$ coinciding with moments $0, \tau, 2\tau, 3\tau,$ then S will reproduce G, since the *cycle* of a fast delay chain coincides with that of one slow element.

The relationship between the states of S and G is independent of the input: for each state $\varkappa_{iG}$ of G there exists such a state of all the fast delays of S at which the state of the initial fast delays of each chain coincides with that of the corresponding slow delays. Calculations show that for each of the r^k states of G there are $r^{k(q-1)}$ states of S reproducing it. The set L_G of S may be either set E, or set R (in which there are no sequences with repeating symbols), or set M which contains all sequences such as

$$\underbrace{\rho_{\alpha_0}\rho_{\alpha_0}\cdots\rho_{\alpha_0}}_{q\text{ times}}\;\underbrace{\rho_{\alpha_1}\rho_{\alpha_1}\cdots\rho_{\alpha_1}}_{q\text{ times}}\;\underbrace{\rho_{\alpha_2}\rho_{\alpha_2}\cdots\rho_{\alpha_2}}_{q\text{ times}}\cdots\underbrace{\rho_{\alpha_S}\rho_{\alpha_S}\cdots\rho_{\alpha_S}}_{q\text{ times}}$$

$$\left(\text{possible } \rho_{\alpha_i}=\rho_{\alpha_{i+1}}\right),$$

or another of the many possible sets (all these sets must have the following property: if the symbols in positions $0, q, 2q, 3q, \ldots$ are extracted from each sequence belonging to a given set and arranged into a new set, then this new set must be the set E).

The clock suitable for this case is a ring scaler (which is an autonomous finite automaton) made up of fast delay elements and emitting a signal indicating the occurrence of a "slow" discrete moment every q "fast" moments.

It may be pointed out that in this example (just as in the preceding one) the synthesis was so successful that machine S not only reproduces, but also represents machine G (both in terms of E and in terms of any other set). The relationship between the states of the two machines in the case of representation remains the same as in the case of reproduction.

10.3 REPRODUCTION OF A SLOW MACHINE ON A FAST ONE IN THE CASE WHEN THE CYCLE OF THE SLOW MACHINE IS GOVERNED BY THE CHANGE OF INPUT STATE

This problem mentioned was already discussed in Chapter 5, where we arrived at a solution. We shall produce here another solution which is suitable for any machine.

Suppose we are given some slow s-machine G whose cycle (that is, clock rate) is governed by change of input. Let this G be given as an interconnection matrix or a state diagram. The set L_G of G contains all the possible sequences except those with two identical symbols in a row. We need a fast s-machine S which reproduces G in terms of L_G, and whose clock rate is related to that of G in the following manner: G operates at instants $t_0, t_1, t_2, \ldots, t_s$ which occur when S reaches equilibrium after any change of input. Assume that the maximum number of fast cycles necessary for S to go from one equilibrium state to another upon a change of input is m. We shall then assume that, at reproduction, the set L_G of S contains all the sequences such as

$$\underbrace{\rho_{\alpha_0}\rho_{\alpha_0}\cdots\rho_{\alpha_0}}_{q_0 \text{ times}}\ \underbrace{\rho_{\alpha_1}\rho_{\alpha_1}\cdots\rho_{\alpha_1}}_{q_1 \text{ times}}\cdots\ \underbrace{\rho_{\alpha_i}\rho_{\alpha_i}\cdots\rho_{\alpha_i}}_{q_i \text{ times}}\ (\rho_{\alpha_i}\neq\rho_{\alpha_{i+1}}), \tag{10.6}$$

where $q_i \geqslant m$ for all $i = 1, 2\ 3, \ldots$. This means that an input to S cannot change until the machine is in equilibrium.

Assuming that $q_i \geqslant q^*$, the set of sequences such as (10.6), will be denoted by T_{q^*}. The sets T_{q^*} satisfy the relationship

$$T_1 \supset T_2 \supset T_3 \supset \ldots \supset T_{q^*} \supset \ldots, \tag{10.7}$$

whereby $T_1 = E$. Thus, provided $q^* \geqslant m$, any set T_q^* can serve as the set L_S of S.

If the condition of replacement of G by S specifies that the two machines must operate synchronously, then condition $q^* \geqslant m$ means that there are at least m cycles of S between two successive cycles of G.

We shall construct machine S by transforming the state diagram of the given machine G. Assume that state $\varkappa_i$ of this diagram has the form of Fig. 10.6. We shall replace $\varkappa_i$ with as many states as there are different paths terminating in that state (a loop path is considered to be both terminating and originating in state $\varkappa_i$). This gives the four states $\varkappa_{i1}, \varkappa_{i2}, \varkappa_{i3}, \varkappa_{i4}$ (surrounded by a dotted line) of Fig. 10.7, where each of these new states also carries a loop path labeled in the same way as the path terminating in that state.

From each of the new states we draw the same paths as those which originated in state $\varkappa_i$ of Fig. 10.6; however, we need not draw those paths carrying the same ρ symbol as the loop at the state from which that path would originate (see Fig. 10.7).

We do the same thing for all states of G, and obtain the state diagram of a machine S which reproduces G in terms of L_G, the relationships between the states of G and S being as follows (Figs. 10.6

and 10.7): at $\rho = \rho_s$, state $\varkappa_{j1}$ of G corresponds to state $\varkappa_{i1}$ of S, where $\varkappa_{i1}$ is the equilibrium state of S at input ρ_s; obviously, at input ρ_s the same state $\varkappa_{i1}$ of S also corresponds to state $\varkappa_i$ of G; similarly, at $\rho = \rho_p$ the state $\varkappa_{j2}$ of G corresponds to state $\varkappa_{i2}$ of S, $\varkappa_{i2}$ being the state of equilibrium of S at $\rho = \rho_p$, and so on. The general correspondence is established in a similar manner. Now it is readily seen from the state diagram that S goes from one state of equilibrium to another at any change of the input and that this transition is accomplished in one "fast" cycle. That is, $m = 1$. In addition, the diagram shows that machine S has no unstable states at any input, because each state has a loop path and therefore is a state of equilibrium for some specific input.

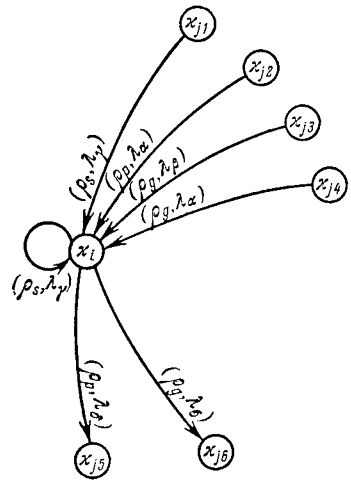


Fig. 10.6.

An input sequence of the reproducing machine S, corresponding to the input sequence $\rho_{\alpha_0}\rho_{\alpha_1}\rho_{\alpha_2} \cdots \rho_{\alpha_i} \cdots (\rho_{\alpha_i} \neq \rho_{\alpha_{i+1}})$ of the slow machine G, has the form of Eq. (10.6), where all the $q_i \geqslant 1$. Using $q = 1$, we find that one of the (corresponding) input sequences of S is $\rho_{\alpha_0}\rho_{\alpha_1}\rho_{\alpha_2} \cdots$, that is, it coincides with the input sequence to G.

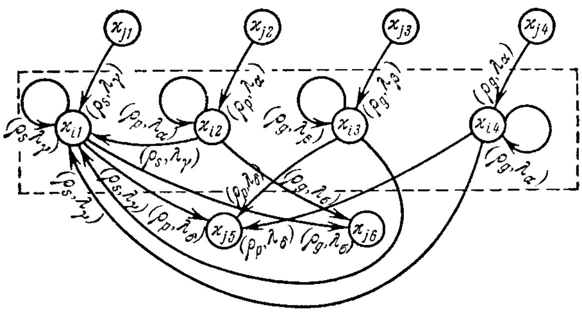


Fig. 10.7.

The instants t_0, t_1, t_2, ..., at which the fast machine S is "viewed" (that is, at which information is extracted from the tape of S) occur one "fast" cycle after the change of the state of input of S. If the input sequence of S is made to coincide with that of G, then information is extracted from S in all fast cycles.

Thus, our transformation of the state diagram of G solves the problem of synthesis of a fast s-machine S which reproduces, in terms of L_G, a given s-machine G responding to the change of the input state. Here we used the state diagram only for better visualization of the problem. In practice, it may be more convenient to transform the interconnection matrix. This is done as follows:

Assume the interconnection matrix C^G of G is given, and let it conform to Fig. 10.6, so that its ith row and column are

$$C^G = \begin{array}{cc} & \begin{array}{ccccccc} \cdots & i & \cdots & j5 & \cdots & j6 & \cdots \end{array} \\ \begin{array}{c} \vdots \\ j1 \\ j2 \\ \vdots \\ i \\ \vdots \\ j3 \\ \vdots \\ j4 \\ \vdots \end{array} & \begin{bmatrix} & \cdots & & & & & \\ & (\rho_s, \lambda_\gamma) & & & & & \\ & (\rho_p, \lambda_\alpha) & & & & & \\ & \cdots & & & & & \\ \cdots & (\rho_s, \lambda_\gamma) & \cdots & (\rho_p, \lambda_\delta) & \cdots & (\rho_g, \lambda_\sigma) & \cdots \\ & \cdots & & & & & \\ & (\rho_g, \lambda_\beta) & & & & & \\ & \cdots & & & & & \\ & (\rho_g, \lambda_\alpha) & & & & & \\ & \cdots & & & & & \end{bmatrix} \end{array}.$$

We shall now replace this row and column with as many rows and columns as there are different pairs in the ith column of C^G. Since there are four such pairs $[(\rho_s, \lambda_\gamma), (\rho_p, \lambda_\alpha), (\rho_g, \lambda_\beta), \text{and} (\rho_g, \lambda_\alpha)]$, we introduce the rows and columns $i1$, $i2$, $i3$ and $i4$:

$$C = \begin{array}{cc} & \begin{array}{ccccccccc} & i1 & i2 & i3 & i4 & \cdots & j5 & \cdots & j6 \end{array} \\ \begin{array}{c} j1 \\ \vdots \\ j2 \\ \vdots \\ i1 \\ i2 \\ i3 \\ i4 \\ \vdots \\ j3 \\ \vdots \\ j4 \\ \vdots \end{array} & \begin{bmatrix} \cdots & (\rho_s, \lambda_\gamma) & & & & & & & \\ & & & & & & & & \\ & & (\rho_p, \lambda_\alpha) & & & & & & \\ & & & & & & & & \\ \cdots & (\rho_s, \lambda_\gamma) & & & & & (\rho_p, \lambda_\delta) & & (\rho_g, \lambda_\sigma) \\ \cdots & (\rho_s, \lambda_\gamma) & (\rho_p, \lambda_\alpha) & & & & & & (\rho_g, \lambda_\sigma) \\ \cdots & (\rho_s, \lambda_\gamma) & & (\rho_g, \lambda_\beta) & & & (\rho_p, \lambda_\delta) & & \\ \cdots & (\rho_s, \lambda_\gamma) & & & (\rho_g, \lambda_\alpha) & & (\rho_p, \lambda_\delta) & & \\ & & & & & & & & \\ & & & (\rho_g, \lambda_\beta) & & & & & \\ & & & & & & & & \\ & & & & (\rho_g, \lambda_\alpha) & & & & \\ & & & & & & & & \end{bmatrix} \end{array}$$

For diagonal elements of the submatrix comprising these four rows and columns we take (in any desired order) all the different pairs of the ith column of C^G. Furthermore, the intersection of row $j1$ and column i of C^G contains the pair (ρ_s, λ_γ); we retain it in row $j1$ of the new matrix C, but place it in the column where (ρ_s, λ_γ) is already present, that is, column $i1$. We do the same with the pairs of rows $j2$, $j3$ and $j4$ of the ith column of C^G. Now the columns $i1$—$i4$ are complete, and each contain only identical pairs. We then fill in rows $i1$—$i4$ of C as follows: all the pairs of the ith row of C^G [except for pair (ρ_s, λ_γ) which is present on the diagonal of C^G] are transposed into all four rows ($i1$—$i4$), retaining these pairs in the same columns as in C^G. However, if the ρ symbol of the pair being transposed into a given row coincides with the ρ symbol of a pair already present in that row, then there is no need for this transposition—the space is left blank. The pair (ρ_s, λ_γ) is transposed into all rows $i1$—$i4$, being placed in that column in which it already appears as a result of filling in the disposal elements. To be specific, the ith row of our example of C^G contains pairs (ρ_s, λ_γ), (ρ_p, λ_δ), (ρ_g, λ_σ). Row $i1$ of C already contains pair (ρ_s, λ_γ); we transpose into it pair (ρ_p, λ_δ) and place it in column $j5$, and the pair (ρ_g, λ_σ) in column $j6$. We add to the row $i2$ the pair (ρ_s, λ_γ) in column $i1$ and the pair (ρ_g, λ_σ) in the column $j6$ [pair (ρ_p, λ_δ) is omitted from column $j5$ since row $i2$ already contains pair (ρ_p, λ_α)]. Rows $i3$ and $i4$ are filled in the same manner.

The above procedure must be repeated for all the ith rows and columns of C^G. As a result, we obtain a matrix C which actually is the interconnection matrix C^S of the fast machine S. Note one property of C^S: *all columns of this matrix contain (only) identical pairs.*

It is obvious that the transformation of the interconnection matrix C^G into C^S is a procedure identical to that employed in the previously described transformation of state diagrams.

Let us conclude this section with two notes.

Note 1. In this section, just as in Section 10.2, we devised the machine S so "successfully" that it not only reproduces but also represents machine G. In representation, the set of allowed input sequences may be any set, including set E, since the inputs to S can change at a rate coinciding with the clock rate of the fast machine S. This is because m of S is 1. In representation, the correspondences between the states of G and S are the same as in reproduction.

Note 2. The above technique is only one of the available methods for synthesizing a fast s-machine S, reproducing a given s-machine G in terms of L_G. Other techniques are also possible, since S is not the only machine reproducing G with respect to L_G. For this reason there arises the problem of minimization of S, that is, the problem of synthesizing the machine S in such a way that it will contain a minimal number of states.

10.4. MINIMIZATION OF THE s-MACHINE OF SECTION 10.3

We shall minimize the machine of Section 10.3, that is, synthesize a machine $S_{\min}$ reproducing the given machine G in terms of L_G, but having the least possible number of internal states. The required machine $S_{\min}$ will have to satisfy two conditions.

Condition 1. Each state of $S_{\min}$ must be an equilibrium state for at least one input.

Condition 2. Regardless of what changes are made at the input, $S_{\min}$ must reach a new equilibrium in one fast cycle (that is, $m_{S_{\min}} = 1$).

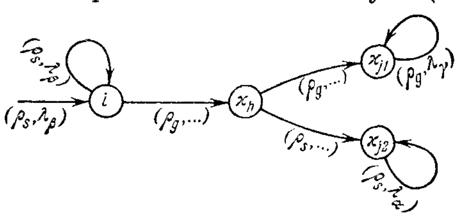


Fig. 10.8.

We shall now prove that these conditions do not restrict the generality of our minimization. Assume that machine $S_{\min}$ does not satisfy condition 1. This would mean that $S_{\min}$ has at least one state x_h which is not a state of equilibrium, and is represented in the state diagram by a circle h not associated with a loop path (Fig. 10.8). If this is so, we can drop this state x_h from the diagram, replacing the path labeled $(\rho_g, \ldots)$* from x_i to x_{j1} (and passing through x_h) by a direct path $(\rho_g, \ldots)$ from x_i to x_{j1}. The path $(\rho_s, \ldots)$ from x_h to x_{j2}, may be dropped because no path with the same label terminates in x_h. We thus obtain Fig. 10.9, from which all the nonequilibrium states have been removed.

This transformation modifies the operation $S_{\min}$ only during the interval between two equilibria of $S_{\min}$, when we do not care what the machine does anyway. The order in which the equilibrium states change remains unaltered and consequently the modified machine will reproduce the given machine G in the same way as before. However, the very fact that we are able to reduce the number of states in $S_{\min}$ by this transformation contradicts our statement that $S_{\min}$ is a minimal machine. Therefore a minimal machine $S_{\min}$ must satisfy condition 1.

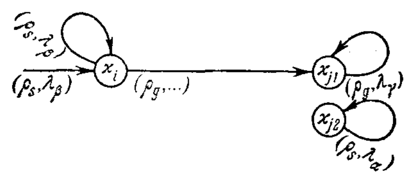


Fig. 10.9.

*Dots in the label indicate that the output symbol can be arbitrary.

In general, for any given machine G there may exist several different minimal machines $S_{\min}$, each reproducing G in terms of L_G. However, all these machines must have the same number of states $k_{\min}$. Our minization problem will be solved when we shall find at least one of these machines.

Let us now turn to condition 2. We shall prove the following statement: if there exists a machine $S_{\min}$, with $k_{\min}$ internal states, reproducing G in terms L_G and not satisfying condition 2, then there must exist another machine $\tilde{S}_{\min}$ with the same number of internal states $k_{\min}$, which also reproduces G in terms of L_G but which satisfies condition 2. This will show that condition 2 does not restrict the generality of the solution. To prove this statement we shall show that the state diagram of $\tilde{S}_{\min}$ can be obtained from that of $S_{\min}$ by a transformation which does not alter the number of states.

Let $S_{\min}$ go from state x_i, which is an equilibrium for $\rho = \rho_p$, to state x_j which is an equilibrium for $\rho = \rho_g$. Let this be accomplished in m fast cycles. Then $S_{\min}$ will go through $(m - 1)$ intermediate states $x_{i1}, x_{i2}, \ldots, x_{i(m-1)}$ (because none of these is an equilibrium state, they cannot contain closed loops). For example, Fig. 10.10 illustrates what happens in a section of some machine at $m = 3$.

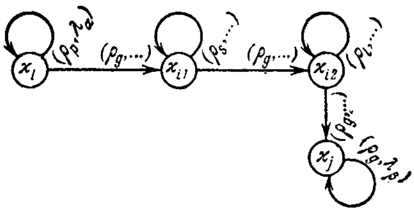


Fig. 10.10.

If the circles i, $i1$, $i2$, ..., $i(m-1)$ are directly connected to circle j by paths labeled (ρ_g, λ_β), then we obtain the state diagram of Fig. 10.11. Now ρ_g shifts the machine from state x_i and from states $x_{i1}, x_{i2}, \ldots, x_{i(m-1)}$ to the state x_j in one fast cycle. If we transform

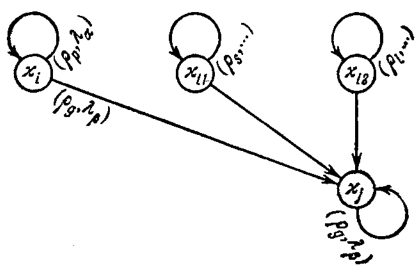


Fig. 10.11.

the entire state diagram of $S_{\min}$ in the same manner, then the state diagram of the resulting machine $\tilde{S}_{\min}$ will have the same number of states. The operation of $S_{\min}$ will differ from that of $\tilde{S}_{\min}$ only during the intervals between equilibria, when we do not care what these machines do anyway. However, an allowable input will change the states of equilibrium in $\tilde{S}_{\min}$ input sequences in the same way as in $S_{\min}$. This proves our statement.

Having proved that conditions 1 and 2 are not restrictive, let us look for our minimal machine among the machines S_1, S_2, S_3, ..., which reproduce the given machine G in terms of L_G and which satisfy these conditions.

As we have already pointed out in the preceding section, reproduction requires that for any input sequence

$$\rho_{\alpha_0}\rho_{\alpha_1}\rho_{\alpha_2}\cdots\rho_{\alpha_s}\cdots \qquad (\rho_{\alpha_i}\neq\rho_{\alpha_{i+1}}) \tag{10.8}$$

of machine G there be an input sequence

$$\underbrace{\rho_{\alpha_0}\rho_{\alpha_0}\cdots\rho_{\alpha_0}}_{q_0\text{ times}}\ \underbrace{\rho_{\alpha_1}\rho_{\alpha_1}\cdots\rho_{\alpha_1}}_{q_1\text{ times}}\cdots\underbrace{\rho_{\alpha_1}\rho_{\alpha_s}\cdots\rho_{\alpha_s}}_{q_s\text{ times}}\ (\rho_{\alpha_i}\neq\rho_{\alpha_i+1})$$

of machine S, where $q_i \geqslant m$ $(i=1, 2, \ldots)$. Since $m=1$ for any S_i, the set L_S of input sequences allowed in S_i can be any set T_{q^*} $(q^*=1, 2, 3, \ldots)$ of sequences such as (10.6), assuming $q_i \geqslant q^*$ $(i=1, 2, 3, \ldots)$; in particular, L_S can be the set $E=T_1$.

Assume we want S_i to reproduce G, and let the set $T_{\bar{q}^*}$ be used as the set L_S. We shall then prove the following statement (A): if state $\varkappa_{\alpha_i}$ of machine S_α and state $\varkappa_{\beta_j}$ of machine S_β are equivalent in terms of set $T_{\bar{q}^*}$, then they are also equivalent with respect to all the sets $T_1, T_2, T_3, \ldots$; that is, they are ***simply equivalent***, since $T_1=E$ (S_α and S_β may also refer to the same machine). If $q>\bar{q}^*$, our statement is obviously true, since in this case $T_q \subset T_{\bar{q}^*}$ [see Section 10.3, Eq. (10.7)]. If $q<\bar{q}^*$, the truth of our statement follows from the fact that with any change of input, all machines S_i will go to an equilibrium in one discrete instant, after which repetition of an input symbol will not change the state of the machine, regardless of the number of times this symbol is fed to the machine.

Every machine S_i reproduces G in terms of set L_G, that is, in terms of the formula of Eq. (10.8). This means that given any initial state $\varkappa_G^0$ and any allowed input to G, there must exist an initial state $\varkappa_{S_i}^0$ of S_i at which that machine, accepting a corresponding input from set $T_{\bar{q}^*}$ [in the form of Eq. (10.6)] and observed only upon attainment of equilibrium after a change of input, generates the same output as G. Without sacrificing generality, we may assume that $\varkappa_{S_i}^0$ will be a state of equilibrium for S_i at $\rho=\rho_{\alpha_0}$; were this not so, then S_i would go in one "fast" cycle from $\varkappa_{S_i}^0$ to $\varkappa_{S_i}^1$ which would have to be a state of equilibrium at $\rho=\rho_{\alpha_0}$. In that case, $\varkappa_{S_i}^1$ would be the state corresponding to $\varkappa_G^0$ of G.

Now consider any two machines S_i and S_j from the set $S_1, S_2, S_3, \ldots$. Suppose that at $\rho=\rho_{\alpha_0}$, state $\varkappa_G^0$ of G corresponds to state $\varkappa_{S_i}^0$ of S_i and to the state $\varkappa_{S_j}^0$ of S_j. Then, since states $\varkappa_{S_i}^0$ and $\varkappa_{S_j}^0$ correspond to the same state of G for the same ρ_{α_0}, the outputs of S_i and S_j, which start from $\varkappa_{S_i}^0$ and $\varkappa_{S_j}^0$, respectively, will coincide for any input from $T_{\bar{q}^*}$ which begins with ρ_{α_0}, this coincidence occurring one "fast" cycle after a change of input. But if there is no change of input, then the machine is also in equilibrium at all other times, and the outputs at these times will also coincide. Thus, the states

$\varkappa^0_{S_i}$ and $\varkappa^0_{S_j}$ are equivalent at all those input sequences from $T_{\bar{q}^*}$ which begin with ρ_{α_0}. But these states will also be equivalent in terms of set $T_{\bar{q}^*}$, since they are equilibrium states at $\rho = \rho_{\alpha_0}$. Indeed, if the input sequence were to begin with some $\rho_{\beta_0} \neq \rho_{\alpha_0}$, then it would be possible to write the sequence

$$\underbrace{\rho_{\alpha_0}\rho_{\alpha_0} \cdots \rho_{\alpha_0}}_{\bar{q}^* \text{ times}}\rho_{\beta_0}, \tag{10.9}$$

which does begin with ρ_{α_0} and does belong to $T_{\bar{q}^*}$. With respect to this sequence, the states $\varkappa^0_{S_i}$ and $\varkappa^0_{S_j}$ are equivalent. At the end of $\bar{q}^*$ cycles, the machines starting from these states, are again in these states; thus the initial conditions are not changed, and we may take the $(\bar{q}^* + 1)$-th fast cycle as the reference time. If we do that, then the sequence (10.9), taken as of the $(\bar{q}^* + 1)$-th fast cycle, begins with ρ_β.

Thus we have shown that all the states of S_i $(i = 1, 2, 3, \ldots)$ corresponding to the same state $\varkappa^0_G$ of G, are equivalent with respect to the chosen set $T_{\bar{q}^*}$ at $\rho = \rho_{\alpha_0}$. It then follows from statement (A) proved above that they are also equivalent with respect to $T_1 = E$, that is, they are simply equivalent.

Let us now assume that set $S_1, S_2, S_3, \ldots$ yields a fast machine $\bar{S}$ which not only reproduces G in terms of L_G, but also represents it in terms of E. This means that for each state of $\bar{S}$ and for any $\rho = \rho_{\alpha_0}$ we can find a state of G, such that there is representation.

Let us take any state $\tilde{\varkappa}_{\bar{S}}$ of $\bar{S}$, and let $\tilde{\varkappa}_{\bar{S}}$ be an equilibrium state for $\rho = \rho_{\alpha_0}$. Then, under representation, the corresponding state of G is $\tilde{\varkappa}_G$. Observing the operation of $\bar{S}$ and G of various inputs to $\bar{S}$ beginning with ρ_{α_0}, we come to the conclusion that if $\tilde{\varkappa}_G$ corresponds, at $\rho = \rho_{\alpha_0}$, to $\tilde{\varkappa}_{\bar{S}}$ under representation, then $\tilde{\varkappa}_{\bar{S}}$ corresponds, at the same ρ_{α_0}, to $\tilde{\varkappa}_G$ under reproduction (that is, machine $\bar{S}$ represents and reproduces machine G). Since $\tilde{\varkappa}_{\bar{S}}$ may be any state of $\bar{S}$, it follows that for each state $\tilde{\varkappa}_{\bar{S}}$ of $\bar{S}$ there exist suitable $\tilde{\varkappa}_G$ and ρ_{α_0} of G such that $\tilde{\varkappa}_{\bar{S}}$ reproduces $\varkappa_G$ at $\rho = \rho_{\alpha_0}$. But from this it follows directly that for any state $\tilde{\varkappa}_{\bar{S}}$ of $\bar{S}$ there exists a state $\tilde{\varkappa}_{S_i}$ equivalent to it in any machine S_i. This means that all machines S_i may be mapped onto machine $\bar{S}$ (some of these S_i may, of course, also be equivalent to $\bar{S}$). Therefore, all we have to do is to minimize machine $\bar{S}$, that is construct a minimal sequential machine $S_{\min}$ equivalent to $\bar{S}$. And this can be done by means of the Aufenkamp-Hohn algorithm (see Section 9.6).* The machine $S_{\min}$ so obtained will be the minimal s-machine reproducing G in terms of L_G.

*For machine S_i, the decomposition of all the states into groups equivalent in terms of set $T_{\bar{q}^*}$ coincides with groupings equivalent in terms of E.

As already pointed out in Section 10.2, the machine S, derived by transforming the interconnection matrix of G, both reproduces G in terms of L_G and represents it in terms of E. This machine also satisfies conditions 1 and 2 of the present section. Consequently, *to obtain* $S_{\min}$ (to be precise, one of the possible minimal machines) ***it is sufficient to minimize S by symmetrical decomposition of its interconnection matrix.*** The result of the minimization does not depend on which of the sets $T_{\bar{q}^*}$ is used as the set L_S of input sequences allowed in S under reproduction. In this case, restricting the number of sets of input sequences does not further reduce the number of states of the reproducing machine.

We shall now construct a minimal s-machine $S_{\min}$ reproducing a given machine G in terms of L_G.

Example. Let the interconnection matrix C^G of a given "slow" machine G operating in alphabets $\{\rho\} = \{1, 2, 3\}$, $\{\varkappa\} = \{1, 2, 3\}$ and $\{\lambda\} = \{1, 2\}$ be

$$C^G = \begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} (1,1) & (2,1) & (3,1) \\ (2,2)\vee(1,2) & 0 & (3,1) \\ (2,2)\vee(1,1) & (3,2) & 0 \end{bmatrix} \end{array}.$$

The state diagram of G is shown in Fig. 10.12.

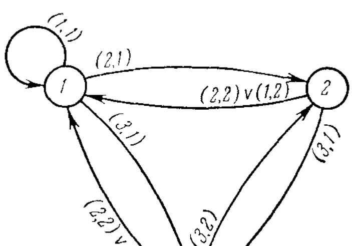


Fig. 10.12.

Transforming C^G as in Section 10.3, we obtain the matrix C^S of the "fast" machine S reproducing G:

$$C^S = \begin{array}{cc} & \begin{array}{cccccc} 1^1 & 1^2 & 1^3 & 2^1 & 2^2 & 3 \end{array} \\ \begin{array}{c} 1^1 \\ 1^2 \\ 1^3 \end{array} & \begin{bmatrix} (1,1) & 0 & 0 & (2,1) & 0 & (3,1) \\ (1,1) & (2,2) & 0 & 0 & 0 & (3,1) \\ 0 & 0 & (1,2) & (2,1) & 0 & (3,1) \end{bmatrix} \end{array}$$

$$
\begin{array}{c}
2^1 \\ 2^2 \\ 3
\end{array}
\left.\begin{array}{cccccc}
0 & 0 & (1,2) & (2,1) & 0 & (3,1) \\
0 & (2,2) & (1,2) & 0 & (3,2) & 0 \\
(1,1) & (2,2) & 0 & 0 & 0 & (3,1)
\end{array}\right]
$$

The state diagram of S is shown in Fig. 10.13. Now let us minimize S. In C^S, rows 2 and 6, as well as 3 and 4 form 1-matrices.

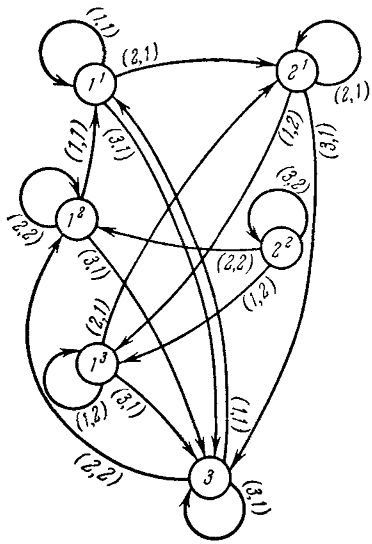


Fig. 10.13.

Let us rewrite C^S so that these rows appear one after another, and then carry out the symmetrical grouping:

$$
C^S = \begin{array}{c|c}
 & \begin{array}{cc|cc|c|c} 1^3 & 2^1 & 1^2 & 3 & 1^1 & 2^2 \end{array} \\
\begin{array}{c} 1^3 \\ 2^1 \\ \hline 1^2 \\ 3 \\ \hline 1^1 \\ \hline 2^2 \end{array} &
\left[\begin{array}{cc|cc|c|c}
(1,2) & (2,1) & 0 & (3,1) & 0 & 0 \\
(1,2) & (2,1) & 0 & (3,1) & 0 & 0 \\
\hline
0 & 0 & (2,2) & (3,1) & (1,1) & 0 \\
0 & 0 & (2,2) & (3,1) & (1,1) & 0 \\
\hline
0 & (2,1) & 0 & (3,1) & (1,1) & 0 \\
\hline
(1,2) & 0 & (2,2) & 0 & 0 & (3,2)
\end{array}\right]
\end{array}.
$$

After all the intermediate steps, we get the interconnection matrix $C^S_{\min}$ of the minimal machine $S_{\min}$ with four states $\varkappa_1, \varkappa_2, \varkappa_3, \varkappa_4$:

$$C^S_{\min} = \begin{array}{c} \\ \varkappa_1 \\ \varkappa_2 \\ \varkappa_3 \\ \varkappa_4 \end{array} \begin{array}{c} \begin{array}{cccc} \varkappa_1 & \varkappa_2 & \varkappa_3 & \varkappa_4 \end{array} \\ \left[\begin{array}{cccc} (1,2)\vee(2,1) & (3,1) & 0 & 0 \\ 0 & (2,2)\vee(3,1) & (1,1) & 0 \\ (2,1) & (3,1) & (1,1) & 0 \\ (1,2) & (2,2) & 0 & (3,2) \end{array}\right] \end{array}.$$

The state diagram of $S_{\min}$ is given in Fig. 10.14.

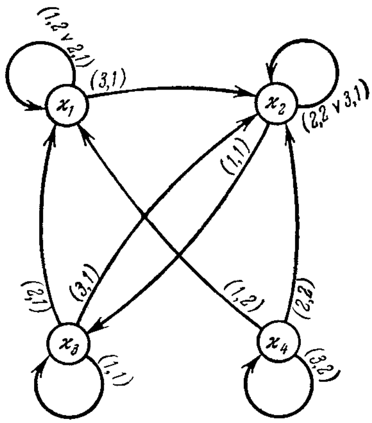


Fig. 10.14.

By virtue of the previously noted fact that each column of C^s can contain only identical pairs, matrix $C^S_{\min}$ has the following property: no column of $C^S_{\min}$ can contain two pairs with identical first and different second subscripts. This means that the state diagram of $S_{\min}$ may be treated, at will, either as the diagram of an s-machine of the P - P type, or as the diagram of an s-machine of the P - Pr type (see the note in Section 3.4). Assume, for example, that the diagram is that of a P - Pr machine. Now, let us show the construction of a relay circuit realizing this machine. To start with, the state diagram yields the tables of the automaton and of the converter of $S_{\min}$ (Tables 10.9 and 10.10).

Table 10.9 can be regarded as a Huffman flow table; all we have to do is to draw squares around the equilibrium states $\tilde{\varkappa}_{\bar{S}}$; that is, those states whose subscripts are the same as the ordinal numbers of the matrix rows. After this Table 10.9 assumes the form of Table 10.11. From Tables 10.10 and 10.11 we can design a relay circuit realizing $S_{\min}$ by means of the method described in Section 5.4. To do this, we assign binary numbers to the symbols $\varkappa$, ρ and λ, as shown in Tables 10.12 - 10.14. Then Tables 10.9 and 10.10 can be expressed in the form of Tables 10.15 and 10.16, from which we derive the combined Table 10.17.

Table 10.9

$x \backslash \rho$	ρ_1	ρ_2	ρ_3
x_1	x_1	x_1	x_2
x_2	x_3	x_2	x_2
x_3	x_3	x_1	x_2
x_4	x_1	x_2	x_4

Table 10.10

$x \backslash \rho$	ρ_1	ρ_2	ρ_3
x_1	λ_2	λ_1	λ_1
x_2	λ_1	λ_2	λ_1
x_3	λ_1	λ_1	λ_1
x_4	λ_2	λ_2	λ_2

Table 10.17 defines three logical functions Y_1, Y_2 and Z of four independent variables x_1, x_2, y_1, and y_2. We shall now assume x_1 and x_2 represent the states of the input contacts, y_1 and y_2—the states of contacts of the secondary relays, and Y_1 and Y_2—the states of the

Table 10.11

$x \backslash \rho$	ρ_1	ρ_2	ρ_3
x_1	$\boxed{x_1}$	$\boxed{x_1}$	x_2
x_2	x_3	$\boxed{x_2}$	$\boxed{x_2}$
x_3	$\boxed{x_3}$	x_1	x_2
x_4	x_1	x_2	$\boxed{x_4}$

Table 10.12

$x \backslash y$	y_2	y_1
x_1	0	0
x_2	0	1
x_3	1	0
x_4	1	1

Table 10.13

$\rho \backslash x$	x_2	x_1
ρ_1	0	0
ρ_2	0	1
ρ_3	1	0

Table 10.14

$\lambda \backslash Z$	Z
λ_1	0
λ_2	1

coils (energized or deenergized) of these secondary relays; the state of the coil of the output relay will be Z. Now, any network made up of contacts x_1, x_2, y_1, and y_2, as well as coils Y_1, Y_2, and Z and realizing the table of logical functions of Table 10.17 will also realize machine S_{min}. A network of this type may be constructed by any method of Chapter 2.

Table 10.15

y \ x	00	01	10
00	00	00	01
01	10	01	01
10	10	00	01
11	00	01	11

Table 10.16

y \ x	00	01	10
00	1	0	0
01	0	1	0
10	0	0	0
11	1	1	1

Now let us compare our minimization method with that of Huffman (Section 5.4). The main difference between the two methods is that

Table 10.17

y_1	0	1	0	1	0	1	0	1	0	1	0	1
y_2	0	0	1	1	0	0	1	1	0	0	1	1
x_1	0	0	0	0	1	1	1	1	0	0	0	0
x_2	0	0	0	0	0	0	0	0	1	1	1	1
Y_1	0	0	0	0	0	1	0	1	1	1	1	1
Y_2	0	1	1	0	0	0	0	0	0	0	0	1
Z	1	0	0	1	0	1	0	1	0	0	0	1

there are no restrictions on the applications of our method, while that of Huffman (as already pointed out in Chapter 5) may only be used to construct those s-machines in which the next state of the automaton of this machine is uniquely determined by the present states of the input and the output of the machine. Where both methods are applicable, they yield identical results, even after minimization.

In concluding this section, let us point out that the algorithm for deriving additional states (Section 10.3) is also applicable when the given slow machine G is subject to Aufenkamp-type constraints. The machine S (which reproduces G in terms of L_G) constructed by means of this algorithm will also be subject to the same constraints. Therefore it should be minimized by the technique described at the end of Section 9.8 (or some other method for full minimization of machines subject to Aufenkamp-type constraints). The fact that this minimization of the fast machine S gives a minimal machine reproducing G may be proved by the same reasoning as that given in the present section, assuming no constraints (the only additional requirement is finding the set of input sequences allowed for each of the states).