THE UNIVERSITY OF MICHIGAN
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Computer and Communication Sciences Department

Technical Report

REALIZATION WITH FEEDBACK ENCODING

Dennis Paul Geller

with assistance from:

Department of Health, Education, and Welfare
National Institutes of Health
Grant No. GM-12236
Bethesda, Maryland

and

National Science Foundation
Grant No. GJ-29989X
Washington, D.C.

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

May 1972

ABSTRACT

REALIZATION WITH FEEDBACK ENCODING

by

Dennis Paul Geller

Chairman: John F. Meyer

Automata theorists have long been concerned with various notions of realization. All of these have, as a basic feature, an input encoder which transforms inputs intended for the realized machine, M', into inputs to the realizing machine, M. In this paper we introduce one modification to this concept. We give the realizing machine a measure of control over the realization by introducing feedback to the input encoder. The input encoder is then a map h from pairs (q,i') to inputs of M, where q is a state of M and i' is an input to M'. We introduce the restriction that for each q, the map h(q,.) is one-to-one and onto.

With this restriction, we are able to demonstrate strong ties with the theory of directed graphs. We introduce and study admissible homomorphisms of graphs and show that they play a role for realization with feedback encoding similar to that played by homomorphisms of machines for classical realization. We also investigate algebraic properties of realization with feedback encoding.

Finally, we discuss two applications of the concept of realization with feedback encoding. The first is to the problem of finding series-parallel decompositions of automata, and the second involves the design of diagnosable machines capable of realizing a given state behavior.

ACKNOWLEDGEMENTS

It is difficult to adequately express appreciation to all those who have helped me reach this stage. I most certainly thank the Research Center for Group Dynamics, Logic of Computers Group and the Computer and Communication Sciences Department, all of The University of Michigan, and the School of Advanced Technology and SUNY Binghamton for the financial aid which they provided; in this line I must also thank the Woodrow Wilson Foundation and the National Science Foundation.

As for the many individuals who helped me there are, first and foremost, the members of my committee. Before there was a committee, Frank Harary provided support, and tried to teach me how to write and ask questions. I will always be grateful to Stephen Hedetniemi and Gary Chartrand, not only for their help, but also for their friendship. This thesis might never have come into existence without the superlative translating, typing and shepherding done by Jan McDougall.

Finally, to my wife Naomi: thanks.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

In this thesis we will be studying a new definition for the concept of realization of finite-state sequential machines. Although their content may vary, definitions of the notion of realization in the literature all fit into a basic paradigm. As the realizing machine will, in general, have a different input alphabet than the realized machine, provision is made for a memoryless input encoder, which transforms inputs "intended for" the realized machine into the input alphabet of the realizing machine. In some cases, (realization [17, p. 28]), this encoder transforms symbols to symbols, and in other cases, (simulation [17, p. 192]), symbols to strings. One feature, however, is always constant: the way a symbol is transformed is invariant. In this work we propose an apparently drastic change to this paradigm: we allow the realizing machine to exert a measure of control over the realization by adding feedback from the realizing machine to the input encoder. In this way, a given symbol may be transformed in any one of a number of different ways, depending on the current state of the realizing machine. We call this new notion of realization, *realization with feedback encoding*.

It is a truism of automata theory that adding even a small amount of feedback to a system can have profound results, and the modification which we propose here is no exception. We show, in Chapter III, and in a different context in Chapter IV, that unless some restrictions are placed on the behavior of the input encoder, the concept becomes trivial, in the sense that there may be no discernible

relation, either behaviorally or structurally, between the realized and realizing machines.

The restrictions which we place on the encoders in Chapter III are natural ones, but they lead to a surprising result. Whether or not one machine can realize another with feedback encoding can be ascertained by examining their state transition diagrams with the input labels on the arcs removed. Thus, graph theoretic tools become quite valuable in the study of realization with feedback encoding. Graph Theory is reviewed in Chapter II, where we present the notion of an admissible homomorphism between graphs. Properties of admissible homomorphisms are discussed, and they are related to other forms of mappings between graphs, most notably those of McNaughton [25] and of Harary, Hedetniemi and Prins [13]. In Chapter III we show that the existence of an admissible homomorphism between the unlabelled state transition graphs of two machines is both necessary and sufficient for realization with feedback encoding of the range machine by the domain machine. We then show how well-known theorems about realization of machines by cascade compositions can be generalized to include the class of realizations with feedback encoding. Using the complexity measure *size*, we then generalize a result of Zeigler [36] and show that it is impossible to realize, even with feedback encoding, all machines by cascades of bounded complexity. Finally, we investigate the addition of a feedback encoder to the special case of simulation, b-*slow simulation*, and in the process we provide a characterization of those digraphs which are powers of other digraphs.

In Chapter IV, we investigate algebraic properties of both realizations and simulations with feedback encoding. We show that it is possible to associate a semigroup with each machine so that the question of whether one machine can realize or simulate another with feedback encoding can be resolved by examining the semigroups associated with the two machines. We also relate the notion of simulation with feedback encoding to the S^*-semigroups defined by Hedetniemi and Fleck [20], and settle a conjecture of theirs in all but one case.

In Chapter V we examine the problem, given a machine, of finding a realizing machine which has a distinguishing sequence. It is shown that allowing realizations with feedback encoding makes the problem in general more tractable even with the additional restrictions, which we assume throughout, that there be no increase in state-set or input-set size, and as little increase as possible in output-set size. In particular, we define a class of machines for which the problem can always be solved when realization with feedback encoding is permitted. For other machines, we show that the difficulties which are reflected in the structure of the state transition graph of the machine to be realized can be modelled as a problem in coloring the lines of bipartite graphs. We develop a new canonical form for such colorings, and then apply it to the distinguishing sequence problem.

CHAPTER I

FUNDAMENTALS

This chapter and the next present some of the basic concepts from the theories of machines and graphs which will be important to the study at hand.

A *machine*, or *finite-state automaton* M, consists of a set Q of *states*, a set I of *inputs*, where both Q and I are finite, and a map $\delta: Q \times I \to Q$. A machine $M = \langle Q, I, \delta \rangle$ can be specified either by giving the map δ explicitly, or by drawing its *state-transition graph*, a labeled directed graph (See Chapter II) whose points are the states of M, having an arc from state q_i to state q_j labeled with input $x \in I$ if and only if $\delta(q_i, x) = q_j$. A machine is *autonomous* if $|I| = 1$.

Example 1.1

The following table for δ describes the machine $M = \langle Q, I, \delta \rangle$ whose state-transition graph is given in Figure 1.1; $Q = \{q_1, q_2, q_3, q_4\}$ and $I = \{x_1, x_2\}$.

M	x_1	x_2
q_1	q_1	q_4
q_2	q_4	q_1
q_3	q_3	q_2
q_4	q_4	q_3

We will make the notational convention that, unless otherwise specified, the state set and input set of a machine M are always denoted by Q and I, respectively; thus, for example, a machine M' has state set Q', and a machine M_2 has input set I_2.

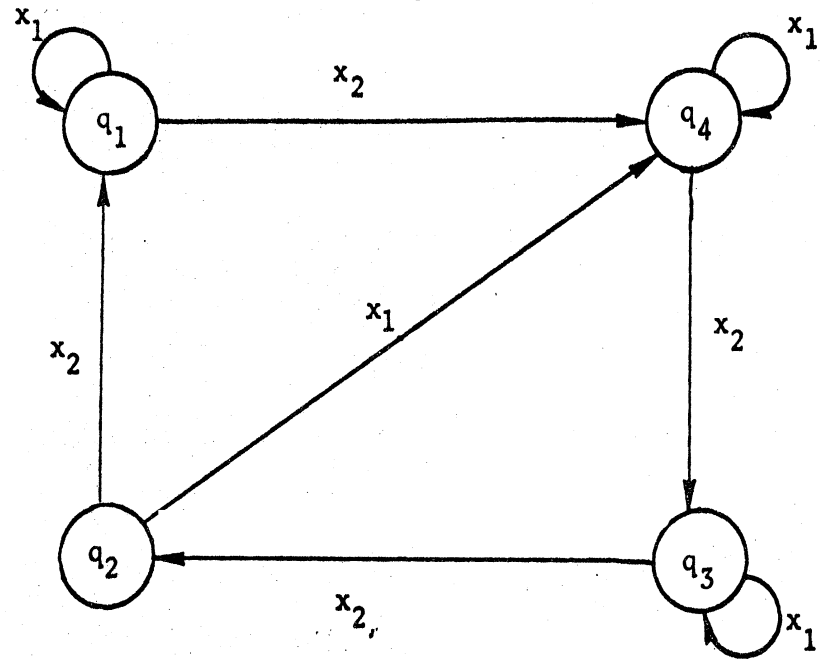


Figure 1.1. A state-transition graph.

We will also omit explicit reference to δ, the *transition function*; if $\delta(q_i,x) = q_j$ we will instead write

$$q_i x = q_j. \tag{1.1}$$

A machine M', is a submachine of a machine M if $Q' \subseteq Q$, $I' \subseteq I$, and $\delta' = \delta|_{Q' \times I'}$.

A *string* y over a set I is a nonempty finite sequence $y = x_1 x_2 \ldots x_n$ of symbols from I; the *length* $\ell(y)$ is the number n of symbols in the sequence. Given any finite set I we define I^+ to be the set of all strings over I.

The transition function δ of a machine M can be extended to a map $\overline{\delta}: Q \times I^+ \to Q$ by setting $\overline{\delta}(q,x) = \delta(q,x)$ for all $x \in I$ and $\overline{\delta}(q,xy) = \overline{\delta}(\overline{\delta}(q,x),y)$, for $x,y \in I^+$; alternately, we could write $q(xy) = (qx)y$. In fact, as is common, we will use the symbol δ or the notation of (1.1) for both δ and $\overline{\delta}$. It is well known that the value of qz for $z \in I^+$ is independent of the way that z is decomposed

into a product $z = xy$ of strings. For the machine of Example 1.1,

$$q_1x_1x_2 = q_4$$
$$q_2x_1x_2 = q_3$$
$$q_3x_1x_2 = q_2$$
$$q_4x_1x_2 = q_3$$

Let $\vec{Q}$ be the vector of length $|Q|$, where $\vec{Q}(i) = q_i$. Each string $x \in I^+$ transforms $\vec{Q}$ to a vector $\vec{Q}x$, where $\vec{Q}x(i) = q_ix$. Note that for the machine of Example 1.1, $\vec{Q}x_2' = \vec{Q}x_2x_2x_2x_2x_2$.

Let $I^* = I^+ \cup \{\Lambda\}$, where Λ, the *empty string*, has the properties that $\ell(\Lambda) = 0$, $\vec{Q}\Lambda = \vec{Q}$ and, for any string $y \in I^+$, $\Lambda y = y\Lambda = y$. Define an equivalence relation on I^* by setting $x \equiv y$ if and only if $\vec{Q}x = \vec{Q}y$. Then the equivalence classes under $\equiv$ are the elements of a semigroup, $S(M)$, where $[x][y] = [xy]$; $S(M)$ is the *semigroup of* M.

One notion which is crucial to our study is that of one machine being able to "simulate" the behavior of another.

We say that M *realizes* M' if there are maps $\phi: Q \overset{\text{onto}}{\rightarrow} Q'$ and $h: I' \overset{\text{onto}}{\rightarrow} I$ such that

$$\phi(q)x' = \phi(qh(x')). \tag{1.2}$$

The map h is called the *input encoder*. If the map ϕ is one-to-one we say that the realization is *isomorphic*, and otherwise it is *homomorphic*. In either case, ϕ is said to be a *homomorphism with substitution property*, or an *SP homomorphism*. An important property of an SP-homomorphism between two machines is that if

$$\phi(q_1) = \phi(q_2) \text{ then for all } x \in I,\ \phi(q_1x) = \phi(q_2x) \tag{1.3}$$

The partition which an SP homomorphism induces on its domain is an *SP partition*.

A machine M *simulates* a machine M' if there are maps $\phi:Q \to Q'$ and $h:I' \to I^+$ such that

$$\phi(q)x' = \phi(qh(x')) \tag{1.4}$$

A special case of simulation is b-slow simulation; M *b-slow simulates* M' if M simulates M' and the input encoder h has the property that for each $x' \in I'$, $\ell(h(x')) = b$.

Often it is not important to know precisely to which state an input takes a machine. Instead, outputs are associated with a machine, and one is interested in the mapping from inputs to outputs which the machine induces.

A *Mealy machine* is a machine $M = \langle Q,I,\delta \rangle$ together with a finite set Y of *outputs* and an *output function* $\lambda:Q \times I \to Y$; $M = \langle Q,I,\delta,\lambda,Y \rangle$. It may often be the case that for any states q_1,q_2 and inputs x_1,x_2, $q_1x_1 = q_2x_2$ implies that $\lambda(q_1,x_1) = \lambda(q_2,x_2)$. In such a machine $\lambda(q,x)$ can be considered to be a function of the next state, qx. Such a machine is called a *Moore machine*, and, as the meaning will always be clear from context, we will write $\lambda(q,x) = \lambda(qx)$. In fact, the two types of machines define the same classes of behaviors. In many applications, Moore machines are instead defined so that $\lambda(q,x) = \lambda(q)$. We chose the formulation which we did (see [1]) to make the derivations in Chapter V more tractable although, as we point out there, we actually lose no generality by doing so.

As with the transition function, we can extend λ to a map with domain $Q \times I^+$. In fact, this can be done in two ways. We define $\overline{\lambda}:Q \times I^+ \to Y$ by $\overline{\lambda}(q,x) = \lambda(qy,x_1)$, where $x = yx_1 \in I^+$ and $x_1 \in I$. On the other hand, $\beta:Q \times I^+ \to Y^+$ is the map $\beta(q,x_1 \ldots x_n) = \lambda(q,x_1)\lambda(qx_1,x_2) \ldots \lambda(qx_1 \ldots x_{n-1},x_n)$. Using the same symbol, λ, for both λ and $\overline{\lambda}$, we see that in the Mealy machine $M = \langle \{q_1,q_2,q_3,q_4\},$

$\{x_1,x_2\}$, δ, λ, $\{0,1\}$> of Figure 1.2,

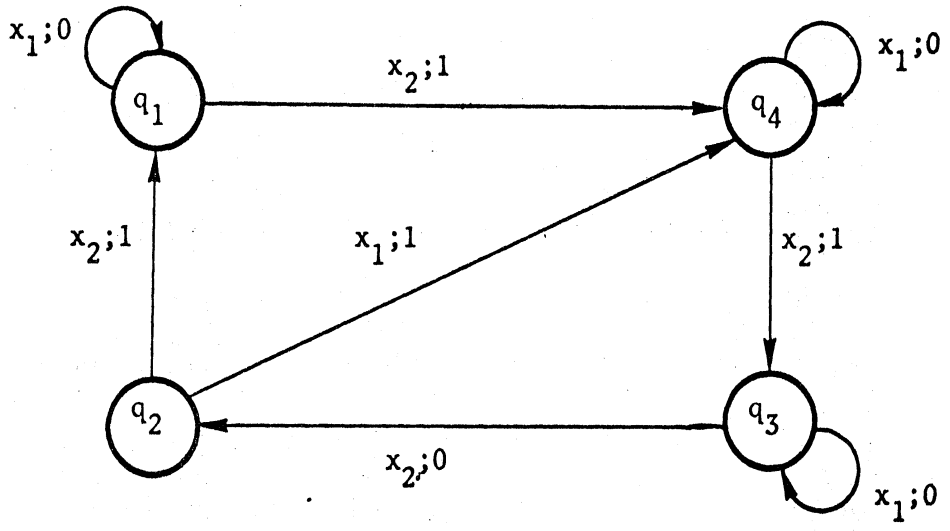


Figure 1.2. A Mealy machine.

$\lambda(q_1,x_1x_2) = \lambda(q_4,x_1x_2) = \lambda(q_2,x_1x_2) = 1$, and $\beta(q_1,x_1x_2) = \beta(q_4,x_1x_2) = 01$, but $\beta(q_2,x_1x_2) = 11$.

We say that two states q_1 and q_2 of a machine are *equivalent* if, for each $x \in I^+$, $\lambda(q_1,x) = \lambda(q_2,x)$. A machine is *reduced* if no two states are equivalent. For every machine M which is not reduced there is a unique reduced machine which is formed by identifying each set of equivalent states to a single state.

If M and M' are machines with output then M *realizes* M' if there are maps $\phi:Q \overset{\text{onto}}{\to} Q'$, $h:I' \overset{\text{onto}}{\to} I$, $g:Y \to Y'$ such that

$$\begin{aligned} \phi(q)x' &= \phi(qh(x')) \\ \lambda'(\phi(q),x') &= g(\lambda(q,h(x'))) \end{aligned} \tag{1.5}$$

It then follows that every machine realizes its reduced machine.

Given machines M_1 and M_2, let Z be a map $Z:Q_1 \times I_1 \to I_2$. The *cascade* $M_1 \circ_Z M_2$ *of* M_1 *and* M_2 with *connecting map* Z is a machine M with $Q = Q_1 \times Q_2$, $I = I_1$, and $(q_1,q_2)x_1 = (q_1x_1, q_2Z(q_1,x_1))$. If Z is in fact independent of the state of M_1, $Z:I_1 \to I_2$, then the cascade

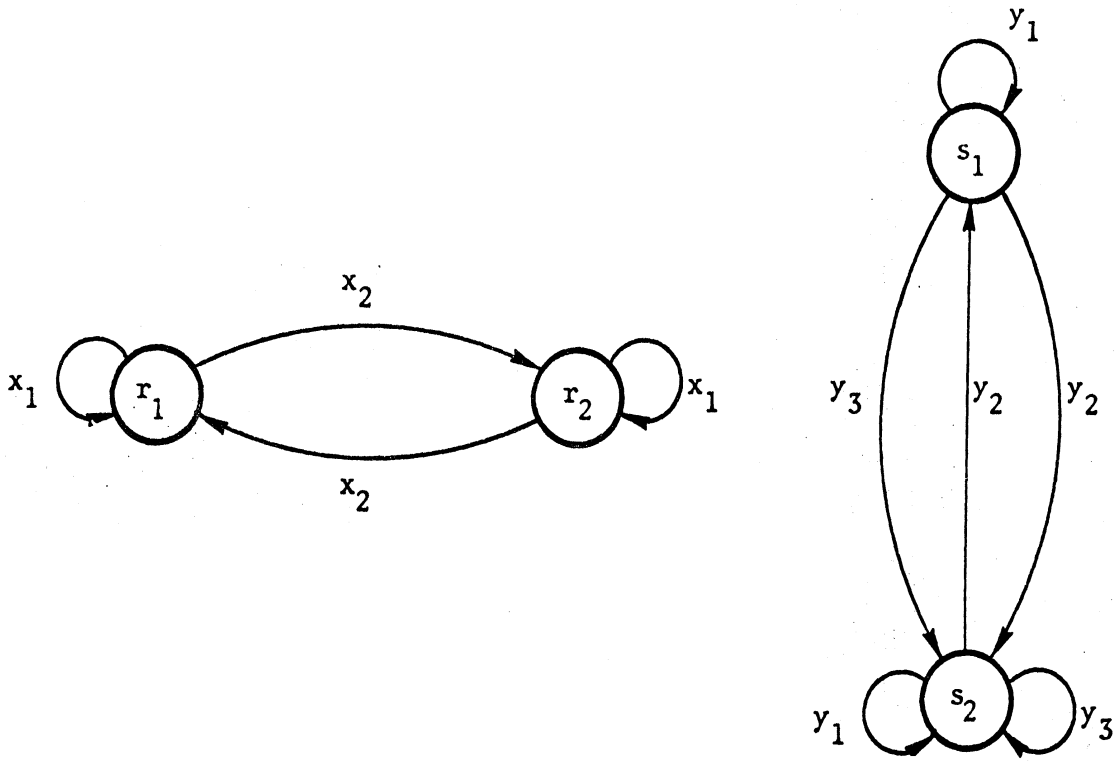
is a *parallel composition.* In a cascade $M_1 \circ_Z M_2$, M_1 is the *front component* and M_2 is the *tail component.*

The operation of cascading machines is, in general, not associative. On the other hand, if we write $M_1 \circ_{Z_1} M_2 \circ_{Z_2} M_3$ we are not actually being ambiguous: once we specify the maps Z_i the cascade will be defined uniquely up to isomorphism. Thus, if we have a cascade M of n machines M_i we can write $M = \Pi M_i$ as a purely notational convenience. Of course, since cascade is a binary operation, every cascade ΠM_i can be written as $\Pi M_i = N_1 \circ_Z N_2$, where, in general, both N_1 and N_2 can be specified as cascades of the M_i.

Among the well-known, in fact almost "folklore", theorems to which we will be referring in the sequel are ([17]).

1. M can be isomorphically realized by a cascade $M_1 \circ_Z M_2$ if and only if M_1 is SP homomorphic image of M.
2. M can be isomorphically realized by a parallel composition $M_1 \circ_Z M_2$ if and only if each of M_1 and M_2 is an SP homomorphic image of M and the meet of the induced SP partitions is the zero partition on Q.

For example, the partition $\{\overline{q_1q_3};\overline{q_2q_4}\}$ on the machine M of Example 1.1 has SP, and is the only such partition which is nontrivial. The cascade $M_1 \circ_Z M_2$ in Figure 1.3 isomorphically realizes M.



Z	x_1	x_2
r_1	y_1	y_2
r_2	y_3	y_2

ϕ	s_1	s_2
r_1	q_1	q_3
r_2	q_2	q_4

Figure 1.3. A cascade.

CHAPTER II

ADMISSIBLE HOMOMORPHISMS

In this chapter we will develop an apparently new notion for homomorphisms of directed graphs, the admissible homomorphisms. A survey of other definitions for homomorphisms can be found in [18]. Our admissible homomorphisms will be seen to lie "between" SP homomorphisms of machines and McNaughton's pathwise homomorphisms [25]. The notion is somewhat tangent to the usual idea of homomorphisms in graph theory [12], but we will develop some relationships to these; in particular, we will strengthen a result of Hedetniemi [19] on the conjunction of two graphs by generalizing it to digraphs and then showing that in many cases the homomorphisms which he considers are, in fact, admissible.

In graph theory, the term digraph usually refers to irreflexive relations. This turns out to be unsuitable for our purposes: most of our results will eventually be applied to state-transition graphs of machines, which often have loops or multiple arcs. We will therefore, unless we specify otherwise, use the term digraph to refer to the more general structures called nets in [15].

A *digraph* D consists of a set V of *points* together with a collection X (repetitions permitted) of ordered pairs, called *arcs*, from V x V. If uv = (u,v) is an arc of D write uv ϵ D or uv ϵ X. If there is at least one arc from u to v then u is *adjacent to* v and v is adjacent *from* u. The expression "uv ϵ X" is used both in the normal set-theoretic usage and also as a predicate, to express "u is adjacent to v". Such dual usage causes no confusion, except possibly when there is more

than one arc from u to v. Where difficulties may arise, we will try to be especially careful in using this notation. Figure 2.1 shows two digraphs; the first is also a digraph in the sense of standard graph-theoretic usage. We will generally use the symbols p for $|V|$ and q for $|X|$.

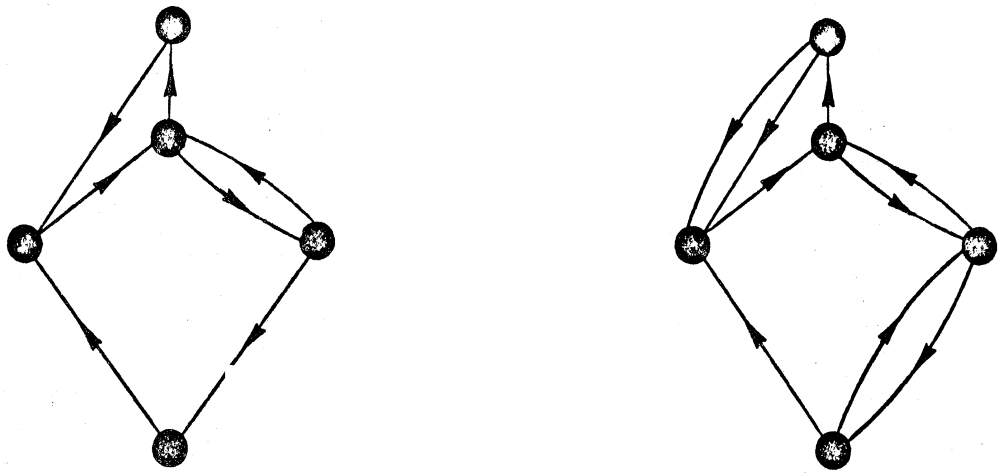


Figure 2.1. Two digraphs.

With each digraph D, we associate a relation γ, defined by $(u,v) \in \gamma$ if and only if $uv \in D$; then $\gamma(u) = \{v | uv \in D\}$ and $\gamma^{-1}(v) = \{u | uv \in D\}$.

With each point of a digraph D, we associate two numbers. The *indegree* id(u) of u is the number of arcs $vu \in D$; note that, since we are permitting multiple arcs, id(u) is not in general equal to $|\gamma^{-1}(u)|$. Similarly the *outdegree* od(u) is the number of arcs $uv \in D$. A digraph is *outregular* (*of degree* d) if all its points have the same outdegree (equal to d).

If a digraph is, in fact, an irreflexive relation, and hence a digraph in the sense of [15], we will say that it is *pure*.

We will use the standard definition for graphs [12]. A *graph* G consists of a set V of *points* together with set X of unordered pairs of distinct points, called *lines*. If there is a line between u and v in G we write uv ε G or uv ε X.

As we mentioned, we will be interested in applying the results of this chapter to state-transition graphs of machines. To do this, we will associate a digraph with each machine. If M = <Q,I,δ> is a machine, the *digraph* D(M) *of* M has for its points the set Q and for its arcs the collection X of arcs uv such that for some x ε I, ux = v. Note that if there are, say, n inputs x_i for which ux_i = v then there are n arcs from u to v in D(M), so that D(M) is always outregular. A machine and its associated digraph are shown in Figure 2.2.

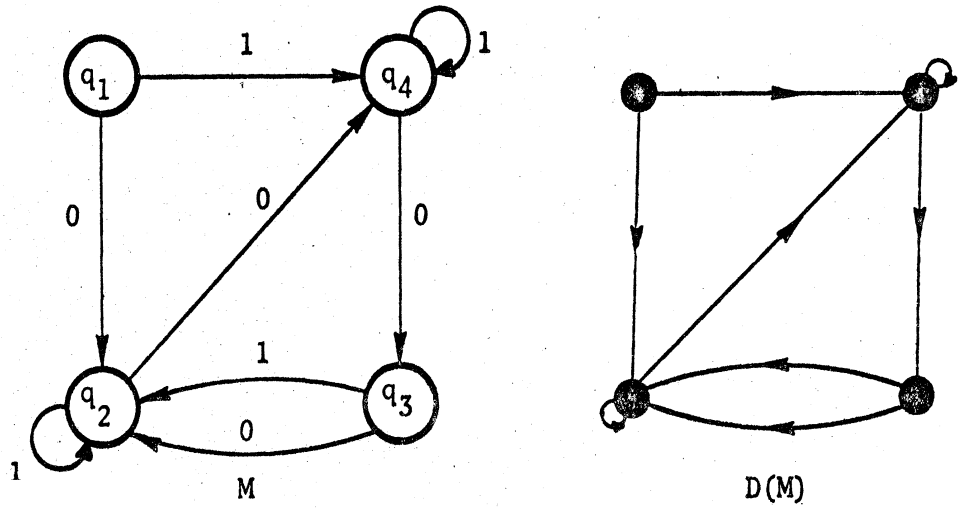


Figure 2.2. A machine and its digraph.

A *walk* in a digraph is a sequence $W = \langle u_0u_1, u_1u_2, \ldots, u_{n-1}u_n \rangle$ of arcs. The *length* of W is n. We may often abbreviate this notation and write W as a sequence of points, $W = u_0u_1 \ldots u_n$, although it is understood that in a digraph with multiple arcs there may be distinct walks which have the same abbreviation. A digraph D is said to be *strong*, or *strongly connected*, if there is a walk $W = v_1 \ldots v_{n-1}v_1$ such that each point of D appears at least once as one of the v_i.

In much of this work we will be dealing with maps between digraphs. Although it will be convenient to treat these as mappings between the point sets of the digraphs, we will provide a slightly unconventional definition. To this end, we introduce some preliminary notation. The *reflexive closure* D^R of a digraph D is the smallest superdigraph of D which has a loop at each point. For a point $u \in D$, let $\vec{S}(u)$ be the set of arcs incident from u, and let $\overleftarrow{S}(u)$ be the set of arcs incident to u.

Given digraphs D and E, by a *mapping* from D onto E we will mean a mapping $\phi: X(D) \to X(E^R)$ which is onto X(E) and which satisfies the following property: for each $u \in V(D)$ there is a $u' \in V(E)$ such that $\phi(\vec{S}(u)) \subseteq \vec{S}(u')$ and $\phi(\overleftarrow{S}(u)) \subseteq \overleftarrow{S}(u')$. A consequence of this definition is that ϕ can be considered as a map from V(D) onto V(E); we will write ϕ for the map in either case, and it will be clear from context whether we are mapping points to points or arcs to arcs.

There are two important features of mappings as we have defined them. Hedetniemi [19] calls a mapping $\phi: G \to H$ between the point sets of two graphs a *full* mapping if for each line $u'v' \in H$ in the image there

is a line $uv \in G$ such that $u' = \phi(u)$ and $v' = \phi(v)$; fullness is automatically a property of mappings as we have defined them. Another important property is that if $x \in X(D)$ is an arc from u to v then $\phi(x)$ is an arc from $\phi(u)$ to $\phi(v)$ in $X(E)$, unless $\phi(u) = \phi(v)$, in which case $\phi(x)$ may not be a loop in $X(E)$, but may instead be a loop in $X(E^R) - X(E)$. The reason for this dichotomy arises from differences between well-established results in the theories of graphs and automata.

Following McNaughton [25], if D and D' are digraphs and $\phi: D \overset{\text{onto}}{\rightarrow} D'$ is a map, then ϕ is said to be *walkwise* if, for any walk $\langle x'_1,\ldots,x'_n\rangle$ in D' there is a walk $\langle x_1,\ldots,x_n\rangle$ in D such that $\phi(x_i) = x'_i$, $i = 1,\ldots,n$; since McNaughton used the term "path" for walk, he called such mappings "pathwise". McNaughton defined "pathwise homomorphisms" between state-transition graphs in conjunction with his studies of the star-height problem. He showed that the "rank" (in the sense of [5]) of a state-transition graph cannot increase under a walkwise homomorphism. Hedetniemi [19] studied maps which were homomorphisms between graphs in the sense of [12], from here on simply called homomorphisms, and also walkwise, and proved a similar result for the cycle rank [12], the number of independent cycles. We will complete this sequence by generalizing Hedetniemi's theorem.

A *cycle* Z in a digraph D is a walk $Z = v_1 \ldots v_n$, where $v_1 = v_n$ and, for $1 \leq i < j < n$, $v_i \neq v_j$. The *length* of a cycle is the number of distinct points. As in [12,p. 37], we can consider each cycle in D to be a formal sum over the q arcs of D, $Z = \sum_{i=1}^{q} \varepsilon_i x_i$, where the ε_i are elements of the two-element field {0,1} and $\varepsilon_i = 1$ if and only if x_i is an arc of Z. Thus, each cycle can be represented by a vector from the vector space of dimension q over the binary field. A set of cycles is *linearly independent* if the cycles are linearly independent as vectors. In a graph G, it is known that the *cycle rank* m(G), the maximum number of independent cycles, is equal to q-p+k, where k is the number of connected components of G. Berge [2] showed that the same equation holds for strong pure digraphs, so that the cycle rank is m(D) = q-p+1. His result can be applied to digraphs as we have defined them.

Lemma 2.1

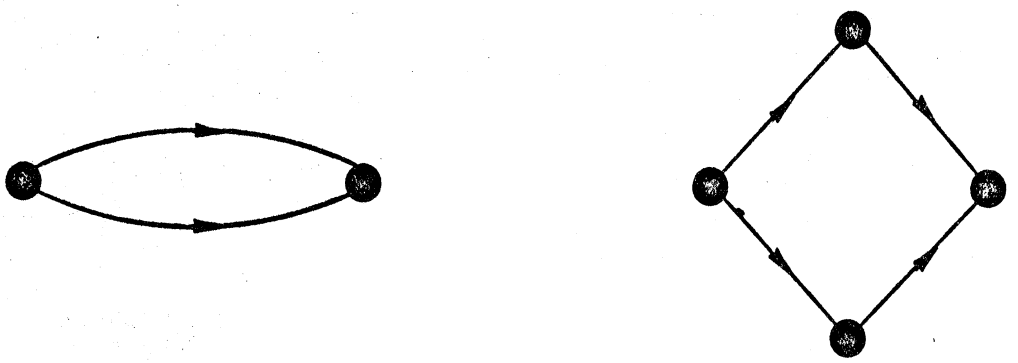
If D is a strong digraph then m(D) = q-p+1.

PROOF: (sketch).

Suppose that D is a strong digraph. We form a new digraph D^+ with $V(D^+) = V(D) \cup X(D)$, and $X(D^+) = \{u_i x_j, x_j u_k \mid u_i u_k = x_j\}$. In effect, D^+ is formed by inserting a new point on each arc of D. It is clear that D^+ is a pure strong digraph, and that $m(D^+) = m(D)$. By Berge's result, $m(D^+) = q(D^+) - p(D^+) + 1$. But $p(D^+) = p(D) + q(D^+) = 2q(D)$. Thus, $m(D) = m(D^+) = 2q(D) - (p(D) + q(D)) + 1 = q(D) - p(D) + 1$.

Berge's result, and hence Lemma 2.1, does not hold for digraphs which are not strong. The digraphs D and D^+ of Figure 2.3 each have

q-p+1 = 1, but each is acyclic. However, if a digraph D has k strong components and every arc belongs to a strong component, then m(D) = q-p+k.



D D^+

Figure 2.3. A digraph and the derived pure digraph.

Corollary

If D is not strong then m(D) is the sum of the cycle ranks of the strong components of D.

Lemma 2.2

If $\phi:D \to D'$ is walkwise and if D' has a cycle of length n, then D has a cycle whose length is a nonzero multiple of n.

PROOF:

Let the cycle Z' in D' have points $v'_1,\ldots,v'_n$, and consider an arbitrarily large walk around $Z':v'_1,v'_2,\ldots,v'_n,v'_1,\ldots$. This walk must have a pre-image which is a walk, say $W = v_{11},v_{12},\ldots,v_{1n},v_{21},\ldots,v_{2n},v_{31},\ldots$, where $\phi(v_{ij}) = v'_j$. Since D is finite there must be repetitions of points in W; suppose that $v_{ij} = v_{k\ell}$ is the first such repetition. Then clearly $j = \ell$ and $v_{ij},v_{i,j+1},\ldots,v_{kj}$ is a cycle whose length is (k-i)n.

For any digraph D and any collection Y of arcs of D, D-Y is a digraph with $V(D-Y) = V(D)$ and $X(D-Y) = X(D)-Y$.

Lemma 2.3

Let D and D' be digraphs and $\phi:V \to V'$ a walkwise map. For any collection $X'_1 = \{u'_1 v'_1, \ldots, u'_n v'_n\}$ of arcs of D', ϕ, considered as a map from $D - \phi^{-1}(X'_1)$ to $D' - X'_1$, is walkwise.

The proof is immediate.

Lemma 2.4

If D is a strong digraph, and x is an arc of D, then $m(D-x) < m(D)$.

PROOF:

Since D is strong, x belongs to some cycle Z of D. Now any linearly independent collection of cycles in D-x is certainly linearly independent in D, but Z cannot be dependent on any of these as $x \notin D-x$.

A strong digraph D is *minimally strong* if for each arc x, D-x is not strong. It is shown in [7] that every minimally strong digraph has at least two points having both indegree and outdegree equal to 1. In particular, each arc x incident to such a point has the property that $m(D-x) = m(D)-1$.

Theorem 2.1

If D and D' are digraphs and if $\phi:V \to V'$ is walkwise, then $m(D) \geq m(D')$.

PROOF:

We proceed by induction. If $m(D) = 0$ then by Lemma 2.2, $m(D') = 0$. Suppose that the theorem is true whenever $m(D) \leq n$ and let $m(D) = n+1$. If $m(D') > m(D)$ then certainly $m(D') \geq 1$. Let x' be an arc in D' such that $m(D'-x') = m(D')-1$. Such an arc certainly exists if D' has a strong component which is not minimally strong, by Lemma 2.1, and, as noted above, a suitable arc also exists if every strong component is minimally strong. Note that x' must belong to a strong component of D', so that there is a closed walk in D' containing x'; therefore some element of $\phi^{-1}(x')$ belongs to a closed walk, and hence a strong component, of D. Then by Lemma 2.4 and the corollary to Lemma 2.1, $m(D-\phi^{-1}(x')) < m(D)$. Using Lemma 2.3 and the inductive hypothesis $m(D-\phi^{-1}(x')) \geq m(D'-x')$. But $m(D) \geq m(D-\phi^{-1}(x'))+1$ and $m(D'-x')+1 \geq m(D')$, so it follows that $m(D) \geq m(D-\phi^{-1}(x'))+1 \geq m(D'-x')+1 = m(D')$.

The proof technique in the theorem is essentially that used in Hedetniemi's proof, but Theorem 2.1 applies to a wider class of maps.

Hedetniemi [19] asked which graphs have no walkwise homomorphic images. During the course of this chapter we will be discussing conditions which are sufficient to insure that a digraph has a walkwise image. Now, however, we present a class of digraphs which have no walkwise images. The digraph which consists precisely of a cycle of length n is an *n-cycle*, denoted C_n.

Theorem 2.2

For any prime p, there is no nontrivial walkwise map whose domain is C_p.

PROOF:

By Lemma 2.2, if ϕ is walkwise, then $\phi(C_p)$ cannot contain any cycles of length greater than one and less than p. On the other hand, since C_p is strong, $\phi(C_p)$ must be strong. The only resolution to these conditions is that either ϕ is an isomorphism or $|V(\phi(C_p))| = 1$; i.e., that ϕ is trivial.

We suspect that the only strong digraphs which have no nontrivial walkwise images are essentially the prime cycles, with loops at some points permitted. For now, we can show that a large number of (strong) digraphs do have walkwise images. A *path* is a walk $v_1v_2 \ldots v_n$ in which no points are repeated. A *simple path in a digraph* D is a path $v_1v_2 \ldots v_n$ such that for $1 < i < n$, each point v_i has indegree and outdegree one in D. An arc is also considered to be a simple path.

Theorem 2.3

Let D be a digraph. If there are points u and v such that there are at least two simple paths from u to v, then D has a nontrivial walkwise image.

PROOF:

We note that if ϕ is a map from D to some digraph D' and if ϕ is an isomorphism from a subdigraph of D, then ϕ is walkwise. Consider first the case in which one of the simple paths is an arc. Let the other simple path from u to v be $uv_1v_2 \ldots v_nv$. If ϕ is the map which takes each v_i to v, and acts as the identity on the other points of D then $\phi(D)$ is isomorphic to $D - \{uv_1, v_1v_2 \ldots v_nv\}$, so that ϕ is walkwise and nontrivial. The same technique works in the case that neither path is an arc. Let the paths be $uv_1 \ldots v_nv$ and $uw_1 \ldots w_mv$, where $n \geq m$. First map each v_i, $i > m$ to v, and then map each v_i, $1 \leq i \leq m$ to w_i. The resulting digraph is again isomorphic to a subdigraph of D, so that the induced map, ϕ, is walkwise.

A map whose only action is to identify two points is an *elementary identification*. For homomorphisms of graphs, there is a very strong interpolation theorem, which states that if $\phi:G \to H$ is any homomorphism which is not elementary, ϕ can be factored into two nontrivial homomorphisms, $\phi_1:G \to H_1$ and $\phi_2:H_1 \to H$ [13]. For walkwise mappings we get a different sort of interpolation property, which we present in the next two theorems.

Theorem 2.4

If $\phi_1:D \to D_1$ and $\phi_2:D_1 \to E$ are both walkwise, then so is $\phi_2\phi_1 = \phi:D \to E$.

PROOF:

This is immediate. Any walk W_E in E has a preimage W_{D_1} in D_1 which is a walk; $\phi_2(W_{D_1}) = W_E$. Also, since W_{D_1} is a walk there is a walk W_D in D such that $\phi_1(W_D) = W_{D_1}$. Clearly $\phi(W_D) = W_E$, so that ϕ is walkwise.

Theorem 2.5

If $\phi: D \to E$ is walkwise and we write $\phi = \phi_2\phi_1$, where $\phi_1: D \to E_1$ and $\phi_2(E_1) = E$, then ϕ_2 is walkwise.

PROOF:

Let $w_1w_2 \ldots w_n$ be a walk in E. Then there is a walk $v_1v_2 \ldots v_n$ in D, where $\phi(v_i) = w_i$. But, $\phi_1(v_1)\phi_1(v_2) \ldots \phi_1(v_n)$ is a walk in E_1. If $u \in \phi_1(\phi^{-1}(w_i))$ then $u \in \phi_2^{-1}(w_i)$. Therefore, $\phi_2(\phi_1(v_i)) = w_i$, and hence $\phi_1(v_1)\phi_1(v_2) \ldots \phi_1(v_n)$ is a walk in E_1 whose image under ϕ_2 is $w_1w_2 \ldots w_n$. Thus, ϕ_2 is walkwise.

This theorem gives us a tool for checking whether a map is walkwise. The map can be expressed as a composition of two maps, of which the second to be applied is elementary, and hence rather simple to check for the walkwise property: if the test fails the original map is not walkwise.

Theorem 2.6

Let $\phi: D \to D'$ be an elementary map which identifies two points u and v to a point $(uv) \in D'$ and which acts as the identity on $V - \{u,v\}$. Then ϕ is walkwise if and only if both of the following hold:

(1) $\gamma v \subset \gamma u$ or $\gamma^{-1}u \subset \gamma^{-1}v$

(2) $\gamma u \subset \gamma v$ or $\gamma^{-1}v \subset \gamma^{-1}u$

PROOF:

Whenever $W' = \ldots w_1(uv)w_2\ldots$ is a walk in D', then either $W_u = \ldots w_1uw_2\ldots$ or $W_v = \ldots w_1vw_2\ldots$ is a walk in D. If $\gamma v \not\subset \gamma u$, let $w_2 \in \gamma v - \gamma u$; then W_v must be a walk in D. But if we take $w_1 \in \gamma^{-1}u$ then we must have $w_1 \in \gamma^{-1}v$; thus $\gamma^{-1}u \subset \gamma^{-1}v$; the rest follows by symmetry.

To prove sufficiency, first label the subconditions:

(α) $\gamma u \subset \gamma v$ (β) $\gamma^{-1} v \subset \gamma^{-1} u$

(μ) $\gamma v \subset \gamma u$ (η) $\gamma^{-1} u \subset \gamma^{-1} v$

Let $W' = \ldots w_1(uv)w_2\ldots$ be a walk in D', and let $W_u = \ldots w_1 u w_2 \ldots$, $W_v = \ldots w_1 v w_2 \ldots$. Clearly $w_1 \in \gamma^{-1}u \cup \gamma^{-1}v$ and $w_2 \in \gamma u \cup \gamma v$. We must show that either W_u or W_v is a walk in D; i.e., that either $w_1 \in \gamma^{-1}u \cap \gamma^{-1}v$ or $w_2 \in \gamma u \cap \gamma v$.

Case ($\alpha\mu$). Then $\gamma u = \gamma v$.

Case ($\beta\eta$). Then $\gamma^{-1}u = \gamma^{-1}v$.

Case ($\alpha\eta$). If $w_2 \in \gamma u$ we are done. If $w_2 \in \gamma v - \gamma u$ then $w_1 v \in D$ by (η), so W_v is a walk.

Case ($\beta\mu$) follows similarly.

Since graphs are symmetric digraphs we have

Corollary

For graphs G and G', if $\phi: G \to G'$ is an elementary identification of u and v to (uv), then ϕ is walkwise if and only if for all $w_1 \in \gamma u$, $w_2 \in \gamma v$, either $w_2 \in \gamma u$ or $w_1 \in \gamma v$; i.e., if and only if either $\gamma u \subseteq \gamma v$ or $\gamma v \subseteq \gamma u$.

Example 2.1

We give an example showing that even if the composition of two maps is walkwise, both need not be walkwise.

Consider the cycles in Figure 2.4. Let ϕ be the map $u_i \mapsto v_{i \pmod 3}$;

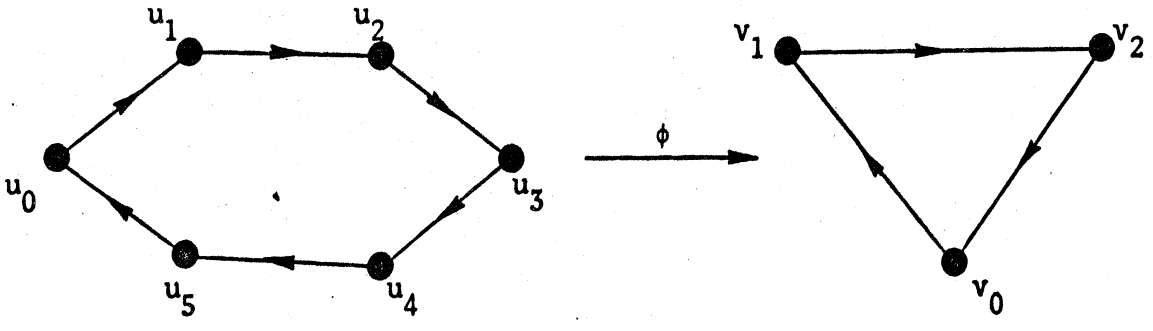


Figure 2.4. A walkwise map.

then ϕ is walkwise. Suppose that we factor ϕ as in Figure 2.5. Using the theorem, we verify that ϕ_2 is walkwise, since $\gamma^{-1}w_2 = \{w_1\}$, $\gamma w_5 = \{w_0\}$, $\gamma w_2 = \{w_0\}$, $\gamma^{-1}w_5 = \{w_1\}$. (This, of course, does not prove that ϕ is walkwise). On the other hand, using Theorems 2.5 and 2.6 we can show that ϕ_1 is not walkwise.

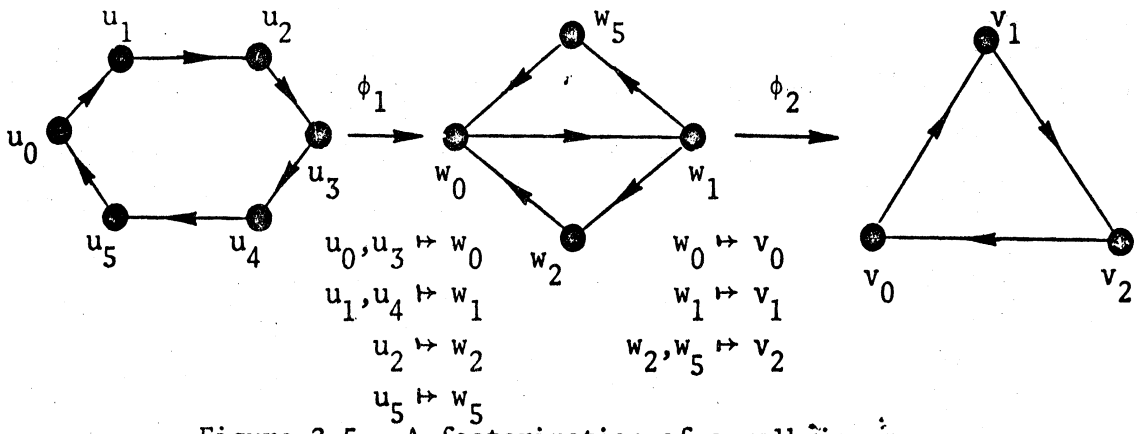


Figure 2.5. A factorization of a walkwise map.

For, if we factor ϕ_1 as in Figure 2.6, then $\gamma r_4 = \{r_5\}$, $\gamma r_1 = \{r_2\}$, $\gamma^{-1}r_4 = \{r_0\}$, $\gamma^{-1}r_1 = \{r_0\}$, so that ϕ_4 is also walkwise, but ϕ_3 is not walkwise, since $u_1 \in \gamma u_0$ and $u_2 \in \gamma^{-1}u_3$ but $u_2 \notin \gamma u_0$ and $u_1 \notin \gamma^{-1}u_3$. In other words, there is no walk whose image is $r_2r_0r_1$.

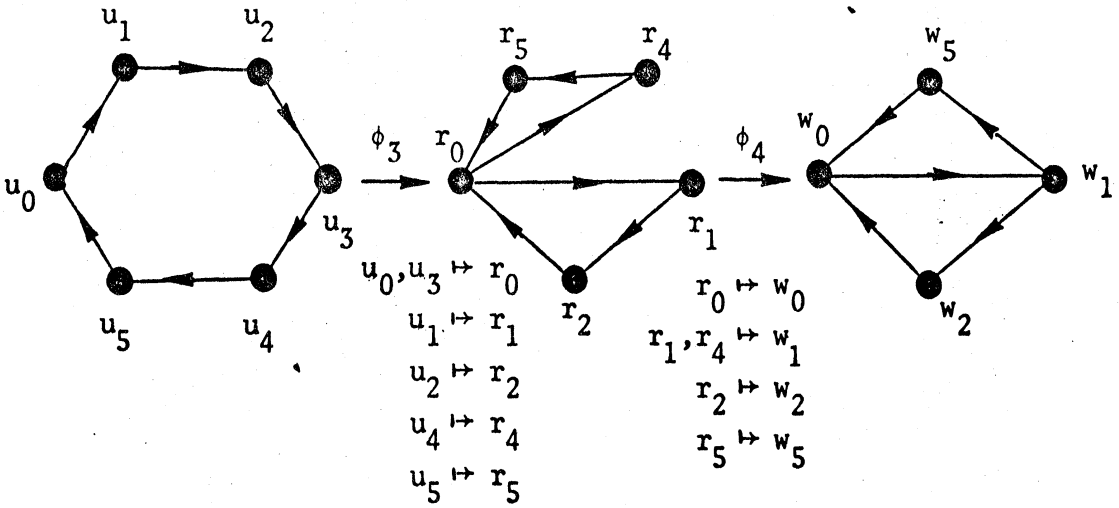


Figure 2.6. A factorization of a map.

We now turn our attention to an important class of mappings, the admissible homomorphisms, which we will show to be walkwise. It is important to note that these need not be homomorphisms in the sense of [12].

Let D and D' be digraphs. A map $\phi:V(D) \to V(D')$ is an *admissible homomorphism* if whenever $\phi(u) = \phi(v)$ and $uw \in D$ then there is a point $\bar{w}$ such that $v\bar{w} \in D$ and $\phi(\bar{w}) = \phi(w)$.

Theorem 2.7

Every admissible homomorphism is walkwise.

PROOF:

Let $\phi:V(D) \to V(D')$ be admissible, let $W' = w'_1 \ldots w'_n$ be a walk in D', and suppose that $w'_1 \ldots w'_{n-1}$ has a walk pre-image $u_1 \ldots u_{n-1}$. Since $w'_{n-1} w'_n \in D'$ and ϕ is full, there are points $w_{n-1} \in \phi^{-1}(w'_{n-1})$, $w_n \in \phi^{-1}(w'_n)$ such that $w_{n-1}w_n \in D$. But then, by admissibility, since $\phi(u_{n-1}) = \phi(w_{n-1})$ there is a point u_n such that $\phi(u_n) = \phi(w_n) = w'_n$, and $u_{n-1}u_n \in D$. Therefore $u_1 \ldots u_n$ is a walk in D and the result follows by induction.

While it is true that a mapping between digraphs can always be written as a map between their point sets, expressing it in this way may not exhibit all possible information. For example, the mapping from digraph D to a subdigraph D' of D which is illustrated in Figure 2.7 would, as a map from V(D) to V(D') seem to be the identity.

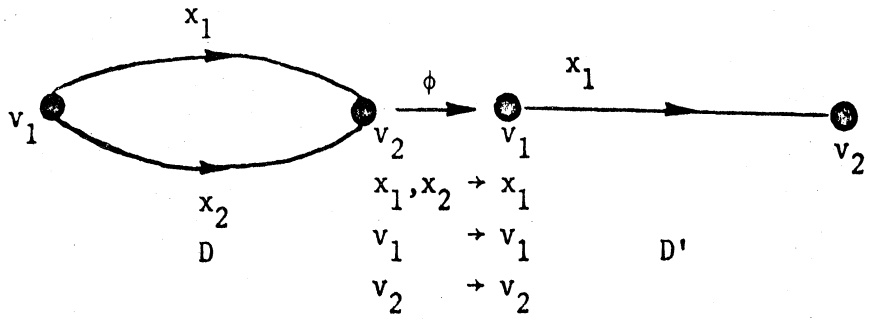


Figure 2.7. A mapping from a digraph to a subdigraph.

On the other hand, it is often possible to write a mapping as a product of elementary identifications of points. From now on we will assume that an elementary identification of adjacent points results in a point with a loop.

Theorem 2.8

Let $\phi:D \to E$ be expressible as $\phi = \varepsilon_n \ldots \varepsilon_1$ where $\varepsilon_i:D_{i-1} \to D_i$ is an elementary identification, and $D_0 = D$, $D_n = E$. Let ϕ_i be $\varepsilon_n \ldots \varepsilon_i:D_{i-1} \to E$. Then if ϕ is admissible, each ϕ_i is.

PROOF:

We first show that $\phi_n = \varepsilon_n$ is admissible. Suppose $\varepsilon_n:D_{n-1} \to D_n = E$ is defined by $\varepsilon_n(u) = \varepsilon_n(v) = (uv)$, and that $uw \in D_{n-1}$.

Let $\Psi_i = \varepsilon_i \ldots \varepsilon_1$. Let $\Psi_{n-1}(u') = u$, $\Psi_{n-1}(v') = v$; since ϕ is full u' can be chosen so that it is adjacent to some w' for which $\phi(w') = \varepsilon_n(w)$. Suppose that w is not one of u,v. Then $\Psi_{n-1}(w') = \phi(w') = w$. Thus there must be a w'' such that $\phi(w'') = \phi(w')$ and $v'w'' \in D$. Since w is not u or v, $\phi(w'') = \phi(w')$ implies that $\Psi_{n-1}(w'') = \Psi_{n-1}(w') = w$. Thus, since $v'w'' \in D$, $vw \in D_{n-1}$.

If w is one of u,v then there is a w'' such that $\phi(w'') = \phi(w') = (uv)$ and $v'w'' \in D$. Thus $\Psi_{n-1}(w'')$ is u or v.

Thus, is either case, there is a $\bar{w}$ such that $v\bar{w} \in D_{n-1}$ and $\epsilon_n(\bar{w}) = \epsilon_n(w)$, so that ϵ_n is admissible.

Now, suppose that $\phi_{j+1} = \epsilon_n \ldots \epsilon_{j+1}$ is admissible. We want to show that $\phi_j = \epsilon_n \ldots \epsilon_j$ is.

Suppose that for $u \neq v$, $\phi_j(u) = \phi_j(v)$ and $uw \in D_{j-1}$.

Case 1. $\epsilon_j(u) \neq \epsilon_j(v)$.

Then, in D_j, $\phi_{j+1}(u) = \phi_{j+1}(v)$, and $u\epsilon_j(w) \in D_j$. Thus, there is a $\bar{w} \in D_j$ such that $v\bar{w} \in D_j$ and $\phi_{j+1}(\bar{w}) = \phi_{j+1}(\epsilon_j(w))$. Since $v\bar{w} \in D_j$ there is a $\bar{w}' \in \epsilon_j^{-1}(\bar{w})$ for which $v\bar{w}' \in D_{j-1}$; clearly $\phi_j(\bar{w}') = \phi_{j+1}(\bar{w}) = \phi_{j+1}(w) = \phi_j(w)$.

Case 2. $\epsilon_j(u) = \epsilon_j(v) = (uv)$.

Since $uw \in D_{j-1}$, $(uv)w \in D_j$. Choose $u' \in \Psi_{j-1}^{-1}(u)$, $v' \in \Psi_{j-1}^{-1}(v)$, $w' \in \Psi_{j-1}^{-1}(w)$, such that $u'w' \in D$; this is always possible by fullness. Since $\phi(u') = \phi(v') = \phi_{j-1}((uv))$, there is a w'' such that $\phi(w'') = \phi(w')$ and $v'w'' \in D$. Let $\bar{w}'$ be $\Psi_{j-1}(w'')$. Then $v\bar{w}' \in D_{j-1}$, and $\phi_j(\bar{w}') = \phi(w'') = \phi(w') = \phi_j(w)$.

A *functional digraph* is a digraph in which each point u has od(u)=1 [12]. If M is an autonomous machine then D(M) is functional. Yoeli and Ginzburg [34] characterized SP homomorphisms of autonomous machines in terms of primitives which they called "elementary" homomorphisms. (Note: keeping track of terminology can be quite tricky here. Yoeli and Ginzburg's use of elementary is different from that of graph theory, where the term is used to mean an elementary identification of nonadjacent points [12]. Also, Yoeli and Ginzburg ([34],[11]) use "admissible" as a synonym for SP.) Hedetniemi asked whether the elementary homomorphisms are always walkwise [18]. That this is indeed true will follow

as a corollary to the next theorem, which will also be proved as a corollary to Theorem 3.5.

Theorem 2.9

Let M and M' be machines and let $\phi:M \to M'$ be SP. Then if $|I| = |I'|$ $\phi:D(M) \to D(M')$ is admissible.

PROOF:

NOTE: We are given a map ϕ from Q to Q'. The first step in showing that ϕ is admissible is to show that ϕ induces a mapping in the formal sense from D(M) to D(M'); that is, we will show that from ϕ we can define a function from X(D(M)) onto X(D(M')) which has the property that for each point u of D(M) there is a point u' of D(M') such that $\phi(\vec{S}(u))\ \vec{S}(u')$ and $\phi\overleftarrow{S}(u))\ \overleftarrow{S}(u')$. Having done this once it will be clear in later proofs that the same technique will let us take other functions between machines and "reinterpret" them as mappings between the digraphs of those machines.

Let x be an arc of D(M) from u to v. Then u and v are states of M and there is an input i such that ui = v, and i is the label on arc x in M. There is an arc x' from $\phi(u)$ to $\phi(v)$ labelled with i (or, more generally, with $h^{-1}(i)$); we take x' to be the image of x. The map thus induced from X(D(M)) to X(D(M')) is onto, since if x' is an arc from $\phi(u)$ to v' induced by input i' then the arc from u to uh(i') which is induced by h(i') will have x' as its image. Since this arc-to-arc mapping is defined in terms of the point-to-point mapping ϕ, it is clear that all the arcs incident to (from) a point u map to arcs incident to (from) $\phi(u)$. We have thus shown that the SP homomorphism ϕ induces a mapping from D(M) to D(M'); in addition, when this mapping is considered as a map from V(D(M)) to V(D(M')), we get precisely

the map ϕ. Notice that having started with a point-to-point map the thing which we really had to provo was that ϕ is full.

Suppose that $\phi(u) = \phi(v)$ and that $uw \in D(M)$. Then there is an input x such that $ux = w$. Since ϕ is SP, $\phi(vx) = \phi(w)$. Thus, letting $\bar{w} = vx$, $\phi(\bar{w}) = \phi(w)$ and $v\bar{w} \in D(M)$, so that ϕ is admissible.

Corollary

Every SP homomorphism is walkwise.

Corollary

Every admissible homomorphism between autonomous machines is SP.

A partition π on the points of a digraph D is an *admissible partition* if whenever $u \equiv_\pi v$ and $uw \in D$ then there is a point $\bar{w}$ such that $v\bar{w} \in D$ and $w \equiv_\pi \bar{w}$.

Lemma 2.5

Any admissible homomorphism induces an admissible partition on its domain. Conversely, if π is an admissible partition on D, then π is the partition induced by some admissible homomorphism whose domain is D.

PROOF:

Let $\phi: D \to D'$ be an admissible homomorphism, and define π by setting $u \equiv_\pi v$ if and only if $\phi(u) = \phi(v)$. Then, since ϕ is admissible, if $u \equiv_\pi v$ and $uw \in D$ there is a point $\overline{w}$ such that $v\overline{w} \in D$ and $\phi(w) = \phi(\overline{w})$. But if $\phi(w) = \phi(\overline{w})$ then $w \equiv_\pi \overline{w}$, so that π is admissible.

On the other hand, if π is admissible, we will define an admissible homomorphism ϕ from D to a digraph D'. The points of D' are the blocks of π, and $\phi(u)$ is just the block $[u]$ of π which contains u. In D', $[u]$ is adjacent to $[v]$ if and only if there are points of D, $u_1 \in [u]$ and $v_1 \in [v]$, such that $u_1v_1 \in D$; note that D' has no multiple arcs. It is easy to see that ϕ is a mapping. If $\phi(u) = \phi(v)$ then $u \equiv_\pi v$. If, also, for some point w, $uw \in D$ then it follows from the admissibility of π that there is a point $\overline{w}$ such that $v\overline{w} \in D$ and $\overline{w} \equiv_\pi w$; but then $\phi(\overline{w}) = \phi(w)$, so that ϕ is an admissible homomorphism which induces the partition π on V(D).

It is interesting to note that, unlike the SP partitions, the admissible partitions of a digraph do not necessarily form a sublattice of the lattice of partitions.

Example 2.2

Consider the digraph D of Figure 2.8. Both $\pi_1 = \{\overline{uv}; \overline{w_1w_3}; \overline{w_2w_4}\}$ and $\pi_2 = \{\overline{uv}; \overline{w_1w_4}; \overline{w_2w_3}\}$ are admissible, but

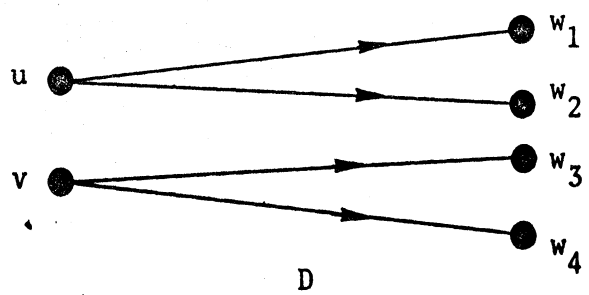


Figure 2.8. A digraph.

the meet $\pi_1 \wedge \pi_2 = \{\overline{uv};\overline{w_1};\overline{w_2};\overline{w_3};\overline{w_4}\}$ is not.

Theorem 2.10

The join $\pi_1 \vee \pi_2$ of two admissible partitions is admissible.

PROOF:

Let π_1 and π_2 be admissible partitions of D, $u \equiv_{\pi_1 \vee \pi_2} v$, and suppose that $uw \in D$. Since $u \equiv_{\pi_1 \vee \pi_2} v$ there is a sequence $u = u_0, u_1, \ldots, u_{t+1} = v$ such that for each $i = 0,1,\ldots,t$, either $u_i \equiv_{\pi_1} u_{i+1}$ or $u_i \equiv_{\pi_2} u_{i+1}$. Then there are points $w = w_0, w_1, \ldots, w_{t+1}$ such that $u_i w_i \in D$ and either $w_i \equiv_{\pi_1} w_{i+1}$ or $w_i \equiv_{\pi_2} w_{i+1}$. Thus $w \equiv_{\pi_1 \vee \pi_2} w_{t+1}$ and $vw_{t+1} \in D$; hence, $\pi_1 \vee \pi_2$ is admissible.

The next theorem is actually a corollary of Theorem 2.9 and the Yoeli and Ginzburg results, but because of its importance, we prove it directly.

Theorem 2.11

Any pure admissible homomorphic image of a cycle C_n is a cycle C_m, where $m|n$.

PROOF:

Let $\phi: C_n \to D$ be admissible, and let π be the induced admissible partition. If the points of C_n, in cyclic order, are $u_0, \ldots, u_{n-1}$, suppose that u_i and u_j, $i < j$, are such that $u_i \equiv_\pi u_j$ and $u_k \equiv_\pi u_m$, $m > k$, implies that $|m-k| \geq |j-i|$. Without loss of generality take $i = 0$, so that $u_0 \equiv_\pi u_j$. Then we claim that for any r, $u_r \equiv_\pi u_{r+j}$ and if $u_r \equiv_\pi u_s$ then $j|(s-r)$. To see this, note first that since $u_0 \equiv_\pi u_j$ and $u_0 u_1 \in C_n$, there must be a point w with $u_1 \equiv_\pi w$ and $u_j w \in C_n$. Since $od(u_j) = 1$, w must be u_{j+1}, so $u_1 \equiv_\pi u_{1+j}$. An inductive argument then establishes the first part of the claim. For the second, if $u_r \equiv_\pi u_s$ and $s > r$, then write $s = r+kj+m$, $m < j$. We

know that $u_r \equiv_\pi u_{r+kj} \equiv u_{r+kj+m}$; but this contradicts the minimality of j.

Now, let m be the smallest integer such that $mj > n-1$. If $k \equiv mj \pmod{n}$ then $u_0 \equiv_\pi u_k$, so that $k \geq j$. But $k \leq j$, for otherwise $(m-1)j > n-1$. Thus $k = j$, and $mj = n$. Hence $j|n$, and $\phi(C_n)$ is C_j.

Theorem 2.12

Let $\phi: D \to D'$ be admissible, and suppose that D and D' are both outregular of degree d. Then for any u and v the number n of arcs from u to $\phi^{-1}(\phi(v))$ is the same as the number n' of arcs from $\phi(u)$ to $\phi(v)$.

PROOF:

Since ϕ is full, every arc from $\phi(u)$ to $\phi(v)$ has a preimage, say u_1v_1. Since $\phi(u) = \phi(u_1)$ and ϕ is admissible, there must be a point $\bar{v} \in \phi^{-1}(\phi(v))$ such that $u\bar{v} \in D$. Thus $n \geq n'$. But this inequality holds for each point v' such that $\phi(u)v' \in D'$. Let the set of all such points be $S' = \{v'_0 = \phi(v), v'_1, \ldots, v'_m\}$; note that $\phi^{-1}(S') = \gamma u$. If we denote the number of arcs from $\phi(u)$ to v'_i by n'_i and the number of arcs from u to $\phi^{-1}(v'_i)$ by n_i, then for each $i = 0,\ldots,m$, $n_i \geq n'_i$. But

$$\sum_{i=0}^{m} n'_i = od(\phi(u)) = d \quad \text{and} \quad \sum_{i=0}^{m} n_i = od(u) = d.$$

Thus, for each i, $n_i = n'_i$.

Note that the outregularity restrictions of the theorem are necessary, as can be seen from the admissible homomorphism in Figure 2.9; there are two arcs from v_1 to v_2, but only one arc from u_1 to $\phi^{-1}(v_2)$.

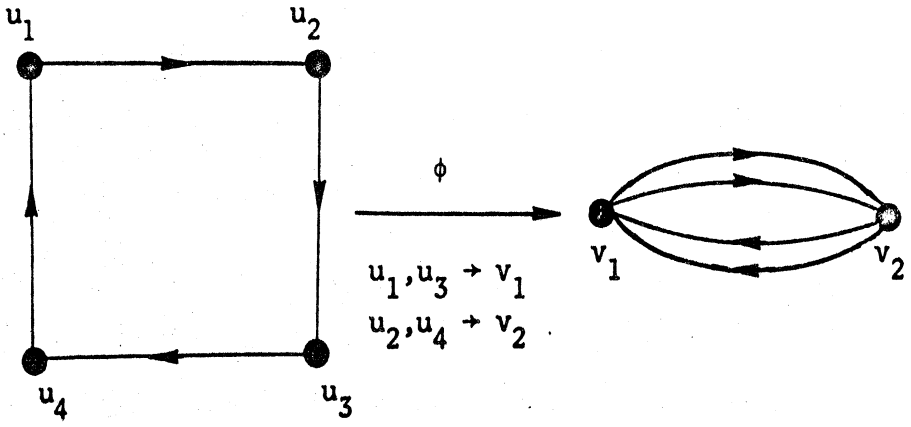


Figure 2.9. An admissible homomorphism.

The existence of an admissible homomorphism with domain D might seem to imply symmetries in D. While this would be hard to state precisely, there is a well-defined converse. An *automorphism of a digraph* D is a map $g:V(D) \overset{\text{onto}}{\to} V(D)$ which satisfies: $uv \in D$ if and only if $g(u)g(v) \in D$. The automorphisms of D form a group, ΓD. Any subgroup $\Gamma' \subseteq \Gamma D$ partitions V(D) into sets $O_1,\ldots,O_r$ such that for each $g \in \Gamma'$, $g(O_i) = O_i$; the O_i are the *orbits* of Γ'.

Theorem 2.13

For any digraph D, the partition π, defined by $v_i \equiv_\pi v_j$ if and only if there is an automorphism $g \in \Gamma(D)$ such that $g(v_i) = v_j$, is admissible.

PROOF:

Suppose $v_i \equiv_\pi v_j$ and $v_iv_k \in D$. If g is an automorphism for which $g(v_i) = v_j$ then v_j is adjacent to $g(v_k)$, and certainly $v_k \equiv_\pi g(v_k)$, so that π is admissible.

Example 2.3

Let D be any digraph with n points in which each point has outdegree t, and suppose that there is an automorphism $g \in \Gamma D$ which is an *n-cycle*; that is, for any $v \in D$ the points $v, g(v), g^2(v),\ldots,g^{n-1}(v)$ are all distinct. Label the points by choosing v_0 arbitrarily and then setting $v_j = g^j(v_0)$. Let $S = \{i_1,\ldots,i_t \mid i_k \leq i_{k+1}\}$ be those integers such that $\gamma v_0 = \{v_{i_1},\ldots,v_{i_t}\}$. We first show that for any $v_j \in D$, the arcs from v_j are $v_jv_{j+i_1},\ldots,v_jv_{j+i_t}$. For, if $v_0v_{i_k} \in D$ then $g^j(v_0)g^j(v_{i_k}) \in D$; that is, $v_jv_{j+i_k} \in D$. But this accounts for t arcs from v_j, and there are only t.

We will refer to such digraphs by listing the i_j and n; i.e., $D = D(i_1,\ldots,i_t,n)$. Now, let $m \mid n$ and define $D'_m(D)$ to be a digraph whose points are the m sets $w_i = \{v_i, v_{i+m}, v_{i+2m},\ldots\}$, where $w_iw_j \in D'_m(D)$ if and only if there are $v_\alpha \in w_i$, $v_\beta \in w_j$ such that $v_\alpha v_\beta \in D$. The w_i, as subsets of V(D), are the orbits of $g^{n/m}$. Thus the map $\phi: D \to D'_m(D)$, which takes each v_j to that set w_i of which it is an element, is admissible. These digraphs $D(i_1,\ldots,i_t,n)$ will be of prime importance in Chapter III.

In Theorem 2.8, we gave a necessary condition for a map ϕ to be admissible. As with walkwise maps, that condition is most useful when ϕ is decomposed into $\phi = \phi_2\phi_1$, where ϕ_2 is an elementary identification. In this case, it is easy to check whether ϕ_2 is admissible. If u and v are not adjacent then γu must equal γv. If, say, uv is an arc then $\gamma u - \{u,v\} = \gamma v - \{u,v\}$ and either vu or vv is an arc.

On the other hand, suppose that $\phi = \phi_2\phi_1$ and ϕ_2 is admissible. What conditions on ϕ_1 will insure that ϕ is admissible?

Theorem 2.14

Let $\phi: D \to F$ be decomposed as $\phi = \phi_2\phi_1$, where $\phi_1: D \to E$ is an elementary identification of u and v to (uv) and $\phi_2: E \to F$ is admissible. Then ϕ is admissible if and only if $\phi(\gamma u) = \phi(\gamma v)$.

PROOF:

Let $W_1 = \gamma u - \gamma v$, $W_2 = \gamma u \cap \gamma v$, $W_3 = \gamma v - \gamma u$.

Then $\phi(\gamma u) = \phi(\gamma v)$ if and only if one of the following holds:

(1) $W_1 \cup W_3 = \emptyset$

(2) $\phi(W_1) \subseteq \phi(W_2) \cup \phi(W_3)$

$\phi(W_3) \subseteq \phi(W_2) \cup \phi(W_1)$

(Necessity): Suppose ϕ is admissible. Since $\phi(u) = \phi(v)$ let $uq_1 \in D$. Then $q_1 \in W_1 \cup W_2$. There must be a $q_2 \in D$ such that $\phi(q_2) = \phi(q_1)$ and $vq_2 \in D$. If $q_1 \in W_2$ there is no problem. Suppose that $q_1 \in W_1$. Then since $vq_2 \in D$, $q_2 \in W_2 \cup W_3$. Thus $\phi(q_1) \in \phi(W_2) \cup \phi(W_3)$, so $\phi(W_1) \subset \phi(W_2) \cup \phi(W_3)$.

(Sufficiency): If (1) holds then ϕ_1 is admissible, so ϕ must be. Otherwise, let $\phi(q_1) = \phi(q_2)$ and $q_1 q_3 \in D$.

Case 1: If neither q_1 nor q_2 is u or v, then $\phi_2(q_1) = \phi_2(q_2)$. Also, $q_1 q_3 \in D$ implies $q_1 \phi_1(q_3) \in E$. So, there is a q_4 in E such that $\phi_2(q_4) = \phi_2(\phi_1(q_3))$ and $q_2 q_4 \in E$. Thus, there is a $q'_4 \in \phi^{-1}{}_1(q_4)$ such that $q_2 q'_4 \in D$ and $\phi(q'_4) = \phi(q_3)$.

Case 2: If $q_1 = u$ and $q_2 = v$, then $q_3 \in W_1 \cup W_2$. If $W_3 = \phi$ then since $\phi(q_3) \in \phi(W_2)$ there is a $q_4 \in W_2$ such that $\phi(q_4) = \phi(q_3)$ and, since $q_4 \in W_2$, $vq_4 \in D$. If W_3 is not empty by the above it suffices to consider the case where $q_3 \in W_1$ and $\phi(q_3) \in \phi(W_3)$. Then there is a $q_4 \in W_3$ such that $\phi(q_4) = \phi(q_3)$; clearly $vq_4 \in D$.

Case 3: If $q_1 = u$ and $q_2 \neq v$ then again $q_3 \in W_1 \cup W_2$. Now, in E, $\phi_2(q_2) = \phi_2((uv))$ and $(uv)\phi_1(q_3) \in E$. Thus there is a $q_4 \in E$ such that $q_2q_4 \in E$ and $\phi_2(q_4) = \phi_2(\phi_1(q_3))$. Since $q_2q_4 \in E$ there is a $q_4' \in \phi_1^{-1}(q_4)$ such that $q_2q_4' \in D$. Also, $\phi(q_4') = \phi_2(q_4) = \phi_2(\phi_1(q_3)) = \phi(q_3)$, so $\phi(q_4') = \phi(q_3)$.

Case 4: $q_1 \neq v$ and $q_2 = u$. Since $q_1q_3 \in D$, $q_1\phi_1(q_3) \in E$. Also, since $\phi(q_1) = \phi(q_2)$, $\phi_2(q_1) = \phi_2((uv))$. So there is a $q_4 \in E$ such that $(uv)q_4$ and $\phi_2(q_4) = \phi_2(\phi_1(q_3))$. Thus there is a $q_4' \in \phi_1^{-1}(q_4)$ such that either uq_4' or vq_4'. Suppose $q_4' \in W_3$; otherwise we are done. Since $q_4' \in W_3$, $\phi(q_4') \in \phi(W_1) \cup \phi(W_2)$. Thus there is a $q_5 \in W_1 \cup W_2$ such that $\phi(q_5) = \phi(q_4')$. Since $\phi(q_4') = \phi_2(q_4) = \phi(q_3)$, $q_4' \equiv q_3 (\mathrm{mod}\ \phi)$. But $\phi(q_5) = \phi(q_4')$; thus $\phi(q_5) = \phi(q_3)$.

Corollary

If ϕ is a map which identifies u and v, write $\phi = \phi_1\phi_2$ where ϕ_1 is the elementary identification of u and v. Then ϕ is admissible if and only if ϕ_2 is admissible and u and v satisfy the condition of the theorem.

The result of applying an elementary identification to a machine is often a nondeterministic machine. Thus, we would not expect to find similar decomposition results for SP homomorphisms.

A *homomorphism* of a digraph D is a map $\phi: D \to D'$ such that if $uv \in D$ then $\phi(u)\phi(v) \in D'$ and $\phi(u) \neq \phi(v)$. The *conjunction* $D_1 \wedge D_2$ of digraphs D_1 and D_2 is a digraph D with $V(D) = V(D_1) \times V(D_2)$, and $(u_1,u_2)(v_1,v_2) \in D$ if and only if $u_1v_1 \in D_1$ and $u_2v_2 \in D_2$. The next

theorem was proved by Hedetniemi for graphs [19].

Theorem 2.15

A pure digraph D is equal to a conjunction $D_1 \wedge D_2$ of pure digraphs if and only if there are full homomorphisms $\phi_1: D \to D_1$, $\phi_2: D \to D_2$ such that the meet of the corresponding induced partitions is the identity partition 0.

PROOF:

If $D = D_1 \wedge D_2$, let ϕ_1 and ϕ_2 be the projection homomorphisms of D to the $D_i: \phi_i(u) = proj_i(u)$. If $\phi_1(u) = \phi_1(v)$ and $u \neq v$ then u and v cannot be adjacent. If u and v are adjacent, then $\phi_1(u)$ and $\phi_1(v)$ are adjacent. If u_1 and v_1 are adjacent in D_1, then, for any $u_2v_2 \in D_2$, $u = (u_1,u_2)$ and $v = (v_1,v_2)$ are adjacent in D, and $\phi_1(u) = u_1$, $\phi_1(v) = v_1$. Thus ϕ_1 is a full homomorphism. A similar proof serves for ϕ_2. If $\phi_1(u) = \phi_1(v)$ and $\phi_2(u) = \phi_2(v)$ then $u = v$, so that induced partitions must have meet equal to 0.

Conversely, if ϕ_1 and ϕ_2 are full homomorphisms of D onto D_1 and D_2 respectively, and the induced partitions have meet equal to zero, then each point u of D can be represented uniquely by the ordered pair $(\phi_1(u),\phi_2(u))$, and this representation defines an isomorphism of D with $D_1 \wedge D_2$. This follows as in [19].

We presented the first part of the proof in detail because we can show that, in many cases, the projection maps are also admissible. A *sink* in a digraph D is a point u such that $od(u) = 0$. In a graph, a sink is called an *isolate*.

Theorem 2.16

Let D_1 and D_2 be pure digraphs, neither of which has a sink, and let $D = D_1 \wedge D_2$. Then the projection homomorphisms

$\phi_i: D \to D_i$ are admissible.

PROOF:

Let $\phi_1((u_1,u_2)) = \phi_1((v_1,v_2))$, so that $u_1 = v_1$. Suppose $(u_1,u_2)(w_1,w_2) \in D$. Then $u_1w_1 \in D_1$ and $u_2w_2 \in D_2$. Since v_2 is not a sink there is a point $\overline{w}_2$ such that $v_2\overline{w}_2 \in D_2$. Now $\phi_1(w_1,\overline{w}_2) = \phi_1(w_1,w_2)$, and $(v_1,v_2)(w_1,\overline{w}_2) \in D$, since $v_1 = u_1$. Thus, ϕ_1 is admissible. Similarly, ϕ_2 is admissible.

Corollary

If G_1 and G_2 are graphs without isolates and if $G = G_1 \wedge G_2$, then the projection homomorphisms $\phi_i: G \to G_i$ are admissible.

Since we know from Theorems 2.7 and 2.1 that for any admissible homomorphism ϕ of a digraph D, $m(D) \geq m(\phi(D))$, it is now natural to ask what the cycle rank of the conjunction of two digraphs is.

Theorem 2.17

Let D_1 and D_2 be strong digraphs with p_i points, and $m(D_i) = k_i+1$. If $D_1 \wedge D_2$ has δ strong components, then $m(D_1 \wedge D_2) = (k_1k_2+\delta) + p_1k_2 + p_2k_1$.

Note: If the greatest common divisor of the lengths of all closed walks in D_i is d_i, then $\delta = \gcd(d_1,d_2)$ (see [16]).

PROOF:

Let D_i have q_i arcs. Then $D_1 \wedge D_2$ has q_1q_2 arcs, and $m(D_1 \wedge D_2) = q_1q_2 - p_1p_2 + \delta$. Now $(q_1-p_1)(q_2-p_2) = k_1k_2$, so that

$$q_1q_2 - p_1q_2 - q_2p_1 + p_1p_2 = k_1k_2,$$

and

$$\begin{aligned} q_1q_2 - p_1p_2 + \delta &= (k_1k_2+\delta) + p_1q_2 + p_2q_1 - 2p_1p_2 \\ &= (k_1k_2+\delta) + p_1(k_2+p_2) + p_2(k_1+p_1) - 2p_1p_2 \\ &= (k_1k_2+\delta) + p_1k_2 + p_2k_1. \end{aligned}$$

Corollary

If G_1 and G_2 are connected graphs with cycle ranks k_i+1, the cycle rank of $G_1 \wedge G_2$ is $(k_1k_2+\delta)+q_1q_2+p_1k_2+p_2k_1$, where $\delta = 2$ unless one of G_1, G_2 has an odd cycle, in which case $\delta = 1$.

A homomorphism of a graph G onto a graph with n points defines a *coloring* of G, an assignment function from V(G) onto a set of n colors such that no two adjacent points are assigned the same color; conversely, every coloring of the points of a graph defines a homomorphism, where two points are identified just when they have the same color. The minimum number of colors in any coloring of a graph G is the *chromatic number*, $\chi(G)$, and the image under the associated homomorphism is the complete graph $K_{\chi(G)}$. The homomorphisms associated with colorings are, in general, not admissible. It might well be of interest to develop a theory of admissible colorings of graphs. We will present one connection between colorings and admissible homomorphisms.

A graph G with chromatic number $\chi(G) = n$ is said to be *uniquely n-colorable* if there is exactly one homomorphism from G onto K_n (see [4], [14]).

Theorem 2.18

If G is uniquely n-colorable then the homomorphism $\phi: G \to K_n$ is admissible.

PROOF:

It is shown in [4] that if G is uniquely n-colorable then in any coloring of G with n colors, every point v is adjacent to at least one point of every color different from that assigned to v. Thus if

$\phi: G \to K_n$ is the unique homomorphism from G onto K_n, suppose that $\phi(u) = \phi(v)$ and $uw \in G$. Then w is assigned a color different from that of u. Since u and v have the same color, it follows, from the result just quoted, that v is adjacent to a point $\overline{w}$ which has the same color as w; i.e., $\phi(\overline{w}) = \phi(w)$. Thus, ϕ is admissible.

CHAPTER III

REALIZATION WITH FEEDBACK ENCODING

A major concern of classical automata theory has been the realization of either the state behavior or the input-output behavior of machines. The main thrust of the research has always been to realize the given machine by a loop-free network; that is, by a network for which the digraph obtained by taking the modules as points and the flow lines between them as directed arcs has no directed cycles. In such a network there is no feedback, except possibly within modules.

As noted by Holland [22], "feedback is a prominent structural feature of most systems which exhibit complex behavior." Nevertheless, as Zeigler [36] showed, many networks can be modelled by cascades; i.e., feedback-free networks. The modules of the new network are taken to be the strong components of the original network. This construction, of course, greatly increases the complexity of the component modules.

There are advantages both to excluding and including feedback in networks. On the one hand, there is a large body of techniques available for analyzing feedback-free systems and also for deriving feedback-free realizations of a given behavior [17]. On the other hand, a feedback-free realization can be artifactual: it often consists of merely masking those parts of the network with feedback, the strong components, and considering them as "black boxes". We thus have a trade-off between simplicity of the components and simplicity of the interconnections. It is certainly to be expected that by restricting in some way the form that feedback is allowed to take we can strike a suitable balance between the two extremes.

Thus, one motivation of the study we are about to undertake is the desirability of defining a restricted form of feedback which will prove tractable in both theory and application. Another is the following. Suppose that we have some given state behavior which we want to realize by a sequential machine. If the behavior is so realizable then, in fact, techniques exist for doing this in an optimal way: the algorithm for deriving reduced machines, first discussed by Moore [30]. However, it is often the case that additional restrictions are placed on the realization. For example, one might want the realizing machine to be expressible as a cascade of modules from a well-defined set. This, of course, has been studied by Zeiger [35], Krohn and Rhodes [24], and others. In general, it turns out that the behavior of the cascade which is derived properly includes the desired behavior. For another example, take the fault diagnosis problem, a small segment of which is discussed in Chapter V. Here we might want a realizing machine, for example, which, if it fails in some way, allows us to determine the cause or location of the fault, or perhaps even to correct it. Even the simplest fault diagnosis properties cannot, in general, be incorporated into the reduced machine which realizes the given behavior, and it is usually necessary for the realizing machine to have more states or inputs (and thus, more state circuitry) than the reduced machine. A second motivation for the study to follow, therefore, is to develop a form of realization which will allow us to add additional constraints and at the same time, will not cause the same state set growth as the classical theory.

As presented in Chapter I, a realization of a machine $M' = \langle Q', I', \delta' \rangle$ by a machine $M = \langle Q, I, \delta \rangle$ is defined in terms of two maps

$$\phi: Q \overset{\text{onto}}{\to} Q'$$

and

$$h: I' \to I$$

such that, for any $q \in Q$ and any $x' \in I'$

$$\phi(q)x' = \phi(qh(x')). \tag{3.1}$$

The function h can be considered to be a combinatorial (i.e., memoryless) module which encodes inputs to M' into inputs to M (see Figure 3.1).

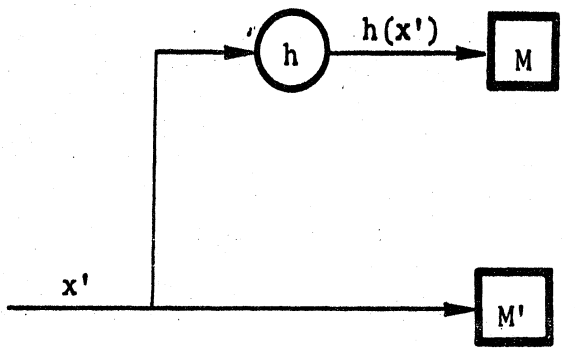


Figure 3.1. Schema for realization.

We will say that M *realizes* M' with *feedback encoding* if there are maps

$$\phi: Q \overset{\text{onto}}{\to} Q'$$

and

$$h: Q \times I' \to I$$

such that for all $q \in Q$ and all $x' \in I'$

$$\phi(q)x' = \phi(qh(q,x')). \tag{3.2}$$

Thus the combinatorial circuit h has as inputs both the input to the realized machine M' and the current state of the realizing machine M. This is expressed diagrammatically in Figure 3.2.

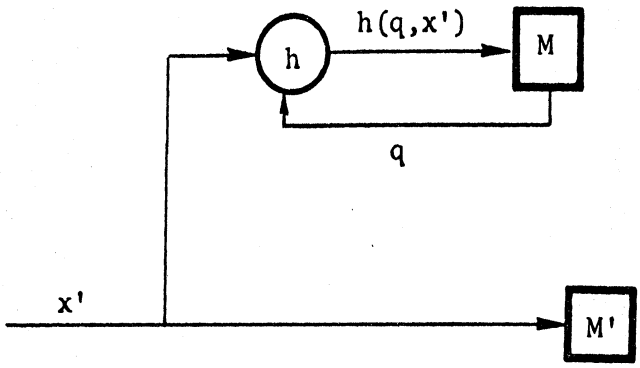


Figure 3.2. Schema for realization with feedback encoding.

We will often find it convenient to consider the map h to be a collection $\{h_q | q \in Q\}$ of maps, where each h_q maps I' to I. Thus, (3.2) could be restated as

$$\phi(q)x' = \phi(qh_q(x')) \tag{3.3}$$

If all the maps h_q were identical, then h would simply be a function from I' to I. We thus have the following simple observation:

Theorem 3.1

If M realizes M' then M realizes M' with feedback encoding.

We now give some examples of pairs of machines M and M' such that M realizes M' with feedback encoding, but not without.

Example 3.1

Let $M_1 = \langle Q, I_1, \delta_1 \rangle$ be the modulo three counter in Figure 3.3, and let $M_2 = \langle R, I_2, \delta_2 \rangle$ be the machine in Figure 3.4. Note that the only actual difference between the two machines is that the input labels at state r_3 have been permuted.

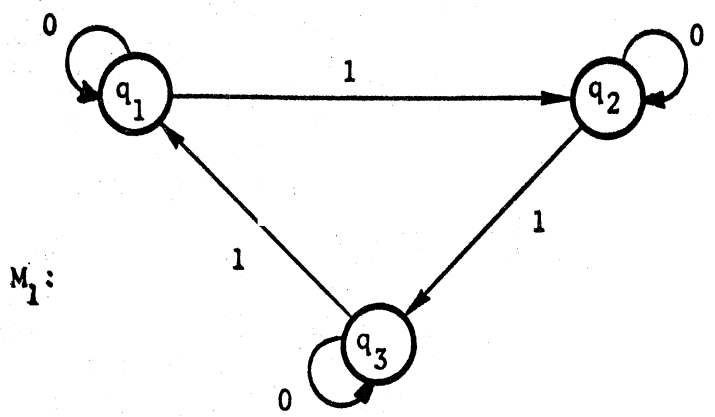


Figure 3.3. A modulo three counter.

To show that M_2 does not realize M_1 we would in general have to consider all maps $\phi: R \overset{\text{onto}}{\rightarrow} Q$ and $h: I_1 \rightarrow I_2$; however, since M_1 is highly symmetric, it is sufficient to look at the map ϕ_1 such that $\phi_1(r_i) = q_i, i = 1,2,3$.

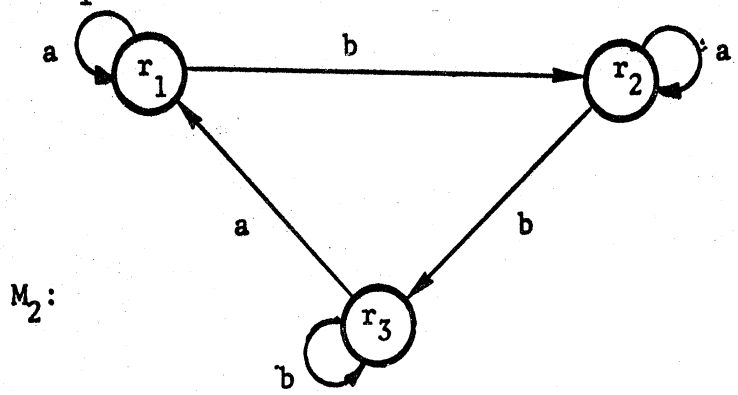


Figure 3.4. A machine which does not realize a modulo three counter.

There are then four candidates for the h to consider. Although it is quite clear, as it will be other examples, that only one-to-one maps need be considered, in this one case we will examine all the h maps for completeness. For each proposed map, we will give an example showing how (3.1) is violated.

Map h	Violation of (3.1)
1. h(0) = h(1) = a	$\phi_1(r_1)1 = q_1 1 = q_2 \neq q_1 = \phi_1(r_1 a)$
2. h(0) = h(1) = b	$\phi_1(r_1)0 = q_1 0 = q_1 \neq q_2 = \phi_1(r_1 b)$
3. h(0) = b h(1) = a	$\phi_1(r_2)1 \quad 1 = q_3 \neq q_2 = \phi_1(r_2 a)$
4. h(0) = a h(1) = b	$\phi_1(r_3)1 = \quad = q_2 \neq q_3 = \phi_1(r_3 b)$

Thus M_2 does not realize M_1. It is clear that the most likely candidate for a suitable h map was the four, which is because the actions of a and b at state r_3 are not symmetric to their actions at r_1 and r_2. By following feedback encoding, we can clear up this difficulty. Let $h: R \times I_1 \to I_2$ be the map

h:

	0	1
r_1	a	b
r_2	a	b
r_3	b	a

By the above discussion it is easy to see that this map, together with the map ϕ_1 which takes each r_i to q_i, satisfies (3.2) in all cases, so that M_2 does indeed realize M_1 with feedback encoding.

Example 3.2

Consider $M_3 = \langle P, I_3, \delta_3 \rangle$ from Figure 3.5, and M_2 from the previous example.

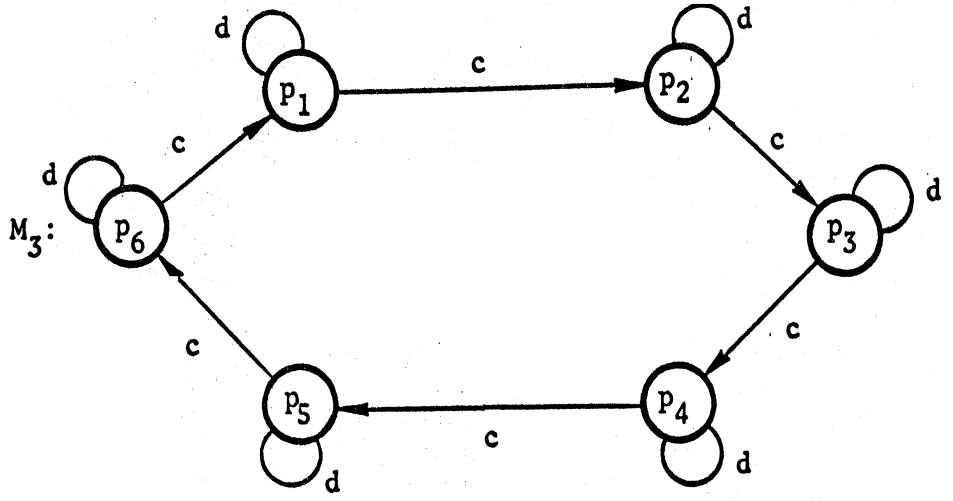


Figure 3.5. A modulo six counter.

We will try to find maps ϕ_2 and h which define a realization of M_2 by M_3. By the symmetry of M_3, it does not matter which state of M_2 we take to be the image of p_1 under ϕ_2, so take $\phi_2(p_1) = r_1$. By (3.1),

$$r_1 b = r_2 = \phi_2(p_1 h(b)) \text{ and } r_1 a = r_1 = \phi_2(p_1 h(a)).$$

Thus h(a) must be d. If h(b) = d then $\phi_2(p_1 d) = \phi_2(p_2) = r_1$, and similarly, for all i, $\phi_2(p_i) = r_1$, which is impossible. Then h(b) = c and $\phi_2(p_2) = r_2$. Then $r_3 = r_2 b = \phi_2(p_2 c) = \phi_2(p_3)$. But then $r_3 a = r_1$ but $\phi_2(p_3 h(a)) = \phi_2(p_3 d) = \phi(p_3) = r_3$, so M_3 cannot realize M_2. However, if we now define ϕ_2 and h by the tables

ϕ_2	
p_1	r_1
p_2	r_2
p_3	r_3
p_4	r_1
p_5	r_2
p_6	r_3

h	a	b
p_1	d	c
p_2	d	c
p_3	c	d
p_4	d	c
p_5	d	c
p_6	c	d

then we get a realization with feedback encoding of M_2 by M_3. Next we display two tables: the entries in the first are the values $\phi_2(p_j)x$ and those in the second are $\phi_2(p_j h_{p_j}(x))$, where $x \in \{a,b\}$. Since the tables

$\phi_2(p_j)x$	a	b
p_1	r_1	r_2
p_2	r_2	r_3
p_3	r_1	r_3
p_4	r_1	r_2
p_5	r_2	r_3
p_6	r_1	r_3

$\phi_2(p_j h_{p_j}(x))$	a	b
p_1	r_1	r_2
p_2	r_2	r_3
p_3	r_1	r_3
p_4	r_1	r_2
p_5	r_2	r_3
p_6	r_1	r_3

are identical, ϕ_2 and h define a realization with feedback encoding. Now let $\phi = \phi_2\phi_1 : P \to Q$ and let $\bar{h}: P \times I_1 \to I_3$ be defined by $\bar{h}(p_j,x) = h_{p_j}(h_{\phi_2(p_j)}(x))$. As a map from $P \times I_1 \to I_2$, $h_{\phi_2(p_j)}$ has the table

$h_{\phi_2(p_j)}$	0	1
p_1	a	b
p_2	a	b
p_3	b	a
p_4	a	b
p_5	a	b
p_6	b	a

Then $\bar{h}(p_j,x)$ has the table

	0	1
p_1	d	c
p_2	d	c
p_3	d	c
p_4	d	c
p_5	d	c
p_6	d	c

We see that $\bar{h}:P \times I_1 \to I_3$ is in fact independent of the state of M_3, so that the composition of the realizations with feedback encoding of examples 3.1 and 3.2 is a realization of M_1 by M_3. It is not true in general that the "composition" of realizations with feedback encoding is a realization. On the other hand, such a composition is always a realization with feedback encoding.

Theorem 3.2

Let M_3 realize M_2 with feedback encoding and let M_2 realize M_1 with feedback encoding. Then M_3 realizes M_1 with feedback encoding.

PROOF:

Let machine M_i have state-set Q_i and input-set I_i. Then we have maps

$$\phi:Q_2 \overset{\text{onto}}{\to} Q_1 \qquad \psi:Q_3 \overset{\text{onto}}{\to} Q_2$$

$$h:Q_2 \times I_1 \to I_2 \qquad g:Q_3 \times I_2 \to I_3$$

such that

$$\text{and} \qquad \begin{aligned} \phi(q_2)x_1 &= \phi(q_2 h(q_2,x_1)) && q_2 \in Q_2,\ x_1 \in I_1 \\ \psi(q_3)x_2 &= \psi(q_3 g(q_3,x_2)) && q_3 \in Q_3,\ x_2 \in I_2. \end{aligned}$$

Then, for any $q_3 \in Q_3$ let q_2 be $\psi(q_3)$ and let q_1 be $\phi(q_2)$. For any $x_1 \in I_1$, $q_1x_1 = \phi(q_2 h(q_2,x_1))$. But $h(q_2,x_1) \in I_2$, so $\phi(q_2 h(q_2,x_1)) = \phi(\psi(q_3 g(q_3,h(q_2,x_1))))$. Thus, letting $\tau(q_3) = \phi(\psi(q_3))$ and $f(q_3,x_1) = g(q_3,h(\psi(q_3),x_1))$, we see that the pair (τ,f) defines a realization with feedback encoding of M_1 by M_3.

Example 3.3

Let M be the reset machine displayed in Figure 3.6. We will show that M realizes with feedback encoding the machine M_1 of Example 3.1.

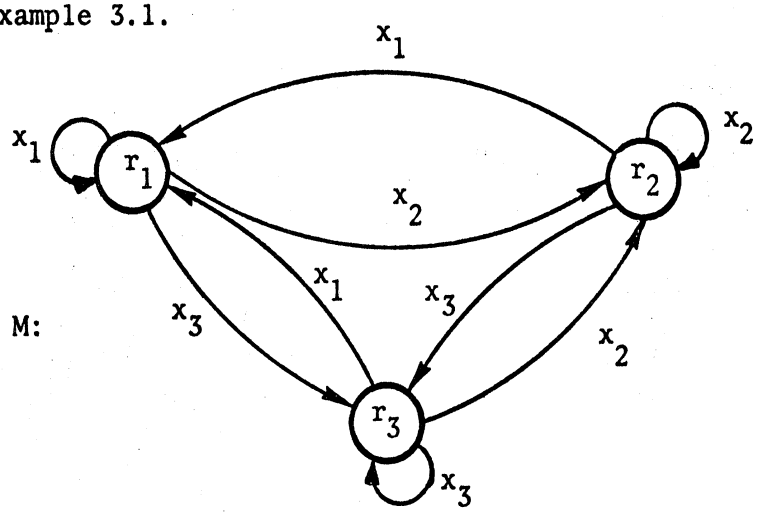


Figure 3.6. A reset machine.

Let ϕ be the map $\phi: r_i \to q_i$, and define h as follows: if $q_j y = q_k$, where $y \in I_1$, then $h(r_j,y) = x_k$. Since machine M is a reset machine, so that for any state r_j, $r_j x_k = r_k$, it then follows that if $\phi(r_j)y = q_j y = q_k$ then $r_j h(r_j,y) = r_k$, so that $\phi(r_j)y = \phi(r_j h(r_j,y))$.

Theorem 3.3

Let $M' = \langle Q', I', \delta' \rangle$ be any machine and let $M = \langle Q, I, \delta \rangle$ be a machine such that

i. $|Q| \geq |Q'|$

ii. for each $q_i \in Q$ there is an input x_i such that for all q_j, $q_j x_i = q_i$.

Then M realizes M' with feedback encoding.

PROOF:

Let ϕ be any map from Q onto Q'. Then, as in Example 3.3, if $\phi(q_i)y' = \phi(q_j)$ define $h(q_i, y')$ to be the reset input x_j; the resulting pair (ϕ, h) defines a realization with feedback encoding. Of course, for a given q_i and x' there may be many states whose image under ϕ is $\phi(q_i)x'$. It does not matter which of these is taken to be the q_i of the proof.

In one sense, Theorem 3.3 would seem to be quite impressive. We know from the Krohn-Rhodes results [24] that reset machines are quite simple, and we have shown that they are sufficient to realize any machine with feedback encoding. On second thought, however, we must have quite a different opinion. Sequential machines can have quite complicated behavior, and it is clear that what we are doing in Theorem 3.3 is lumping all the behavior into the h-map. Any sequential machine can, in fact, be modelled by a machine whose memory is a reset machine and whose transition function is computed by a memoryless circuit. While it is often desirable to transfer some of the complexities of a machine to a combinatorial circuit in this way there is a, perhaps ill-defined, point at which it becomes fatuous: if our purpose is to study the

complexities of sequential machines, i.e., machines with memory, it does us no good to define a canonical form in which all the complexity resides in a memoryless component.

It appears that the type of feedback we have defined does not quite strike the balance alluded to in the introduction to this chapter. We must therefore place some restrictions on the types of feedback encoding which we shall allow.

Given machines $M = \langle Q, I, \delta \rangle$ and $M' = \langle Q', I', \delta' \rangle$ we will say that a map $h: Q \times I' \to I$ is a *type I feedback encoder* if, for each $q \in Q$, the map $h_q: I' \to I$ is one-to-one and onto. Recalling Example 3.3, it is clear that machine M cannot realize the modulo three counter M_1 with type I feedback encoding. Unfortunately, this example obscures the point somewhat: M cannot realize M_1 with type I feedback encoding because $|I| > |I_1|$. Since, in studying realization, we are interested in whether one machine can capture the behavior of another, we certainly do not want to say that one machine cannot realize another solely because the cardinalities of their input sets are not equal. Even in the classical realization theory, where the encoder is independent of the state, this restriction does not appear. Thus, we relax the previous definition in the following way: a map $h: Q \times I' \to I$ is a *type II feedback encoder* if there is a subset $\bar{I} \subseteq I$ such that for each $q \in Q$ the range of the map h_q is $\bar{I}$ and h_q is one-to-one. Even with a type II encoding M cannot realize M_1 in Example 3.3.

In the sequel we will rarely use the adjectives "type I" and "type II". Whenever we refer to feedback encoding we will assume it to be type I, unless explicitly stated otherwise. Now, since if M realizes M' with type II feedback encoding it is clear that a submachine of M

realizes M' with type I feedback encoding, most of the results we obtain for type I encodings will readily translate into results for type II encodings, and we will rarely make this translation explicit.

A word is in order here about the restriction that the maps h_q be one-to-one. Example 3.3 shows that this cannot be relaxed very much before the concept of realization with feedback encoding becomes both unrestricted and unrealistic. However, in practice, in applying the concept of realization with feedback encoding it might be desirable to introduce slight relaxations, depending of course, on the resultant increase in complexity of the h map.

We are now ready to begin our study of realization with feedback encoding.

We first wish to point out that the composition property developed in Theorem 3.2 has not been lost. Consider the proof of that theorem, and suppose that both h and g are type I feedback encodings. We wish to show that $f:Q_2 \times I_1 \to I_3$ is type I; i.e., that for each q_3, f_{q_3} is both one-to-one and onto. For each q_3 and each $x_3 \in I_2$, we know that there is an $x_2 \in I_2$ such that $g(q_3,x_2) = x_3$. Also, there is an x, such that $h(\psi(q_3),x_1) = x_2$. Thus $f(q_3,x_1) = g(q_3,h(\psi(q_3),x_1)) = x_3$, and so f_{q_3} is onto. Now, since $f_{q_3}(x_1) = g(q_3,h(\psi(q_3),x_1))$, and since both g_{q_3} and $h_{\psi(q_3)}$ are one-to-one, it follows immediately that f_{q_3} is one-to-one. We have thus proved:

Theorem 3.4

Let M_3 realize M_2 with feedback encoding and let M_2 realize M_1 with feedback encoding. Then M_3 realizes M_1 with feedback encoding.

This theorem, of course, refers to type I feedback encodings. On the other hand, if one of the realizations is of type II, then so will be the composed relation.

We next show that with the restrictions we have introduced, we have limited the power of the concept. The next theorem is important for another reason as well; it demonstrates a relationship between realization with feedback encoding and the structures of the state graphs of the realized and realizing machines. Recall from Chapter II that for any machine M, D(M) is its underlying digraph.

Theorem 3.5

If M realizes M' with feedback encoding then there is an admissible homomorphism from D(M) onto D(M'). Conversely, if there is an admissible homomorphism from D(M) onto D(M'), and if $|I| = |I'|$ then M realizes M' with feedback encoding.

PROOF:

Let the realization be defined by maps $\phi:Q \to Q'$ and $h:Q \times I' \to I$. Suppose that $\phi(u) = \phi(v)$ and that $uw \in D(M)$. We must show that there is a $\overline{w}$ such that $v\overline{w} \in D(M)$ and $\phi(\overline{w}) = \phi(w)$. Since $uw \in D(M)$ there is an input x such that $ux = w$. Since $h_u:I' \to I$ is onto there is an $x' \in I'$ such that $h_u(x') = x$. Then $\phi(u)x' = \phi(w)$ so that $\phi(v)x' = \phi(w)$.

Let $\overline{w} = vh_v(x')$. Then by (3.3), $\phi(\overline{w}) = \phi(w)$, and, since $\overline{w} = vh_v(x')$, $v\overline{w}$ is certainly an arc of D(M). Therefore, ϕ considered as a mapping from D(M) to D(M') will be admissible once, as in Theorem 2.9, we can verify that it is full. Since M realizes M' with feedback encoding, if there is a single arc $u'v' \in D(M')$ then this arc must have a pre-image arc in D(M). But if there is more than one arc between

u' and v' in D(M') then there are inputs, say $x'_1,\ldots,x'_r$, such that $u'x'_1 = u'x'_2 = \ldots = v'x'_r = v'$. Then for each $u \in \phi^{-1}(u')$ there are $v_1,\ldots,v_r$ such that $uh_u(x'_i) = v_i$ and $\phi(v_i) = v'$, for i=1,...,r. Then each arc $uv_i \in D(M)$ is a pre-image for one of the arcs between u' and v'. While it may be the case that the v_i are not all distinct, since h_u is one-to-one we are assured that there will be r distinct arcs from u to points in $\phi^{-1}(v')$. Thus, $\phi:D(M) \to D(M')$ is admissible.

For the converse, suppose ϕ is an admissible map from D(M) onto D(M') and that $|I| = |I'|$. Let u be a state of M and suppose that

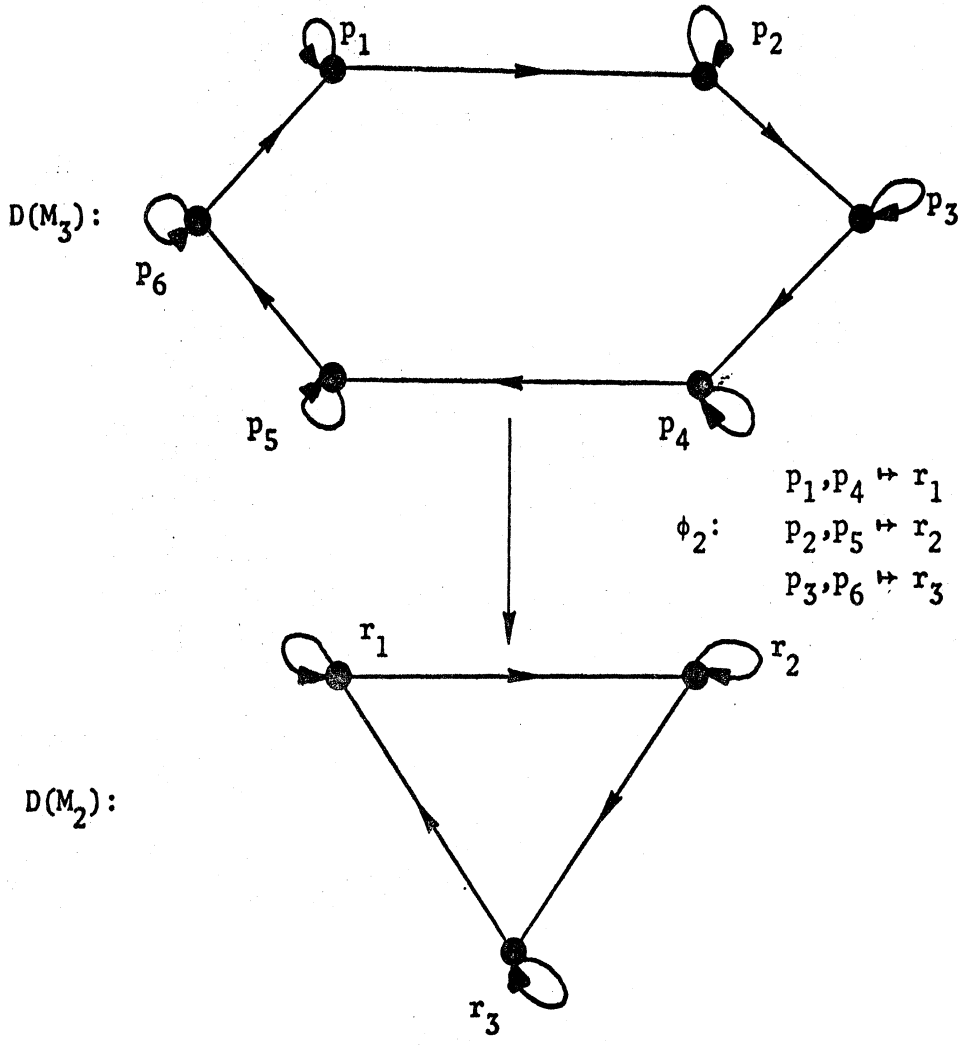


Figure 3.7: An admissible homomorphism based on Example 3.2.

$\phi(u)x' = w'$. Since ϕ is a map from D(M) to D(M') there must be states $\overline{u}$ and $\overline{w}$ in M such that $\phi(\overline{u}) = \phi(u)$, $\phi(\overline{w}) = w'$, and $\overline{uw} \in D(M)$. But then, by admissibility, there is a $w \in M$ such that $\phi(w) = w'$ and $uw \in D(M)$. Thus, we define $h_u(x')$ to be that input x which induces the arc uw. It is clear that once we verify that this assignment can be done so that $h_u: I' \to I$ is one-to-one and onto the pair (ϕ,h) will define a realization with feedback encoding. But, the one-to-one and onto properties will be satisfied just when the cardinality of the set of arcs from u' to w' is the same as the cardinality of the set of arcs from u to points in $\phi^{-1}(w')$, and we proved this property of admissible maps in Theorem 2.12. Thus, M realizes M' with feedback encoding.

As we noted above, a type II feedback encoding is associated with realization by a submachine. The implication there was that we were discussing *spanning* submachines; submachines with the same state set but restricted input set. However, as in the classical theory of realization we want to allow a machine M' to be realized with feedback encoding by a machine M, where the actual realization is done by a submachine of M which might have either fewer states or fewer inputs, or both. Theorem 3.5 carries through to cover these generalizations in a straightforward manner.

Corollary

If M' is realized with feedback encoding by a submachine $\overline{M}$ of M then there is an admissible homomorphism from $D(\overline{M})$ onto D(M').

It is sometimes common in the classical realization theory to say that M realizes M' when, in fact, the realization is by a submachine of M. We will, in general, only say that M realizes M' with feedback

encoding when the realization is by all of M.

Corollary

If ϕ is an SP homomorphism from M to M' and $|I| = |I'|$ then ϕ is an admissible homomorphism from D(M) to D(M').

PROOF:

We know that $\phi: M \to M'$ is an SP homomorphism if and only if M realizes M'. But any realization is trivially a realization with feedback encoding, as we saw in Theorem 3.1.

Although Examples 3.1 and 3.2 were presented before we had the concept of type I feedback encoding, the realizations in each are easily seen to involve type I feedback encodings. Thus, the converse to the second corollary is certainly false.

We have not, thus far, distinguished the situations in which ϕ is an isomorphism and that in which ϕ is a homomorphism: in the former case ϕ and h define an *isomorphic realization with feedback encoding*, and in the latter, a *homomorphic realization with feedback encoding*. As we mentioned in the introductory remarks to this chapter, we will eventually wish to show that, by introducing feedback encoding, realizations with extra constraints can sometimes by achieved with less cost, in terms of the switching circuitry, than with the conventional concept of realization. In particular, in Chapter V we will give instances in which such constrained realizations with feedback encoding can be achieved by machines with no more states than the reduced machine which exhibits the behavior we wish to realize. To this end, we will essentially restrict our study to isomorphic realizations with feedback encoding.

Corollary

If M isomorphically realizes M' with feedback encoding, then $D(M) \cong D(M')$.

In the case that $D(M) \cong D(M')$, Hedetniemi and Fleck [20] say that M and M' are *graph-isomorphic*.

Much of the classical theory is concerned with realizing machines by cascades of other machines. We can also study realization by cascades, where of course the realization will involve feedback encoding. Most of the theorems we derive will be seen to generalize results about realization without feedback encoding.

Lemma 3.1

If a cascade $M_1 \circ_Z M_2$ realizes M isomorphically with feedback encoding, then there is an admissible homomorphism from D(M) onto $D(M_1)$.

PROOF:

Let the realization with feedback encoding be defined by maps $\phi: Q_1 \times Q_2 \to Q$ and $h: (Q_1 \times Q_2) \times I \to I_1$, so that for any input $x \in I$ and any state (q_1,q_2) in $Q_1 \times Q_2$, $\phi((q_1,q_2))x = \phi((q_1,q_2)h((q_1,q_2),x))$. Let $\rho = \phi^{-1}$, and for a state $q \in Q$ let $q(j)$ be the state $\text{proj}_j\rho(q)$. Now, define an equivalence relation on the states of Q by $q \equiv q'$ if and only if $q(1) = q'(1)$.

Suppose that $q_1 \equiv q_2$ and that for some $x \in I$, $q_1x = q_3$. Let

$$S_1 = \{\text{proj}_1\rho(q_1y) \mid y \in I\} = \{q_1(1)h(\rho(q_1),y) \mid y \in I\}$$

and

$$S_2 = \{\text{proj}_1\rho(q_2y) \mid y \in I\} = \{q_2(1)h(\rho(q_2),y) \mid y \in I\}.$$

Since $q_1(1) = q_2(1)$ by hypothesis, and since $\{h(\rho(q_1),y) \mid y \in I\} = \{h(\rho(q_2),y) \mid y \in I\}$, it follows immediately that

$$S_1 = \{q_1(1)h(\rho(q_1),y)\} = \{q_1(1)h(\rho(q_2),y)\} = S_2.$$

Thus there is some state $q_4 \in Q$ and an input $\bar{y} \in I$ such that $q_2\bar{y} = q_4$ and $q_4(1) = q_3(1)$; i.e., $q_4 \equiv q_3$. Thus $\text{proj}_1\rho: Q \to Q_1$ defines an admissible homomorphism from D(M) onto $D(M_1)$.

This lemma also follows from the Hartmanis and Stearns results [17], for M_1 induces an SP partition on the cascade, and this converts to an admissible partition on M.

Corollary

If M' is a submachine of a cascade $M_1 \circ_Z M_2$ such that $\{proj_1(q') | q' \in Q'\} = Q_1$ and if M' realizes a machine M isomorphically with feedback encoding, where $|I'| = |I_1|$, then there is an admissible homomorphism from D(M) to $D(M_1)$.

Lemma 3.2

Suppose there is an admissible homomorphism from D(M) to $D(M_1)$, where $|I| = |I_1|$. Then there is a machine M_2 such that M can be isomorphically realized with feedback encoding by a submachine of a cascade $M_1 \circ_Z M_2$.

PROOF:

Suppose that ϕ is an admissible homomorphism from D(M) onto $D(M_1)$. We must exhibit a machine M_2 and a connecting map Z such that $M_1 \circ_Z M_2$ isomorphically realizes M with feedback encoding. Now, the map ϕ induces a partition π on the states of M; suppose that there are r partition classes P_i and that the largest of these contains s states. Number the states in each P_i as $1,2,\ldots,n_i$, where $n_i \leq s$, and form a new partition π' with s blocks B_i, where B_i consists exactly of those states, at most one from each P_j, which were numbered i. Then each state $q \in M$ can be associated with exactly one pair (P_i,B_j) of blocks such that $q \in P_i \cap B_j$. Clearly the blocks P_i are identifiable with the states of M_1. We know from Theorem 3.5 that there is a map $h: Q \times I_1 \rightarrow I$ which is one-to-one and onto and which, taken together with ϕ, defines a

realization with feedback encoding of M_1 by M. We can represent the state of M associated with blocks P_i and B_j by q_{ij}. Now, suppose under some input x, $q_{ij}x = q_{km}$. Then, by (3.3), $\phi(q_{ij})h_{q_{ij}}^{-1}(x) = \phi(q_{km})$; we are permitted to refer to $h_{q_{ij}}^{-1}$ since h is a Type I feedback encoding.

Machine M_2 will have the blocks B_j for its states. Now if $q_{ij}x = q_{km}$, and $x' = h_{q_{ij}}^{-1}(x)$, we will want $(P_i,B_j)x' = (P_ix',B_jZ(P_i,x')) = (P_k,B_jZ(P_i,x')) = (P_k,B_m)$. We already know that $P_ix' = P_k$, so we must define M_2 and Z in such a way that $B_jZ(P_i,x') = B_m$. Let M_2 have r inputs y_i. If $(P_i,B_j)x' = (P_k,B_m)$ then in M_2 we define $B_jy_k = B_m$; since not every pair (P_i,B_j) defines a state of M, this may leave us with don't-care conditions, which can be assigned arbitrarily. Now we define $Z(P_i,x') = y_k$, where $P_ix' = P_k$. With these definitions it follows that if $q_{ij}x = q_{km}$ then $(P_i,B_j)x' = (P_ix',B_jZ(P_i,x')) = (P_k,B_jy_k) = (P_k,B_m)$. If ρ is the map which assigns to (P_i,B_j) the state q_{ij} then for each $x \in I$, $\rho((P_i,B_j))x = \rho((P_i,B_j)h_{q_{ij}}^{-1}(x))$. Thus if $g:(\{P_i\}x\{B_j\}) \times I$ is defined by $g((P_i,B_j),x) = h_{q_{ij}}^{-1}(x)$, ρ and g will define an isomorphic realization with feedback encoding of M by that submachine of $M_1 \circ_Z M_2$ consisting of the pairs of states (P_i,B_j) for which $P_i \cap B_j \neq \phi$.

Combining the lemmas, we have

Theorem 3.6

A machine M can be isomorphically realized with feedback encoding by a submachine M' of a cascade $M_1 \circ_Z M_2$, where $|I'| = |I_1|$, if and only if there is an admissible homomorphism from D(M) onto the subdigraph of $D(M_1)$ induced by the states in $\{proj_1(q')|q' \in Q'\}$.

Corollary

A machine M can be isomorphically realized with feedback encoding by a submachine of a cascade $M_1 \circ_Z M_2$ if and only if there is an admissible homomorphism from D(M) onto the digraph of a submachine of M_1.

Having established Theorem 3.6, it is natural to investigate what happens if the feedback to the h map in a realization with feedback encoding by a cascade is from only one of the components of the cascade. Unfortunately, as we shall see, there does not seem to be too much to say about such situations.

Theorem 3.7

Let M and M_1 be machines. Then M has an SP homomorphic image which is graph-isomorphic to M_1 if and only if M can be isomorphically realized with feedback encoding by a cascade M_1 o_Z M_2 in such a way that the encoding map h has domain Q_1 x I.

PROOF:

If M_1 is graph-isomorphic to an SP homomorphic image $\overline{M}_1$ of M then in any isomorphic realization of M by a cascade $\overline{M}_1$ o_Z M_2, $\overline{M}_1$ can be replaced by M_1 together with the appropriate feedback encoder, which, of course, is independent of the state of M_2.

On the other hand, let the realization with feedback encoding be defined by the maps $\phi:Q_1 \times Q_2 \to Q$ and $h:Q_1 \times I \to I_1$. Let $\rho = proj_1\phi^{-1}$ be the admissible homomorphism guaranteed by Theorem 3.5; $\rho:Q \to Q_1$.

As in the proof of Lemma 3.1, for each state q_i, $\phi^{-1}(q_i) = (q_i(1),q_i(2))$. Now, suppose that $\rho(q_1) = \rho(q_3)$, $q_1x = q_2$ and $q_3x = q_4$. Then $\phi((q_1(1),q_1(2)))h(q_1(1),x) = \phi((q_2(1),q_2(2)))$ and $\phi((q_3(1),q_3(2)))h(q_3(1),x) = \phi((q_4(1),q_4(2)))$, where $q_2(1) = q_1(1)h(q_1(1),x)$ and $q_4(1) = q_3(1)h(q_3(1),x)$. But since $\rho(q_1) = \rho(q_3)$, $q_1(1) = q_3(1)$, and therefore $q_2(1) = q_4(1)$. Thus, if $\rho(q_1) = \rho(q_3)$ then $\rho(q_1x) = \rho(q_3x)$, so that ρ induces an SP partition on M, and hence an SP homomorphism $\overline{\phi}$ is reinterpreted as a digraph mapping, as in the proof of Theorem 2.9, it is identical to ρ; thus M_1 is graph-isomorphic to $\overline{M}_1$.

We will give two examples in which the encoder is independent of the state of the front component.

Example 3.4

Let M be the machine

M	0	1
1	1	5
2	2	6
3	4	3
4	6	2
5	5	1
6	3	4

Consider the following two machines and connecting map Z:

M_1	0	1
q_1	q_1	q_2
q_2	q_2	q_1

M_2	0	1
r_1	r_1	r_1
r_2	r_2	r_3
r_3	r_3	r_2

Z	0	1
q_1	0	1
q_2	1	0

Then the cascade $M_1 \circ_Z M_2$ is

$M_1 \circ_Z M_2$	0	1
$P_1 = q_1r_1$	$q_1r_1 = P_1$	$q_2r_1 = P_5$
$P_2 = q_1r_2$	$q_1r_2 = P_2$	$q_2r_3 = P_6$
$P_3 = q_1r_3$	$q_1r_3 = P_3$	$q_2r_2 = P_4$
$P_4 = q_2r_2$	$q_2r_3 = P_6$	$q_1r_2 = P_2$
$P_5 = q_2r_1$	$q_2r_1 = P_5$	$q_1r_1 = P_1$
$P_6 = q_2r_3$	$q_2r_2 = P_4$	$q_1r_3 = P_3$

Under the maps $\phi: P_i \mapsto i$ and h,

h	0	1
P_1	0	1
P_2	0	1
P_3	1	0
P_4	0	1
P_5	0	1
P_6	1	0

$M_1 \circ_Z M_2$ realizes M with feedback encoding, and the encoding map depends only on the input and the state of M_2.

Notice that M has the SP partition $\{\overline{15};\overline{2346}\}$. One might ask, therefore, whether it is "worth" using a realization with feedback encoding when, in fact, there is a cascade which realizes M without feedback encoding. The answer, of course, depends on the intended application, but in this case M must be realized without feedback encoding by the cascade of a two state machine and a four state machine, which certainly requires more state circuitry than the cascade $M_1 \circ_Z M_2$.

In the next example, which will be quite similar, the realized machine will have no nontrivial cascade realization.

Example 3.5

Let M_2 be as in the previous example, and let M_1 instead be isomorphic to M_2, with states $\{q_i \mid i=1,2,3\}$. Under the connecting map $Z(q_i,x) = x$ if $i \neq 3$ and $1-x$ if $i = 3$, the cascade $M_1 \circ_Z M_2$ is:

M_1 o$_Z$ M_2	0	1
q_1r_1	q_1r_1	q_2r_2
q_1r_2	q_1r_2	q_2r_3
q_1r_3	q_1r_3	q_2r_1
q_2r_1	q_2r_1	q_3r_2
q_2r_2	q_2r_2	q_3r_3
q_2r_3	q_2r_3	q_3r_1
q_3r_1	q_3r_2	q_1r_1
q_3r_2	q_3r_3	q_1r_2
q_3r_3	q_3r_1	q_1r_3

Now, with the maps $\phi(q_ir_j) = 3(i-1)+j$ and $h(q_ir_j,x) = x$ if $j \neq 3$ and $1-x$ if $j = 3$, the following machine M is isomorphically realized with feedback encoding by M_1 o$_Z$ M_2:

M	0	1
1	1	2
2	2	6
3	4	3
4	4	8
5	5	9
6	7	6
7	8	1
8	9	2
9	3	7

However, in this case M has no nontrivial SP partitions, as can be shown in the usual manner by demonstrating that for any pair (i,j)

of states, an SP partition in which $i \equiv j$ has only one block.

It should be clear by this point that whether or not a machine M has an isomorphic cascade realization with feedback encoding depends on the structure of D(M). We will now strengthen this impression by giving a large class of families F of digraphs such that if some cascade has a submachine whose digraph is in F then one component of the cascade also has a submachine whose digraph is in F. We begin by proving a number-theoretic lemma (Lemma 3.4).

Recall the following common notations. If integer n divides integer m we write $n|m$, and if not we write $n\nmid m$. The greatest common divisor of a set $\{i_1,\ldots,i_n\}$ of integers is denoted $(i_1,\ldots,i_n)$.

Lemma 3.3

Let $g = (a,m)$. Then the congruence $ax \equiv b \pmod m$ has no solutions if $g\nmid b$. If $g|b$ then for any $t = 0,1,\ldots,g-1$, $x \equiv (b/g)x_0+t(m/g) \pmod m$ is a solution, where x_0 is any solution to $(a/g)x \equiv 1 \pmod{m/g}$.

PROOF:

See [32, Theorem 2.13]. Note that $(a/g)x \equiv 1 \pmod{m/g}$ always has a solution $x_0 = (a/g)^{\phi(m/g)}$, where the value of $\phi(n)$ is the number of positive integers less than or equal to n which are relatively prime to n.

Lemma 3.4

If $(i_1,\ldots,i_t,n) = g$ then for any r there is a solution to $\sum_{j=1}^{t} w_j i_j \equiv r \pmod n$ if and only if $g|r$.

PROOF:

The case t = 1 is just Lemma 3.3. Suppose the result to hold when t = s-1 and consider the congruence $\sum_{j=1}^{s} w_j i_j \equiv r \pmod{n}$. For a fixed w_s, this equation has a solution if and only if the equation

$$\sum_{j=1}^{s-1} w_j i_j \equiv r - w_s i_s \pmod{n} \tag{3.4}$$

has a solution.

Let $g' = (i_1,\ldots,i_{s-1},n)$. By hypothesis, (3.4) has a solution just when $g' \mid (r-w_s i_s)$. Thus we must find a solution to $w_s i_s \equiv r \pmod{g'}$, which, by Lemma 3.3, has a solution of and only if $(g',i_s) \mid r$. But $(g',i_s) = g$, so there is a solution to $\sum_{1}^{s} w_s i_s \equiv r \pmod{n}$ if and only if $g \mid r$.

Now, recall that in Example 2.3 we presented a class of n-state, t-input machines, each of which had an n-cycle in the automorphism group $\Gamma D(M)$, and showed that the digraphs of these machines could be completely specified by n and the numbers $i_j, j=1,\ldots,t$, such that $\gamma q_0 = \{q_{i_j}\}$. We also showed, for any $m \mid n$, that there is an m-state machine M', with digraph $D(M') = D'_m(M)$, where $\Gamma D(M')$ contains an m-cycle, such that $D'_m(M)$ is an admissible homomorphic image of D(M). We use these facts in the next theorem which shows that, at least for machines whose digraphs are sufficiently symmetric, cascade decompositions are structure preserving.

Theorem 3.8

If M, with digraph $D(M) = D(i_1,\ldots,i_t,n)$ is a submachine of a cascade $M_1 \circ_Z M_2$, and if the g.c.d. $(i_1,\ldots,i_t,n) = 1$, then there is a machine M'_m with digraph $D'_m(M)$ such that either

i. $m|n$, $m > 1$, and M'_m is a submachine of M_1; or

ii. $m = 1$ and M'_m is a submachine of M_2.

PROOF:

Let ϕ be the admissible homomorphism of $M_1 \circ_Z M_2$ onto M_1 guaranteed by Theorem 3.6. We first assume that $|\phi(M)| > 1$, and consider the map ϕ restricted to M.

Let k be the smallest integer for which there are states q_i, q_{i+k} such that $\phi(q_i) = \phi(q_{i+k})$; by symmetry we can take $i = 0$. For each $1 \le r \le t$, there is an s_r, $1 \le s_r \le t$, such that $\phi(q_{i_r}) = \phi(q_{k+i_{s_r}})$. If, for each $1 \le r \le t$, $s_r = r$ then, for all j, $\phi(q_{i_j}) = \phi(q_{k+i_j})$. [Note: if, for some r, i_r and i_{r+1} are the same integer we could have $s_r = r+1$ and $s_{r+1} = r$ without affecting this conclusion]. Otherwise, let r_0 be the smallest integer such that $i_{s_{r_0}} < i_{r_0}$. Then $\phi(q_{i_{r_0}}) = \phi(q_{k+i_{s_{r_0}}})$ and, since $i_{s_{r_0}} - i_{r_0} < 0$, $(k+i_{s_{r_0}}) - i_{r_0} < k$, a contradiction, unless $|\{\phi(q_i)\}| = n$, in which case $\phi|_M$ is an isomorphism.

Thus $\phi(q_0) = \phi(q_k)$, $\phi(q_{i_j}) = \phi(q_{k+i_j})$ for all $1 \le j \le t$ and, in general, if $\alpha = w_1 i_1 + w_2 i_2 + \ldots + w_t i_t$ then $\phi(q_\alpha) = \phi(q_{\alpha+k})$, for any $n_1, \ldots, n_t \ge 0$.

But, by Lemma 3.4, since $(i_1, \ldots, i_t, n) = 1$, there is a solution to $\alpha = \Sigma w_j i_j \equiv r \pmod n$ for any r, so that for all s and t, $\phi(q_s) = \phi(q_{s+kt})$. But then, if $\phi(q_r) = \phi(q_s)$, where $s > r$, we can write $s-r = hk+t$, $t < k$ and conclude that $\phi(q_{r+hk}) = \phi(q_s)$, a contradiction to the minimality of k since $s-(r+hk) = t < k$, unless $t = 0$. Thus $\phi(q_r) = \phi(q_s)$ if and only if $s \equiv r \pmod k$. Then $D(\phi(M)) \cong D_k(M)$.

We have shown that if $|\phi(M)| = m > 1$ then there is a submachine M'_m of M_1 with the desired properties. We now consider the case where $|\phi(M)| = 1$.

Let $\phi(q_0) = q^*$. Then, as a submachine of $M_1 \circ_Z M_2$, M has states $q_i = (q^*,q_i(2))$; then the states $\{q_i(2)\}$ induce a submachine with digraph isomorphic to D(M), where if x is the input which takes q_i to q_j in M, the input which takes $q_i(2)$ to $q_j(2)$ is $Z(q^*,x)$. Therefore, in fact, the states $\{q_i(2)\}$ induce a submachine of M_2 which is isomorphic to M.

Corollary

If the machine M'_m has $D(M'_m) = D'_m(M) = D(j_1,\ldots,j_t,m)$, then $(j_1,\ldots,j_t,m) = 1$.

PROOF:

The parameter j_k for M_m is the residue of i_k module m; that is, $0 \le j_k \le m-1$ and $j_k \equiv i_k \pmod m$. Now suppose $g = (j_1,\ldots,j_t,m)$. Then $g|m$ and for each k, $g|j_k$. But since $j_k \equiv i_k$ (module m), there is an $\ell_k \ge 0$ such that $i_k = j_k + m\ell_k$, so that $g|i_k$. Also, since $g|m$ and $m|n$, $g|n$. Thus $g|(i_1,\ldots,i_t,n)$. Hence $g = 1$.

We now proceed to consider the complexity of cascade realizations, with feedback encoding. These results are motivated by Zeigler's [36] generalization of the Burks-Wang conjecture [3]. Up to now, we have been concerned only with cascades of two machines, but it is clear that we can generalize our results to the more general iterated cascades as defined in Chapter I.

Consider, for example, Theorem 3.8. Suppose that M, as defined in the theorem is a submachine of a cascade ΠN_i. Generalized cascade is a binary procedure, so we can find the major connective and write $\Pi N_i = L_1 \circ_{Z_1} L_2$. The theorem guarantees the existence of a machine

M'_m in either L_1 or L_2. Since the parameters for M'_m satisfy $(j_1,\ldots,j_t,m) = 1$, if that component of the cascade $L_1 \circ_Z L_2$ which contains M'_m is itself a cascade, we can repeat the procedure. Eventually we will find a submachine M'_r with digraph $D'_r(M)$ of some component N_s of the cascade ΠN_i. This proves

Corollary

If M, with digraph $D(M) = D(i_1,\ldots,i_t,n)$ is a submachine of a generalized cascade ΠN_j, and if $(i_1,\ldots,i_t,n) = 1$ then there is a submachine M'_r with digraph $D'_r(M)$ of some component N_k of the cascade.

Suppose now that we have a cascade $N = \Pi N_i$, and let machine N_i have p_i states. We define the *size* of the cascade, denoted size(N), to be the $\max\{p_i\}$. Let S_σ be the collection of all generalized cascades N having size(N) $\leq \sigma$. We will show that S_σ is not *universal* for isomorphic realization with feedback encoding; i.e., that (for any σ) there is a machine M which cannot be isomorphically realized with feedback encoding by any element of S_σ. In fact, we will show a slightly stronger result. First, however, we will show that this statement is not true for unrestricted feedback encoding, by proving the following corollary to Theorem 3.3.

Corollary

Let $M = \langle Q_M, I_M, \delta_M \rangle$ be any machine. Then there is a cascade $N = \langle Q_N, I_N, \delta_N \rangle = \Pi N_i$ of size 2 and maps $\rho: Q_M \to Q_N$, $h: Q_N \times I_M \to I_N$, where ρ is one-to-one, such that for each $q \in Q_M$ and $x \in I_M$,

$$qx = \rho^{-1}(\rho(q)h(\rho(q),x)). \tag{3.5}$$

PROOF:

Note first that if h were a Type II feedback encoder then the pair (ρ^{-1},h) would be defining a realization with feedback encoding by a submachine of M.

Let $n = \{\log_2|Q_M|\}$, and let N_n have 2^n states $\{q_0,\ldots,q_{2^n-1}\}$ and 2^n inputs $\{x_0,\ldots,x_{2^n-1}\}$, where for any q_i and x_j, $q_ix_j = q_j$. By Theorem 3.3, maps ρ and h certainly exist and satisfy the condition of the theorem. We need only demonstrate, therefore, that N_n is a cascade of size 2. We do this inductively.

Consider the machines N_{n-1} and N_1, with input sets $\{y_i|i=0,\ldots,2^{n-1}-1\}$ and $\{x_i|i=0,1\}$. Modify N_{n-1}, to get a machine N'_{n-1} by replacing each y_i by two inputs, y_{i0} and y_{i1}, where for each q_j, $q_jy_{i0} = q_jy_{i1} = q_i$. It is easy to verify that with connecting map Z such that $Z(q_j,y_{i0}) = x_0$ and $Z(q_j,y_{i1}) = x_1$ the cascade N'_{n-1} o_Z N_1 is isomorphic to N_n. Had we expressed $N_n = N_{n-1}$ o_Z N_1 we would be done. As is, we must now show that N'_{n-1} can be further decomposed.

Let r be an s-bit binary number $r = b_1 \ldots b_s$, and let $proj_j(r) = b_j$; for $i < j$ let $proj_{i,j}(r)$ be $b_ib_{i+1}\ldots b_j$. Let N_1^* be the machine with states $\{q_0,q_1\}$ and inputs $\{y_0,\ldots,y_{2^{n-1}-1}\}$ such that $q_ix_j = q_{proj_1(j)}$. Let Z_1 be the map $Z_1(q_i,y_j) = x_{proj_2(j)}$; then the cascade $N_2^* = N_1^*$ o_{Z_1} N_1 has states $\{q_0,q_1,q_2,q_3\}$ and inputs $\{y_i|i=0,\ldots,y_{2^{n-1}-1}\}$ such that $q_iy_j = q_{proj_{1,2}(j)}$. Let Z_2 be the map $Z_2(q_i,y_j) = x_{proj_3(y_j)}$; then $N_3^* = N_2^*$ o_Z N_1 has state $\{q_i|i=0,\ldots,n\}$ and inputs $\{y_i|i=0,\ldots,2^{n-1}-1\}$, such that $q_iy_j = q_{proj_{1,3}(j)}$. Continuing in this manner, we see that N_r^* has 2^n states and 2^{n-1} inputs such that the action of each y_j in N_r^* is isomorphic to the action of $y_{proj_{1,r}(j)}$ in N_r. Then N_{n-1}^* is isomorphic to N'_{n-1}, so that N_n is a cascade of size two.

We now return our attention to restricted feedback encodings.

Theorem 3.9

For any σ and any t there is a t-input machine M which cannot be isomorphically realized with feedback encoding by any submachine of any element of S_σ.

PROOF:

Let p be a prime larger than both σ and t and consider any p-state machine M with digraph $D(i_1,\ldots,i_t,p)$ as described in Theorem 3.8 and Example 2.4. Such a machine can, in fact, always be constructed such that $i_t \neq 0$. If M is isomorphically realized by a submachine N of an element ΠN_i of S_σ then N also has digraph $D(i_1,\ldots,i_t,p)$. Then since p is prime, Theorem 3.8 assures us that at least one component N_i of the cascade contains a submachine with digraph $D'_p(N) \cong D(N)$. But this submachine has $p > \sigma$ states so that N_i has more than σ states, and hence size $(\Pi N_i) > \sigma$.

Corollary

For any σ and any t, there is a t-input machine M which cannot be isomorphically realized by any submachine of any element of S_σ.

In the classical theory of realization a similar result holds for both homomorphic realization and for simulation [36]. We will see in the next chapter that the corresponding result does not hold for simulation with feedback encoding. For homomorphic realizations, we can prove a result slightly more restricted than Theorem 3.9.

If a cascade $M_1 \circ_Z M_2$ contains an n-*cycle* C_n, i.e., a strong, autonomous, n-state machine, then by the corollary to Lemma 3.1, M_1 contains an admissible homomorphic image of C_n. By Theorem 2.11, any admissible homomorphic image of C_n is some C_m, where $m|n$. Thus,

for some $m|n$, M_1 contains C_m. If C_n is defined by input x and has states $\langle q_0,\ldots,q_{n-1}\rangle$ consider the string $y = Z(q_0(1),x)Z(q_1(1),x)\ldots Z(q_{m-1}(1),x)$. It can be shown (see below) that the states, in M_2,

$$q_0(2), q_0(2)y, q_0(2)y^2, \ldots, q_0(2)y^{\frac{n}{m}-1}$$

are all distinct, and that $q_0(2)y^{\frac{n}{m}} = q_0(2)$. The string y thus induces a *string cycle* in M_2; we call $\frac{n}{m}$ the *string period*. If p is a prime which divides n, then either $p|m$ or $p|\frac{n}{m}$, so one of M_1 or M_2 must have at least p states. Thus, $\text{size}(M_1 \circ_Z M_2) \geq p$. We would like to be able to continue this process and show that if any cascade M contained C_n, and $p|n$, then $\text{size}(M) > p$. If $p|m$ and M were a cascade, we could indeed continue, since M_1 contains a cycle C_n. At some point in this "unfolding", however, we may come across the situation where the machine which is guaranteed to have at least p states is the tail component of a cascade, and then we have the problem of decomposing a string cycle of string period t, where $p|t$, into string cycles, one of whose string periods is a multiple of p. This can always be done.

Lemma 3.5 (Zeigler [36])

Let M be a cascade $M_1 \circ_Z M_2$ which contains a string cycle of period n. Then there is some $k|n$ such that M_1 contains a string cycle of string period k and M_2 contains a string cycle of string period n/k.

We are now ready to state a complexity result for homomorphic realization with feedback encoding.

Theorem 3.10

For any σ there is a machine which cannot be homomorphically realized with feedback encoding by any submachine of any element of S_σ.

PROOF:

Let M be C_p, where $p > \sigma$ is a prime. If a submachine N of $\Pi N_i \in S_\sigma$ homomorphically realizes M then, by the corollary to Theorem 3.5, D(N) is an admissible homomorphic pre-image of C_p. Then, by Theorem 2.11, for some n, N is C_{np}. Applying Lemma 3.5 as outlined above, we can conclude that one of the N_i has at least p states, contradicting the hypothesis that $p > \sigma$.

We stated that this result is more restricted than Theorem 3.9. In fact, since for autonomous machines the notions of homomorphism and admissible homomorphism coincide, it really says nothing new (Corollary 3.6). The distinction between Theorems 3.9 and 3.10 is that we have no detailed knowledge about admissible homomorphic pre-images of complex structures, such as the digraphs $D(i_1,\ldots,i_t,n)$ of Theorem 3.9. Undoubtedly there is some relationship between these digraphs, their admissible homomorphic pre-images, and the admissible homomorphic images of these, but what this might be is unclear at present.

We stated above that simulation with feedback encoding would be discussed in Chapter III. It is proper at this point, however, to consider a weakened form of this. A b-slow simulation was defined in Chapter I. We say that M *isomorphically b-slow simulates* M' with *feedback encoding* if there are a constant b and maps $\phi: Q \to Q'$ and $h: Q \times I' \to I^+$, where ϕ is both one-to-one and onto, and for all $q,x',\ell(h(q,x')) = b$, such that

$$\phi(q)x' = \phi(qh(q,x')) \tag{3.6}$$

Of course, if b = 1 this reduces to an isomorphic realization with feedback encoding. What we are doing in an isomorphic b-slow simulation

with feedback encoding is assigning to each arc from state q_1' to state q_2' in D(M') a walk of length b from $\phi^{-1}(q_1')$ to $\phi^{-1}(q_2')$ in D(M). The next definition may make this clear.

For any digraph D with point set V = V(D) the *absolute n-th power* $D^{<n>}$ *of* D has $V(D^{<n>}) = V(D)$ and $uv \in D^{<n>}$ if and only if there is a walk of length n joining u to v in D. Figure 3.8 shows a digraph and its absolute square. Thus M isomorphically b-slow simulates M' with feedback encoding if and only D(M') is a subdigraph of $D(M)^{<b>}$. As an aid to determining whether M' can be isomorphically b-slow simulated by some machine, we can give necessary and sufficient conditions for the existence of a digraph D such that $D^{<b>} \cong D(M')$, although these conditions are unwieldy, at best, even for the case b = 2; if $D^{<b>} = D'$ we say that D is an *absolute* b-*th root* of D'.

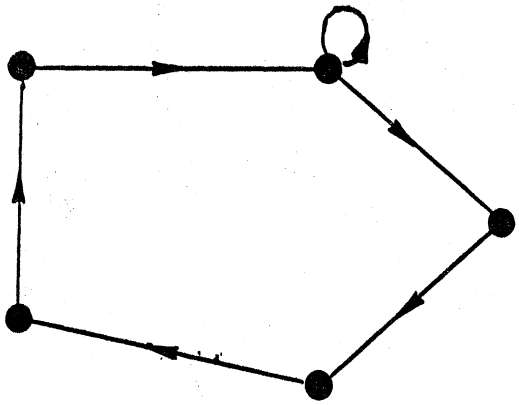
Rather than directly prove the characterization theorem, we prove a similar result, whose proof is virtually identical. If D is a digraph with point set V = V(D), the n-*th power* D^n has $V(D^n)$ and $uv \in D^n$ if and only if there is a walk of length at most n from u to v in D; if $E^n = D$ then E is an n-*th root* of D.

Let $S_1, S_2, \ldots, S_{2n-1}$ be sets, not necessarily disjoint, subject to the constraints that $S_n \neq \emptyset$ and if $S_{n-j} = \emptyset (S_{n+j} = \emptyset)$ then for all $k > j$ $S_{n-k} = \emptyset (S_{n+k} = \emptyset)$. Let $K = K_n(S_1, \ldots, S_{2n-1})$ be the digraph with

$$V(K) = \bigcup_{j=1}^{2n-1} S_j \text{ and } X(K) = [\bigcup_{j=1}^{n-1} S_j x (S_n \cup \ldots \cup S_{n+j})] \cup [\bigcup_{j=n}^{2n-2} S_j x (S_{j+1} \cup \ldots \cup S_{2n-1})].$$

Theorem 3.11

Let D be a digraph and let $n \geq 2$. There exists a digraph E such that $E^n = D$ if and only if there is a collection of subdigraphs $K_i = K_n(S_{i,1}, \ldots, S_{i,2n-1})$ of D associated with the



D

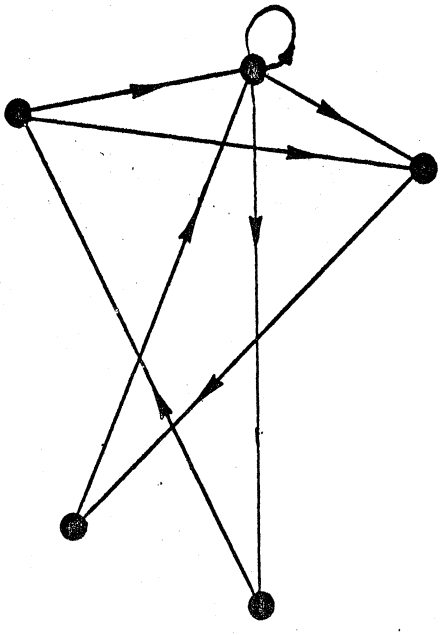


Figure 3.8. A digraph and its absolute square.

points u_i of D such that

(1) $S_{i,n} = \{u_i\}$

(2) $X(D) = X(K_i)$

(3) $u_i \in S_{j,n-1}$ if and only if $u_j \in S_{i,n+1}$

(4) for any $0 < r < n-1$ and $s = r+1$: $u_k \in S_{i,n-s}$ if and only if there is a $u_j \in S_{i,n-r}$ such that $u_k \in S_{j,n-1}$; $u_k \in S_{i,n+s}$ if and only if there is a $u_j \in S_{i,n+r}$ such that $u_k \in S_{j,n+1}$.

PROOF:

Let E be a digraph. For each $u_i \in E$ define $S_{i,j}$ to be $\gamma_E^{j-n}(u_i)$; in particular $S_{i,n} = \{u_i\}$. In E^n, for each $1 \leq j \leq n-1$, each point of $S_{i,j}$ is adjacent to each point of $S_{i,n}, S_{i,n+1}, \ldots, S_{i,n+j}$, and if $n \leq j \leq 2n-2$, each point of $S_{i,j}$ is adjacent to each point of $S_{i,j+1}, \ldots, S_{i,2n-1}$. Each arc u_iu_j of E^n is determined by a path $u_iu_k \ldots u_j$ of length $r \leq n$, so that $u_i \in S_{k,n-1}$ and $u_j \in S_{k,n+r}$, and $u_iu_j \in K_k$. Conditions (3) and (4) follow from the properties of Γ_E.

Conversely, let D have such a collection. Define a digraph E by setting $V(E) = V(D)$ and $u_iu_j \in E$ just when $u_i \in S_{j,n-1}$. We show $E^n = D$ by demonstrating that, for each u_i and each $1 \leq k \leq 2n-1$, $S_{i,k} = \gamma_E^{k-n}(u_i)$. It will then follow that $u_iu_j \in E^n$ if and only if $u_iu_j \in D$. For, if $u_iu_j \in D$, u_iu_j is in some K_k; thus for some r and s, $u_i \in S_{k,r}$ and $u_j \in S_{k,s}$, where $r < n < s$ and $s \leq r+n$. Then there is a $u_{t_1} \in \gamma_E^{(r+1)-n}(u_k)$ such that $u_i \in S_{t_1,n-1}$, so that $u_iu_{t_1} \in E$, and $u_{t_1} \in S_{k,r+1}$. We then find t_2 such that $u_{t_1}u_{t_2} \in E$ and $u_{t_2} \in S_{k,r+2}$; we continue until we find t_b such that $u_{t_{b-1}}u_{t_b} \in E$ and $u_{t_b} \in S_{k,n-1}$, so that $u_{t_b}u_k \in E$. Similarly we find a sequence $t_d, t_{d-1}, \ldots, t_{b+3}, t_{b+2}$

where $d = s-r-1$, such that $u_k u_{t_{b+2}}, u_{t_{b+2}} u_{t_{b+3}}, \ldots, u_{t_d} u_j$ are all arcs of D. This defines a walk of length $s-r \leq n$ between u_i and u_j in E; thus $u_i u_j \in E^n$. If $u_i u_j \in E^n$ then u_i and u_j are joined by a walk $u_i u_{t_1} \ldots u_{t_k} u_j$ of length at most n. Now, $u_i \in \gamma_E^{-1}(u_{t_1}) = S_{t_1,n-1}$ and $u_j \in \gamma^k(u_{t_1}) = S_{t_1,n+k}$. Since $k < n$, $n+k \leq n+(n-1)$; thus $u_i u_j \in D$.

We proceed to show that $S_{i,k} = \gamma_E^{k-n}(u_i)$. If $k = n-1$ then $\gamma_E^{k-n}(u_i) = \gamma_E^{-1}(u_i) = \{u_j | u_j u_i \in E\} = \{u_j | u_j \in S_{i,n-1}\} = S_{i,k}$. If $k = n+1$, $\gamma^{k-n}(u_i) = \{\gamma_E(u_i)\} = \{u_j | u_i u_j \in E\} = \{u_j | u_i \in S_{j,n-1}\}$. But $u_i \in S_{j,n-1}$ just when $u_j \in S_{i,n+1}$; thus $\gamma_E(u_i) = S_{i,n+1}$.

Suppose the result holds for $n-r \leq k \leq n+r$, and let $s = r+1$. Then $\gamma_E^{(n-s)-n}(u_i) = \gamma_E^{-s}(u_i) = \gamma_E^{-1}(\Gamma_E^{-r}(u_i)) = \gamma_E^{-1}(S_{i,n-r}) = \cup\{\gamma_E^{-1}(u_j) | u_j \in S_{i,n-r}\}$. For any $u_j, \gamma_E^{-1}(u_j) = S_{j,n-1}$. Thus if $u_p \in \gamma_E^{-1}(u_j)$ and $u_j \in S_{i,n-r}$ then $u_p \in S_{i,n-r-1} = S_{i,n-s}$. On the other hand, if $u_p \in S_{i,n-r-1}$ then there is a u_j such that $u_p \in S_{j,n-1}$ and $u_j \in S_{i,n-r}$, so that $u_p \in \gamma_E^{-1}(u_j) \subset \gamma_E^{-1}(S_{i,n-r}) = \gamma_E^{-1}(\gamma_E^{-r}(u_i)) = \gamma_E^{-s}(u_i)$ Similarly we can show that $S_{i,n+s} = \gamma^s(u_i)$, establishing the theorem.

The results in [8] and [31] follow as corollaries for the case $n = 2$.

Now, let $H = H_n(S_1, \ldots, S_{2n-1})$ be the digraph with $V(H) = \bigcup_{j=1}^{2n-1} S_j$ and $X(H) = \bigcup_{j=1}^{n-1} [S_j \times S_{j+n}]$. The proof of the next result is virtually identical to that of the preceding and will be omitted.

Theorem 3.12

Let D be a digraph and $n \geq 2$. There is a digraph E such that $E^{<n>} = D$ if and only if there is a collection $H_i = H_n(S_{i,1}, \ldots, S_{i,2n-1})$ associated with the points u_i of D which satisfies conditions (1) - (4) of Theorem 3.11.

Corollary

A digraph D has an absolute square root if and only if there are sets S_i and T_i associated with the points u_i of D such that (1) each point of S_i is adjacent to each point of T_i; (2) for each arc x of D there is some u_i for which x joins S_i and T_i; and (3) $u_i \in S_j$ if and only if $u_j \in T_i$.

CHAPTER IV

ALGEBRAIC STRUCTURES AND SIMULATION WITH FEEDBACK ENCODING

An important structure associated with a finite automaton M is its semigroup, S(M). The semigroup, in fact, is quite crucial to the decomposition theory of Krohn and Rhodes [24]. While it would certainly be desirable to be able to associate an algebraic structure with a machine M which gives us the same information about realizations with feedback encoding that S(M) gives for classical realization, S(M) is clearly not the structure we would wish to use. The semigroup S(M) reflects quite accurately the state-behavior of M, while, as we have shown, the theory of realization with feedback encoding depends much less on behavioral properties than on (digraph) structural ones. Hedetniemi and Fleck [20] defined the S^*-semigroup of an automaton and, with Oehmke [33], showed that it had at least some of the properties we would want any algebraic structure to exhibit for us to consider it to be the "natural" system to study in conjunction with realization with feedback encoding. Unfortunately, as we will show, the properties they found do not extend in the way they should. In fact, rather than being able to apply the S^*-semigroup to the present study, the converse holds: we are able to apply the notion of simulation with feedback encoding to a conjecture of Hedetniemi and Fleck on S^*-semigroups. With the work of Hedetniemi, Fleck, and Oehmke as motivation, we will define other semigroups and study some of their properties vis-à-vis realization with feedback encoding.

We first define a simulation with feedback encoding. Let $M = \langle Q, I, \delta \rangle$ and $M' = \langle Q', I', \delta' \rangle$ be machines. Then M *simulates* M' *with feedback*

encoding if there are maps $\phi: Q \overset{\text{onto}}{\to} Q'$ and $h: Q \times I' \to I^+$ such that for each $q \in Q$ and $x' \in I'$

$$\phi(q)x' = \phi(qh(q,x')). \tag{4.1}$$

If $|Q| = |Q'|$ then we have an *isomorphic simulation with feedback encoding*, and, if not, a *homomorphic simulation with feedback encoding*.

Example 4.1

Let M and M' be the machines in Figure 4.1.

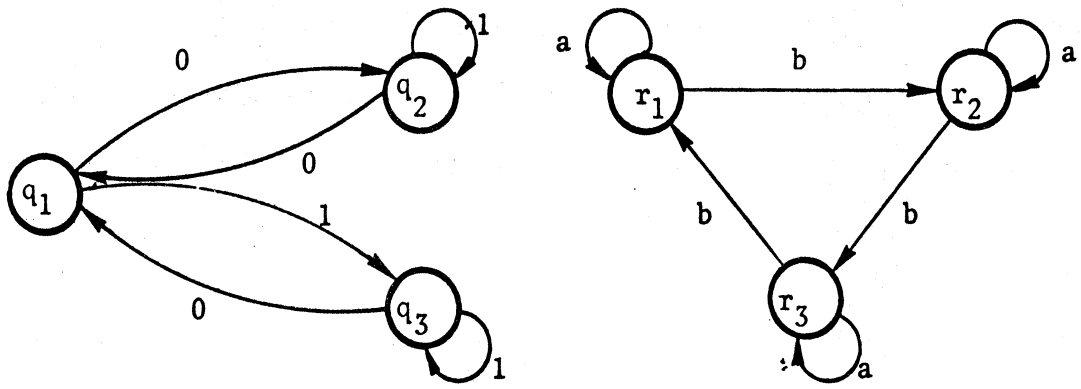


Figure 4.1. Simulation with feedback encoding.

Then by Theorem 3.5 it is clear that M does not realize M' with feedback encoding. But M does simulate M' with feedback encoding. One such simulation is defined by the pair (ϕ,h), where $\phi(q_i) = r_i$, $i = i,2,3$, and h has the following table:

Q \ x'	a	b
q_1	00	0
q_2	1	01
q_3	1	0

Theorem 4.1

If M_1 simulates M_2 with feedback encoding, and M_2 simulates M_3 with feedback encoding, then M_1 simulates M_3 with feedback encoding.

PROOF:

Let the simulation of M_2 by M_1 be defined by maps ϕ and h, and that of M_3 by M_2 be defined by maps ψ and g.

Let $q_1 \in Q_1$ and $x_3 \in I_3$. Suppose that $g(\phi(q_1),x_3) = x_{21} \ldots x_{2n}$; then $\psi(\phi(q_1))x_3 = \psi(\phi(q_1)x_{21} \ldots x_{2n})$. Let $q_{11} = q_1 h(q_1,x_{21})$, $q_{12} = q_{11} h(q_{11},x_{22}),\ldots,q_{1n} = q_{1,n-1} h(q_{1,n-1},x_{2n})$. Then $\phi(q_{11}) = \phi(q_1 h(q_1,x_{21})) = \phi(q_1)x_{21}$, $\phi(q_{12}) = \phi(q_{11} h(q_{11},x_{22})) = \phi(q_1,h(q_1,x_{21})h(q_{11},x_{22}))$, etc. Thus, if we define $f(q_1,x_3)$ to be the string $h(q_1,x_{21})h(q_{11},x_{22}) \ldots h(q_{1,n-1},x_{2n})$, where $q_{11},q_{12},\ldots,q_{1n}$ and $x_{21} \ldots x_{2n}$ are as above, then $\psi(\phi(q_1))x_3 = \psi(\phi(q_1 f(q_1,x_3)))$, so that M_1 does indeed simulate M_3 with feedback encoding.

We now proceed to define and study the algebraic structures mentioned above.

Let S and S' be semigroups with zero, where, without confusion, we will use the same symbol, 0, for both the zero of S and the zero of S'. A map $\Phi:S \to S'$ is a *zero-free homomorphism* if it satisfies

1. $\mathrm{Ker}(\Phi) = \{0\}$;
2. If $ab \neq 0$ then $\Phi(a)\Phi(b) = \Phi(ab)$;
3i. If $\Phi(a)b' \neq 0$ then there is some $b \in \Phi^{-1}(b')$ such that $ab \neq 0$;
3ii. If $a'\Phi(b) \neq 0$ then there is some $a \in \Phi^{-1}(a')$ such that $ab \neq 0$.

(4.2)

We will see examples of zero-free homomorphisms which are not homomorphisms later. For the present, we give some basic properties of zero-free homomorphisms.

Theorem 4.2

a) Let S_1, S_2 and S_3 be semigroups, and let $\Phi: S_1 \to S_2$ and $\Psi: S_2 \to S_3$ be zero-free homomorphisms. Then

i. If $\Phi(a)\Phi(b) = 0$ then $ab = 0$.

ii. $\Psi\Phi: S_1 \to S_3$ is a zero-free homomorphism.

iii. If Φ is one-to-one then Φ is an isomorphism.

b) If $\Phi: S_1 \to S_2$ is a semigroup homomorphism with $\mathrm{Ker}(\Phi) = \{0\}$ then Φ is a zero-free homomorphism.

PROOF:

a) For (i), suppose that $ab \neq 0$. Then $\Phi(a)\Phi(b) = \Phi(ab) \neq 0$ since $\mathrm{Ker}(\Phi) = \{0\}$. The result follows by contraposition.

For (ii) we must verify the four properties of (4.2). First, suppose that $\Psi\Phi(s) = 0$. Then $\Psi(\Phi(s)) = 0$ so $\Phi(s) \in \mathrm{Ker}(\Psi)$ and hence $\Phi(s) = 0$. But then $s \in \mathrm{Ker}(\Phi)$, so $s = 0$. Next, suppose that, in S_1, $st \neq 0$. Then $\Phi(s)\Phi(t) = \Phi(st)$. By (i), $\Phi(st) \neq 0$, so $\Psi(\Phi(s))\Psi(\Phi(t)) = \Psi(\Phi(s)\Phi(t)) = \Psi(\Phi(st))$.

Finally, suppose $\Psi\Phi(a)b \neq 0$, where $b \in S_3$. Then $\Psi(\Phi(a))b \neq 0$ so there is a $d \in S_2$ such that $\Psi(d) = b$ and $\Phi(a)d \neq 0$. Then, there is an $e \in \Phi^{-1}(d)$ such that $ae \neq 0$; $\Psi\Phi(e) = \Psi(d) = b$. This verifies 3i, and the proof for 3ii is virtually identical.

For (iii) we need only check that if $ab = 0$ then $\Phi(ab) = \Phi(a)\Phi(b)$. In fact, $\Phi(ab) = \Phi(a)\Phi(b) = 0$; for, since Φ is one-to-one it follows from 3i and 3ii that if $\Phi(a)\Phi(b) \neq 0$ then $ab \neq 0$, a contradiction. Thus $\Phi(a)\Phi(b) = 0$, and $\Phi(ab)$ is certainly zero since $ab = 0 \in \mathrm{Ker}(\Phi)$.

b) Suppose that $\Phi: S_1 \to S_2$ is a homomorphism, and that $\mathrm{Ker}(\Phi) = \{0\}$. Since $\Phi(ab) = \Phi(a)\Phi(b)$ for all $a,b \in S_1$, condition 2 of (4.2) certainly holds. If $\Phi(a)b' \neq 0$ then for any b such that $\Phi(b) = b'$, $\Phi(a)\Phi(b) = \Phi(ab)$. Thus $ab \neq 0$ since $\Phi(0) = 0 \neq \Phi(a)\Phi(b)$.

Let $M = \langle Q, I, \delta \rangle$ be a machine. An ordered three-tuple (s,x,t) is a *triple for* M, or simply a *triple*, if $s,t \in Q$, $x \in I^+$, and $sx = t$. A triple is *elementary* if the length $\ell(x) = 1$; i.e., if $x \in I$.

Let $S(M)$ be the set of all triples of M together with a distinguished zero element 0. We introduce an operation on $S(M)$ by the following rules:

1. $(s,x,t_1)(t_2,y,r) = (s,xy,r)$ if $t_1 = t_2$

   0 if $t_1 \neq t_2$

2. $b0 = 0b = 0$ for each $b \in S(M)$.

Lemma 4.1

For any machine M, $S(M)$ is a semigroup.

PROOF:

Clearly the operation is defined for each pair of elements in $S(M)$. Thus, we need only show associativity. Let $b_1, b_2, b_3 \in S(M)$. Since 0 commutes with every element of $S(M)$, if any of the b_i are 0 then certainly $(b_1b_2)b_3 = b_1(b_2b_3) = 0$. Suppose that $b_1 = (q,x,r_1)$, $b_2 = (r_2,y,s_2)$, $b_3 = (s_3,z,t)$. If $r_1 = r_2$ and $s_2 = s_3$ then $b_1b_2 = (q,xy,s_2)$, $b_2b_3 = (r_2,yz,t)$, and $(b_1b_2)b_3 = (q_1,(xy)z,t) = (q,x(yz),t) = b_1(b_2b_3)$. If $r_1 \neq r_2$ and $s_1 \neq s_2$ then $b_1b_2 = b_2b_3 = 0$, so $(b_1b_2)b_3 = b_1(b_2b_3) = 0$. If $r_1 = r_2$ and $s_1 \neq s_2$ then $b_2b_3 = 0$ and $b_1(b_2b_3) = 0$. Now, $b_1b_2 = (q,xy,s_1)$ and so $(b_1b_2)b_3 = 0$. Similarly, if $r_1 \neq r_2$ and $s_1 = s_2$ then $(b_1b_2)b_3 = b_1(b_2b_3) = 0$. Thus $S(M)$ is a semigroup.

Suppose that M realizes M' with feedback encoding, where the realization is defined by maps ϕ and h. Although, for each $s \in Q$,

h_s is a map from I' to I, h_s can be extended to a map $\bar{h}_s$ from I'^+ to I^+ in the following way. If $x' \in I'$, $\bar{h}_s(x') = h_s(x')$; if $x',y' \in I'^+$, $\bar{h}_s(x'y') = \bar{h}_s(x')\bar{h}_{sh_s(x')}(y')$. We must, of course, verify that if $w'z'$ is a different way of writing $x'y'$ then $\bar{h}_s(w'z') = \bar{h}_s(x'y')$.

Lemma 4.2

a) For each $s \in Q$, $\bar{h}_s$ is a function.

b) For each $s \in Q$, $x' \in I'^+$, $\phi(s)x' = \phi(s\bar{h}_s(x'))$.

c) For each $s \in Q$, $x \in I^+$ there is a unique $x' \in I'^+$ such that $h_s(x') = x$.

PROOF:

a) We proceed by induction. The result is true for strings of length 1 since h_s is a function, and true for strings of length 2 since, for such a string, there is only one decomposition into smaller strings. Suppose it to be true for strings of length n-1 and let $x' = w'_1 \ldots w'_n$, where each $w'_i \in I'$. Choose $1 \leq i < j \leq n-1$. We must show that $\bar{h}_s(w'_1 \ldots w'_i)\bar{h}(s\bar{h}_s(w'_1 \ldots w'_i), w'_{i+1} \ldots w'_n) = \bar{h}_s(w'_1 \ldots w'_j)\bar{h}(s\bar{h}_s(w'_1 \ldots w'_j), w'_{j+1} \ldots w'_n)$. Let $s_1 = \bar{h}_s(w'_1 \ldots w'_i)$. By induction we can write $\bar{h}_{s_1}(w'_{i+1} \ldots w'_n) = \bar{h}_{s_1}(w'_{i+1} \ldots w'_j)\bar{h}_{s_2}(w'_{j+1} \ldots w'_n)$, where $s_2 = s_1\bar{h}_{s_1}(w'_{i+1} \ldots w'_j)$. Also by induction, $\bar{h}_s(w'_1 \ldots w'_i)\bar{h}_{s_1}(w'_{i+1} \ldots w'_j) = \bar{h}_s(w'_1 \ldots w'_j)$. Thus, $\bar{h}_s(w'_1 \ldots w'_i)\bar{h}_{s_1}(w'_{i+1} \ldots w'_n) = \bar{h}_s(w'_1 \ldots w'_j)\bar{h}_{s_2}(w'_{j+1} \ldots w'_n)$, and the two expressions are equal.

b) Let $s \in Q$ and $x' = w'_1 \ldots w'_n$, where each $w'_i \in I'$. If $\ell(x') = 1$ then the result holds since $\bar{h}_s(x') = h_s(x')$. Assume the result holds when $n \leq m-1$ and let $n = m$. Choose $1 \leq i \leq n-1$. Then $\phi(s)x' = [\phi(s)w'_1 \ldots w'_i]w'_{i+1} \ldots w'_n = \phi(s\bar{h}_s(w'_1 \ldots w'_i))w'_{i+1} \ldots w'_n$. If $s_1 = s\bar{h}_s(w'_1 \ldots w'_i)$ then $\phi(s)x' = \phi(s_1)w'_{i+1} \ldots w'_n$ $= \phi(s_1\bar{h}_{s_1}(w'_{i+1} \ldots w'_n))$, which, by part a), is equal to $\phi(s\bar{h}_s(x'))$.

c) This follows immediately from part a) and the fact that it is true for h_s by the properties of a Type I feedback encoding.

From this point, we will use the symbol h_s for both h_s and $\bar{h}_s$. For a state $s \in Q$ and a string $x \in I^+$ we will write $h_s^{-1}(x)$ for the unique string $x' \in I'^+$ such that $h_s(x') = x$.

Theorem 4.3

If M realizes M' with feedback encoding then there is a zero-free homomorphism from $S(M)$ onto $S(M')$.

PROOF:

Let the realization be defined by maps ϕ and h. If $b = (s,x,t)$ is a triple of M, define $\Phi(b) = (\phi(s),h_s^{-1}(x),\phi(t))$. By Lemma 4.2, $\phi(s)h_s^{-1}(x) = \phi(sx) = \phi(t)$, so that $\Phi(b)$ is a triple for M'. Also, define $\Phi(0) = 0$. We will prove that Φ is a zero-free homomorphism from $S(M)$ onto $S(M')$. First, we show that Φ is onto. Let $b' = (s',x',t')$ be a triple of M'. Choose $s \in \phi^{-1}(s')$, and let $x = h_s^{-1}(x')$. If $t = sx$ then $\phi(t) = \phi(sx) = \phi(s)x' = t'$, so $\Phi((s,x,t)) = b'$.

Clearly $\mathrm{Ker}(\Phi) = \{0\}$. Suppose that $b_1 = (s,x,t)$ and $b_2 = (t,y,r)$, so that $b_1b_2 = (s,xy,r) \neq 0$. Now $\Phi(b_1)\Phi(b_2) = (\phi(s),h_s^{-1}(x),\phi(t))\cdot(\phi(t),h_t^{-1}(y),\phi(r)) = (\phi(s),h_s^{-1}(x)h_t^{-1}(x)h_t^{-1}(y),\phi(r))$. By Lemma 4.2, $h_s^{-1}(x)h_t^{-1}(y) = h_s^{-1}(xy)$, so $\Phi(b_1)\Phi(b_2) = (\phi(s),h_s^{-1}(xy),\phi(t)) = \Phi(b_1b_2)$.

Suppose that $\Phi(b)d' \neq 0$, where $b = (s,x,t)$. Then $\Phi(b) = (\phi(s),x',\phi(t))$, so that for some $y' \in I'^+$, $r' \in Q'$, $d' = (\phi(t),y',r')$. If $d = (t,h_t(y'),th_t(y'))$, then $\Phi(d) = d'$ and $bd \neq 0$. Similarly, if $d'\Phi(b) \neq 0$ then there is a d such that $\Phi(d) = d'$ and $db \neq 0$.

Corollary

If M isomorphically realizes M' with feedback encoding then $S(M) \cong S(M')$.

PROOF:

This follows from Theorem 4.2.2.

We can now give the examples mentioned above of zero-free homomorphisms which are not homomorphisms. Consider the map Φ guaranteed by Theorem 4.3. If s_1 and s_2 are two states of M for which $\phi(s_1) = \phi(s_2)$ then any triples of the form $b_1 = (r,x,s_1)$ and $b_2 = (s_2,y,t)$ have product $b_1b_2 = 0$, while $\Phi(b_1)\Phi(b_2) = (\phi(r),xy,\phi(t)) \neq 0$.

We will also prove a converse to this theorem. If $b = (s,x,t)$ is a triple of machine M define $i(b) = s$ and $f(b) = t$. A state s of M is *reachable* if there is a triple b such that $f(b) = s$.

Let S be a semigroup with zero. For any $s \in S$ define $s^{\perp} = \{t \mid st \neq 0\}$ and $s^{\top} = \{t \mid ts \neq 0\}$. We can then define equivalence relations $\equiv_{\top}$ and $\equiv_{\perp}$ on S: $s \equiv_{\perp} t$ if $s^{\perp} = t^{\perp}$, and $s \equiv_{\top} t$ if $s^{\top} = t^{\top}$.

Lemma 4.3

For any machine M in which every state is reachable the relation $\equiv_{\perp}$ on S(M) has finite index which is equal to $1 + |Q|$. In fact, $b_1 \equiv_{\perp} b_2$ if and only if $f(b_1) = f(b_2)$, and $[0]_{\perp} = \{0\}$.

PROOF:

Clearly $0^{\perp} = \emptyset$, while for any $b = (s,x,t)$ there is some triple (t,y,r), so that $b^{\perp} \neq \emptyset$. Thus $[0]_{\perp} = \{0\}$.

Note that, by definition, $bd \neq 0$ if and only if $f(b) = i(d)$. Thus if $b_1^{\perp} = b_2^{\perp}$ then $f(b_1) = f(b_2)$, and conversely. Therefore under $\equiv_{\perp}$ there is one equivalence class for each reachable state of Q_1 and one for 0 and so, if every state is reachable the index of $\equiv_{\perp}$ is $|Q|+1$.

Lemma 4.4

For any machine M in which every state is reachable the relation $\equiv_\tau$ on $S(M)$ has finite index, equal to $1+|Q|$; $[0]_\tau = \{0\}$, and $b_1 \equiv_\tau b_2$ if and only if $i(b_1) = i(b_2)$.

PROOF:

The proof is similar to that for Lemma 4.3. In this case we need the reachability criterion to guarantee that $[0]_\tau = \{0\}$.

Theorem 4.4

Let M and M' be machines in which each state is reachable. If $|I| = |I'|$ and there is a zero-free homomorphism $\Phi: S(M) \overset{\text{onto}}{\to} S(M')$, then M realizes M' with feedback encoding.

PROOF:

We first show that $b_1 \equiv_\perp b_2$ implies that $\Phi(b_1) \equiv_\perp \Phi(b_2)$. Suppose that $b_1 \equiv_\perp b_2$ and that $\Phi(b_1)d' \neq 0$, so that $d' \in \Phi(b_1)^\perp$. Then there is a d such that $\Phi(d) = d'$ and $b_1 d \neq 0$ by (4.2). Since $b_1 \equiv_\perp b_2$, $b_2 d \neq 0$. Then $\Phi(b_2)\Phi(d) = \Phi(b_2)d' = \Phi(b_2 d) \neq 0$ since $\mathrm{Ker}(\Phi) = \{0\}$. Thus $d' \in \Phi(b_2)^\perp$. Similarly, $\Phi(b_2)^\perp \subseteq \Phi(b_1)^\perp$, so $\Phi(b_1)^\perp = \Phi(b_2)^\perp$ and hence $\Phi(b_1) \equiv_\perp \Phi(b_2)$. By a similar argument, $b_1 \equiv_\tau b_2$ implies $\Phi(b_1) \equiv_\tau \Phi(b_2)$.

It then follows from Lemmas 4.3 and 4.4 that $f(b_1) = f(b_2)$ implies that $f(\Phi(b_1)) = f(\Phi(b_2))$ and $i(b_1) = i(b_2)$ implies that $i(\Phi(b_1)) = i(\Phi(b_2))$. Therefore, Φ induces maps $\phi_i: Q \to Q'$ and $\phi_f: Q \to Q'$ such that $\Phi((s,x,t)) = (\phi_i(s),x',\phi_f(t))$. Now, suppose that $b_1 = (r,x,s)$ and $b_2 = (s,y,t)$, so that $b_1 b_2 \neq 0$. Then $\Phi(b_1) = (\phi_i(r),x',\phi_f(s))$, $\Phi(b_2) = (\phi_i(s),y',\phi_f(t))$. But we know that $\Phi(b_1)\Phi(b_2) \neq 0$; thus, for each reachable state s, $\phi_i(s) = \phi_f(s)$. Since every state is reachable, $\phi_i = \phi_f = \phi$, a map from Q to Q'. Furthermore, ϕ is onto, since Φ is.

Let $x \in I^+$, $x' \in I'$, be such that for some triple $b = (s,x,t)$, $\Phi(b) = (\phi(s),x',\phi(t))$. If we write $x = yz$, where neither y nor z is empty, then $b = (s,y,sy)(sy,z,t)$. Thus $\Phi((s,y,sy))\Phi((sy,z,t)) = \Phi(b) = (\phi(s),x',\phi(t))$. But this is an impossible situation: the second components of $\Phi((s,y,sy))$ and $\Phi((sy,z,t))$ are nonempty strings, say y' and z', so that $x' = y'z'$, a contradiction of the hypothesis that $x' \in I'$. Thus, every pre-image of an elementary triple of M' is an elementary triple of M.

We now show that if $b' = (s',x',t')$ and $\phi(s) = s'$, then there is a triple b with $i(b) = s$ such that $\Phi(b) = b'$. Since s is reachable, there is some triple $d = (r,y,s)$. Now, $\Phi(d) = (\phi(r),y',\phi(s))$, so $\Phi(d)b' \neq 0$. Thus, since Φ is a zero-free homomorphism, there is a triple b such that $db \neq 0$ and $\Phi(b) = b'$. But $db \neq 0$ implies $i(b) = f(d) = s$.

Now let $|I| = |I'| = n$. Choose any $s' \in Q'$ and any s such that $\phi(s) = s'$. If the n elementary triples b'_j with $i(b'_j) = s'$ are $b'_j = (s',x'_j,t'_j)$ then for each b'_j there is a triple b_j with $\Phi(b_j) = b'_j$ and $i(b_j) = s$. Furthermore, as we have shown, each such b_j is an elementary triple. Also, since $|I| = |I'| = n$, there are exactly n elementary triples b with $i(b) = s$. Therefore, for each $s \in Q$ and each $x' \in I'$ there is a unique $x \in I$ such that $\Phi((s,x,sx)) = (\phi(s),x',\phi(sx))$; we can then define a map $h:Q \times I' \to I$ by setting $h(s,x') = x$, where $\Phi((s,x,sx)) = (\phi(s),x',\phi(sx'))$. The pair (ϕ,h) defines a realization with feedback encoding.

Corollary

If M and M' are machines in which each state is reachable, with $|I| = |I'|$, and if $S(M') \cong S(M)$, then M isomorphically realizes M' with feedback encoding.

We can get a result parallel to Theorem 4.3 for the case in which one machine simulates another. Let S and S' be semigroups with zero; we say that S' *zero-free divides* S if there is a subsemigroup S_1 of S which contains the zero of S such that there is a zero-free homomorphism from S_1 onto S'.

Theorem 4.5

Let M and M' be machines such that M simulates M' with feedback encoding, where the simulation is defined by maps ϕ and h. If the extended map $h:Q \times I'^{+} \to I^{+}$ is one-to-one for each $s \in Q$, then S(M') zero-free divides S(M).

Note: A sufficient condition for the extended map to be one-to-one will be given as Theorem 4.15.

PROOF:

For each triple $b' = (s',x',t')$ define $\Psi(b') = \{(s,h_s(x'),sh_s(x')) \mid \phi(s) = s'\}$, and define $\Psi(0) = 0$. Let $S_1 = \Psi(S(M'))$. We first show that S_1 is a subsemigroup of S(M). What we must verify is that if $b_1b_2 = b_3$ in S(M) and if $b_1,b_2 \in S_1$, then $b_1b_2 = b_3 \in S_1$. This is immediate if any of b_1,b_2,b_3 are 0. Suppose that $b_1 = (s,x,t) \in \Psi((s',x',t'))$, and $b_2 = (t,y,r) \in \Psi((t',y',r'))$. Then $(s',x',t')(t',y',r') = (s',x'y',r')$. Since $x = h_s(x')$ and $y = h_t(y')$, $xy = h_s(x'y')$ and $s(xy) = r$. Thus $b_1b_2 = (s,xy,r) \in \Psi((s',x'y',r')) \subseteq S_1$. Now, for each $b \in S_1$, define $\Phi(b) = \Psi^{-1}(b)$; Φ is properly a function since if $b = (s,x,t)$ is in $\Psi((s_1',x_1',t_1'))$ and also in $\Psi((s_2',x_2',t_2'))$ then $s_1' = s_2' = \phi(s)$ by definition of Ψ and $x_1' = x_2'$, since h_s is one-to-one, so that $t_1' = s_1'x_1' = t_2'$. The proof that Φ is a zero-free homomorphism is now identical to the proof of Theorem 4.3, once we note that, by the assumption that each

h_s is one-to-one, for each $s \in Q$, $x \in I^+$ such that $(s,x,sx) \in S_1$ there is a unique string $x' = h_s^{-1}(x) \in I'^+$ such that $h_s(x') = x$; this is, of course, the analogue of Lemma 4.2c.

Let M and M' be two machines, and $h:Q \times I' \to I^+$; we say that a subset $Q_1 \subseteq Q$ is *closed under* h if, for each $s \in Q_1$, $x' \in I'$, $sh_s(x') \in Q_1$. If there is also a map $\phi:Q_1 \overset{\text{onto}}{\to} Q'$ which satisfies (4.1), then we will say that M' *divides* M *with feedback encoding*; we call Q_1 the *core* of the division.

Corollary

If M' divides M with feedback encoding in such a way that the map $h_s:I'^+ \to I^+$ is one-to-one for each s, then $S(M')$ zero-free divides $S(M)$.

It is the corollary to Theorem 4.5, rather than the theorem itself, which has a converse. To prove it, we need one additional concept from the theory of semigroups. If S is a semigroup and $s,t \in S$ then s *divides* t, written $s|t$ if there is an r such that either $sr = t$ or $rs = t$; t is *prime* if there is no s which divides it. Note that for any machine M the only primes in $S(M)$ are the elementary triples.

Theorem 4.6

Let M and M' be machines such that every state of M' is reachable and $S(M')$ zero-free divides $S(M)$. Then M' divides M with feedback encoding, and the extended map $h_s:I'^+ \to I^+$ is one-to-one for each s.

PROOF:

Let $S \subseteq S(M)$ be the subsemigroup for which there is a zero-free homomorphism $\Phi:S \overset{\text{onto}}{\to} S(M')$. We first show that if $\Phi(b)$ is prime, then b is. For suppose b is not prime; since 0 is not prime and $\Phi(0) = 0$, $b \neq 0$. Then $b = b_1b_2$, so that $\Phi(b) = \Phi(b_1)\Phi(b_2)$ is also not prime.

Next, we show that $\{i(b)|b\varepsilon S\} = \{f(b)|b\varepsilon S\}$. Let $b = (t,x,s)$ and $\Phi(b) = (t',x',s')$. Since every state of M' is reachable, there is a $d' = (r',y',t')$. Since $d'\Phi(b) \neq 0$ we can find d such that $\Phi(d) = d'$ and $db \neq 0$. Thus $f(d) = i(b)$ and $\{i(b)|b\varepsilon S\} \subseteq \{f(b)|b\varepsilon S\}$. The other direction follows similarly, except that there we rely not on the fact that every state of M' is reachable, but rather only on the fact that M' is a (complete) machine. Let $Q_1 = \{f(b)|b\varepsilon S\}$. Since for each $s \varepsilon Q_1$ there are triples b_1 and b_2 such that $i(b_1) = s = f(b_2)$ we have all the machinery needed to prove the natural analogue of Lemmas 4.3 and 4.4. The relations $\equiv_\tau$ and $\equiv_\perp$ have finite index, equal to $1+|Q_1|$; $b_1 \equiv_\perp b_2$ if and only if $f(b_1) = f(b_2)$, $b_1 \equiv_\tau b_2$ if and only if $i(b_1) = i(b_2)$, and $[0]_\tau = [0]_\perp = \{0\}$. Exactly as in Theorem 4.5, we can now show that $b_1 \equiv_\tau b_2$ implies $\Phi(b_1) \equiv_\tau \Phi(b_2)$ and $b_1 \equiv_\perp b_2$ implies $\Phi(b_1) \equiv_\perp \Phi(b_2)$, so that there is a map $\phi: Q_1 \overset{\text{onto}}{\to} Q'$ such that $\Phi((s,x,t)) = (\phi(s),x',\phi(t))$.

Let $|I'| = n$ and suppose that for some $s' \varepsilon Q'$ the elementary triples b'_j for which $i(b'_j) = s'$ are $b'_j = (s',x'_j,t'_j)$. Choose any $s \varepsilon Q_1$ such that $\phi(s) = s'$. We show that for each j there is a triple b_j with $i(b_j) = s$ such that $\Phi(b_j) = b'_j$. For, if d is any triple with $f(d) = s$ then $\Phi(d)b'_j \neq 0$, so there is a $b_j \varepsilon \Phi^{-1}(b'_j)$ for which $db_j \neq 0$; certainly $i(b_j) = s$.

Then, as we have shown, each such b_j is prime in S.

Now, for each $s \varepsilon Q_1$, $x' \varepsilon I'$, define $h_s(x')$ to be that string x for which $\Phi((s,x,sx)) = (\phi(s),x',\phi(sx))$. Then ϕ and h satisfy (4.1), so that M' divides M with feedback encoding, where the core of the division is Q_1. Since Φ is a function, each $h_s: I' \to I^+$ must be one-to-one.

We now show that the map h which we have defined extends so that for each $s \in Q_1$, $h_s: I'^+ \to I^+$ is one-to-one. We have already shown that $h_s: I' \to I^+$ is one-to-one. That argument, which shows that the preimage of a prime is prime, can extend to show that for each n, $h_s: I'^n \to I^+$ is one-to-one, but this is not sufficient. What we need to prove, as in Lemma 4.2, is that for each $x \in I^+$ such that $(s,x,sx) \in S$, there is a unique string $x' \in I'^+$ such that $h_s(x') = x$. As noted, this is true if (s,x,sx) is prime. We show that each triple $b \in S$ which is not prime has a unique prime decomposition; i.e., if $b = b_1 \ldots b_n = d_1 \ldots d_m$, where each b_i and d_j is prime, then n = m and $b_i = d_i$, for i = 1,...,n. This is certainly true in S(M'). If $b' = (s',x',t')$ is prime then $\ell(x') = 1$. Thus if the $b_i' = (p_i',x_i',q_i')$ and the $d_i' = (u_i',y_i',v_i')$ are primes and $(s',x',t') = b_1' \ldots b_n' = d_1' \ldots d_m'$ then $n = m = \ell(x')$. Then $x' = x_1' \ldots x_n' = y_1' \ldots y_n'$, so that for i = 1,...,n, $x_i' = y_i'$. If $b' = b_1' \ldots b_n' = d_1' \ldots d_n'$ then $p_1' = u_1' = s'$, so that $u_1'y_1' = p_1'x_1' = q_1' = v_1' = u_2' = p_2'$, etc. With this at our disposal, suppose that $b = b_1 \ldots b_n = d_1 \ldots d_m$. Then $\Phi(b) = \Phi(b_1) \ldots \Phi(b_n) = \Phi(d_1) \ldots \Phi(d_m)$. Since each b_i and d_j is prime, each $\Phi(b_i)$ and $\Phi(d_j)$ is, so that n = m and, for i = 1,...,n, $\Phi(b_i) = \Phi(d_i)$. Now $i(d_1) = i(b_1) = i(b)$, and since h_s is one-to-one on I', this means that $d_1 = b_1$. It then follows that $i(d_2) = i(b_2)$, so that $d_2 = b_2$, <u>etc.</u> Hence, each element of S has a unique prime decomposition.

Now, for $s \in Q_1$ and $x' = x_1' \ldots x_n' \in I'^+$ define $h_s(x') = h(s,x_1')h(sx_1',x_2') \ldots h(sx_1' \ldots x_{n-1}',x_n')$. If $h_s(x_1' \ldots x_n') = h_s(y_1' \ldots y_m')$ then $(s,x,sx) = (s,h_s(x_1'),sh_s(x_1'))\ b_1 = (s,h_s(y_1'),sh_s(y_1'))b_2$.

Since $(s,h_s(x_1'),sh_s(x_1'))$ and $(s,h_s(y_1'),sh_s(y_1'))$ are primes, they must be equal, since (s,x,sx) has a unique prime decomposition, and hence $x_1' = y_1'$. It follows inductively that $x_2' = y_2'$,..., and finally that $n = m$ and $x_1' \ldots x_n' = y_1' \ldots y_n'$, so that $h_s: I'^+ \to I^+$ is one-to-one.

We should note that it is possible for M to simulate M' with feedback encoding without having each map $h_s: I'^+ \to I^+$ be one-to-one, even if every state in each machine is reachable.

Example 4.2

Let M and M' be as in Figure 4.2. Let $\phi(p_i) = s_i$ and suppose that h has the following table:

h	α	β
p_1	a	ab
p_2	a	bc
p_3	c	b

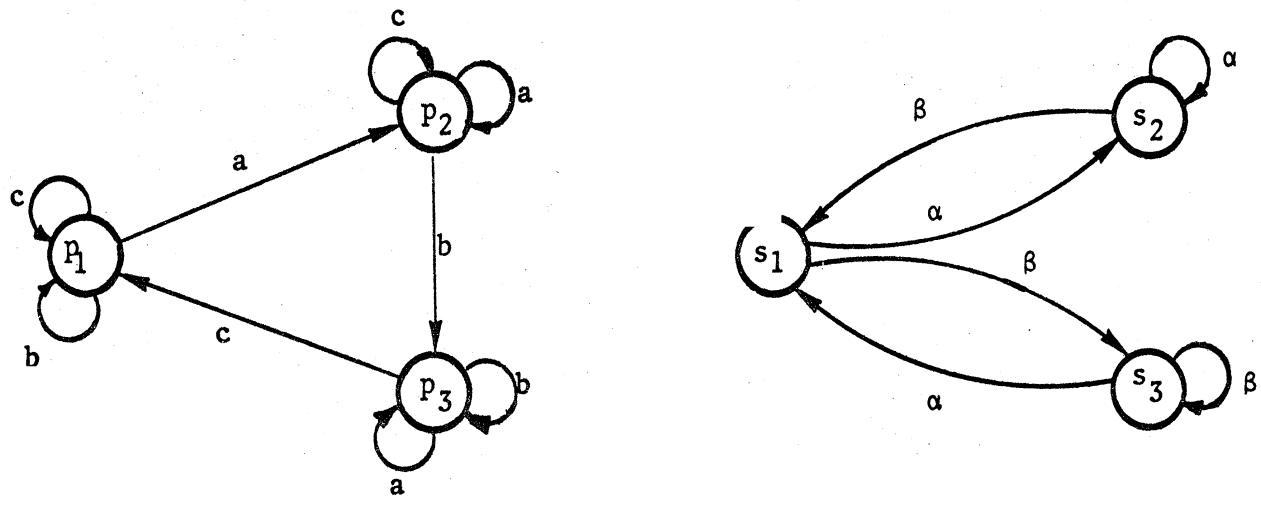


Figure 4.2. Another simulation with feedback encoding.

It is easy to verify that ϕ and h define a simulation with feedback encoding, and, in fact, that $h_{p_i} : I' \to I^+$ is one-to-one for each i. But $h(p_1,\alpha)h(p_1h(p_1,\alpha),\beta) = ah(p_2,\beta) = abc = h(p_1,\beta)h(p_3,\alpha) = h(p_1,\beta)h(p_1h(p_1,\beta),\alpha)$, so that $h_{p_1}(\alpha\beta) = h_{p_1}(\beta\alpha)$.

We have thus shown that the semigroups $S(M)$ reflect quite accurately the relationship of M to other machines as far as realization or simulation with feedback encoding. One important feature of the classical semigroup of a machine is that for every finite semigroup S there is a machine such that $S = S(M)$. A similar result cannot be possible for the semigroups $S(M)$; for example, it is clearly impossible for $S(M)$ to ever be a group. We can, however, characterize those infinite semigroups S such that, for some machine, $S = S(M)$. To do this, we need some preliminary definitions.

Let P be a partially ordered set under the relation $\leq$. If $a,b \in P$ we say that b *covers* a if $a < b$ and there is no $c \in P$ such that $a < c < b$; the *cover of* a, cov(a), is the set of all b which cover a. We will call a partially ordered set P an n-*tree* if it has the following properties:

i. P has a least element;

ii. for each $a \in P$, $|\text{cov}(a)| = n$;

iii. if $a \neq b$ then $\text{cov}(a) \cap \text{cov}(b) = \emptyset$. (4.3)

For any semigroup S we can define a relation $\leq$ by defining $a < b$ if there is an element c such that $ac = b$, and $a \leq b$ if either $a = b$ or $a < b$. This relation may, but need not, be a partial order, since it is not necessarily antisymmetric.

Theorem 4.7

Let S be an infinite semigroup. Then there is a machine M with $|I| = n$ such that $S = S(M)$ if and only if

i. S has a zero, 0.

ii. The relations $\equiv_\perp$ and $\equiv_\tau$ have the same finite index, and $[0]_\tau = [0]_\perp = \{0\}$.

iii. If $st \neq 0$ then $st \in [s]_\tau \cap [t]_\perp$.

iv. Each block of the equivalence relation $\equiv_\tau$, except for $[0]_\tau$, is a disjoint union of n n-trees under the relation $\leq$.

PROOF:

We have already shown the necessity of each condition except iv. Notice first that $\leq$ must be antisymmetric, for if $(s,x,t) \leq (u,y,v)$ then $s = u$ and, more important, $\ell(y) > \ell(x)$. For each state $s \in Q$ we can find exactly n primes, the elementary triples $b_i = (s,x_i,sx_i)$, $i = 1,\ldots,n$. For each triple b_i, $cov(b_i) = \{(s,x_ix_j,sx_ix_j) | x_j \in I\}$, etc. Thus, under $\leq$, each elementary triple is the least element of an n-tree, and each block B of $\equiv_\tau$ is the union of the n n-trees generated by the elementary triples in B.

To prove sufficiency, let m+1 be the index of $\equiv_\tau$. We will define a machine with state set $Q = \{q_1,\ldots,q_m\}$ and input set $I = \{x_1,\ldots,x_n\}$ such that $S(M) = S$.

Let the blocks of $\equiv_\tau$ be $B_0 = [0]_\tau$, $B_1,\ldots,B_m$, and let the n-trees of B_i be $T_{i1},\ldots,T_{in}$.

Let the blocks of $\equiv_\perp$ be $D_0 = [0]_\perp$, $D_1,\ldots,D_m$. There is a natural correspondence between the B_i and the D_j. If $s \in t^\tau$ then $st \neq 0$, so $t \in s^\perp$, and conversely. With each $B_i = [s_i]_\tau$ associate that D_j for which $D_j = [t_j]_\perp$ and $t_j^\perp = B_i$. If necessary, renumber the D_j so that B_i is associated in this manner with D_i.

With each $s \in S$ we will associate a triple. If $s \in B_i \cap D_j$ then s will be associated with $(q_i,-,q_j)$; the second element of the triple is yet to be assigned. For each block B_i, let $s_1,\ldots,s_n$ be the least elements of the n-trees of B_i, and assign as the second element of the triple associated with s_j the symbol x_j. We proceed inductively; suppose that the string y has been assigned to element t, and let $\{t_1,\ldots,t_n\} = \text{cov}(t)$. Then assign as the second element of the triple for t_j the string yx_j.

This association between triples and elements of S is an isomorphism between S and $S(M)$, where M is the machine whose elementary triples are exactly the mn least elements of the n-trees of S.

Before proceeding we mention one interesting auxiliary property of the semigroups $S(M)$. Suppose that the machine M is actually a finite-state acceptor; that is, there is a starting state q_0 and a set of final states $F \in Q$, so that we can associate with M the event $E(M) \subseteq I^+$ of strings x for which $q_0x \in F$.

Theorem 4.8

If M is a finite state acceptor then there is a right ideal R and a left ideal L of $S(M)$ such that

$E(M) = \{\text{proj}_2(s) \mid s \in L \cap R - \{0\}\}$.

PROOF:

Let $L = \{s \mid f(s) \in F\} \cup \{0\}$ and $R = \{s \mid i(s) = q_0\} \cup \{0\}$. Then L is a left ideal since, for any $t \in S(M)$, $s \in L$, if $ts \neq 0$ then $f(ts) = f(s) \in F$. Similarly, R is a right ideal. Then $L \cap R - \{0\}$ is the set $\{s \mid i(s) = q_0$ and $f(s) \in F\}$, so that the second components of the strings of $L \cap R - \{0\}$ are all the walks from q_0 to F; i.e., $E(M)$.

We now discuss briefly a new finite semigroup for a machine M. Let $H(M)$ be the set of ordered pairs $(s,t) \in Q \times Q$ such that, for some x, $(s,x,t) \in S(M)$, together with a zero element, 0, with the same multiplication rule as in $S(M)$; i.e., $(s,t)(u,v) = (s,v)$ if $t = u$, and $(s,t)(u,v) = 0$ if $t \neq u$. We call $H(M)$ the *reachability semigroup* of M.

Theorem 4.9

For any machine M, $H(M)$ is a homomorphic image of $S(M)$.

PROOF:

Let $\Phi: S(M) \to H(M)$ be defined by $\Phi((s,x,t)) = (s,t)$ and $\Phi(0) = 0$. Let b_1, b_2 be elements of $S(M)$; if either is zero then $b_1 b_2 = 0$, and $\Phi(b_1)\Phi(b_2) = 0 = \Phi(b_1 b_2)$. Suppose that neither b_1 nor b_2 is zero. If their product is zero then $f(b_1) \neq i(b_2)$, so that certainly $\Phi(b_1)\Phi(b_2) = 0 = \Phi(b_1 b_2)$. Finally, if $b_1 = (s,x,t)$, and $b_2 = (t,y,r)$, then $\Phi(b_1) = (s,t)$, $\Phi(b_2) = (t,v)$, and $\Phi(b_1 b_2) = \Phi((s,xy,r)) = (s,r) = \Phi(b_1)\Phi(b_2)$. Thus Φ is a homomorphism. Note, incidentally, that since $\mathrm{Ker}(\Phi) = \{0\}$, Φ is also a zero-free homomorphism by Theorem 4.2b.

Because the map Φ, as defined in the theorem, is so obviously important in studying the H-semigroups we find it convenient to also consider it as an operator ∂ on S-semigroups, so that for any M, $\partial(S(M)) = H(M)$.

Not surprisingly, zero-free homomorphisms between S-semigroups tend to induce zero-free homomorphisms between the corresponding H-semigroups. We state this in the context of Theorem 4.3.

Theorem 4.10

If M realizes M' with feedback encoding then there is a zero-free homomorphism from $H(M)$ onto $H(M')$.

PROOF:

Let $\Phi: S(M) \to S(M')$ be the zero-free homomorphism guaranteed by Theorem 4.3. Note that we showed in the proof of Theorem 4.3, although we did not have the notation to express it, that if $\partial(b_1) = \partial(b_2)$ then $\partial(\Phi(b_1)) = \partial(\Phi(b_2))$. Thus, for each $(s,t) \in H(M)$ choose $b \in S(M)$ such that $\partial(b) = (s,t)$ and define $\Psi((s,t)) = \partial\Phi(b)$; by the preceeding discussion the value of $\Psi((s,t))$ is independent of the choice of $b \in \partial^{-1}((s,t))$.

Certainly $\mathrm{Ker}(\Psi) = \{0\}$. Suppose that $d_1 = (s,t)$ and $d_2 = (t,r)$ are elements of $H(M)$; for any choice of $b_i \in \partial^{-1}(d_i)$, $b_1 b_2 \neq 0$, and $\partial(b_1 b_2) = d_1 d_2$. Since Φ is zero-free, $\Phi(b_1)\Phi(b_2) = \Phi(b_1 b_2)$, and so $\partial\Phi(b_1)\partial\Phi(b_2) = \partial\Phi(b_1 b_2) \neq 0$. But $\partial\Phi(b_1 b_2)$ is just $\Psi(d_1 d_2)$, and so $\Psi(d_1)\Psi(d_2) = \partial\Phi(b_1)\partial\Phi(b_2) = \Psi(d_1 d_2)$.

If $\Psi(d_1)d_2' \neq 0$, choose $b_1' \in \partial^{-1}(\Psi(d_1))$ and $b_2' \in \partial^{-1}(d_2')$; then $b_1' b_2' \neq 0$. In particular, if $b_1 \in \partial^{-1}(d_1)$ then $\Phi(b_1) \in \partial^{-1}(\Psi(d_1))$, so that $\Phi(b_1)b_2' \neq 0$ for any $b_2' \in \partial^{-1}(d_2')$. Then, since Φ is zero-free, there is a $b_2 \in \Phi^{-1}(b_2')$ such that $b_1 b_2 \neq 0$. Let $d_2 = \partial(b_2)$, then $d_1 d_2 \neq 0$ and, by definition, $\Psi(b_2) = d_2'$. Similarly, if $d_1'\Psi(d_2) \neq 0$ then there is some $d_1 \in \Psi^{-1}(d_1')$ such that $d_1 d_2 \neq 0$. Thus, Ψ is a zero-free homomorphism.

The converse of this theorem is not true. As an example, let M_1 and M_2 be two machines whose digraphs have the same transitive closure. Then $H(M_1) \cong H(M_2)$, but it is not necessarily true that M_1 isomorphically realizes M_2 with feedback encoding. For an example, see Figure 4.2; since the two digraphs D(M) and D(M') are strong they have the same transitive closure. Thus the progressive relationships we found between the S-semigroups and the concepts of realization and simulation with

feedback encoding do not hold for the H-semigroups. In fact, Theorem 4.10 is a special case of

Theorem 4.11

If M simulates M' with feedback encoding then there is a zero-free homomorphism between H(M) and H(M').

PROOF:

Let the simulation be defined by maps ϕ and h. Define $\Phi((s,t))$ to be $(\phi(s),\phi(t))$, and $\Phi(0) = 0$. If we have $(s,t),(t,r) \in H(M)$ then $\Phi((s,t))\Phi((t,r)) = (\phi(s),\phi(t))(\phi(t),\phi(r)) = (\phi(s),\phi(r)) = \Phi((s,t)(t,r))$. The other properties of zero-free homomorphisms follow as in previous proofs, and will be omitted.

As with the S-semigroups we can define the relations $\equiv_\tau$ and $\equiv_\perp$ and the functions f and i. We also need the notion of an elementary element of H(M); (s,t) is *elementary* if $st \in D(M)$.

Theorem 4.12

If M and M' are machines in which every state is reachable, and if H(M') is a zero-free homomorphic image of H(M), then M simulates M' with feedback encoding.

PROOF:

Let $\Phi: H(M) \to H(M')$ be a zero-free homomorphism. We must first show that Φ induces a map from Q to Q' as in Theorem 4.4. Since the relations $\equiv_\tau$ and $\equiv_\perp$ on S-semigroups were shown to be directly dependent on the functions i and f, most of the results we obtain about them carry through directly for H-semigroups, and the proofs will be omitted.

In particular, $f(b_1) = f(b_2)$ implies $f(\Phi(b_1)) = f(\Phi(b_2))$, and $i(b_1) = i(b_2)$ implies $i(\Phi(b_1)) = i(\Phi(b_2))$, so that Φ induces maps ϕ_i and ϕ_f, and $\Phi((s,t)) = (\phi_i(s),\phi_f(t))$. If $b_1 = (s,t)$ and $b_2 = (t,r)$

are elements of $H(M)$, then $\Phi(b_1)\Phi(b_2) \neq 0$, so that $\phi_i(t) = \phi_f(t)$. Since every state of M is reachable, ϕ_i and ϕ_f agree on each $t \in Q$; call the function thus defined ϕ.

Let (s',t') be elementary in $H(M')$, and let x' be an input such that $s'x' = t'$. We claim, first of all, that for each $s \in \phi^{-1}(s')$ there is a $t \in \phi^{-1}(t')$ such that $(s,t) \in H(M)$. To see this, we first show that if $\phi(s) = s'$ then there are elements b_1 and b_2 in $H(M)$ such that $s = i(b_1) = f(b_2)$. Certainly, there must either be a b_1 for which $i(b_1) = s$ or a b_2 for which $f(b_2) = s$. Suppose the former. Since s' is reachable, there is an element $(r',s') \in H(M')$. Then $(r',s')\Phi(b_1) \neq 0$ so there is some $(r,s) \in H(M)$.
A similar argument completes the demonstration. Now, to show that there is an element (s,t), where $\phi(t) = t'$, we need only choose some element (r,s) and note that since $\Phi((r,s))(s',t') \neq 0$ there is an element (s,t) such that $\Phi((s,t)) = (s',t')$ by the properties of zero-free homomorphisms. But as we have shown, if $\Phi((s,t)) = (s',t')$ then $\phi(t) = t'$.

Thus if $s'x' = t'$, for each s such that $\phi(s) = s'$ there is a state t reachable from s such that $\phi(t) = t'$. Choose any string $x \in I^+$ for which $sx = t$ and define $h_s(x') = t'$. Then the pair (ϕ,h) defines a simulation with feedback encoding. Note however that h_s may not even be one-to-one on symbols.

We now turn our attention to the third and final algebraic structure which we will study. We will first discuss its relationship to realization with feedback encoding, and some of its shortcomings in that regard, and then go on to show how the concept of simulation with feedback encoding applies to a conjecture about these semigroups.

For any machine M, the *S*-semigroup* $S^*(M)$ has for its elements all finite sets of triples [we will write 0 for the empty set $\emptyset$ of triples],

where if $U = \{b_i | i = 1,\ldots,n\}$ and $V = \{d_j | j = 1,\ldots,m\}$ are elements of $S^*(M)$ then $UV = \{b_i d_j | i = 1,\ldots,n; j = 1,\ldots,m\}$. We list some of the properties of $S^*(M)$ in the next theorem; for proofs see [33] and [20]; a state s is *terminal* if for all $x \in I$, $sx = s$.

Theorem 4.13

a) If M isomorphically realizes M' with feedback encoding then $S^*(M) \cong S^*(M')$

b) If M and M' are machines in which each state is reachable and there is at least one nonterminal state, and if $S^*(M) \cong S^*(M')$, then M isomorphically realizes M' with feedback encoding.

These results make it seem reasonable to expect that we can find generalizations in the sense of Theorems 4.3 - 4.6. Suppose, for example, we had M realizing M' homomorphically with feedback encoding, the realization being defined by maps ϕ and h. It would be natural, as would in fact be done in proving Theorem 4.13a, to define, for each triple $b = (s,x,t)$ of M, $\phi^*(b) = (\phi(s), h_s^{-1}(x), \phi(t))$ (ϕ^* is just the map $\Phi: S(M) \to S(M')$ of Theorem 4.3.) We would then define $\Phi(\{b_i | i = 1,\ldots,n\})$ to be $\{\phi^*(b_i)\}$. Let $S = \{b_j\}$ and $T = \{d_k\}$, $S,T \in S^*(M)$, and suppose $ST \neq 0$. For each pair b_j and d_k for which $b_j d_k \neq 0$, it will follow that $\phi^*(b_j d_k) = \phi^*(b_j)\phi^*(d_k)$, so that $\Phi(ST) \subseteq \Phi(S)\Phi(T)$. But, should there be a pair b_j, d_k such that $b_j d_k = 0$ but $\phi^*(b_j)\phi^*(d_k) \neq 0$, then $\Phi(ST) \subsetneq \Phi(S)\Phi(T)$, so that Φ would not even be a zero-free homomorphism. Such a pair b_j, d_k would certainly exist unless $|Q| = |Q'|$.

The preceding discussion, of course, only shows that one particular approach to the problem is infeasible. Recalling the proofs for previous theorems, as well as the proof of Theorem 4.13b which provided much

of the motivation for the work in this chapter, where we showed that a zero-free homomorphism between two S-semigroups or two H-semigroups induces a map between the state-sets of the two machines, we can show

Theorem 4.14

Let M and M' be two machines where M is strong. If there is a zero-free homomorphism $\Phi: S^*(M) \to S^*(M')$ and a map $\phi: Q \to Q'$ such that for each singleton $b = \{(s,x,t)\} \in S^*(M)$, $\Phi(b) = \{(\phi(s),x',\phi(t))\}$, then $|Q| = |Q'|$.

PROOF:

Suppose not, and let $\phi(q_1) = \phi(q_2) = q'$. Since q_1 is reachable there is some $b_1 = \{(r,x,q_1)\} \in S^*(M)$. For $y,z \in I^+$, let $b_2 = \{(q_1,y,q_1y), (q_2,z,q_2z)\}$. Since $b_1b_2 \neq 0$, $\Phi(b_1)\Phi(b_2) = \{(\phi(r),x'y',\phi(q_1y)),(\phi(r),x'z',\phi(q_2z))\} = \Phi(b_1b_2) = \{(\phi(r),(xy)',\phi(q_1y))\}$. Among other things, this would imply that if q_3 is reachable from q_1 and q_4 is reachable from q_2 then $\phi(q_3) = \phi(q_4)$. Since M is strong, every state is reachable from every other, so that ϕ must be one-to-one.

The hypothesis that M be strong is probably more than is needed to reach the conclusion of the theorem, but it certainly does show that the S^*-semigroups are not especially useful in a study of realization, with or without feedback encoding. On the other hand, Hedetniemi and Fleck [20] made the following conjecture, which we can use the concept of simulation with feedback encoding to settle in almost all cases:

Let M and M' be any two strong machines with the same number of states. Then $S^*(M)$ is isomorphic to a subsemigroup of $S^*(M')$, and $S^*(M')$ is isomorphic to a subsemigroup of $S^*(M)$.

We first prove a crucial result, to which we alluded earlier. It gives both the sufficient condition which we mentioned in connection with Theorem 4.5, and also shows, as we promised in Chapter III, that simulation with feedback encoding is an essentially uninteresting concept.

Theorem 4.15

Let M be a strong machine with at least two inputs, and let M' be any machine with $|Q'| \leq |Q|$. Then M' divides M with feedback encoding, and the feedback encoding can be chosen in such a way that the maps $h_s: I'^+ \to I^+$ are one-to-one.

PROOF:

Since M is strong, for any $s,t \in Q$ there is a string w_{st} with $\ell(w_{st}) \geq 1$ such that $sw_{st} = t$.

Let $\overline{Q}$ be any subset of Q with cardinality $|Q'|$, and $\phi: \overline{Q} \overset{\text{onto}}{\to} Q'$ be any map. Let $I' = \{x'_1, \ldots, x'_a\}$ and let $\eta \neq \overline{\eta}$ be two elements of I. For each $s \in \overline{Q}$, $x'_j \in I'$, let $t = \phi^{-1}(sx'_j)$ and define $h_s(x'_j) = \eta^j \overline{\eta} w_{qt}$, where $q = s\eta^j\overline{\eta}$. Then $\phi(sh_s(x'_j)) = \phi(s)x'_j$, so that M' divides M with feedback encoding; if $|Q| = |Q'|$ then M simulates M' with feedback encoding. Each map h_s is certainly one-to-one on strings. We show that the extended maps are also one-to-one.

Let $j_1 \ldots j_m$ and $k_1 \ldots k_{m'}$ be two strings from $(I')^+$ and let $h_s(j_1 \ldots j_m) = h_s(k_1 \ldots k_{m'}) = w$. Then, by definition, there are states $r,t \in \overline{Q}$ such that $w = h_s(j_1)h_r(j_2 \ldots j_m) = h_s(k_1)h_t(k_2 \ldots k_{m'})$. But there is a unique positive integer n such that the prefix of w having length n+1 is the string $\eta^n\overline{\eta}$. This uniquely determines

$j_1 = k_1 = x'_n$, so that $r = t = sx'_n$. Then $h_r(j_2 \ldots j_m) = h_r(k_2 \ldots k_{m'})$, and we can repeat the above process until we arrive at $m = m'$ and $j_p = k_p$, $p = 1,2,\ldots,m$.

Corollary

If M is strong, with at least two inputs, and M' has $|Q'| \leq |Q|$, then $S(M')$ zero-free divides $S(M)$.

Corollary

If M is strong, and M' has $|Q'| \leq |Q|$ then M' divides M with feedback encoding.

PROOF:

We need only cover the case in which $|I| = 1$. As in the theorem, we choose any $\overline{Q} \leq Q$ with $|\overline{Q}| = |Q'|$ and any map $\phi:\overline{Q} \overset{\text{onto}}{\to} Q'$. For $s \in \overline{Q}$ and $x'_j \in I'$, let $t = \phi^{-1}(\phi(s)x'_j)$, and define $h_s(x'_j) = w_{st}$. Then $\phi(s)x'_j = \phi(sh_s(x'_j))$, so M' divides M with feedback encoding.

Theorem 4.16

Let M be a strong automaton with n states and at least two inputs, and let M' be an automaton with $n' \leq n$ states. Then $S^*(M')$ is isomorphic to a subsemigroup of $S^*(M)$.

PROOF:

We use the maps ϕ and h of Theorem 4.15 to define the isomorphism. Let $b' = \{(s',x',t')\}$ be a singleton in $S^*(M')$ and set $g(b') = \{(s,h_s(x'),t) \mid \phi(s) = s'\}$; since ϕ is one-to-one, $g(b')$ is a singleton. Note that this implies that $\phi(t) = t'$. Also since h_s is one-to-one for each $s \in \overline{S}$, $g(b'_1) = g(b'_2)$ if and only if $b'_1 = b'_2$; i.e., g is one-to-one. Let $b'_1 = \{(s'_1,x'_1,r')\}$ and $b'_2 = \{(r',x'_2,t'_2)\}$. Let $b_1 = \{(s_1,x_1,r)\} = g(b'_1)$, $b_2 = \{(r,x_2,t_2)\} = g(b'_2)$. Then $b_1 \circ b_2 = \{(s_1,h_{s_1}(x'_1x'_2),t_2)\}$ is a singleton of $S^*(M)$ and, as $\phi(t_2) = t'_2$, $b_1 \circ b_2 = g(b'_1 \circ b'_2)$. Thus

$g(b_1')\circ g(b_2') = g(b_1'\circ b_2')$. On the other hand, if $b_1' = \{(s_1',x_1',t_1')\}$, $b_2' = \{(s_2',x_2',t_2')\}$, $g(b_1') = \{(s_1,x_1,t_1)\}$, $g(b_2') = \{(s_2,x_2,t_2)\}$ and $t_1' \neq s_2'$ then $t_1 \neq s_2$, so that $b_1'\circ b_2' = 0$ and $g(b_1')\circ g(b_2') = 0$.

Now let V' be any element of $S^*(M')$; V' is a finite set of triples of M'. Extend g to g^* by $g^*(V') = \{g(b')|b\varepsilon V'\}$. Let $S^*_{M'}(M)$ be $\{g^*(V')|V'\varepsilon S^*(M')\}$.

Now, if $V_1', V_2' \varepsilon S^*(M')$,

$$\begin{aligned} g^*(V_1')\circ g^*(V_2') &= [\cup\{g(b_i')|b_i'\varepsilon V_1'\}]\circ[\cup\{g(d_j')|d_j'\varepsilon V_2'\}] \\ &= \bigcup_{i,j}\{g(b_i')\circ g(d_j')|b_i'\varepsilon V_1', d_j'\varepsilon V_2'\} \\ &= \bigcup_{i,j}\{g(b_i'\circ d_j')|b_i'\varepsilon V_1', d_j'\varepsilon V_2'\} \\ &= \cup\{g(f_k')|f_k'\varepsilon V_1'\circ V_2'\} \\ &= g^*(V_1'\circ V_2'). \end{aligned}$$

Thus $S^*_{M'}(M)$ is a subsemigroup of $S^*(M)$, and g^* is a homomorphism. We wish to show that g^* is 1-1. Suppose $g^*(V_1') = g^*(V_2')$. Choose a triple $b_1'\varepsilon V_1'$, and let $\{b\} = g(\{b_1'\})$. Then there is a triple $b_2'\varepsilon V_2'$ such that $\{b\} = g(\{b_2'\})$. If $b = (s,w,t)$, $b_1' = (\phi(s),x_1',\phi(t))$ and $b_2' = (\phi(s),x_2',\phi(t))$, where $w = h_s(x_1') = h_s(x_2')$. Then, by the lemma, $x_1' = x_2'$, so $b_1' = b_2'$ and $V_1' \subseteq V_2'$. The symmetric argument gives $V_1' = V_2'$ so that g^* is 1-1, and hence g^* is an isomorphism between $S^*(M')$ and $S^*_{M'}(M)$.

Corollary

Let M and M' be strong machines, with $|Q| = |Q'|$. Then, unless M is autonomous but M' is not, $S^*(M')$ is isomorphic to a subsemigroup of $S^*(M)$.

OF:

If M is not autonomous then $S^*(M') \cong S^*_{M'}(M)$, by the theorem.

On the other hand, if both machines are autonomous then they are isomorphic, so the result follows from Theorem 4.13a.

This corollary settles the Hedetniemi-Fleck conjecture except that it does not decide the case in which M is the unique (up to isomorphism) autonomous, strong, n-state machine C_n and M' is not autonomous. Due to the dependence of Theorem 4.16 on the fact that M had at least two inputs, we prefer to make the following counter-conjecture:

Conjecture

There is a strong, n-state machine M' such that $S^*(M')$ is not isomorphic to any subsemigroup of $S^*(C_n)$.

CHAPTER V

DISTINGUISHING SEQUENCES

Having presented some of the theoretical properties of realizations with feedback encoding, we now turn our attention to a specific applications area. Actually, we have already discussed one application in Example 3.4, where we showed that the use of feedback encoding can lead to more efficient cascade realizations.

Given a behavior which is to be realized, one often places additional requirements on the realizing machine; perhaps the simplest such requirement is that the machine be reduced. Another requirement is that the machine have a distinguishing sequence, an input string whose output sequence uniquely identifies the machine's starting state. In this chapter we will develop techniques for realizing some machine M', with feedback encoding, by a machine M having a distinguishing sequence. It is important to note that the machine which has the distinguishing sequence is M, and not M together with the feedback encoder.

Up to now we have been concerned only with the realization of state-behaviors, but in this chapter, where we will be studying distinguishing sequences, we will instead study the realization of input-output behaviors. We will need to define the realization with feedback encoding of a machine having outputs in two ways, one for the Moore case and one for the Mealy.

If $M = \langle Q, I, \delta, \lambda, Y \rangle$ and $M' = \langle Q', I', \delta', \lambda', Y' \rangle$ are Moore machines, then M *realizes* M' *with feedback encoding* if there are maps $\phi: Q \xrightarrow{\text{onto}} Q'$, $h: Q \times I' \to I$, a type I feedback encoder, and $g: Y \to Y'$, such that

$$\phi(q)x' = \phi(qh_q(x'))$$

and

$$\lambda'(\phi(q)) = g(\lambda(q)). \tag{5.1}$$

Since in most of what follows the realizations will be isomorphic, the *output decoder*, g, may often be just a one-to-one correspondence, and can be omitted if we assume, without loss of generality that $\lambda'(\phi(q)) = \lambda(q)$, giving the alternate conditions

$$\begin{aligned} \phi(q)x' &= \phi(qh_q(x')) \\ \lambda'(\phi(q)) &= \lambda(q) \end{aligned} \tag{5.2}$$

Similarly, if M and M' are Mealy machines, then M *realizes* M' *with feedback encoding* if there are maps $\phi: Q \overset{\text{onto}}{\to} Q'$, $h: Q \times I' \to I$, a type I feedback encoder, and $g: Y \to Y'$, such that

$$\begin{aligned} \phi(q)x' &= \phi(qh_q(x')) \\ \lambda'(\phi(q),x') &= g(\lambda(q,h_q(x')) \end{aligned} \tag{5.3}$$

or, if we take Y = Y' and g to be the identity map,

$$\begin{aligned} \phi(q)x' &= \phi(qh_q(x')) \\ \lambda'(\phi(q),x') &= \lambda(q,h_q(x')) \end{aligned} \tag{5.4}$$

Example 5.1

Let M_1 and M_2 be the Mealy machines in Figure 5.1. Then M_2 realizes M_1 with feedback encoding. For example, we

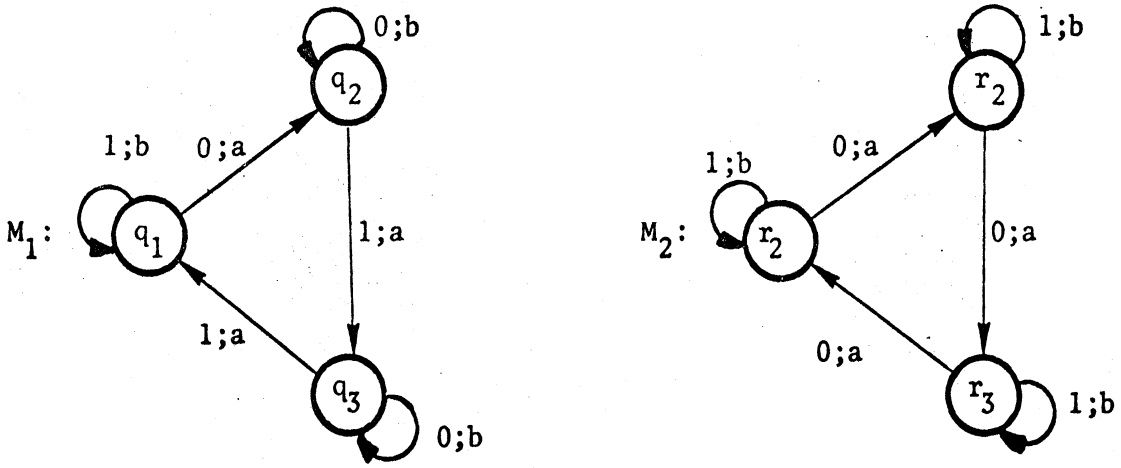


Figure 5.1. Realization with feedback encoding.

check (5.4) at state r_2, where $\phi(r_i) = q_i$. For $x' = 0$, $h_{r_2}(x') = 1$, so that $q_2 0 = q_2 = \phi(r_2 1)$, and $\lambda_1(q_2,0) = b = \lambda_2(r_2,1)$; for $x' = 1$, $h_{r_2}(x') = 0$, $q_2 1 = q_3 = \phi(r_2 0)$ and $\lambda_1(q_2,1) = a = \lambda_2(r_2,0)$.

From Example 5.1, we can see the relationship between the state transition graphs of two Mealy machines when one realizes the other isomorphically with feedback encoding. Not only are the digraphs isomorphic, but the isomorphism preserves the output labels, so that only the input labels are permuted.

As is well known, the concepts of Mealy and Moore machines are essentially interchangeable. We show in Theorem 5.1 that this interchangeability carries over to realizing machines.

Theorem 5.1

Let M and M' be Mealy machines, and $\tilde{M}$, $\tilde{M}'$ their Moore equivalents. If M realizes M' with feedback encoding then $\tilde{M}$ realizes $\tilde{M}'$ with feedback encoding.

PROOF:

We are given maps $\phi: Q \xrightarrow{\text{onto}} Q'$ and $h: Q \times I' \to I$ such that $\phi(q)x' = \phi(qh_q(x'))$ and $\lambda'(\phi(q),x') = \lambda(q,h_q(x'))$. Now, $\tilde{M}$ has states q_{pr}, where q_p is a state of M and some transition into q_p has output w_r; $\tilde{\lambda}(q_{pr}) = w_r$. Similarly, $\tilde{M}'$ has states q'_{st}.

Let $\tilde{\phi}$ be the map defined by $\tilde{\phi}(q_{pr}) = q'_{st}$ if and only if $r = t$ and $\phi(q_p) = q'_s$. Let $\tilde{h}$ be defined by $\tilde{h}_{q_{pr}} = h_{q_p}$ for all states $q_{pr} \in \tilde{M}$.

Then, for any state q_{pr} and input $x' \in \tilde{I}'$, $q_{pr}\tilde{h}_{q_{pr}}(x') = q_{st}$ if and only if $q_p h_{q_p}(x') = q_s$ and $\lambda(q_p,h_{q_p}(x')) = w_t$. Thus $\tilde{\phi}(q_{pr}\tilde{h}_{q_{pr}}(x')) = \tilde{\phi}(q_{st})$, $\phi(q_s) = \phi(q_p h_{q_p}(x')) = \phi(q_p)x'$, so that $\tilde{\phi}(q_{pr}\tilde{h}_{q_{pr}}(x')) = \tilde{\phi}(q_{pr})x'$; and also,

$\tilde{\lambda}(q_{pr}, \tilde{h}_{q_{pr}}(x')) = \tilde{\lambda}(q_{pr}\tilde{h}_{q_{pr}}(x')) = \tilde{\lambda}(q_{st}) = w_t = \tilde{\lambda}'(\tilde{\phi}(q_{st})) = \tilde{\lambda}'(\tilde{\phi}(q_{pr}), x')$.

Hence $\tilde{M}$ realizes $\tilde{M}'$ with feedback encoding.

Recall that for a state q and string $x_1 \ldots x_n$, $\beta_q(x_1 \ldots x_n) = \lambda(q, x_1)\lambda(qx_1, x_2) \ldots \lambda(qx_1 \ldots x_{n-1}, x_n)$; of course, in a Moore machine this reduces to $\beta_q(x_1 \ldots x_n) = \lambda(qx_1)\lambda(qx_1x_2) \ldots \lambda(qx_1 \ldots x_n)$. Defining β in this manner we are omitting the output associated with the current state, q, and therefore ignoring a potentially useful item of information. Since this information is usually available, especially in actual circuits, it is worthwhile to examine the effects of this omission on the work to follow. All the constructions which we give will be valid for either definition of β, but bounds involving the lengths of input sequences will be larger by 1 than they would be with the "usual" definition of β. The advantage to the form taken here, as we shall see, is to make it possible to treat certain behavioral characteristics of a machine as though they were strictly structural.

We say that a string x is a *distinguishing sequence* for a machine M if for any states $q \neq q'$, $\beta_q(x) \neq \beta_{q'}(x)$. Clearly [21] any machine which has a distinguishing sequence is reduced, but the converse does not hold. We will first present some examples of machines taken from the literature which do not have distinguishing sequences, and show how each can be isomorphically realized with feedback encoding by a machine having a distinguishing sequence.

Example 5.2 [21, p. 116]

The machine M_1 of Figure 5.2 has no distinguishing sequence. However, machine M_2 certainly realizes M_1 with feedback encoding, since the only difference is a permutation of the input labels at q_1. The string x = 1 1 0 0 1 is a distinguishing

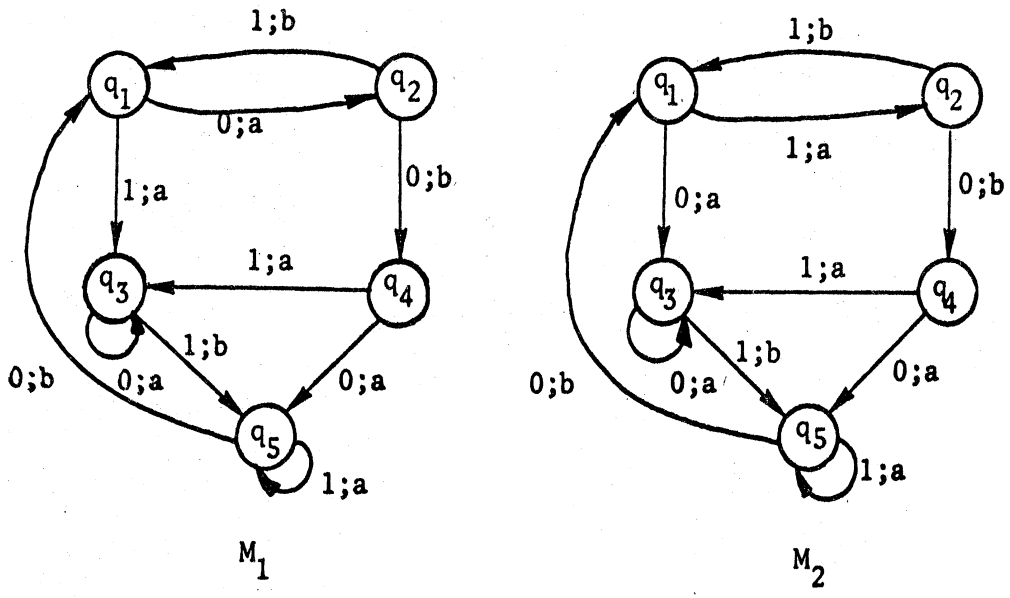


Figure 5.2. A diagnosable realization.

sequence for M_2; in fact, the β-function has the values

$$\beta_{q_1}(x) = a\ b\ a\ a\ b$$
$$\beta_{q_2}(x) = b\ a\ b\ a\ a$$
$$\beta_{q_3}(x) = b\ a\ b\ a\ b$$
$$\beta_{q_4}(x) = a\ b\ b\ a\ b$$
$$\beta_{q_5}(x) = a\ a\ b\ a\ b$$

Note that the action of x is to carry every state to q_5, so that x is also a synchronizing sequence [21]; in general, however, a distinguishing sequence need not be synchronizing, and a synchronizing sequence certainly need not be distinguishing.

Example 5.3 [21, p. 120]

Consider Figure 5.3. Again M_1 has no distinguishing sequence, while M_2, which realizes M_1 with feedback encoding, has one. In fact if x = 00 then

$$\beta_{q_1}(x) = a\ a$$
$$\beta_{q_2}(x) = b\ a$$
$$\beta_{q_3}(x) = a\ b$$
$$\beta_{q_4}(x) = b\ b$$

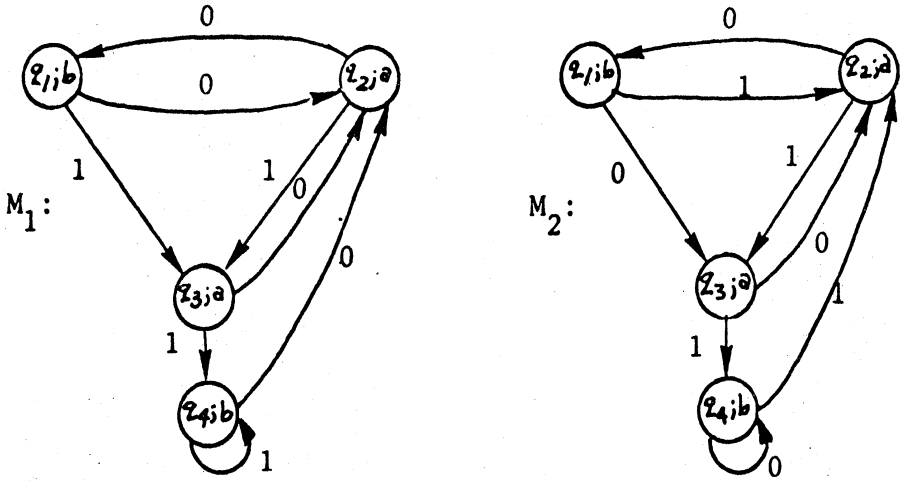


Figure 5.3. Another diagnosable realization.

In the case that all the symbols in a distinguishing sequence x are identical we say that x is a *repeated symbol distinguishing sequence* [27].

Example 5.4 [27, p. 325]

Referring to the machines in Figure 5.4, M_1 has no distinguishing sequence, M_2 realizes M_1 with feedback encoding, and M_2 has a repeated symbol distinguishing sequence; in fact, if $x = 1^4$, then

$$\beta_{q_1}(x) = b\ a\ a\ a$$
$$\beta_{q_2}(x) = a\ a\ a\ a$$
$$\beta_{q_3}(x) = a\ a\ a\ b$$
$$\beta_{q_4}(x) = a\ a\ b\ b$$
$$\beta_{q_5}(x) = a\ b\ b\ a$$
$$\beta_{q_6}(x) = b\ b\ a\ a$$

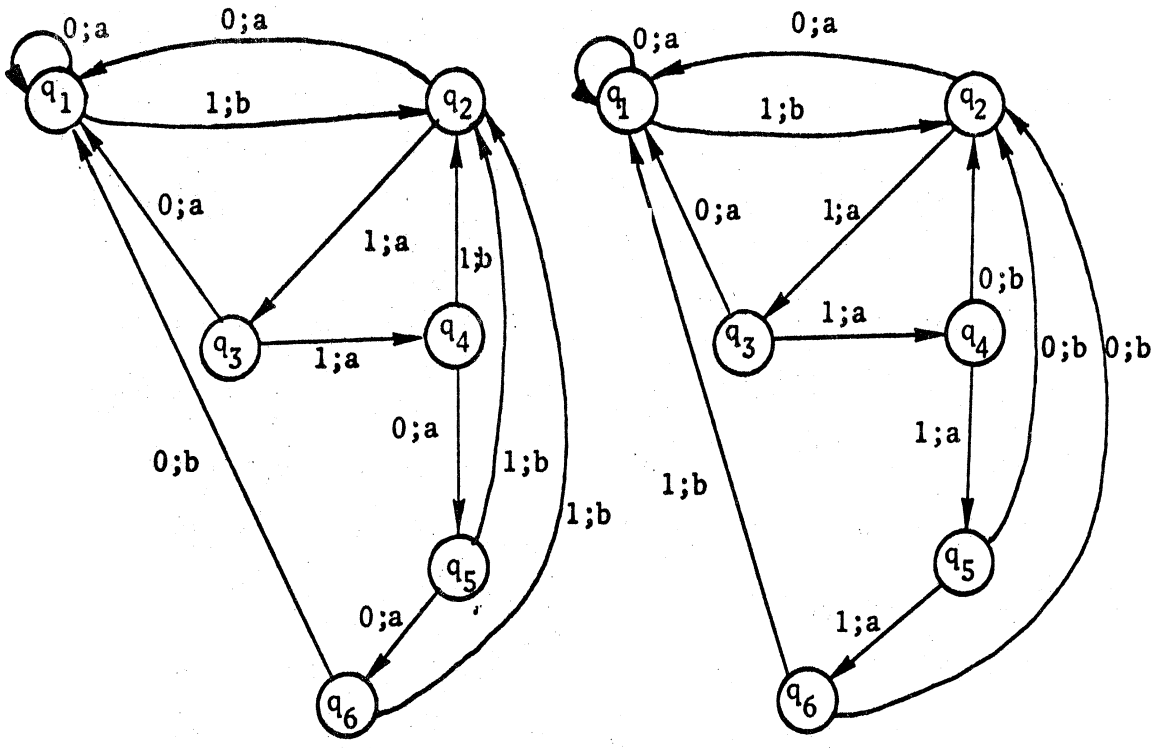


Figure 5.4. A repeated symbol diagnosable realization.

Before we begin our study in detail, a word is in order about just what it is that we wish to accomplish. The existence of a distinguishing sequence is properly a behavioral, rather than a structural, property. Nevertheless, as the previous examples show, feedback encoding techniques can, for some machines, produce realizing machines with distinguishing sequences and yet cause no increase in state size, input set size, or output set size. We will develop some results which will give techniques, in some cases, for finding realizing machines with distinguishing sequences. These techniques will not cover all the possibilities, however. For example, while they could be successfully applied to machine M_1 of Figure 5.2, they would not produce the same realizing machine as is given in Example 5.2. Rather than simply giving techniques, we are more concerned with trying to demonstrate that feedback encoding techniques can be an effective tool. The usefulness of these techniques does not lie solely in the theorems

which we are presenting.

In a machine M, two states q_1 and q_2 are said to *merge* if, for some input $x \in I$, $q_1x = q_2x$; states q_1 and q_2 *converge* if, under some input x, they merge and if, also, $\lambda(q_1,x) = \lambda(q_2,x)$. If q_1 and q_2 converge to q_3 under x we write $(q_1,q_2)x = q_3$. Note that in a Moore machine two states converge if and only if they merge.

Lemma 5.1 [21]

If a reduced machine is convergence-free then it has a distinguishing sequence.

The converse is not true; a machine need not be convergence-free to have a distinguishing sequence (see, for example, machine M_2 of Figure 5.2). However, if a machine is reduced and k states converge to a single state then by state-splitting techniques, and adding additional output symbols, the convergence can be eliminated; this requires adding on the order of $\{\frac{k}{2}\}$ new states and output symbols [27]. Of the two, addition of new states is the more costly, and, as we have noted above, the techniques we present will be geared towards producing no increase in state-set size, although some increase in output-set size will not always be avoidable. We will first concentrate on the problem of designing realizing machines which have repeated symbol distinguishing sequences.

Let $\langle s_i \rangle = \langle s_0,s_1,\ldots,s_{m-1} \rangle$ be a sequence of symbols, where it is understood that any reference to s_j is for j modulo m. For any integer n, we say that $\langle s_i \rangle$ has *property* P(n) if there is an integer i_n for which $s_{i_n} \neq s_{i_n+n}$.

Let $M = \langle Q,\{1\},\delta,\lambda,Y \rangle$ be a strong autonomous machine with m states, where $q_i 1 = q_{i+1}$, and let $\lambda(M)$ be $\langle \lambda(q_0,1),\ldots,\lambda(q_{m-1},1) \rangle$.

Theorem 5.2

A strong autonomous machine M is reduced if and only if, for all $n = 1,2,\ldots,[\frac{m}{2}]$, $\lambda(M)$ has $P(n)$.

PROOF:

Suppose first that $\lambda(M)$ has $P(n)$ for each $1 \leq n \leq [\frac{m}{2}]$, but that states q_i and q_{i+k} are equivalent. Then for any $x \in \{1\}^+$, $\lambda(q_i,x) = \lambda(q_{i+k},x)$, so that for $r = \min\{k,m-k\}$, $1 \leq r \leq [\frac{m}{2}]$ and for every state q_j, $\lambda(q_j,1) = \lambda(q_{j+r},1)$, which would indicate that $\lambda(M)$ does not have $P(r)$, a contradiction.

On the other hand, suppose that M is reduced, and choose $1 \leq n \leq [\frac{m}{2}]$. Since q_0 and q_n are not equivalent, there is some input sequence x, $\ell(x) \geq 1$, such that $\lambda(q_0,x) \neq \lambda(q_n,x)$. Thus if $r = \ell(x)$, then $\lambda(q_{r-1},1) \neq \lambda(q_{n+r-1},1)$, so that $\lambda(M)$ has $P(n)$.

Of course, Theorem 5.2 just says that $\lambda(M)$ has no proper subperiods. If $\langle s_i \rangle$ has $P(n)$ for each $1 \leq n \leq [\frac{m}{2}]$ we say that it has *property* P; conversely, if $\langle s_i \rangle$ does not have $P(n)$ for some n we say that P *fails* (*for* n).

Lemma 5.2

Let $\langle s_i \rangle = \langle s_0,s_1,\ldots,s_{m-1} \rangle$.

a) If P fails for n it fails for the g.c.d. (n,m).

b) If P fails for n_1 and n_2 it fails for (n_1,n_2).

PROOF:

a) Let $g = (n,m)$. It will be sufficient to show that $s_0 = s_g$. Since, by hypothesis, $s_0 = s_n = s_{2n} = \ldots$, we need only show that, for some k, $kn \equiv g \pmod m$. By Lemma 3.3, this congruence has a solution when $(n,m) \mid g$; but $g = (n,m)$, so that $s_0 = s_g$.

b) Let $g = (n_1,n_2)$. In this case we look for k_1 and k_2 which satisfy $n_1k_1+n_2k_2 \equiv g(\text{mod } m)$. Since $(n_1,n_2,m)|g$, Lemma 3.4 guarantees a solution to the congruence, so that, again, $s_0 = s_g$.

Theorem 5.3

Let $\overline{s}_0 \neq s_0$. If $<s_0,\ldots,s_{m-1}>$ does not have P then $<\overline{s}_0,s_1,\ldots,s_{m-1}>$ has P.

PROOF:

Let $n^* = \min\{n|P \text{ fails for } n\}$. We show that for $n \leq n^*$, $<\overline{s}_0,s_1,\ldots,s_{n-1}>$ has P(n).

If $n^* = 1$ then the s_i are identical, and hence the new sequence has P. If $n^* = 2$, then $m \geq 4$. Thus $\overline{s_0} \neq s_2$ so P does not fail for $n = 2$. But since s_0 must have been different from s_1, as otherwise we would have had $n^* = 1$, we still have $s_2 = s_0 \neq s_1$, so the new sequence also has P(1). In general, we know $m \geq 2n^*$. Now suppose for $k < n^*$, $s_{i_k} \neq s_{i_k+k}$. Then since P fails for n^*, $s_{i_k+n^*} = s_{i_k} \neq s_{i_k+k} = s_{i_k+k+n^*}$. Furthermore $i_k+n^* \not\equiv i_k(\text{mod } m)$ and $i_k+k+n^* \not\equiv i_k+k(\text{mod } m)$, since $m \geq 2n^*$. Thus if we change s_0 to $\overline{s_0}$ we can not, for $n < n^*$, cause P(n) to fail for the sequence $<\overline{s_0},s_1,s_2,\ldots,s_{m-1}>$. But, since $\overline{s_0} \neq s_0$, $\overline{s_0} \neq s_{n^*} = s_0$ so the new sequence has P(n*).

Now, let $n_1^* = \min\{n|P(n) \text{ fails for } <s_i>\}$ and $n_2^* = \min\{n|P(n) \text{ fails for } <\overline{s}_0,s_i,\ldots,s_{m-1}>\}$. Then $n_2^* > n_1^*$. But could have started with $<\overline{s}_0,s_i,\ldots,s_{m-1}>$ and changed $\overline{s}_0$ to s_0, giving $<s_i>$. Thus $n_2^* < n_1^*$. This is impossible, so $<\overline{s}_0,s_1,\ldots,s_{m-1}>$ has P.

The next result holds for both Moore and Mealy machines.

Lemma 5.3

Let M be a machine and suppose that in the labelled digraph consisting of D(M) together with state and output labels,

there is a spanning cycle whose output sequence has property P. Then M can be isomorphically realized with feedback encoding by a machine M' with a repeated symbol distinguishing sequence.

PROOF:

We choose M' to have the same labelled digraph as M. Let the states, in their order along the cycle, be $q_0,q_1,\ldots,q_{m-1}$. We will define an h map such that at such state q_i, one input x_i for which $q_ix_i = q_{i+1}$ is coded as $h_{q_i}(x_i) = 1' \in I'$, and assign the other values of h arbitrarily, preserving the type I property. Then in M' the input symbol 1' will define a strong autonomous submachine M'' of M', which, by hypothesis will be reduced. But it is known [27] that a reduced autonomous machine has a (repeated symbol) distinguishing sequence, y. Since M'' has the same state-set as M', y is a repeated symbol distinguishing sequence for M'.

A *2-factor* of a digraph is a spanning subdigraph in which each point u has id(u) = od(u) = 1; a 2-factor is a union of directed cycles. Suppose that a machine M has a 2-factor $Z = Z_1 \cup \ldots \cup Z_t$ consisting of t directed cycles. As in Lemma 5.3, we can define a machine M' which realizes M isomorphically with feedback encoding, such that Z is the subdigraph induced by a single input symbol, x'. We now need only make sure that the autonomous submachine defined by x' is reduced. This can be done, in the worst case, by adding t new output symbols, $w_1,\ldots,w_t$. By Theorem 5.3, if we change one output label in Z_i to w_i, Z_i will be reduced. Furthermore, since $w_i \neq w_j$, no state in Z_i can be equivalent to a state in w_j. Of course, in general we could expect to need fewer than t new output symbols. In the next lemma

we summarize the known conditions for a digraph to have a 2-factor: if D(M) satisfies any of these conditions then M can be isomorphically realized with feedback encoding by a machine with a repeated symbol distinguishing sequence.

Lemma 5.4

a) A digraph D has a 2-factor if and only if for each set S of points, $|S| \leq |\gamma S|$ [2].

b) If a strong digraph D has p points and, for each point v, $id(v)+od(v) \geq p$, then D has a spanning cycle [9].

Of course, the fewer the number of cycles in a 2-factor, the fewer the number of output symbols which will have to be changed. On the other hand, the more cycles in the 2-factor, the shorter will be the length of the distinguishing sequence. This trade-off is expressed in the next theorem: by $[r > 0]$ in the theorem we mean the logical variable which takes the value 1 if $r > 0$ and 0 otherwise.

Theorem 5.4

Let M be a machine with p states and suppose that D(M) has a 2-factor which contains t cycles, of which r are 1-cycles (loops). Then M can be isomorphically realized with feedback encoding by a machine which has a distinguishing sequence. Furthermore, the length L of the distinguishing sequence and the number N of output symbols which must be changed satisfy

$$L \leq p-2t+r+1$$

$$N \leq t-[r > 0]$$

$$L+N \leq p.$$

PROOF:

We have already indicated how the existence of a 2-factor implies the existence of a realizing machine which has a distinguishing sequence. In the worst case, we need to change one output symbol on each of the t-r cycles of length greater than one, both to reduce the cycle (see Theorem 5.3) and to make the cycles inequivalent. In the worst case this requires adding t-r new output symbols, although this can undoubtedly be reduced in practice. Each of the r loops is already reduced, so we need add or change at most r-1 output symbols to make them inequivalent. This gives $N \leq (t-r)+(r-1) = t-1$ if $r > 0$ and $N \leq t$ if $r = 0$; hence $N \leq t-[r > 0]$. The length L of the distinguishing sequence is at most the length of a distinguishing sequence for the largest cycle, which is one less than the length of the cycle ([27], [30]). For a given t and r the longest cycle is attained when all but one of the t-r cycles of length greater than 1 are 2-cycles, using 2(t-r-1) of p-r states. The remaining cycle then has length $p-(2(t-r)-2)-r = p-2t+r+2$, so that $L \leq p-2t+r+1$. Now, if $r > 1$, $L+N \leq t-1+p-2t+r+1 = p-t+r$, which takes its maximum when all cycles are 1-cycles, so that $t = r$. If $r = 0$, then $L+N \leq t+p-2t+1 = p-t+1$, which takes a maximum when $t = 1$, and the 2-factor is a spanning cycle. Thus, $L+N \leq p$.

The bound $L \leq p-2t+r+1 \leq p$ compares quite favorably with the bound $L \leq (p-1)p^p$ given in [10], although it is not known if the latter is a best possible upper bound.

Notice that while the theorem describes a sufficient procedure, it is certainly not necessary. The machine in Example 5.4 has a spanning cycle, but we were able to find a shorter distinguishing sequence by inspection.

The simplest way for a machine to satisfy the conditions of the theorem is for each of its states to have the same indegree; such a machine is called *homogeneous* in [29], where it is shown that any machine whose reduced machine is strong is behaviorally equivalent to a homogeneous machine. Theorem 5.4 could then be applied to the homogeneous machine. Of course, the homogeneous machine has a much larger state set, but this disadvantage may be offset, as Miller and Winograd [29] note:

> McNaughton and Booth [26], however, found that in the 2-input case a particularly uniform circuit structure (for the homogeneous machine) resulted for the state to state circuitry ... the uniform structure may be quite advantageous in practice, and ... can be readily seen to extend to the p-input case.

On the other hand, even if the condition of Lemma 5.4a does not hold, we can use the techniques outlined above to produce a realizing machine with a distinguishing sequence.

Note first that any machine can be isomorphically realized by a machine with a repeated symbol distinguishing sequence, by choosing an autonomous submachine and adding output symbols so as to make it reduced [28]. With feedback encoding a similar technique applies, but there is more freedom in choosing the substructure to reduce, and consequently less output augmentation may be necessary.

Any autonomous machine, whether or not it is a union of directed cycles, is a *functional digraph*: a digraph in which each point has out-degree exactly 1. Given any functional subdigraph $\overline{D}$ of D(M), we can, with feedback encoding, make $\overline{D}$ the digraph of an autonomous submachine $\overline{M}$, and then add output symbols to make $\overline{M}$ reduced. Thus, for a machine M, we would choose a functional subdigraph $\overline{D}$ which was as "close to" being reduced as possible, and then add output symbols so as to make it reduced. The technique in doing this is first to make each cycle of $\overline{D}$ reduced, and to make the cycles inequivalent, as we would if $\overline{D}$ were a 2-factor. We then continue, making each component of $\overline{D}$ reduced. Two states can be equivalent only if they belong to the same component and there is a homomorphism which identifies them; homomorphisms of autonomous machines (as we noted in Chapter II the notions of SP and admissible homomorphisms coincide for autonomous machines) have been studied in detail in [34]. We can therefore state:

Remark

Any machine can be isomorphically realized with feedback encoding (in the sense of (5.1) or (5.3)) by a machine with a repeated symbol distinguishing sequence.

Example 5.5

The machine M of Figure 5.5 is reduced, has no distinguishing sequence, and has no 2-factor, since

$$|\gamma\{q_1,q_2,q_3,q_4,q_5\}| = |\{q_2,q_3,q_4,q_5\}| = 4 < |\{q_1,q_2,q_3,q_4,q_5\}|.$$

Suppose that we choose the functional subdigraph $\overline{D}$ of Figure 5.6; the outputs, but not the inputs, associated with the arcs are shown.

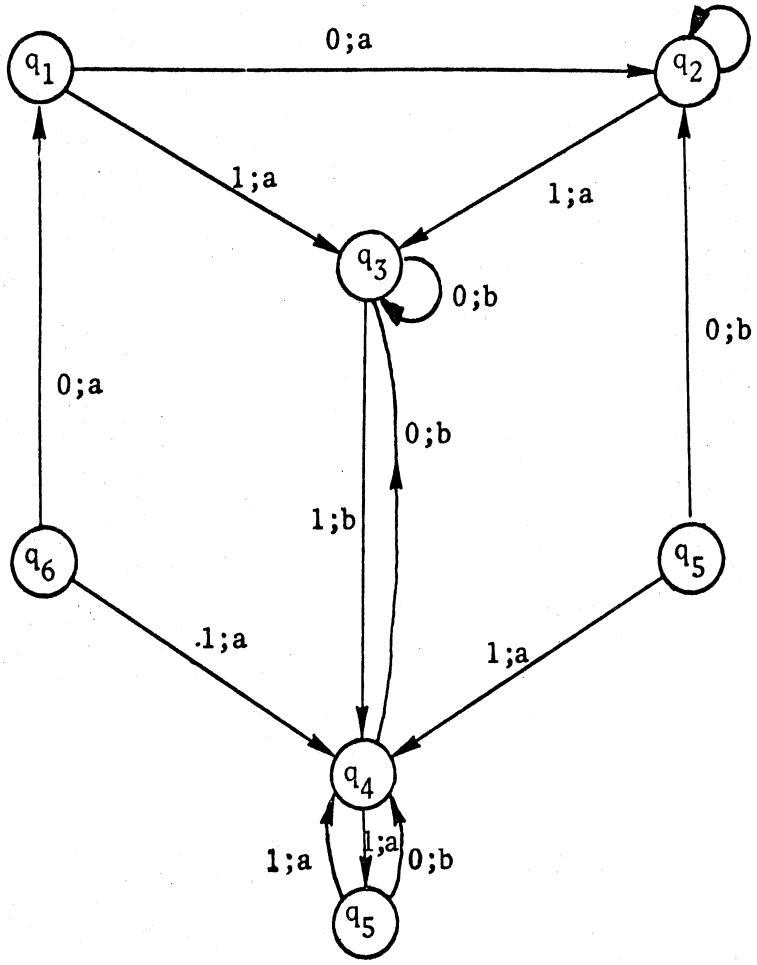
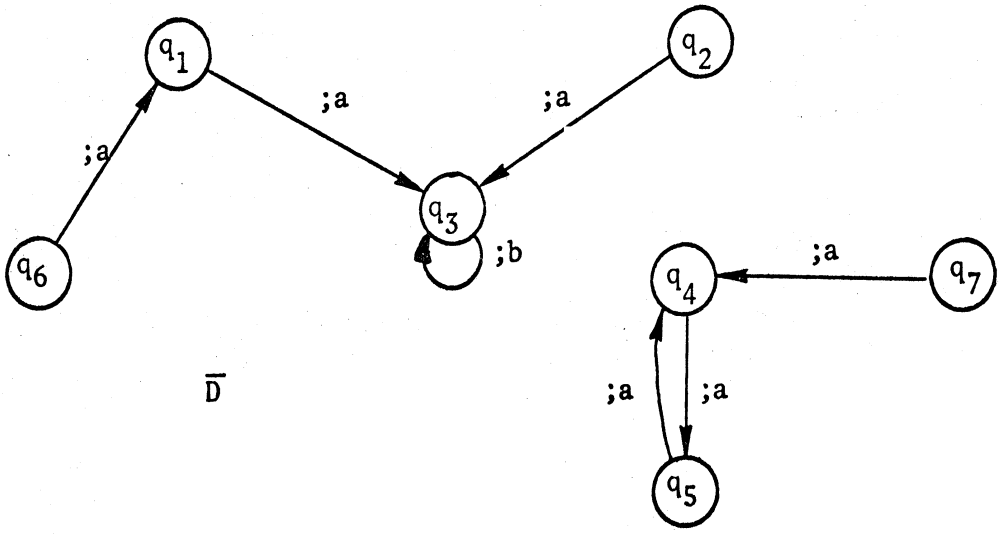
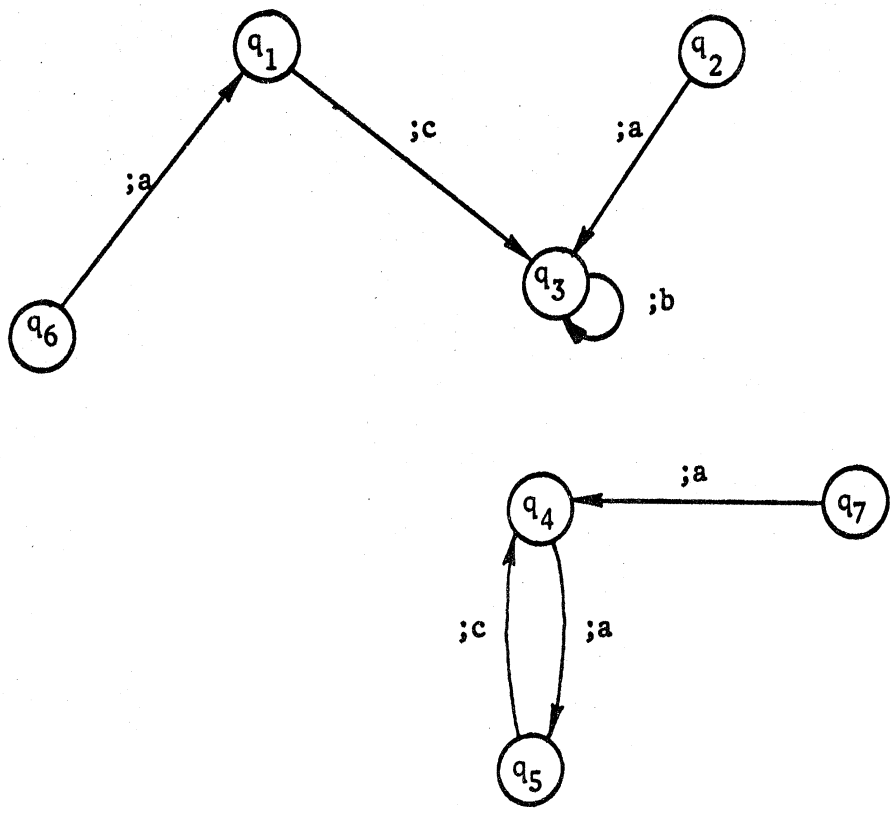


Figure 5.5. A machine without a 2-factor.



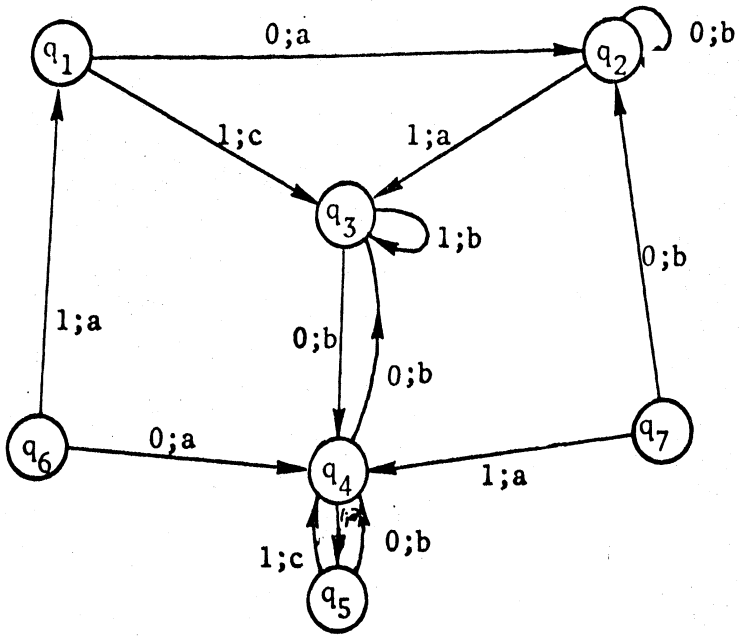
$\overline{D}$

If we change the output label on the arc q_5q_4 to a new symbol, c, then the lower component in Figure 5.6 will be reduced. To reduce the upper component, we need only change the symbol on one of the arcs q_1q_3 or q_2q_3; since we already know that c must be decoded to an a, we can relabel q_1q_3 with c. The resulting reduced autonomous machine $\overline{D}'$ is shown in Figure 5.7, and the machine M' which realizes M isomorphically with feedback encoding is given in Figure 5.8.



$\overline{D}$

Figure 5.7. A reduced autonomous machine.



M'

Figure 5.8. A machine with a distinguishing sequence.

In fact, x = 111 is a distinguishing sequence, for

$$\beta_{q_1}(x) = c\ b\ b$$
$$\beta_{q_2}(x) = a\ b\ b$$
$$\beta_{q_3}(x) = b\ b\ b$$
$$\beta_{q_4}(x) = a\ c\ a$$
$$\beta_{q_5}(x) = c\ a\ c$$
$$\beta_{q_6}(x) = a\ c\ b$$
$$\beta_{q_7}(x) = a\ a\ c.$$

A reduced machine will fail to have a distinguishing sequence only if two states converge under some input. Using classical techniques, as we noted above, once a convergence is found to interfere with the existence of a distinguishing sequence, state-splitting and/or augmentation of outputs must be employed. With feedback encoding, however, it is

often possible to eliminate the convergence without adding states or output symbols. We first state a theorem indicating when this can be done for a 2-input (Moore) machine.

Theorem 5.5

Let M be a Moore machine with two inputs x_1 and x_2 and exactly one convergence: $(q_1^1, q_2^1)x_1 = q_1^2$. Then, if there is no state q such that $qx_2 = q_1^2$, M can be isomorphically realized with feedback encoding by a convergence-free machine with the same output function.

PROOF:

Note first that since M has only one convergence, $q_1^1 x_2 \neq q_2^1 x_2$. Let $q_2^1 x_2 = q_2^2$ and $q_1^1 x_2 = q_0^2$ (see Figure 5.9).

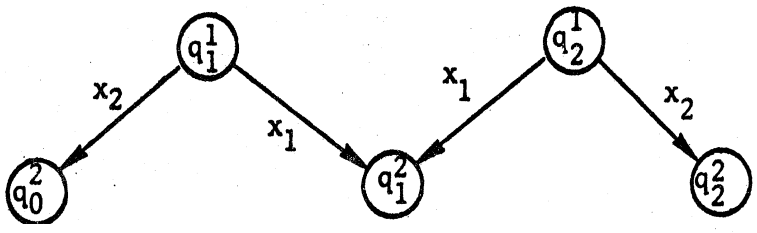


Figure 5.9. A convergence.

We can recode inputs at q_1^1, eliminating the convergence, unless there is a state q_0^1 such that $q_0^1 x_1 = q_0^2$; similarly, we can recode at q_2^1 unless there is a q_3^1 such that $q_3^1 x_1 = q_2^2$. Suppose that we can recode neither at q_1^1 nor at q_2^1 (see Figure 5.10).

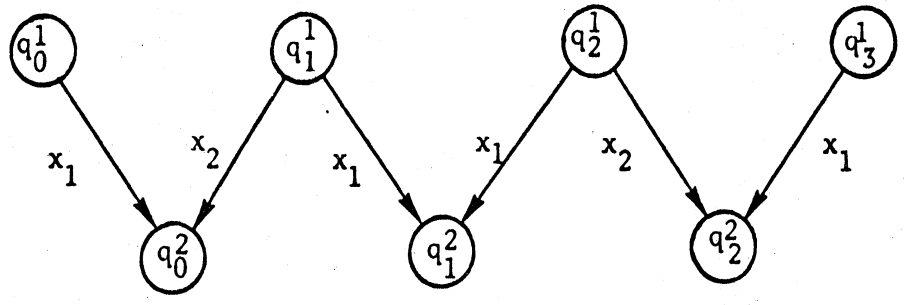


Figure 5.10. A step in the proof of Theorem 5.5.

If, for example, there is no state q_4^1 such that $q_4^1 x_1 = q_3^1 x_2$ then we can recode the inputs at q_3^1 without causing a new convergence; this, in turn, permits us to recode the inputs at q_2^1, and hence eliminate the convergence.

Continuing in this manner gives rise to sequences of states

$$\ldots, q_{-1}^1, q_0^1, q_1^1, q_2^1, q_3^1, q_4^1, \ldots$$
$$\ldots, q_{-1}^2, q_0^2, q_1^2, q_2^2, q_3^2, \ldots$$

such that

if $j > 1$

$$q_j^1 x_1 = q_{j-1}^2$$
$$q_j^1 x_2 = q_j^2$$

and if $j \leq 1$

$$q_j^1 x_1 = q_j^2$$
$$q_j^1 x_2 = q_{j-1}^2.$$

Since M is finite the process of extending these sequences must have repetitions. Suppose the first repetition is that two of the q_j^2 coincide. If we are at the stage of choosing q_j^2 and find that it is the same as some previously chosen q_k^2 then q_k^2 must be an x_2 image of two distinct states, by virtue of the way q_j^2 is chosen, unless $k = 1$. If $k \neq 1$ this implies that M has two convergences, which is a contradiction. If $k = 1$ then assume without loss of generality that

$j > 0$ and continue the process at the "other end" of the sequence. It then becomes impossible for the process to fail for lack of a new q^1_{-n}, for this would have to imply that M has a second convergence.

The process therefore must stop because of an inability to choose a new q^1_{j+1} (or q^1_{j-1}). But then we can progressively relabel inputs at $q^1_j, q^1_{j-1}, \ldots$ (or at $q^1_j, q^1_{j+1}, \ldots$) until we finally relabel at q^1_2 (or q^1_1), eliminating the convergence. At no time have we changed any output label: this proves the theorem.

The technique used in Theorem 5.5 is applicable to Moore or Mealy machines. More important, it may be used successfully with machines which have more convergences than specified in the theorem. One straightforward generalization is given by the following corollary.

Corollary

If M has exactly n convergences $(q^i_1, q^i_2)x^i_1 = q^i$, where the x^i_1, $i = 1, \ldots, n$ are distinct, and if there are n additional inputs x^i_2 such that there is no state q for which $qx^i_2 = q^i$, then the convergences can be eliminated.

PROOF:

For each i, apply the theorem to that submachine of M defined by the two inputs x^i_1 and x^i_2.

The corollary, of course, was phrased to insure that the n applications of the theorem would not conflict. Certainly, we can expect that the same techniques will apply to many machines which do not meet the strict condition of the corollary, by breaking up the convergences one at a time. Unfortunately, the extent to which the technique can be reapplied depends both on behavioral and structural properties of the given machine. While useful as a heuristic, this

approach to eliminating convergences cannot easily be expressed in algorithmic form, with clearcut rules for choosing, given at some stage a convergence $(q_1,q_2,\ldots,q_n)x_1 = q$, which subconvergence $(q_i,q_j)x_1 = q$ to break up, and which input $x_2 \neq x_1$ to use.

In contrast to this essentially local attack on the problem of eliminating convergences, we will develop a global procedure for reducing, not the number of convergences, but rather the number of merges. With Moore machines as we have defined them the concepts of "merge" and "converge" are, of course, identical, but this, as has been pointed out, is not always a useful identification to make. Thus, the global technique we propose may often be inefficient, as it will reduce many merges which are not convergences. For machines which have a large number of convergences, however, we can expect the technique to substantially decrease the number of additional states or outputs which are required for a diagnosable realization. Unfortunately, since the technique deals with merges and not convergences, exact results on the number of additional states or output symbols which will be saved are not available. In fact, it is possible to devise examples where the procedure, while reducing the number of merges, increases the number of convergences. This will become clearer as we get into the actual mechanics of the procedure.

Let D be a digraph, and number the points $u_1,\ldots,u_p$ so that $id(u_1) \leq id(u_2) \leq \ldots \leq id(u_p)$. The *indegree sequence of* D is the sequence $\langle id(u_1), id(u_2),\ldots,id(u_p)\rangle$. Suppose that D is the digraph of a Mealy machine, D = D(M), and let M have t inputs. Let $\hat{\Xi}(D)$ be the quantity $\sum_{i=1}^{p} \max\{0,id(u_i)-t\}$. For any arc uv, let $\xi(uv)$ be the input label on the arc. We will say that there are n merges at a state u if there are n+1 arcs $v_1u,v_2u,\ldots,v_{n+1}u$ such that

$\xi(v_1u) = \xi(v_2u) = \ldots = \xi(v_{n+1}u)$; note that if all these arcs had the same output label it would be necessary to add n new output symbols to eliminate all the convergences. Let $\Xi(u)$ be the total number of merges at u, and $\Xi(M)$ be the total number of merges in M.

Lemma 5.5

For any machine M with t inputs, $\Xi(M) \geq \hat{\Xi}(D(M))$.

PROOF:

We will show that for each state u, $\Xi(u) \geq \max\{0, id(u)-t\}$ Clearly, the equation holds for each state u with $id(u) \leq t$. If the number of different input labels on arcs leading to state u is $m \leq t$, where $id(u)>t$, then $\Xi(u) = id(u)-m$. Thus, if $id(u) > t$ then $\Xi(u) = id(u)-m \geq id(u)-t$.

Theorem 5.6

Any machine M can be isomorphically realized with feedback encoding by a machine M' with $\Xi(M') = \hat{\Xi}(D(M))$.

To prove this theorem we need to investigate those properties of the assignment function ξ which will guarantee that $\Xi(M) = \hat{\Xi}(D(M))$. To this end, we introduce some additional notions from graph theory.

A bigraph [12, p. 17] is a graph G whose point-set V can be partitioned into two sets V_1 and V_2 such that all lines of G join points of V_1 with points of V_2. A *line-coloring* of a graph is an assignment of colors to the lines in such a way that any two lines which are incident with the same point receive different colors. The smallest n such that G can be line-colored with n colors is the *line-chromatic number* $\chi'(G)$. Clearly, the line-chromatic number of a graph G is not less than $\Delta(G)$, the maximum of the degrees of the points of G. For a bigraph, a stronger statement can be made.

Lemma 5.6 ([23, p. 171])

For any bigraph G, $\chi'(G) = \Delta(G)$.

Now, let M be a machine with states $\{v_1,\ldots,v_p\}$, and form a bigraph G with points $\{v_{1i},v_{2i}|i=1,\ldots,p\}$ where v_{1i} is adjacent to v_{2j} if and only if $v_iv_j \in D(M)$; these are the only lines in G. Note how the degrees of the points of G are related to these of the points of D(M): $\deg(v_{1i}) = od(v_i)$ and $\deg(v_{2i}) = id(v_i)$. If we "color" each line $v_{1i}v_{2j}$ of G with the corresponding input symbol $\xi(v_iv_j)$, then if two lines are incident with the same point v_{1i} they are colored differently, since M is a deterministic machine. If lines x and y with $\xi(x) = \xi(y)$ are incident to point v_{2i} there is a merge. As Lemma 5.5 shows, there must be merges at states v_i with $id(v_i) > t$, the number of inputs. To prove Theorem 5.6 we will show how to assign inputs to the arcs of D(M) in such a way that the resulting machine is complete and deterministic, hence realizing M with feedback encoding, and the only merges occur at v_i with $id(v_i) > t$. If the line-chromatic number of the associated bigraph G is $\chi'(G) \geq t$, we will color the lines of G from a set of colors $\{\beta_1,\ldots,\beta_{\Delta(G)}\}$ in such a way that only colors from $\{\beta_1,\ldots,\beta_t\}$ are used to color lines incident with points v_{2i} with $\deg(v_{2i}) \leq t$. If we translate each color β_i in $\{\beta_1,\ldots,\beta_t\}$ to the input symbol x_i and assign input symbols from $\{x_1,\ldots,x_t\}$ to lines colored from $\{\beta_{t+1},\ldots,\beta_\Delta\}$ in such a way as to give a complete deterministic machine M', then there will be no merges at states v_i with $id(v_i) \leq t$. Furthermore, if $id(v_j) > t$ then since in the coloring of G each color $\beta_1,\ldots,\beta_t$ appears once on a line incident with v_{2j}, the number of merges at v_j will be exactly $id(v_j)-t$. Thus we will have $\Xi(M') = \hat{\Xi}(D(M))$. To prove Theorem 5.6, it is then necessary only to demonstrate that a coloring of the prescribed type always exists.

Theorem 5.7

Let G be a bigraph in which $\max\{\deg u \mid u \in V_1\} = n = d_0$, and suppose that the degrees greater than n which are realized in V_2 are $n < d_1 < d_2 < \ldots < d_r = \Delta$. Then there is a line coloring of G from $\{\beta_1,\ldots,\beta_\Delta\}$ such that all lines colored

$$\beta_{d_i+1},\ldots,\beta_{d_{i+1}}$$

are incident to points of degree greater than d_i, for $i = 0,\ldots,r-1$.

To prove the theorem we first develop a sequence of lemmas.

Lemma 5.7

Let G be a bigraph such that all points of maximum degree are in V_2. Then G can be line colored from $\{\beta_1,\ldots,\beta_\Delta\}$ such that the only lines colored β_Δ are incident with points of maximum degree.

)F:

We suppose the result to be true for bigraphs with q-1 lines. Let G have q lines and let all points of maximum degree be in V_2; let $x = uv$ be incident with one such point v. If $\Delta(G-x) < \Delta(G) = \Delta$ then v was the only point of maximum degree in G. So any line-coloring of G-x from $\{\beta_1,\ldots,\beta_{\Delta-1}\}$ extends to a line-coloring of G in which only x is colored β_Δ.

If $\Delta(G-x) = \Delta(G) = \Delta$ then we can color G-x with Δ colors so that all lines colored β_Δ are incident with points of maximum degree; in particular, no line colored β_Δ is incident with v. If there is no β_Δ-line at u, then x can be colored β_Δ in G. Otherwise, there is a β_Δ-line uv_1 (where $\deg v_1 = \Delta$). Since $\deg u < \Delta$ there is some color α which does not appear at u. Clearly however, there is a line v_1u_1

colored α. Thus we get a sequence $<u = u_0, v_1, u_1, v_2, \ldots, >$ such that each v_i has maximum degree, each $u_j v_{j+1}$ is colored β_Δ and each $v_j u_j$ is colored α. At each step of this process we are choosing a new point. If we have just chosen v_j then u_j cannot be a previously chosen u_k: $u_j \neq u_0$ as α does not appear at u_0 and u_j cannot be some other previously chosen u_k for otherwise there would be two lines colored α incident to u_k. Similarly, if u_j has just been chosen, v_{j+1} cannot be a previously chosen v_k or otherwise there would be two lines colored β_Δ at v_k. Of course, since G-x is a bigraph, no v_j can be equal to any u_k. Then, since G-x is finite, this process must terminate when we are unable to choose a new u_j or a new v_{j+1}. Since deg $v_j = \Delta$, the process cannot stop with a v_j, so it must stop at some u_j, at which there is no β_Δ-line. We have thus defined a component of the subgraph $G\text{-}x|_{\alpha,\beta_\Delta}$, and can interchange the colors α and β_Δ in this subgraph, preserving the validity of the coloring. But now β_Δ does not appear at u, so x can be colored with β_Δ.

Lemma 5.8

In a bigraph G, suppose $\max\{\deg u | u \varepsilon V_1\} = n$ and that there are at least two degrees greater than or equal to n realized by points of V_2, the two largest being $n \leq \Delta' < \Delta$. Then there is a line-coloring of G from $\{\beta_1, \ldots, \beta_\Delta\}$ such that all lines colored $\beta_{\Delta'+1}, \ldots, \beta_\Delta$ are incident with points of maximum degree.

PROOF:

Clearly the result is true whenever n = 1. Suppose it to be true for n-1, and suppose that in G, $\max\{\deg u | u \varepsilon V_1\} = n$. Note that if $\Delta - \Delta' = 1$ then the result holds by Lemma 5.7.

Suppose now that the result is true for $\Delta - \Delta' = t-1$, and that in G, $\Delta - \Delta' = t$, where $v_1, \ldots, v_r$ are the points of degree Δ. We remove

an independent set X of r lines, one adjacent to each of $v_1,\ldots,v_r$, to get $G':\Delta(G') = \Delta(G)-1$. If, in G', $\max\{\deg u|u\varepsilon V_1\} = n$ then G' can be colored with $\Delta(G)-1$ colors in the prescribed manner: the only lines colored $\beta_{\Delta'+1},\ldots,\beta_{\Delta-1}$ are incident with the v_r. Now, the lines of X can be colored β_Δ.

Otherwise, $\max\{\deg u|u\varepsilon V_1\} = n-1$. By induction on n, a line-coloring of G' with the desired properties can be achieved, and this coloring uses only $\Delta-1$ colors, as above. Again, the lines of X can be colored β_Δ.

Lemma 5.9

Let G be a bigraph in which $\max\{\deg u|u\varepsilon V_1\} = n$, $\Delta(G) = \Delta > n$, and suppose there are no points of degree $n+1,\ldots,\Delta-1$. Then G has a line-coloring from $\{\beta_1,\ldots,\beta_\Delta\}$ such that all lines colored $\beta_{n+1},\ldots,\beta_\Delta$ are incident with points of maximum degree.

PROOF:

By Lemma 5.7 we know the result is true, for any n, if $\Delta = n+1$. Also, the result is trivially true whenever n = 1. Suppose that the result holds when $\max\{\deg u|u\varepsilon V_1\} = n-1$, and let G have $\max\{\deg u|u\varepsilon V_1\} = n$. Since we know the result holds for $\Delta = n+1$ suppose that it holds for $\Delta = n+k-1$, and let G have $\Delta(G) = \Delta = n+k$. Suppose the points of degree Δ are $v_1,\ldots,v_r$. Remove an independent set X of lines which covers $\{v_1,\ldots,v_r\}$, and let G-X = G'. If in G', $\max\{\deg u|u\varepsilon V_1\} = n$ then the resulting graph satisfies the conditions of the theorem with $\Delta = n+k-1$, so there is a line-coloring where all lines colored $\beta_{n+1},\ldots,\beta_{n+k-1}$ are incident with the v_i. Then the lines in X can be colored β_{n+k} and the result holds.

Otherwise, in G', $\max\{\deg u | u\varepsilon V_1\} = n-1$. Then the result holds for G' by induction unless there were points of degree n in V_2. If so, G' satisfies the conditions of Lemma 5.8 with $\Delta'(G') = n$, $\Delta(G') = \Delta-1$, and so there is a line-coloring of G' from $\{\beta_0,\ldots,\beta_{\Delta-1}\}$ such that all lines colored $\beta_{\Delta'+1} = \beta_{n+1},\ldots,\beta_{\Delta-1}$ are incident with the v_i. By coloring the lines of X with β_Δ the result holds. If V_2 had no points of degree n, then by the inductive hypothesis on n, in the line-coloring of G' all lines colored $\beta_n,\beta_{n+1},\ldots,\beta_{\Delta-1}$ are incident with the v_i. We can again color the lines of X with β_Δ, proving the theorem.

PROOF OF THEOREM 5.7:

The result is trivial for n = 1. Also, by Lemma 5.9 it holds whenever r = 1, so we can assume it true for bigraphs in which $\max\{\deg u | u\varepsilon V_1\} \leq n-1$ and also for bigraphs in which $\max\{\deg u | u\varepsilon V_1\} = n$ and r-1 degrees greater than n are realized in V_2. Let G have $\max\{\deg u | u\varepsilon V_1\} = n$ and let degrees $n < d_1 <\ldots< d_r = \Delta$ be realized in V_2. We first prove the result in the case $d_r-d_{r-1} = 1$. Suppose we remove an independent set X covering the points of degree d_r, to get a graph G'. If the maximum degree of the points in V_1 is reduced to n-1, then G' has a line-coloring of the desired type; in particular, all lines colored $\beta_{d_{r-2}},\ldots,\beta_{d_{r-1}}$ are incident with points of degree d_{r-1} in G', those being the points of degree d_{r-1} or d_r in G. Then by coloring the lines of X with β_Δ, the desired coloring results. If in G' the maximum degree of the points in V_1 is n, then the inductive hypothesis on r guarantees the desired coloring for G', and again we can color the lines of X with β_Δ.

Now, suppose the result holds for $d_r-d_{r-1} = t-1$ and suppose that,

in G, $d_r - d_{r-1} = t$. Let $v_1,\ldots,v_s$ be the points of degree $d_r = \Delta$. We can remove an independent set of lines X which covers $\{v_1,\ldots,v_s\}$, giving a graph G' with maximum degree $\Delta-1$, and next largest degree d_{r-1}; note that $(\Delta-1)-d_{r-1} = t-1$. If, in G', $\max\{\deg u | u\in V_1\} = n$ we get a line coloring of G' from $\{\beta_1,\ldots,\beta_{\Delta-1}\}$ with the desired properties by the inductive hypothesis on t. If not, we get a line-coloring by the inductive hypothesis on n. Either way, we can color the lines of X with β_Δ.

We state, without proof as it bears no direct relationship to our study, the following generalization of Theorem 5.7.

Theorem 5.8

Let G be a bigraph which has points of degree $d_1 < d_2 <\ldots< d_r = \Delta$. Then G has a line-coloring from $\{\beta_1,\ldots,\beta_\Delta\}$ such that all lines colored

$$\beta_{d_i+1},\ldots,\beta_{d_{i+1}}$$

are incident to points of degree at least d_{i+1}.

PROOF:

See [6].

Example 5.6

Consider the machine M in Figure 5.11. There are convergences at states q_3, q_4 and q_5, and M certainly does not have a distinguishing sequence. The associated bigraph G has $\Delta(G) = 3$, and so can be line-colored with three colors. Two such colorings are given in Figures 5.12 and 5.13. In Figures 5.14 and 5.15 we show the deterministic incomplete machines with three inputs which the colorings induce.

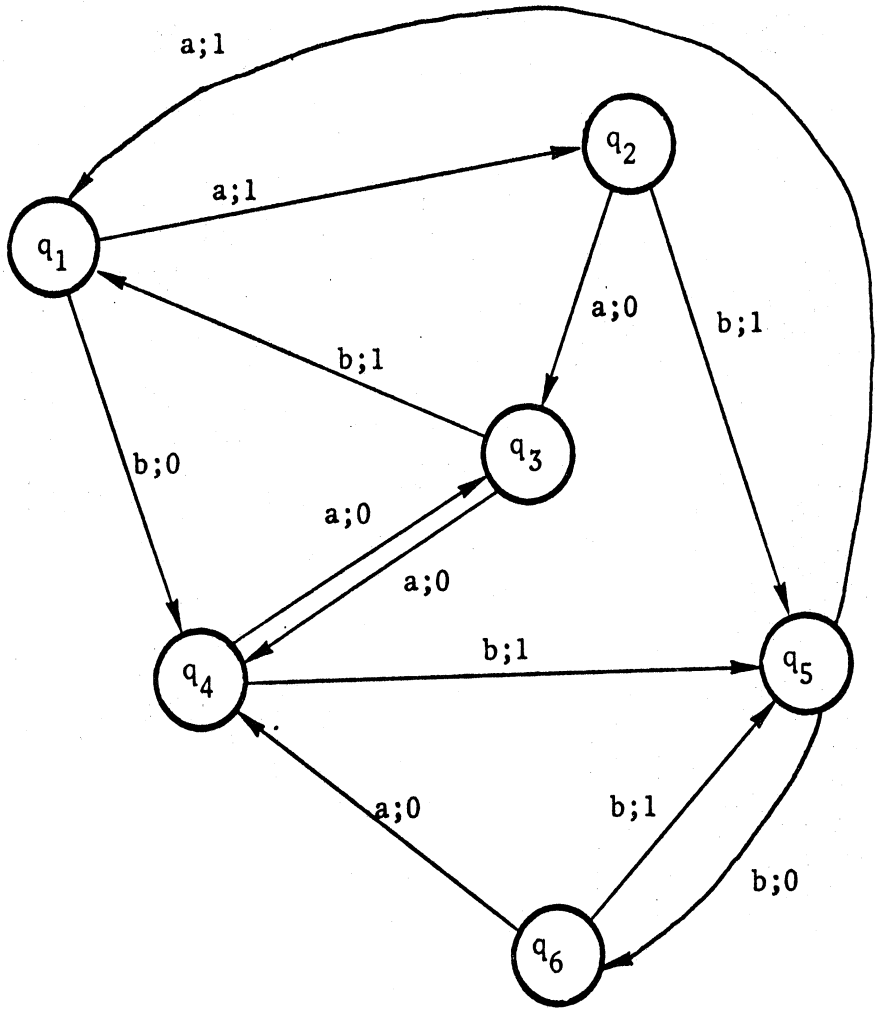


Figure 5.11. A machine with convergences at three states.

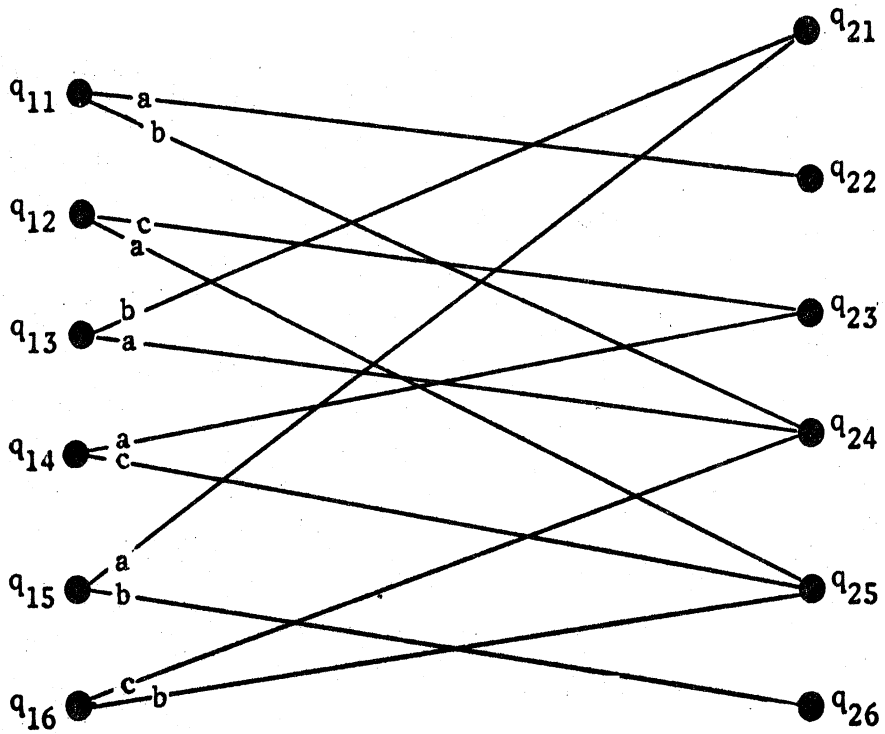


Figure 5.12. A line coloring of a bigraph.

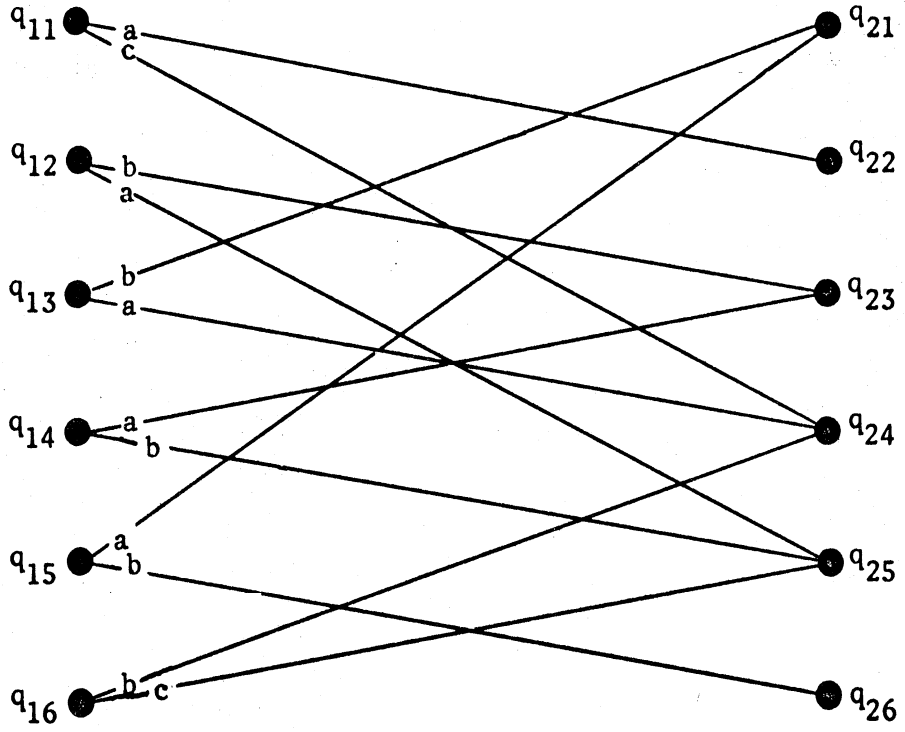


Figure 5.13. A line coloring satisfying the conditions of Theorem 5.7.

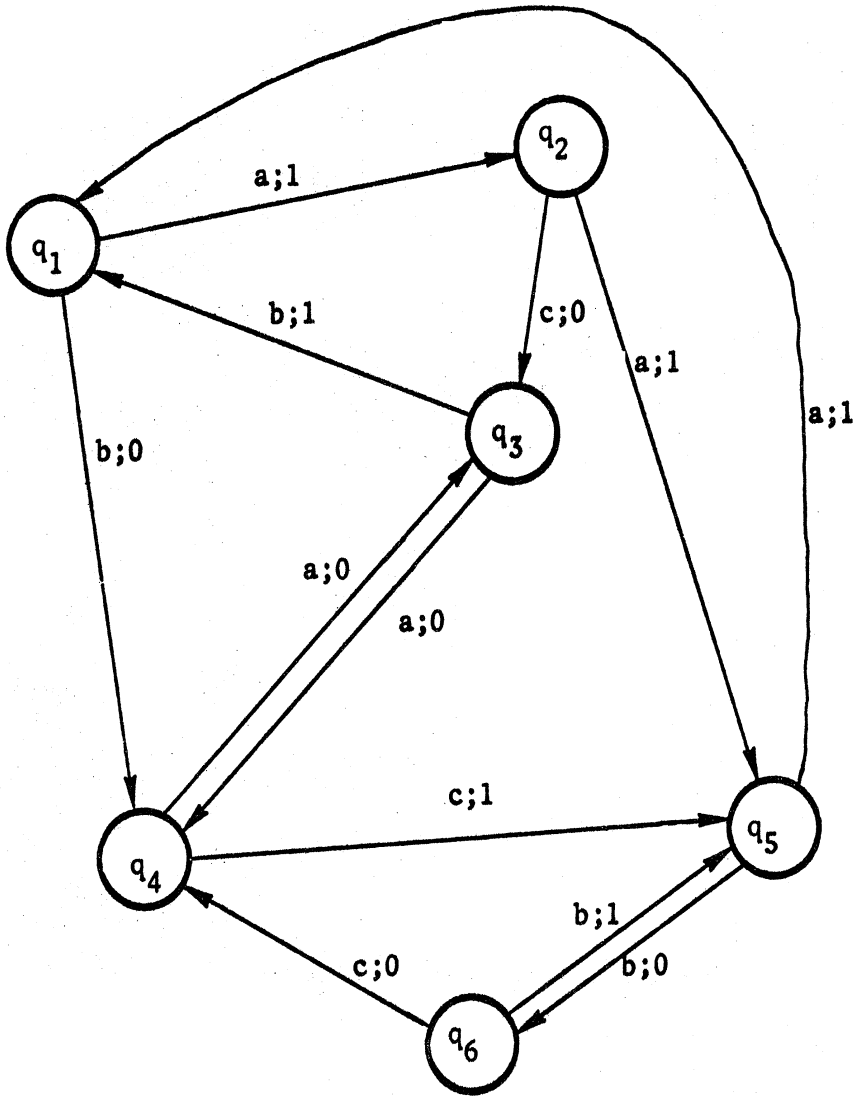


Figure 5.14. An incomplete machine derived from Figure 5.12.

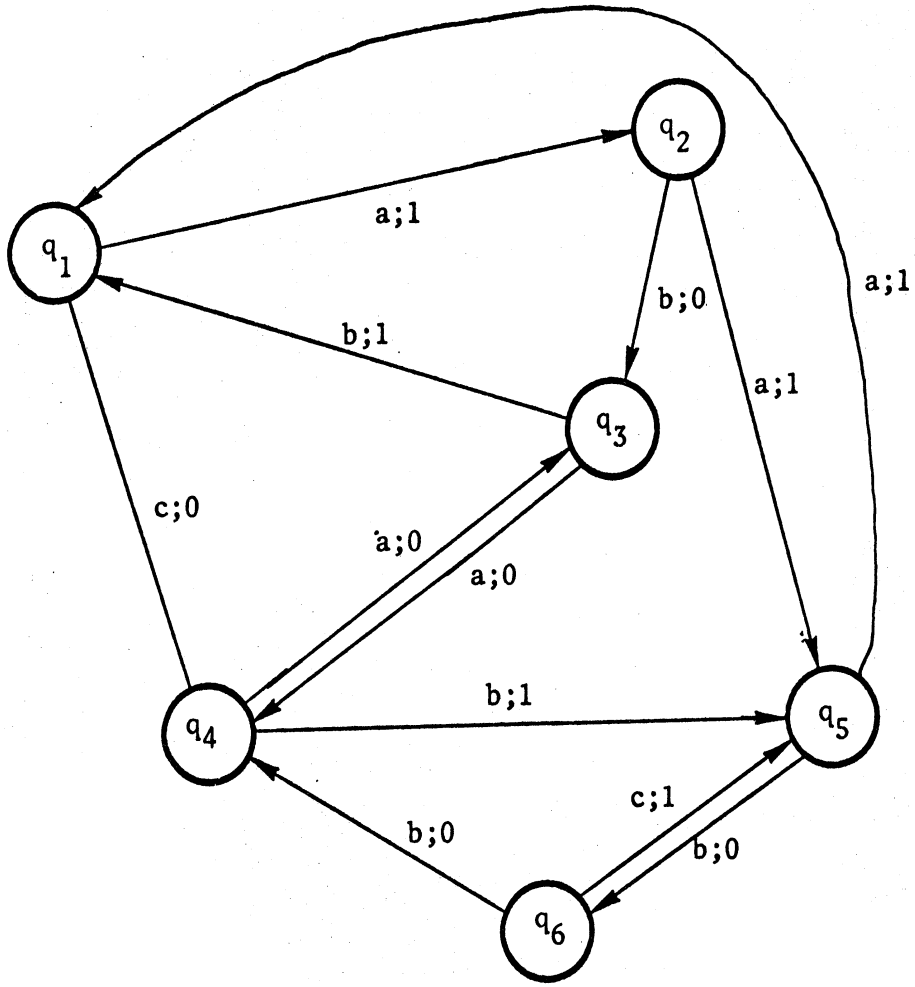


Figure 5.15. An incomplete machine derived from Figure 5.13.

In each machine we would change the labels c to an a or b so as to make the resulting machine both complete and deterministic. For the machine of Figure 5.14 there would still be three convergences. However, as the coloring in Figure 5.13 satisfies the conditions of Theorem 5.7, the only arcs labelled c are incident to one of the states q_4 or q_5 having maximum indegree, when the labels on the machine in Figure 5.15 are changed, the resulting machine, shown in Figure 5.16, has only two convergences.

As we have noted, the procedure which we have outlined reduces merges rather than convergences, and for machines M in which $\hat{\Xi}(D(M))$ is small but nonzero may actually increase the number of convergences. Nevertheless there are two cases in which the procedure can be shown to be of definite advantage. We state these as corollaries to Theorem 5.6.

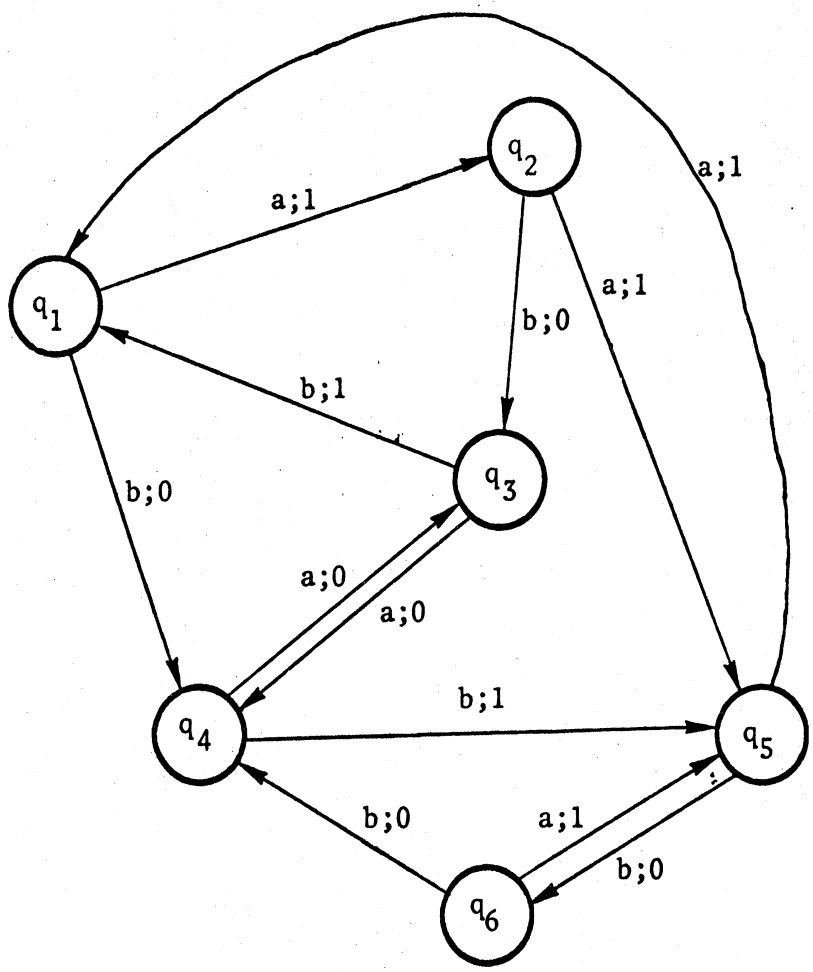


Figure 5.16. A machine M with $\hat{\Xi}(D(M))$ convergences.

Corollary

If M has t inputs then M can be isomorphically realized with feedback encoding by a convergence free machine if and only if every state q has id(q) = t.

Such machines, of course, also satisfy Theorem 5.4; this corollary, therefore, provides an alternate method of attack.

Any machine M can be isomorphically realized with feedback encoding by a machine which has at most $\hat{\Xi}(D(M))$ convergences, since every convergence is a merge. Of special interest is the case when M has more than $\hat{\Xi}(D(M))$ convergences.

Corollary

If M has more than $\hat{\Xi}(D(M))$ convergences then M can be isomorphically realized with feedback encoding by a machine with fewer convergences.

CHAPTER VI

SUMMARY

We have studied both properties and applications of realization with feedback encoding, and, at this point, we should look back to see what we did, and did not, do.

Most of the study of the properties of these realizations was motivated by the classical theory of automata. Thus, for example, admissible homomorphisms play the same role for realizations with feedback encoding that SP homomorphisms play for realizations. There are subtle differences between these two classes of mappings, however. On one hand, as we pointed out in Chapter II, admissible homomorphisms do not exhibit all the lattice properties of SP homomorphisms. Thus, the full power of the Hartmanis-Stearns techniques [17] cannot be applied to admissible homomorphisms. On the other hand, admissible homomorphisms being defined on digraphs rather than on machines, are somewhat easier to manipulate and study. In this regard, an added advantage is that, as mappings between digraphs or graphs, they have interest independent of their applications to machines; Theorem 2.15 is one example of this. Another advantage of admissible maps is the procedure for checking admissibility which is embodied in Theorems 2.7 and 2.13.

Of course, we have actually only scratched the surface in studying admissible homomorphisms. One major question which is still outstanding is, which digraphs have no admissible homomophic images; related to this is the problem of finding those digraphs which have no walkwise images. We could also ask, what properties of digraphs

are preserved by admissible homomorphisms, in the sense that the property "$m(D) \geq k$" is preserved?

There appears to be little more that can be said about the basic properties of realizations with feedback encoding. One problem upon which we did not touch is the meaning of realization with feedback encoding when applied to logical nets. For example, Zeigler [36] shows that for any integer r there is a machine M such that any logical net which isomorphically realizes M has a strong component S which contains a point whose indegree, in S, is greater than r. While we strongly suspect that a similar result would hold for isomorphic realization with feedback encoding, it is not clear how to attack the problem. Zeigler's proof techniques depend heavily on behavioral properties, which of course are blurred by realization with feedback encoding, and so resolution of the problem would probably depend on a better understanding of the relationship between net structure and transition graph structure.

The semigroup of a machine is quite important to the theory of realization. While we studied the S-semigroups in great detail in chapter IV, our motivation was to be able to develop decomposition properties, and we must conclude that this goal is very likely unattainable. For, we showed that zero-free homomorphisms or divisions between S-semigroups are both necessary and sufficient for realizations or divisions with feedback encoding, which makes it difficult to suppose that a finite algebraic structure with similar properties could be found. And certainly, even if decomposition properties could be related to the S-semigroups, there is no real hope of usefully applying these infinite structures to the problems of finite automata theory.

We also discussed two applications of realizations with feedback encoding. In Chapter III we showed that cascade realizations with feedback encoding can be more economical, in terms of the sizes of the state sets of the component machines, than realizations without feedback encoding, and in Chapter V we showed how to apply feedback encoding to the design of realizing machines with distinguishing sequences. Perhaps other applications could have been developed, but these are sufficient to show the value, and limits, of realization with feedback encoding. For, and this cannot be stressed too strongly, there are no clear rules on when to use the techniques we have developed. Even when, as with the distinguishing sequence problem, we can guarantee the applicability of the techniques to any machine, whether or not they are of any value will depend quite strongly on the specific problem. We have shown the potential power of realization with feedback encoding, but in any application it will be just one of a number of tools which can be tried, and will work better in some cases than in others.

BIBLIOGRAPHY

1. Arbib, M.A. *Theories of Abstract Automata*, Prentice-Hall, Englewood Cliffs, N.J. 1969.

2. Berge, C. *The Theory of Graphs*, Methuen, London. 1964.

3. Burks, A.W. and Wang, H. "The Logic of Automata: Parts I and II," *J.A.C.M.*, *4*, pp. 193-297, 1967.

4. Chartrand, G. and Geller, D. "On Uniquely Colorable Planar Graphs," *J. Combinatorial Theory*, *6*, pp. 271-278, 1969.

5. Eggan, L.C. "Transition Graphs and the Star-Height of Regular Events," *Michigan Math. J.*, *10*, pp. 385-397, 1963.

6. Geller, D. "How to Color the Lines of a Bigraph," Technical Report 032960-12-T, Department of Computer and Communication Sciences, The University of Michigan, April 1971.

7. ___, "Minimally Strong Digraphs," *Proc. Edinburgh Math. Soc.*, *17*, pp. 15-22, 1970.

8. ___, "The Square Root of a Digraph," *J. Combinatorial Theory*, *5*, pp. 320-321, 1968.

9. Ghouila-Houri, A. "Une Condition Suffisante d'Existence d'un Circuit Hamiltonien," *C.R. Acad. Sci. Paris*, *251*, pp. 495-497, 1960.

10. Gill, A. *Introduction to the Theory of Finite-State Machines*, McGraw-Hill, New York. 1962.

11. Ginzburg, A. *Algebraic Theory of Automata*, Academic Press, New York. 1968.

12. Harary, F. *Graph Theory*, Addison-Wesley, Reading. 1969.

13. Harary, F.; Hedetniemi, S.T.; and Prins, G. "An Interpolation Theorem for Graphical Homomorphisms," *Port. Math.*, *26*, pp. 453-462, 1967.

14. Harary, F.; Hedetniemi, S.T.; and Robinson, R.W. "Uniquely Colorable Graphs," *J. Combinatorial Theory*, *6*, pp. 264-270, 1969.

15. Harary, F.; Norman, R.Z.; and Cartwright, D. *Structural Models*, John Wiley & Sons, New York. 1965.

16. Harary, F. and Trauth, C.A.,Jr. "Connectedness of Products of Two Directed Graphs," *J. SIAM Applied Math.*, *14*, pp. 250-254, 1966.

17. Hartmanis, J. and Stearns, R.E. *Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, Englewood Cliffs, N.J. 1966.

18. Hedetniemi, S.T. "Homomorphisms of Graphs," Technical Report 03105-42-T, Department of Computer and Communication Sciences, The University of Michigan, December 1965.

19. ___, "Homomorphisms of Graphs and Automata," Technical Report 03105-44-T, Department of Computer and Communication Sciences, The University of Michigan, July 1966.

20. Hedetniemi, S.T. and Fleck, A.C. "*S*-Semigroups of Automata," Technical Report 6, THEMIS PROJECT, University of Iowa, 1970.

21. Hennie, F.C. *Finite State Models for Logical Machines*, John Wiley & Sons, New York. 1968.

22. Holland, J.H. "Cycles in Logical Nets," Technical Report 2722, 2794-4-7, Logic of Computers Group, The University of Michigan, 1959.

23. König, D. *Theorie der Endlichen und Unendlichen Graphen*, Chelsea, New York. 1950.

24. Krohn, K. and Rhodes, J. "Algebraic Theory of Machines I: Prime Decomposition Theorem for Finite Semigroups and Machines," *Trans. AMS, 116*, pp. 450-464, 1965.

25. McNaughton, R. "The Loop Complexity of Pure-Group Events," *Information and Control, 11*, pp. 167-176, 1967.

26. McNaughton, R. and Booth, T.M. "General Switching Theory," Technical Documentary Report ASD-TDR-62-599, Moore School of Electrical Engineering, University of Pennsylvania, 1962.

27. Meyer, J.F. "Theory and Design of Reliable Spacecraft Data Systems," Systems Engineering Lab, The University of Michigan, September 1970.

28. Meyer, J.F. and Yeh, K. "Diagnosable Machine Realizations of Sequential Behavior," *Digest 1971 International Symposium on Fault Tolerant Computing*, pp. 22-25, March 1971.

29. Miller, R.E. and Winograd, S. "On the Number of Transitions Entering the States of a Finite Automaton," IBM Research Note NC-320, 1963.

30. Moore, E.F. "Gedanken-Experiments on Sequential Machines," in *Automata Studies* (C.E. Shannon and J. McCarthy, eds.) Princeton University Press, Princeton, 1956.

31. Mukhopadhyay, A. "The Square Root of a Graph," *J. Combinatorial Theory, 2,* pp. 290-295, 1967.

32. Niven, I. and Zuckerman, H. *An Introduction to the Theory of Numbers,* John Wiley & Sons, New York. 1960.

33. Oehmke, R.H. "On *S**-Semigroups of Automata," Technical Report 8, THEMIS PROJECT, University of Iowa, 1969.

34. Yoeli, M. and Ginzburg, A. "On Homomorphic Images of Transition Graphs," *J. Franklin Inst., 278,* pp. 291-296, 1964.

35. Zeiger, H.P. "Loop-Free Synthesis of Finite-State Machines," Ph.D. Thesis, Department of Electrical Engineering, M.I.T. 1964.

36. Zeigler, B.P. "On the Feedback Complexity of Automata," Technical Report 08226-6-T, Department of Computer and Communication Sciences, The University of Michigan. Also Ph.D. Thesis. 1969.