Journal of Pure and Applied Algebra 28 (1983) 155-177
North-Holland Publishing Company 

# SYNERGY IN MACHINES

Gerald S. EISMAN
*Department of Mathematics, Saint Mary's College, Moraga, CA 94575, USA**

Communicated by J. Rhodes
Received November 1981

The capability of a finite state machine constructed of component machines in a composition with feedback is shown to be greater than the capabilities of series-parallel (or cascade) compositions of these same components. A measure of the amount of feedback in a construction is defined and a hierarchy of classes of machines is obtained by increasing the amount of feedback permitted in the members of each class.

## Introduction

The Krohn-Rhodes Prime Decomposition Theorem characterizes the structure of semigroups in terms of their basic building blocks, the simple groups and the unit semigroups. Moreover, the algebraic theory carries over to the structure of finite state machines in that each machine can be simulated by another which is constructed by a series-parallel composition of the corresponding fundamental machines.

The nature of a series-parallel composition is that it induces a partial ordering on its components in such a way that the functioning of a component cannot be affected by the functioning of another component which is higher in the ordering. That is, there is no feedback.

This paper examines the capability of a machine which is constructed by a composition in which there is feedback. In particular, the composition has circular feedback. That is, the machine is constructed of other machines hooked in a circle. Certain restrictions on the hook-ups between components in a circle are placed, and a certain interpretation on how a circle of components produces output is given. Then it is shown that the capability of a machine so constructed may be greater than that of every series-parallel composition of these same component machines. In addition, a hierarchy of classes of machines is obtained by allowing the members of a higher class to contain a circular feedback composition of members of the classes below. This hierarchy is shown to be proper at the lowest levels.

*The research for this paper was supported in part by the Saint Mary's College Faculty Development Fund.

The basic unit machine from which all constructions of increasing complexity will be built is a nerve net consisting of a single neuron with no axons to itself and no inhibition on its axons. If the input to the neuron exceeds the threshold of the neuron then the neuron is turned on; otherwise it is turned off. At each moment the effect of the input to such a neuron is independent of what state the neuron was in previously. That is, there is no memory. The level 0 class of machines in the above-mentioned hierarchy contains those machines which can be simulated by a series-parallel (or cascade) composition of these basic unit machines. At level $i$, machines which can be simulated by a cascade composition of level $i-1$ machines and circular compositions of level $i-1$ machines are included.

The semigroup of state transition maps induced by inputs into a machine may be used to measure the capabiliy of a machine. At level 0, because of the absence of memory, the associated semigroups may be constructed from unit semigroups containing only constant maps. At level 1, a neuron may have a path to itself, and its present state may depend on its past, thus memory is introduced. The associated semigroups may be constructed from unit monoids. At level 2 it is shown that the semigroups may contain cyclic subgroups, but that all subgroups are solvable. At level 3 the special linear groups, $\mathrm{PSL}_n(Z_2)$, appear in a natural way, and at level 4 the commutator subgroups of the orthogonal groups, $O_{2m}^{+}(2)$, appear.

This paper is organized in the following manner.

In Section 1 preliminary algebraic and machine notation, definitions, and theorems are given.

In Section 2 the formal definition of circular feedback is presented, and the hierarchy of machines with increasing feedback is defined.

In Section 3 the properness of the hierarchy for levels 0, 1, 2, and 3 is shown by considering the corresponding semigroups, and the semigroups at level 4 are examined.

Section 4 contains some comments on motivation and generalizations.

## 1. Preliminaries

Most of the notation, definitions, and theorems in this section are taken from articles in [1]. A reader familiar with this material may wish to skip to Section 2.

There are three areas which need preliminary definitions. These are machines, semigroups, and the relation between machines and semigroups.

*Machine preliminaries*

**1.1. Definition.** A *state output automaton* or *machine* is a 6-tuple $M=(Q,A,B,q_0,\delta,\lambda)$ where $Q$ is a finite set called the set of *states*, $A$ is a finite set called the *input alphabet*, $B$ is a finite set called the *output alphabet*, $q_0\in Q$ is the *initial state*, $\delta$ is a function $\delta:Q\times A\to Q$ called the *next-state function*, and $\lambda$ is a function $\lambda: Q \to B$ called the *output function*. (Note that in this version $\lambda$ depends only on $Q$ and produces a single symbol in $B$.)

The interpretation of the action of the machine is that if at time $t$ it is in state $q$ and is receiving input $a$ then it changes to state $\delta(q,a)$ and outputs $\lambda(\delta(q,a))$.

Extend $\delta$ to a function $\hat{\delta}: Q \times A^+ \to Q$ by defining for each $q \in Q$ and $a \in A$, $\hat{\delta}(q,a) = \delta(q,a)$ and for $a_1, a_2, \ldots, a_n \in A$, $n > 1$, $\hat{\delta}(q, a_1 a_2 \cdots a_n) = \delta(\hat{\delta}(q, a_1 a_2 \cdots a_{n-1}), a_n)$.

As a sequence of inputs from $A$ enters the machine a sequence of outputs from $B$ exits. In this way each machine computes a function from $A^+$ to $B$.

**1.2. Definition.** Let $M = (Q, A, B, q_0, \delta, \lambda)$ be a machine. Then the *function of the machine*, $M^f$, is the function $M^f: A^+ \to B$ defined by $M^f(a_1 a_2 \cdots a_n) = \lambda(\hat{\delta}(q_0, a_1 a_2 \cdots a_n))$ for $a_1, a_2, \ldots, a_n \in A$, $n \geq 1$.

One machine can be simulated by another if the function of the first can be computed by the second by encoding the inputs to the former so that they are compatible with the latter, then computing the function of the second, and then decoding the outputs.

**1.3. Definitions.** Let $M_i = (Q_i, A_i, B_i, q_{0i}, \delta_i, \lambda_i)$ for $i = 1, 2$ be two machines. Then $M_2$ *simulates* $M_1$ if there exists homomorphisms $h_1: A_1^* \to A_2^*$ and $h_2: B_2^* \to B_1^*$ such that $h_2 \circ M_2^f \circ h_1 = M_1^f$. In this case we say $M_1$ *divides* $M_2$ written $M_1 | M_2$.

Two obvious ways to hook-up machines are series and parallel compositions. Here the cascade composition is presented which contains series and parallel compositions as special cases. In a cascade of $M_2$ with $M_1$, the input into $M_1$ depends only on external input, and the input to $M_2$ depends on external input and the previous state of $M_1$.

**1.4. Definition.** Let $M_i = (Q_i, A_i, B_i, q_{0i}, \delta_i, \lambda_i)$ for $i = 1, 2$ be two machines. Let $A$ be a finite set, $h$ a function $h: A \times Q_1 \to A_2$, and $g$ a function $g: A \to A_1$. A *cascade composition of* $M_2$ *with* $M_1$ *with connecting map h*, $M_1 \times_h M_2$, is a machine $M = (Q_1 \times Q_2, A, B_1 \times B_2, q_{01} \times q_{02}, \delta, \lambda)$ where for $q_i \in Q_i$, $i = 1, 2$ and $a \in A$, $\delta((q_1, q_2), a) = (\delta_1(q_1, g(a)), \delta_2(q_2, h(a, \lambda_1(q_1))))$ and $\lambda(q_1, q_2) = (\lambda_1(q_1), \lambda_2(q_2))$.

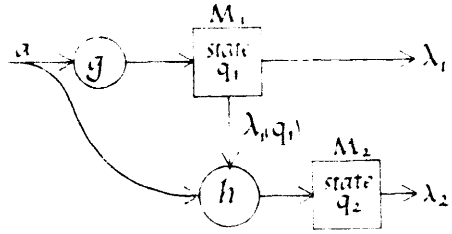



Fig. 1

Diagramatically, a cascade composition appears as in Figure 1. Note that if the value of the function $h$ is independent of the state $q_1$ of $M_1$ then the cascade is a parallel composition, and if the value of $h$ is independent of the external input, $a$, then the cascade is a series composition where the output from $M_1$ to $M_2$ is delayed one unit of time. The advantages of the cascade composition are discussed in [1]. The essential property in each of these types of composition is that there are no loops, i.e., no feedback. The cascade composition can easily be extended to three or more machines.

**1.5. Definition.** Let $\mathcal{M}$ be a collection of machines. Define *cascade* $(\mathcal{M})$ = set of machines which can be simulated by a cascade composition of members of $\mathcal{M}$.

*Semigroup preliminaries*

**1.6. Definition.** A *right action semigroup* is a triple $(X, S, \Theta)$ where $X$ and $S$ are non-empty sets, $\Theta$ is a function, $\Theta: X\times S\to X$, and for all $s_1, s_2\in S$ there exists $s_3\in S$ such that $\Theta(\Theta(x,s_1),s_2)=\Theta(x,s_3)$ for all $x\in X$. In this case we write $s_1s_2=s_3$. When $\Theta$ is understood we write $(X, S)$ for $(X, S, \Theta)$ and $xs$ for $\Theta(x,s)$. The action is *faithful* if for all $s_1, s_2\in S$ if $s_1\neq s_2$ then for some $x\in X$, $xs_1\neq xs_2$. If the action is faithful then multiplication is well defined in $S$ and $S$ is called the *underlying semigroup* of $(X, S)$.

Given a semigroup $S$, let $S^1=S$ if $S$ contains an identity, and otherwise $S^1=S\cup\{1\}$ where $1\notin S$, multiplication in $S$ is unchanged and 1 is the identity of $S\cup\{1\}$. Let $(S^1, S)$ denote the right action semigroup $(S^1, S, \Theta)$ where $\Theta(s_1, s)=s_1s$ for all $s_1\in S^1$ and $s\in S$.

**1.7. Definition.** Given two right action semigroups $(X, S)$ and $(Y, T)$, $(X, S)$ *divides* $(Y, T)$, $(X, S)\mid(Y, T)$, if there exists a subset $Y'$ of $Y$, a subsemigroup $T'$ of $T$, a function, $h'$, from $Y'$ onto $X$, and a homomorphism, $h$, from $T'$ onto $S$ such that for all $y'\in Y'$ and $t'\in T'$ (1) $y't'\in Y'$ and (2) $h'(y't')=h'(y')h(t')$. Pictorially we have the following commutative diagram:

$$\begin{array}{ccc} Y' & \xrightarrow{\;t'\;} & Y' \\ \downarrow{\scriptstyle h'} & & \downarrow{\scriptstyle h'} \\ X & \xrightarrow{\;h(t')\;} & X \end{array}$$

Division in right-action semigroups can be shown to be transitive. Given two semigroups $S$ and $T$, we say $S$ divides $T$ if $(S^1, S)\mid(T^1, T)$.

Semigroups may be combined by the direct product or the more general Wreath product which contains the direct product as a special case.

**1.8. Definition.** Let $(X_i, S_i)$ for $i=1,2$ be two right action semigroups. The *Wreath product of* $(X_1, S_1)$ *and* $(X_2, S_2)$, $(X_1, S_1) \wr (X_2, S_2)$, is the right action semigroup $(X, S)$ where $X = X_1 \times X_2$ and $S$ is the set of all pairs $(s_1, h)$ where $s_1 \in S_1$, $h$ is a function, $h: X_1 \to S_2$, and where for all $x = (x_1, x_2) \in X$ and $s = (s_1, h) \in S$, $xs = (x_1, x_2)(s_1, h) = (x_1 s_1, x_2 h(x_1))$. Note that if the function $h$ is restricted to being a constant function then $S$ is the direct product $S_1 \times S_2$. Note that if the action in the factors is faithful then so is the action in the Wreath product.

The product can easily be extended to three or more right action semigroups.

**1.9. Definition.** Let $\mathscr{S}$ be a collection of faithful right action semigroups. Define $\bar{\mathscr{W}}(\mathscr{S})$ = *wreath closure divisor of* $\mathscr{S}$ = set of right action semigroups $(Y, T)$ such that $(Y,T) \mid (X_1, S_1) \wr (X_2, S_2) \wr \cdots \wr (X_n, S_n)$ for $n \geq 1$, $(X_i, S_i) \in \mathscr{S}$, $i = 1, \ldots, n$. Define $\mathscr{W}(\mathscr{S})$ = set of underlying semigroups of the members of $\bar{\mathscr{W}}(\mathscr{S})$.

Certain *combinatorial* semigroups, i.e., containing no non-trivial subgroups, will be important in the sequel. The action of each is given in the tables below. $C_i$ denotes the constant map to $i$. $I$ denotes the identity map. The name of the underlying semigroup of each appears below the table.

| $x^s$ | $C_1$ |
|---|---|
| 1 | 1 |

$U_0$

| $x^s$ | $C_0$ | $C_1$ |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 1 |

$U_1$

| $x^s$ | $C_1$ | $I$ |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |

$U_2$

| $x^s$ | $C_0$ | $C_1$ | $I$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |

$U_3$

These semigroups are called *unit semigroups*. They are particularly important in view of the following definitions and theorem. The theorem is the analogue for semigroups to the Jordan-Hölder Theorem for groups.

**1.10. Definition.** A semigroup $S$ is *irreducible* if $(S^1, S) \mid (S_1^1, S_1) \wr (S_2^1, S_2)$ implies $(S^1, S) \mid (S_1^1, S_1)$ or $(S^1, S) \mid (S_2^1, S_2)$.

**1.11. Definition.** Let $S$ be a semigroup. Let *Primes*$(S) = \{G$: $G$ is a simple group and $(G, G) \mid (S^1, S)\}$. Let $\mathscr{S}$ be a collection of semigroups. Let *Primes*$(\mathscr{S}) = \bigcup_{S \in \mathscr{S}}$ Primes$(S)$.

**1.12. Theorem.** Prime decomposition theorem for finite semigroups (Krohn–Rhodes).

(a) *The set of finite irreducible semigroups = the set of simple groups* $\cup$ *the set of unit semigroups.*

(b) *Let* $S$ *be a finite semigroup,* $\mathscr{S}$ *a collection of finite semigroups. Then* $S \in \mathscr{W}(\mathscr{S} \cup \{U_3\})$ *iff* Primes$(S) \subseteq$ Primes$(\mathscr{S})$.

For a proof, see [1].

*Relationships between machines and semigroups*

Each input string to a machine $M=(Q, A, B, q_0, \delta, \lambda)$ induces a transition map on $Q$. That is, let $w \in A^+$ and define $f_w : Q \to Q$ by $f_w(q) = \delta(q, w)$.

**1.13. Definition.** Let $M=(Q, A, B, q_0, \delta, \lambda)$ be a machine. Let $(Q, M^S, \Theta)$ be the right action semigroup where $M^S = \{f : Q \to Q$ such that there exists $w \in A^+$ with $f = f_w\}$ and $\Theta : Q \times M^S \to Q$ is defined by $\Theta(q, f) = f(q)$. The underlying semigroup, $M^S$, is called the *semigroup of the machine.* Let $\mathcal{M}$ be a collection of machines. Define $\mathcal{M}^S = \{M^S : M \in \mathcal{M}\}$.

**1.14. Definition.** Let $S$ be a semigroup. Define $S^M$ = *machine of* $S = (S^1, S, S, 1, \delta, \lambda)$ where $\delta(s_1, s_2) = s_1 s_2$ for all $s_1, s_2 \in S^1$ and $\lambda(s) = s$ for all $s \in S^1$.

The following theorem gives the important relationship between the semigroups of the components of a cascade composition of machines and the semigroup of the whole machine. The theorem states that the semigroup of the total divides the Wreath product of the semigroups of the components. (Again the proof is in [1].)

**1.15. Theorem.** *Let* $\mathcal{M}$ *be a collection of machines. Then* $(\text{cascade}(\mathcal{M}))^S \subseteq \mathcal{W}(\mathcal{M}^S)$. □

Theorem 1.15 yields the following useful result. If $\mathcal{M}$ is a collection of machines and $T$ is an irreducible semigroup such that $T$ does not divide $M^S$ for each $M \in \mathcal{M}$ then $T$ does not divide the semigroup of each cascade composition of members of $\mathcal{M}$. This fact will be used frequently in the following sections.

## 2. Circular feedback in nerve nets

The reader unfamiliar with nerve nets may wish to use [4] as a reference. The nerve net here contains a minor restriction in the structure and a major modification in the interpretation of the usual model. The restriction is that the nets will possess only excitory axons. That is, the firing of one neuron will not inhibit the firing of another. Such nets are called *inhibition free.*

**2.1. Definition.** An (*inhibition free*) *nerve net, L*, is a 4-tuple $L=(N, A, I, t)$ where $(N, A)$ is a directed graph ($A$ may contain multiple edges), $I$ is a finite set each member of which is assigned to a member of $N$ (each member of $I$ is an edge with no beginning node but ending at its assigned member of $N$), and $t$ is a function, $t : N \to \mathbb{R}$. $N$ is called the set of *neurons*, $A$ the set of *internal axons*, $I$ the set of *input axons*, and $t$ the *threshold assignment function.* For each $n \in N$, $t(n)$ is called the *threshold* of $n$. Let $B^n \subseteq A$ be the set of axons beginning at $n$. Let $E^n \subseteq A \cup I$ be the axons ending at $n$.

Pictorially, neurons will be represented as isoceles triangles, each enclosing a number equal to its threshold, e.g., ◁1. Each input axon will be represented as a line ending in a solid dot near a leg of the triangle representing the neuron assigned to the input axon, e.g.,

————• ◁1.

And each internal axon, $(n_1, n_2)$ for $n_i \in N$ $i = 1, 2$, as a line from the base of the triangle representing $n_1$ to a solid dot near a leg of the triangle representing $n_2$, e.g.,

◁1————• ◁1.

For simplicity if there are more than one axon between two neurons, they may all be represented as a branching line, e.g.,

◁1————<: ◁1.

The restriction to inhibition free nets is necessary for the results in Section 3 and is discussed further in Section 4.

At each moment of time (time is considered as discrete moments) a neuron is either firing (on = 1) or not firing (off = 0). If it is firing, it sends a *pulse* along each internal axon beginning at the neuron. A neuron will fire if the number of internal or input axons incident to it which contain pulses equals or exceeds its threshold. In this way each neuron computes a logical function on its incident axons, called its threshold function. We will consider all pulses to travel synchronously along all axons and the computation of the threshold functions to be instantaneous.

A nerve net may be considered as a machine by letting its state at time $t$ be the set of neurons in the net which are firing at time $t$. The major modification mentioned above is in the interpretation of how the net changes state. Here the important feature is that certain designated subsets of the net will 'stabilize' before they send pulses along their axons to other portions of the net. In the usual model the neuron is the unit of computation, computing its threshold function and sending an output. Here a subset of neurons compute a more complex function before sending information to the rest of the net. This model will be called a *locally stable*

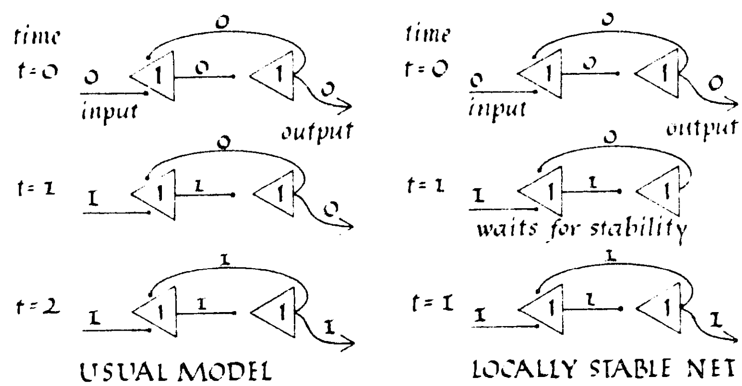



Fig. 2

*nerve net*. Some comments on its origin appear in Section 4.

Before proceeding to a formal definition the example in Figure 2 demonstrates the difference between the usual model and this one. The locally stable net computes in one unit of time the function that takes two units of time to compute in the usual model.

More generally, in a locally stable net, a subset of neurons may be partitioned into smaller sets, each set of which is partitioned into yet smaller sets, and so on, where the smaller sets at each level must stabilize before their union stabilizes.

**2.2. Definition.** A *locally stable nerve net*, (l.s.n.), is a net $L=(N,A,I,t)$ and a finite sequence of partitions of $N, P_1, P_2, \ldots, P_r$, where $P_i$ is a refinement of $P_{i+1}$, $i=1,\ldots,r-1$. $r$ is called the *rank* of $L$, and if the partition sequence is empty $r=0$. The partitions, $P_i$, determine a sequence of increasing subsets of $A$. Let $A_i$, the *i-th level axons*, = the axons in $A$ whose endpoints are members of the *same* element of $P_i$, $i=1,\ldots,r$. Let $A_{r+1}=A$. Since $P_i$ is a refinement of $P_{i+1}$ then $A_i \subseteq A_{i+1}$, $i=1,\ldots,r$.

**2.3. Definition.** Let $L=(N,A,I,t)$ with partition sequence $P_1, P_2, \ldots, P_r$ be an l.s.n. Let 0 be a subset of $N$. Then a *machine interpretation of L with output* 0 is a machine $M=(2^N, 2^I, 2^0, S_0, \delta, \lambda)$ where $S_0 \subseteq N$, $\lambda : 2^N \to 2^0$ is defined by $\lambda(S)=S \cap 0$ for each $S \subseteq N$, and $\delta$ is defined by the following procedure.

For each set $A' \subseteq A \cup I$ let $F(A')$ designate the set of members of $A'$ which carry a pulse. Let $A_1, A_2, \ldots, A_{r+1}$ be determined from $P_1, \ldots, P_r$ as above. Now suppose $M$ is in state $S$ and $J$ is input.

First all input axons in $J$ and all internal axons beginning at neurons in $S$ carry a pulse and no other axons carry a pulse. That is, $F(A \cup I)=J \cup \bigcup_{n \in S} B^n$.

If $r=0$, the next state is computed directly from the axons carrying pulses. That is, $\delta(S,J)=\{n \in N: |F(A \cup I) \cap E^n| \geq t(n)\}$.

Otherwise, begin the *level* 1 *computation* as follows. $F(A \cup I \sim A_1)$ is held fixed while for each $S_1$ in $P_1$ the neurons in $S_1$ simultaneously compute their threshold functions, change $F(A_1)$ only, recompute their threshold functions with the new value of $F(A_1)$, again change $F(A_1)$, and repeat this process until $F(A_1)$ does not change. At this point $F(A_2)$ is changed to correspond to the threshold function computed by their beginning neurons. Call this the *completion* of the level 1 computation.

Now suppose the level $i$ computation is complete for some $i$, $1 \leq i \leq r-1$. To perform the level $i+1$ computation the level $i$ computation is repeated until $F(A_{i+1})$ does not change. At this point $F(A_{i+2})$ is changed, and the level $i+1$ computation is complete.

When the level $i, i=1,\ldots,r-1$, computation is complete the level $i+1$ computation is begun. When the level $r$ computation is complete $M$ is in the next state $=\delta(S,J)$.

This procedure is defined formally in Section 4.

A few important observations should be made here.

*Note* 1: When the level $i$ computation is complete, the computation is begun again at level 1, since to perform the level $i+1$ computation the level $i$ computation is repeated, and to perform the level $i$ computation the level $i-1$ computation is repeated, and so on.

*Note* 2: It is quite possible that at some level the computation is never completed because the net does not stabilize. In this case the next state function is not defined.

*Note* 3: Suppose $P_r$ contains the single element $N$. Then when the $r$-th level computation is complete the entire net is stable. In this case if the input $J$ is repeated the net will remain in the same state. That is, if $\delta(S,J)=S_1$ then $\delta(S_1,J)=S_1$.

Let $L$ be an l.s.n. Let $L^S=M^S$ where $M$ is a machine interpretation of $L$. (Since $M^S$ is independent of 0, $L^S$ is well defined.) If $\mathscr{L}$ is a collection of l.s.n.'s let $\mathscr{L}^S=\{L^S: L\in\mathscr{L}\}$.

The concept behind local stability is that the unit of computation in a net need not be the neuron but rather a subset of neurons. In fact the sequence of partitions is a way of constructing larger units of computation from smaller ones. The interesting case occurs when some sets in $P_i$ are connected in a loop by axons in $A_{i+1}$.

For each $i=1,\dots,r$ and each $Q\in P_i$ define the subnet $L_Q=(Q,A_Q,I_Q,t_Q)$ where $A_Q=$ the axons in $A_i$ with both endpoints in $Q$, i.e., $A_Q=A_i\cap\bigcup_{n\in Q}B^n\cap\bigcup_{n\in Q}E^n$, $I_Q=$ the input axons and the internal axons that end in $Q$ but do not begin in $Q$, i.e., $I_Q=\bigcup_{n\in Q}E^n\sim\bigcup_{n\in Q}B^n$, and $t_Q$ is the threshold assignment function, $t$, restricted to $Q$. Each member $R\in P_{i+1}$ is a union of members of $P_i$ so that $L_R$ is constructed from the subnets $L_Q$ such that $Q\subseteq R$, where the subnets are linked together by axons in $A_R\sim A_i$.

For each $R\in P_{i+1}$ consider the digraph whose vertices are the subnets $L_Q: Q\subseteq R$, $Q\in P_i$, and whose edges correspond to axons in $A_{i+1}$ between $Q_1,Q_2\subseteq R$. If this digraph has no closed paths, i.e., no cycles, then clearly $L_R$ is a cascade composition of $L_Q: Q\subseteq R$. Algebraically then, by Theorem 1.15, the semigroup of a machine interpretation of $L_R$ divides a Wreath product of the semigroups of the $L_Q$'s.

In this paper a particular class of nets will be examined in which $L_R$ *does* exhibit feedback for at least one $R\in P_{i+1}$, for each $i$. In particular the case will be examined where the digraph above is a simple closed path, that is, a circle. Some interesting algebraic properties result.

For the following definitions let $L=(N,A,I,t)$ with partition sequence $P_1,P_2,\dots,P_r$ be a locally stable net. Let $P_0$ be the discrete partition of $N$, i.e., $P_0=\{\{n\}: n\in N\}$. Let $A_1,\dots,A_r$ be determined as above.

**2.4. Definition.** Let the *i-th level graph, $G_i$*, $i=1,\dots,r+1$, be the digraph whose vertices are the members of $P_{i-1}$ and whose edges $=\{(Q_1,Q_2)$: $Q_1,Q_2\in P_{i-1}$ and there exists an axon in $A_i$ from a neuron in $Q_1$ to a neuron in $Q_2\}$.

It should be noted that $G_i$ does not contain multiple edges. However a particular count on the multiplicity in an edge will be important in the sequel.

**2.5. Definition.** $G_i$, $i \geq 2$, is *k-linked* if for each edge $(Q_1, Q_2)$ in $G_i$ there are at most $k$ members, $Q_{11}, Q_{12}, \ldots, Q_{1k}$, of $P_{i-2}$ such that for each $j = 1, \ldots, k$ there is an axon in $A_i$ from a neuron in $Q_{1j}$ to a neuron in $Q_2$. $L$ is *k-linked* if $G_i$ is $k$-linked $i = 2, \ldots, r$.

**2.6. Definition.** $L$ is *locally connected* if for each $i = 1, \ldots, r$ and each $Q$ in $P_i$ the subgraph of $G_i$ with vertices $= \{Q_1 : Q_1 \in P_{i-1}$ and $Q_1 \subseteq Q\}$ and all edges in $G_i$ connecting these vertices is a connected graph. (It should be noted that these subgraphs are the connected components of $G_i$.)

**2.7. Definition.** $L$ has *circular feedback* if for $i = 1, \ldots, r$ the connected components of $G_i$ are singletons or circles. That is, if $C$ is a connected component of $G_i$, then $C$ consists of a single vertex or $C$ consists of distinct vertices $v_1, v_2, \ldots, v_m$ and edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{m-1}, v_m), (v_m, v_1)$.

$G_{r+1}$ is not included in the above definition. In fact the structures to be studied are those in which all feedback occurs locally, i.e., in the graphs $G_1, \ldots, G_r$.

**2.8. Definition.** $L$ is *ultimately sequential* if $G_{r+1}$ contains no closed paths. (It should be noted that if $L$ is ultimately sequential then $L$ is a cascade composition of $\{L_Q : Q \in P_r\}$.)

The remainder of this paper concerns the class of inhibition free, locally stable, locally connected, 2-linked, ultimately sequential nerve nets with circular feedback. These shall be called *circular feedback nets*, the class referred to as CFN. Comments on the restrictions for membership in this class appear in Section 4.

**2.9.** Let $\mathrm{CFN}_r$ be the class of nets $L$ in CFN whose partition sequence has length $r$.

It should be noted that if $L \in \mathrm{CFN}_r$ has partition sequence $P_1, \ldots, P_r$ then an equivalent net in $\mathrm{CFN}_{r+1}$ can be constructed by letting $P_{r+1} = P_r$. Therefore $\mathrm{CFN}_r^S \subseteq \mathrm{CFN}_{r+1}^S$ and there exists a hierarchy in $\mathrm{CFN}^S$. The properness of this hierarchy is the subject of the next section.

## 3. The CFN hierarchy

In this section the semigroups of the machine interpretations of the nets in $\mathrm{CFN}_r$ will be examined for $r = 0, 1, 2, 3, 4$. It will be shown that $\mathrm{CFN}_r^S \subsetneq \mathrm{CFN}_{r+1}^S$ for

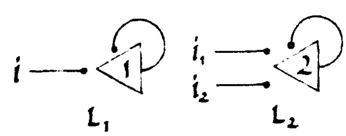
$r=0,1,2$ so that the class of machines which can be simulated by a net in $\mathrm{CFN}_r$ is a smaller class than those that can be simulated by a net in $\mathrm{CFN}_{r+1}$.

**3.1. Theorem.** *Let* $L \in \mathrm{CFN}_0$. *Then* $L^S \in W(U_0, U_1)$.

**Proof.** Since $L$ has rank 0 there is no partition sequence, and since $L$ is ultimately sequential then $L$ is a cascade composition of single neurons. Consider the machine of a single neuron and its input axons. The states of the machine $=\{\text{off}, \text{on}\}$ but regardless of what state the machine is in, it will be turned on if the input exceeds its threshold and turned off otherwise. Therefore the semigroup of the machine is isomorphic to either $U_0$ or $U_1$. The result follows from Theorem 1.15. □

**3.2. Theorem.** *There exists* $L_1, L_2 \in \mathrm{CFN}_1$ *for which* $L_1^S \cong U_2$ *and* $L_2^S \cong U_3$.

Fig. 3



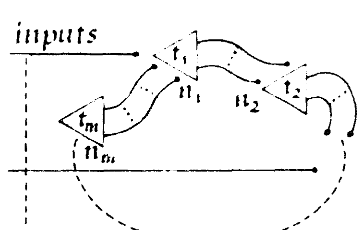

**Proof.** Consider the net in Figure 3 where both have the partition consisting of the singleton of the neuron. In $L_1$ if input $i$ is off it induces the identity map on the states of $L_1$, and if it is on it induces the constant map, $C_1$. Therefore $L_1^S \cong U_2$. In $L_2$ the input $i_1 = i_2 = \text{off}$ induces the map $C_0$, the input $i_1 = i_2 = \text{on}$ induces $C_1$, and the inputs $i_1 = \text{on}$ and $i_2 = \text{off}$ or $i_1 = \text{off}$ and $i_2 = \text{on}$ induce $I$. Therefore $L_2^S \cong U_3$. □

**3.3. Corollary.** $\mathrm{CFN}_0^S \subsetneq \mathrm{CFN}_1^S$. □

To completely describe rank 1 nets we need the following lemma.

**3.4. Lemma.** *Suppose* $L=(N, A, I, t) \in \mathrm{CFN}_1$ *and the* 1*st level graph of* $L$ *consists of a single component, and suppose* $a$ *is an input to* $L$ (*interpreted as a machine*). *Then either* $a$ *induces a constant map on the states of* $L$ *or the range of the map induced by* $a$ *consists of two states,* $X_0 = \emptyset$ *and* $X_1 = N$, *the map is the identity map on these two states, and the map is undefined on all other states.*

Fig. 4



**Proof.** By hypothesis, $L$ is a single neuron or a single circle as in Figure 4. In the former case $L \in \text{CFN}_0$. So assume the latter, and suppose $a$ is an input to $L$ which does not induce a constant map, i.e., $f_a \neq C_X$ for each $X \subseteq N$. Then the range of $f_a$ contains at least two states, $X_0$ and $X_1$. Then there must be some neuron which is off in one of these states ($X_0$ say) and on in the other. Number the neurons $1, 2, \ldots, m$ beginning with this neuron. Since $L$ is a locally stable net, the neurons are stable in either $X_0$ or $X_1$. Clearly the condition of neuron 2 is completely determined from the input and the condition of neuron 1. Similarly for $3, 4,, \ldots, m$. So if the state of any neuron in $X_0$ is the same as its state in $X_1$, then the state of all neurons must be the same. Then since the net is inhibition free, all neurons must be off in $X_0$ and on in $X_1$. Since the next state function is determined by the net being stable then $f_a$ is the identity map on $X_0$ and $X_1$.

Now consider the action of $f_a$ on some state $X$. Suppose neuron $j$ is on in state $X$. Since $f_a$ maps $X_1$ to $X_1$ and the state of neuron $j+1 \pmod m$ is determined by $a$ and $j$ then in the first step of the level 1 computation neuron $j+1$ is turned on. In the next step neuron $j+2$ is turned on and so on. Similarly if neuron $j$ is initially off. Therefore the level 1 computation can only be completed if $X = X_1$, or $X = X_0$. □

**3.5. Theorem.** *Let* $L \in \text{CFN}_1$. *Then* $L^S \in W(U_0, U_1, U_2, U_3)$.

**Proof.** By definition of rank 1, $L$ has a single partition $P_1$ each member of which, along with its internal axons, is a single neuron or a circle. Since $L$ is ultimately sequential, it is sufficient to consider only one member of $P_1$ by Theorem 1.15. So consider the subnet of a single member of $P_1$. Suppose an input sequence $a_1 a_2 \cdots a_n$ induces a permutation on some states. Clearly if $f_{a_j}$ is a constant map for some $j$ then so is $f_{a_1 a_2 \cdots a_n}$. Therefore by Lemma 3.4 each $f_{a_j}$ is the identity map, and the permutation is trivial. The result follows from Theorem 1.12. □

**3.6. Theorem.** *For each* $n \geq 1$ *there exists* $L_n \in \text{CFN}_2$ *such that* $L_n^S$ *contains the cyclic group* $Z_n$.

**Proof.** Consider the net $L_n$ in Figure 5, where the partition sequence of $L_n$ is $P_1 = \{\{1\}, \ldots, \{2n\}\}$ and $P_2 = \{1, 2, \ldots, 2n\}$. Let input $a$ send a pulse along the input axons to each of the odd numbered neurons, and input $b$ send a pulse to each of the even numbered neurons. Let $S_i$, $i = 1, \ldots, 2n$, be the state where the $i$-th

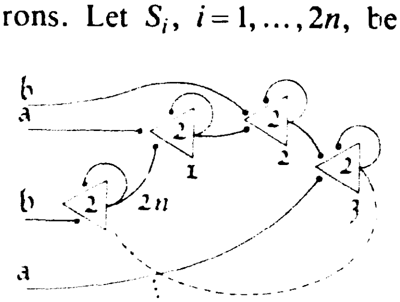



Fig. 5

neuron is on and all others are off. Then on these states $f_a$ and $f_b$ are the maps

$$f_a(S_i) = \begin{cases} S_i & \text{if } i \text{ is odd,} \\ S_{i+1(\text{mod } 2n)} & \text{if } i \text{ is even,} \end{cases}$$

and

$$f_b(S_i) = \begin{cases} S_i & \text{if } i \text{ is even,} \\ S_{i+1} & \text{if } i \text{ is odd,} \end{cases}$$

and $f_{ab}$ is the permutation of order $n, f_{ab}(S_i) = S_{i+2(\text{mod } 2n)}$ on the set

$$\{S_i : i = 2, 4, \ldots, 2n\}. \quad \square$$

**3.7. Corollary.** $\text{CFN}_1^S \subsetneq \text{CFN}_2^S$. $\square$

Suppose that $L \in \text{CFN}_2$. Since $L$ is ultimately sequential then $L$, interpreted as a machine, is a cascade composition of the subnets, $L_Q$ for each $Q$ in $P_2$. It will be shown that $L_Q^S$ contains only solvable subgroups from which it follows by Theorems 1.15 and 1.12 that $L^S$ contains only solvable subgroups. We proceed with a series of lemmas.

**3.8. Lemma.** *Let L be a net as in Lemma* 3.4 *interpreted as a machine. Let $a_1$ and $a_2$ be two inputs into L such that $a_1 \subseteq a_2$. Let $X_0 = \emptyset$, $X_1 = N$. Then either*

(a) *$f_{a_1}$ and $f_{a_2}$ are both the identity map on $\{X_0, X_1\}$,*
(b) *$f_{a_1}$ is the identity map and $f_{a_2} = C_{X_1}$ = the constant map onto $X_1$,*
(c) *$f_{a_2}$ is the identity map and $f_{a_1} = C_{X_0}$, or*
(d) *$f_{a_1} = C_{S_1}$ and $f_{a_2} = C_{S_2}$ for some $S_1$ and $S_2$ such that $S_1 \subset S_2$.*

**Proof.** The result follows immediately from Lemma 3.4, the fact that the net is inhibition free, and the hypothesis that $a_2$ provides equal or more input pulses than $a_1$ to each neuron. $\square$

The remaining lemmas concern a subnet, $L_Q$, for some $Q \in P_2$. Such a net is pictured in Figure 6. The set $Q = Q_1 \cup Q_2 \cup \cdots \cup Q_m$, $m < 1$, where $Q_j \in P_1$, $j = 1, \ldots, m$, is either a single neuron or a circle of neurons.

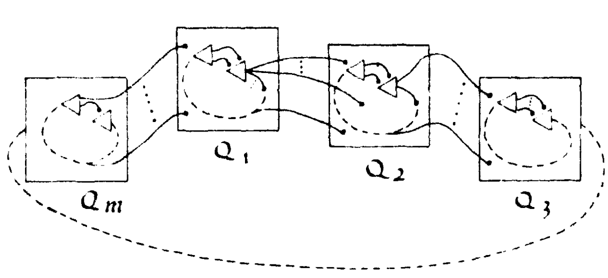



Fig. 6

Let $G_Q$ be a subgroup of $L_Q^S$ and let $\mathcal{S}_G = \{S_1, S_2, \ldots, S_p\}$ be a maximal set of states on which $G_Q$ acts transitively. Let $w = a_1 a_2 \cdots a_n$ be an input sequence into $L_Q$

which induces the permutation $f_w=\pi$ on $\mathscr{S}_G$. Assume $G_Q$ is non-trivial.

For each $a_i$ occurring in $w$ the level 1 computation may be repeated several times with the net going through many intermediate states before it stabilizes. Call the completion of a level 1 computation a *moment*. For different states of the net and an input, $a_i$, it might take a different number of moments to complete the level 2 computation. However when the level 2 computation is complete, the entire net is stable so that recomputing the level 1 computation will not change the state of any neuron in the net. Therefore it may be assumed that it takes the same number of moments to compute the next state function for each initial state and each input. Suppose the permutation induced by $w$ takes a total of $t$ moments to complete. Suppose the net is in state $S_i$ initially when the sequence is input. Then after $k$ moments, $k=0,\dots,t$, call the state of the net $S_{ik}$. Clearly $S_{ik}\neq S_{jk}$, $i\neq j$, as otherwise $S_i$ and $S_j$ would be mapped onto the same state by $f_w$.

Since $L_Q$ is 2-linked then for each $j=1,\dots,m$ there exists either 1 or 2 neurons in $Q_j$ with axons leading to $Q_{j+1(\text{mod } m)}$. Let $N_j=\{n\in Q_j$: there exists $m$ such that $(n,m)\in A_2\sim A_1\}$. It turns out that the state of the $N_j$'s determine the states $S_{ik}$. Call $Q_j\cap S_{ik}$, for each $i,k,j$, the *j-th coordinate of* $S_{ik}$. Let $P_{ik}=\bigcup_{j=1,\dots,m} N_j\cap S_{ik}$. Call $N_j\cap S_{ik}$ the *j-th coordinate of* $P_{ik}$.

**3.9. Lemma.** *For each* $k=0,\dots,t$, $P_{ik}\neq P_{hk}$ *if* $i\neq h$.

**Proof.** Suppose $P_{ik}=P_{hk}$ for some $i,h$ at moment $k$. The inputs into $Q_{j+1(\text{mod } m)}$ are completely determined by the input axons and $N_j$, so the inputs into $Q_{j+1(\text{mod } m)}$ are the same for $S_{ik}$ and $S_{hk}$, $j=1,\dots,m$. By Lemma 3.4 each input induces either a constant map or the identity map. In the former case the $(j+1)$-st coordinates of $S_{ik+1}$ and $S_{hk+1}$ are equal. In the latter case the map is undefined unless the $(j+1)$-st coordinate of $S_{ik}$ and $S_{hk}$ are either $\emptyset$ or $Q_{j+1}$, both of which can be determined by the $(j+1)$-st coordinate of $P_{ik}$ and $P_{hk}$. Therefore $S_{ik+1}=S_{hk+1}$ but then $S_{it}=S_{ht}$ contradicting that $\pi$ is a non-trivial permutation. □

Call two sets $X$ and $Y$ *comparable* if $X\subseteq Y$ or $Y\subseteq X$. It cannot be the case that for some $i$ and $h$ the $j$-th coordinates of $P_{ik}$ and $P_{hk}$ are comparable for all $j$ with the containment always in the same direction.

**3.10. Lemma.** *There does not exist i, h with* $i\neq h$ *such that* $N_j\cap S_{ik}\subseteq N_j\cap S_{hk}$ *for* $j=1,\dots,m$.

**Proof.** Suppose for some $i\neq h$, $N_j\cap S_{ik}\subseteq N_j\cap S_{hk}$ for $j=1,\dots,m$. Then by Lemma 3.8 $N_j\cap S_{ik+1}\subseteq N_j\cap S_{hk+1}$ for $j=1,\dots,m$, and so $N_j\cap S_{it}\subseteq N_j\cap S_{ht}$ for all $j$. That is, the $j$-th coordinate of $\pi(S_i)$ is contained in the $j$-th coordinate of $\pi(S_h)$. Let $S_1'=\pi(S_i)$ and $S_2'=\pi(S_h)$. Since $G_Q$ is transitive there exists a permutation $\pi'$ in $G_Q$ mapping $S_1'$ to $S_2'$. Suppose $\pi'$ has order $r$. By using Lemma 3.8 repeatedly it is clear that for $j=1,\dots,m$ since the $j$-th coordinate of $S_1'\subseteq j$-th coordinate of $S_2'$ then the $j$-th coordinate of $\pi'(S_1')\subseteq$ the $j$-th coordinate of $\pi'(S_2')$ and so the $j$-th coordinate of $\pi'(S_2')\subseteq$ the $j$-th coordinate of $(\pi')^2(S_2')$ and so on, implying that $N_j\cap S_1'\subseteq N_j\cap\pi'(S_1')\subseteq\cdots\subseteq N_j\cap(\pi')^r(S_1')=N_j\cap S_1'$. Therefore by Lemma 3.9 $S_1'=S_2'$ contradicting $i\neq h$. □

With two neurons it would appear that $N_j$ could be in four possible states. However, at most two can occur at each moment, and if two do occur, they are comparable.

**3.11. Lemma.** *For each $k=0,\ldots,t$ and each $j=1,\ldots,m$, $|\{N_j\cap S_{ik}: i=1,\ldots,p\}|=1$ or 2. Moreover if $Y_1=N_j\cap S_{ik}$ and $Y_2=N_j\cap S_{hk}$ and $p>2$ then $Y_1$ and $Y_2$ are comparable.*

**Proof.** If $p\leq 2$ the result is trivial. If $N_j$ consists of a single neuron then $N_j\cap S_{ik}=\emptyset$ or $N_j$, and the result follows. Suppose for some $j$, $N_j$ consists of two neurons, $n_{j1}$ and $n_{j2}$. Then $N_j\cap S_{ik}$ has four possibilities: $X_0=\emptyset$, $X_1=N_j$, $X_2=\{n_{j1}\}$, and $X_3=\{n_{j2}\}$. Of these only $X_2$ and $X_3$ are incomparable. Assume $p>2$.

Suppose for some $j$ and two states $S_{i0}$ and $S_{h0}$, $N_j\cap S_{i0}$ and $N_j\cap S_{h0}$ are comparable. By using Lemma 3.8 repeatedly, $N_{j+k}\cap S_{ik}$ and $N_{j+k}\cap S_{hk}$ are comparable for $k=0,1,2,\ldots$ (though the order of containment may change). Moreover, at the $t$-th moment, the level 2 computation is complete, and the net is stable. That is, repeating the level 1 computation does not change the state of the net. Therefore $S_{it}$ and $S_{ht}$ are comparable in all coordinates.

Suppose two states, $S_{it}$ and $S_{ht}$, are incomparable in every coordinate. Then each coordinate of $S_{it}$ must be $X_2$ or $X_3$ with each coordinate of $S_{ht}$ being the other. By Lemma 3.8 the map induced on the $j$-th coordinate of $S_{it}$ or $S_{ht}$ must be a constant map which when restricted to $N_j$ is $C_{X_2}$ or $C_{X_3}$. Let $S_{mt}$ be a third state. Then the $j$-th coordinate of $S_{mt}$ is comparable to the $j$-th coordinate of either $S_{it}$ or $S_{ht}$. Suppose it is $S_{it}$. By lemma 3.8, then the $(j+1)$-st, $(j+2)$-nd, ... coordinates will be comparable with the containment in the same order contradicting Lemma 3.10. Therefore, if $p>2$, each pair of states, $S_{it}$ and $S_{ht}$, are comparable in every coordinate, and this must be true for each moment.

Now suppose for some $k,j$, $|\{N_j\cap S_{ik}: i=1,\ldots,p\}|=3$. Then the elements of this set must be $X_0$, $X_1$, and $X_2$ or $X_0$, $X_1$, and $X_3$ because they are comparable. Suppose in the previous moment (mod $t$) and the previous coordinate (mod $m$) only two elements occur. Since the union of the ranges of the maps induced by these elements has three elements, then by Lemma 3.4 one must induce the identity map. But by Lemma 3.8 and the fact that these two elements are comparable, one induces the identity and the other is $C_{X_0}$ or $C_{X_1}$. In either case the ranges of the two maps contain only two elements. Therefore at moment 0 all coordinates must contain three elements, $X_0, X_1, X=X_2$ or $X_3$. Moreover the maps induced by these in the next moment must be $C_{X_0}, C_{X_1}, C_Y$, $Y=X_2$ or $X_3$ respectively. But if this is the case there are two states, one of which is $X_0$ in each coordinate, the other of which is

$X_1$ in each coordinate contradicting Lemma 3.10. □

By the previous lemmas the states $S_{ik}$ may be represented faithfully by the following sequences. For each $i=1,p$ and $k=0,\dots,t$ let $T_{ik}=(t_{ik1},t_{ik2},\dots,t_{ikm})$ where if the set of $j$-th coordinates at moment $k$ contains only one element then $t_{ikj}=c$ and if the set of $j$-th coordinates contains two elements then $t_{ikj}=1$ if the $j$-th coordinate of $S_{ik}$ is the greater of these two and $t_{ikj}=0$ otherwise. Moreover, by Lemma 3.10, it is not the case that one sequence is greater than or equal to another of those at the same moment in every coordinate. Therefore every sequence must contain at least one 1 and at least one 0.

Consider the sequence, $T_{ik}$, as written in a circle so that $t_{ik1}$ follows $t_{ikm}$. Then this circle can be divided into *segments* where each segment consists entirely of 1's and $c$'s or 0's and $c$'s. Pictorially, for each $T_{ik}$, we have a figure as in Figure 7.

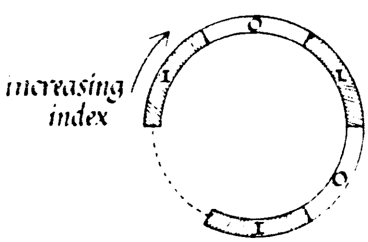



Fig. 7

The shaded areas represent segments of 1's and $c$'s and the unshaded areas represent segments of 0's and $c$'s. It will be shown that the number of segments are the same for all $T_{ik}$, that at each moment the segments merely 'rotate' about the ring, and that the segments of one $T_{ik}$ cannot rotate 'faster' than the segments of another. This will force the group $G_Q$ to be a solvable group.

Let $m_{ik}^0$ be the first coordinate of $T_{ik}$ where a 0 occurs. let $m_{ik}^1$ be such that the $m_{ik}^1+1$ coordinate of $T_{ik}$ is 1, but the $m_{ik}^0, m_{ik}^0+1,,\dots,m_{ik}^1$ coordinates of $T_{ik}$ are 0 or $c$. (Here addition is taken mod $m$.) $m_{ik}^1$ is the last coordinate to the right of $m_{ik}^0$ not equal to 1. Let $m_{ik}^2$ be the last coordinate to the right of $m_{ik}^1$ not equal to 0. Define $m_{ik}^3,\dots,m_{ik}^{f_{ik}}$ in this manner until the cycle is exhausted. Note that $m_{ik}^0$ may have been in the middle of a segment but that $m_{ik}^1,\dots,m_{ik}^{f_{ik}}$ mark the right-hand end of consecutive segments where the odd superscripted ones begin with a 0 and the even with a 1.

The sequences $M_{ik}=m_{ik}^1,\dots,m_{ik}^{f_{ik}}$ faithfully represent the states $S_{ik}$.

**3.12. Lemma.** *For each* $k=0,\dots,t$, $M_{ik}\neq M_{hk}$ *if* $i\neq h$.

**Proof.** Since the coordinates of $T_{ik}=c$ exactly where the coordinates of $T_{hk}=c$ then if $M_{ik}=M_{hk}$ then $T_{ik}=1$ where $T_{hk}=1$ and $T_{ik}=0$ where $T_{hk}=0$. Therefore if $M_{ik}=M_{hk}$ then $T_{ik}=T_{hk}$ and $P_{ik}=P_{hk}$ contradicting Lemma 3.9. □

It will be shown that the number of segments in $T_{ik}$ cannot increase in $T_{ik+1}$ for each $k$, and hence can never decrease since repeating the permutation several times will yield the identity map. This will be true for each permutation in $G_Q$, and since $G_Q$ is transitive the number of terms in $M_{ik}$ is the same for all $i, k$.

**3.13. Lemma.** *For each* $i = 1, \dots, p$ *and each* $k = 0, \dots, t-1$, $f_{ik} \geq f_{ik+1}$.

**Proof.** Consider a segment $t_{ikj} = 1$, $t_{ikj+1} = 1$ or $c, \dots, t_{ikj+r} = 1$ or $c$ of $T_{ik}$ where $t_{ikj+r+1} = 0$. In the next moment none of $t_{ik+1s}$ can equal 0 for $j+1 \leq s \leq j+r$. For suppose $t_{iks-1} = 1$. Then the $(s-1)$-st coordinate of $P_{ik}$ is the greater of the two possible states occurring among the $(s-1)$-st coordinates and so by Lemma 3.8 the $s$-th coordinate of $P_{ik+1}$ must be 1 or $c$. Similarly if $t_{iks-1} = c$ then $t_{ik+1s}$ could $= 0$ only if the input into $Q_s$ induces the identity map in which case $t_{ik+1s} \neq 0$ since $t_{iks} \neq 0$. Therefore a segment of 1's and $c$'s cannot create a 0 in its interior. Similarly for a segment of 0's and $c$'s. A segment can be completely lost if the tail of the segment immediately preceding it joins the head of the segment immediately after it in the next moment, but a new segment can never be created. □

Let $f$ = the number of terms in $M_{ik}$ for each $i$ and $k$. Since the number of segments remains fixed, the segments can be pictured as following each other around the circle.

The segments of $T_{ik}$ become the segments of $T_{ik+1}$ in the obvious way. Renumber the superscripts for $m^e_{ik+1}$, $e = 1, \dots, f$, $k = 0, \dots, t-1$ so that $m^e_{ik+1}$ marks the right-hand end of the segment that the $e$-th segment of $T_{ik}$ becomes at moment $k+1$. It should be noted here that it is not necessarily the case that if $S_{it} = S_{h0}$ then $m^e_{it} = m^e_{h0}$. For example, it is possible for a sequence (i.e., a state) to be mapped to itself without each of its segments being mapped to itself.

The segments of one sequence will never travel around their circle faster than the segments of another sequence.

**3.14. Lemma.** *For some moment* $k$ *suppose the right-hand end of a segment in* $T_{ik}$ *beginning with* $b = 0$ *or* 1 *agrees with the right-hand end of a segment in* $T_{hk}$ *beginning with* $b$. *Then these right-hand ends agree at moment* $k+1$.

**Proof.** Consider a segment in $T_{ik}$ beginning with a 1. The value of the coordinate immediately to the right of the segment is a 0. Suppose a segment in $T_{hk}$ beginning with a 1 ends in the same coordinate as this segment. If both segments end in $c$ and in the next moment $c$ induces the identity then the right-hand end of both segments doesn't change in the next moment. If the $c$ induces a constant map then both segments are extended to the right where they may join a string of $c$'s. By Lemma 3.13 this string of $c$'s cannot completely 'cover' the next segment. The coordinate immediately to the right of the string must equal 0 and the end of the string will be the right-hand end of both segments. If both segments end in a 1 and the 1 induces the identity map the right-hand ends stay fixed. If the 1's induce a constant map then the right-hand end will be extended by a chain of $c$'s or a 1 followed by a chain of $c$'s, but again will have the same right-hand end. The result is similar if the segment begins with a 0. □

**3.15. Lemma.** *Consider two sequences $T_{ik}$ and $T_{hk}$ at moment $k$. Consider the r-th segment of $T_{ik}$ beginning with a b. Suppose the s-th segment of $T_{hk}$ is such that it begins with a b, and there is no other segment of $T_{hk}$ beginning with b between the right-hand end of the s-th segment and the right-hand end of the r-th segment of $T_{ik}$. Then this relation will hold between these segments at moment $k+1$. That is, if r and s are both even or both odd, and there exists a positive integer $d<m$ such that $m_{hk}^s+d\equiv m_{ik}^r \pmod m$ and $m_{hk}^s+c\not\equiv m_{hk}^{s+2(\mathrm{mod}\, f)} \pmod m$ for $c=1,2,\dots,d-1$, then there exists a positive integer e such that $m_{hk+1}^s+e\equiv m_{ik+1}^r \pmod m$ and $m_{hk+1}^s+c\not\equiv m_{hk+1}^{s+2(\mathrm{mod}\, f)} \pmod m$ for $c=1,2,\dots,e-1$.*

**Proof.** Suppose $b=1$, the other case being similar. As in the proof of (3.14) the right-hand ends of the $r$-th segment of $T_{ik}$ and the $s$-th segment of $T_{hk}$ can only be extended by a string of $c$'s or a 1 followed by a string of $c$'s at moment $k+1$. The end of the $s$-th segment of $T_{hk}$ will remain to the left of the $r$-th segment unless the string of $c$'s adjoining the former reaches the string of $c$'s adjoining the latter in which case they will have the same right-hand end, that is, $m_{ik+1}^r\equiv m_{hk+1}^s$.

Consider the pairs

$$(m_{ik}^r, m_{hk}^s), (m_{ik}^{r+2(\mathrm{mod}\, f)}, m_{hk}^{s+2(\mathrm{mod}\, f)}), \dots, (m_{ik}^{r+f-2(\mathrm{mod}\, f)}, m_{hk}^{s+f-2(\mathrm{mod}\, f)}).$$

By Lemma 3.14 if the members of one of these pairs are equal at moment $k$ then they will be equal at each subsequent moment. Suppose $q$ of these pairs are equal. Since the input sequence $w$ may be repeated to induce the identity permutation, it is clear that the number, $q$, must remain fixed at each moment. However, if $m_{ik+1}^r\neq m_{hk+1}^s \pmod m$ then $q$ is increased by at least 1. □

Consider the set of right-hand ends of all segments beginning with a 1, of all sequences, $T_{i0}, i=1,\dots,p$, and the set of right-hand ends of all segments beginning with a 0. That is, let $M^1=\{m_{i0}^r$: $r$ is even, $i=1,\dots,p\}$ and $M^0=\{m_{i0}^r$: $r$ is odd, $i=1,\dots,p\}$. Input $w$ induces two maps $f_w^1: M^1\to M^1$ and $f_w^0: M^0\to M^0$ defined by the following. For $b=1$ or $0, i=1,\dots,p$, $r\equiv b+1 \pmod 2$, $f_w^b(m_{i0}^r)=m_{it}^r$. By Lemma 3.14 these maps are well defined, and since $f_w$ is a permutation then so are $f_w^0$ and $f_w^1$.

Let $\hat{M}=M^0\times M^1$. Let $w_1$ be any input sequence inducing the permutation $f_{w_1}$ in $G_Q$. Then $f_{w_1}^b, b=0,1$, may be defined in a similar way as $f_w^b$ was defined. (The total number of moments $t$ may be different.) Then $w_1$ induces the permutation $\hat{f}_{w_1}=f_{w_1}^0\times f_{w_1}^1$ on $\hat{M}$. Let $\hat{G_Q}=\{\hat{f}_{w_1}$: $w_1$ induces $f_{w_1}$ in $G_Q\}$.

**3.16. Lemma.** *$G_Q$ is a homomorphic image of $\hat{G_Q}$.*

**Proof.** Let $h: G_Q^{\wedge} \to G_Q$ be the map $h(f_{w_1}^{\wedge}) = f_{w_1}$. Clearly $f_{w_1}^{\wedge} f_{w_2}^{\wedge} = f_{w_1 w_2}^{\wedge}$, and so $h$ is a homomorphism. □

Finally, we have the results for rank 2 nets.

**3.17. Theorem.** *Let* $L \in \mathrm{CFN}_2$. *Then* $L^S \in W(U_0, U_1, U_2, U_3, Z_p$: *p is prime*).

**Proof.** It is sufficient to show $G_Q^{\wedge}$ is solvable by lemma 3.16, Theorems 1.12 and 1.15. Suppose $w$ induces $f_w^{\wedge} \in G_Q^{\wedge}$. For each $y$ in $M^b$ ($b = 0, 1$), let the immediate successor of $y$ be the element $z \in M^b$, $z \neq y$, such that there does not exist $z_1 \in M^b$ and positive integers $c, d$ such that $y + c \equiv z_1 \pmod m$, $z_1 + d \equiv z \pmod m$, and $c + d$ is the least positive integer such that $y + c + d \equiv z \pmod m$. let $\pi_b$ be the permutation on $M^b$ defined by $\pi_b(y) =$ immediate successor of $y$. Clearly $\pi_b$ is a single orbit permutation on $M^b$. By Lemma 3.15 $f_w^b$ commutes with $\pi_b$. Suppose for some $y, y_1 \in M^b$, $f_w^b(y) = y_1$. Since $\pi_b$ has only one orbit then $y_1 = \pi_b^s(y)$ for some $s$. Then for each $j \geq 0$, $f_w^b(\pi_b^j(y)) = \pi_b^j(f_w^b(y)) = \pi_b^j \pi_b^s(y) = \pi_b^s(\pi_b^j(y))$ and so $f_w^b = \pi_b^s$. Therefore $G_Q \subseteq \langle \pi_0 \rangle \times \langle \pi_1 \rangle$ and the result follows. □

A more complicated argument can be used to show that $G_Q^{\wedge}$ is, in fact, cyclic. This will not be presented here.

At the next rank it is possible to obtain unsolvable subgroups. In fact it can be shown that the symmetric groups, $S_n$, can be obtained for all $n$, and hence every group appears as a subgroup. However the following construction yields certain unsolvable groups in a natural way, consistent with the previous constructions.

For $n > 2$ and $p$ prime consider the $n \times n$ matrices over $Z_p$, $E_{ij}$, whose every component is 0 except for the $i, j$ component which equals 1. Let $I_n$ be the $n \times n$ identity matrix. Consider the set $\mathcal{M} = \{M_{12}, M_{23}, \ldots, M_{n-1n}, M_{n1}\}$ where $M_{ij} = I_n + E_{ij}$, let $V =$ the vector space $Z_p^n$ written as row vectors, and let $G$ be the group generated by the elements of $\mathcal{M}$. Since for $j \neq i-1$, $M_{ij} M_{jj+1} M_{ij}^{-1} M_{jj+1}^{-1} = M_{ij+1}$ (where the addition to $j$ is taken mod $n$) then for $j \neq i$ it is clear that $M_{ij} \in G$. It is well known (see [3]) that $\{M_{ij} : i \neq j\}$ generate $\mathrm{SL}_n(Z_p)$.

The effect of multiplying a vector, $v \in V$, by a matrix $M_{ii+1}$ is the addition of the $i$-th component of $v$ to the $(i+1)$-st component. Let $\mathcal{M}_1 = \mathcal{M} - \{M_{n1}\}$, and let $G_1 = \langle \mathcal{M}_1 \rangle$. Then the effect of multiplying a vector by an element in $G_1$ is a series of additions of lower components to higher components. The inclusion of $M_{n1}$ in the set of generators for $G$ necessitates some 'feedback'. A machine modelling this feedback may be constructed in $\mathrm{CFN}_3$.

**3.18. Theorem.** *For each* $n \geq 2$ *there exists a net,* $L \in \mathrm{CFN}_3$, *such that* $L^S$ *contains* $\mathrm{PSL}_n(Z_2)$.

**Proof.** Consider the net in Figure 8.

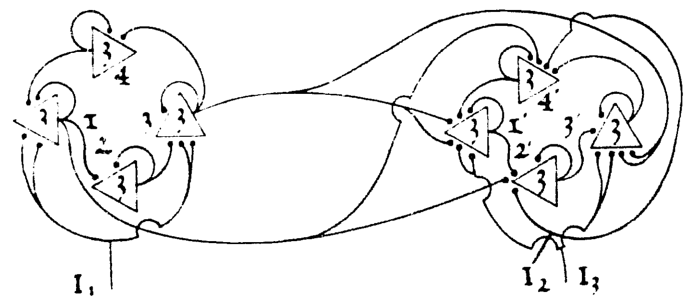


Fig. 8

There are two rank 2 nets, $L_1$ containing neurons $\{1,2,3,4\}$, and $L_1'$ containing neurons $\{1',2',3',4'\}$, and some axons from $L_1$ to $L_1'$. This net is a series composition of $L_1$ and $L_1'$.

Let $I$ ($I'$) be the state of $L_1$ ($L_1'$) defined by neuron 1 (1′) on and neurons 2, 3, 4 (2′, 3′, 4′) off, and let 0 (0′) be the state defined by neuron 3 (3′) on and neurons 1, 2, 4 (1′, 2′, 4′) off. Let $a$ be the input $a=\{I_1,I_2\}$, and $b$ be the input $b=\{I_1,I_3\}$. Note that input $I_1$ fixes both $I$ and 0. The permutation on the set $\{I,0\}\times\{I',0'\}$ induced by the input sequence $ab$ is $f_{ab}=((0,0'))((0,I'))((I,0')(I,I'))$. If $I$ ($I'$) and 0 (0′) are considered as 1 and $0\in Z_2$ then $f_{ab}$ adds (mod 2) the 'value' of $L_1$ to $L_1'$. In this way $f_{ab}$ represents the matrix $M_{ii+1}$ from the discussion preceding the theorem. Clearly a rank 3 net, $L$, such that $L^S$ contains $\mathrm{PSL}_n(Z_2)$ can be constructed by linking $n$ copies of the subnet $L_1$ in a circle along with the appropriate input axons, where the axons between consecutive elements of the circle are as in Figure 8. □

**3.19. Corollary.** $\mathrm{CFN}_2^S \subsetneq \mathrm{CFN}_3^S$. □

Since $\mathrm{PSL}_n(Z_2)$ contains the symmetric group, $S_n$, as a subgroup, it is possible to obtain each finite group as a subgroup of $L^S$ for some rank 3 net, $L$. That is, it is possible to find input sequences which induce permutations which generate the desired group. But these input sequences may be arbitrarily long. A more difficult task is to construct a net whose semigroup contains the desired group for which the input sequences inducing the generators of the group are as short as possible. By Note 3 after Definition 2.3 the shortest sequence that can induce a non-trivial permutation on states has length 2.

**3.20. Definition.** A group, $G$, is *length $k$ realizable at rank $r$* if there exists a rank $r$ net, and a set of input sequences $W=\{w_1,w_2,\ldots,w_m\}$ such that $|w_i|\le k$, $i=1,\ldots,m$, and $\langle f_{w_1},\ldots,f_{w_m}\rangle\cong G$.

It should be noted that $\mathrm{PSL}_n(Z_2)$ is length 2 realizable at rank 3.

**3.21. Theorem.** *For each $m\ge 3$ the commutator subgroup of $O_{2m}^+(2)$ is length 2 realizable at rank 4.*

**Proof.** Let $V$ be the vector space $V \cong Z_2^{2m}$ and number the dimensions of $V$ $1, 2, 3, \ldots, m, \hat{1}, \hat{2}, \ldots, \hat{m}$. Let $M_{i,j,k,l}$ be the matrix $M_{i,j,k,l} = M_{i,j}M_{k,l}$. Let $G$ be the commutator subgroup of $O_{2m}^{+}(2)$. It is known (see [3]) that $G$ is generated by the set

$$\mathcal{M} = \{M_{i,j,\hat{j},\hat{i}}, M_{j,i,\hat{i},\hat{j}}, M_{i,\hat{j},j,\hat{i}}, M_{\hat{i},j,\hat{j},i} : 0 < i < j \leq m\}.$$

Let $\mathcal{M}_1$ be the set

$$\mathcal{M}_1 = \{M_{1,2,\hat{2},\hat{1}}, M_{2,3,\hat{3},\hat{2}}, \ldots, M_{n-1,n,\hat{n},\widehat{n-1}}, M_{n,1,\hat{1},\hat{n}}\} \cup \{M_{1,\hat{2},2,\hat{1}}, M_{\hat{1},2,\hat{2},1}\}.$$

Since

$$M_{i,j,\hat{j},\hat{i}} M_{j,j+1,\widehat{j+1},\hat{j}} M_{i,j,\hat{j},\hat{i}} M_{j,j+1,\widehat{j+1},\hat{j}} = M_{i,j+1,\widehat{j+1},\hat{i}}$$

for $j \neq i-1$ then $M_{i,j,\hat{j},\hat{i}} \in \langle \mathcal{M}_1 \rangle$ for all $i \neq j$. Moreover since if $i < j$, $j \neq 2$,

$$M_{j,2,\hat{2},\hat{j}} M_{1,\hat{2},2,\hat{1}} M_{j,2,\hat{2},\hat{j}} M_{i,1,\hat{1},\hat{i}} M_{j,2,\hat{2},\hat{j}} M_{1,\hat{2},2,\hat{1}}$$

$$M_{j,2,\hat{2},\hat{j}} M_{i,1,\hat{1},\hat{i}} M_{1,\hat{2},2,\hat{1}} M_{i,1,\hat{1},\hat{i}} M_{1,\hat{2},2,\hat{1}} M_{i,1,\hat{1},\hat{i}} = M_{i,\hat{j},j,\hat{i}} \in \langle \mathcal{M}_1 \rangle$$

and similarly $M_{\hat{i},j,\hat{j},i} \in \langle \mathcal{M}_1 \rangle$ then $\langle \mathcal{M}_1 \rangle = G$.

Let $L$ and $\hat{L}$ be two copies of the net constructed in the proof of Theorem 3.18 where the copies of the subnet $L_1$, in $L$ are numbered $1, 2, 3, \ldots, m$ and the copies in $\hat{L}$ are numbered $\hat{1}, \hat{2}, \ldots, \hat{m}$, and the links between these subnets in $L$ go from the $i$-th subnet to the $(i+1)$-st subnet (mod $m$) and the links in $\hat{L}$ go from the $\hat{i}$-th subnet to the $(\widehat{i-1})$-st subnet (mod $m$) for each $i$. Let $L_4$ be a net constructed from $L$ and $\hat{L}$ with axons from the subnets 1 and 2 of $L$ (subnets $\hat{1}$ and $\hat{2}$ of $\hat{L}$) to the subnets $\hat{1}$ and $\hat{2}$ of $\hat{L}$ (subnets 1 and 2 of $L$). Clearly with the appropriate input axons and connecting axons, $L_4$ can be constructed so that $L_4^S \cong G$. Pictorially $L_4$ is shown in Figure 9. Each $Q_j$ ($Q_{\hat{j}}$) is a copy of $L_1$. Since there are axons from only two $Q_j$ ($Q_{\hat{j}}$) $L_4$ is 2-linked, and $L_4 \in \mathrm{CFN}_4$. □

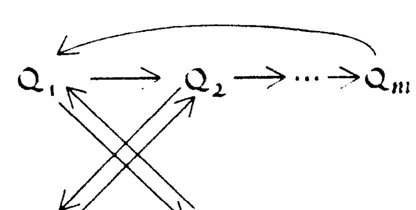



Fig. 9

## 4. Comments

In this section the necessity of the restrictions placed on nets for membership in CFN are discussed, and some comments are made on motivation.

(i) *Local stability.* The concept of local stability, though not formally defined, appears in [2]. There the author calls the stable states 'real' and the intermediate states 'imaginary' and conjectures that the introduction of imaginary values to logic has great power. He also constructs a rank 2 net which contains $Z_2$ as a subgroup of its associated semigroup.

The following procedure formally defines the next-state function, $\delta: 2^N \times 2^I \to 2^N$, for a machine interpretation of a locally stable net, $L=(N, A, I, t)$ with partition sequence $P_1, P_2, \ldots, P_r$. It has been noted that the level $i$ computation may take several intermediate steps before stabilizing. Call these steps *instants*, indexed by the variable $u$ which is initially 0.

For each $n \in N$ let $B^n \subseteq A$ be the set of axons beginning at $n$ and let $E^n \subseteq A \cup I$ be the axons ending at $n$. Let $N^u \subseteq N$ be the set of neurons 'on' at instant $u$. For each $S \subseteq A \cup I$ let $S^u$ = set of axons in $S$ which carry a pulse at instant $u$. Let $A_j$, $j=1,\ldots,r+1$, be defined from $P_j$, $j=1,\ldots,r$, as before. Let $\bar{A}_j = A \cup I \sim A_j$. Let $B_j^n = B^n \cap A_j$.

Suppose the net is in state $S$ and $J$ is input. Consider the following recursive procedure.

*Input:* $i$ = level of computation; $u$ = beginning instant
*Output:* $u'$ = instant of completion; $N^{u'}$ = state of machine at completion
*Procedure:* LEVEL COMP $(i, u)$
Step 1: $v \leftarrow u$
Step 2: if $i \neq 1$ then go to step 3
else $N^{v+1} \leftarrow \{n \in N: |E^n \cap (A \cup I)^v| \geq t(n)\}$
$(A \cup I)^{v+1} \leftarrow \bar{A}_1^v \cup \bigcup_{n \in N^{v+1}} B_1^n$
$v \leftarrow v+1$
if $A_1^v \neq A_1^{v-1}$ then go to step 2
else $(A \cup I)^v = \bar{A}_2^v \cup \bigcup_{n \in N^v} B_2^n$
$u' \leftarrow v$
return
Step 3: do LEVEL COMP $(i-1, v)$
if $A_i^{u'} \neq A_i^v$ then $v \leftarrow u'$ go to step 3
else $(A \cup I)^{u'} = \bar{A}_{i+1}^v \cup \bigcup_{n \in N^{u'}} B_{i+1}^n$
return

To compute $\delta(S, J)$, let $u=0$, let $N^0 = S$, let $(A \cup I)^0 = \bigcup_{n \in S} B^n \cup J$ and do LEVEL COMP $(r, 0)$. When it returns, $\delta(S, J) = N^{u'}$.

(ii) *Inhibition-free.* The proofs of Lemmas 3.4 and 3.8 and subsequent theorems depend on the nets being inhibition-free. In particular Theorem 3.17 (the semigroup of a rank 2 net is solvable) is no longer true if inhibition is allowed. In fact, given a permutation on $n$ letters, it is possible to construct a rank 2 net with inhibition such that some input sequence induces the permutation, though it is not clear that this can be done if the length of the sequence is restricted to be $\leq 2$.

(iii) *2-linked*. Though 2-linked nets appear naturally at ranks 3 and 4, it is not clear that 2 is a necessary upperbound on the number of links. It can be shown that there exists a 5-linked rank 2 net whose associated semigroup contains $S_5$. Therefore if Theorem 3.17 is to hold the nets must be at most 4-linked. However lemma 3.11 depends heavily on 2-linkedness, and it is not clear that an equivalent lemma can be proven with 3 or more links.

(iv) *Circular feedback*. It is clear that the $i$-th level graphs, $G_i$, $i = 1, \ldots, r$, each must contain a closed path for there to be a distinction in the associated semigroup of nets of different rank. Moreover it can be shown that if a connected component of $G_i$ is allowed more than a simple circuit than it is possible to realize any group at rank 2.

## References

[1] M.A. Arib, Algebraic Theory of Machines, Languages and Semigroups (Academic Press, New York, 1968).
[2] G.S. Brown, Laws of Form (Julian Press, New York, 1972).
[3] R.W. Carter, Simple Groups of Lie Type (John Wiley and Sons, New York, 1972).
[4] R. McNaughton and S. Papert, Counter-Free Automata (MIT Press, Cambridge, 1971).