INFORMATION AND CONTROL **49**, 239–264 (1981)

Threshold Nets and Cell-Assemblies*

A. PISTORELLO

Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Italy

C. ROMOLI

Istituto di Scienze dell'Informazione, Universita di Pisa, Italy

AND

S. CRESPI-REGHIZZI

Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Italy

Motivated by the cell-assemblies theory of the brain, we propose a new formal model of threshold nets (TN). TN are patterned after Petri nets, with a very different firing rule, which removes all tokens upon firing of a transition. The generative power of threshold nets, with and without inhibition, is compared with traditional families of languages. Excitatory TN languages are included by the noncounting regular languages and form an infinite hierarchy for increasing values of threshold. Inhibitory nets are included by the context-sensitive languages. Two new net operators, motivated by the phenomena of growth, learning and brain damage are introduced and compared with Boolean operators.

INTRODUCTION

A great number of theoretical studies aimed at understanding nervous systems have been developed recently. Our work tries to outline a formal approach to Hebb's theory (Hebb, 1949) on cerebral organization, recently reformulated by Braitenberg (Braitenberg, 1973, 1974, 1978), which proposes the "cell-assembly" as the significant unit of mental processing.

Roughly, a cell-assembly is a set of neurons so strictly interconnected by excitatory synapses, that for a particular pattern of external stimuli it reaches a high level of excitation, and maintains it. In addition a mechanism is postulated to control activity of a cell-assembly through changes in threshold of neurons. Simple cell-assemblies can be connected with each other to form more complex ones.

Braitenberg conjectures that each unitary mental act (perception or

---

* Work supported by C.N.R.

0019-9958/81/060239–26$02.00/0

abstract concept) is related to the activity of a cell-assembly, which is triggered by stimulation, from sense organs or other cell-assemblies.

The activity of cell-assemblies can be turned off by inhibition either localized or spread over large areas of the cortex.

As a formal model of a cell-assembly we propose a place-transition net, similar to a Petri net (Petri, 1962), consisting of the elements depicted in Fig. 1.

Tokens, which represent elementary amounts of stimulation, come into places from afferent transitions, and remain there until the transition is enabled and fires (firing models the spike of a neuron).

A transition is enabled when the number of input tokens equals or exceeds its threshold; firing consists of empting input places and sending a token along each output arc to efferent places.

Although modelled after Petri nets, our nets have undergone substantial changes to conform to nervous net behavior:

— Transition enabling requires the overall presence in the input of a sufficient number of tokens, no matter how they are distributed in the places.

— Upon firing of a transition *all* input places are emptied.

— A place cannot be input to more than one transition.

In spite of these differences we have found that the formal approach used for analyzing firing sequences in Petri nets (e.g., Crespi-Reghizzi, 1976; Hack, 1975) can be useful also for threshold nets. Among the numerous formal models proposed for neuronal activity, the best known is the study by McCulloch and Pitts (McCulloch and Pitts, 1943).

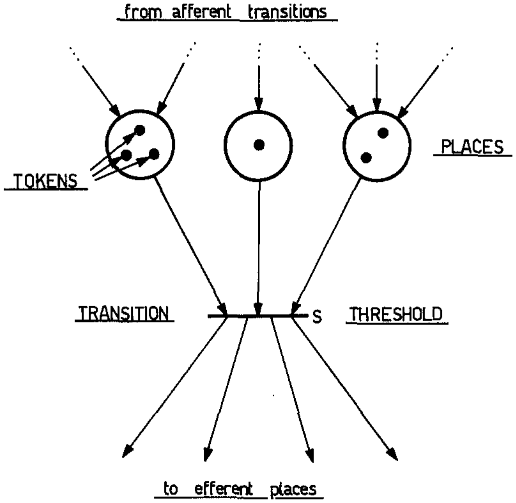

FIG. 1. Elements of a threshold net.

In their "nerve nets," firing of "cells" and transmission of impulses are strictly synchronous, parallel firings are allowed, and cells are not able to keep memory of pulses arriving at different times.

Different approaches between nerve nets (NN) and our threshold nets (TN) result in opposite formal characteristics: while NN are synchronous, deterministic and parallel automata; TN are, like Petri nets, asynchronous, serial and nondeterministic.

We think that a realistic temporal analysis of neuronal nets is far more complex than the one proposed by McCulloch and Pitts, and could hardly fit in a formal analysis.

Thus we have preferred to avoid the problems of precise timing and synchronization: this is accomplished by considering as plausible firing sequences all sequences which may occur for any firing delays of enabled transitions. As firing delays are unspecified our model is clearly nondeterministic.

Following Braitenberg's approach, in the first part of our work we consider all excitatory nets, convinced that this simplification may still lead to useful insight.

In the last chapter we shall consider nets with inhibitory connections too, and show the resulting increase in expressive power.

The paper is organized as follows:

- — Definition of threshold nets.
- — Study of main characteristics of firing sequences languages.
- — Definition of new operators for TN.
- — Study of the effects of changes in threshold and connections.
- — Inhibitory nets.
- — Concluding remarks.

Some uninteresting formal proofs, which are just outlined here, can be found in the thesis (Pistorello and Romoli, 1980), which also contains some developments not included here.

Basic knowledge of formal language theory is occasionally assumed; the reader is referred to any standard textbook (e.g., Hopcroft and Ullman, 1969).

1. DEFINITIONS

DEFINITION 1. An excitatory threshold net (ETN) is a system made of the following five components:

P = a finite set of *places*;

T = a finite set of *transitions*;
P and T are disjoint sets;

I = *input function*, is a function mapping each transition t onto the set of *input* places $I(t)$;
the *conflict-free* condition holds:
$\forall t, t' \in T, I(t) \cap I(t') = \varnothing$;

O = *output function*, maps each transition t into the multiset[1] or bag $O(t)$ of output places;

S = the *threshold*, maps each transition t to a nonnegative integer $S(t)$.

The state of an ETN is described by a *marking* M which maps each place p to a nonnegative integer. M can be extended to a set of places as follows:

$$M(\{p_1,\ldots,p_n\}) = \sum_{i=1}^{n} M(p_i).$$

It is convenient to visualize an ETN by a graphical representation like the one used for Petri nets (Fig. 2), where places are represented by circles and transitions by bars.

There is an arc from each place in $I(t)$ to t.

Similarly from t to each place p in $O(t)$ there are $\#(p, O(t))$ arcs.

Thresholds are written near each transition.

The marking which assigns m to a place p, is represented by m dots or *tokens* inside the node p.

A transition t is *enabled* if $M(I(t)) \geqslant S(t)$.

An enabled transition t can *fire* generating the new marking M' such that, for each p,

$$M'(p) = \mathit{if}\ p \in I(t) \qquad \mathit{then}\ \#(p, O(t))$$
$$\mathit{else}\ M(p) + \#(p, O(t)).$$

The *firing function* $f(M, t) = M'$ if t is enabled in M, otherwise is undefined.

In this model the firing of a transition t first clears the input places $I(t)$; then each output place p receives as many tokens as there are arcs from t to p.

In Fig. 2 only t_2 is enabled; when t_2 fires it generates the new marking $M'(p_1) = 1, M'(p_2) = 2, M'(p_3) = 1$, which can be written as $M' = (1, 2, 1)$.

---

[1] A multiset or bag is a set which allows repetitions of its elements. The number of occurrences of an element e_i in the bag B is denoted by $\#(e_i, B)$. Similarly we denote by $\#(t, v)$ the number of occurrences of an element t in the string v.

The T.N. (P, T, I, O, S) where:

$P = \{p_1, p_2, p_3\}$;
$T = \{t_1, t_2\}$;
$I(t_1) = \{p_2, p_3\}$; $I(t_2) = \{p_1\}$
$O(t_1) = \{p_1^2\}$; $O(t_2) = \{p_1, p_3\}$
$S(t_1) = 3$; $S(t_2) = 2$
with marking:
$M_0(p_1) = 3$; $M_0(p_2) = 2$; $M_0(p_3) = 0$
can be rappresented as:

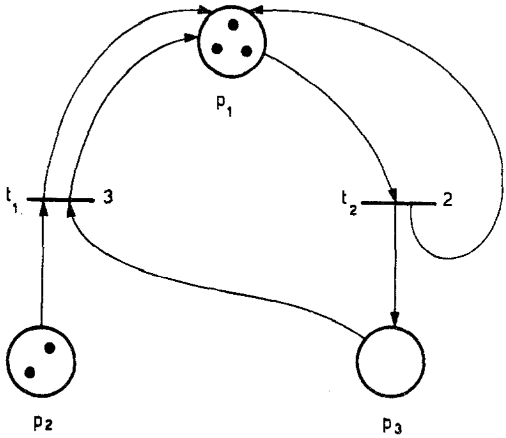

FIG. 2. An example of an ETN.

Concerning time, we make the following assumptions:

— An enabled transition will fire after a finite, unspecified time.
— Firing is instantaneous.
— No two transitions may fire at the same time.

The behaviour of an ETN is characterized by the firing sequences. In the net of Fig. 2 t_2 fires, then t_1, then again t_2. Thus $t_2 t_1 t_2$ is a firing sequence.

Let M_0 be the *initial* marking of ETN, and t_0 a transition enabled by M_0. Firing t_0 leads to M_1 which again enables t_1, etc. Then we say that

$$t_0 t_1 \cdots$$

is a firing sequence of ETN with marking M_0.

We can extend the function f to a firing sequence $t_0 t_1 \cdots t_n$ as follows:

$$f(M_0, t_0 t_1 \cdots t_n) = f(f \cdots (f(M_0, t_0), t_1) \cdots t_n).$$

The *set of firing sequences* of an ETN with marking M_0 is

$$L = \{t_0 t_1 \cdots t_n \in T^* \mid f(M_0, t_0 \cdots t_n) \text{ is defined}\}.$$

For simplicity, without loss of generality, we shall occasionally assume that $M_0 \neq 0$ only in a single place p_{t_0}, such that for no $t \in T$, $p_{t_0} \in O(t)$. We denote this as "*single-source* hypothesis."

Noticing that redistribution of tokens between input places of a transition does not affect its enabling (thanks to the conflict free hypothesis), we can merge all input places of any transition t in a single place p_t, with no effect on firing sequences.

Formally, this is stated by:

Statement 2. The nets $N_1 = ((P, T, I, O, S), M_0)$ and $N_2 = ((P', T, I', O', S), M'_0)$, where

$$T = \{t_1, \ldots, t_m\};$$

$$P' = \{p_{t_1}, \ldots, p_{t_m}\};$$

$$\forall t \in T, \qquad I'(t) = \{p_t\};$$

$$\forall t_i, t_j \in T, \#(p_{t_i}, O'(t_j)) = \sum_{p \in I(t_i)} \#(p, O(t_j));$$

$$\forall t \in T, M'_0(p_t) = M_0(I(t));$$

generate the same set of firing sequences. ▮

Therefore from now on we will always deal with nets in which each transition t has a single input place p_t without loss of generality.

At this early point the reader should already realize that ETN behavior is very different from that of Petri nets.

We will show in Section 3 that ETN belong to the class of finite state automata, while it is known that Petri nets are more powerful (Hack, 1975).

2. LANGUAGES GENERATED BY ETN

Firing sequences of an ETN can be considered as strings over the alphabet T.

DEFINITION 3. The *language L generated* by an ETN is

$$L = \{w \in T^* \mid f(M_0, w) \text{ is defined}\}.$$

According to (Braitenberg, 1973) a subset of a neuronal network is a cell-assembly when the excitation of its elements is maintained without the contribute of external stimuli.

Accordingly, in our model a cell-assembly is a subnet which, with a particular initial marking and after severing all connections with the rest of the network, generates infinitely long firing sequences.

A *prefix* of a string x is any string y such that $x = yz$ (x, y, z can be null).

If a string x is in L, then all prefixes of x are in L, too, since if $f(M_0, x)$ is defined, then so is $f(M_0, y)$. Therefore for convenience we shall assume that, for any language L that we define, *L stands for the union of all prefixes* of the strings of L.

Next we argue that an ETN is equivalent to a finite state machine, hence the language L is regular (or type 3).

Two markings M and M' of an ETN are *undistinguishable* if they generate the same firing sequences.

We argue that a sufficient condition for two markings M and M' to be undistinguishable is:

$$\begin{aligned} \forall t \in T, \quad & [M(p_t) = M'(p_t)] \\ \text{or} \quad & [t \text{ is enabled in both } M \text{ and } M']. \end{aligned} \tag{1}$$

In fact if two markings M and M' satisfy condition (1), they enable identical sets of transitions. Whichever transition fires, the two markings M_1 and M'_1 obtained satisfy condition (1). By induction, the sets of firing strings obtained from M and M' are identical.

Condition (1) is not, however, necessary for undistinguishibility. Relation (1) and undistinguishibility are equivalence relations, which partition the set of all markings of an ETN into equivalence classes. By the previous reasoning the latter partition is coarser then the former.

Each equivalence class of the undistinguishibility relation can be identified with a state of the net; since the number of classes is smaller than the number of classes of partition (1), which is finite, it follows that an ETN is equivalent to a finite state automaton; however, not all finite state languages are generated by ETN.

THEOREM 4. *The family of languages generated by* ETN, *denoted by* $\mathscr{L}$ (ETN), *is strictly included by the family of regular languages.*

Proof. Any regular or even finite language which does not contain all prefixes of its strings is not in $\mathscr{L}$ (ETN). ▮

Reduction of an ETN to a finite automaton is not very useful, because it hides net structure; a more expressive description using regular expressions is next presented after some preliminary notation.

For reasons of simplicity we restrict the development to networks without *self-loops*, that is such that for each transition t, $O(t) \cap I(t) = \varnothing$.

Although it is not true that for each ETN there exists an equivalent self-loop-free ETN, it is always possible to find a self-loop free net which is equivalent in a weaker sense, to be defined next.

An ETN N *covers* another N', if there exists a homomorphism $h\colon T' \to T$ such that, for any initial markings of N and N',

$$h(L(N')) = L(N).$$

LEMMA 5. *Given a* ETN N, *it is possible to construct a self-loop free net* N' *which covers* N.

Proof. Let $T_s \subseteq T$ the set of transitions having a self-loop, and let

$$T' = T \cup \{t' \mid t \in T_s\},$$

where t' are new symbols;

$$P' = P \cup \{p_{t'}\}.$$

The input/output functions of N' are now defined:

$$\begin{aligned} &\forall t \in T\colon I'(t) = I(t);\\ &\forall t \in T - T_s\colon O'(t) = O(t);\\ &\forall t \in T_s, \text{ if } r \in T \text{ with } t \neq r\colon \#O'(t, p_r) = \#O(t, p_r)\\ &\qquad \#O'(t, p_t) = 0,\\ &\qquad \#O'(t, p_{t'}) = \#O(t, p_t);\\ &\forall t' \in T'\colon I'(t') = p_{t'}\\ &\qquad \#O'(t', p_t) = \#O(t, p_t);\\ &\qquad \#O'(t', p_r) = 0, \quad \text{if} \quad r \neq t;\\ &S'(t) = S(t), \forall t \in T; \qquad S'(t) = 1, \forall t \in T' - T. \end{aligned}$$

We define the homomorphism:

$$\begin{aligned} h(t) &= t, \qquad t \in T;\\ h(t') &= \lambda. \end{aligned}$$

An example of this transformation is shown in Fig. 3. ∎

DEFINITION 6. An *afferent* of a transition t is a transition t' whose firing brings at least one token into p_t.

The *set of afferents* of t is denoted by $C(t)$.

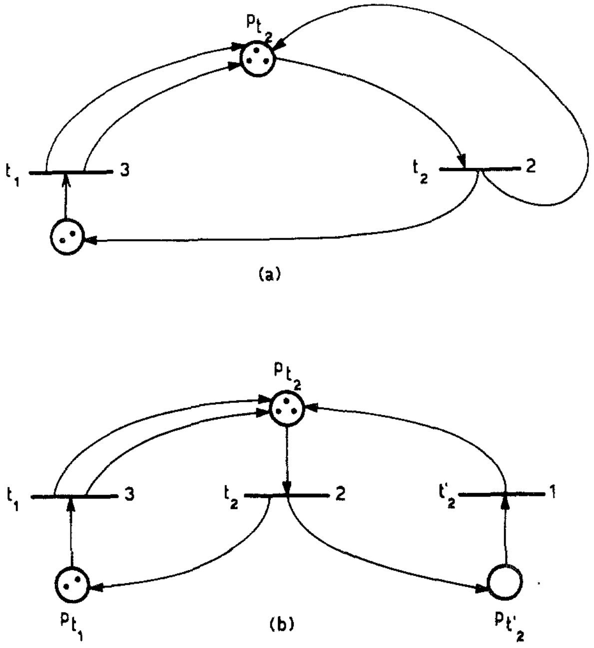

FIG. 3. A self-loop free ETN (b) covering the ETN (a).

DEFINITION 7. An antecedent of t is a string in $(T - \{t\})^*$ whose firing is sufficient to enable t. Formally the *set* $a(t)$ of antecedents of t is:

$$a(t) = \left\{ v \in (T - \{t\})^* \;\middle|\; \sum_{x \in T} (\#(x, v) \cdot \#(P_t, O(x))) \geqslant s(t) \right\}$$

Because $a(t)$ is an infinite set, we define the finite set $A(t)$ of *minimal antecedents* of t.

DEFINITION 8. The set of minimal antecedents of t is: $A(t) = \{v \in a(t) \mid$ no string derived from v by at least one erasure is in $a(t)\}$. Clearly $A(t) \subseteq (C(t))^*$. ∎

We must also recall the *shuffle* operator (Eilenberg, 1974, 1976): $\mathrm{sh}(v, w)$ is the set of all strings $z = v_1 \cdot w_1 \cdot \cdots \cdot v_n \cdot w_n$, where $v_1 \cdot \cdots \cdot v_n = v$, $w_1 \cdot \cdots \cdot w_n = w$, and $v_i, w_i \in T^*$. Intuitively the shuffle of two independent sequences of events of unknown duration represents the totally time-ordered series of events. Using the shuffle we can express the antecedents of t as:

$$a(t) = \mathrm{sh}(A(t), (T - \{t\})^*) \tag{1}$$

We are now ready to write an expression for the language L of firing sequences of an ETN with the single source hypothesis, assuming that the transition t_0 is initially enabled (otherwise $L = \{\lambda\}$).

THEOREM 9.

$$\text{(a)}\quad L = \bigcap_{t \in T - \{t_0\}} L_t \cap L_{t_0},$$

where

$$\text{(b)}\quad \textit{for}\quad t \neq t_0, L_t = (a(t) \cdot t)^* \cdot (T - \{t\})^*,$$

$$\text{(c)}\quad L_{t_0} = t_0 \cdot (T - \{t_0\})^*,$$

and t_0 is the only transition initially enabled.

Proof. First we show that $v \in L \Rightarrow v$ satisfies (a). Clearly $v = t_0 \cdot w$, hence $v \in L_{t_0}$. To show that $v \in L_t$, for any t, consider all occurrencies of t in v:

$$v = w_1 \cdot t \cdot w_2 \cdot t \cdot \cdots \cdot w_n \cdot t \cdot z,$$

where for $i = 1,..., n$, $w_i \in (T - \{t\})^*$ and $z \in (T - \{t\})^*$. Since v is a firing sequence, each firing of t must be enabled by the immediately preceding w_i, hence $w_i \in a(t)$.

Second, we show that a string v in $\bigcap_{t \in T - \{t_0\}} L_t \cap L_{t_0}$ is a firing sequence. Clearly v starts with t_0 since $v \in L_{t_0}$. Since $v \in L_t$, for any $t \in T$ occurring in v, we can write

$$v \in a(t) \cdot t \cdot \cdots \cdot a(t) \cdot t \cdot \cdots \cdot z,$$

where t does not occur in z: therefore each occurrence of t is enabled.

Since each symbol in v is enabled v is a firing sequence. ▮

We shall see in Section 5 that Theorem 9 holds also for inhibitory threshold nets.

The family of *noncounting* or *aperiodical* languages was introduced by McNaughton and Papert (1971).

These are regular languages which are recognized by a counter-free automaton, that is a finite state machine which cannot count modulo n, $n > 1$.

However, such a machine may count up to a finite threshold. It is interesting, that among several formal characterizations of noncounting languages, there is one in terms of nerve nets, which are however quite different from ETNs. The next results relate ETN and noncounting languages.

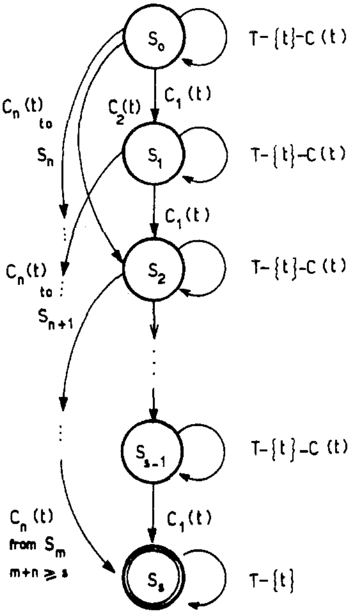

FIG. 4. The regular automaton which recognizes the language $a(t)$; with $C_i(t)$ we denote the set $\{t' \in C(t) \mid \#(p_t, O(t')) = i\}$.

THEOREM 10. *The family* $\mathscr{L}$ (ETN) *is strictly contained within the noncounting (regular) languages.*

Proof. It is known that noncounting languages coincide with the languages represented by star-free regular expressions. Therefore we have to show that expression (b) and (c) of Theorem 9 can be transformed into star-free expressions.

This is immediately done for (c), since

$$L_{t_0} = \bar{\bar{L}}_{t_0} = \overline{(T - \{t_0\}) \cdot \bar{\varnothing}}.$$

It is straightforward that $a(t)$ is noncounting since its reduced automaton (Fig. 4) does not contain any loop of length > 1. Therefore $\overline{a(t)}$ is noncounting. Expressing L_t as:

$$L_t = \bar{\bar{L}}_t = \overline{(\overline{a(t)} \cdot t \cdot \bar{\varnothing}) \cup (\bar{\varnothing} \cdot t \cdot \overline{a(t)} \cdot t \cdot \bar{\varnothing})}$$

we conclude that L_t, hence L, is noncounting. The fact that the inclusion is strict follows from the proof of Theorem 4.

We recall the equivalent definition of a noncounting language: L is noncounting if there exists an integer k such that, if $xy^kz \in L$, then every string xy^hz with $h \geqslant k$ belongs to L. For ETN languages, we prove next that:

THEOREM 11. *If a language* $L \in \mathscr{L}$ (ETN) *contains a string in* $x \cdot \mathrm{sh}(y^2, z)$, $x, y, z \in T^*$, *such that* $\forall t$ *occurring in* y, $C(t) \cap z = \varnothing$ *then to* L *belong all strings* $x \cdot \mathrm{sh}(y^n, z)$, $n \geqslant 2$.

Proof. When a transition $t \in y$ fires, it empties all input places which must be replenished with enough tokens before the second firing of t. As $z \cap C(t) = \varnothing$, t must be enabled by firing of transitions of y. Hence the loop formed by y is self-sustained and can fire an arbitrary number of times. ∎

Intuitively, Theorem 11 shows that the constant k of the previous definition of noncounting languages can be always taken to be equal to 2.

3. CHANGES IN THRESHOLD AND TOPOLOGY

In some cortical models it is assumed that changes in threshold are required to account for attention mechanism and control of activity of neurons. On the other hand learning is usually associated with establishing or reinforcing synaptical connections.

The effects on net behavior of local variations in the values of S and O functions can be analyzed through the changes in the set of firing sequences.

When the threshold of some $t \in T$ increases from $S(t)$ to $S'(t)$ some antecedents of t in the original net eventually become insufficient to enable t, causing deletion of some former firing sequences from L (ETN).

More precisely, let us consider the set of antecedents of t (see Definition 7) as a function of the threshold $s = S(t)$, $a(t, s)$. Then for $s' > s$ we have:

$$a(t, s) = a(t, s') \cup \left\{ v \in (T - \{t\})^* \;\middle|\; s \leqslant \sum_{x \in T} (\#(x, v) \cdot \#(p_t, O(x))) < s' \right\},$$

where the two sets on the right-hand side are disjoint. By the same reasoning one could treat changes in the topology of the net, caused by addition or deletion of arcs.

Languages of ETN can be classified in an infinite hierarchy based on values of threshold.

Let us denote by $\mathscr{L}$ (ETN$_s$) the family of languages of ETNs possibly with self-loops, in which the *maximum* value of $S(t)$ is s.

We have the following:

THEOREM 12. *If* $s < s'$ *than* $\mathscr{L}$ (ETN$_s$) $\subsetneqq \mathscr{L}$ (ETN$_{s'}$).

Proof. For every net N in ETN$_s$ with initial marking M_0 we can define in

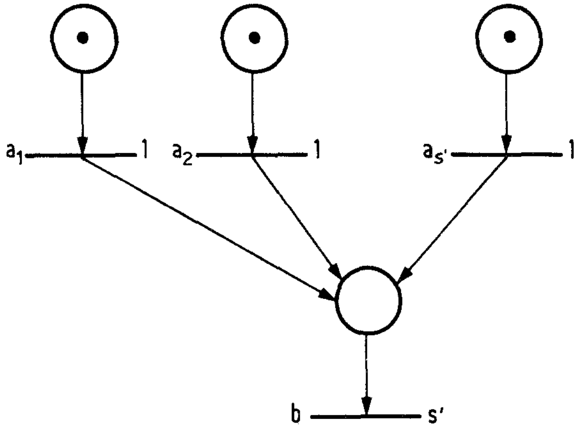

FIG. 5. The net of Theorem 12.

ETN_s a net $N' = (P, T, I, O', S')$ with marking M_0', which generates the same firing sequences, by:

$$\forall t, t' \in T', t \neq t', S'(t) = s';$$

$$\#(p_t, O'(t)) = \#(p_t, O(t)) + (s' - S(t));$$

$$\#(p_t, O'(t')) = \#(p_t, O(t'));$$

$$M_0'(p_t) = M_0(p_t) + (s' - S(t)).$$

Notice that N' has a uniform threshold.

The effect of raising by d the threshold of t in N' is balanced by keeping in p_t d additional tokens. This is easily obtained by making $M_0'(p_t) = M_0(p_t) + d$ and by adding to t d arcs (self-loops)[2] which bring d tokens into p_t anytime t fires. Hence, any ETN with maximum threshold s, can be transformed to an equivalent net with threshold s' by suitably raising the threshold of all transitions.

To show that inclusion is proper, let us consider the net of Fig. 5. The language generated is the set of all possible permutations of a_i, $1 \leqslant i \leqslant s'$, followed by b.

It can be shown that no net with threshold less than s' can generate such a language (Pistorello and Romoli, 1980). ▮

---

[2] If we do not want to use self-loops, we can define a net whose language covers $L(N)$ (Lemma 5); equality of $L(N)$ and $L(N')$ cannot be granted.

4. OPERATORS FOR ETN

Traditional language operators are not suited for use with ETN, because $\mathscr{L}$ (ETN) is not closed neither with respect to Boolean operators (union, intersection, complement), nor to catenation.

THEOREM 13. *The family $\mathscr{L}$ (ETN) is not closed with respect to Boolean operators and catenation.*

Proof. As nonclosure with respect to complement, union and catenation is straightforward, we shall only consider intersection.

As a counterexample let us consider the nets N_1 and N_2 of Fig. 6, which generate, respectively, the languages:

$$L_1 = \mathrm{sh}(\{a'ax, aa'x, axa'ax, aa'ax\}, \{b'b, bb'b\})$$

$$L_2 = \mathrm{sh}(\{b'bx, bb'x, bxb'bx, bb'bx\}, \{a'a, aa'a\})$$

Consider the intersection $L_3 = L_1 \cap L_2$ and suppose there exist an ETN N_3

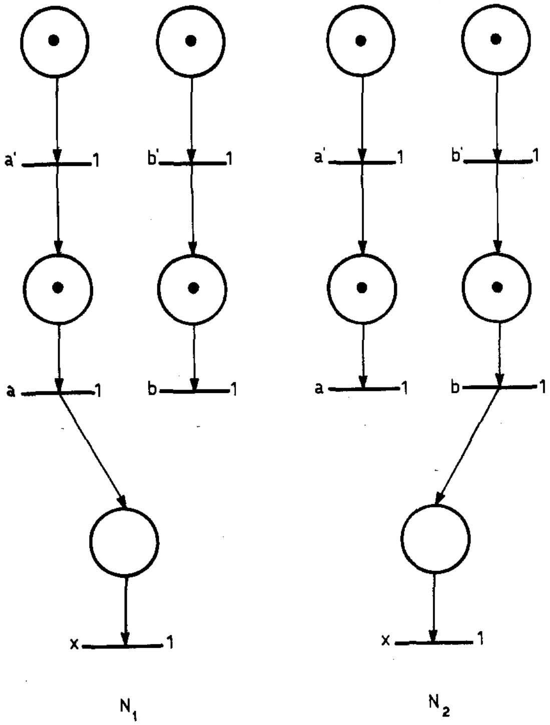

FIG. 6. The net of the counterexample of Theorem 13.

which recognizes L_3. Since x is enabled by a in L, and by b in L_2, whatever is the structure of N_3 we must have

$$A(x) = \{ab, ba\},$$

that is, by Definition 6,

$$S(x) \leqslant \#(p_x, O(a)) + \#(P_x, O(b)),$$

and

$$\#(P_x, O(a')) = \#(P_x, O(b')) = 0.$$

On the other hand the strings

$$aa'ax, bb'bx$$

are not in L_3, whereas all their proper prefixes are in L_3. Hence $aa'a$ and $bb'b$ must not enable x, hence

$$2 \cdot (P_x, O(a)) < S(x),$$
$$2 \cdot (P_x, O(b)) < S(x),$$

which contradict the previous inequality.

Instead of Boolean operators we propose two net operators, called *overlap* and *match*, similar to union and intersection, whose definition is strictly bound to net structure.

DEFINITION 14. The *overlap* of two single source nets N_1 and N_2, denoted by $N_1 \sqcup N_2$, is a net N_3 such that:

— The set T_3 of transitions of N_3 is the union of sets T_1 T_2 of transitions of N_1 and N_2;

— the output function O_3 of N_3 derives from functions O_1 and O_2 of N_1 and N_2 by:

— $\forall t, t' \in T_3, \#(p_t, O_3(t')) = \max[\#(p_t, O_1(t')), \#(p_t, O_2(t'))]$;

— $\forall t \in T_3$, if $t \in T_1 \cap T_2$ then $S_3(t) = \min[S_1(t), S_2(t)]$ else $S_3(t)$ equals $S_1(t)$ or $S_2(t)$;

— $\forall t \in T_3$, $M_0(t) = \max[M_0(p_t)$ in N_1, $M_0(P_t)$ in $N_2]$.

An example is shown in Fig. 7.

The language generated by N_3 is somewhat larger than the union of the languages generated by N_1 and N_3.

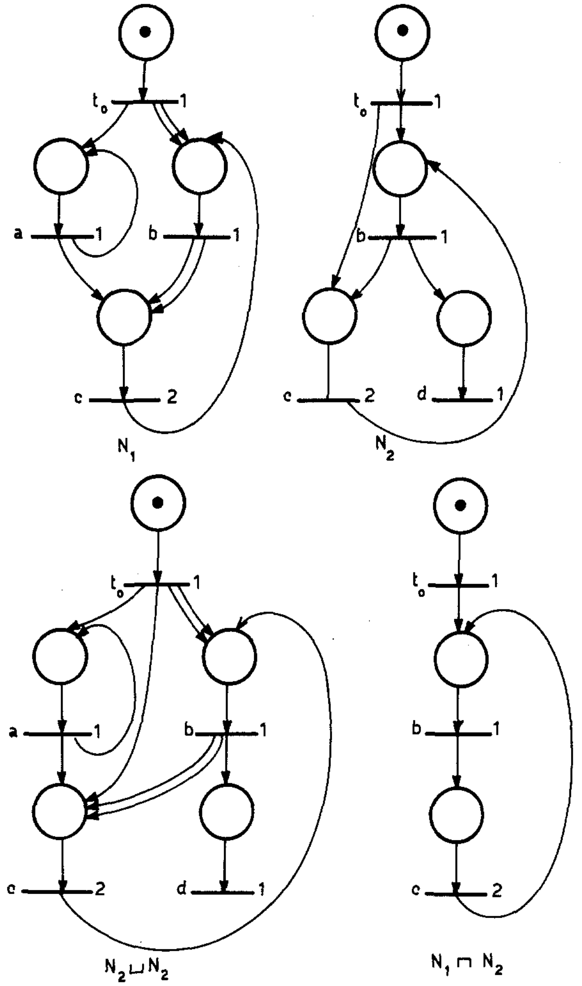

FIG. 7. Net operations.

THEOREM 15. *Given languages L_1 and L_2, generated by nets N_1 and N_2, the language $L_3 = L\ (N_1 \sqcup N_2)$ includes both L_1 and L_2.*

Proof. Let us call $a_1(t)$, $a_2(t)$, $a_3(t)$ the sets of antecedents of t, respectively, in N_1, N_2, and N_3. Each string v belonging to $a_1(t) \cup a_2(t)$, for any $t \in T_1 \cup T_2$, also belongs to $a_3(t)$, since $\sum_{x \in T}(\#(x, v) \cdot \#(p_t, O_3(x))) > \max_{i=1,2}[\#(p_t, O_i(x))] \geqslant S_3(t)$. Let us respectively call $L_{1,t}$, $L_{2,t}$, $L_{3,t}$ the language related to t (see (b) of Theorem 9), in each net N_1, N_2, N_3.[3]

---

[3] If any transition t does not belong to T_1 (or T_2) we assume $L_{1,t} = T_1^*$ (or $L_{2,t} = T_2^*$), so that $L_{i,t}$ is defined for any $i = 1, 2, 3$ and $t \in T_3$, without affecting L_1 and L_2.

Since for any $t \in T_3$, $a_1(t) \cup a_2(t) \subseteq a_3(t)$ we have:

$$(L_{1,t} \cup L_{2,t}) \subseteq L_{3,t}.$$

Through some simple set operations, we derive:

$$\left(\bigcap_{t \in T_3} L_{1,t}\right) \cup \left(\bigcap_{t \in T_3} L_{2,t}\right) \subseteq \bigcap_{t \in T_3} L_{3,t}$$

$$L_1 \cup L_2 \subseteq L_3. \quad \blacksquare$$

Similarly we define the match of two nets.

DEFINITION 16. The *match* of two single source nets N_1 and N_2, denoted by $N_1 \sqcap N_2$, is the net N_3 such that

— $T_3 = T_1 \cap T_2$;
— $\forall t \in T_3$, $S_3(t) = \max[S_1(t), S_2(t)]$;
— $\forall t, t' \in T_3$, $\#(p_t, O_3(t')) = \min[\#(p_t, O_1(t')), \#(p_t, O_2(t'))]$;
— $\forall t \in T_3$, $M_0(t) = \min[M_0(t)$ in N_1, $M_0(t)$ in $N_2]$.

In Fig. 7 an example is shown.

By applying the same reasoning of Theorem 16, it is possible to prove:

THEOREM 17. *Given two nets N_1 and N_2 which generate languages L_1 and L_2,*

$$L(N_1 \sqcap N_2) \subseteq L_1 \cap L_2. \quad \blacksquare$$

In most cases $L(N_1 \sqcup N_2)$ strictly includes $L_1 \cup L_2$ and $L(N_1 \sqcap N_2)$ is strictly included in $L_2 \cap L_2$, like in the examples of Fig. 6, but these are not general rules.

The mathematically oriented reader should resist the temptation to believe that $\sqcup$, $\sqcap$ and $\subseteq$ define a lattice of nets and languages (similar to the well-known lattice of regular languages with Boolean operators); in general $L(N_1 \sqcup N_2)$ and $L(N_1 \sqcap N_2)$ are not the l.u.b. and g.l.b. of $L(N_1)$ and $L(N_2)$ with respect to language inclusion.

5. INHIBITORY TN

Inhibition is certainly a necessary ingredient of any brain theory. We have therefore studied some properties of threshold nets when inhibitory connections are allowed.

Next we propose two models of inhibitory threshold nets (ITN) with unbounded and bounded inhibition.

The following changes are made to the definitions of ETN. An arc from t to $p_{t'}$ can be inhibitory as well as excitatory. Upon firing of t, an inhibitory arc decrements the token count of $p_{t'}$ by one. Accordingly the marking of a place can be also negative.

DEFINITION 18. An inhibitory threshold net (ITN) is a system made of six components: P, T, I, O, H, S, where P, T, I, S are as in Definition 1 (i.e., ETN).

The *excitatory output* function O maps t into the multiset of excited output places.

The *inhibitory output* function H maps t into the multiset of inhibited output places. We assume that, for any $t \in T$, $O(t)$ and $H(t)$ are disjoint, i.e., there exists no place p s.t. $\#(p, O(t)) > 0$ and $\#(p, H(t)) > 0$.

The disjointness hypothesis derives from neurophysiological evidence suggesting that a neuron receives from another one either excitatory or inhibitory pulses, but not both. The marking M maps each place p to a *signed* integer. A transition t is *enabled* if $M(I(t)) \geqslant S(t)$. The *firing* of t generates the marking M', such that for each p,

$$\begin{aligned} M'(p) = \text{if} \quad p \in I(t) \qquad & \text{then} \quad \#[(p, O(t)) - \#(p, H(t))] \\ & \text{else} \quad [M(p) + \#(p, O(t)) - \#(p, H(t))]. \end{aligned}$$

The definition of the firing function f and of the language L of firing sequences generated by a net remains the same as for ETN.

Moreover we assume that for any $t \in T$, $I(t) = \{p_t\}$.

For each transition t we define the set $CE(t)$ of *excitatory afferents* and $CH(t)$ of *inhibitory afferents* as:

$$CE(t) = \{t' \in T \mid p_t \in O(t')\},$$

$$CH(t) = \{t' \in T \mid p_t \in H(t')\}.$$

The union of $CE(t)$ and $CH(t)$ is the set of *afferents* of t, denoted $C(t)$.

DEFINITION 19. For an ITN the set of *antecedents* of t, $a(t)$, is:

$$a(t) = \left\{ v \in (T - \{t\})^* \;\middle|\; \sum_{x \in CE(t)} [\#(x, v) \cdot \#(p_t, O(x))] - \sum_{x \in CH(t)} [\#(x, v) \cdot \#(p_t, H(x))] \geqslant S(t) \right\}.$$

Since Definition 19 defines a set with the same properties of $a(t)$ of Definition 6, with respect to firing of t, Theorem 9 holds also for ITN.

Introduction of inhibition into net systems often causes a noticeable increase of computing power. For example, it is known (Valk, 74) that Petri nets become equivalent to Turing machines.

What is noteworthy of TN is that the increase due to inhibition is less sweeping.

THEOREM 20. *The family* $\mathscr{L}$ (ITN) *of languages generated by inhibitory* TN *is properly included by the family of context-sensitive* (*i.e., type* (1) *languages, but not by the one of context-free* (*i.e., type* (2) *languages.*

Proof. Every language L_t is context-free, since it is straightforward to build a push-down stack automaton which recognizes it. Hence the language L, intersection of context-free languages, is context-sensitive.

An example is provided by the net of Fig. 8, which generates $\{a^n b^n b^+ dc^n c^+ e \mid n \geqslant 1\}$ obviously not a context-free language.

To show that inclusion is proper it suffices to consider that not all context-sensitive languages are closed with respect to the prefix operation. ▮

To complete this section we consider threshold nets with bounded accumulation of tokens in places. Motivation for this variant comes from the observation that it is physically unsound to assume unlimited accumulation of tokens in a place: for this would amount to unlimited polarization or depolarization of a neuron body.

Consider an inhibitory net ITN and a positive integer k; a marking M,

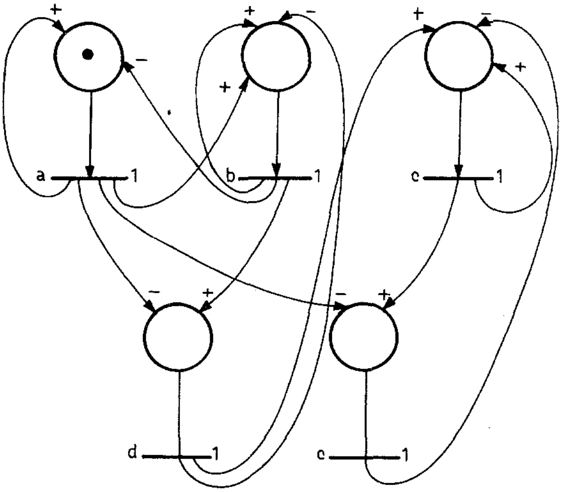

FIG. 8. The ITN generating the language $\{a^n b^{n+m} dc^{n+k} e;\ n \geqslant 1,\ m \geqslant 1,\ k \geqslant 1\}$.

such that $|M(p)| \leqslant k$, for each p in P, is termed *k-bounded.* Let us assume that $|M_0(p_{t_0})| \leqslant k$.

DEFINITION 21. The language generated by ITN with bound k is

$$\begin{aligned} L_k(\mathrm{ITN}, M_0) = \{w \mid {} & w = v_1 \cdots v_n \text{ and for each } 1 \leqslant i \leqslant n \text{ and } p \text{ in } P, \\ & M_i = f(M_0, v_1 \cdots v_i) \text{ is } k\text{-bounded}\}. \end{aligned}$$

It is obvious that $L_k(\mathrm{ITN}, M_0) \subseteq L(\mathrm{ITN}, M_0)$, for any ITN and M_0. The language L_k is now recognized using a finite amount of memory.

DEFINITION 22. The set of antecedents of a transition t with bound k is:

$$a_k(t) = \{w \in a(t) \mid \text{no prefix of } w \text{ brings more than } k \text{ tokens into } p_t\},$$

which can be reformulated as:

$$\begin{aligned} a_k(t) = \Bigg\{ & w \in (T - \{t\})^* \mid w = v_1 \cdots v_n, \text{ and for } 1 \leqslant i \leqslant n, \\ & \left| \sum_{j=1}^{i} \#(p_t, O(v_j)) - \sum_{j=1}^{i} \#(p_t, H(v_j)) \right| \leqslant k \\ & \text{and } \sum_{j=1}^{n} \#(p_t, O(v_j)) - \sum_{j=1}^{n} \#(p_t, H(v_j)) \geqslant S(t) \Bigg\}. \end{aligned}$$

If we substitute $a(t)$ of Theorem 9 with $a_k(t)$, Theorem 9 provides an expression of $L_k(\mathrm{ITN}, M_0)$: this is obvious since Theorem 9 holds for ITN and Definition 22 restricts $a(t)$ to the strings which meet the bound k on place p_t. As for $\mathscr{L}$ (ETN), it is possible to show the inclusion of $\mathscr{L}_k$ (ITN) in the noncounting languages.

THEOREM 23. *The family* $\mathscr{L}_k$ (ITN) *of languages generated by* ITN *with bound k, is included by the noncounting languages.*

Proof. Following the pattern of the proof of Theorem 10, it is possible to show that (a) and (b) can be rewritten as star-free expression.

First we substitute every occurrence of $(T - \{t\})^*$ by the equivalent $\overline{\overline{\varnothing} \cdot t \cdot \overline{\varnothing}}$. Then consider the expression $(a_k(t) \cdot t)$ and first prove that $a_k(t)$ is noncounting. Let us use in the following, for any $w \in (T - \{t\})^*$ the notation $e(t, w)$ instead of $\sum_{x \in T} [\#(x, w) \cdot (\#(p_t, O(x)) - \#(p_t, H(x)))]$ to denote the number of positive or negative tokens brought into p_t by the transitions of w. It is immediate to verify that $e(t, v \cdot w) = e(t, v) + e(t, w)$. The language $a_k(t)$ can be recognized by a finite-state automaton A such that:

— A has $2k+2$ states:

$$S_i, i=-k,\ldots, O,\ldots, k \text{ and } S_{\text{halt}};$$

state S_i corresponds to the marking $M(p_t)=i$ and S_{halt} corresponds to trespassing the bound k in p_t.

— S_0 is the initial state.

— States $S_{S(t)}, S_{S(t)+1},\ldots, S_k$ are final states.

— In state S_i, the next state of A upon encountering $t' \in (T-\{t\})^*$ is the state $S_{i+e(t,t')}$ if $|i+e(t,t')| \leqslant k$ else S_{halt}.

Note that the next state function in state S_i for a string w in T^* is, for every prefix v of w, if $|i+e(t,v)| \leqslant k$ then $S_{i+e(t,w)}$ else S_{halt}.

Let us now assume that a string $z=v \cdot w^{2k+1}x \in a_k(t)$, where $v, w, x \in (T-\{t\})^*$ and $w \neq \lambda$.

It must be: $e(t,w)=0$, since otherwise no string of the form $v \cdot w^{2k+n}x$, with $n \geqslant 1$, would belong to $a_k(t)$, because $|e(t,w^{2k+n})| > 2k$ and $|e(t, v \cdot w^{2k+1})| > k$. Therefore the automaton, after analyzing the string $v \cdot w^{2k+1}$, reaches either the state $S_{e(t,v)}$ or S_{halt}. At this point, after encountering any further occurrences of w, it always returns to the same state. Hence for $n \geqslant 2k+1$, $v \cdot w^{n+1} \cdot x \in a_k(t) \Leftrightarrow v \cdot w^n \cdot x \in a_k(t)$, that is $a_k(t)$ is noncounting. We can rewrite the expression

$$(a_k(t) \cdot t)^* = \overline{(\overline{a_k(t)} \cdot t \cdot \bar{\varnothing}) \cup (\bar{\varnothing} \cdot t \cdot \overline{a_k(t)} \cdot t \cdot \bar{\varnothing})}. \quad \blacksquare$$

Note that $\mathscr{L}_k$ (ITN) are noncounting but Theorem 11 does not apply to $\mathscr{L}_k$ (ITN).

As a particular case of $\mathscr{L}_k$ (ITN) we can consider the family $\mathscr{L}_k$ (ETN) of languages generated by excitatory TN with bound k.

The effect of the introduction of inhibition and of the bound k on the families of languages for different kind of threshold networks is summarized in Table I, which in position (j, i) lists an example of a language belonging to $\mathscr{L}_i - \mathscr{L}_j$.

Let us prove some of the cases in the table:

EXAMPLE 1. $L_1 = \{((ab)^n c)^*\}$ $n > 1$, fixed.

The language belongs to $\mathscr{L}_k$ (ETN) (and $\mathscr{L}_k$ (ITN)), for $k=2n$, since it is generated by the net a of Fig. 9.

L_1 cannot belong to $\mathscr{L}$ (ETN) since, for Theorem 11 any language $\in \mathscr{L}$ (ETN) containing the string $(ab)^n$, must also contain $(ab)^{n+1}$. The fact that $L_1 \notin \mathscr{L}$ (ITN) can be shown by contradiction. Let us suppose that there exists an ITN recognizing L_1. Since $(ab)^n$, but not $(ab)^{n-1}$, is an antecedent

TABLE I

$\mathscr{L}_i$ / $\mathscr{L}_j$	$\mathscr{L}$(ETN)	$\mathscr{L}_k$(ETN)	$\mathscr{L}$(ITN)	$\mathscr{L}_k$(ITN)
$\mathscr{L}$(ETN)		$L_1 = \{((ab)^n \cdot c)^*\}$, $n > 1$, fixed	Any language $\in$ $\mathscr{L}$(ITN) which is not regular noncounting (see Theorem 20)	$L_3 = \{(ab)^n \cdot c\}$, $n > 1$, fixed
$\mathscr{L}_k$(ETN)	$L_2 = \{(ab \cdot \mathrm{sh}((ab)^*, c))^*\}$		Any language $\in$ $\mathscr{L}$(ITN) which is not regular noncounting (see Theorem 20)	$L_4 = \{(ab)^* \cdot acb\}$
$\mathscr{L}$(ITN)	$\varnothing$	$L_1 = \{((ab)^n \cdot c)^*\}$, $n > 1$, fixed		$L_3 = \{(ab)^n \cdot c\}$, $n > 1$, fixed
$\mathscr{L}_k$(ITN)	$L_2 = \{(ab \cdot \mathrm{sh}((ab)^*, c))^*\}$	$\varnothing$	Any language $\in$ $\mathscr{L}$(ITN) which is not regular noncounting (see Theorem 20)	

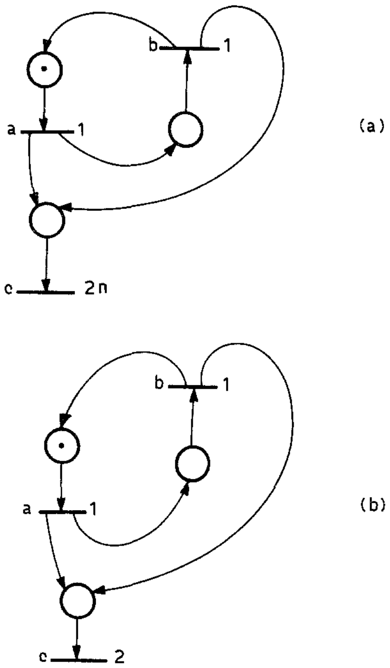

FIG. 9. (a) The $2n$-bounded ETN of Example 1. (b) The ETN of Example 2.

of c, it follows that ab brings a positive amount of tokens into p_c. Hence also $(ab)^{n+i}c$ should be a firing sequence: a contradiction.

EXAMPLE 2. $L_2 = (ab \cdot \mathrm{sh}((ab)^*, c))^*$.

L_2 belongs to $\mathscr{L}$ (ETN) since it is generated by the net b of Fig. 9.

L_2 cannot belong to $\mathscr{L}_k$ (ITN) or $\mathscr{L}_k$ (ETN), since c is enabled by the firing of ab. This means that, in every net which generates L_2, a firing of ab brings some positive tokens into p_c. As ab can fire any number of times before c, there is no bound on the marking of p_c.

EXAMPLE 3. $L_3 = \{(ab)^n c\}$, $n > 1$, is recognized by the ITN of Fig. 10a with bound $k = 2n$. Reasoning as in Example 1, one can prove that $L_3 \notin \mathscr{L}$ (ETN) and $L_3 \notin \mathscr{L}$ (ITN).

EXAMPLE 4. $L_4 = \{(ab)^* acb\}$ is recognized by the ITN of Fig. 10b. To show that L_4 is not in $\mathscr{L}_k$ (ETN) observe that p_c must receive some excitation from a, and that the number of tokens in p_c can grow unbounded (because of $(ab)^*$).

The relationship between the various families is summarized in Fig. 11.

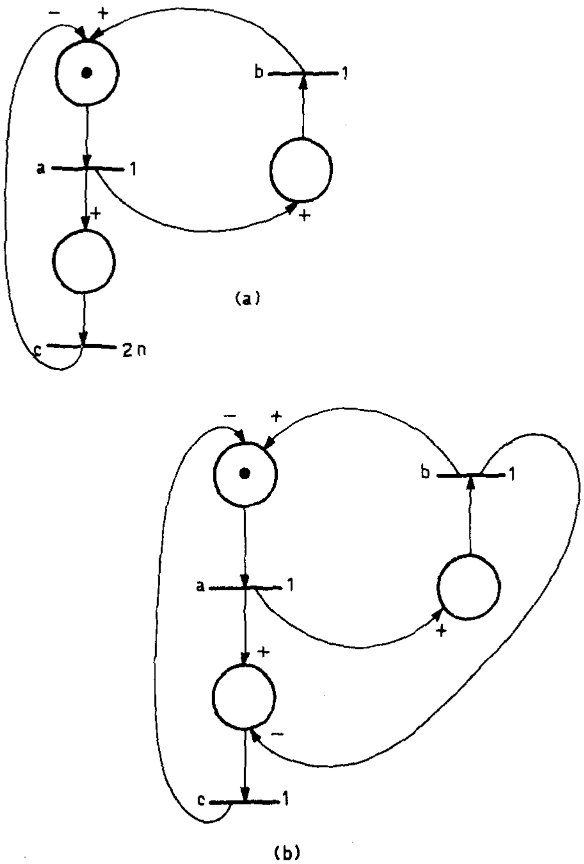

FIG. 10. (a) The $2n$-bounded ITN recognizing L_3. (b) The $2n$-bounded ITN recognizing L_4.

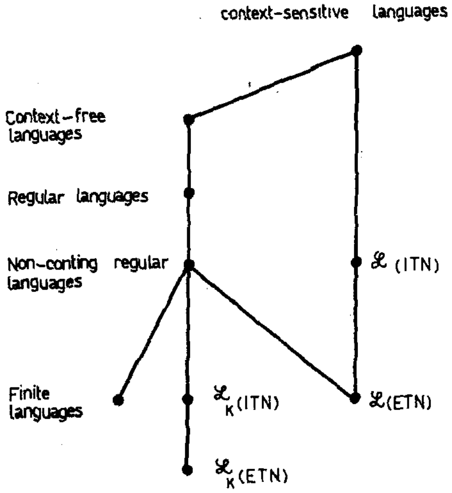

FIG. 11. Relationships between various families of languages.

CONCLUDING REMARKS

The cell assembly theory of neuronic networks has motivated this work. We have introduced and formally characterized two sorts of threshold nets, derived from Petri nets, where the firing rules and the token game are modified.

The first model, excitatory threshold nets (ETN) includes only excitatory connections; the second one, inhibitory threshold nets, has excitatory as well as inhibitory connections. Our analysis of TN has obtained the following results.

Excitatory TN are reducible to finite state automata; sets, or languages, of firing sequences can be effectively described by regular expressions, and are strictly included by the noncounting languages already considered by McNaughton and Papert in connection with a simplified model of nerve nets. The family of languages of ETN is not closed with respect to Boolean operations and catenation, but we have introduced two new operators, overlap and match, to manipulate nets and languages, in a way suggestive of phenomena of growth, learning or damage of cortex.

An infinite hierarchy based on the values of threshold has been evidenced.

The introduction of unbounded inhibition extends the generative power of ETN beyond finite state (type 3) and context-free (type 2) languages; the family of languages generated by inhibitory TN is strictly included by the context-sensitive languages (type 1).

If a finite bound on the amount of inhibition and excitation is inposed, the generative power of ITN and unbounded ETN are noncomparable.

On the theoretical side much work remains to be done on a precise characterization of threshold languages with respect to existing families of languages, in particular the aperiodic hierarchies (Brzozowski, 1971).

To conclude let us make a disclaim: The formal models proposed were inspired by the cell-assembly theory of the brain, but this paper does not attempt to closely explain any cerebral or mental phenomena like associative memory, logical reasoning or learning. In our opinion a formal study of threshold nets should provide solid foundations for extending in the future the analysis to structured patterns of behavior in organized threshold nets.

ACKNOWLEDGMENTS

We are indepted to V. Braitenberg for suggesting the subject and for providing noninhibitory criticism. We also thank D. Mandrioli for his help in revising the previous drafts.

RECEIVED: December 20, 1980; REVISED: June 26, 1981

REFERENCES

BRAITENBERG, V., "Cell-Assemblies in the Cerebral Cortex," Max-Plank Inst. fuer biologische Kybernetik, Tuebingen, 1973.

BRAITENBERG, V., Thoughts on cerebral cortex, *J. Theor. Biol.* **46** (1974).

BRAITENBERG, V., "Cortical Architectonics: General and Areal Architectonics of the Cerebral Cortex," New York, 1978.

BRZOZOWSKI, J. A., CULIK, K., AND GABRIELLIAN, A., Classification of noncounting events, *J. Comput. System Sci.* **5** (1971).

CAIANELLO, E. R., "Neuronal Networks," Springer-Verlag, Berlin/New York, 1968.

CRESPI-REGHIZZI, S. AND MANDRIOLI, D., Petri nets and Szilard languages, *Inform. Contr.* **3** (1976).

EILENBERG, S., "Automata, Languages and Machines," Academic Press, New York, Vol. A, 1974; Vol. B, 1976.

HACK, M., "Petri Net Languages," MIT Press, Cambridge, Mass., 1975.

HEBB, D. O., "The Organization of Behavior," Wiley, New York, 1949.

HOPCROFT, J. AND ULLMAN, J., "Formal Languages and Their Relation to Automata," Addison–Wesley, Reading, Mass., 1969.

MCCULLOCH, W. AND PITTS, W., A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* **5** (1943).

MCNAUGHTON, R. AND PAPERT, S., "Counter-Free Automata," MIT Press, Cambridge, Mass., 1971.

PETRI, C., "Kommunication mit Automaten," Thesis, Universität Bonn, 1962.

PISTORELLO, A. AND ROMOLI, C., "Reti a soglia e assembramenti cellulari," Thesis, ISI, Università di Pisa, 1980.

VALK, R., "Dynamic Petri Nets," Universität Hamburg, 1974.