Turing Machines, Finite Automata and Neural Nets*

MICHAEL ARBIB

Sydney, Australia†

Abstract. This paper[1] compares the notions of Turing machine, finite automaton and neural net. A new notation is introduced to replace net diagrams. "Equivalence" theorems are proved for nets with receptors, and finite automata, and for nets with receptors and effectors, and Turing machines. These theorems are discussed in relation to papers of Copi, Elgot and Wright; Rabin and Scott; and McCulloch and Pitts. It is shown that sets of positive integers "accepted" by finite automata are recursive; and a strengthened form of a theorem of Kleene is proved.

1. *Neural Nets and Finite Automata*

In this section a new symbolism is introduced, which obviates the need for drawing net diagrams; an 'equivalence' theorem is proved for neural nets with suitable effectors, and finite (one-way one-tape) automata; and it is shown how this theorem may be used to "telescope" the papers of Copi, Elgot and Wright [2], and of Rabin and Scott [6].

Symbolism: $i\langle j\rangle$: the jth digit of the binary representation of the integer i.

$A(\&,m)$: the element A is a conjunction with m inputs.

$A(\vee,m)$: the element A is a disjunction with m inputs.

$A(\sim)$: the element A is a negation.

$A(d)$: the element A is a delay.

$A^{(i)} = B$: the ith input of the element A is taken from (the output of the element) B.

DEFINITION 1. Let the net N have q delay elements $d_0, \cdots, d_{q-1}$. We shall say that N is in internal state s_j $(0 \leqq j < 2^q)$ at time t if and only if $j\langle i\rangle = d_{i-1}(t)$. We denote by S the set of all internal states of N. (If $q = 0$, N has just one state s_0.)

Since $d_i(0) = 0$ $(i = 0,\cdots,q-1)$, N is in internal state s_0 at time $t = 0$.

It is immediate from the definitions of [2] that:

LEMMA 1. *The internal state of N at time $t+1$ is uniquely determined by the input to N, and the internal state of N, at time t.*

We now introduce the receptor TSn.

DEFINITION 2. TSn is a tape scanner which can "recognize" the 2^n letters

* Received October, 1960.

† Presently at the Massachusetts Institute of Technology, Cambridge, Mass.

[1] The reader will find the following, or its equivalent, a necessary and sufficient prerequisite for the study of this paper: the paper [2], the first part of the book [3], and sections 1, 2, 3, 5 and 6 of the paper [6]. The notation of these three references will be used in this paper without further explanation.

$\sigma_0, \sigma_1, \cdots, \sigma_{2^n-1}$ of an alphabet Σ_n, scanning one square of the tape at each moment of time and moving the tape one square to the left between successive moments. TSn has n output wires $w_1, \cdots, w_n$ such that if TSn scans σ_k at time t then

$$w_a(t) = k\langle a\rangle \qquad (a = 1,\cdots,n).$$

With this preparation, we may now state and prove our "equivalence" theorem for neural nets and finite automata.

THEOREM 1. *Given a net N and any output o of N whose firing depends only on the internal state of N, then a finite automaton $\mathfrak{A}$ can be found whose function is equivalent to that of the net, by considering the input sequences to N realized by o, and the tapes accepted by $\mathfrak{A}$ to be identical; and vice versa.*

PROOF. *Part I.* Let the net N be given with n inputs $i_1, \cdots, i_n$, q delay outputs $d_0, \cdots, d_{q-1}$, and the chosen output o.

If the w_a of TSn is connected to the i_a $(a = 1,\cdots,n)$, then it is clear that any input sequence to N may be uniquely replaced by a tape scanned by TSn.

Let the set F of internal states of N be defined as

$$\{s_j \mid N \text{ is in the internal state } s_j \text{ at time } t \Rightarrow o(t) = 1\}.$$

By Lemma 1, N defines a single-valued function M from $S \times \Sigma_n$ to S.

Then the finite automaton $\mathfrak{A} = (S, M, s_0, F)$ with alphabet Σ_n is *equivalent* to the net N.

Part II. Let $\mathfrak{A} = (S, M, s_0, F)$ be a finite automaton with alphabet Σ, where

$$\Sigma = \{\sigma_0, \sigma_1, \cdots, \sigma_l\}$$
$$S = \{s_0, s_1, \cdots, s_k\}$$
$$F = \{s_{i_1}, \cdots, s_{i_j}\}$$

and choose n, q such that $2^{q-1} \leqq k < 2^q$, $2^{n-1} \leqq l < 2^n$.

Now construct a net N with n inputs i_{a1} $(a = 1,\cdots,n)$ and a chosen output o which is *equivalent* to $\mathfrak{A}$. N comprises the following $n+kl+3q+j+1$ elements:

(1) n elements $i_{a0}(\sim)$; $i_{a0}^{(1)} = i_{a1}$ $(a = 1,\cdots,n)$,
(2) kl elements $I_{ij}(\&,n+q)$ $(i = 1,\cdots,k;\ j = 1,\cdots,l)$,
(3) q elements $D_{b1}(\vee, m_b)$ $(b = 1,\cdots,q;\ m_b \leqq kl)$,
(4) q elements $\bar{D}_{b1}(d)$; $\bar{D}_{b1}^{(1)} = D_{b1}$ $(b = 1,\cdots,q)$,
(5) q elements $\bar{D}_{b0}(\sim)$; $\bar{D}_{b0}^{(1)} = \bar{D}_{b1}$ $(b = 1,\cdots,q)$,
(6) j elements $S_d(\&,q)$; $S_d^{(b)} = \bar{D}_{b,i_d\langle b\rangle}$ $(d = 1,\cdots,j;\ b = 1,\cdots,q)$,
(7) 1 element $o(\vee,j)$; $o^{(d)} = S_d$ $(d = 1,\cdots,j)$,

with the further connections,

$$I_{ij}^{(a)} = \bar{D}_{a,i\langle a\rangle} \text{ for } 1 \leqq a \leqq n; \quad I_{ij}^{(n+a)} = i_{a,j\langle a\rangle} \text{ for } 1 \leqq a \leqq q,$$

and such that D_{b1} receives one input from each of the $m_b \in I_{ij}$ for which $s_k = M(s_i,\sigma_j)$ and $k\langle b\rangle = 1$.

The net N so constructed is *equivalent* to $\mathfrak{A}$.

I shall now show how the above theorem, which is short and relatively simple, may be used to "telescope" portions of papers [2] and [6].

If we assume the results of Copi, Elgot and Wright, then theorems 3, 4 (since, clearly, S is regular $\Rightarrow S^*$ {in the sense of [6]} is regular), 5, 13, 14 and corollary 5.1 of [6] follow immediately, and Definition 7 and Theorem 6 become unnecessary.

Conversely, if we assume the results of Rabin and Scott, then the analysis and synthesis theorems of [2] follow immediately from their Theorem 14, on applying our Theorem 1. Then all we require of [2] are the definitions of net, input, output, etc.

In the latter case, we may still wish to retain the discussion of regular sets and expressions contained in [2] for its intrinsic interest. Should this be so, we may then replace the inductive proof that "all sets of the form $T(\mathfrak{A})$ are regular", outlined in the proof of theorem 14 of [6], by the following more elegant proof:

Let $\mathfrak{A}$ be the automaton (S, M, s_0, F). Say that the $S \times \Sigma$ pair $a_{l_k} = (s_{i_k}, \sigma_{j_k})$ stands in the direct transition relationship to the pair a_{l_j} if and only if $s_{i_k} = M(s_{i_j}, \sigma_{j_j})$. A sequence $a_{l_1}, \cdots, a_{l_n}$ is called a transition sequence if and only if a_{l_r} stands in the direct transition relationship to $a_{l_{r-1}}$ $(r = 2, \cdots, n)$.

Clearly, if a set of sequences of $S \times \Sigma$ pairs (s_{i_k}, σ_{j_k}) is regular, then the derived set of sequences of symbols σ_{j_k} is regular.

Now consider the $b_k = (s_0, \sigma_k)$ and the $c_{ij} = (s_i, \sigma_j)$ where $s_i \in F$. Clearly a tape is in $T(\mathfrak{A})$ if and only if it is derived from a transition sequence from a b_k to a c_{ij}. Thus $T(\mathfrak{A})$ is derived from the union of the sets of transition sequences from a b_k to a c_{ij}. Hence by Lemma 1 of [2] and the above remarks, $T(\mathfrak{A})$ is regular.

2. *Neural Nets and Turing Machines*

In this section we prove an "equivalence" theorem for neural nets with suitable effectors and receptors, and Turing machines; and use this to prove assertions made, without proof, by McCulloch and Pitts [5].

We first introduce the receptor $TS'n$, and the effectors TM, TPn and CO.

DEFINITION 3. $TS'n$ is a tape scanner identical with TSn, save that it does not move the tape.

DEFINITION 4. TM is a tape mover with two input wires L and R. At time t, TM moves the tape one square to the left if $L(t) = 1$ and $R(t) = 0$; to the right if $L(t) = 0$ and $R(t) = 1$; and not at all otherwise.

DEFINITION 5. TPn is a tape printer which can print the letters of Σ_n. TPn has n input wires $\bar{w}_1, \cdots, \bar{w}_n$, and prints σ_k at time t if and only if $\bar{w}_a(t) = k\langle a\rangle$ $(a = 1, \cdots, n)$.

DEFINITION 6. CO is a cut-out with one input wire. The activation of this wire forces CO to cause $TS'n$, TPn and TM to cease firing. With this preparation, we may now state and prove our 'equivalence' theorem for neural nets and Turing machines.

THEOREM 2. *Given a net N with n inputs, if we choose any $n+3$ outputs of N,* *then we can find a Turing machine Z whose function is equivalent to that of N, in that if we identify the inputs of N with the output wires of $TS'n$, n of the outputs of N with the input wires of TPn, two of the remaining output wires of N with the input wires of TM, and the last output of N with the input wire of CO, then the machine so obtained behaves exactly as Z; and vice versa.*

PROOF. For the purpose of this proof we shall:

(a) Follow Turing [7], rather than Davis [3], and define a Turing machine as a set of quintuples $q_i S_j S_k D q_l$, each such to be interpreted: if the machine is in state q_i and the symbol scanned is S_j, then print S_k, execute D (where D can take the values: L, move the tape one square to the left; R, move the tape one square to the right; N, do not move the tape) and change the state of the machine to q_l.

(b) Write s_i for q_i, σ_j for S_j, in consonance with the notation of Section 1.

Part I. Let the net N be given with n inputs $i_1, \cdots, i_n$, q delay outputs $d_0, \cdots, d_{q-1}$, and the chosen outputs $o_1, \cdots, o_n, o_L', o_R', o''$.

We connect the w_a of $TS'n$ to the i_a of N, and the o_a of N to the $\bar{w}_a$ of TPn $(a = 1, \cdots, n)$, the o_a' of N to the a of TM $(a = L, R)$ and o'' to the input wire of CO.

Clearly, the symbol σ_j scanned at time t and the internal state of N at time t together determine the state of the $n+3$ chosen outputs at time t, and (by Lemma 1) the internal state s_l of N at time $t+1$. This being so, we define the Turing machine Z as follows:

If $o''(t) = 1$ whenever the symbol-state pair is $s_i\sigma_j$ at time t, then Z contains no quintuple starting with $s_i\sigma_j$.

If $o''(t) = 0$ whenever the symbol-state pair is $s_i\sigma_j$ at time t, and then the state of the remaining outputs is such that TPn prints σ_k at time t, and TM executes D, then the quintuple $s_i\sigma_j\sigma_k D s_l$ is in Z.

Clearly, Z is *equivalent* to N.

Part II. Let Z be a Turing machine with the internal states and alphabet of $\mathfrak{A}$ of Theorem 1 (II), and choose n and q as in that proof. Let

$$I = \{(i,j) \mid Z \text{ contains a quintuple starting with } s_i\sigma_j\}$$

contain $\bar{\bar{I}}$ elements.

Now construct a net N with n inputs i_{a1} $(a = 1, \cdots, n)$ and $n+3$ outputs o_a $(a = 1, \cdots, n)$, o_L', o_R', o''. N comprises the elements described in 1–5 of Theorem 1 (II) (with the inputs of I_{ij} as described therein) plus the following $n+3$ elements,

(1′) 1 element $o''(\vee, kl - \bar{\bar{I}})$ with one input from each I_{ij} with $(i,j) \notin I$,

(2′) 2 elements $o_a'(\vee, m_a')$ $(a = L, R;\ m_a' \leq kl)$,

(3′) n elements $o_a(\vee, \bar{m}_a)$ $(a = 1, \cdots, n;\ \bar{m}_a \leq kl)$, with the further connections:

$$\begin{Bmatrix} D_{b1} \\ o_a \\ o_a' \end{Bmatrix} \text{ receives one input from each of the } \begin{Bmatrix} m_b \\ \bar{m}_a \\ m_a' \end{Bmatrix} I_{ij}$$

$$\text{for which } s_i\,\sigma_j\,\sigma_k\,D s_l \text{ is in } Z, \text{ and } \begin{Bmatrix} l\langle b\rangle = 1. \\ k\langle a\rangle = 1. \\ a = D. \end{Bmatrix}$$

Then the net N is *equivalent* to Z.

The following rather interesting paragraph is quoted from McCulloch and Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity" [5, p. 129]:

> It is easily shown: first, that every net, if furnished with a tape, scanners connected to afferents, and suitable efferents to perform the necessary motor operations, can compute only such numbers as can a Turing machine; second, that each of the latter numbers can be computed by such a net; and that nets with circles can be computed by such a net; and that nets with circles can compute, without scanners and a tape, some of the numbers the machine can, but no others, and not all of them. This is of interest as affording a psychological justification of the Turing definition of computability and its equivalents, Church's λ-definability and Kleene's primitive recursiveness: If any number can be computed by an organism, it is computable by these definitions, and conversely.

Before discussing this, we must first recall the following definition from Turing [7]:

DEFINITION 7. If a Turing machine is supplied with a blank tape and set in motion, starting from the correct initial state, the subsequence of symbols printed by it which are either 0 or 1 will be called the *sequence* computed by the machine. The real number whose expression as a binary decimal is obtained by prefacing this sequence by a decimal point is called the *number computed by this machine.*

Armed with this definition and Theorem 2, we may now prove precisely stated assertions, which I hope accurately mirror those of McCulloch and Pitts.[2]

The first and second assertions merely constitute a reformulation of Theorem 2.

I am not quite sure what "nets with circles can be computed by such a net" means. Perhaps it is the equivalent of the following statement which follows immediately from Theorem 2 and the discussion of Universal Turing machines contained in Davis [3, Sec. 4.3] or Turing [7, Sec. 6,7]:

> "The numbers computed by nets N with circles may be computed by such a net U, namely one corresponding to a Universal Turing machine, in that if U is provided with effectors and receptors as in Theorem 2, and presented with a tape bearing a coded description of N (i e the description of the corresponding Turing machine Z) then U will compute precisely the number which would be computed by Z "

The last of the above assertions is given precise form in the following definition and theorem.

DEFINITION 8. We shall say that the number a is computed by a net N without receptors and effectors if we can choose an output o of N such that, for $t = 0, 1, 2, \cdots,$ $o(t)$ is the tth bit in the binary decimal of a.

THEOREM 3. *There exists a computable number which cannot be computed by a net N without receptors and effectors.*

PROOF. Let N be any such net. Then the state of any output o at time t is completely determined by the internal state of N at time t, which is itself completely determined by the internal state of N at time $t-1$. Since there are only a finite number of internal states of N, it follows that after an initial 'settling-

[2] Note that every McCulloch-Pitts net is a net in the sense of Copi, Elgot and Wright, but not vice versa. This, however, does not vitiate the following discussion.

down' portion of time that the states of o must become periodic, i.e. any number computed by N must have a recurring binary decimal expansion and is thus rational.

Thus the number e, which Turing [7, Sec. 10] has shown to be computable, cannot be computed by N.

Analogously to Definition 8 we have:

DEFINITION 9. We shall say that a set S of numbers can be computed by a net N without receptors and effectors if we can choose an output o of N such that, for $t = 0,1,2,\cdots$, $o(t) = 1$ if and only if $t \in S$.

We may now prove the analogue of Theorem 3. This is essentially Theorem II of Burks and Wright [1].

THEOREM 4. *There exists a primitive recursive set of numbers which cannot be computed by a net N without receptors and effectors.*

PROOF. Consider the set $S = \{0,1,4,9,16,\cdots\}$. Then

$$t \in S \Leftrightarrow t = 0 \vee \sim \left[\underset{y=0}{\overset{t}{\mathfrak{M}}} (t = y^2) = 0 \right]$$

whence S is primitive recursive, by various results of [3, Sec. 3.4, 3.5]. But, by the argument used in proving Theorem 3, S cannot be computed by any such net N.

The restriction made in the above theorem, that N have no accessories, is indeed necessary, as we shall show by constructing an automaton (which by the method of Theorem 2 is clearly *equivalent* to a net N with accessories and a chosen output) with an output o, and such that $o(t) = 1 \Leftrightarrow t$ is a square. The construction is based on the simple equations:

$$(n+1)^2 - n^2 = 2n+1$$
$$(n+2)^2 - (n+1)^2 = 2n+1 + 2$$

The automaton, $\mathfrak{M}$ say, is a modified Turing machine with an output o. $\mathfrak{M}$ sweeps back and forth along the tape, one square at a time, going two squares further on each sweep. $o(t) = 0$ at each moment of time, save at the end of each sweep. $\mathfrak{M}$ comprises quadruples of the form $q_i S_j (S_k D; S) q_m$, where D is as in Theorem 2; S_j, S_k take the values 1, b (=blank); and S takes the values 0, 1. The quadruple is to be interpreted as: If $\mathfrak{M}$ is in the state q_i and the symbol scanned is S_j at time t, then print S_k on the tape, execute D, set $o(t) = S$, and change the state of $\mathfrak{M}$ to q_m.

At time $t=0$, $\mathfrak{M}$ is in state q_1, and the tape is blank. The quadruples of $\mathfrak{M}$ are:

$q_1 b(bN;1)q_2$	setting up of $\mathfrak{M}$ at time $t=0, 1$
$q_2 b(1N;1)q_3$	
$q_3 1(1L;0)q_3$	leftward sweep
$q_3 b(1L;0)q_4$	
$q_4 b(1N;1)q_5$	
$q_5 1(1R;0)q_5$	rightward sweep

$q_5 b(1R;0)q_6$

$q_6 b(1N;1)q_3$

Clearly, $\mathfrak{M}$ is the required automaton.

3. *Finite Automata and Turing Machines*

In this section we characterise the sets of sequences of positive integers defined by finite automata; deduce that regular sets are recursive; and prove a strengthened form of Kleene's theorem that regular sets are primitive recursive.

Symbolism: If $(n_1,\cdots,n_k)$ is a sequence of positive integers,[3] $\overline{(n_1,\cdots,n_k)}$ shall denote the tape $\bar{n}_1 \cdots \bar{n}_k$, where $\bar{n}$ denotes the tape $\underbrace{1\cdots 1}_{n+1}$ blank of length $n+2$. If Λ_N is the null sequence of positive integers, $\bar{\Lambda}_N$ shall denote the tape whose single square is blank.

Definition 10. A set S of sequences of positive integers is said to be finitely acceptable if there exists a finite automaton $\mathfrak{A}$ such that

$$(n_1,\cdots,n_k) \in S \Leftrightarrow \overline{(n_1,\cdots,n_k)} \in T(\mathfrak{A}),$$

and we shall say that $\mathfrak{A}$ accepts S.

Clearly, $\mathfrak{A}$ provides an effective method for deciding whether or not a sequence is in S. This suggests the following theorem.

Theorem 5. *Any finitely acceptable set S of sequences of positive integers is recursive.*

Proof. We prove this theorem by showing that, given a finite automaton $\mathfrak{A}$ which accepts S, we can construct a Turing machine $\mathfrak{A}_{TM}$ such that

$$\Psi^{(k)}_{\mathfrak{A}_{TM}}(n_1,\cdots,n_k) = C_S((n_1,\cdots,n_k)).$$

Let $\mathfrak{A} = (\bar{S},M,s_0,F)$ have alphabet Σ including b (=blank) and 1. Let:

$$\bar{S} = \{s_0,s_1,\cdots,s_{k-1}\}$$

$$\left.\begin{aligned} M(s_i,b) &= s_{i_b} \\ M(s_i,1) &= s_{i_1}\end{aligned}\right\} \quad 0 \leqq i_b,i_1 < k$$

Then the required Turing machine $\mathfrak{A}_{TM}$ has the internal states $q_1, \cdots, q_5$, $s_0, \cdots, s_{2k-1}$, alphabet $\Sigma_n\{\lambda, \mu\}$, and is the set of quadruples:

q_1	any	L	q_2	
q_2	b	λ	q_2	
q_2	λ	R	s_0	
s_i	1	R	s_{i_1}	$0 \leqq i < k$
s_i	b	R	s_{k+i}	
s_{k+i}	1	1	s_{i_b}	
s_{k+i}	b	μ	s_{i_b}	
s_i	μ	b	q_3	if $s_i \in F$
s_i	μ	1	q_3	if $s_i \notin F$
q_3	b or 1	L	q_4	
q_4	not b or λ	b	q_4	
q_4	b	L	q_4	
q_4	λ	b	q_5	

[3] 0 included.

This theorem suggests that, under the representation of Definition 10, any regular set of sequences of positive integers is recursive. However, Kleene [4, Theorem 8] has essentially stated (with only the barest outline of a proof) that such regular sets are primitive recursive relative to the input predicates, so long as there is an upper bound for the values of the integers. I give below a full proof of an even stronger result, using the methods of Davis [3, Sect. 3.4, 3.5, 4.1] in which the upper bound of Kleene's result is removed. In the following theorem we are essentially considering regular sets on alphabets with a denumerable infinity of letters.

DEFINITION 11. The Gödel number, $\mathrm{gn}\{(n_1,\cdots,n_k)\}$, of the sequence $(n_1,\cdots,n_k)$ of positive integers is defined to be

$$\prod_{j=1}^{k} \mathrm{Pr}\,(j)^{n_j+1}.$$

We set $\mathrm{gn}\{\Lambda_N\} = 1$.

THEOREM 6. *All regular sets of sequences of positive integers are primitive recursive, in the sense that if E is such a set, then the set E' is primitive recursive, where*

$$x \in E' \Leftrightarrow [x = \mathrm{gn}\{(n_1,\cdots,n_k)\} \wedge (n_1,\cdots,n_k) \in E].$$

PROOF. Clearly, any set whose sole member is 2^n, for some integer n, is primitive recursive.

Assume now that E' and F' are primitive recursive. Then $(E \vee F)' = E' \vee F'$, and is thus primitive recursive. Clearly

$$x \in (E\cdot F)' \Leftrightarrow \bigvee_{y,z=1}^{x}\{x = y*z \wedge y \in E' \wedge z \in F'\}.$$

Thus

$$\begin{aligned} C_{(E\cdot F)'}(x) &= \prod_{y,z=1}^{x} \alpha\{\alpha(|x - y*z|)\cdot\alpha(C_{E'}(y))\cdot\alpha(C_{F'}(z))\} \\ &= \bar{R}\{C_{E'}(y);C_{F'}(z)\}, \quad \text{say.} \end{aligned}$$

Hence $(E\cdot F)'$ is primitive recursive.

We now define the function $E(n,x)$ to be $C_{(E^n)'}(x)$ for each n. Thus

$$\begin{aligned} E(0,x) &= \alpha(\alpha(|x-1|)) \\ E(n+1,x) &= \bar{R}\{C_{E'}(y);E(n,z)\}. \end{aligned}$$

Thus $E(n,z)$ is primitive recursive. But

$$C_{(E^*)'}(x) = \prod_{n=0}^{x} E(n,x)$$

Hence $(E^*)'$ is primitive recursive. Our theorem then follows from the inductive definition of regular set.

Finally, we show that finitely acceptable sets of positive integers constitute a *proper* subset of the primitive recursive sets of positive integers.

The following theorem is similar to Theorem 4, and can be proved by a modification of the proof of that theorem, using Theorem 1.[4] However, we here proceed otherwise.

THEOREM 7. *There exists a primitive recursive set of positive integers which is not finitely acceptable.*

PROOF. Let $\mathfrak{A} = (S,M,s_0,F)$ be a finite automation with $m+1$ internal states $s_0, s_1, \cdots, s_m$. Then if we present $\mathfrak{A}$ with a tape of the form $1^{m+1}x$ there must be $0 \leqq k < 1 \leqq m+1$ with

$$M(s_0, {}_0x_k) = M(s_0, {}_0x_1).$$

The operation of $\mathfrak{A}$ may then be represented as

$$1^k \overset{s_{i_1}}{} 1 \overset{s_{i_2}}{} 1 \overset{\cdots}{\cdots} 1 \overset{s_{i_{l-k}}}{} 1^{m+1-l}x.$$

Now consider any $n \geqq k$, where

$$n \equiv n' + k - 1 \pmod{l-k} \qquad (1 \leqq n' \leqq l-k).$$

Then $M(s_0, 1^n) = s_{i_{n'}}$. Thus, if

$$\{s_{i_{j_1}}, \cdots, s_{i_{j_r}}\} = \{s_{i_1}, \cdots, s_{i_{l-k}}\} \cap F,$$

the set of integers $\geqq k$ which are accepted by $\mathfrak{A}$ is

$$\{k - 1 + i_{j_a} + b(l-k) \mid a = 1,2,\cdots,r; \quad b = 0,1,2,\cdots\}.$$

Thus $\mathfrak{A}$, which was chosen arbitrarily, cannot accept the primitive recursive set

$$S = \{0, 1, 4, 9, 16, \cdots\}.$$

The theorem is proved.

REFERENCES

1. BURKS, A. W., AND WRIGHT, J. B Theory of logical nets. *Proc. IRE 41*, No 10 (1953), 1357–1365.
2 COPI, ELGOT AND WRIGHT Realization of events by logical nets *J. ACM 5* (1958), 181–196.
3. DAVIS, MARTIN *Computability and Unsolvability*. McGraw-Hill, New York, 1958.
4. KLEENE, S. C. Representation of events in nerve nets and finite automata. In *Automata Studies*, ed by C. E. Shannon and J. McCarthy, Princeton University Press, 1956, pp. 3–41.
5. MCCULLOCH, W. S., AND PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys 5* (1943), 115–133.
6. RABIN, M. O., AND SCOTT, D. Finite automata and their decision problems *IBM J., Res. Dev. 3* (1959), 114–125.
7. TURING, A. M On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* (2), *42* (1936–7), 230–265; with a correction, Ibid. *43* (1947), 544–546.

[4] This theorem has been proved by Kleene, and is clearly implied in one of the results of Rabin and Scott. However, we repeat it here with an explicit proof for completeness of exposition.