数理解析研究所講究録
第 454 巻 1982 年 94-109

Pictures of functions and their acceptability by automata

Akira Nakamura

Department of Applied Mathematics
Hiroshima University
Hiroshima, 730 Japan

Hiroakira Ono

College of Integrated Arts and Sciences
Hiroshima University
Hiroshima, 730 Japan

1. Introduction

Pictorial pattern recognition is a well-established and active field, with numerous applications. In this paper, we consider some interesting problems in this field from the mathematical standpoint. First, we define pictures of functions of one variable in the natural number theory as patterns of input symbols {0,1} in the cell space of the first quadrant. That is, the picture of function $y=f(x)$ is a pattern in which a cell $(x,f(x))$ is 1 and the other cells are all 0. This is a reasonable definition in the digital consideration.

The set of pictures of *all* functions is denoted by F and called the trivial set. It is not so difficult to define a kind of automaton accepting the F. In this paper, we consider an automaton which accepts the set of pictures of all computable functions of one variable. Since a picture of a computable function is not finite but infinite in its size (i.e., the picture spreads out in the first quadrant), it is necessary to introduce a different concept of acceptability from the usual one. To this end, we introduce

three-dimensional automata which are defined as a generalization of that on ω-tapes in [1].

Now, let C be a set of pictures of all computable functions of one variable in the natural number theory, and D be a nontrivial set of pictures of functions containing a noncomputable function.
(The definition of D will be given in the section 3.)
The first purpose of this paper is to show that the set C consists of patterns in the bottom planes of input arrays accepted by a nondeterministic three-dimensional automaton on ω^3-tapes. Then, it is proved that D is deterministically obtained in a similar way. That is, it is shown that D is the set of pictures in the bottom planes of three-dimensional input arrays accepted by such a deterministic automaton. Since D is not the trivial set, it seems that this is a surprising fact.

Let G_f be a picture of function f in the natural number theory. An initial segment of the picture G_f is generally defined as an initial rectangular part of this picture. The set of all initial segments of picture G_f is denoted by $[G_f]_I$. Also, this definition is generalized to a set S of pictures of functions by defining $[S]_I=\{[G_f]_I \mid G_f \in S\}$. Although the picture of a function is infinite in its size, we can finitely treat it by considering all its initial segments. In this case, we are able to use the usual concept of array acceptors such as Turing rectangular array acceptors defined in [4]. The third purpose of this paper is to prove that $[C \cup D]_I$ is accepted by a *deterministic* Turnig rectangular array acceptor. This result also seems to be of interest.

2. Definitions

In this paper, we consider exclusively a finite set Σ of two input symbols 0 and 1 (i.e., $\Sigma=\{0,1\}$).

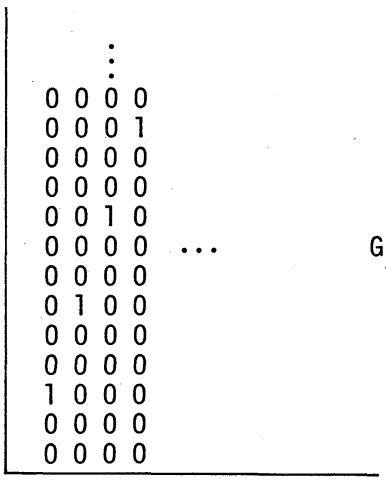
Definition 2.1 A two-dimensional tape of infinite size is a mapping $\sigma: \omega \times \omega \rightarrow \Sigma$ where ω is the set of positive integers, and a three- dimensional tape of infinite size is a mapping $\rho: \omega \times \omega \times \omega \rightarrow \Sigma$ where ω is the same as before.

Hereafter, we say simply a two-dimensional (or three-dimensional) tape in omission of "of infinite size". In the above definition, ω denotes the set of all positive integers. Even if ω represents the set of all non-negative integers, the essential point of our theory is the same. Also, functions of this paper mean total functions.

Definition 2.2 A picture of a function $y=f(x)$ is a two-diemnsioanl tape in which every cell $(x,f(x))$ is 1 and all other cells are 0.

The picture of the Definition 2.2 is denoted by G_f. For example, the picture of $y=3x$ is as follows:

Fig. 1



As mentioned before, (0,f(0)) is neglected.

Definition 2.3 A nondeterministic automaton on ω^3-tapes is a 9 tuple $(q_0,Q,\tau_x,\tau_y,\tau_z,\tau_{xy},\tau_{yz},\tau_{xz},\tau)$ satisfying the following conditions:

(1) $q_0 \in Q$ and q_0 is the initial state,

(2) Q is the finite set of states,

(3) $\tau_x : Q \times \Sigma \longrightarrow 2^Q$

$\tau_y : Q \times \Sigma \longrightarrow 2^Q$

$\tau_z : Q \times \Sigma \longrightarrow 2^Q$

$\tau_{xy} : Q \times Q \times \Sigma \longrightarrow 2^Q$

$\tau_{yz} : Q \times Q \times \Sigma \longrightarrow 2^Q$

$\tau_{xz} : Q \times Q \times \Sigma \longrightarrow 2^Q$

$\tau : Q \times Q \times Q \times \Sigma \longrightarrow 2^Q$.

An automaton on ω^3-tapes is called deterministic if every transition function of (3) in the Definition 2.3 is a mapping into Q.

Informally, τ_x corresponds to the usual transition function along the x-axis, and τ_y, τ_z are similar. τ_{xy} is the transition function of the xy-plane in the on-line tessellation acceptor in [2], and τ_{yz}, τ_{xz} are similar. See the Fig.2 .

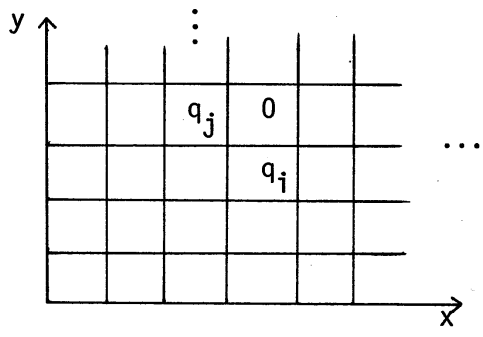


Fig.2

q_i is a state of a cell (m+1,n,1) and q_j is a state of a cell (m,n+1,1), and 0 is an input symbol of the a cell (m+1,n+1,1). Then, a state of the cell (m+1,n+1,1) is determined by $\tau_{xy}(q_i,q_j,0)$. τ is the transition function of three-dimensional on-line tessellation acceptor. See Fig.3 .

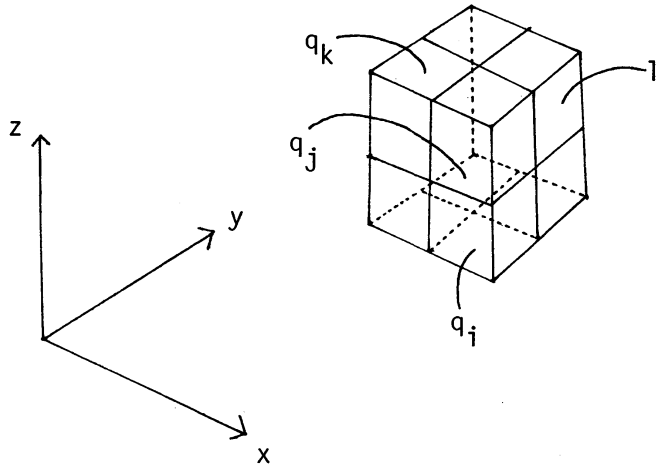


Fig.3

q_i,q_j,q_k are states of the cells $(m+1,n,\ell),(m,n+1,\ell),(m,n,\ell+1)$, respectively. 1 is an input symbol of the cell $(m+1,n+1,\ell+1)$. Then, a state of the cell $(m+1, n+1, \ell+1)$ is determinied by $\tau(q_i,q_j,q_k,1)$.

Note that an automaton on ω^3-tapes is a variant of the corresponding definition for the two-dimensional on-line tessellation acceptor [2] and also it corresponds to the finite causal ω^2-systems [3] .

Definition 2.4 For every $s \in \Sigma^{\omega\times\omega\times\omega}$, the run r over s of an automaton A on ω^3-tapes is recursively defined as $r \in Q^{\omega\times\omega\times\omega}$ satisfying the following conditions:

For every i,j,k,

(1) $r(1,1,1,) \in \tau(q_0,q_0,q_0,s(1,1,1))$

(2) $r(i+1,1,1) \in \tau_x(r(i,1,1),s(i+1,1,1))$

$r(1,j+1,1) \in \tau_y(r(1,j,1),s(1,j+1,1))$

$r(1,1,k+1) \in \tau_z(r(1,1,k),s(1,1,k+1))$

(3) $r(i+1,j+1,1) \in \tau_{xy}(r(i+1,j,1),r(i,j+1,1),s(i+1,j+1,1)$

$r(1,j+1,k+1) \in \tau_{yz}(r(1,j+1,k),r(1,j,k+1),s(1,j+1,k+1))$

$r(i+1,1,k+1) \in \tau_{xz}(r(i+1,1,k),r(i,1,k+1),s(i+1,1,k+1))$

(4) $r(i+1,j+1,k+1) \in \tau(r(i+1,j,k),r(i,j+1,k),r(i,j,k+1),s(i+1,j+1,k+1))$.

The run r is a three- dimensional infinite array of states which is obtained by successively applying the transition functions of automaton A to s. We use the notation Rn(A,s) to represent the set of all runs of automaton A to s.

Definition 2.5 Let ψ be a mapping from A into B. Then, In (ψ) is defined as follows:

$$In(\psi) = \{b \mid \psi^{-1}(\{b\}) \text{ is infinite}\}.$$

Definition 2.6 $s \in \Sigma^{\omega \times \omega \times \omega}$ is said to be accepted by an acceptor A=(A,F) on ω^3-tapes iff the following conditions are satisfied:

(1) A is an automaton on ω^3-tapes,

(2) $F \subseteq Q$ is the set of accepting states,

(3) $\exists r(r \in Rn(A,s) \ \& \ In(r) \cap F \neq \phi)$.

The above definition of acceptability is a generalization of that called B-automaton on ω-tapes in [1]. Since the picture of function in the natural number theory is not finite but infinite in its size, we use Definition 2.6 concerning its acceptability. It is mathematically of interest to consider input arrays of infinite size. If we want to discuss pictures of functions from the engineering side and develop the standard theory, other definitions

are necessary. The best way to this end is to consider the initial segment of input of infinite size. An initial segment of two-dimensional tapes with infinite size is generally defined as follows:

Definition 2.7 An initial segment of two-dimensional tape with infinite size is its partial array over cells (x,y) satisfying the following (1),(2), or (3):

For some m,

(1) $1 \leq x \leq m$ and $1 \leq y \leq \max_{1\leq i\leq m} \mu_j[(i,j)=1]$ if $\max_{1\leq i\leq m} \mu_j[(i,j)=1] \geq m$,

(2) $1 \leq x \leq m$ and $1 \leq y \leq m$ if $\max_{1\leq i\leq m} \mu_j[(i,j)=1] < m$,

(3) $1 \leq x \leq m$ and $1 \leq y$ if $\exists i(1\leq i\leq m \ \& \ \forall j(i,j)=0))$,

where μ of the above (1)-(2) is the μ-operation in the number theoty.

For the case (3) of Definition 2.7, the initial segment of given tape is infinite in the direction of y-axis. Also, it is easily known that every initial segment of G_f of function is a rectangular array. We call the m-th initial segment for the Definition 2.7.

The (m+1)-th initial segment is called the next of the m-th one.

For example, see Fig. 4 and Fig. 5:

Fig. 4

0	0	0	1
0	0	0	0
0	0	1	0
0	0	0	0
0	1	0	0
0	0	0	0
1	0	0	0
0	0	0	0

4-th initial segment of G_{2x}

Fig. 5

0	0	0	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	0	0
0	0	0	0	0

5-th initial segment of G_f,

$f(x)=\begin{cases} 2x & \text{if } x\leq 2 \\ 3 & \text{if } x\geq 3 \end{cases}$

For the two-dimensional tape P, the set of its initial segments for every m is denoted by $[P]_I$. Also, this definition is generalized to the set S of two-dimensional tapes by defining $[S]_I=\{[P]_I \mid P \in S\}$.

The usual Turing acceptors (TA's) in the one-dimensional input are well-known. Turing rectangular array acceptors (TRAA's) are defined as a natural generalization of TA. That is, it is defined as an acceptor over rectangular array inputs which can move "left", "right", "up", "down", or "no move". Since the details of TRAA's are described in [4], we do not review it here but give only the formal definition.

Definition 2.8 An array automaton M is a triple (Q,V,δ), where

Q is the set of states,

V is the set of symbols,

δ: $Q\times V\times\Delta \longrightarrow 2^{Q\times V\times\Delta}$ (or $\longrightarrow Q \times V \times \Delta$, in the deterministic case) is the transition function, and

$\Delta\equiv\{L,R,U,D,N\}$ ("left","right", "up", "down", "no move") is the set of move directions.

Definition 2.9 A Turing rectangular array acceptor is a triple (M,q_0,Q_A) over rectangular array inputs, where M is an array automaton, q_0 is M's initial state and Q_A is a set of accepting states.

Acceptability by a TRAA is defined in the uaual way.

3. Theorems

Let us now consider a two-dimensional pattern $r \in \Sigma^{\omega\times\omega\times\{1\}}$. When this r is the restriction of an $s \in \Sigma^{\omega\times\omega\times\omega}$, r is called the bottom picture of s.

To prove one of the main theorem of this paper, we give some preliminary lemmas. Let f(x) be a computable function and R_f be the set of rectangular arrays over {0} whose lengths of column and row are m and f(m) for some m, respectively. See Fig. 6 .

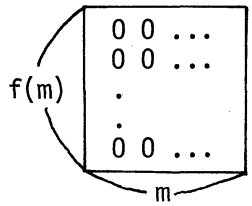


Fig.6

Lemma 3.1 There is a deterministic TRAA which accepts R_f.

Proof:

Let us consider an element $\pi \in R_f$ whose lengths of column and row are m and f(m), respectively. Since f is a computable function, there exists a Turing machine M_1 which computes this f. Thus, there is a sequence of configurations $a_1, a_2, \ldots, a_s$ which represents this computation. Now, we define a TRAA M_2 as follows: At first, m cells (m,1),(m,2),...,(m,m) are marked which correspond to the value m of variable x. Then, by making use of cells (m,1), (m,2),...,(m,m),.... this TRAA M_2 simulates the above M_1. Finally, M_2 accepts an element π of R_f if and only if the result by M_1 equals to f(m). //

Note that the TRAA of Lemma 3.1 does not use the west, east, and south blank spaces.

Lemma 3.2 There is a deterministic TRAA which accepts $[G_f]_I$, where f is a computable function.

Proof:

Every element of R_f consists of symbol 0 only as shown in Fig.6 . As

compared with this R_f, G_f consists of pictures such that every cell (i,f(i)) is 1 and all other cells are 0. Thus, by making use of the M_2 of Lemma 3.1, we define a TRAA M_3 as follows:

i) M_3 simulates the M_2 for every column i,

ii) The correctness of i-th column is ensured only after the correctness of (i-1)-th column,

iii) M_3 goes to an accepting state at the upper-right-corner of every m-th initial segment.

From the above consideration, we get this lemma. //

By the same way, $[F]_I$ is accepted by a TRAA where F is the set of pictures of all functions.

In the previous lemmas, inputs were the set of rectangular arrays. From now on, we will consider the case of three-dimensional tapes as inputs.

Theorem 3.3 (The first main theorem) There is a nondeterministic acceptor $A=(A,F)$ on ω^3-tapes as follows:

The set C of pictures of all computable functions of one variable is exactly the set of bottom pictures of three-dimensional input arrays accepted by this A.

Proof:

At the beginning of the proof, we consider the sequence of configurations which leads to the acceting state q_a for a $\pi_0 \in [G_f]_I$ in Lemma 3.2. Let the sequence denote $\pi_0, \pi_1, \ldots, \pi_k$. From this sequence, we construct a three-dimensional array as follows in Fig.7 . That is, π_0 is written on a plane which is parallel to the xy-plane, and π_{i+1} is aslant put over π_i. In this case, the z-axis corresponds to the time step. As seen later, it is very important that π_{i+1} is aslant put over π_i.

Let H be the set of three-dimensional input arrays h's whose every bottom

represents a picture $G_f \in C$ and h (i,j,k) is arbitrary for $i \geq 1$, $j \geq 1$, $k \geq 2$.

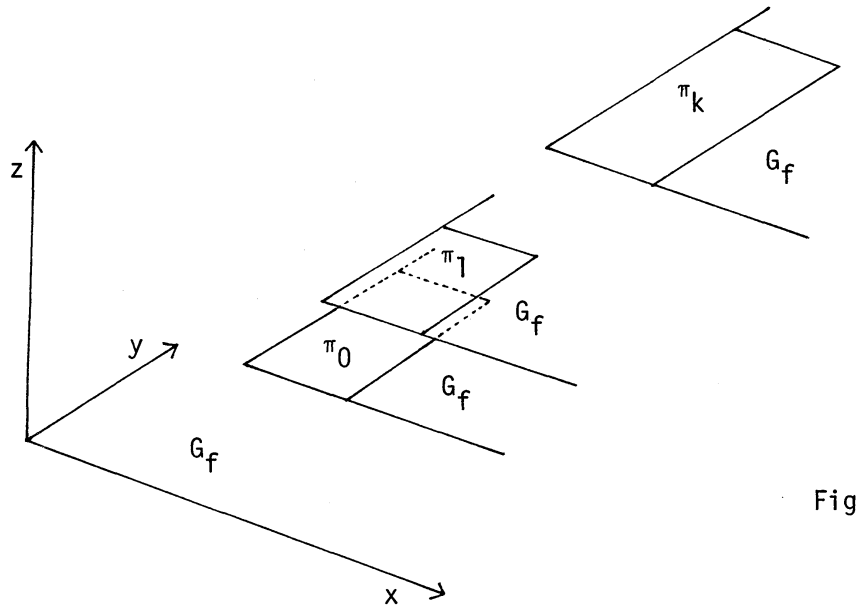


Fig. 7

We will now define a nondeterministic acceptor $A=(A,F)$ on ω^3-tapes which accepts the set H. At first, A checks whether or not a certain initial segment of bottom of given input array is correct. The checking of m-th initial segment is done as follows: Firstly, A marks a special symbol over all cells of this initial segment. In this case, when this initial segment is infinite in its size (see (3) of Definition 2.7) A continue to mark the symbol and so A never fall into an aceepting state q_a. After terminating to mark, A begins the following act. A checks again whether or not this initial segment represents a partial picture of a function. Then, by nondeterministically guessing A writes a Turing machine (exactly say, the code of a Turing machine) to compute a function f on cells $(1,1),(1,2),\ldots,(1,\lambda)$. This code is fixed in the subsequent bihaviors of A. According to this code, A checks again whether or not this initial segment is correct. In this case, A acts as the universal Turing machine. A writes states at every cell such that the above-mentioned π_{i+1} follows to π_i. By shifting π_{i+1} by one cell to

the x-axis and the y-axis from the preceding π_i, A can write every configuration in the fashion of three-dimensional on-line. Also, let us assume that the given bottom input is always written on the every plane containing π_i. (See Fig. 7.) This is possible, because at every time step the shifting is done as mentioned before. This method is analogous to the technique used in [3].

If a cell (x,y,1) in the m-th initial segment is illegally 0, A falls into the dead state q_d at some cell. Also if a cell (x,y,1) in this segment is illegally 1, A falls into the dead state q_d. After once fallen into the dead state, A always write the dead state q_d. If a cell (x,y,1) is legally 1 (i.e., y=f(x)) and the m-th initial segment is right, A ever falls into the accepting state q_a at some cell of three-dimension and the A begins to check the next initial segment. Further, if the previous mentioned code of a Turing machine is incorrectly guessed, then A falls into the dead state q_d. Notice that A can do this checking. Also, A never fall into the accepting state q_a for the case that A does not halt.

Thus, it is sufficient for the proof of this theorem to define an acceptor on ω^3-tapes as $A=(A,\{q_a\})$. Because, it is easily shown that for every h $\exists r(r \in Rn(A,h)$ & $In(r)\cap\{q_a\}\neq\phi)$ iff $h \in H$. //

It is shown by the similar method to Theorem 3.3 that there is an acceptor A on ω^3-tapes as follows: The set F of pictures of all functions of one variable is exactly the set of bottom of three-dimensional array accepted by this A.

Problem Is there a *deterministic* acceptor $A=(A,F)$ on ω^3-tapes which accepts the set H ?

To prove the second main theorem of this paper, we give here a lemma. Let us consider the number ν which is obtained from the code of Turing

machine computing a computable function f of one variable. This is easily obtained as follows: If the number of symblos used in coding of Turing machine is L, ν is defined as a L-ary number. By making use of a code number of a Turing machine, every computable function of one variable is denoted by $f_\nu(x)$. Obviously, all code numbers can be linearly arranged such that $\nu_1 < \nu_2 < \ldots < \nu_i < \nu_{i+1} < \ldots$. Let us denote the set of all code numbers by C_N. We consider a subset $T \subseteq C_N$. If T is a finite set, then $T=\{\nu_{n_1}, \nu_{n_2}, \ldots, \nu_{n_k}\}$ where $\nu_{n_i} < \nu_{n_{i+1}}$, and if T is an infinite set, then $T=\{\nu_{n_1}, \nu_{n_2}, \ldots\}$ where $\nu_{n_i} < \nu_{n_{i+1}}$. Then, by making use of a set T we define a function $q_T(x)$ in the natural number theory as follows:

The case of a finite set $T=\{\nu_{n_1}, \nu_{n_2}, \ldots, \nu_{n_k}\}$:

$$g_T(x) = \begin{cases} 1 & \text{if } 1 \leq x < \nu_{n_i} \\ f_{\nu_{n_i}}(\nu_{n_i})+1 & \text{if } \nu_{n_i} \leq x < \nu_{n_{i+1}} \\ f_{\nu_{n_k}}(\nu_{n_k})+1 & \text{if } \nu_{n_k} \leq x \ . \end{cases}$$

The case of an infinite set $T=\{\nu_{n_1}, \nu_{n_2}, \ldots\}$

$$g_T(x) = \begin{cases} 1 & \text{if } 1 \leq x < \nu_{n_1} \\ f_{\nu_{n_i}}(\nu_{n_i})+1 & \text{if } \nu_{n_i} \leq x < \nu_{n_{i+1}} \end{cases} .$$

Now, we define the set V as follows: $V=\{g_T(x) \mid T \subseteq C_N\}$. Further, let D be the set of pictures of elements of V.

Lemma 3.4 The set D contains a picture of a noncomputable function. Also the set D is not the trivial set.

Proof:

Let us consider the function $q_{C_N}(x)$. It is obvious that $g_{C_N}(x)$ is an element of V. Let us assume that $g_{C_N}(x)$ is a computable function. Then, it

must be computable by a Turing machine whose code number is a ν_i. Thus, $g_{C_N}(x)$ can be denoted by $f_{\nu_i}(x)$. Therefore, we have $g_{C_N}(\nu_i)=f_{\nu_i}(\nu_i)$. But, $g_{C_N}(\nu_i)=f_{\nu_i}(\nu_i)+1$ from the definition. This is a contradiction. Hence, we know that $g_{C_N}(x)$ is not a computable function.

Also, it is easy to prove that the set D is not the trivial set. //

Theorem 3.5 (The second main theorem) There is a deterministic acceptor $A=(A,F)$ on ω^3-tapes as follows:

The set D is exactly the set of bottom pictures of three-dimensional input arrays accepted by this A.

Proof:

First of all, let us define a characteristic point of picture of function f as follows: A cell (x,f(x)) is said a characteristic point iff $f(x)\neq f(x-1)$. (See Fig.8 .)

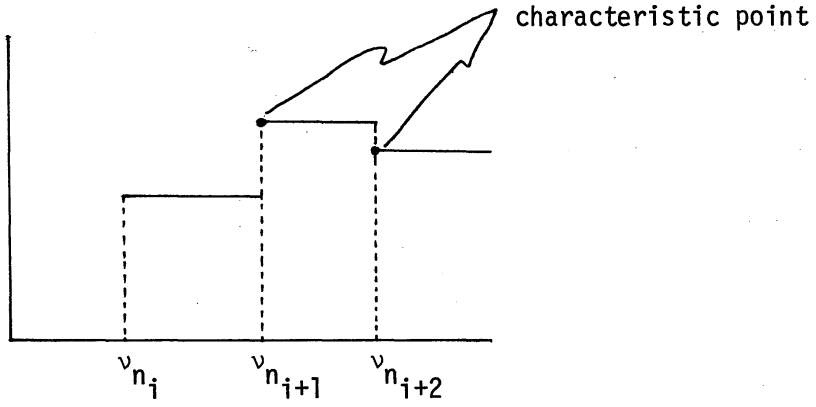


Fig.8

The proof of this theorem is similar to Theorem 3.3 . A checks whether or not a m-th initial segment of bottom of given input represents a partial picture of a function. If this segment represents a function, A searches a characteristic point contained in it. Obviously, A can do it. By making use of every characteristic point (x,y,1), A can deterministically accepts

this initial segment. The acceptance is done as follows: A can encode a Turing machine from the code number x of a characteristic point and then A acts as the universal Turing machine according to this code. For a characteristic point (x,y,1), if $y=f_x(x)+1$ then A falls into the accepting state q_a. For non-characteristic point (x,y,1)=1, A falls into q_a iff (x-1,y,1)=(x,y,1)=1.

After terminating an acceptance of the m-th initial segment, A goes the next initial segment, and then A repeats this process. Of course, the acceptance of (m+1)-th initial segment is ensured only after the acceptance of m-th initial segment.

The other parts of proof are quite similar to Theorem 3.3. That is, we define $A=(A,\{q_a\})$. Because, it is shown that for every given input h, $\exists r(r\in Rn(A,h)\ \&\ In(r)\wedge\{q_a\}\neq\phi)$ iff a bottom of h is in the set D. Further, it is obvious from the above construction of A that A is deterministic. //

Let us now give the final theorem of this paper.

Theorem 3.6 (The third main theorem) Let C and D be the same as before. $[C\vee D]_I$ is accepted by *deterministic* TRAA.

Proof:

It is easily proved from the previous discussion that $[C\vee D]_I$ is accepted by a nondeterministic TRAA. However, it is well-known that deterministic TRAA's are equivalent to nondeterministic TRAA's with respect to the acceptability. //

In the same consideration to Theorem 3.6, it is obvious that $[C]_I$ and $[D]_I$ are accepted by deterministic TRAA's.

Acknowledgement

A. Nakamura, one of the authors, would like to thank Professor Azriel Rosenfeld of University of Maryland for valuable comments.

References

[1] Y. Choueka: Theories of automata on ω-tapes: A simplified approach, J. Computer and System Sciences, 8 (1974) 117-141.

[2] K. Inoue and A. Nakamura: Some properties of two-dimensional on-line tessellation acceptors, Information Sciences, 13 (1977) 95-121.

[3] A. Nakamura: On causal ω^2-systems, J. computer and System Sciences, 10 (1975) 225-265.

[4] A. Rosenfeld: Picture Languages, Academic Press, New york, 1979 .