数理解析研究所講究録
353巻 1979年 266-275

266

SPACE COMPLEXITY ON TWO-DIMENSIONAL CONNECTED TAPES

HIROSHI UMEO, KENICHI MORITA AND KAZUHIRO SUGATA

FAC. OF ENGINEERING SCIENCE, OSAKA UNIVERSITY

ABSTRACT A concept of two-dimensional space complexity has played an important role in the study of two-dimensional tape automata on rectangular inputs. Recently automata on arbitrarily shaped two-dimensional connected tapes have been studied by several investigators, but little has been known about them. In this paper we extend the concept of space complexity on rectangular inputs so that it can deal with automata on connected tapes. A connected tape reduction theorem, log|space| connected-tape traversal algorithm, halting theorem, hierarchy theorem, Boolean closure theorem, and a characterization theorem are our main results.

1. INTRODUCTION

Recently there has been increasing interest in the study of two-dimensional tape automata([1]∿[11]). The concept of two-dimensional space complexity proposed by the authors has provided us a unified treatment of two-dimensional tape automata([2],[3],[4]). This concept has been defined on rectangular inputs. On the other hand several investigators have considered automata which operate on arbitrarily shaped two-dimensional arrays([5],[6],[7],[8]). In this paper we extend the concept of space complexity defined on rectangular inputs to the one on arbitra-

rily shaped two-dimensional arrays. Some results established earlier are interpreted in a unified way from a viewpoint of space complexity, and moreover several results are newly obtained. In our considerations, a tape traversal algorithm is essentially made use of.

2. Preliminaries

ADJACENCY AND CONNECTIVITY A two-dimensional plane is divided into cells. (i,j) is assigned to each cell, where i and j are integers(Fig.1). Two cells (i,j) and (i',j') are said to be adjacent if and only if $|i - i'| + |j - j'| \leq 1$. A connectivity of two cells is defined by a reflexive and transitive closure of an adjacency. An array of cells is said to be connected when any two cells on the array are connected.

PATHWISE-CONNECTED TAPE AND SIMPLY-CONNECTED TAPE Let Σ be a set of alphabets. A pathwise-connected tape(P-tape) over Σ is a finite connected array of cells marked with alphabets in Σ, where boundaries of the array are delineated by a special symbol B not in Σ(Fig.2). The width of all boundaries is supposed to be one cell. Note that a pathwise-connected tape has some holes

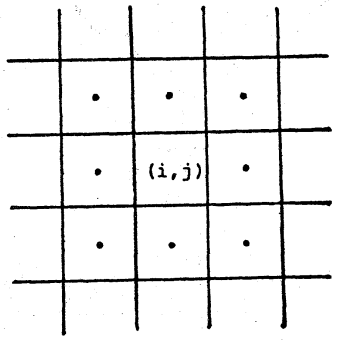

Fig.1 Two-dimensional space.

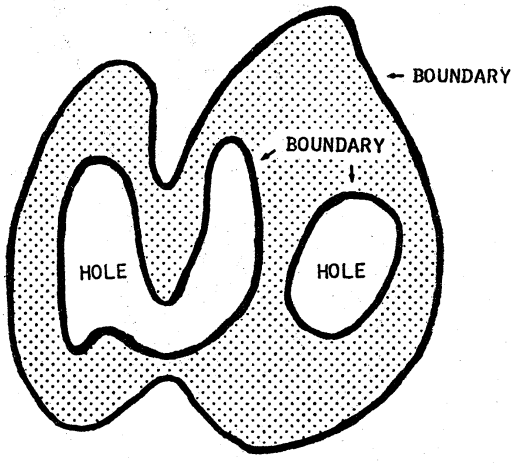

Fig.2 Pathwise-connected tape.

in its interior in general.

Any P-tape which has no holes is called a simply-connected tape (S-tape) (Fig.3). The size of an S(P)-tape x is the total number of cells on x, including its boundary cells.

S(P)-TAPE TURING MACHINE An S(P)-tape Turing machine M is a deterministically operating S(P)-tape acceptor which walks about on S(P)-tape using its storage memories. M consists of an S(P)-tape (input tape), a four-way read-only input head, a finite state control, a two-way read/write storage head and a semi-infinite one-dimensional structured storage tape, as illustrated in Fig.4. Both the input and the storage heads are not allowed to drop off its input and storage tapes, respectively. We do not give an description of operations of M, since they are easily formalized in the same way as those of one- or two-dimensional Turing machines([2]). The set of S(P)-tapes accepted by M is denoted by $T^{S(P)}(M)$. In this paper we consider only deterministically operating machines.

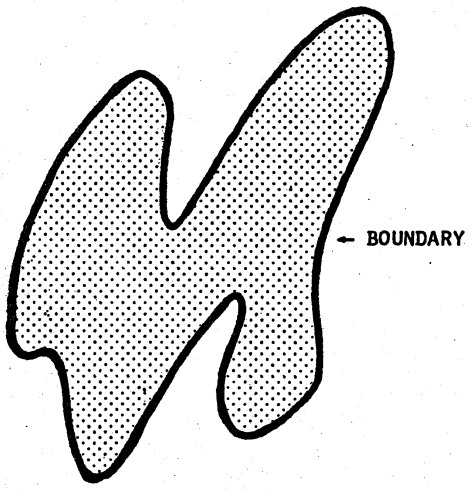

Fig.3 Simply-connected tape.

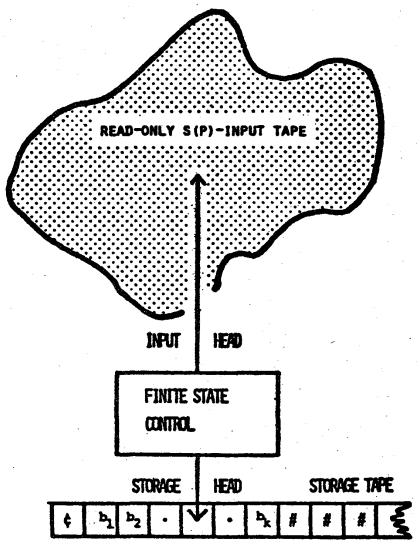

Fig.4 Two-dimensional S(P)-tape Turing machine.

S(P)-TAPE SPACE COMPLEXITY Let M be an S(P)-tape Turing machine, and L(n) be a function from $(N - \{],2,3,\ldots,8\})^{\dagger}$ into $[R^{+}]$, where N and R^{+} are the set of natural numbers and non-negative real numbers, respectively. M is called an L(n) S(P)-tape bounded Turing machine if the storage tape of M cannot be visited more than L(n) tape squares from the left end for any S(P)-tape of size n. The function L(n) is referred to as an S(P)-tape function. The family of all L(n) S(P)-tape bounded Turing machines is denoted by $TM^{s(p)}[L(n)]$. Let

$$\mathcal{L}^{s(p)}[TM - L(n)] = \{T^{s(p)}(M) \mid M \in TM^{s(p)}[L(n)]\}.$$

S(P)-TAPE CONSTRUCTIBILITY An S(P)-tape function L(n) is said to be constructible if and only if there exists an S(P)-tape Turing machine which, for each S(P)-tape of size n, lays off L(n) tape squares using only L(n) tape squares on its storage tape.

3. Space Complexity on Connected Tapes

In this section several properties of $TM^{s(p)}[L(n)]$ are considered from a viewpoint of space complexity. A connected-tape traversal algorithm given by several investigators has played a fundamental role in this study([8],[10],[11]). Let's begin with a tape traversal algorithm for any S(P)-tape.

3.1. S(P)-tape Traversal Algorithm

Tape traversal problem is that "is there an automaton which can halt after visiting all cells on any S(P)-tape, without stepping off, starting from any cell of the tape?" and "by what algorithm is it traversed, if possible?".

---

† Note that $n \geq 9$, since the minimal size of connected tapes is equal to nine.

The authors and Shah have shown independently that any S(P)-tape can be traversed by a finite state automaton supplied with one marker(five markers)† ([8],[11]). A finite state automaton on S(P)-tape supplied k markers, $\{m_1,m_2,\ldots,m_k\}$ is called a k-marker automaton($MA^{s(p)}(k)$)([1],[3],[8]).

The tape traversal algorithm has the following properties.

1. The automaton begins to traverse at a unique point of any S(P)-tape. This unique point is called the traverse starting square.
2. In traversing, the automaton can distinguish the square which has been already scanned from those not scanned.
3. The automaton finishes traversing and halts after returning to the traverse starting square.

It is shown from above observations that an $MA^{s(p)}(1)$ ($MA^{s(p)}(5)$) M can think of any S(P)-tape of size n as a string of length n, and conversely, M can also embed a string of length n into an S- or P-tape of size n. This tape traversal technique is often used in the following.

3.2. Space Complexity on Connected Tapes

A tape reduction theorem for two-dimensional connected tapes follows from the same idea in one-dimensioanl space complexity.

Theorem 1 S(P)-tape Reduction

For any L(n) and any constant c > 0,

$$\mathcal{L}^{s(p)}[TM - L(n)] = \mathcal{L}^{s(p)}[TM - c\cdot L(n)].$$

Lemma 1

For any positive integer k and for any $MA^{s(p)}(k)$, M,

---

† Recently Blum and Kozen have shown that any P-tape can be traversed by a two-marker automaton([10]).

there exists a machine M_1 such that $T^{s(p)}(M_1) = T^{s(p)}(M)$ and $M_1 \varepsilon TM^{s(p)}[\log n]$†.

Proof sketch: The input head of M_1 directly simulates the moves of the finite state control of M. The horizontal and the vertical displacements between the finite state control of M and its markers placed are stored in the storage tape of M_1 in a binary representation. At each step of M, M_1 updates the contents of its storage tape. M_1 can easily know whether a marker is placed or not on the square currently scanned by referrencing its own storage tape. Since the reletive displacements are at most n squares for any S(P)-tape of size n, the length log n storage tape of M_1 is sufficient for the simulation. Q.E.D.

Lemma 2

Log n is S(P)-tape constructible.

From lemma 1 a log|space| tape traversal algorithm is obtained.

Theorem 2 Log|space| tape traversal algorithm

There exists a machine M satisfying the following conditions: (1) $M \varepsilon TM^{s(p)}[\log n]$ and (2) M halts after traversing any S(P)-tape with its read-only input head.

The following is a halting property for $TM^{s(p)}[L(n)]$. Note that we do not need constructibility of L(n).

Theorem 3 Halting

Let L(n) be any S(P)-tape function such that $L(n) \geq \log n$. Then, for any M in $TM^{s(p)}[L(n)]$, there exists a machine M^h satisfying the following conditions: (1) M^h halts for any S(P)-tape, (2) $T^{s(p)}(M^h) = T^{s(p)}(M)$, and (3) $M^h \varepsilon TM^{s(p)}[L(n)]$.

---

† From Th.1 constant factors are irrelevant, so we do not need to specify logarithmic bases.

Proof: First we construct $M^{h'}$ in $TM^{s(p)}[L(n) + \log n]$, satisfying (1) and (2). $M^{h'}$ operates as follows: At first $M^{h'}$ lays off log n tape squares(See lemma 2). While M uses k squares, $M^{h'}$ simulates M on the first track using k squares and concurrently counts the step number of M on the second track using k + log n squares. In the simulation $M^{h'}$ halts when M halts. Next we show how $M^{h'}$ detects the looping conditions of M. Let M have s states and t storage symbols. The number of configurations of M on the array of size n are at most $s \cdot n \cdot t^k \cdot k (=N)$, where k is the length of the storage tape being used. Since $N < (2 \cdot s \cdot t)^{k + \log n}$, for any n, $M^{h'}$ can detect the condition in which M loops by counting the number of moves of M in base $2 \cdot s \cdot t$. Therefore when M does not halt within N steps, $M^{h'}$ halts and rejects the input. From Th.1 It is easily seen that $M^{h'}$ can be reconstructed to satisfy (1), (2) and (3). Q.E.D.

The next theorem establishes an infinite hierarchical relation between $TM^{s(p)}[L(n)]$. We can show the existance of the set L of S(P)-tapes, which is not a set of rectangles, such that $L \notin \mathcal{L}^{s(p)}[TM - L_1(n)]$ and $L \in \mathcal{L}^{s(p)}[TM - L_2(n)]$, for $L_1(n)$ and $L_2(n)$ in Th.4, by the use of conventional diagonalization technique and tape traversal algorithm.

Theorem 4 Hierarchy

Let $L_1(n)$ and $L_2(n)$ be any constructible S(P)-tape functions satisfying the following conditions. Then, $\mathcal{L}^{s(p)}[TM - L_1(n)] \subsetneq \mathcal{L}^{s(p)}[TM - L_2(n)]$.

Conditions: (I) $L_1(n) \leq L_2(n)$, and (II) there exists an infinite sequence i = 1,2,....., such that

(1) $9 \leq n_1 < n_2 < \ldots\ldots, < n_i < \ldots\ldots$

(2) $\lim_{i \to \infty} \frac{L_1(n_i)}{L_2(n_i)} = 0$, (3) $\frac{L_2(n_i)}{\log n_i} > k$, for some $k > 0$.

Halting theorem is applicable to establish the following Boolean closure results.

Theorem 5 Boolean Closure

Let L(n) be any S(P)-tape function such that $L(n) \geq \log n$. Then, $\mathcal{L}^{S(P)}[TM\text{-}L(n)]$ is closed under complement, union, and intersection, and forms a Boolean algebra.

Proof sketch: From Th.2 we can assume without loss of generality that any machine in $TM^{S(P)}[L(n)]$ halts in an accepting or rejecting states for any S(P)-tape. Therefore, for any M in $TM^{S(P)}[L(n)]$, we can easily construct a machine M_1 in $TM^{S(P)}[L(n)]$ such that $T^{S(P)}(M_1) = \overline{T^{S(P)}(M)}$. In the same way we can construct a machine M in $TM^{S(P)}[L(n)]$ such that $T^{S(P)}(M) = T^{S(P)}(M_1) \cup T^{S(P)}(M_2)$ for any M_1 and M_2 in $TM^{S(P)}[L(n)]$. Moreover, $T^{S(P)}(M_1) \cap T^{S(P)}(M_2) = \overline{\overline{T^{S(P)}(M_1)} \cup \overline{T^{S(P)}(M_2)}}$. This completes the proof.Q.E.D.

The last application of space complexity is a computational characterization of two-dimensional tape automata on connected tapes. We consider the following automata. The details of each automaton are omitted(See the references).

(1) FINITE STATE AUTOMATON(FSA)([7],[8]) An FSA, consisting of a read-only input head and a finite state control, is an automaton which walks about on S(P)-tape.

(2) MARKER AUTOMATON(MA)([8],[11]) An MA is just an FSA provided with a finite fixed number of labelled markers which it carries about itself and leaves on squares on the input as temporary markers. Let $MA = \bigcup_{k=1}^{\infty} MA^{S(P)}(k)$.

(3) MULTI-HEAD FINITE STATE AUTOMATON(HA)([8]) A k-head finite state automaton($HA^{S(P)}(k)$) consists of a finite state control and k read-only input heads which operate on S(P)-tape.

Let $HA = \bigcup_{k=1}^{\infty} HA^{s(p)}(k)$.

(4) Array Bounded Automaton(ABA)([5],[6],[7]) An ABA consists of a finite state control and an input head, where this head can read and write symbols on the tape, but its operation is bounded within the input tape.

(5) Bounded Cellular Space(BCS)([5],[9]) A BCS is a parallel operating finite array of identical FSA's called cells, each connected to its four nearest neighbours.

Computational abilities of these automata are characterized as follows(Proofs are omitted):

Theorem 6 Computational Characterization

(1) $\mathcal{L}^{s(p)}[FSA] = \mathcal{L}^{s(p)}[TM - k]$, for some positive constant k.

(2) $\mathcal{L}^{s(p)}[MA] = \mathcal{L}^{s(p)}[TM - \log n]$

(3) $\mathcal{L}^{s(p)}[HA] = \mathcal{L}^{s(p)}[TM - \log n]$

(4) $\mathcal{L}^{s(p)}[ABA] = \mathcal{L}^{s(p)}[TM - n]$

(5) $\mathcal{L}^{s(p)}[BCS] = \mathcal{L}^{s(p)}[TM - n]$.

The following two corollaries are immediate from Th.4, Th.5, and Th.6.

Corollary 6.1 Computational Relation

$$\mathcal{L}^{s(p)}[FSA] \subsetneq \mathcal{L}^{s(p)}[MA] = \mathcal{L}^{s(p)}[HA]$$
$$\subsetneq \mathcal{L}^{s(p)}[ABA] = \mathcal{L}^{s(p)}[BCS].$$

Corollary 6.2 Boolean Closure

$\mathcal{L}^{s(p)}[MA]$, $\mathcal{L}^{s(p)}[HA]$, $\mathcal{L}^{s(p)}[ABA]$ and $\mathcal{L}^{s(p)}[BCS]$ forms a Boolean algebra.

4. Conclusions

In this paper we introduced a concept of space complexity on connected-tapes and then investigated several interesting properties of automata on connected tapes with an aid of this concept.

Further studies are left for the class of machines with tape bounds less than $\lim_{i\to\infty} \frac{L(n_i)}{\log n_i} = 0$, such as "is there a hierarchy in $TM^{s(p)}[L(n)]$?", "can a halting property be assumed for $TM^{s(p)}[L(n)]$?" and so on.

REFERENCES

[1] M.Blum and C.Hewitt;"Automata on a two-dimensional tape",IEEE Symp on SWAT.,p.155,(1967).

[2] K.Morita, H.Umeo,and K.Sugata;"Computational complexity of L(m,n) tape-bounded two-dimensional tapeTuring machines", Trans. IECE of Japan, Vol.60-D, No.11, P.982,(197).

[3] H.Umeo, K.Morita and K.Sugata;"Some closure properties of the class of sets of patterns accepted by deterministic two-dimensional L(m,n) tape-bounded Turing machines", Trans. IECE of Japan, Vol.62-D, NO.1, p.32, (1979).

[4] K.Morita, H.Umeo and K.Sugata;"Language recognition abilities of several two-dimensional tape automata and their relation to tape complexities", Trans. IECE of Japan, Vol.60-D, No.12, p.1077,(1977).

[5] A.R.Smith;"Two-dimensional formal languages and pattern recognition by cellular automata",IEEE Symp on SWAT.,p.144,(1971).

[6] D.L.Milgram and A.Rosenfeld;"Array automata and array grammars", IFIP'71, Conference Proceedings, North-Holland, p.69,(1971).

[7] D.L.Milgram;"A region crossing problem for array-bounded automata", Inf. and Cont.,p.147,(1976).

[8] H.Umeo,K.Morita and K.Sugata;"Sone notes concerning the shapes of two-dimensional input tapes", Trans.IECE of Japan, Vol.60-D, No.8, p.57,(1977).

[9] W.T.Beyer;"Recognition of topological invariants by iterative arrays",Ph.D. Dissertation, MIT,(1969).

[10] M.Blum and W.Kozen;"On the power of the compass", Proc.19th IEEE Annual Symp. On Foundations of Computer Science, Oct. (1978).

[11] A.N.Shah;"Pebble automata on arrays", Computer Graphics and Image Processing,3,p.236,(1974).