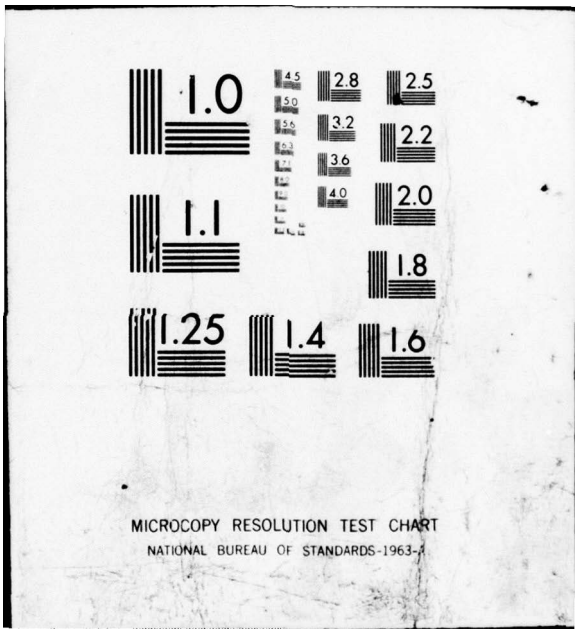
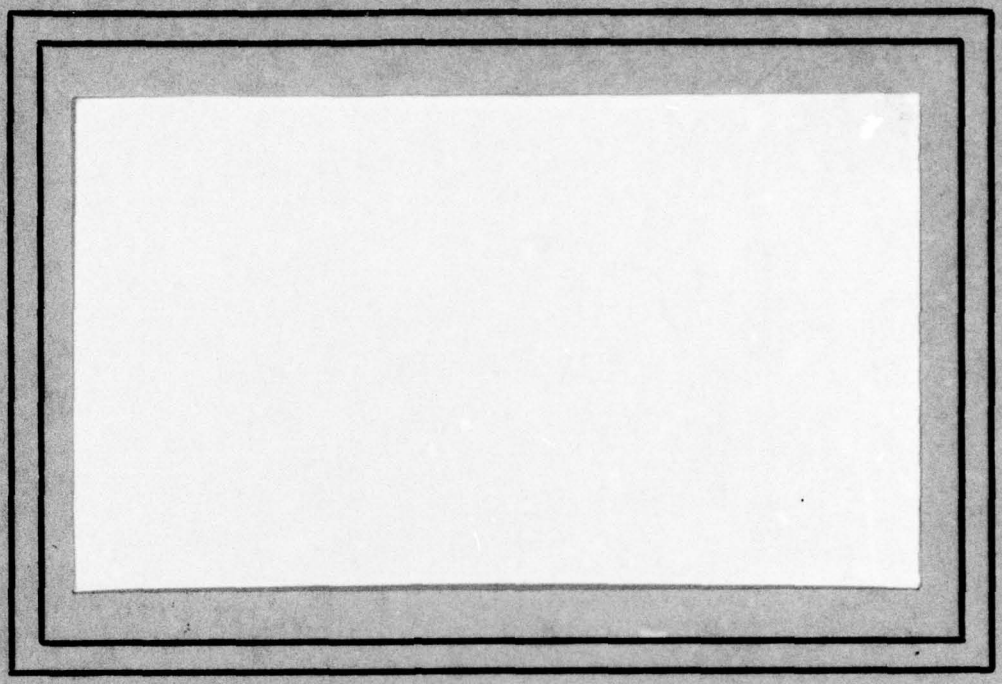
AD-A050 466 MARYLAND UNIV COLLEGE PARK COMPUTER SCIENCE CENTER F/G 9/2
NONDETERMINISTIC BOTTOM-UP PYRAMID ACCEPTORS.(U)
DEC 77 A NAKAMURA, C R DYER AFOSR-77-3271

UNCLASSIFIED TR-616 AFOSR-TR-78-0176 NL

| OF |
AD
A050466

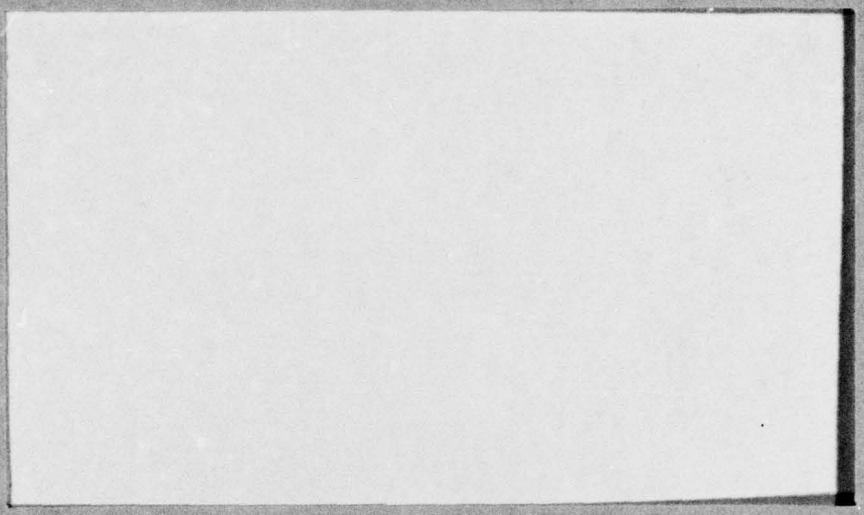
END
DATE
FILMED
3-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



COMPUTER SCIENCE TECHNICAL REPORT SERIES

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742



AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(18) 1. REPORT NUMBER: AFOSR-TR-78-0176

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): NONDETERMINISTIC BOTTOM-UP PYRAMID ACCEPTORS.

(9) 5. TYPE OF REPORT & PERIOD COVERED: Interim rept.,

(14) 6. PERFORMING ORG. REPORT NUMBER: TR-616

(10) 7. AUTHOR(s): Akira/Nakamura [redacted] Charles R./Dyer

(15) 8. CONTRACT OR GRANT NUMBER(s): ✓AFOSR 77-3271✓

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Maryland
Computer Science Center
College Park, MD 20742

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(16) 61102F (17) 2304 A2

11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Office of Scientific Research/NM
Bolling AFB, DC 20332

(11) 12. REPORT DATE: Dec [redacted] 77

(12) 13. NUMBER OF PAGES: 13 p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
formal languages
cellular automata
nondeterministic acceptors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Several properties and capabilities of nondeterministic bottom-up pyramid cellular acceptors (NBPA's) are presented. NBPA's are a special case of the pyramid cellular acceptors, proposed by Dyer and Rosenfeld. The main result is that the class of languages accepted by nondeterministic bounded cellular array acceptors is the same as that accepted by NBPA's.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

403 018

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TR-616
AFOSR-77-3271 December 1977

NONDETERMINISTIC BOTTOM-UP PYRAMID ACCEPTORS

by

Akira Nakamura[1,2]
Charles R. Dyer[2]

ABSTRACT

Several properties and capabilities of nondeterministic bottom-up pyramid cellular acceptors (NBPA's) are presented. NBPA's are a special case of the pyramid cellular acceptors proposed by Dyer and Rosenfeld. The main result is that the class of languages accepted by nondeterministic bounded cellular array acceptors is the same as that accepted by NBPA's.

---

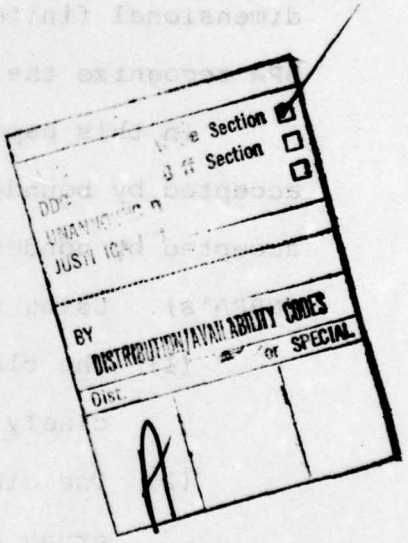
[1]Department of Applied Mathematics
Hiroshima University
Hiroshima, Japan

[2]Computer Science Center
University of Maryland
College Park, MD 20742

The support of the Directorate of Mathematical and Information Sciences, U. S. Air Force Office of Scientific Research, under Grant AFOSR-77-3271, is gratefully acknowledged, as is the help of Mrs. Shelly Rowe in preparing this paper. This paper will be presented at the Languagesand Automata Symposium in Japan in February 1978.

ACKNOWLEDGMENT

The first author would like to thank Professor Azriel Rosenfeld for arranging a very enjoyable stay at the Computer Science Center, University of Maryland, November 1977 - January 1978 and for fruitful discussions.



1. Introduction

In [1, 2], Dyer and Rosenfeld introduced cellular pyramid acceptors, which are multilayer stacks of cellular arrays. The stack consists of a bottom array of 2^r by 2^r cells in which an input pattern is given; the next lowest layer of 2^{r-1} by 2^{r-1} cells; and so on, until the top layer is a single cell. Dyer and Rosenfeld have shown that many useful recognition tasks are executed by this acceptor in time proportional to the logarithm of the diameter of the input. They also introduced a bottom-up pyramid acceptor (BPA), which is a simplified version of the pyramid acceptor (PA), and proposed a number of interesting open problems about BPA's. One of the problems is as follows: Can a BPA simulate a 2-dimensional finite-state acceptor? Another one is: Can a BPA recognize the connectedness of a set of 1's in its input?

In this paper we shall show that the class of languages accepted by bounded cellular array acceptors (CA's) is also accepted by nondeterministic bottom-up pyramid acceptors (NBPA's). Using this result we also show that:

(1) The class of languages accepted by NBPA's is precisely the class accepted by NCA's.

(2) The class of languages accepted by rectangular array bounded acceptors (RABA's) is also accepted by NBPA's. In particular, the two-dimensional finite-state languages are accepted by NBPA's.

(3) The emptiness problem for NBPA's is unsolvable.

(4) An NBPA can recognize the connectedness of the

set of 1's in its input pattern (of 1's and 0's).

(5) The class of languages accepted by nondeterministic two-dimensional multi-pass on-line tessellation acceptors (NMPOTA's) [3, 4] is also accepted by NBPA's.

The idea used in proving the main result is to consider the sequence of configurations (i.e., the states of all the cells in a cellular automaton). The base array of cells in an NBPA can nondeterministically guess at each step of the simulation the next states of their corresponding cells in the CA. The non-base cells can then verify whether or not the succession of configurations is a legal sequence for the CA.

2. Bottom-up pyramid acceptors

In this section we review some basic concepts about pyramid acceptors (PA's). A bounded cellular array acceptor (CA) is a finite, rectangular array of identical finite state machines (FSM's), or cells. Each of these cells is a quadruple $M = (Q_N, Q_T, \delta, A)$, where Q_N is a nonempty, finite set of states, $Q_T \subseteq Q_N$ is a finite set of input states, $A \subseteq Q_N$ is the set of accepting states, and $\delta: Q_N^5 \rightarrow Q_N$ is the state transition function, mapping the current state of M and its four nearest neighbors into M's next state. If the mapping is into sets of states, i.e., $\delta: Q_N^5 \rightarrow 2^{Q_N}$, then M is nondeterministic. In addition, there exists a special boundary state $\# \in Q_N$. The state transition function is restricted so that the boundary state can never be exited from or entered. Consequently, only those cells initially in a non-# state can ever be in a non-# state. A configuration of a CA and acceptability by a CA are defined in the standard way.

A pyramid cellular acceptor is a pyramidal stack of 2-dimensional CA's, where the bottom array has size 2^r by 2^r, the next lowest 2^{r-1} by 2^{r-1}, and so forth, the (r+1)st layer consisting of a single cell, called the root. Each cell is defined as an identical FSM, $M = (Q_N, Q_T, \delta, A)$. Q_N, Q_T and A are defined as before. Each cell now has nine neighbors -- four son cells in a 2-by-2 block in the level below, four brother cells in the current level, and one father cell in the level above. The nine neighbors are shown in Figure 1.

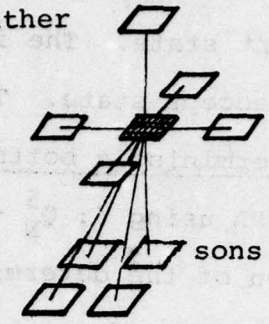


Figure 1

The transition function δ maps 10-tuples of states into states -- or sets of states, in the nondeterministic case. The input pattern is stored as the initial states of the bottom array, henceforth called the base array. The root is the accepting cell. The whole pyramid is surrounded by the boundary state # as before. A configuration of a PA is defined in the standard way and acceptability of an input pattern (configuration) by a PA is also defined in the usual way.

Now, alternative neighborhood definitions can be made which restrict information transmission through a PA. In particular, we now define a simplification in which the only neighbors of a cell are its sons, so that state information can move only one way up the pyramid. A <u>bottom-up pyramid acceptor</u> (BPA) is a PA whose state transition function is modified to be $\delta: Q_N^5 \rightarrow Q_N$. In this case, the next state of a cell depends only on the current states of that cell and its four sons. As in the case of the PA, the input defines the

start states of the base array, the other cells being initialized to a quiescent state. The input is accepted if the root ever enters an accept state. This BPA is called deterministic. A nondeterministic bottom-up pyramid acceptor (NPBA) is defined as a BPA using $\delta: Q_N^5 \rightarrow 2^{Q_N}$ instead of the state transition function of the deterministic BPA. (Generally, we use the notation N in front of the names of acceptors to specify nondeterminism. The absence of this letter implies that the machine is deterministic.)

3. NBPA's and CA's

In this section we prove that an NBPA can simulate deterministic and nondeterministic BCA's in almost real time. Using this result we also compare the language recognition capability of NBPA's with that of other automata.

Theorem 1. For an arbitrary deterministic or nondeterministic CA, there is an NBPA which simulates it in real time following a log diameter time startup delay.

Proof: Given a CA whose FSM is $M = (Q_N, Q_T, \delta, A)$, each input pattern determines a sequence of array configurations, defined by the repeated application of the state transition function δ simultaneously at every cell. Input arrays which are not square and whose side lengths are not powers of two can be padded with #'s at their right and bottom sides.

Each base cell in the NBPA nondeterministically chooses at each step a state from Q_N, while also remembering its previous state. Thus at the end of time t each cell c stores a pair of states (p,q) from Q_N, where p and q are the states chosen by c at times t and t-1, respectively. To check whether or not the new configuration legally follows from the previous one, we verify that the new state of each cell c is in the range of δ given the previous states of c and of its four nearest neighbors. This is accomplished by the non-base cells which check that each 3x3 block of state pairs is legal. That is, if

$$\begin{matrix} (s_{i_1},s_{j_1}) & (s_{i_2},s_{j_2}) & (s_{i_3},s_{j_3}) \\ (s_{i_4},s_{j_4}) & (s_{i_5},s_{j_5}) & (s_{i_6},s_{j_6}) \\ (s_{i_7},s_{j_7}) & (s_{i_8},s_{j_8}) & (s_{i_9},s_{j_9}) \end{matrix}$$

is a 3x3 block of state pairs at time t, then a legal transition occurred at cell 5 at time t if $s_{i_5} \in \delta(s_{j_5},s_{j_2},s_{j_4},s_{j_6},s_{j_8})$.

In [2] it was shown how a deterministic BPA can detect arbitrary local patterns in log diameter time steps. That algorithm is easily modified to verify that each 3x3 block of base cell states is one of a finite number of local patterns. Hence at each time t > log diameter, the root can decide whether the base's configuration at time t - log diameter was a legal successor to the base configuration at time (t - log diameter) - 1.

In addition, the root can simultaneously check at each step whether or not the upper-left corner base cell was in an accept state. Hence if the CA goes through a sequence of configurations leading to acceptance, the NBPA can nondeterministically guess the sequence, check its validity, and determine that an accepting state was entered. Note that it does not matter whether the CA is deterministic or nondeterministic.

Corollary 1.1 The class of languages accepted by NBPA's is precisely the class accepted by NCA's.

Proof: In [1] it was shown how a one-dimensional deterministic CA can simulate a one-dimensional deterministic PA. That construction is readily modified for BPA's, and for the case in which both acceptors are nondeterministic. From this result and Theorem 1 the corollary follows immediately.

Corollary 1.2 The class of languages accepted by rectangular array bounded acceptors is also accepted by NBPA's. In particular, the two-dimensional finite-state languages are accepted by NBPA's.

Proof: It has been shown [5] that the CA (NCA) languages are precisely the ABA (NABA) languages.

Corollary 1.3 An NBPA can recognize the connectedness of a set of 1's on a rectangular background of 0's.

Proof: Beyer [6] and Levialdi [7] defined a CA which recognizes this language.

Corollary 1.4 The class of languages accepted by nondeterministic two-dimensional multi-pass on-line tessellation acceptors is also accepted by NBPA's.

Proof: In [4] it was shown that L(NMPOTA) = L(NABA).

Corollary 1.5 The emptiness problem for NBPA's is unsolvable.

Proof: In [8] it was shown that the emptiness problem for two-dimensional finite-state languages is unsolvable; thus the corollary follows from Corollary 1.2.

References

1. C. R. Dyer and A. Rosenfeld, Celluar pyramids for image analysis, Technical Report 544, Computer Science Center, University of Maryland, College Park, MD., May 1977.

2. C. R. Dyer, Cellular pyramids for image analysis, 2, Technical Report 596, Computer Science Center, University of Maryland, College Park, MD., November 1977.

3. K. Inoue and A. Nakamura, Some properties of two-dimensional on-line tessellation acceptors, to appear in Information Sciences.

4. K. Inoue and A. Nakamura, Two-dimensional multi-pass on-line tessellation acceptors, Technical Report AL76-7, The Institute of Electronics and Communication Engineers of Japan, Tokyo, Japan, April 1976 (in Japanese).

5. A. R. Smith, III, Two-dimensional formal languages and pattern recognition by cellular automata, IEEE 12th Annual Symposium on Switching and Automata Theory, 1971, 144-152.

6. W. T. Beyer, Recognition of topological invariants by iterative arrays, MAC TR-66, Massachusetts Institute of Technology, Cambridge, Mass., October 1969.

7. S. Levialdi, On shrinking binary picture patterns, CACM 15, 1972, 7-10.

8. J. Mylopoulos, On the application of formal language and automata theory to pattern recognition, Pattern Recognition 4, 1972, 37-51.