ON THE ESSENTIAL EQUIVALENCE OF TWO FAMILIES OF LEARNING AUTOMATA+

M. A. L. Thathachar* and B. J. Oommen**

SCS - TR - 26
May 1983

* Department of Electrical Engineering, Indian Institute of Science, Bangalore: 560012, INDIA

** School of Computer Science, Carleton University, Ottawa, K1S 5B6, CANADA

+ Supported by the Natural Sciences and Engineering Research Council of Canada

ON THE ESSENTIAL EQUIVALENCE OF TWO FAMILIES OF LEARNING AUTOMATA

M. A. L Thathachar+ and B. J. Oommen++

ABSTRACT

Fixed Structure Stochastic Automata (FSSA) have been used to learn the best of a finite set of actions by interacting with a random environment. Two families of such automata are the Tsetlin Automata and the Krylov Automata. In this paper, it is shown that a Krylov Automaton which possesses a certain number of states and which interacts with an environment E_1 is equivalent to a Tsetlin Automaton possessing the same number of states but which interacts with an environment E_2. The relationship between the environments has also been derived. A tremendous gain in computation can thus be obtained in the study of the Krylov Automaton (which is essentially stochastic) by studying the corresponding deterministic Tsetlin automaton in the modified environment.

Apart from being of computational significance, this demonstrates a new way of studying certain families of Fixed Structure Stochastic Automata (FSSA) using deterministic automata thus simplifying both the analysis and the computation.

+ Department of Electrical Engineering, Indian Institute of Science, Bangalore : 560012, India.
++ Department of Computer Science, Carleton University, Ottawa, Ontario : K1S 5B6, Canada.

Index Terms : Equivalence of Learning Automata, Tsetlin and Krylov Automata, Variable Structure Stochastic and Deterministic Automata, Learning Machines, Cybernetics.

I. INTRODUCTION

Stochastic automata have been studied in the literature and have been used to model biological learning systems. The literature concerning such automata is extensive, but for the sake of brevity we merely cite the works of Narendra and Thathachar[1,2] and the book by Tsetlin[4].

The learning process of the automaton can be described as follows. Consider Fig. I. The environment with which the automaton interacts offers the latter a finite set of actions. The automaton is constrained to choose one of these actions. Once the action is chosen, the automaton is penalized by the environment, the penalty probability being dependent on the action chosen. A learning automaton is one which learns the action with the minimum penalty probability and which ultimately chooses this with a higher probability compared to the other actions.

In this paper we are concerned with a distinct class of automata, namely, those with a fixed structure. Tsetlin, who initiated work in this area, was probably the first to propose one such learning machine[3]. This automaton is deterministic and possesses some optimal properties in certain environments. Subsequently others, such as Krinsky and Krylov[4] proposed Fixed Structure Stochastic Automata (FSSA) which have superior properties because they possessed similar optimal properties in all environments.

In this paper we shall show that the family of Tsetlin automata and Krylov automata are essentially equivalent. In other words we demonstrate that a N-state Krylov automaton interacting with an environment E_1 behaves exactly as a Tsetlin automaton would if it interacted with an environment E_2. The relationship between the parameters of the corresponding environments is also shown.

The results that we highlight are not merely of pathological importance. Let us suppose, we are interested in studying the behaviour of an N-state Krylov automaton in the environment E_1. Since this requires probabilistic state transitions, the automaton must be studied by simulation, and at every instant a random number generator must be used to determine its next state. However, this problem can be equivalently studied by observing the characterstics of the deterministic N-state Tsetlin automaton in the environment E_2. The computational gain thereby is tremendous.

But the most important contribution of this paper is the new approach by which FSSA can be studied. If by manipulation of the transition probabilities, the stochasticity of the automaton can be associated with the environment, the FSSA can be replaced by a deterministic automaton interacting with a modified environment. Usually, the latter problem is far more easily studied.

The paper is organized as follows. We shall first introduce the terminology used in the literature. In the next section, we shall describe the Tsetlin automaton. Finally, in Section III we shall describe the Krylov automaton and demonstrate the essential equivalence between these automata.

I.1 Fundamentals

An Automaton is a quintuple $\mathbb{A} = \{ S, A, B, F(.,.), G(.) \}$, where, (1) $S = \{s_1, s_2, \ldots, s_N\}$ is its set of states.

(2) $A = \{a_1, a_2, \ldots, a_R\}$ is its set of actions.

(3) $B = \{0, 1\}$ is the set of possible inputs to the automaton. The input at time instant 'n' is b(n). b(n)=1 indicates that the automaton has been penalized.

(4) F(.,.) is a map from S X B to S. It determines the next state of the automaton at time 'n+1' if its state at time 'n' is known. It is called the transition function (or matrix) and can be either deterministic or stochastic.

(5) The output function (or matrix), G(.), determines the output or the action chosen by the automaton at any time and is a function of the state in which the automaton is. With no loss of generality, this map from S to A can be always considered deterministic[1, 2, 6].

The automaton learns the optimal action in A by interacting with an environment. The latter is a triple {A, B, C}, where :

(1) A is the set of actions $\{a_1, a_2, \ldots, a_R\}$. One of these actions is the input to the environment.

(2) $B = \{0, 1\}$ is its set of outputs. The output at time instant 'n', b(n), is 1 if it penalizes the automaton.

(3) $C = \{c_1, c_2, \ldots, c_R\}$ is the set of penalty probabilities characterizing the environment with :

$$c_j(n) = \Pr [b(n) = 1 \mid a(n) = a_j]$$

We assume that $c_j(n)$ is independent of 'n'.

The automaton-environment interaction can be described by Fig. I. Suppose the automaton is in state s(n) at time 'n'. Based on G(.) the action selected by it is G(s(n)). This serves as the input to the environment which immediately responds to the action by either a 0 or 1. Depending on the feedback it receives, $b(n)$, and $F(.,.)$, the automaton goes into a new state $s(n+1)$ at the next time instant to decide on a new action.

The probability $q_i(n)$ is the probability that the automaton is in state a_i at the nth time instant. Similarly, $p_j(n)$ is the probability that it chooses the jth action at this time.

I.2 Learning Criteria

With no apriori information, the automaton chooses the actions with equal probability. The expected penalty is thus initially M_o, where,

$$M_o = \sum_{i=1}^{R} p_i(0)\, c_i = \frac{1}{R} \sum_{i=1}^{R} c_i \quad (\text{since } p_i(0) = 1/R)$$

An automaton is said to learn expediently if, as time tends towards infinity, the expected penalty is less than M_o. We denote the expected penalty at time 'n' as $E[M(n)]$. The automaton is said to be optimal if $E[M(n)]$ equals the minimum penalty probability in the limit as time goes towards infinity.

It is ϵ optimal if $\lim_{n\to\infty} E[M(n)] < c_{min} + \epsilon$ where

$c_{min} = \min_i \{c_i\}$, for any arbitary $\epsilon > 0$ by suitable choice of some parameter of the automaton. Thus the limiting value of $E[M(n)]$ can be as close to c_{min} as desired.

For the rest of the paper we will be dealing with the two action case, i.e., $A = \{a_1, a_2\}$. The results derived for two actions can be extended for R-actions trivially.

II THE TSETLIN AUTOMATON

The 2-action Tsetlin automaton $T_{2N,2}$ has 2N states and is defined as follows :

$$T_{2N,2} = \{\{s_1, s_2, \ldots, s_{2N}\}, \{a_1, a_2\}, \{0,1\}, L(.,.), G_1(.)\}$$

If the automaton is in any of the states $\{s_1, \ldots, s_N\}$ it chooses the action a_1. Otherwise it chooses the action a_2.

The L(.,.) map is <u>deterministic</u> and is described as below :
(1) If b = 0, (the automaton chooses a_i (i=1,2) and it gets a favourable response), the L map requires that the automaton go towards the most extreme state corresponding to that action -- s_1 or s_{N+1} -- one step at a time.
(2) If b = 1 (it gets an unfavourable response), the automaton moves towards the opposite action one step at a time.
This is shown graphically by the transition map of Fig. II.

The G_1 map is given by the matrix :

$$G_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \cdot \\ \cdot \\ N \\ N+1 \\ \cdot \\ 2N \end{array} \begin{array}{c} \begin{array}{cc} 1 & \quad 2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \qquad (1)$$

For clarity we have numbered the rows and colums of G_1. The entry G_{ij} specifies the probability of the automaton choosing action a_j if it is in state s_i.

The L(.,.) map is given by the matrices L^0 and L^1, where, $L^b{}_{ij}$ is the probability of the automaton going from state s_i to state s_j when an input b is received. As before, for clarity, the rows and colums of the matrices are numbered.

$$
L^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \cdot \\ N \\ N+1 \\ N+2 \\ \cdot \\ 2N \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & \ldots & N & N+1 & N+2 & \ldots & 2N \end{array} \\
\left[\begin{array}{ccccccccc}
1 & 0 & 0 & \ldots & 0 & 0 & . & \ldots & 0 \\
1 & 0 & 0 & & 0 & 0 & & & 0 \\
0 & 1 & 0 & & 0 & 0 & & & 0 \\
 & & & & & & & & \\
 & & & & & & & & \\
0 & 0 & 0 & 1 & 0 & 0 & & & 0 \\
0 & 0 & 0 & & 0 & 1 & 0 & & 0 \\
0 & 0 & 0 & & 0 & 1 & 0 & & 0 \\
 & & & & & 0 & 1 & & 0 \\
0 & 0 & 0 & & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\end{array}
\quad (2)
$$

$$
L^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \cdot \\ N \\ N+1 \\ N+2 \\ \cdot \\ 2N \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & \ldots & N & N+1 & N+2 & \ldots & 2N \end{array} \\
\left[\begin{array}{ccccccccc}
0 & 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 1 & 0.. & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & 1.. & 0 & 0 & 0 & \ldots & 0 \\
 & & & & & & & & \\
 & & & & & & & & \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 1 \\
0 & 0 & 0 & \ldots & 0 & 0 & 1 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & 1.. & 0 \\
 & & & & & & & & \\
0 & 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0
\end{array}\right]
\end{array}
$$

If the environment has penalty probabilies c_1 and c_2, the overall markov chain, L^* that governs the state occupational probabilities is obtainted by multiplying the matrices L^0 and L^1 with the probabilities with which they are the transition maps determining the behaviour of the automaton. Thus if $d_i = 1 - c_i$, L^* is given by (3) below.

$$
L^* = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ . \\ N \\ N+1 \\ N+2 \\ . \\ 2N \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & \ldots & N & N+1 & N+2 & \ldots & 2N \end{array} \\
\left[\begin{array}{ccccccccc}
d_1 & c_1 & 0 & 0.. & 0 & 0 & 0 & 0.. & 0 \\
d_1 & 0 & c_1 & 0.. & 0 & 0 & 0 & 0.. & 0 \\
0 & d_1 & 0 & c_1.. & 0 & 0 & 0 & 0.. & 0 \\
0 & 0 & d_1 & 0c_1. & 0 & 0 & 0 & 0.. & 0 \\
 & & & & & & & & \\
0 & 0 & 0 & ..d_1 & 0 & 0 & 0 & 0.. & c_1 \\
0 & 0 & 0 & 0.. & 0 & d_2 & c_2 & \ldots & 0 \\
0 & 0 & 0 & 0.. & 0 & d_2 & 0 & c_2.. & 0 \\
 & & & & & & & & \\
0 & 0 & 0 & 0.. & c_2 & 0 & 0 & ..d_2 & 0
\end{array} \right]
\end{array}
\qquad (3)
$$

Note that by the form of $G_1(.)$, for all n,

$$p_1(n) = \sum_{i=1}^{N} q_i(n) \qquad \text{and} \qquad p_2(n) = \sum_{i=N+1}^{2N} q_i(n)$$

L^* represents an ergodic markov chain[4, 5]. The steady state value of the (2N X 1) probability vector, $\underline{q}$, is given by the solution of the matrix equation :

$$L^{*T} \underline{q}(\infty) = \underline{q}(\infty) \qquad (4)$$

Tsetlin[4] has solved (4) using difference equations and shown that the limiting action probabilities are :

$$p_1(\infty) = \frac{A_1(\lambda_1^N - 1)}{\lambda_1 - 1} \qquad p_2(\infty) = \frac{A_2(\lambda_2^N - 1)}{(\lambda_2 - 1)}$$

where $\lambda_i = \frac{d_i}{c_i}$

The constants A_1 and A_2 are solved for using the constraint that $p_1(\infty) + p_2(\infty) = 1$, and the equations which involve $q_1(\infty)$, $q_N(\infty)$, $q_{N+1}(\infty)$ and $q_{2N}(\infty)$ on the right hand side.

Since $M(n) = c_1 p_1(n) + c_2 p_2(n)$, we obtain,

$$M(\infty) = \frac{\frac{1}{c_1^{N-1}} \cdot \frac{c_1^N - d_1^N}{c_1 - d_1} + \frac{1}{c_2^{N-1}} \cdot \frac{c_2^N - d_2^N}{c_2 - d_2}}{\frac{1}{c_1^N} \frac{c_1^N - d_1^N}{c_1 - d_1} + \frac{1}{c_2^N} \frac{c_2^N - d_2^N}{c_2 - d_2}}$$

It can be shown that if c_{min} is less than 0.5, then $M(\infty)$ can be made as close to c_{min} as desired by correspondingly increasing N. Thus the Tsetlin automaton is ϵ optimal whenever $c_{min} < 0.5$.

III THE KRYLOV AUTOMATON

The 2N-state Krylov automaton for the 2-action environment is given by the quintuple :

$$Kr_{2N} = \{\{s_1, s_2, \dots s_{2N}\}, \{a_1, a_2\}, \{0, 1\}, K(.,.), G_1(.)\}$$

The output function $G_1(.)$ is identical to that of the $T_{2N,2}$, and is given by (1).

The state transitions are determined by K(.,.) and are stochastic. If b = 0, the automaton moves one state towards the extreme state corresponding to the present action it has chosen. If b = 1 the automaton moves one state either toward the extreme state corresponding to that action or towards the set of states corresponding to the other action. Krylov's automaton assigns equal probabilities to both these transitions. The transition diagram is shown in Fig. III.

Note that K^0 is identical to L^0 of the Tsetlin automaton. The matrix K^1 is given below.

Interacting with an environment possessing penalty probabilities c_1 and c_2, the matrix that represents the overall markov chain K^* is given below with $d_i = 1 - c_i$.

Studying K^* and L^* of (3) we note the following :

(1) There is a non-zero entry in K^* if and only if there is one in the corresponding position in L^*.

(2) Every c_i in L^* is replaced by $c_i/2$ in K^*. Similarly, d_i in L^* is replaced by $d_i + c_i/2$.

(3) Other than for the first and the last columns corresponding to each action, both the matrices are doubly stochastic.

$$
K^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \cdot \\ N \\ N+1 \\ N+2 \\ \cdot \\ 2N \end{array}
\begin{array}{ccccccccc}
1 & 2 & 3 & \ldots & N & N+1 & N+2 & \ldots & 2N \\
\tfrac{1}{2} & \tfrac{1}{2} & 0 & 0.. & 0 & 0 & 0 & 0.. & 0 \\
\tfrac{1}{2} & 0 & \tfrac{1}{2} & 0.. & 0 & 0 & 0 & 0.. & 0 \\
0 & \tfrac{1}{2} & 0 & \tfrac{1}{2}.. & 0 & 0 & 0 & 0.. & 0 \\
 & & & & & & & & \\
 & & & & & & & & \\
0 & 0 & 0 & ..\tfrac{1}{2} & 0 & 0 & 0 & 0.. & \tfrac{1}{2} \\
0 & 0 & 0 & \ldots & 0 & \tfrac{1}{2} & \tfrac{1}{2} & 0.. & 0 \\
0 & 0 & 0 & \ldots & 0 & \tfrac{1}{2} & 0 & \tfrac{1}{2}.. & 0 \\
 & & & & & & & & \\
0 & 0 & 0 & \ldots & \tfrac{1}{2} & 0 & 0 & 0.\tfrac{1}{2} & 0
\end{array}
$$

$$
K^* = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \cdot \\ N \\ N+1 \\ N+2 \\ \cdot \\ 2N \end{array}
\begin{array}{ccccccccc}
1 & 2 & 3 & \ldots & N & N+1 & N+2 & \ldots & 2N \\
\frac{d_1+c_1}{2} & \frac{c_1}{2} & 0 & 0.. & 0 & 0 & 0 & 0.. & 0 \\
\frac{d_1+c_1}{2} & 0 & \frac{c_1}{2} & 0.. & 0 & 0 & 0 & 0.. & 0 \\
0 & \frac{d_1+c_1}{2} & 0 & \frac{c_1}{2}.. & 0 & 0 & 0 & 0.. & 0 \\
 & & & & & & & & \\
 & & & & & & & & \\
0 & 0 & 0 & .\frac{d_1+c_1}{2} & 0 & 0 & 0 & 0.. & \frac{c_1}{2} \\
0 & 0 & 0 & 0.. & 0 & \frac{d_2+c_2}{2} & \frac{c_2}{2} & 0.. & 0 \\
0 & 0 & 0 & 0.. & 0 & \frac{d_2+c_2}{2} & 0 & \frac{c_2}{2}.. & 0 \\
 & & & & & & & & \\
0 & 0 & 0 & 0.. & \frac{c_2}{2} & 0 & 0 & .\frac{d_2+c_2}{2} & 0
\end{array}
$$

Due to the above three observations, the ergodic markov chain represented by K^* can be solved trivially, by merely substituting in the solution for (4) $c_i/2$ and $(d_i + c_i/2)$ instead of c_i and d_i respectively. This leads us to the interesting conclusion that the Krylov automaton interacting with an environment with penalty probabilities (c_1, c_2) behaves exactly as a Tsetlin automaton would if it interacted with an environment with penalty probabilities $(c_1/2, c_2/2)$. Since c_1 and c_2 are probabilities the ϵ optimality of the Krylov automaton in <u>all</u> environments follows from the ϵ optimality properties of the Tsetlin automaton.

IV CONCLUSIONS

In this paper we have considered two families of learning automata due to Tsetlin and Krylov. We have shown that the N-state Krylov automaton interacting with an environment with penalty probabilities c_1 and c_2 behaves exactly as a N-state Tsetlin automaton would if it interacted with an environment with penalty probabilities $c_1/2$ and $c_2/2$ respectively.

In the Krylov automaton, the transitions made on receiving a penalty were to the adjacent states, and these transitions were made with equal probability. However, the form of K^* indicates that we need not necessarily make these transitions equiprobable. The probability of moving to the set of states corresponding to the alternate action can be arbitary. If this probability is zero, we will not even have an expedient learning automaton. In the case when this probability is unity, we have the Tsetlin automaton which is ϵ optimal whenever $c_{min} < 0.5$. As this probability is varied from 0.5 to unity, we encounter a variety of stochastic automata, each of them guarenteeing ϵ optimality for restricted values of c_{min}.

Krinsky[4] has also defined a deterministic automaton that guarentees ϵ optimality in all environments. Just as the Krylov automaton is shown to behave just like a Tselin automaton in a transformed environment, it is possible that a class of stochastic automata can be designed behaving similar to Krinsky's.

We have also shown that often the study of a family of stochastic automata can be greatly simplified. This can be acheived by manipulating the probabilities so that the stochasticity of the automaton can be associated with the environment. An equivalent determinsitc automaton can be then used in conjunction with a modified environment to study the original stochastic automaton.

REFERENCES

1. Narendra, K.S., and Thathachar, M. A. L., Forthcoming Book on Learning Automata.

2. Narendra, K.S., and Thathachar, M. A. L., "Learning Automata -- A Survey", IEEE Trans. Syst.Man and Cybern.,Vol. SMC-4, 1974, pp.323-334.

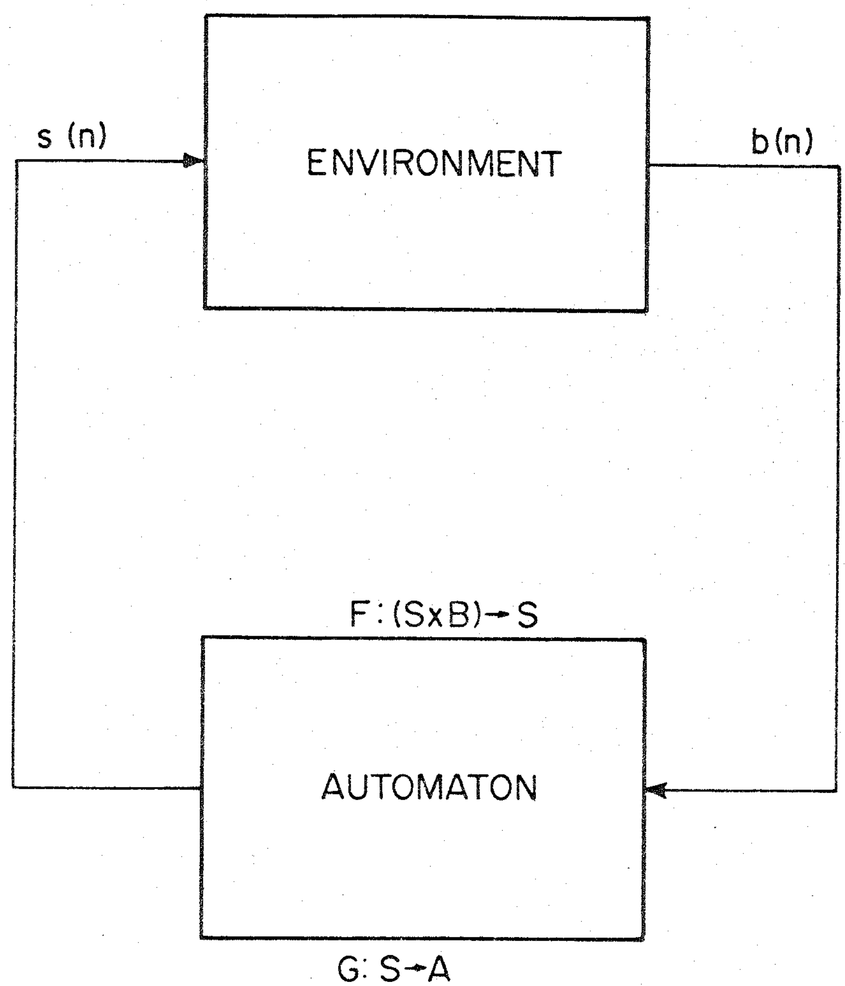
3. Tsetlin, M.L., "On the Behaviour of Finite Automata in Random Media ", Automat. Telemekh., Vol.22, 1961, pp.1345-1354.

4. Tsetlin, M.L., "Automaton Theory and the Modelling of Biological Systems", New York and London, Academic, 1973.

5. Isaacson, D.L., and Madson, R.W., "Markov Chains : Theory and Applications", Wiley, 1976.

6. Paz, A., "Introduction to Probabilistic Automata", New York, Academic, 1971.

LIST OF FIGURES

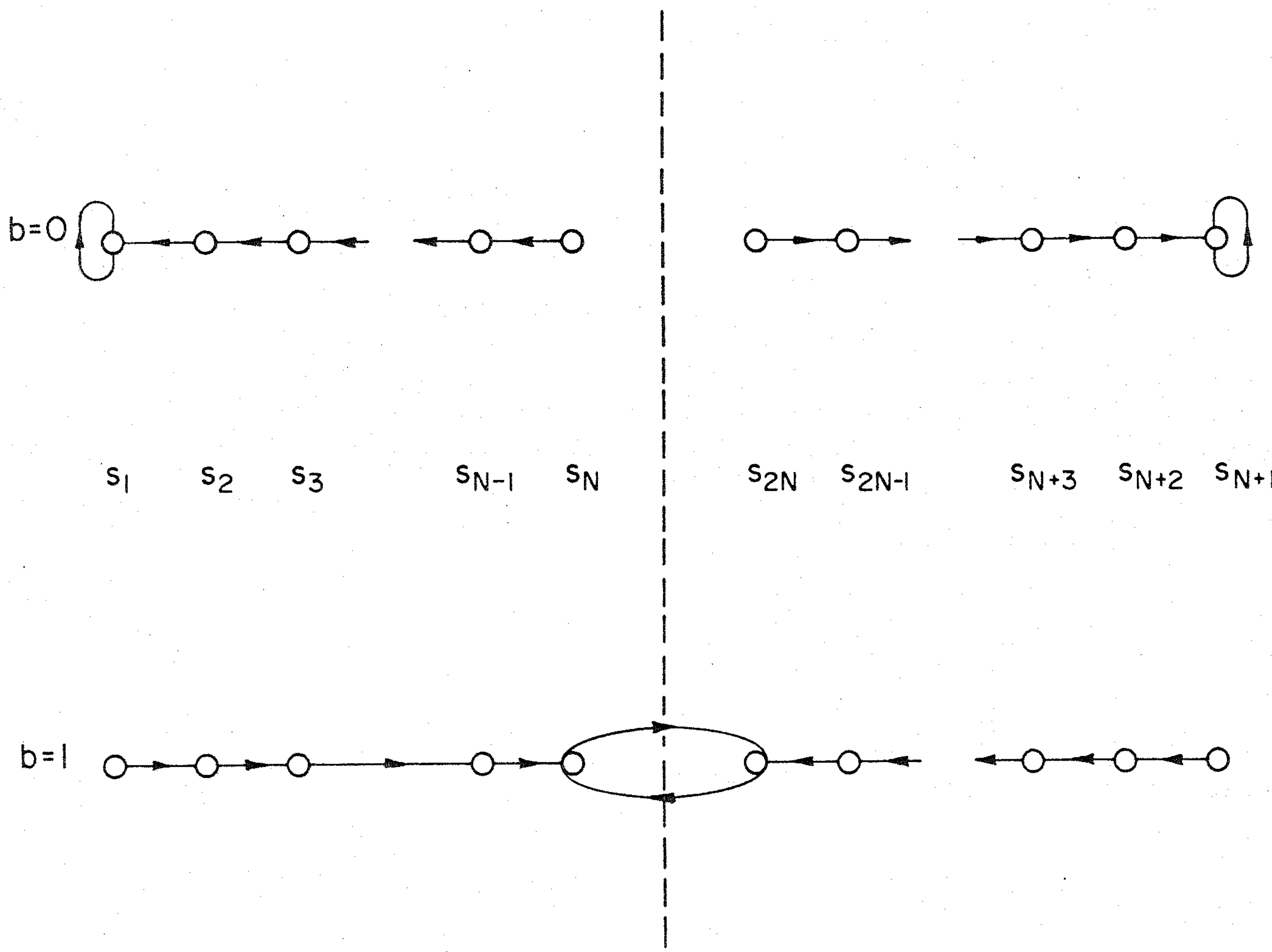


$b(n) \in \{0,1\} = B$

$s(n) \in \{s_1, s_2, \ldots s_N\} = S$

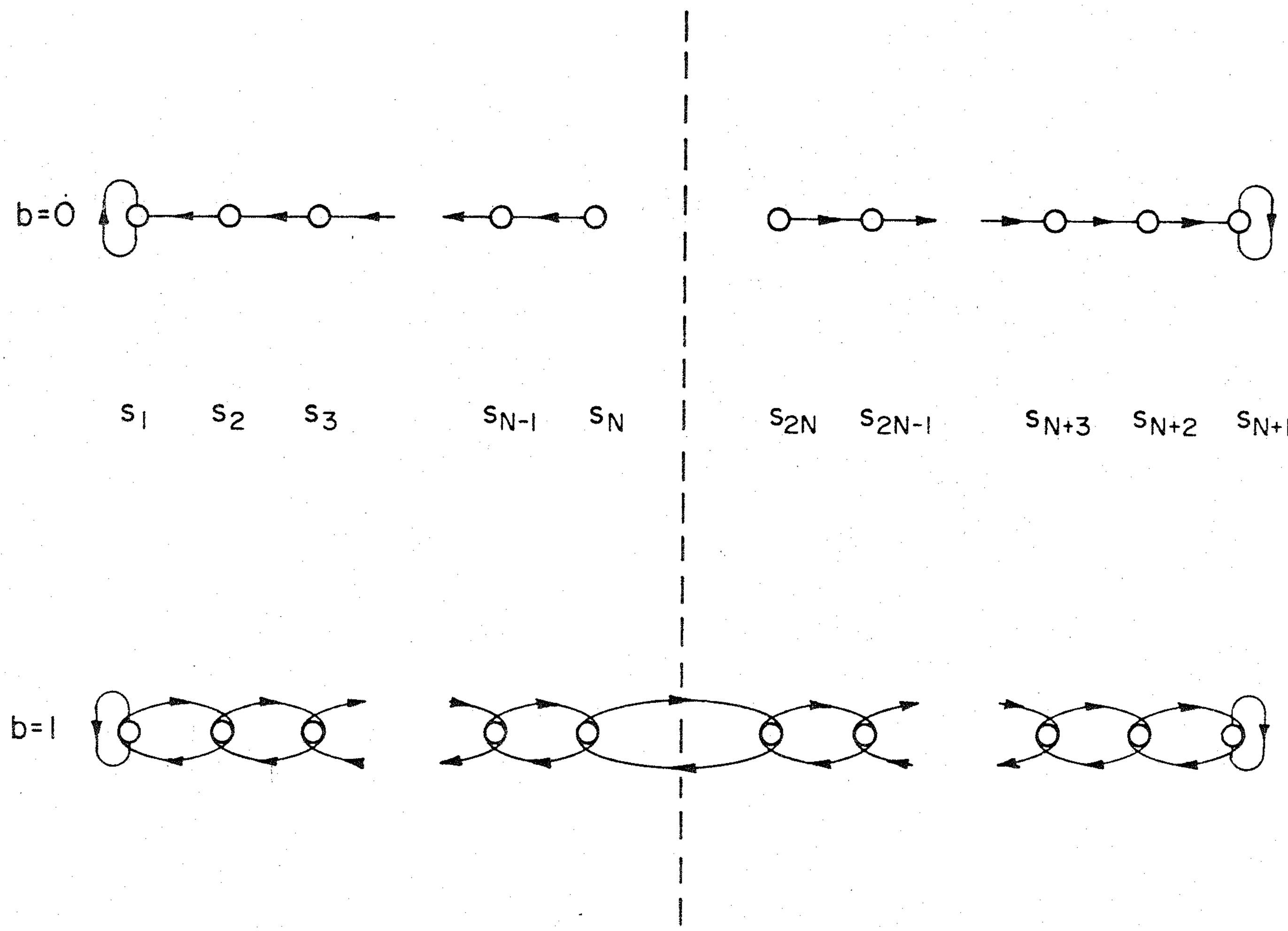
$a(n) \in \{a_1, a_2, \ldots a_R\} = A$

FIG. 1: THE AUTOMATON-ENVIRONMENT INTERACTION



(All Transitions w.p. 1)

FIG. 11: THE TSETLIN AUTOMATON



(All transitions for b = 1 w. p. 1/2)

FIG. III: THE KRYLOV AUTOMATON

CARLETON UNIVERSITY
School of Computer Science

Bibliography of SCS Reports

SCS-TR-1 THE DESIGN OF CP-6 PASCAL
Jim des Rivieres and Wilf R. LaLonde, June 1982.

SCS-TR-2 SINGLE PRODUCTION ELIMINATION IN LR(1) PARSERS:A SYNTHESIS
Wilf R. LaLonde, June 1982.

SCS-TR-3 A FLEXIBLE COMPILER STRUCTURE THAT ALLOWS DYNAMIC PHASE ORDERING
Wilf R. LaLonde and Jim des Rivieres, June 1982.

SCS-TR-4 A PRACTICAL LONGEST COMMON SUBSEQUENCE ALGORITHM FOR TEXT COLLATION
Jim des Rivieres, June 1982.

SCS-TR-5 A SCHOOL BUS ROUTING AND SCHEDULING PROBLEM
Wolfgang Lindenberg, Frantisek Fiala, July 1982.

SCS-TR-6 ROUTING WITHOUT ROUTING TABLES
Nicola Santoro, Ramez Khatib, July 1982.

SCS-TR-7 CONCURRENCY CONTROL IN LARGE COMPUTER NETWORKS
Nicola Santoro, Hasan Hural, July 1982.

SCS-TR-8 ORDER STATISTICS ON DISTRIBUTED SETS
Nicola Santoro, Jeffrey B. Sidney, July 1982.

SCS-TR-9 OLIGARCHICAL CONTROL OF DISTRIBUTED PROCESSING SYSTEMS
Moshe Krieger, Nicola Santoro, August 1982.

SCS-TR-10 COMMUNICATION BANDS FOR SELECTION IN A DISTRIBUTED SET
Nicola Santoro, Jeffrey B. Sidney, September 1982.

SCS-TR-11 A SIMPLE TECHNIQUE FOR CONVERTING FROM A PASCAL SHOP TO C SHOP
Wilf R. LaLonde, John R. Pugh, November 1982.

SCS-TR-12 EFFICIENT ABSTRACT IMPLEMENTATIONS FOR RELATIONAL DATA STRUCTURES
Nicola Santoro, December 1982.

SCS-TR-13 ON THE MESSAGE COMPLEXITY OF DISTRIBUTED PROBLEMS
Nicola Santoro, December 1982.

SCS-TR-14 A COMMON BASIS FOR SIMILARITY MEASURES INVOLVING TWO STRINGS
R.L. Kashyap and B.J. Oommen, January 1983.

SCS-TR-15 SIMILARITY MEASURES FOR SETS OF STRINGS
R.L. Kashyap and B.J. Oommen, January 1983.

SCS-TR-16 THE NOISY SUBSTRING MATCHING PROBLEM
R.L. Kashyap and B.J. Oommen, January 1983.

SCS-TR-17 DISTRIBUTED ELECTION IN A CIRCLE WITHOUT A GLOBAL SENSE OF ORIENTATION
E. Korach, D. Rotem, N. Santoro, January 1983.

.../2

SCS-TR-18 A GEOMETRICAL APPROACH TO POLYGONAL DISSIMILARITY AND THE CLASSIFICATION OF CLOSED BOUNDARIES
R.L. Kashyap and B.J. Oommen, January 1983.

SCS-TR-19 SCALE PRESERVING SMOOTHING OF POLYGONS
R.L. Kashyap and B.J. Oommen, January 1983.

SCS-TR-20 NOT-QUITE-LINEAR RANDOM ACCESS MEMORIES
Jim des Rivieres, Wilf LaLonde and Mike Dixon, August 1982, Revised March 1, 1983.

SCS-TR-21 SHOUT ECHO SELECTION IN DISTRIBUTED FILES
D. Rotem, N. Santoro, J. B. Sidney, March 1983.

SCS-TR-22 DISTRIBUTED RANKING
E. Korach, D. Rotem, N. Santoro, March 1983

SCS-TR-23 A REDUCTION TECHNIQUE FOR SELECTION IN DISTRIBUTED FILES : I
N. Santoro, J. B. Sidney, April 1983.

SCS-TR-24 LEARNING AUTOMATA POSSESSING ERGODICITY OF THE MEAN : THE TWO ACTION CASE
M.A.L. Thathachar and B. J. Oommen, May 1983

SCS-TR-25 ACTORS - THE STAGE IS SET
John R. Pugh, June 1983

SCS-TR-26 ON THE ESSENTIAL EQUIVALENCE OF TWO FAMILIES OF LEARNING AUTOMATA
M. A. L. Thathachar and B. J. Oommen, May 1983