Proceedings of Grammar Systems Week 2004, edited by E. Csuhaj-Varjú and Gy. Vaszil,
MTA SZTAKI, Budapest, pages 287 - 299, 2004.

Distributed Probabilistic Finite Automata

G. N. Santhana Krishnan, Kamala Krithivasan, Ashish Choudhary

Dept of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai, India 600036.
kamala@iitm.ernet.in

Abstract

In this paper we consider multiple choice probabilistic automaton and show that it can be simulated by a (single choice) probabilistic automaton. We define distributed probabilistic automata with four cooperating modes and show that distribution does not give any additional power.

1 Introduction

Recent developments in the field of Computer Science are towards processing information that are distributed among geologically different locations. In various fields like Artificial Intelligence, Cognitive Psychology etc, we have to deal with more and more complex tasks distributed among set of 'processors', which are working together in a well defined way. Parallel computers, computer nets, distributed databases and knowledge sources are practical materialization of this idea. Formal Language Theory constructs, namely Grammar Systems, modelling distribution and parallelism were coined in the early nineties. [1] deals with many types of Grammar Systems. Distributed Automata have been considered in [2].

Very often, in the world of distributed information processing systems, there arise situations where the system is random and its course of operation nondeterministic. Probabilistic grammars were introduced as mathematical generative models for capturing the randomness in a classical computing environment [3, 4]. Probabilistic grammar systems were studied in [5, 6] with application to network-load modelling [7].

In this paper we study distributed probabilistic automata. We first study multiple choice probabilistic automaton where we include the idea of both randomness and non-determinism. We show that a multiple choice probabilistic automaton can be simulated by a single choice probabilistic automaton. We define distributed probabilistic automata and consider four standard modes of cooperation between the components, viz, $*, = k, \leq k, \geq k$ modes. We show that a distributed probabilistic automaton can be simulated by a multiple choice probabilistic automaton. Since multiple choice probabilistic automaton can be simulated by a single choice probabilistic automaton, we see that distribution does not increase the power of probabilistic automaton in any of the four modes of cooperation.

In the next section we define probabilistic automata. In section 3 we define multiple choice probabilistic automata and show the equivalence to single choice probabilistic automata. In section 4 we define distributed probabilistic automaton with four modes of cooperation and show that in any mode, it can be simulated by a multiple choice probabilistic automaton. The paper concludes with some remarks in section 5.

2 Probabilistic Automaton

In this section, we consider the definition of probabilistic automaton. For the definition and details of probabilistic grammars, the reader is referred to [3, 4]. A study of probabilistic automata is done in [8].

Definition 2.1 A finite probabilistic automaton over a finite alphabet V is an ordered triple $\mathbf{PA} = (S, s_0, M)$ where $S = \{s_1, s_2, s_3, ..., s_n\}$ is a finite set with $n \geq 1$ elements(the set of internal states), s_0 is an n-dimensional stochastic row vector(the initial distribution) and M is a mapping of V into the set of n-dimensional stochastic matrices. For $x \in V$, the $(i,j)^{th}$ entry in the matrix $M(x)$ is denoted by $p_j(s_i, x)$ and referred to as the transient probability of **PA** to enter into the state s_j, after being in the state s_i after consuming the input x.

As an example, consider the following example. $\mathbf{PA}_1 = (\{s_1, s_2\}, (1,0), M)$ over the alphabet $\{x, y\}$ where

$$M(x) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad M(y) = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

The initial distribution indicates that s_1 is the initial state. From the matrices M we see that s_1 changes to s_2 with a probability of $1/2$ on reading the symbol y. This can be indicated in a diagram with the states being the nodes and arcs having labels in the form $x(p)$ where x is the symbol while p is the probability of transition from one node to the other on scanning the symbol x.

For a finite probabilistic automaton, we increase the domain of M from V to V^* as follows:

1. $M(\epsilon) = I$
2. $M(wx) = M(w)M(x)$

Now for a word w, the $(i,j)^{th}$ entry of $M(w)$ would denote the probability that the automaton would move to state s_j after getting the input w if it were initially in state s_i.

Definition 2.2 Let $\mathbf{PA} = (\{s_1, \ldots, s_n\}, s_0, M)$ be a finite probabilistic automaton over V and $w \in V^*$. The stochastic row vector $s_0M(w)$ is termed as the distribution of states caused by the word w and is denoted by $\mathbf{PA}(w)$.

We notice that for a word w, the i^{th} entry of $\mathbf{PA}(w)$ is the probability that the automaton is in state s_i starting from the initial distribution s_0.

Definition 2.3 Let $\overline{s}_1$ be a $n-$dimensional column vector, each component of which equals either 0 or 1, and the **PA** as in the previous definition. Let η be a real number such that $0 \leq \eta \leq 1$. The language accepted in **PA** by $\overline{s}_1$ with the cut-point η is defined by

$$\mathrm{L}(\mathbf{PA}, \overline{s_1}, \eta) = \{w | s_0 M(w) \overline{s_1} \geq \eta\}.$$

A language L is $\eta-$stochastic if for some **PA**, η and $\overline{s_1}$, $L = L(\mathbf{PA}, \overline{s_1}, \eta)$. A language L is stochastic if it is $\eta-$stochastic for some η.

Theorem 2.1 *Every regular language is stochastic. Further, every regular language is $\eta-$stochastic for every $0 \leq \eta \leq 1$.*

Theorem 2.2 *Every 0-stochastic language is regular.*

3 Multiple-Choice Probabilistic Automaton

Now we consider multiple choice probabilistic automaton.

Definition 3.1 (Multiple-Choice Probabilistic Automaton) A finite Multiple-Choice probabilistic Automaton over a finite alphabet V is an ordered triple **NDPA** $= (S, s_0, M)$ where $S = \{s_1, s_2, \ldots, s_n\}$ is a finite set with $n \geq 1$ elements (the set of internal states), s_0 is an $n-$dimensional stochastic row vector(the initial distribution) and M is a collection of mapping of V into the set of $n-$dimensional stochastic matrices. For $x \in V$, the $(i,j)^{th}$ entry in each of the matrices in $M(x)$ is denoted by $p_j(s_i, x)$ and referred to as the transient probability of **NDPA** to enter into the state s_j after being in the state s_i and consuming the input x.

We define the language accepted in a similar fashion as above, with the consideration that whenever a matrix needs to be chosen for an alphabet symbol a, the choice is made non-deterministically among the matrices in $M(a)$. We talk about two different modes of acceptance, namely the $max-$mode and the $p-$mode.

Definition 3.2 $L_{max}(\mathbf{NPA}, \overline{s_1}, \eta) = \{w | max\{s_0 M(w)\overline{s_1}\} \geq \eta\}$, where $\overline{s_1}$, η and s_0 are as defined above and $\{s_0 M(w)\overline{s_1}\}$ denotes the set of all derivations of the string w.

Definition 3.3 $L_p(\mathbf{NPA}, \overline{s_1}, \eta) = \{w | prob\{s_0 M(w)\overline{s_1}\} \geq \eta\}$, where $\overline{s_1}$, η and s_0 are as defined above and $prob\{s_0 M(w)\overline{s_1}\}$ denotes the probability of the derivation of the string w considering all possible derivations of the string w.

Let $L_{max}(\mathbf{NPA})$ denote the family of languages generated by multiple choice probabilistic automata in the $max-$ mode, $L_p(\mathbf{NPA})$ denote the family of languages generated in the $p-$mode and $L(\mathbf{PA})$ denote the family of languages generated by probabilistic automata, i.e. let it denote the family of stochastic languages.

Theorem 3.1 $L_p(\mathbf{NPA}) = L(\mathbf{PA})$.

Proof. Let $M_1(x), M_2(x), \ldots, M_k(x)$ be the various choices for the alphabet x. Let the system be in distribution $\overline{s_1}$, and the input which is read next is x. The probability that the new state is s_j is :

$$p_j = \sum_{l=1}^{k} p(l) * (\sum_{i=0}^{n} \overline{s_{1i}} * p_{lj}(s_i, x))$$

Where

- p_j is the probability that the automaton is in state s_j after reading the alphabet x.
- $\overline{s_{1i}}$ is the i^{th} entry in $\overline{s_1}$. In other word, it stands for the probability that the system were in the state s_i initially.
- $p_{lj}(s_i, x))$ is the transient probability in $M_l(x)$ that the automaton would change from state s_i to s_j on encountering the symbol x.
- $p(l)$ is the probability of choice of the matrix $M_l(x)$.

Noting that the matrices are chosen non-deterministically, the above equation reduces to

$$p_j = \sum_{l=1}^{k} (1/k) * (\sum_{i=0}^{n} \overline{s_{1i}} * p_{lj}(s_i, x))$$

Now the above can be written in the vector notation as

$$\overline{s_2} = \sum_{l=1}^{k} (1/k) * \overline{s_1} * M_l(x)$$

or in other words,

$$\overline{s_2} = \overline{s_1} * \sum_{l=1}^{k} (1/k) * M_l(x)$$

If we consider the **PA** with (S, s_0, M), with the set of all matrices for each alphabet replaced with their arithmetic mean, we obtain the same distribution of states at each stage. So, this clearly proves that Multiple-Choice probabilistic automata are at least as powerful as probabilistic automata. Also, since every probabilistic automaton is also a multiple-choice probabilistic automaton, the inclusion holds the other way too. Hence the theorem. □

4 Distributed Probabilistic Automaton

We now define the various kinds of distributed probabilistic automaton with certain restrictions as to transfer of control, and we study various families of languages accepted.

Definition 4.1 (**Distributed Probabilistic Automaton**) A distributed finite probabilistic automaton over a finite alphabet V is an ordered $(k+2)$-tuple **DPA** $= (S, s_0, M_1, M_2, \ldots, M_k)$, where $S = \{s_1, s_2, \ldots, s_n\}$ is a finite set with $n \geq 1$ elements(the set of internal states), s_0 is an $n-$dimensional stochastic row vector(the initial distribution) and each of the M_i's is mapping of V into the set of $n-$dimensional stochastic matrices. For $x \in V$, the $(j,k)^{th}$ entry of each of the matrices in $M_i(x)$ is denoted by $p_{ik}(s_j, x)$ and referred to as the transient probability of **DPA** to enter into the state s_k, after being in the state s_j after consuming the input x in component i.

We now define the various modes of cooperation. To achieve that end, we first define the various modes of transition.

Definition 4.2 (**Transition Modes**) There are four possible modes of transitions.

- $*$-mode: The transition from M_i to some other component M_j occurs at any arbitrary stage.
- =k-mode: The transition from M_i to some other component M_j occurs after exactly k transitions in M_i.
- $\geq k$-mode: The transition from M_i to some other component M_j occurs after at least k transitions in M_i.
- $\leq k$-mode: The transition from M_i to some other component M_j occurs before k+1 transitions in M_i.

We note that conventional $t-$mode does not make sense in this context since all the matrices involved are stochastic, and hence the derivation in any M_i cannot terminate unless the string has been completely processed. Also the derivation always starts with M_1 (without loss of generality, this can be assumed).

Definition 4.3 $L_{(p,\alpha)}(DPA, \eta) = \{w \in V^* | prob\{s_0 M(w)\overline{s_1}\} \geq \eta\}$, where α is the transition mode, $\alpha \in \{*, = k, \leq k, \geq k | k \geq 1\}$ and $prob\{s_0 M(w)\overline{s_1}\}$ denotes the probability of derivation of the string w considering all possible derivations of the string w.

Definition 4.4 $L_{(max,\alpha)}(DPA, \eta) = \{w \in V^* | max\{s_0 M(w)\overline{s_1}\} \geq \eta\}$, where α is the transition mode, $\alpha \in \{*, = k, \leq k, \geq k | k \geq 1\}$ and $\max\{s_0 M(w)\overline{s_1}\}$ refers to the maximum probability of the derivation of the string w amongst all the possible derivations of it.

We study the acceptance power of **DPA** in the various modes of transition. We prove that $L_{(p,\alpha)}$ generates just a stochastic language in the following theorem. We leave the study of $L_{(max,\alpha)}$ as an open problem.

Theorem 4.1 *Given $\alpha \in \{*, = k, \leq k, \geq k\}$ and given a* **DPA M**, *there exists a* **PA M**$''$ *such that $L_{(p,\alpha)}(s_0, M, \eta) = L(s_0, M'', \eta)$.*

Proof. **Case 1: $\alpha = *$.**

Let $\mathbf{M} = (S, s_0, M_1, M_2, \ldots, M_l)$. Let the cardinality of the set S is n. We try to define a corresponding **PA** $\mathbf{M}'' = (S'', s_0'', M'')$ in the following way.

$S'' = \{[q, i] | 1 \leq i \leq l, q \in S\}$. Without loss of generality, we assume that the machine starts working in the first component. s_0'' is a nl stochastic row vector, which defines the new initial state distribution in $\mathbf{M}''$. Since we have assumed that the machine $\mathbf{M}$ starts working in the first component, so if the initial state distribution in $\mathbf{M}$ is $(p_{q_1}, p_{q_2}, \ldots, p_{q_n})$, then the initial state distribution in machine $\mathbf{M}''$ would be represented as:

$$s_0'' = (p_{[q_1,1]}, p_{[q_2,1]}, \ldots, p_{[q_n,1]}, 0, 0, \ldots, 0(nl - n)times)$$

At any instance of time, the machine $\mathbf{M}$ can be in any one of the component. Suppose at current instance t, the machine $\mathbf{M}$ is in component i, then the state distribution in $\mathbf{M}$ would have been represented as:

$$s_t = (p_{q_1}, p_{q_2}, \ldots, p_{q_n})$$

The corresponding state configuration in machine $\mathbf{M}''$ would be represented by a nl stochastic row vector as:

$$s_t'' = (0, 0, \ldots, p_{[q_1,i]}, p_{[q_2,i]}, \ldots, p_{[q_n,i]}, 0, 0, \ldots, 0)$$

where the non-zero entries are from the position $n(i-1)+1$ to $n(i-1)+n$ and $p_{[q_j,i]} = p_{q_j}, \forall 1 \leq j \leq n$. Also the final state configuration has to be changed in $\mathbf{M}''$. If the final state configuration in $\mathbf{M}$ is $(p_{q_1}, p_{q_2}, \ldots, p_{q_n})^T$ where each of the values $p_{q_i}, \forall i, 1 \leq i \leq n$ is either 0 or 1, then the corresponding final state configuration in $\mathbf{M}''$ would be represented by a nl-dimensional column vector:

$$(p_{[q_1,1]}, p_{[q_2,1]}, \ldots, p_{[q_n,1]}, p_{[q_1,2]}, p_{[q_2,2]}, \ldots, p_{[q_n,2]}, \ldots, p_{[q_1,l]}, p_{[q_2,l]}, \ldots, p_{[q_n,l]})^T$$

where $p_{[q_i,j]} = p_{q_i}, \ \forall i, j, 1 \leq i \leq n, 1 \leq j \leq l$. From each of the M_i, we have l possible matrices to choose and from each of the matrices, we choose the matrix for given alphabet. We can encompass both these steps into a single step. The change of M_i to M_j can be done by having the present stochastic vector, representing the current state distribution, multiplied by the following matrix:

$$M(i,j) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where M is a (nl)x(nl) matrix and I is a nxn identity matrix which starts off at the entry $(i-1)n+1$, $(j-1)n+1$. It can be easily seen that the matrix carries the transition from M_i to M_j. For example, consider $M(1,2)$. This is a matrix which carries out the transformation from M_1 to M_2. Considering that $n = 2, l = 3$ we have:

$$M(1,2) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Suppose the state were $\overline{s} = (p_1, p_2, 0, 0, 0, 0)$.

Then $\overline{s_1} = \overline{s} * M(1,2)$, which is

$$(p_1, p_2, 0, 0, 0, 0) * \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which is $(0, 0, p_1, p_2, 0, 0)$.

We thus see that $M(i,j)$ transfers control from M_i to M_j.

For each alphabet a, we define the matrix as follows:

$$M''(a) = \begin{bmatrix} M_1(a) & 0 & . & 0 \\ 0 & M_2(a) & . & 0 \\ . & . & . & . \\ . & 0 & . & M_l(a) \end{bmatrix}$$

Consider the same example as above. Suppose, we have $l = 3$ and $n = 2$. Suppose, the present vector is $(p_1, p_2, 0, 0, 0, 0)$, and we read the symbol a. The matrix for a would be

$$M''(a) = \begin{bmatrix} p_{11}(s_1,a) & p_{12}(s_1,a) & 0 & 0 & 0 & 0 \\ p_{11}(s_2,a) & p_{12}(s_2,a) & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{21}(s_1,a) & p_{22}(s_1,a) & 0 & 0 \\ 0 & 0 & p_{21}(s_2,a) & p_{22}(s_2,a) & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{31}(s_1,a) & p_{32}(s_1,a) \\ 0 & 0 & 0 & 0 & p_{31}(s_2,a) & p_{32}(s_2,a) \end{bmatrix}$$

Now,consider $\overline{s_2}$.

$\overline{s_2} = \overline{s} * M''(a)$, which is

$$(p_1,p_2,0,0,0,0) * \begin{bmatrix} p_{11}(s_1,a) & p_{12}(s_1,a) & 0 & 0 & 0 & 0 \\ p_{11}(s_2,a) & p_{12}(s_2,a) & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{21}(s_1,a) & p_{22}(s_1,a) & 0 & 0 \\ 0 & 0 & p_{21}(s_2,a) & p_{22}(s_2,a) & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{31}(s_1,a) & p_{32}(s_1,a) \\ 0 & 0 & 0 & 0 & p_{31}(s_2,a) & p_{32}(s_2,a) \end{bmatrix}$$

which is $(p_1 * p_{11}(s_1,a) + p_2 * p_{11}(s_2,a), p_1 * p12(s_1,a) + p_2 * p_{12}(s_2,a), 0, 0, 0, 0)$.

We can view it in other way. The above can be seen as follows in the block matrix-multiplication format.

$$\begin{bmatrix} s_{n,1} * M_1(a) & s_{n,2} * 0 & s_{n,3} * 0 \\ s_{n,1} * 0 & s_{n,2} * M_2(a) & s_{n,3} * 0 \\ s_{n,1} * 0 & s_{n,2} * 0 & s_{n,3} * M_3(a) \end{bmatrix}$$

where $s_{n,1}, s_{n,2}$ and $s_{n,3}$ represent the first, second and third n elements in the current state configuration. In the context of the above example, $s_{n,1}, s_{n,2}$ and $s_{n,3}$

represents the first, second and last two elements in the current state configuration. So, we now see that $M''(a)$ helps us process the string as if it were under the control of one M_i.

In order to incorporate the transition from M_i to M_j, we premultiply $M''(a)$ by $M(i,j)$.

Consider the above example. Instead of having just $M''(a)$, we will have $M(1,2)*M''(a)$, $M(1,3)*M''(a)$, $M(2,1)*M''(a)$, $M(2,3)*M''(a)$, $M(3,1)*M''(a)$, $M(3,2)*M''(a)$ and of course $M''(a)$ itself. For each alphabet, thus we have $l*(l-1)+1$ matrices to choose from. We are now left with the $p-$mode of acceptance in the multiple-choice probabilistic automaton, which we have already proved to be equivalent to probabilistic automaton.

Case 2: $\alpha = =\mathbf{k}$.

Suppose we are given a distributed probabilistic automaton $\mathbf{M}$ with l components and n states

$$\mathbf{M} = (S, s_0, M_1, M_2, M_3, \ldots, M_l)$$

Without loss of generality, we assume that initially the automaton is in the first component. Since the mode of cooperation is $= k$, the automaton should spend exactly k steps in the current component before switching to some other component, other than the present component.

To simulate the working of this distributed automaton with a single probabilistic automaton, we look at $\mathbf{M}$ in a different way. We think that we have exactly k copies of each of the components $M_1, M_2, M_3, \ldots, M_l$ i.e

$$M_{11}, M_{12}, M_{13}, \ldots, M_{1k}$$

$$M_{21}, M_{22}, M_{23}, \ldots, M_{2k}$$

$$\ldots$$

$$M_{l1}, M_{l2}, M_{l3}, \ldots, M_{lk}$$

Here $M_{ij} = M_i$, $\forall i, j$, $1 \leq i \leq l, 1 \leq j \leq k$.

All the transition matrices in $M_{11}, M_{12}, M_{13}, \ldots, M_{1k}$ are identical as in M_1. Similarly all the transition matrices in $M_{21}, M_{22}, M_{23}, \ldots, M_{2k}$ are identical as in M_2 and so on. Now the original distributed probabilistic automaton $\mathbf{M}$ is converted into another distributed probabilistic automaton $\mathbf{M}'$ as follows:

$$\mathbf{M}' = (S', s_0', M_{11}, M_{12}, \ldots, M_{1k}, M_{21}, M_{22}, \ldots, M_{2k}, \ldots, M_{l1}, M_{l2}, \ldots, M_{lk})$$

where

- S' is the new set of states in $\mathbf{M}'$ such that $S' = \{[q, ij] | 1 \leq i \leq l, 1 \leq j \leq k, q \in S\}$. In $\mathbf{M}'$ since the components are replicated, we also include the number of the component and the number of the copy of the component to which the state belongs along with each state in the new set of states. The cardinality of the set S' is nlk.

- s'_0 is a $n * l * k$ stochastic row vector which denotes the new initial probability distribution of $\mathbf{M}'$. If $s_0 = (p_{q_1}, p_{q_2}, \ldots, p_{q_n})$ then we have :

$$s'_0 = (p_{[q_1,11]}, p_{[q_2,11]}, \ldots, p_{[q_n,11]}, 0, 0, \ldots, 0(n * l * k - n)times)$$

  such that $p_{q_i} = p_{[q_i,11]}\ \forall i, 1 \leq i \leq n$. Here p_{q_j} is interpreted as the probability of being in state q_j in $\mathbf{M}$ and $p_{[q_j,11]}$ is interpreted as the probability of being in the state $[q_j, 11]$ in $\mathbf{M}'$.

- $M_{11}, M_{12}, \ldots, M_{1k}, M_{21}, M_{22}, \ldots, M_{2k}, \ldots, M_{l1}, M_{l2}, \ldots, M_{lk}$ are the set of transition matrices such that :

$$M_{i1} = M_{i2} = M_{i3} = \ldots, = M_{ik} = M_i, \forall i, 1 \leq i \leq l.$$

The probabilistic state distribution at any instance of time is given by a $n * l * k$ stochastic row vector such that if at current instance the M_{ij}^{th} component is active then the probability distribution is:

$$(0, 0, \ldots, 0, p_{[q_1,ij]}, p_{[q_2,ij]}, \ldots, p_{[q_n,ij]}, 0, 0, \ldots, 0)$$

where the non-zero entries are from the position $[(i-1) * k + j - 1] * n + 1$ to $[(i-1)k + j - 1] * n + n$.

In the original **DPA M**, if the system is in some component M_i, then it remains in M_i for exactly k steps and after that it non-deterministically switches to some other component M_j, $1 \leq j \leq l, j \neq i$ for next k steps. This step is simulated in $\mathbf{M}'$ by forcing the transition from M_{i1} to M_{i2}, from M_{i2} to M_{i3},..., from $M_{i(k-1)}$ to M_{ik} and then from M_{ik} to M_{j1} , $1 \leq j \leq l$, $j \neq i$. Also $\mathbf{M}'$ makes sure that the system spends only one step in any component M_{ij}, $1 \leq i \leq l, 1 \leq j \leq k$ at any instance of time. The change of component from M_{ij} to $M_{ij(+1)}$, $\forall i, j$, $1 \leq i \leq l, 1 \leq j < k$ is done in $\mathbf{M}'$ by having the present stochastic row vector representing the current state distribution multiplied by the following matrix:

$$M(ij, i(j+1)) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where $M(ij, i(j+1))$ is a $(n * l * k)$ x $(n * l * k)$ matrix and I is a $n * n$ identity matrix staring at the entry $[(i-1)k + j - 1]n + 1$, $[(i-1)k + j]n + 1$.

The change from the component M_{ik} to M_{j1}, $\forall i, j$, $1 \leq i \leq l, 1 \leq j \leq l, i \neq j$ is done in $\mathbf{M}'$ by having the present stochastic row vector representing the current state distribution multiplied by the following matrix:

$$M(ik, j1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where $M(ik, j1)$ is a $(n * l * k)$ x $(n * l * k)$ matrix and I is a $n * n$ identity matrix staring at the entry $(ik - 1)n + 1$, $[(j-1)]n + 1$.

To understand the above notions, we take an example. Suppose we have $l = 3, k = 4, n = 2$. Then in $\mathbf{M}'$ the state configuration is represented by a 1 x 24 stochastic row vector. Suppose that at current instance the machine $\mathbf{M}'$ is in the component M_{22}, then the present configuration is :

$$\overline{s_1} = (0,0,0,0,0,0,0,0,0,0,p_{[q_1,22]},p_{[q_2,22]},0,0,0,0,0,0,0,0,0,0,0,0)$$

To switch the current component from M_{22} to M_{23} we multiply $\overline{s_1}$ with the following matrix:

$M(22,23)=$

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Multiplying $\overline{s_1}$ with $M(22,23)$ we get new state distribution as :

$$\overline{s_2} = (0,0,0,0,0,0,0,0,0,0,0,0,p_{[q_1,23]},p_{[q_2,23]},0,0,0,0,0,0,0,0,0,0)$$

such that $p_{[q_1,23]} = p_{[q_1,22]}$ and $p_{[q_2,23]} = p_{[q_2,22]}$.

Now we construct an equivalent probabilistic automata $\mathbf{M}''$ having single component with multiple choice for each alphabet symbol from $\mathbf{M}'$ as follows:

$$\mathbf{M}'' = (S'', s_0'', M'')$$

where

- $S'' = S'$.

- $s''_0 = s'_0$.
- M'' is the set of transition matrices.

The final state configuration in $\mathbf{M}''$ has to be changed. If the final state configuration in $\mathbf{M}$ is $(p_{q_1}, p_{q_2}, \ldots, p_{q_n})^T$, then the corresponding final state configuration in $\mathbf{M}''$ is represented as a nlk-dimensional column vector
$(0, 0, \ldots, 0(nk-n)times, p_{[q_1,1k]}, p_{[q_2,1k]}, \ldots, p_{[q_n,1k]}, 0, 0, \ldots, 0(nk-n)times, p_{[q_1,2k]}, p_{[q_2,2k]}, \ldots, p_{[q_n,2k]}, \ldots, 0, 0 \ldots, 0(nk-n)times, p_{[q_1,lk]}, p_{[q_2,lk]}, \ldots, p_{[q_n,lk]})^T$
where $p_{[q_i,jk]} = p_{q_i}$, $\forall i, j, 1 \leq i \leq n, 1 \leq j \leq l$. The matrices in the set M'' is constructed from the matrices $M_{11}, M_{12}, \ldots, M_{lk}$ as follows:

For each alphabet a, we define the transition matrix $M''(a)$ as follows:

$$M''(a) = \begin{bmatrix} M_{11}(a) & & & & & & & & \\ & M_{12}(a) & & & & & & & \\ & & . & & & & & & \\ & & & M_{21}(a) & & & & & \\ & & & & M_{22}(a) & & & & \\ & & & & & . & & & \\ & & & & & & M_{2k}(a) & & \\ & & & & & & & . & \\ & & & & & & & & M_{l1}(a) \\ & & & & & & & & & M_{l2}(a) \\ & & & & & & & & & & . \\ & & & & & & & & & & & M_{lk}(a) \end{bmatrix}$$

Here $M_{ij}(a) = M_i(a)$, $\forall i, j$, $1 \leq i \leq l, 1 \leq j \leq k$.

Example: Suppose we have $l = 2, k = 2, n = 2$, then $M^{\text{“}}(a)$ is defined as:

$$\begin{bmatrix} p_{[11,1]}(s_1,a) & p_{[11,2]}(s_1,a) & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{[11,1]}(s_2,a) & p_{[11,2]}(s_2,a) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{[12,1]}(s_1,a) & p_{[12,2]}(s_1,a) & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{[12,1]}(s_2,a) & p_{[12,2]}(s_2,a) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{[21,1]}(s_1,a) & p_{[21,2]}(s_1,a) & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{[21,1]}(s_2,a) & p_{[21,2]}(s_2,a) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{[22,1]}(s_1,a) & p_{[22,2]}(s_1,a) \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{[22,1]}(s_2,a) & p_{[22,2]}(s_2,a) \end{bmatrix}$$

where

$$\begin{bmatrix} p_{[11,1]}(s_1,a) & p_{[11,2]}(s_1,a) \\ p_{[11,1]}(s_2,a) & p_{[11,2]}(s_2,a) \end{bmatrix} = \begin{bmatrix} p_{[12,1]}(s_1,a) & p_{[12,2]}(s_1,a) \\ p_{[12,1]}(s_2,a) & p_{[12,2]}(s_2,a) \end{bmatrix} = M_1(a)$$

Similarly

$$\begin{bmatrix} p_{[21,1]}(s_1,a) & p_{[21,2]}(s_1,a) \\ p_{[21,1]}(s_2,a) & p_{[21,2]}(s_2,a) \end{bmatrix} = \begin{bmatrix} p_{[22,1]}(s_1,a) & p_{[22,2]}(s_1,a) \\ p_{[22,1]}(s_2,a) & p_{[22,2]}(s_2,a) \end{bmatrix} = M_2(a)$$

Now as in the case of $*$-mode, we will premultiply $M''(a)$ with the matrices which forces $\mathbf{M}''$ to switch the component from M_{ij} to $M_{i(j+1)}$ if $j < k$ or otherwise from M_{ik} to M_{j1} , $\forall 1 \leq i \leq l, 1 \leq j \leq l, i \neq j$.

Example: In the above example, we will have $M(11,12)(a) * M''(a)$, $M(21,22)(a) * M''(a)$, $M(12,21)(a) * M''(a)$, $M(22,11)(a) * M''(a)$.

So now in $\mathbf{M}''$ we have $l(k-1)+l(l-1)$ matrices in total for each alphabet symbol a. Thus from the distributed probabilistic automaton $\mathbf{M}$ we get a multiple choice probabilistic automaton $\mathbf{M}''$, and we have already proved that any multiple choice probabilistic automaton can be simulated by a probabilistic automaton(with single choice). Thus a distributed probabilistic automaton with $= k$-mode of cooperation in p-mode of acceptance is equivalent to a probabilistic automaton.

Case 3: $\alpha = \leq k$ **.**

The idea of the proof in this case will be same as that of $= k$-mode. Suppose we are given a distributed probabilistic automaton with l components

$$\mathbf{M} = (S, s_0, M_1, M_2, \ldots, M_l)$$

We construct $\mathbf{M}'$ from $\mathbf{M}$ and $\mathbf{M}''$ from $\mathbf{M}'$ in the same way as in the case of $= k$-mode. The final state configuration here has to be changed as it is changed in $= k$-mode, with slight difference. If the final state distribution in $\mathbf{M}$ is $(p_{q_1}, p_{q_2}, \ldots, p_{q_n})^T$, then the corresponding final state configuration in $\mathbf{M}''$ is represented by a nlk-dimensional column vector:
$(p_{[q_1,11]}, p_{[q_2,11]}, \ldots, p_{[q_n,1k]}, p_{[q_1,12]}, p[q_2,12], \ldots, p_{[q_n,12]}, \ldots, p_{[q_1,lk]}, p_{[q_2,lk]}, \ldots,$
$p_{[q_n,lk]})^T$, where $p_{q_i} = p_{[q_i,jm]}$, $\forall i,j,m, 1 \leq i \leq n, 1 \leq j \leq l, 1 \leq m \leq k$. The only difference here occurs in the choice of matrices in $\mathbf{M}''$. Since the machine $\mathbf{M}$ is working in $\leq k$-mode, the system remains in any component at most k steps after which it has to forcibly switch to some other component. This is simulated in $\mathbf{M}''$ by forcing the machine to make the transition from M_{ij} to $M_{i(j+1)}$ or M_{r1}, $\forall i,j,r : 1 \leq i \leq l, j < k, 1 \leq r \leq l, r \neq i$ or from M_{ik} to M_{r1} , $\forall i,k,r : 1 \leq i \leq l, 1 \leq r \leq l, r \neq i$.

Example : Considering the same example as in $= k$-mode with $l = 2, k = 2, n = 2$, in $\mathbf{M}''$ we will have $M(11,12) * M''(a)$, $M(11,21) * M''(a)$, $M(12,21) * M''(a)$, $M(21,22) * M''(a)$, $M(21,11) * M''(a)$, $M(22,11) * M''(a)$.

So, for each alphabet symbol a we will now have $ll(k-1) + l(l-1)$ matrices in total in $\mathbf{M}''$. Thus the distributed probabilistic automaton $\mathbf{M}$ is reduced to a multiple choice probabilistic automaton $\mathbf{M}''$ and we have proved that any multiple choice probabilistic automaton is equivalent to a probabilistic automaton with single choice for each alphabet symbol. Hence the theorem holds for the $\leq k$-mode also.

Case 4: $\alpha = \geq k$**.**

The proof is similar to the $\leq k$-mode. Suppose we are given a distributed probabilistic automaton with l components

$$M = (S, s_0, M_1, M_2, \ldots, M_l)$$

We construct $\mathbf{M}'$ from $\mathbf{M}$ and $\mathbf{M}''$ from $\mathbf{M}'$ in the same way as in the case of $= k$-mode. The only difference here occurs in the choice of matrices in $\mathbf{M}''$. Since the machine $\mathbf{M}$ is working in $\geq k$-mode, the machine spends at least k steps in each component before switching the component. This is simulated in $\mathbf{M}''$ by forcing the transition from M_{ij} to $M_{i(j+1)}$, $\forall i,j, 1 \leq i \leq l, j < k$ or otherwise from M_{ik} to M_{ik} or M_{r1}, $\forall i,r, 1 \leq i \leq l, 1 \leq r \leq l, r \neq i$. Also the final state configuration has to be changed exactly in the same way as in the case of $= k$-mode.

Example: Considering the same example as in $\geq k$-mode with $l = 2, k = 2, n = 2$, we will have $M(11, 12) * M''(a)$, $M(12, 12) * M''(a)$, $M(12, 21) * M''(a)$, $M(21, 22) * M''(a)$, $M(22, 22) * M''(a)$, $M(22, 11) * M''(a)$.

So, in general for each alphabet symbol a we will have $l(k-1) + l + l(l-1)$ matrices in $\mathbf{M}''$. Thus the original distributed probabilistic automaton is reduced to a multiple choice probabilistic automaton $\mathbf{M}''$ and we have shown that any multiple choice probabilistic automaton is equivalent to a probabilistic automaton with single choice for each alphabet symbol. Hence the theorem holds for the case $\alpha = \geq k$-mode also.

5 Conclusion

In this paper, we have considered multiple choice probabilistic automaton and we have shown that it can be simulated by a (single choice) probabilistic automaton. We defined distributed probabilistic automaton with four cooperating modes. We have shown that in each of the modes, a distributed probabilistic automaton can be simulated by a multiple choice probabilistic automaton and hence by a (single choice) probabilistic automaton. Here we have not considered probability of transition between components, which can also be taken into account and the above results can be established with appropriate modifications in the proofs.

References

[1] E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Păun. Grammar Systems, A Grammatical Approach to Distribution and Cooperation, Gordon and Breach, London 1994.

[2] K. Krithivasan, M. Sakthi Balan and P. Harsha, Distributed processing in automata, *International Journal of Foundation of Computer Science*, 10(4), p 443-464, 1999.

[3] K. S. Fu, Stochastic automata, stochastic languages and pattern recognition, *IEEE Symposium on Decision Control*, p 31-49, 1970.

[4] A. Salomaa, Probabilistic and weighted grammars, *Information and Control*, 15, p 529-544, 1969.

[5] K. Arthi and K. Krithivasan, Probabilistic parallel communicating grammar systems, *International Journal of Computer Mathematics*, 79(1) p 1-26, 2002.

[6] K. Arthi, K. Krithivasan and E. Csuhaj-Varjú, On the rule number of complexity of components of probabilistic cooperating grammar system, *JALC*, 7, p 433-446, 2002.

[7] K. Arthi, K. Krithivasan and S. V. Raghvan, A generative model for capturing user behaviour in computer networks, *Proceedings of SCI 2001*, Vol 5, p 162-167, 2001.

[8] Azaria Paz. Introduction to Probabilistic Automata, Academic Press, 1971.