70-20,960

ELLIS, Clarence Arthur, 1943-
PROBABILISTIC LANGUAGES AND AUTOMATA.

University of Illinois at Urbana-Champaign,
Ph.D., 1970
Mathematics

University Microfilms, Inc., Ann Arbor, Michigan

PROBABILISTIC LANGUAGES AND AUTOMATA

BY

CLARENCE ARTHUR ELLIS
B.A., Beloit College, 1964
M.S., University of Illinois, 1966

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois, 1970

Urbana, Illinois

UNIVERSITY OF ILLINOIS

THE GRADUATE COLLEGE

October 1969

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY

SUPERVISION BY CLARENCE ARTHUR ELLIS

ENTITLED PROBABILISTIC LANGUAGES AND AUTOMATA

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF Doctor of Philosophy in Computer Science

David E. Muller
In Charge of Thesis

J R Pasta
Head of Department

Recommendation concurred in†

David Kuck
Robert S. Northcote
Jurg Nievergelt
Franz E. Hohn

Committee
on
Final Examination†

† Required for doctor's degree but not for master's.

D517

ACKNOWLEDGMENT

Profuse thanks are due to Professor D. E. Muller for his advice and encouragement during the preparation of this thesis. The author is also indebted to the Department of Computer Science, University of Illinois, for its support and to Miss Barbara Hurdle for her typing of this thesis.

Finally, the author extends his appreciation to his wife, Anna, whose patience and understanding assistance were of great value.

TABLE OF CONTENTS

1. INTRODUCTION

In recent years, much work has been done on extensions of the theory of finite automata to obtain models of acceptors and translators of programming languages. Examples are pushdown store automata[5], stack automata[6], minimax automata[23], and balloon automata[10]. There are many others. The purpose of this thesis is not to simply introduce another type of automaton, but to describe a general concept which can be adapted to any of the automata present in the literature.

It is quite natural to assign probabilities (or frequencies) to the strings of a language to try to get some quantitative measure of "efficiency" of grammars and translators. The model obtained by doing this is called a probabilistic language, which may be considered a fuzzy set[27], containing all valid sentences of the language together with a grade-of-membership function for these sentences. Acceptors and generators for these probabilistic languages are defined as Probabilistic Automata and Probabilistic Grammars, respectively. Specifically, Context Free Probabilistic Languages are explored in depth in this thesis.

This investigation does not consider how one would find the "best" grammar or automaton for a language, or how to improve a given grammar. Indeed, the meaning of "best" is open to many interpretations. The related idea of finding good approximation grammars for languages is also unexplored. It is hoped that the tools developed here will lead to quantitative analysis in these and other areas.

2. BASIC DEFINITIONS AND NOTATION

This section presents notation and concepts which have been previously defined in the literature and are heavily used in this paper. Then these definitions are altered to form probabilistic analogues. A <u>language</u> over a set T of terminal symbols is a subset of the set T^* of all strings over T. A <u>phrase structure grammar</u> over a set T is a system (N, P, S) in which N is a finite set of nonterminal symbols, P is a set of rules (called productions) of the form $(\Psi \to \xi)$ where ζ is any string of symbols of $T \cup N$ (denoted $\zeta \varepsilon (T \cup N)^*$) and Ψ is any non-empty string of symbols of $T \cup N$, (denoted $\Psi \varepsilon (T \cup N)^+$). Ψ is called the generatrix of the production, and ζ is called the replacement string. $S \varepsilon N$ is the initial nonterminal.

Notation: Hereafter, when discussing languages and grammars, A, B, and C will always denote elements of N, while X, Y, and Z denote strings over N. Similarly, a, b, $c \varepsilon T$; x, y, $z \varepsilon T^*$; $\alpha_1, \beta_1, \gamma \varepsilon T \cup N$; $\Psi, \chi, \zeta \varepsilon (T \cup N)^*$. If $\chi = \alpha_1 \alpha_2 \ldots \alpha_n$ then the length of χ is $\ell(\chi) = n$. The null string is denoted by $\lambda(\ell(\lambda) = 0)$ and the empty set is ϕ. $\hat{L}$ always denotes a language, $\hat{G}$ is a grammar, and $\hat{A}$ is an automaton. I denotes the set of <u>positive</u> integers, ¶ the rationals, and R the reals. Let $\hat{G} = (N, P, S)$ be a grammar. If $\chi = \Psi_1 \Psi \Psi_2$ and $(\Psi \to \zeta) \varepsilon P$, then we write $\chi \to \Psi_1 \zeta \Psi_2$. If $\exists$ strings $\zeta_0 \zeta_1 \ldots \zeta_n$ such that $\zeta_{i-1} \to \zeta_i$, then we write $\zeta_0 \Rightarrow \zeta_n$ and we say there is a <u>derivation</u> of ζ_n from ζ_0 with respect to $\hat{G}$. The language $\hat{L}$ generated by a grammar $\hat{G}$ is $L(\hat{G}) = \{x | S \Rightarrow x, x \varepsilon T^*\}$. If $\hat{L}$ is generated by a grammar with all productions $(\Psi \rightarrow \zeta) \ni \Psi \varepsilon N$, then $\hat{L}$ is a context free language. If further, ζ is of the form aB for some $a \varepsilon T$, $B \varepsilon N \cup \{\lambda\}$ in all productions of $\hat{G}$, then $\hat{L}$ is a regular language.

3. PROBABILISTIC GRAMMARS AND LANGUAGES

Definition: A Probabilistic Language (P language) over T is a system $\hat{L} = (L, \mu)$ where L is a class of words formed from T and μ is a measure on the set L. If μ is a probability measure, then $\hat{L}$ is a Normalized Probabilistic Language (NP language).

Definition: A Probabilistic Grammar (P grammar) over T is a system $\hat{G} = (N, P, \Delta)$ where N is the finite set of nonterminals, $A_1, A_2, \ldots A_n$, Δ is an n-dimensional vector, $(\delta_1 \ldots \delta_n)$ with δ_i being the probability that A_i is chosen as the initial nonterminal, and P is a finite set of probabilistic productions, $\Psi_i \xrightarrow{p_{ij}} \zeta_j$, with $\Psi_i \varepsilon (N \cup T)^+$, ζ_j $\varepsilon (N \cup T)^+$, and p_{ij} ε R ($p_{ij} \neq 0$). If Δ is stochastic, if $0 < p \leq 1$, and if $\sum_j p_{ij} = 1$ for every generatrix Ψ_i contained in productions of P, then $\hat{G}$ is a Normalized Probabilistic Grammar (NP grammar).

If all productions of $\hat{G}$ are of the form $A \xrightarrow{p} aB$ or $A \xrightarrow{p} a$, $A \varepsilon N$, $B \varepsilon N$, $a \varepsilon T$, then $\hat{G}$ is called a left linear P grammar. The probability of a derivation of ζ_n from ζ_0 is defined as

$pr(\zeta_0 \Rightarrow \zeta_n) = \sum_{i=1}^{k} \prod_{j=1}^{k_i} p_{ij}$ where k is the number of derivations of ζ_n from ζ_0, k_i is the number of derivation steps, $\zeta_{i,j-1} \xrightarrow{p_{ij}} \zeta_{i,j}$ used in the i-th derivation, and p_{ij} is the probability associated with the j-th step of the i-th derivation. The derived probability of a terminal string $x \varepsilon T^{+}$ with respect to a left linear grammar $\hat{G}$ is $\mu(x) =$

$\sum_{i=1}^{n} (\delta_i \; pr(A_i \Rightarrow x))$ where $N = \{A_1, A_2, \ldots A_n\}$, $\Delta = (\delta_1 \; \delta_2 \ldots \delta_n)$.

The P language generated by $\hat{G}$ is $\hat{L} = (T^{+}, \mu)$ where $\mu(x) =$ the derived probability of x. An admissible P grammar (see Greibach[8]) is a grammar in which there exists a derivation of some $x \varepsilon T^{+}$ from each $A \varepsilon N$. A generalized admissible P grammar is one in which there exists a production with A in the generatrix for each $A \varepsilon N$.

Theorem 1: Every normalized left linear admissible P grammar $\hat{G}$ generates a normalized P language.

Proof: Define an (n+1) x (n+1) matrix $U = [u_{ij}]$ as follows:

$$u_{ij} = \sum_{\substack{a \, \varepsilon \, T \\ (A_i \rightarrow aA_j) \varepsilon P}} pr(A_i \rightarrow aA_j), \; i \leq n, \; j \leq n$$

$$u_{ij} = \sum_{\substack{b \, \varepsilon \, T \\ (A_i \rightarrow b) \varepsilon P}} pr(A_i \rightarrow b), \; i \leq n, \; j = n + 1$$

$$u_{ij} = 0, \quad i = n + 1, \; j \leq n$$

$$u_{n+1, \, n+1} = 1$$

$u_{i, n+1}$ is by definition the total probability of a derivation from A_i of a terminal string of length 1. Considering powers of the matrix U, $u^{k}_{i, n+1}$ gives the total probability of derivation from A_i of a string of length $\leq k$. If U^{k} is pre-multiplied by the row vector Δ augmented by zero, $\Delta' = (\delta_1 \; \delta_2 \ldots \delta_n, 0)$ then the (n+1)-st element in the resulting vector represents the sum of the derived probabilities of all $x \varepsilon T^{+} \ni \ell(x) \leq k$.

Finally, $\sum_{x \in T^+} \mu(x) = \lim_{k \to \infty} \sum_{\substack{x \in T^+ \\ \ell(x) \leq k}} \mu(x) = \lim_{k \to \infty} (\Delta' \cdot U^k)_{n+1}$. Since $\hat{G}$ is normalized, U is a stochastic matrix; and since $\hat{G}$ is admissible, $\exists\, k \in I \ni u^k_{i,n+1} > 0$ for i = 1, 2...n+1. Thus, using the theory of Markov Chains[6], $U^k = \begin{pmatrix} t^k_1 \\ t^k_2 \\ t^k_{n+1} \end{pmatrix}$ where each row vector t^k_i approaches a steady state vector t as k approaches infinity. $t^k_{n+1} = (0\ 0 \ldots 0 1) \forall\, k \in I$ implies $t = (0\ 0 \ldots 0\ 1)$ and $\lim_{k \to \infty} (\Delta' \cdot U^k) = \Delta' \lim_{k \to \infty} (U^k) = \Delta' \cdot \begin{pmatrix} t \\ t \\ \vdots \\ t \end{pmatrix}$. Thus, the (n+1)-st element is $\left(\Delta' \cdot \begin{pmatrix} t \\ t \\ \vdots \\ t \end{pmatrix}\right)_{n+1} = \sum_{i=1}^{n+1} (\delta_i) = 1$. QED.

A P language which is generated by a left linear P grammar is called a regular P language.

Theorem 2: There exists a regular language L with a probability $\mu(x)$ assigned to each $x \in L$ such that no left linear P grammar generates (L, μ).

Proof: The proof will consist simply of exhibiting such a language.

(1) Let $T = \{a\}$; then T^+ is the set of strings $\{a^n | n \in I\}$.

(2) Assign probabilities to these strings $\mu(a^{n+1}) = \frac{1}{\sqrt{\tau_n}}$, $n > 0$,

where $\tau_1 = 4$, τ_i = smallest prime $\ni \tau_i > \max(\tau_{i-1}, 2^{2i})$ for $i > 1$.

(3) Assign $\mu(a) = 1 - \sum_{i=1}^{\infty} \frac{1}{\sqrt{t_n}}$. This guarantees that $\sum_{n=1}^{\infty} \mu(a^n) = 1$.

Next we show that no left linear P grammar generates the language (T^+, μ).

(1) Suppose the grammar $\hat{G} = (N, P, \Delta)$ is alleged to generate (T^+, μ). Then all $\mu(a^n)$ are in the field of numbers generated by the rationals with field extensions p_i where p_i is the probability associated with the i-th production of P if $0 < i \leq |P|$, and p_i is the probability δ_j in the vector Δ if $i = |P| + j$. This field is denoted $¶(p_1 \ldots p_k)$, where $k = |P| + |N|$.

(2) If all p_i are in the field ¶ or are algebraic extensions of it, then the total extension is of finite degree. Consider the extension $(\frac{1}{\sqrt{t_1}}, \frac{1}{\sqrt{t_2}}, \ldots)$. This may be written as a union of fields each of which is a finite extension of degree 2 of the previous field. Thus, $\bigcup_{n=1}^{\infty} ¶(\frac{1}{\sqrt{t_1}}, \frac{1}{\sqrt{t_2}}, \ldots, \frac{1}{\sqrt{t_n}})$ is a field[26] whose degree must be infinite. Thus all of these irrationals cannot be within the finite degree algebraic field extension $¶(p_1 \ldots p_k)$. Since all derived probabilities under the grammar $\hat{G}$ of finite strings are expressible as finite sums of products, these derived probabilities must be within $¶(p_1 \ldots p_k)$. Thus (T^+, μ) cannot be derived using $\hat{G}$.

(3) If some of the p_i are transcendental extensions, then $¶(p_1 \ldots p_k)$ can be obtained by a pure transcendental extension $¶(p_1 \ldots p_\ell) = Q$

followed by an algebraic extension of finite degree $Q(p_{\ell+1}\cdots p_k)$. In this case, $\frac{1}{\sqrt{\tau_i}} \notin \P(p_{\ell+1}\cdots p_k)$ implies $\frac{1}{\sqrt{\tau_i}} \notin Q(p_{\ell+1}\cdots p_k)$ by the following argument. Let the polynomial $f(x) = x^2 - \frac{1}{\tau}$ be irreducible over $\P_i = \P(p_1\cdots p_i)$ but reducible over $\P_{i+1} = \P_i(p_{i+1})$, where p_{i+1} is transcendental $(i \le \ell)$. Then $f(x) = (x - \alpha)(x + \alpha)$, $\alpha \in \P_i(p_{i+1})$. α is expressible as $\frac{g(p_{i+1})}{h(p_{i+1})}$ where $\frac{g}{h}$ is in reduced form and not in $\P_i$. $(\frac{g}{h})^2 = \frac{1}{\tau} \in R$. $g^2 - \frac{1}{\tau}h^2 = 0$. But this equation implies that p_{i+1} is algebraic over $\P_i$ which is a contradiction $\Rightarrow \Leftarrow$. Thus if $f(x)$ is irreducible over $\P_i$, then it is irreducible over $\P_{i+1}$. This can be applied not 1 but ℓ times to yield

$\frac{1}{\sqrt{\tau}} \notin \P \Rightarrow \frac{1}{\sqrt{\tau}} \notin \P(p_1\cdots p_\ell)$. Using the previous part (2) of this proof for the algebraic elements $p_{\ell+1}\cdots p_k$, we get

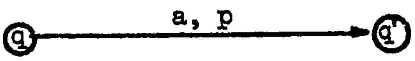
$\exists\, \tau \ni \frac{1}{\sqrt{\tau}} \notin \P(p_1\cdots p_k)$. QED.

4. PROBABILISTIC AUTOMATA

The idea of the probabilistic finite automaton was originally conceived by Rabin.[17] Basically, if an automaton is in some state q, and receives an input a, then it can move into any state, and the probability of moving into state q' is p(q, a, q'). Rabin requires that $\sum_{q' \varepsilon Q} p(q, a, q') = 1$ (called type 1 normalization in this paper) for all q in the set of states Q, and for all a ε T. Practical motivation for this requirement is that these automata can model sequential circuits which are intended to be deterministic, but which exhibit stochastic behavior because of random malfunctioning of components. Thus p(q, a, q') is interpreted as the conditional probability of q' given q and a, pr(q'|q,a), so by the theorem of total probability, $\sum_{q' \varepsilon Q} pr(q'|q,a) = 1.$

Other interpretations may give rise to other normalizations. For example, in performing the state identification experiment with a probabilistic automaton, one might interpret p(q, a, q') as pr(q, q'|a). This implies a normalization by summing over all possible q, q' values. $\sum_{q \varepsilon Q} \sum_{q' \varepsilon Q} p(q, a, q') = 1.$ In fact, eight different types of probabilistic automata can be defined by the various interpretations listed in the following table.

Normalizations for Probabilistic Finite Automata



TYPE	INTERPRETATION	NORMALIZATION		
1	pr(q'|q, a)	$\sum_{q' \varepsilon Q}$	p(q, a, q') = 1	∀ q ε Q, a ε T
2	pr(q|a, q')	$\sum_{q \varepsilon Q}$	p(q, a, q') = 1	∀ q', ∀a
3	pr(a|q, q')	$\sum_{a \varepsilon T}$	p(q, a, q') = 1	∀ q, ∀ q'
4	pr(q', a|q)	$\sum_{a \varepsilon T} \sum_{q' \varepsilon Q}$	p(q, a, q') = 1	∀ q
5	pr(q, a|q')	$\sum_{a \varepsilon T} \sum_{q \varepsilon Q}$	p(q, a, q') = 1	∀ q'
6	pr(q, q'|a)	$\sum_{q \varepsilon Q} \sum_{q' \varepsilon Q}$	p(q, a, q') = 1	∀ a
7	pr(q, a, q'|)	$\sum_{q} \sum_{a} \sum_{q'}$	p(q, a, q') = 1	
0	pr(|q, a, q')		p(q, a, q') = 1	∀ q, a, q'

One of the important theorems concerning finite automata, which was first proved by Kleene in 1956[11] states that for every left linear grammar, there exists an automaton which accepts all and only the strings generated by the left linear grammar and conversely, there is a left linear grammar which generates all and only the strings accepted by any finite automaton. Surprisingly, an identical theorem was proved by Chomsky and Schutzenberger in 1963[3] concerning context free languages and pushdown store automata. The analogous problems for probabilistic automata are attacked in this paper. If the symbols a ε T are interpreted as outputs instead of inputs, then the automaton becomes a generator similar to a grammar. In this case, type 4 normalization must be chosen so that an NP grammar will correspond to an NP automaton.

Definition: A <u>Probabilistic Automaton</u> (P automaton) over T is a system $\hat{A} = (Q, M, S, \Xi)$ where Q is a finite set of states, S is a finite set of storage tape symbols, Ξ is an initial state vector and M is a function, called a probabilistic transition function, which has associated with it a second function p. The specific nature of these functions determines the type of P automaton defined. If Ξ is a stochastic vector and if $\hat{A}$ is constrained to some normalization type, then $\hat{A}$ is a <u>Normalized Probabilistic Automaton</u> (NP automaton). Cases in which $S = \phi$ will be simplified to $\hat{A} = (Q, M, \Xi)$. Particular classes of automata are obtained by attaching constraints to the general definition. The following table lists some of the automata definable, and their range ($\mathcal{R}(M)$) and domain ($\mathcal{D}(M)$) constraints on the mapping M, and their normalization constraints.

<u>Types of Automata</u>

1. Deterministic Finite Automaton

Norm Constraints:	Type 1
RD Constraints:	$\mathcal{D}(M) = Q \times T$, $\mathcal{R}(M) \subseteq Q$

2. Nondeterministic Finite Automaton

Norm:	Type 0
RD:	$\mathcal{D}(M) = Q \times T$, $\mathcal{R}(M) \subseteq \mathcal{P}(Q)$

3. Probabilistic Rabin Automaton

Norm:	Type 1
RD:	$\mathcal{D}(M) = Q \times T$, $\mathcal{R}(M) \subseteq \mathcal{P}(Q)$

4. Probabilistic Ellis Automaton

Norm:	Type 4
RD:	$\mathcal{D}(M) = Q \times T$, $\mathcal{R}(M) \subseteq \mathcal{P}(Q) \cup \{\lambda\}$