ON FORMAL LANGUAGES, PROBABILITIES, PAGING AND DECODING ALGORITHMS.

Joffroy BEAUQUIER (*)
Loys THIMONIER (**) (*~)

(*) Laboratoire de Recherche en Informatique (LRI)-Université de PARIS-SUD Bâtiment 490 - 91 405 ORSAY Cédex - FRANCE
(**) U.E.R. de Mathématiques - Université d'AMIENS 33 rue St Leu 80039 AMIENS Cédex FRANCE.

ABSTRACT

In previous papers [BT 83] [BT 84] [BT 85], it was indicated how a probabilistic parameter, namely the Bernoullian density, could be computed by means of an explicit formula or numerically with a given precision for several structural types of formal languages L. We present here two general methods for this computing, a deterministic one and a Monte-Carlo method, using the generating function of the prefix-free language Pref(L) associated with L, and a recognition algorithm for Pref(L) assumed to be context-free. The results are applied for the obtainment of new algorithms for two classical problems: paging and decoding, within a probabilistic framework of language theory.

I. INTRODUCTION

In the present paper, we are mainly interested in the theoretical aspect of problems, regardless of the implementation of corresponding algorithms. So it should be viewed as a probabilistic contribution to language theory, with applications to the solution of practical problems. The reader is assumed to be familiar with formal languages [Har 78] and discrete probabilities [HU 79].

First we must recall what the probabilistic notion of Bernoullian density [BT 83] is. Let L be a formal language over a finite alphabet A= a1,...,am, i.e. a part of the free monoid A* generated by A; A is provided with a probability distribution p=[p1,...,pm]. A sequence of Bernoullian drawings (with replacement in A) is executed possibly producing a first word w of L which stops the process (w is necessarily a prefix-free word of L : w=w1w2, wEL = w1=w, w2=E the empty word); otherwise, the process never stops, producing an infinite word, whose no left factor is in L; such a process, finite or infinite, is called a Bernoulli process.

The Bernoullian density of L is the probability of obtaining a finite process. According to the Law of large numbers, this notion corresponds to the outcome frequency of a (prefix-free) word of L during the repetition of a large number of processes (e.g. during a simulation). Let Pref(L) be the set of the prefix-free words of L (prefix-free language [Bea 83] associated with L); the probability of producing the word w=a1...am after l drawings is p(w)=p1...pl and we have \( \delta_{L,p}(w) = \sum_{w \in \text{Pref}(L)} p(w) \) (by consideration of mutually exclusive events) = \( \sum_{n=1}^{\infty} \sum_{w \in \text{Pref}(L)^n} p(w) \). We denote by \( \delta_{L,p} \) the value corresponding to the equally likely case (playing always an important part in the probabilistic problems). \( \delta_{L,p} = \sum_{n=1}^{\infty} \binom{\sum_{i=1}^{m} c_i}{n} \sum_{w \in \text{Pref}(L)^n} p(w) \) (C with \( \alpha_1...\alpha_m=\text{number of words of Pref(L)} \) \( \pi^n \), of commutative image \( a_1^{\alpha_1}...a_m^{\alpha_m} \)).

We consider now the question of computing \( \delta_{L,p} \) by means of an explicit formula or numerically with a given precision. It occurs about the theoretical problem of obtaining numerical parameters characterizing formal languages, and also about practical problems using probabilistic algorithms. Let \( R_{N,p} \) be the remainder of order N in the series of sum \( \delta_{L,p} \). \( R_{N,p} \) is the sum of all the terms in p1...pm with \( \alpha_1+...+\alpha_m > N \); we must find N to have \( R_{N,p} < \beta \). This is a non obvious question, depending on the properties of the words of the language to infinity.

It was already solved for some languages [BT 83] [BT 84] [BT 85], with diverse structural properties. It can be noticed that, in case of regular languages, \( \delta_{L,p} \) is obtained by a matrix inversion, therefore with a polynomial time \( O(s^p) \) (\( s = \log_2 7 \approx 2.81 \) [BH 74]), if s is the number of states of a finite automaton recognizing Pref(L) [BT 84]. We try here to find links between these problems of
computation, and, on the one hand the structural properties of \( L \), on the other hand its combinatorial properties (here the sequence of the number \( n \) of prefix-free words of length \( n \)). In section II we give two general methods for this computing: a deterministic one and a Monte-Carlo method. They use a recognition algorithm for Pref(L) assumed to be context-free, the generating function of Pref(L), and a counting condition allowing to obtain for \( R_{N,p} \) an upper bound in one variable \( p_+ = \max(p_1, \ldots, p_m) \).

The case of an unambiguous language Pref(L) is considered, with the example of Dyck languages. Then we obtain and compare time complexities of the two methods.

In section III, the previous results are applied for the obtainment of a new probabilistic paging algorithm, whereas in section IV we consider a new decoding algorithm for a message transmitted through a noisy channel.

II. COMPUTATION OF THE BERNOUILLIAN DENSITY BY MEANS OF A GENERATING FUNCTION AND A RECOGNITION ALGORITHM.

II.1. A deterministic method

The generating function of a language \( L \) over \( A \) is the sum \( \lambda(z) = \sum_{n=1}^{\infty} \lambda_n z^n \) with \( \lambda_n = \text{card}(L \cap A^n) \). Its convergence radius \( \rho_L \), called the convergence radius of \( L \), verifies \( \rho_L \leq m \), if \( m = \text{card} A \). We consider now the prefix-free language Pref(L) associated with \( L \), and its generating function \( \zeta(z) = \sum_{n=1}^{\infty} \sigma_n z^n \) with \( \sigma_n = \text{card}(\text{Pref}(L) \cap A^n) \). By means of the probabilistic expression of \( \delta_{L,p} \), we obtained in the equally likely case \([BT 84]\) : \( \delta_{L,p} = \frac{1}{m} \).

In the general case, let \( p_+ = \max(p_1, \ldots, p_m) \) an elementary result is: \( \delta_{L,p} = \sum_{n=1}^{\infty} (p_+) \delta_{L,p}^n \), the remainder of order \( N \) is bounded by the remainder of the order \( N \) of \( \delta_{L,p} \), if \( p_+ \in [1/m, 1] \). We know an upper bound of this last remainder (practically, with geometrical series), we can compute \( \delta_{L,p} \), with a given precision. Obviously \( \delta_{L,p} \), so we assume that the following condition is met (counting condition): one has at one's disposal an upper bound for \( \delta_{L,p} \); polynomial \( \delta_{L,p} \) or exponential \( \delta_{L,p} \).

We consider first the case of an exponential upper bound for \( \delta_{L,p} \); such a language is \( \text{PAL}_m \) (the palindroms over \( a_1, \ldots, a_m \)), for which \( \delta_{L,p} \leq \frac{1}{m} \leq \frac{1}{m} \). We obtain: (Proposition II.1.1) if \( \delta_{L,p} \leq \frac{1}{m} \) (so \( \text{Pref}(L) \geq 1/a \)), then \( \delta_{L,p} \) can be computed with a given precision.

We consider now the case of a polynomial upper bound for \( \delta_{L,p} \); such a language is Goldstine's language \( \{a_{n_1} b a_{n_2} b \ldots a_{k} b k \geq 1/n_i > 0 / i \} \) for which \( \delta_{L,p} \leq \frac{1}{2n} \) (and \( \delta_{L,p} \leq \frac{1}{2n} \)). We obtain: (Proposition II.1.2) if \( \delta_{L,p} \leq \frac{1}{m} \), then \( \delta_{L,p} \) admits an upper bound which is a \( O(N^{-\frac{1}{p}}) \) term, computable by means of a geometrical series in \( p_+ \) and its derivatives, and \( \delta_{L,p} \) can always be computed with a given precision.

Time complexity

The effective use of this method needs an algorithm for recognizing the words of Pref(L). If this one is context-free, the Cocke-Kasami-Younger algorithm \([Mar 78]\) allows e.g. this recognition, with an \( O(n^3) \) time if \( n \) is the length of the word. So it can be obtained: (Proposition II.1.3) if \( N \) is the first number, obtained by means of the propositions II.1.1 or II.1.2, yielding \( R_{N,p} < \varepsilon \) (\( \varepsilon \) is the given precision), then this deterministic method admits a time complexity \( O(N^3 m^3) \).

II.2. The case of an unambiguous context-free language

A classical theorem of Chomsky and Schützenberger \([CS 63]\) states the algebraicity of the generating function of an unambiguous context-free language \( L \) from the obtainment of a system of polynomial equations whose a component is the generating function \( \lambda(z) \) of \( L \), thanks to substitution rules in an unambiguous grammar \( G \) generating \( L \).