The complexity of generating an exponentially distributed variate

Philippe Flajolet, Nasser Saheb

IRIA
CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. 954.90.20

Rapports de Recherche

N° 159

THE COMPLEXITY OF GENERATING AN EXPONENTIALLY DISTRIBUTED VARIATE

Philippe FLAJOLET
Nasser SAHEB

Septembre 1982

THE COMPLEXITY OF GENERATING

AN EXPONENTIALLY DISTRIBUTED VARIATE

Philippe FLAJOLET - Nasser SAHEB

INRIA
78150 Rocquencourt
France

June 1982

Abstract :

We analyze in detail an almost optimal algorithm for generating an exponentially distributed variate. The algorithm is due to Knuth and Yao and relies on a method which goes back to J. Von Neumann. It is shown that it can generate k bits of an exponentially distributed variate using an average of about k+5.67974692 coin flippings. This solves a problem left open by Knuth and Yao [KY 76].

Résumé :

Cet article présente l'analyse détaillée d'un algorithme quasi-optimal de génération d'une variable exponentiellement distribuée. L'algorithme qui est dû à Knuth et Yao repose sur une méthode qui remonte à John Von Neumann. On montre que par cette méthode, on engendre k bits d'une variable exponentiellement distribuée en utilisant une moyenne d'approximativement k+5.67974692 tirages binaires élémentaires. Cette analyse répond à une question ouverte de Knuth et Yao.

1 - Introduction

In [KY 76], Knuth and Yao address a number of essential questions related to the generation of random numbers with non-uniform distributions. One of their results states that for any distribution over the reals, a random variable X with that distribution can be generated by a tree algorithm which will output k bits of X after inputting less that (k+2) random (uniform) bits on the average. However the optimum tree algorithm that achieves this bound has the disadvantage of being in general infinite.

In the particular case of generating an exponentially distributed variate an old method exists which is due to J. Von Neumann [VN 51]. Knuth and Yao have worked a careful bit level implementation of this method. Based on 1000 simulations, they conjecture that the average cost c(k) of producing k bits of the exponential variable is about

$$\bar{c}(k) \approx k + 5.4 \pm 0.2 + o(1). \qquad (1)$$

We ran 1 000 000 simulations and found for this sample the empirical estimate of $\bar{c}(k)$:

$$\bar{c}(k) \approx k + 5.675996 + o(1) \qquad (2)$$

which is very slightly off the conjectured bounds (1). From the analysis given below will result that in fact

$$\bar{c}(k) = k + \gamma + o(1)$$

where

$$\gamma = e\left(2+\frac{3}{e-1}\right) + \sum_{k\geq 0} \frac{1}{e^{1/2^k}-1}\left[\frac{e}{4^k} + 1 - e^{1/2^k}\left(1-\frac{1}{2^k}\right)\right] \qquad (3)$$

a constant which numerically evaluates to 5.67974 69285 27492, in good agreement with (2).

This result together with companion results on the distribution of costs is proved by expressing the characteristic parameters of the algorithm in terms of digital search trees -or tries-, a structure also closely associated to radix sorting [Kn 73].

The plan of the paper is as follows : in Section 2, we present the basic Von Neumann-Knuth-Yao algorithm ; Section 3 contains the analysis itself and the proof of the basic result corresponding to Equation (3). Section 4 concludes with a calculation of the distribution of costs.

2 - The Von Neumann-Knuth-Yao algorithm

The problem considered in this paper is the generation of a random number X with an exponential distribution, i-e such that

$$Pr(X \leq x) = 1 - e^{-x} \tag{4}$$

or equivalently

$$Pr(x < X \leq x+dx) = e^{-x}\, dx. \tag{5}$$

The basic idea of Von Neumann is to generate (4) by synthesizing Taylor expansion of the cumulative distribution function in a probabilistic manner. Let $\{Y_j\}_{j\geq 0}$ be a sequence of independently uniformly distributed variables over the real interval [0;1]. With probability 1, there will be an n such that

$$Y_0 > Y_1 > Y_2 \cdots > Y_{n-1} \leq Y_n. \tag{6}$$

Let G_n denote event (6). Then, the probability that we have

$$x < Y_0 \leq x+dx$$

and G_n is found to be

$$\left[\frac{x^{n-1}}{(n-1)!} - \frac{x^n}{n!} \right] dx \quad . \tag{7}$$

Thus if we draw an infinite sequence $Y = \{Y_j\}_{j\geq 0}$ and keep the first element Y_0 if n is odd -we call the event a success-, discard the sequence otherwise -call this element a failure-, the variable Y_0 is defined with probability $(1-e^{-1})$ and summing (7) over odd values of n, we find :

$$Pr(x<Y_0\leq x+dx) = e^{-x}dx \qquad 0\leq x<1 \quad .$$

Thus Y_0 generates the portion $0\leq x<1$ of the distribution (5).

If we repeatedly draw sequences Y until the associated n satisfying event G_n is odd, the number of draws D will satisfy :

$$Pr(D=k) = (1-e^{-1})\, e^{-(k-1)} \quad . \qquad (8)$$

From this can be seen that $D + Y_0$ (with Y_0 in the last sequence drawn) is an exponentially distributed variable. The Von Neumann algorithm is thus :

1. Set D := 0 ;
2. Generate $Y_0, Y_1, Y_2, \ldots$ until finding the first $n\geq 1$ such that $Y_{n-1}<Y_n$
3. If n is even, set D := D+1 and return to step 2, otherwise output $D + Y_0$.

In programming the algorithm, one only needs at each stage to retain the current value of Y_0 and the last two Y_j to be compared. With this observation, the algorithm becomes :

```
program EXP-VN ;
var D,N : integer ; U,V,X,YO : real ;
begin
        D := 0 ;
        repeat
             D := D+1 ; N := 1 ;
             U := uniform ; V := uniform ; YO := U ;
             while U > V do
                 begin U := V ; V := uniform ; N := N+1
                 end
        until odd (N) ;
        X := D-1 + YO ;
        output (X)
end
```

There the procedure uniform generates a variable uniformly distributed over the real interval [0;1].

The Knuth-Yao implementation of this algorithm consists in generating U and V in the above algorithm (i-e the Y_j's) bit by bit in such a manner that only those bits that are necessary for comparison are generated. The implementation can take advantage of the fact that we actually only need to keep track of the bits of U (the current Y_j) computed so far : instead of drawing the bits of V (the current Y_{j+1}), one need only flip coins to decide if these coincide with the corresponding bits of U. We refer the reader to the reference [KY 76] for a more detailed presentation. With these conventions, we obtain the following algorithm, a mere stylistic variant of Knuth and Yao's.

```
program EXP-KY
var D,N,S,T : integer ;
    U,YO : array [1..∞] of integer ;
  begin
  D := 0 ;
  repeat
    D := D+1 ; N := 1 ; T := 0 ;
    repeat T := T+1 ; U[T] := flip ; YO[T] := U[T] ;
    until flip = 1 ;
    S := T ;
    while U[T] = 1 do
          begin N := N+1 ; U[T] := 0 ; I := 0 ;
          repeat I := I+1 ; if I > T then U[I] := flip ;
          until flip = 1 ;
          T := I ;
          end
until odd (N) ;
produce (D-1,YO,etc)
end
```

In the above program, the following conventions have been adopted : U and YO are vectors of potentially infinite dimension ; "flip" is a procedure that generates random bits with equal probabilities for 0 and 1's ; as to "produce", it produces the representation of a real number of integral part D-1, of first S fractional bits equal to YO[1], YO[2]... YO[5], the succeeding bits (etc) being randomly chosen 0-1 bits. In the Knuth-Yao mixed unary-binary representation where the integral part is represented by a sequences of ones of corresponding length, the decimal point is represented by a separating zero, and the bits in the fractional parts are represented as usual in binary, the procedure "produce" reads :

```
for I := 1 to D-1 do output (1) ;
output (0) ;
for I := 1 to S do output (YO[I]) ;
for I := 1 to ∞ do output (flip) ;
```

This procedure can of course be trivially modified to output only k bits of the result by truncating the sequence of output bits in an appropriate way.

The cost of the algorithm measured by the number of elementary coin flips for generating k bits of the exponential variate when C coin flips have taken place in the body of the program (the outer repeat loop) is simply :

$$c(k) = \begin{cases} k + C - S - D & \text{if } k \geq S + D \\ C & \text{if } k < S + D \end{cases} \quad . \tag{9}$$

The parameter of interest in the analysis of the KY algorithm is therefore the quantity :

$$G = C - S - D \tag{10}$$

This integer valued random variable, called the balance of the algorithm, is of yet unknown distribution. From the developments that follow, will result that G has expectation γ given by (3). Also, as will be clear from forthcoming developments, S and D have geometrically decreasing distributions so that taking average values of (9) :

$$\overline{c}(k) = k + \gamma + o(1)$$

where the $o(1)$ is a fast decreasing function of k.

3 - The basic parameters and the analysis

To work out the analysis of program KY, we shall proceed in a top down manner. In 3.1 below, we determine some of the basic probabilities involved together with the extraction of the essential parameters ; in 3.2, we study some statistics on digital search trees related to this problem and finally conclude with the analysis.

3.1 - The basic parameters

We let $\mathcal{B}$ denote the set $\{0,1\}^{\omega}$ of infinite binary sequences, occasionally assimilated to real numbers over the interval $[0;1]$, with $>$ denoting the lexicographical ordering on $\mathcal{B}$.

It will prove convenient in the sequel to conceive of the Knuth-Yao algorithm as operating on infinite sequences $Y = \{Y_j\}_{j\geq 0}$, where $Y_j \in \mathcal{B}$, as a recognizer of those sequences Y such that

$$Y_0 > Y_1 > Y_2 > \ldots > Y_{n-1} \leq Y_n \qquad \text{with n odd.}$$

More generally, given a finite sequence of elements of $\mathcal{B}$: $X = (X_0,X_1,X_2,\ldots,X_{n-1})$, we let $\sigma(X)$ denote the total number of bit inspections necessary to determine by pairwise comparisons the order pattern of all pairs of consecutive elements in X. Starting with an example, let $X = (X_0,X_1,X_2,X_3)$ with

$X_0 = \underline{1}\ 0\ 1\ 0\ 1\ 1\ 0\ \ldots$

$X_1 = \underline{0}\ \underline{0}\ \underline{1}\ \underline{1}\ 0\ 1\ 1\ \ldots$

$X_2 = \underline{0}\ \underline{0}\ \underline{1}\ \underline{0}\ 1\ 1\ 0\ \ldots$

$X_3 = \underline{0}\ \underline{1}\ 1\ 1\ 0\ 0\ 1\ \ldots$;

The order pattern is $X_0 > X_1 > X_2 < X_3$, i-e :

$(X_0>X_1)$; $(X_1>X_2)$; $(X_2<X_3)$.

Determining the ordering between X_0 and X_1 necessitates inspection of 1 bit of X_0 and X_1, so that

$$\sigma(X_0,X_1) = 1+1 = 2$$

To determine the order pattern of (X_0,X_1,X_2), we see that the comparison between X_1 and X_2 necessitates inspection of 4 bits of each of X_1 and X_2 ; However, of the 4 bits of X_1, one has already been read in the course of the comparison of X_0 and X_1, and should not be counted twice in σ. Thus

$$\sigma(X_0,X_1,X_2) = 1 + 4 + 4 = 9$$

and similarly

$$\sigma(X_0,X_1,X_2,X_3) = 1 + 4 + 4 + 2 = 11$$

We can now define σ precisely. Let $\delta(U,V)$ for $U,V \in \mathcal{B}$ be the rank of the first bit in U, that differs from the corresponding bit in V (ranks are numbered starting from 1). We have the recurrence :

$$\sigma(X_0,X_1,X_2,..,X_n) = \sigma(X_0,X_1,X_2,...,X_{n-1}) + \delta(X_{n-1},X_n) + |\delta(X_{n-1},X_n) - \delta(X_{n-1},X_{n-2})|^+$$

where for $x \in R$:

$$|x|^+ = \text{if } x \geq 0 \text{ then } x \text{ else } 0$$

We shall see that the analysis of the KY algorithm reduces to the study of various statistics on σ and δ.

Given an infinite sequence $Y = \{Y_j\}_{j\geq 0}$ of elements of $\mathcal{B}$, we let $\nu(Y)$ denote the number $n \geq 1$ such that

$$Y_0 > Y_1 > Y_2 > \ldots > Y_{n-1} \leq Y_n .$$

(As already noted, $\nu(Y)$ exists with probability one). We say that Y is a <u>success</u> if $\nu(Y)$ is odd, a <u>failure</u> otherwise and use the notations

$$F = \{Y/\nu(Y) \text{ even}\} ; \qquad S = \{Y/\nu(Y) \text{ odd}\} ;$$

$$G_n = \{Y/\nu(Y) = n\} ; \qquad E_n = \{Y/Y_0 > Y_1 > Y_2 > \ldots > Y_{n-1}\}$$

in accordance with the notations of (6). The succession of Y-draws in the algorithms is formally represented by the series

$$S + FS + FFS + FFFS + \ldots \qquad (11)$$

where sum denotes disjoint union of events and product denotes succession. Since G_n decomposes as the difference of two events :

$$G_n = E_n \backslash E_{n+1},$$

we have

$$g_n = Pr(G_n) = \frac{1}{n!} - \frac{1}{(n+1)!} = \frac{n}{(n+1)!} \quad . \tag{12}$$

From the definitions :

$$S = \sum_{p\geq 0} G_{2p+1} \quad , \qquad F = \sum_{p\geq 1} G_{2p}$$

so that the corresponding probabilities are

$$s = Pr(S) = \sum_{p\geq 0} \frac{1}{(2p+1)!} - \frac{1}{(2p+2)!} = 1 - e^{-1} \quad ,$$

$$f = Pr(F) = \sum_{p\geq 1} \frac{1}{(2p)!} - \frac{1}{(2p+1)!} = e^{-1} \quad , \tag{13}$$

with which we can check that the probability of (11) is, as expected :

$$\frac{s}{I-f} = 1 \quad .$$

We first analyze the losses in equation (10). The number of bit inspections made by algorithm KY in determining $\nu(Y)$ is precisely

$$\ell(Y) = \sigma(Y_0, Y_1, \ldots, Y_{\nu(Y)}) \quad .$$

We let λ_n denote the conditional expectation :

$$\lambda_n = E(\ell(Y) \mid \nu(Y) = n) = E(\sigma(Y_0, Y_1, \ldots, Y_n) \mid G_n) \tag{14}$$

We have for the expectation of ℓ conditioned by a success

$$CS = E(\ell(Y) \mid Y \in S) = \frac{1}{s} \sum_{p \geq 0} \lambda_{2p+1}\, g_{2p+1} \qquad (15)$$

and upon conditioning by a failure

$$CF = E(\ell(Y) \mid Y \in F) = \frac{1}{f} \sum_{p \geq 1} \lambda_{2p}\, g_{2p} \qquad (16)$$

Now with the decomposition (9), we find that the total number of bit inspections made in drawing Y's until a success occurs is :

$$CT = s(CS) + fs(CF+CS) + f^2 s(2CF+CS) + f^3 s(3CF+CS) + \ldots$$

which simplifies to

$$CT = CS + \frac{1}{e-1}\, CF \quad .$$

This last quantity is thus nothing but the expectation of the quantity C (total number of flip coins) in the balance equation (10). Using (12), (13), (14), we find that, introducing the generating functions :

$$\lambda(z) = \sum_{n \geq 0} \lambda_n \frac{z^n}{n!} \quad ; \quad \Lambda(z) = \sum_{n \geq 0} \lambda_n \frac{z^{n+1}}{(n+1)!} \quad , \qquad (17)$$

we have :

$$CT = \frac{1}{1-e^{-1}}\, (\lambda(1) - \Lambda(1)). \qquad (18)$$

We now turn to the analysis of gains in equation (10). The expected value of the D-parameter in that equation is, again from decomposition (11) :

$$1s + 2fs + 3f^2 s + \ldots = \frac{s}{(1-f)^2} \qquad (19)$$

which evaluates to $(1-e^{-1})^{-1}$.

Let b(Y) be the number of bit inspections of Y_0 made by algorithm KY in determining $\nu(Y)$. We proceed for b as we did for ℓ. We have $b(Y) = \delta(Y_0,Y_1)$. Let β_n be the conditional expectation

$$\beta_n = E(b(Y) \mid \nu(Y)=n) = E(\delta(Y_0,Y_1) \mid G_n) \qquad (20)$$

the expectation of b conditionned by a success is :

$$BS = E(b(Y) \mid Y\in S) = \frac{1}{s} \sum_{p\geq 0} \beta_{2p+1}\, g_{2p+1} \quad ;$$

(which is also the expectation of the number of bits of Y_0 inspected for the last sequence Y examined : the algorithm terminates with a success).

Introducing again the generating functions

$$\beta(z) = \sum_{n\geq 1} \beta_n \frac{z^n}{n!} \quad ; \qquad B(z) = \sum_{n\geq 1} \beta_n \frac{z^{n+1}}{(n+1)!} \qquad (21)$$

and using (12), we find :

$$BS = \frac{1}{2s} (\beta(1) - \beta(-1) - B(1) - B(-1)) \; . \qquad (22)$$

Putting together equations (18), (19), (22) we find

Proposition 1 :

The expectation γ *of the balance parameter* G *of the Knuth-Yao algorithm satisfies the equation*

$$\gamma = \frac{1}{1-e^{-1}} [-1 + \lambda(1) - \Lambda(-1) - \frac{1}{2} (\beta(1) - \beta(-1) - B(1) - B(-1))] \; ,$$

with λ, Λ, β, B *the generating functions associated to the* ℓ *and* b *parameters of equations (14), (17) and (20), (21).*

3.2 - Some related statistics on digital search trees

The first step in the evaluation of the quantities λ_n and β_n of the previous section is to get rid of part of the conditioning represented by G_n. To that purpose, as companions to the already defined quantities :

$$\lambda_n = E(\sigma(Y_0,Y_1,\ldots,Y_{n-1},Y_n) \mid G_n)$$

$$\beta_n = E(\delta(Y_0,Y_1) \mid G_n)$$

we introduce

$$\mu_n = E(\sigma(Y_0,Y_1,\ldots,Y_{n-1},Y_n) \mid E_n) \qquad (23)$$

$$\gamma_n = E(\delta(Y_0,Y_1) \mid E_n) \qquad (24)$$

and the related parameter

$$\ell_n = E(\sigma(Y_0,Y_1,\ldots,Y_{n-1}) \mid E_n) \; . \qquad (25)$$

Initial values are

$$\lambda_1 = 2 \; , \quad \lambda_2 = 6 \quad ; \quad \beta_1 = 2 \; , \quad \beta_2 = \frac{11}{6} \quad ; \quad \mu_1 = 4 \; , \quad \mu_2 = \frac{20}{3} \quad ;$$

$$\gamma_1 = 0 \; , \quad \gamma_2 = 2 \quad ; \quad \ell_1 = 0 \; , \quad \ell_2 = 4 \; ,$$

and a general relation is given by the following proposition :

Proposition 2 :

The parameters β, λ *are related to the parameters* γ, μ, ℓ *by the equations, valid for* $n \geq 1$:

$$\beta_n = (1+\frac{1}{n})\,\gamma_n - \frac{1}{n}\,\gamma_{n+1} + 4\delta_{n,1} \qquad (26)$$

$$\lambda_n = (1+\frac{1}{n})\,\mu_n - \frac{1}{n}\,\ell_{n+1} \qquad (27)$$

PROOF :

For $Y_0, Y_1, \ldots$ in $\mathcal{B}$, we have

$$\{(Y_0, Y_1, \ldots, Y_n) \mid E_{n-1}\} = \{(Y_0, Y_1, \ldots, Y_n) \mid E_n\} \cup \{(Y_0, Y_1, \ldots, Y_{n-1}) \mid E_{n-1} \setminus E_n\} \qquad (28)$$

the respective probabilities of events of the r.h.s. given E_{n-1} being $\frac{1}{n+1}$ and $\frac{n}{n+1}$. Thus decomposing conditional expectations of σ according to (28), we get the relation

$$\mu_n = \frac{1}{n+1} \ell_n + \frac{n}{n+1} \lambda_n,$$

an equality equivalent to (26). The same reasoning applies modulo the special case n=1 to parameter γ giving equation (27).

□

The next stage in the evaluation of the quantities appearing in Proposition 2, is to provide inductive definitions for the parameters of which they represent average values. These definitions translate into recurrences over the average values which in turn correspond to functional (difference) equations for exponential generating functions that can be solved explicitly.

Let $\omega \subset \mathcal{B}$ be a finite set of elements of $\mathcal{B}$; let $\omega/0$ and $\omega/1$ denote the sets

$$\omega/0 = \{\alpha \mid 0\alpha \in \omega\}$$

$$\omega/1 = \{\alpha \mid 1\alpha \in \omega\},$$

so that :

$$\omega = 0(\omega/0) + 1(\omega/1).$$

With the example of Section 3.1, $\omega = \{X_0, X_1, X_2, X_3\}$ we have for instance :

$$\omega/0 = \{0\ 1\ 1\ 0\ 1\ 1\ \ldots\ ;\ 0\ 1\ 0\ 1\ 1\ 0\ \ldots\ ;\ 1\ 1\ 1\ 0\ 0\ 1\ \ldots\}$$

$$\omega/1 = \{0\ 1\ 0\ 1\ 1\ 0\ \ldots\} .$$

We can extend the definitions of parameters ℓ, β... to sets as follows : let $\omega = \{Y_0, Y_1, Y_2, \ldots, Y_{n-1}\}$ where the elements of ω are indexed in such a way that $Y_0 > Y_1 > Y_2 > \ldots > Y_{n-1}$. We define

$$\ell(\omega) = \sigma(Y_0, Y_1, Y_2, \ldots, Y_{n-1}) ;$$

$$\mu(\omega;Z) = \sigma(Y_0, Y_1, Y_2, \ldots, Y_{n-1}, Z) ;$$

$$\gamma(\omega) = \delta(Y_0, Y_1), \qquad (29)$$

and we have :

Proposition 3 :

The parameters γ, ℓ, μ *defined in (28) admit the inductive definitions, valid for* $|\omega| \geq 2$ *:*

$$\gamma(\omega) = \begin{cases} 1 + \gamma(\omega/1) & \text{if } |\omega/1| \geq 1 \\ 1 + \gamma(\omega/0) & \text{if } |\omega/1| = 0 ; \end{cases}$$

$$\ell(\omega) = \ell(\omega/0) + \ell(\omega/1) + |\omega| ;$$

$$\mu(\omega,Z) = \begin{cases} \ell(\omega/1) + \ell(\omega/0) + |\omega| + 1 & \text{if } |\omega/0| \neq 0,\ |\omega/1| \neq 0,\ Z[1] = 0 ; \\ \ell(\omega/1) + \mu(\omega/0;Z) + |\omega| + 1 & \text{if } |\omega/0| \neq 0,\ |\omega/1| \neq 0,\ Z[1] = 1 ; \\ \mu(\omega/1) + |\omega| + 1 & \text{if } |\omega/0| = 0\ ;\ Z[1] = 0 ; \\ \ell(\omega/1) + |\omega| + 1 & \text{if } |\omega/0| = 0\ ;\ Z[1] = 1 ; \\ \ell(\omega/0) + |\omega| + 1 & \text{if } |\omega/1| = 0\ ;\ Z[1] = 0 ; \\ \mu(\omega/0) + |\omega| + 1 & \text{if } |\omega/1| = 0\ ;\ Z[1] = 1. \end{cases}$$

with the initial conditions :

$$\ell(\omega) = \gamma(\omega) = 0 \qquad \text{if } |\omega| \leq 1$$

$$\mu(\omega,Z) = 0 \qquad \text{if } |\omega| = 0 \ .$$

PROOF

(i) The recurrence of γ : if $|\omega/1| \geq 2$ then both Y_0 and Y_1 start with a 1 ; obviously

$$\delta(Y_0,Y_1) = 1 + \delta(Y_0/1,Y_1/1),$$

so that

$$\gamma(\omega) = 1 + \gamma(\omega/1).$$

If $|\omega/1| = 1$ then Y_0 starts with a 1, Y_1,Y_2 start with a 0 and

$$\delta(Y_0,Y_1) = 1$$

so that

$$\gamma(\omega) = 1 = 1 + \beta(\omega/1)$$

using the initial conditions.

(ii) The recurrence of ℓ : if both $\omega/0$ and $\omega/1$ are each of cardinality ≥ 2, then by inspection

$$\begin{aligned}\ell(\omega) &= \ell(0.\omega/0) + \ell(1.\omega/1) \\ &= \ell(\omega/0) + |\omega/0| + \ell(\omega/1) + |\omega/1| \\ &= \ell(\omega/0) + \ell(\omega/1) + |\omega|.\end{aligned}$$

other cases follow by similar arguments using the initial conditions.

(iii) The recurrence of μ : we shall only discuss the first two cases. When $|\omega/1|$ and $|\omega/0|$ are both non empty. If $Z[1] = 0$, then one only need examine this first bit of Z to determine that

$$Z > \min(\omega),$$

so that

$$\mu(\omega) = \ell(\omega) + 1.$$

If $z[1] = 1$, we need to compare $Z/0$ to $\omega/0$ to determine the relation between Z and $\min(\omega) = 0.\min(\omega/0)$, so that

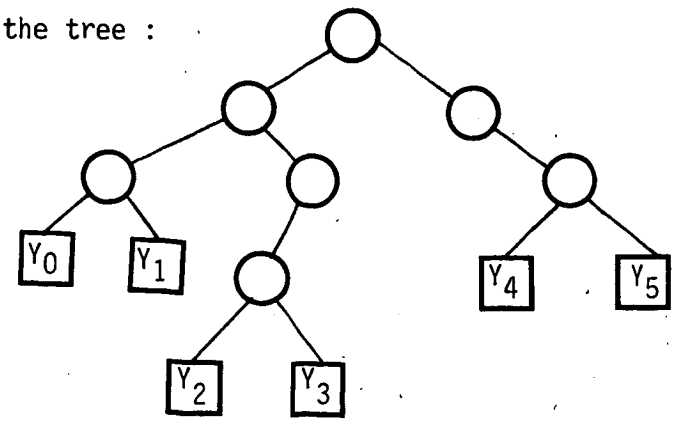
$$\begin{aligned}\mu(\omega;Z) &= \ell(1.\omega/1) + \mu(0.\omega/0;Z) \\ &= \ell(\omega/1) + |\omega/1| + \mu(\omega/0;Z/0) + 1 + |\omega/0| \ .\end{aligned}$$

Other cases follow by similar arguments. □

These parameters have natural interpretations in terms of binary trees associated to subsets of $\mathcal{B}$. Given ω, if $|\omega| = 1$, we represent it as single leaf labelled with ω. If $|\omega| \geq 2$, we obtain its representation by appending to a common root, the representations of $\omega/0$ and $\omega/1$ constructed recursively in the same way. Such trees are called digital search trees or tries [Kn 73] and occur in very diverse contexts in computer algorithms [FS 82]. For instance to

$$\omega = \{000..., 001..., 0100..., 0101..., 110..., 11...\}$$

is associated the tree :



With this representation, $\ell(\omega)$ is the path length of the tree associated to ω, and $\gamma(\omega)$ is the length of its rightmost branch.

Introducing the exponential generating functions of quantities ℓ_n, δ_n, μ_n,

$$\ell(z) = \sum_{n\geq 0} \ell_n \frac{z^n}{n!} \quad ; \quad \gamma(z) = \sum_{n\geq 0} \delta_n \frac{z^n}{n!} \quad ; \quad \mu(z) = \sum_{n\geq 0} \mu_n \frac{z^n}{n!} ,$$

we have :

Proposition 4 :

The generating functions relative to parameters γ, ℓ, μ *satisfy the difference equations :*

$$\gamma(z) = (1+e^{z/2})\ \gamma(\tfrac{z}{2}) + e^z - 1 - z$$

$$\ell(z) = 2e^{z/2}\ \ell(\tfrac{z}{2}) + z(e^z-1)$$

$$\mu(z) - \ell(z) = \tfrac{1}{2}(1+e^{z/2})\ (\mu(\tfrac{z}{2})-\ell(\tfrac{z}{2})) + e^z + z - 1 .$$

PROOF :

The starting point is, for $\omega \subset \mathcal{B}$ of cardinality n, the expression of the Bernoulli probabilities :

$$Pr(|\omega/0| = k) = \frac{1}{2^n}\binom{n}{k} . \qquad (30)$$

Let u_n, v_n, w_n be the expectations of u, v, w on subsets of $\mathcal{B}$ of cardinality n, with corresponding exponential generating functions

$$u(z) = \sum_{n\geq 0} u_n \frac{z^n}{n!} \quad ; \quad v(z) = \sum_{n\geq 0} v_n \frac{z^n}{n!} \quad ; \quad w(z) = \sum_{n\geq 0} w_n \frac{z^n}{n!} .$$

If

$$u(\omega) = v(\omega) + w(\omega) \qquad (31)$$

then, taking expectations

$$u_n = v_n + w_n$$

and

$$u(z) = v(z) + w(z) . \qquad (32)$$

If

$$u(\omega) = v(\omega/0) \cdot w(\omega/1), \tag{33}$$

then by (29) :

$$u_n = \frac{1}{2^n} \sum_{0 \le k \le n} \binom{n}{k} v_k \, w_{n-k}$$

so that

$$u(z) = v(\tfrac{z}{2}) \, w(\tfrac{z}{2}) \ . \tag{34}$$

Lastly :

$$u(\omega) \equiv 1 \quad \Longrightarrow \quad u(z) = e^z \tag{35}$$

$$u(\omega) = |\omega| \quad \Longrightarrow \quad u(z) = ze^z \ . \tag{36}$$

The schemes (30) - (35) make it possible to translate directly inductive definitions of valuations obtained by sums and products form elementary valuations. For instance the definition of $\ell(\omega)$ in Proposition 3, can be put under the form† :

$$\ell(\omega) = \ell(\omega/0) \ V(\omega/1) + \ell(\omega/1) \ V(\omega/0) + |\omega| - \delta_{|\omega|,0} - \delta_{|\omega|,1}$$

with $V(\omega)$ the constant valuation 1. This form translates directly into :

$$\ell(z) = e^{z/2} \, \ell(z/2) + e^{z/2} \, \ell(z/2) + ze^z - z$$

$$= 2e^{z/2} \, \ell(z/2) + z(e^z - 1)$$

with the conditions

$$\ell(0) = \ell'(0) = 0 \ .$$

---

† Here $\delta_{i,j}$ denotes the Kronecker delta symbol : $\delta_{i,j} = 1$ if $i=j$, $\delta_{i,j} = 0$ otherwise.

Similarly, writing :

$$\gamma(\omega) = \gamma(\omega/1) + \gamma(\omega/0)\ \delta_{|\omega/1|,0} + 1 - \delta_{|\omega|,0} - \delta_{|\omega|,1}$$

with $\gamma(\emptyset) = 0$, $\gamma(\{a\}) = 0$, we find :

$$\gamma(z) = \gamma(\tfrac{z}{2})\ e^{z/2} + \gamma(\tfrac{z}{2}) + e^{z} - 1 - z$$

$$= (1+e^{z/2})\ \gamma(\tfrac{z}{2}) + e^{z} - 1 - z$$

with the initial conditions

$$\gamma(0) = \gamma'(0) = 0 \quad .$$

We shall not spell out the details for parameter μ which also follows by similar arguments.

□

The functional equations of Proposition 4 are all of the form

$$\phi(z) = a(z)\ \phi(\tfrac{z}{2}) + b(z), \tag{37}$$

(with ϕ the unknown function), which can be solved formally by iteration, giving :

$$\phi(z) = \sum_{k\geq 0} b(\frac{z}{2^k}) \prod_{0\leq j<k} a(\frac{z}{2^j}) \quad . \tag{38}$$

In the case when

$$a(z) = ce^{z/2}, \tag{39}$$

we find :

$$\phi(z) = \sum_{k\geq 0} c^k e^{z(1-2^{-k})}\ a(\frac{z}{2^k}), \tag{40}$$

and the equation for $\ell(z)$ is of this type.

A more interesting case is

$$a(z) = c(1+e^{z/2}) \; ; \tag{41}$$

let

$$A(z) = c^k \prod_{0\le j<k} a(\frac{z}{2^j}) \; .$$

Setting $q = \exp(z2^{-k})$, we have :

$$A(z) = c^k(1+q)\ (1+q^2)\ (1+q^4)\ \ldots\ (1+q^{2^{k-1}}),$$

and, calculating $(1-q)\ A(z)$, the products collapse, so that $A(z)$ simplifies to :

$$A(z) = c^k \frac{1-q^{2^k}}{1-q} = c^k \frac{1-e^z}{1-e^{z2^{-k}}} \; . \tag{42}$$

Applying these schemes to the equations satisfied by γ, ℓ, μ, we have thus established :

Proposition 5 :

The generating functions relative to parameters γ, ℓ, μ admit the explicit expressions :

$$\gamma(z) = \sum_{k\ge 0} \frac{1-e^z}{1-e^{z2^{-k}}}\ (e^{z2^{-k}} - 1 - z2^{-k}) \; ;$$

$$\ell(z) = z \sum_{k\ge 0} e^z - e^{z(1-2^{-k})} \; ;$$

$$\mu(z) - \ell(z) = \sum_{k\ge 0} 2^{-k} \frac{1-e^z}{1-e^{z2^{-k}}}\ (e^{z2^{-k}} + z2^{-k} - 1) \; . \tag{43}$$

These expansions are easily checked to be convergent for all values of z.

From Proposition 5, it is possible to derive explicit expressions for the quantities γ_n, μ_n, ℓ_n (whence β_n, λ_n) by performing standard Taylor expansions. In fact the result for ℓ_n is known ([Kn 73 p. 131]). However all that is needed here is the expression of corresponding generating functions at $z = \pm 1$.

Instantiating the results of Proposition 5 at $z = \pm 1$ and using the relations of Proposition 2 leads to an explicit expression for the constant γ. We thus have :

Lemma 1 :

The expectation γ of the balance parameter of the Knuth-Yao algorithm is given by Equation (3).

To conclude the analysis, we need to estimate the effect of truncations when k bits of the exponential variate are drawn. Equation (9) shows that the cost of generating k bits satisfies :

$$c(k) = k + C - S - D + |S+D-k|^+ ,$$

so that taking expectations in (43) :

$$\bar{c}(k) = k + \gamma + E(|S+D-k|^+) \quad . \tag{44}$$

We prove :

Lemma 2 :

The trucation term

$$u_k = E(|S+D-k|^+)$$

in Equation (43) satisfies :

$$u_k = O(2^{-k}) \ .$$

PROOF :

We let P_m denote the probability that S + D = m in the KY algorithm. From the independence of S and D follows that

$$P_m = \sum_j Pr(S=j)\ Pr(D=m-j)\ . \qquad (45)$$

The distribution of D is known by (8) :

$$Pr(D=\alpha) = (e-1)\ e^{-\alpha}\ .$$

We thus need to estimate the distribution of S.

(i) In the notations of Section 3.2, define :

$$p_{n,\ell} = Pr(\delta(Y_0,Y_1) = \ell \mid E_n)$$

$$q_{n,\ell} = Pr(\delta(Y_0,Y_1) = \ell \mid G_n)\ .$$

Decomposing G_n as $E_n \backslash E_{n+1}$, we obtain a relation analogous to Proposition 2 :

$$nq_{n,\ell} = (n+1)p_{n,\ell} - p_{n+1,\ell}\ . \qquad (46)$$

The problem is thus reduced to the analysis of the distribution of the lefmost branch in a digital search tree built on n binary sequences (the $p_{n,\ell}$). Let ξ_ℓ ($\ell \geq 0$) be the characteristic variable

$\xi_\ell(Y_0,Y_1,\ldots,Y_{n-1}) = 1$ if the leftmost branch of the tree built on $Y_0,Y_1\ldots$ has length (number of edges) exactly ℓ, and 0 otherwise.

Then

$$p_{n,\ell} = E(\xi_\ell)$$

From the inductive definition of ξ_ℓ with $\ell \geq 2$,

$$\xi_\ell(\omega) = \xi_{\ell-1}(\omega/0) + \delta(\omega/0)\ \xi_{\ell-1}(\omega/1)$$

there follows by the techniques of Proposition 4 the recurrence :

$$p_\ell(z) = (1+e^{z/2})\ p_{\ell-1}(z) \qquad (\ell \geq 2) \tag{47}$$

Where we have set

$$p_\ell(z) = \sum_{n\geq 0} p_{n,\ell}\ \frac{z^n}{n!} \quad .$$

Initial values of $p_\ell(z)$ are :

$$p_0(z) = 1 + z\ ; \qquad p_1(z) = \frac{z}{2}\ (e^{z/2}-1)$$

and from the recurrence (47), we get ;

$$p_\ell(z) = \frac{z}{2^\ell}\ \frac{e^z-1}{e^{z/2^k}-1)} \quad . \tag{48}$$

(ii) Returning to the evaluation of the distribution of S, we have :

$$Pr(S=\ell) = \frac{1}{1-e^{-1}} \sum_{p\geq 0} g_{2p+1}\ q_{2p+1,\ell} \quad ,$$

which with relation (46) and the expression of g in (12) becomes

$$Pr(S=\ell) = \frac{1}{1-e^{-1}} \sum_{p\geq 0} \frac{p_{2p+1,\ell}}{(2p+1)!} - \sum_{p\geq 0} \frac{p_{2p+1,\ell}}{(2p+1)!}$$

$$= \frac{1}{1-e^{-1}}\ p_\ell(-1) \quad .$$

Thus using (48), for $\ell \geq 1$, we get :

$$Pr(S=\ell) = \frac{1}{2^\ell}\ \frac{1}{1+e^{-1/2^\ell}} \tag{49}$$

(iii) We can now combine the two bounds :

$$Pr(D=\alpha) < c_1 e^{-\alpha} \qquad Pr(S=\ell) < c_2 2^{-\ell}$$

to majorize P_m given by Equation (45). We have :

$$P_m < [\text{coefficient of } x^m \text{ in}] \frac{c_1}{1-xe^{-1}} \frac{c_2}{1-x2^{-1}}$$

$$P_m < c_3 2^{-m} .$$

This permits to conclude on u_k :

$$u_k = E(|S+D-k|^+)$$

$$= \sum_{m \geq k} (m-k) P_m = \sum_{j \geq 1} j P_{k+j}$$

$$< c_3 2^{-k} \sum_{j \geq 1} j 2^{-j}$$

$$< c_4 2^{-k} .$$

This completes the proof of Lemma 2. □

Combining Lemma 1 and Lemma 2, we have thus proved :

Theorem 1 :

The expected cost of generating k *bits of an exponentially distributed variate using the Knuth-Yao algorithm satisfies*

$$\overline{c}(k) = k + \gamma + O(2^{-k})$$

where

$$\gamma = e(2 + \frac{3}{e-1}) + \sum_{k \geq 0} \frac{1}{e^{1/2^k} - 1} \left[\frac{e}{4^k} + 1 - e^{1/2^k} (1 - \frac{1}{2^k}) \right] .$$

We notice, in passing, that the relation

$$\sum_{\ell \geq 0} p_\ell(z) = e^z$$

which expresses that the probabilities $p_{n,\ell}$ of Lemma 2 summed over ℓ add up to 1, leads modulo the change of variable $q = e^z$ to the rather curious identity :

$$\frac{1}{\log q} - \frac{1}{q-1} = \sum_{\ell \geq 1} \frac{1}{2^\ell} \frac{1}{1+q^{1/2^\ell}} \quad .$$

4 - Distribution estimates

In this section, we briefly consider the problem of determining the probability distribution of the balance factor of the KY algorithm given by :

$$G = C - S - D$$

These probabilities could theoretically by determined by considering all the possible execution paths of the algorithm. However, the combinatorial process involved quickly grows out of control, so that here again an approach based on generating functions is used.

If the random variable D takes the particular value $\delta + 1$, G can be rewritten as :

$$G = (C_1 - 1) + (C_2 - 1) + \ldots + (C_\delta - 1) + \hat{C} - 1 \qquad (50)$$

where C_j is the number of bit inspections made in the j-th (unsuccessful) trial of a Y-sequence, and $\hat{C} = C_{\delta+1} - S$ is the difference between the number of bit inspections in the last (successful) trial and the value of S. We now introduce a convenient notation for generating functions : for a random variable X, we define :

$$X(q) = \sum_k Pr(X=k)\, q^k = E(q^k) \qquad (51)$$

With this notation, the initial problem is thus to determine G(q). From (50), taking generating functions and summing over all possible values δ of D, we get :

$$G(q) = \sum_{\delta \geq 0} \left(\frac{C(q)}{q}\right)^{\delta} \frac{\hat{C}(q)}{q}$$

$$G(q) = \frac{\hat{C}(q)}{q - C(q)} \; . \qquad (52)$$

The problem is thus reduced to the determination of C(q) and $\hat{C}(q)$. This in turn reduces to the study of the probability distributions of parameters like ℓ, μ,... on $\mathcal{B}^n$.

We shall only briefly present the derivation of equations for the path length parameter ℓ. Let

$$\ell_{n,k} = Pr(\ell(\omega)=k)$$

with $\omega \in \mathcal{B}^n$. We introduce the corresponding generating functions :

$$\ell_n(q) = \sum_k \ell_{n,q}\, q^k = E(q^{\ell(\omega)}),$$

$$\ell(q,z) = \sum_n \ell_n(q) \frac{z^n}{n!} \; .$$

From the inductive definition of ℓ in Proposition 4 follows that

$$q^{\ell(\omega)} = q^{\ell(\omega/0)}\, q^{\ell(\omega/1)}\, q^{|\omega|} \qquad \text{when } |\omega| \geq 2 \; . \qquad (53)$$

Thus the generating function $\ell(q,z)$ satisfies the difference equation :

$$\ell(q,z) = [\ell(q,\tfrac{qz}{2})]^2 + z(1-q). \qquad (54)$$

This equation is equivalent to the recurrence

$$\ell_n(q) = \left(\frac{q}{2}\right)^n \sum_i \binom{n}{i} \ell_i(q)\, \ell_{n-i}(q) + \delta_{n,1}(1-q) \qquad (55)$$

whose probabilistic interpretation is obvious.

We can similarly express the inductive definitions of μ and μ-δ by equations like (53). These translate into functional equations of the form (54) or into corresponding recurrences like (55). We shall not spell out the computations here, and we only state the final result :

Theorem 2 :

The distribution of costs in the Knuth-Yao algorithm has a generating function given by :

$$G(q) = \frac{\nu(q,1) - \nu(q,-1) - \psi(q,1) - \psi(q-1) + 2}{2q - \mu(q,1) - \mu(q,-1) + \ell(q,1) - \ell(q,-1)}$$

where functions ℓ, μ, ν, ψ *satisfy the difference equations :*

$$\ell = \ell_2^{\,2} + z(1-q)$$

$$\mu = \frac{q}{2}(1+\ell_2)\,\mu_2 + \frac{q}{2}(\ell-\ell_2) + (1+z)(1-q)$$

$$q\psi = (1+\ell_2)\,\psi_2 - \ell_2 + q - 1$$

$$\nu = \nu_2 + \frac{1}{2}(\psi_2-1)(\mu_2+\ell_2)$$

in which for a bivariate generating function $\phi(q,z)$, *we have used the notations :* $\phi = \phi(q,z)$; $\phi_2 = \phi(q,\frac{qz}{2})$.

These functional equations (or rather the implied recurrences) make it possible to compute the probability distribution of G. Table 1 gives the first values of $\pi_n = Pr(G=n)$ that have been determined using the Macsyma system for symbolic calculations. It should also be feasible, in principle, to determine moments of higher order of G explicitly (leading to expressions analogous to that of γ). The numerical value of the standard deviation of G calculated from our numerical data appears to be close 7.13.

n	π_n	π'_n	π''_n
0	$1/2^2$	0.250000	0.24913
1	$1/2^3$	0.125000	0.12471
2	$3/2^5$	0.093750	0.09409
3	$4/2^6$	0.062500	0.06421
4	$17/2^8$	0.066406	0.06694
5	$25/2^9$	0.048828	0.04716
6	$95/2^{11}$	0.046386	0.04713
7	$141/2^{12}$	0.034423	0.03558
8	$591/2^{14}$	0.036071	0.03554
9	$857/2^{15}$	0.026153	0.02578
10	$3519/2^{17}$	0.026847	0.02740
11	$5541/2^{18}$	0.021137	0.02021
12	$21331/2^{20}$	0.020342	0.01979

Table 1 : For n = 0..12, display of the values of the exact probabilities $\pi_n = Pr(G=n)$ of the corresponding numerical values (π'_n) and of the empirical estimates resulting from 10^5 simulations (π''_n).

REFERENCES

[FS 82] P. Flajolet, D. Sotteau
"A recursive partitionning process of computer science" in II World Conference on Mathematics at the Service of Man, Las Palmas (1982), pp. 25-30.

[Kn 73] D.E. Knuth
"The art of computer programming"
Vol. 3 Sorting and Searching, Addison Wesley, Reading 1973.

[KY 76] D.E. Knuth, A.C. Yao
"The complexity of nonuniform random number generation"
in Algorithms and Complexity, Academic Press, New-York (1976).

[VN 51] J. Von Neumann
"Various techniques used in connection with random digits"
notes by G.E. Forsythe, National Bureau of Standards,
Applied Math Series, 12 (1951). Reprinted in Von Neumann's
Collected Works 5 (Pergamon Press, 1963), pp. 768-770.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique