Average Cost of Duval's Algorithm for Generating Lyndon Words

Jean BERSTEL
Michel POCCHIOLA

Laboratoire d'Informatique, URA 1327 du CNRS
Département de Mathématiques et d'Informatique
Ecole Normale Supérieure

LIENS - 92 - 8

March 1992

Average Cost of Duval's Algorithm for Generating Lyndon Words *

Jean Berstel [1] (berstel@litp.ibp.fr)
Michel Pocchiola [2] (pocchiol@dmi.ens.fr)

(1) Laboratoire d'Informatique Théorique et Pratique (LITP)
Institut Blaise Pascal
4, place Jussieu, 75252 Paris Cédex 05

(2) Laboratoire d'Informatique de l'Ecole normale supérieure (LIENS)
URA 1327, CNRS, 45 rue d'Ulm, 75230 Paris Cédex 05

March 30, 1992

Abstract

The average cost of Duval's algorithm for generating all Lyndon words up to a given length in lexicographic order is proved to be asymptotically equal to $(q+1)/(q-1)$, where q is the size of the underlying alphabet. In particular, the average cost is independent of the length of the words generated. A precise evaluation of the constants is also given.

1 Introduction

Several years ago, J.-P. Duval [4] has presented an amazingly simple algorithm for generating all Lyndon words up to a given length in lexicographic order. He observed that the worst-case behavior of his algorithm, for computing the next Lyndon word is linear, and he left as an open problem to determine the average-case running time. We answer this question by showing that the average number of operations required for computing a Lyndon word of length at most n is constant, and independent of n. More precisely, we show that the cost is asymptotically equal to $(q+1)/(q-1)$, where q is the size of the alphabet.

Given a totally ordered alphabet, a *Lyndon word* is a word that is smaller than all its conjugates, for the lexicographic ordering. Lyndon words were introduced by Lyndon [9] under the name "standard lexicographic sequences" in

*Partially supported by PRC "Mathématiques et Informatique"

order to give a base for the free Lie algebra over A (see Lothaire [8], Reutenauer [10]). One of the basic properties of the set of Lyndon words is that every word admits a uniquely factorizable as a non increasing product of Lyndon words. There is also a close relationship between Lyndon words and irreducible polynomials over a finite field (Golomb [6]).

There are several algorithms dealing with Lyndon words. Booth [1] shows how to compute, in linear time, the smallest among the conjugates of a given word. This is in fact an application of another algorithm by Duval [3] that computes, in linear time, the factorization of a word into Lyndon words. The algorithm for systematic generation of Lyndon words is similar, in structure, to algorithms for systematic generation of trees [11, 14] or of other combinatorial objects [12]. For these objects, known algorithms have constant average running time. We show that the same holds for Duval's algorithm : the average cost is given by

$$\frac{q+1}{q-1}\left(1+\frac{2q}{(q^2-1)n}+O\left(\frac{1}{n^2}\right)\right). \tag{1}$$

We even give an evaluation of the constant of the big-O in order to describe the behaviour of the average cost for all values of n.

The paper is organized as follows: the next section reviews Duval's algorithm and gives an expression for the cost. Then the asymptotic constant running time is proved. The last section contains the effective constants. We conclude by some remark about possible developments.

2 The algorithm

Let A be a totally ordered alphabet, and let $\prec$ denote the *lexicographical* ordering induced on the free monoid A^*. Recall that the *conjugacy class* of a word w is the set of all words uv such that $w = vu$. A *Lyndon word* is a word that is smaller than all other elements in its conjugacy class. For example, if $A = \{0, 1\}$ with $0 \prec 1$, then the 14 Lyndon words of length at most 5 in lexicographic ordering are :

```
0
0 0 0 0 1
0 0 0 1
0 0 0 1 1
0 0 1
0 0 1 0 1
0 0 1 1
0 0 1 1 1
0 1
0 1 0 1 1
0 1 1
```

```
0 1 1 1
0 1 1 1 1
1
```

Denote by a and z the minimal and the maximal letter in the alphabet A, and by $\nu(b)$ the letter following $b \neq z$ in the total ordering of A. If w is a word of the form $w = ubz^h$, with $b \neq z$, then we denote by $P(w)$ the word $u\nu(b)$.

Consider a fixed integer n. Duval's algorithm computes, from a given Lyndon word w, the next Lyndon word $N(w)$ of length at most n in two steps:

Algorithm.

Input : An integer n, and a Lyndon word $w \neq z$ of length at most n.

Step 1.- Compute the word $v = D(w) = w^h w'$, where $h \geq 1$ and w' is the proper prefix of w defined by $n = h|w| + |w'|$.

Step 2.- Compute the word $P(v)$.

Output : $P(D(w))$.

Duval proved that $N(w) = P(D(w))$. The implementation of the algorithm is straigthforward.

For the evaluation of the cost of the algorithm, we need some notation. We denote by $\mathcal{L}$ the set of Lyndon words, and by $\mathcal{L}_n$ the set of Lyndon words of length at most n. Also, let ℓ_n be the number of Lyndon words of length n, and let

$$L_n = \ell_1 + \cdots + \ell_n$$

be the number of Lyndon words of length at most n. Finally, we set

$$\Lambda_n = L_1 + \cdots + L_n.$$

Proposition 2.1 *The total cost C_n of Duval's algorithm for generating all Lyndon words of length at most n is*

$$C_n = 2\Lambda_n - L_n - 2n + 1$$

and the average cost γ_n is

$$\gamma_n = C_n/L_n \leq 2\Lambda_n/L_n - 1.$$

Proof. Let n be fixed. The cost of computing $D(u)$ for a word u is $n - |u|$. The resulting word $v = D(u)$ has length n. The cost for computing $u' = P(v)$ is $n - |u'| + 1$. Thus the cost for computing the next Lyndon word $u' = N(u)$ is $2n + 1 - (|u| + |u'|)$. Consequently, the total cost of Duval's algorithm for generating all Lyndon words of length at most n is

$$C_n = (2n+1)(L_n - 1) - \sum_{w \in \mathcal{L}_n - \{a\}} |w| - \sum_{w \in \mathcal{L}_n - \{z\}} |w|.$$

Since,

$$\sum_{w \in \mathcal{L}_n} |w| = \sum_{h=1}^{n} h\ell_h = nL_n - \Lambda_n,$$

the expressions follow. ∎

3 Average cost of Duval's algorithm

Recall that every word over the alphabet A admits a unique non increasing factorization into Lyndon words:

$$A^* = \prod_{u \in \mathcal{L}}{}' u^*$$

where the prime means that the product is decreasing. If A has q elements, then taking generation functions, one gets:

$$\frac{1}{1-qz} = \prod_{u \in \mathcal{L}} \frac{1}{1-z^{|u|}} = \prod_{n \geq 1} \left(\frac{1}{1-z^n} \right)^{\ell_n}$$

Setting

$$\ell(z) = \sum_{n \geq 1} \ell_n z^n$$

one gets

$$\log \frac{1}{1-qz} = \sum_{k \geq 1} \frac{1}{k} \ell(z^k)$$

whence, by Möbius inversion:

$$\ell(z) = \sum_{k=1}^{\infty} \frac{\mu(k)}{k} \log(\frac{1}{1-qz^k}). \tag{2}$$

Proposition 3.1 *The average cost γ_n of Duval's algorithm, for an alphabet with q letters, is given by*

$$\gamma_n = \frac{q+1}{q-1} \left(1 + \frac{2q}{(q^2-1)n} + O\left(\frac{1}{n^2} \right) \right). \tag{3}$$

Proof. For the proof, we use the transfer technique for the asymptotics of generating functions to the asymptotics of their coefficients, as developed in [5].

Let $\ell(z)$, $L(z)$ and $\Lambda(z)$ be the generating functions of the integers ℓ_n, L_n and Λ_n. The generating series $\ell(z)$ of Lyndon's words given above is analytic in the complex plane, excepted on the half-line of the reals $x \geq 1/q$; moreover

$$\ell(z) - \log \frac{1}{1-qz} \tag{4}$$

is analytic in the complex plane excepted on the half-line of the reals $x \geq 1/\sqrt{q}$.

We consider the hierarchy of functions

$$f_k(z) = (qz-1)^k \log \frac{1}{1-qz} \qquad (k \geq 0). \tag{5}$$

With the Taylor series expansion of $1/(1-z)$ and $1/(1-z)^2$ in the neighborhood of $r = 1/q$

$$\frac{1-r}{1-z} = \sum_{i=0}^{k} \frac{(z-r)^i}{(1-r)^i} + \frac{1}{(1-r)^k} \frac{(z-r)^{k+1}}{1-z} \tag{6}$$

$$\frac{(1-r)^2}{(1-z)^2} = \sum_{i=0}^{k} (i+1) \frac{(z-r)^i}{(1-r)^i} + \frac{1}{(1-r)^k} (z-r)^{k+1} \left\{ \frac{k+1}{1-z} + \frac{1}{(1-z)^2} \right\} \tag{7}$$

we obtain the following asymptotic developments as z goes to $1/q$

$$(1 - \frac{1}{q})\, L(z) = a_0 f_0(z) + a_1 f_1(z) + \cdots + a_k f_k(z) + O(f_{k+1}(z)) \tag{8}$$

$$(1 - \frac{1}{q})^2\, \Lambda(z) = a_0 f_0(z) + 2a_1 f_1(z) + \cdots + (k+1) a_k f_k(z) + O(f_{k+1}(z)) \tag{9}$$

with

$$a_i = \frac{1}{(q-1)^i}.$$

Now, for $n \geq k+1$,

$$[z^n] f_k(z) = \frac{q^n}{n \binom{n-1}{k}}.$$

Thus we can apply the transfer theorem of [5] and we obtain the following expressions

$$(1 - \frac{1}{q})\, L_n = \frac{q^n}{n} \left\{ \sum_{i=0}^{i=k} \frac{1}{(q-1)^i} \frac{1}{\binom{n-1}{i}} + O\left(\frac{\log n}{n^{k+1}} \right) \right\}$$

$$(1 - \frac{1}{q})^2\, \Lambda_n = \frac{q^n}{n} \left\{ \sum_{i=0}^{i=k} \frac{1}{(q-1)^i} \frac{i+1}{\binom{n-1}{i}} + O\left(\frac{\log n}{n^{k+1}} \right) \right\}.$$

The proposition follows. ∎

4 Evaluation of the constant

In this section, we evaluate the constant of the big-O which figures in proposition 3.1. The transfer theorem, though effective, does not give this constant explicitely. Our evaluation is obtained by elementary majoration techniques. The result is the following

Proposition 4.1 *The average cost γ_n of Duval's algorithm satisfies, for a q-letter alphabet and for all $n \geq 11$, the inequality*

$$\gamma_n \leq \frac{q+1}{q-1}\left(1 + \frac{2q}{(q^2-1)(n-1)} + \frac{61q}{(q^2-1)(q-1)(n-1)^2}\right).$$

For the clarity of exposition, we decompose the proof into several lemmas. The first lemma allows us to replace ℓ_n by q^n/n in developments of L_n and Λ_n. The two next lemmas give an upper bound for Λ_n and a lower bound for L_n.

Lemma 4.1 *For all $n \geq 1$, we have*

$$\frac{q^n}{n}\left(1 - \frac{q}{(q-1)q^{n/2}}\right) \leq \ell_n \leq \frac{q^n}{n}.$$

Proof. See exercise 3.27 page 142 of [7]. ∎

Recall that the functions $f_k(z)$ are defined by

$$f_k(z) = (qz-1)^k \log \frac{1}{1-qz}$$

We introduce an operator Φ by setting

$$\Phi f(z) = \frac{f(z)}{1-z}.$$

Next, we consider the developments (6) and (7), to the order 3 and 2 respectively, and we multiply them by $f_0(z)$. This gives

$$(1-\frac{1}{q})\,\Phi f_0(z) = f_0(z) + \frac{1}{q-1}f_1(z) + \frac{1}{(q-1)^2}f_2(z) + \frac{1}{q(q-1)^2}\Phi f_3(z) \quad (10)$$

$$(1-\frac{1}{q})^2\,\Phi^2 f_0(z) = f_0(z) + \frac{2}{q-1}f_1(z) + \frac{1}{q(q-1)}\left\{2\Phi f_2(z) + \Phi^2 f_2(z)\right\} \quad (11)$$

Lemma 4.2 *For all $n \geq 11$ and $q \geq 2$ one has*

$$(1-\frac{1}{q})\,L_n \geq \frac{q^n}{n}(1 + \frac{1}{(q-1)(n-1)}).$$

Proof. The lemma is readily verified by numerical computation in the domain

$$D = \{\,(q,n) \mid 2 \leq q \leq 5, 11 \leq n \leq 25\,\}.$$

In order to prove it for the other values of q and n, we first show that the coefficient of z^n in $\Phi f_3(z)$ is positive. Indeed, one has

$$f_3(z) = -qz + \frac{5}{2}q^2z^2 - \frac{11}{6}q^3z^3 + \sum_{n=4}^{\infty}\frac{6q^n}{(n-3)(n-2)(n-1)n}z^n \quad (12)$$

Since the coefficient u_n of z^n in $f_3(z)$ is positive for $n \geq 4$, it suffices to observe that

$$\begin{aligned} & u_1 + u_2 + u_3 + u_4 + u_5 + u_6 \\ = \;& q\,(-1 + \frac{5}{2}q - \frac{11}{6}q^2 + \frac{1}{4}q^3 + \frac{1}{20}q^4 + \frac{1}{60}q^5) \\ = \;& 1/60\, q(q-2)(q^4 + 5q^3 + 25q^2 - 60q + 30) \end{aligned}$$

is positive, and this is straightforward.

Using Lemma 4.1 we get that

$$L_n \geq \sum_1^n \frac{q^k}{k} - \frac{q}{q-1} \sum_1^n \frac{q^{k/2}}{k}. \tag{13}$$

The first sum of (13) can be bounded from below, in view of (10), by

$$\frac{q}{q-1}\frac{q^n}{n}\left\{1 + \frac{1}{(q-1)(n-1)} + \frac{2}{(q-1)^2(n-1)(n-2)}\right\}.$$

We show that on the complement of the domain D,

$$\frac{q^n}{n}\frac{2}{(q-1)^2(n-1)(n-2)} \geq \sum_1^n \frac{q^{k/2}}{k}.$$

For this, we bound each term in the right-hand side by $\frac{q^{n/2}}{n}$, and thus the whole right-hand side by $q^{n/2}$. Consequently, it suffices to prove that

$$\frac{2}{(q-1)^2 n(n-1)(n-2)} \geq \frac{1}{q^{n/2}}$$

Since the expression

$$d(q,n) = \frac{(q-1)^2 n(n-1)(n-2)}{q^{n/2}} - 2$$

is decreasing in n and in q for all $q \geq 2$ and $n \geq 11$, it suffices to observe that $d(6,11)$ and $d(2,26)$ are negative to conclude the proof. ∎

In order to prove the proposition, we now introduce a $F(n,N,q)$ which will allow us to parametrize the constant of the big-O. For this, note that

$$f_2(z) = qz - \frac{3}{2}q^2 z^2 + \sum_{n=3}^{\infty} \frac{2q^n}{(n-2)(n-1)n} z^n \tag{14}$$

and set

$$u_n = [z^n] f_2(z) \quad v_n = [z^n]\Phi f_2(z) \quad w_n = [z^n]\Phi^2 f_2(z)$$

and define

$$a(N) = \frac{1}{1 - \dfrac{N+1}{(N-2)q}}.$$

Then by definition

$$F(n,N,q) = G(n,N,q) + H(n,N,q)$$

with

$$G(n,N,q) = 2\frac{n-1}{n-2}\frac{q-1}{q}a(N)(2+a(N))$$

$$H(n,N,q) = n(n-1)^2\frac{q-1}{q^{n+1}}\left\{(n-N+3)v_{N-1} + w_{N-1}\right\}.$$

Observe that G and H are decreasing in n (for H, this holds for $n \geq 8$ as one may verify by taking the logarithmic derivative). Next, letting first go n to infinity and then q to infinity, one sees that $F(n,N,q)$ is bounded from below by 6. The proposition we look for is a consequence of the more general statement:

Proposition 4.2 *The average cost γ_n of Duval's algorithm, for a q-letter alphabet and for all $n \geq N \geq 11$, satisfies the inequality*

$$\gamma_n \leq \frac{q+1}{q-1}\left(1 + \frac{2q}{(q^2-1)(n-1)} + \frac{2(F(n,N,q)-1)q}{(q^2-1)(q-1)(n-1)^2}\right).$$

Furthermore, $F(n,N,q)$ decreases in n and q.

Lemma 4.3 *For all $n \geq N \geq 6$ and all $q \geq 2$, one has*

$$(1-\frac{1}{q})^2\,\Lambda_n \leq \frac{q^n}{n}\left(1 + \frac{2}{(q-1)(n-1)} + \frac{F(n,N,q)}{(q-1)^2(n-1)^2}\right).$$

Proof. In view of equation (11) it suffices to prove that

$$[z^n]\left\{2\Phi f_2(z) + \Phi^2 f_2(z)\right\} \leq \frac{F(n,N,q)\,q^{n+1}}{n(n-1)^2(q-1)}.$$

We show first that for $n \geq N \geq 6$ and $q \geq 2$,

$$u_N + \cdots + u_n \leq a(N)u_n$$

and

$$v_N + \cdots + v_n \leq (n-N+1)v_{N-1} + a(N)^2 u_n$$

Indeed, the inequality

$$u_{n+1}/u_n = q(n-2)/(n+1) \geq q(N-2)/(N+1) = u_{N+1}/u_N > 1$$

implies, setting $b = u_{N+1}/u_N$, that

$$u_N + \cdots + u_n \leq u_n \left(1 + b + b^2 + \cdots + b^{n-N}\right) \leq \frac{1}{1-b} u_n = a(N) u_n .$$

This proves the first inequality. The second follows by observing that

$$v_N + \cdots + v_n - (n - N + 1) v_{N-1} = \sum_{k=N}^{n} (u_N + \cdots + u_k).$$

Combining these two inequalities, the lemma follows after some elementary algebraic manipulations. ∎

Proof of proposition 4.2. Set $r = 1/\big((q-1)(n-1)\big)$; by the two preceeding lemmas, and with $c = F(n, N, q)$, one gets

$$\begin{aligned}
(1 - \frac{1}{q}) \frac{\Lambda_n}{L_n} &\leq \frac{1 + 2r + cr^2}{1 + r} \\
&\leq (1 + 2r + cr^2)(1 - r + r^2) \\
&\leq 1 + r + (c-1)r^2 - (c-2)r^3 + cr^4 \\
&\leq 1 + r + (c-1)r^2
\end{aligned}$$

since $cr^4 \leq (c-2)r^3$ for $c \geq 6$. The inequality follows.

We have already proved that F is a decreasing function of n. To prove that F is decreasing in q, we show that this holds separately for the functions G and H. It is straightforward to see that G is decreasing in q. For H, one may proceed by proving that both that v_{N-1}/q^{n-1} and w_{N-1}/q^{n-1} are decreasing functions of q. The first expression can be written as

$$\frac{v_{N-1}}{q^{n-1}} = \frac{1}{q^{n-11}} \frac{v_{10}}{q^{10}} + \frac{u_{11} + \cdots + u_{N-1}}{q^{n-1}}.$$

In this expression, the second term is decreasing with q because each u_i is, up to a positive multiplicative constant, an ith power of q. The first term is decreasing because v_{10}/q^{10} is decreasing for integral values of q as may be verified (for instance by some symbolic manipulation system). One proceeds in a similar manner to prove that w_{N-1}/q^{n-1} decreases, using the fact that v_k/q^k is decreasing for $k \geq 11$. ∎

The following table, obtained with Maple [2], gives several values of the function $F(n, N, q)$ which allow to adjust the constant of our proposition as a function of q and n.

q	$F(11, 11, q)$	$F(20, 11, q)$	$F(\infty, 11, q)$	$F(\infty, \infty, q)$
2	31.17	16.32	15	8
3	16.92	9.63	9.12	7
4	12.94	8.31	7.87	6.66
10	8.61	6.91	6.55	6.22
∞	6.66	6.33	6	6

In particular, the value of $F(11, 11, 2)$ gives the proposition 4.1. We conclude by comparing the real value of the cost γ_n to the bound, denoted $\Gamma_{n,N}$, as given in the proposition 4.2, for some values of n and q.

q	γ_{11}	$\Gamma_{11,11}$	γ_{20}	$\Gamma_{20,15}$	γ_{100}	$\Gamma_{100,15}$
2	3.47	4.61	3.26	3.36	3.0417	3.0449
3	2.18	2.27	2.09	2.10		
4	1.77	1.79	1.716	1.719		
10	1.248	1.249	1.2354	1.2356		

This shows that our bound is rather good.

5 Conclusion

We have shown that the computation of the next Lyndon word in the set of Lyndon words up to some fixed length requires constant time. In the same paper [4], Duval has presented another algorithm that generates all Lyndon words of fixed length in lexicographic order. It is an easy consequence of our result that the average cost of this second algorithm is asymptotically bounded by $(q + 1)/q$. However, we were unable to give a sharp asymptotic estimation.

Another open problem is to prove a stronger claim, namely that Duval's algorithm has *amortized* constant worst-case running time, in the sense of Tarjan [13]. This would mean that the computation of an interval of Lyndon word costs a constant times the length of the interval plus the difference of some potential. Such a potential seems to be difficult to find, perhaps because the computational cost increases for the "last" words in a sequence.

References

[1] K. S. Booth. Lexicographically least circular substrings. *Inform. Proc. Letters*, 10:240–242, 1980.

[2] B.W. Char, K.O. Gettes, G.H. Gonnet, M.B. Monagan, and S.M. Watt. *Maple V Language Reference Manual.* Springer-Verlag, 1991.

[3] J.-P. Duval. Factorizing words over an ordered alphabet. *J. Algorithms*, 4:363–381, 1983.

[4] J.-P. Duval. Génération d'une section des classes de conjugaison et arbre des mots de Lyndon de longueur bornée. *Theoret. Comput. Sci.*, 60:255–283, 1988.

[5] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM J. Discr. Math.*, 3(2):216–240, may 1990.

[6] S. W. Golomb. Irreducible polynomials, synchronizing codes, primitive necklaces and the cyclotomic algebra. In *Proc. Conf. Combinatorial Math. and Its Appl.*, pages 358–370, Chapel Hill, 1969. Univ. of North Carolina Press.

[7] R. Lidl and H. Niederreiter. *Finite Fields.* Cambridge University Press, 1984.

[8] M. Lothaire. *Combinatorics on Words.* Addison-Wesley, 1983.

[9] R. C. Lyndon. On Burnside problem I. *Trans. American Math. Soc.*, 77:202–215, 1954.

[10] C. Reutenauer. *Free Lie Algebras.* In press, 1992.

[11] F. Ruskey and T.C. Hu. Generating binary trees lexicographically. *SIAM J. Comput.*, 6:745–758, 1977.

[12] D. Stanton and D. White. *Constructive Combinatorics.* Springer-Verlag, 1986.

[13] R. E. Tarjan. Amortized computational complexity. *SIAM J. Alg. Discr. Meth.*, 6:306–318, 1985.

[14] R. A. Wright, B. Richmond, A. Odlyzko, and B. D. McKay. Constant time generation of free trees. *SIAM J. Comput.*, 15:540–548, 1986.