Lyndon words, permutations and trees

Christophe Hohlweg and Christophe Reutenauer [a,b,*]

[a] *IRMA; 7 rue René Descartes; 67084; Strasbourg; France*

[b] *Université du Québec à Montréal; Département de mathématiques; case postale 8888, succursale Centre-ville; Montréal, Québec; H3C 3P8*

1 Introduction

A well-known combinatorial construction, that has many applications in Computer Science [6], [15], [8], [18], maps bijectively permutations in S_n onto binary, planary trees, with labels in $\{1, \ldots, n\}$, increasing from root to leaves; see [5], [3], [17]. See [16], p. 23-24, for a formal definition of this bijection ($\pi \mapsto T(\pi)$ in his notation). This construction is illustrated in Figure 1; the inverse mapping is simply the projection.

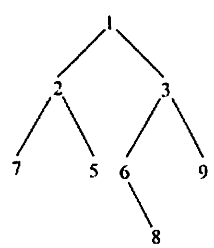

Figure 1

On the other hand, to each Lyndon word is associated a binary, planary complete tree with leaves labelled by the letters of the word; again, the inverse mapping is the projection. See Section 3 for a formal definition. Motivation for this construction comes from the theory of groups and Lie algebras; the tree encodes a nonassociative operation, either a commutator in the free group [1], or a Lie bracketting [9]; both constructions lead to bases of the free Lie

* This work was done during the time where the second author was professeur at the Université Louis Pasteur, Strasbourg.

Email address: `hohlweg@math.u-strasbg.fr`, `christo@math.uqam.ca` (Christophe Hohlweg and Christophe Reutenauer).

Preprint submitted to Elsevier Science *22 January 2003*

algebra consisting of the collection of all Lie polynomials defined by Lyndon words.

In this note, we show that this second construction reduces to the first: indeed, one can associate to each (Lyndon) word a permutation, that we call its *suffix standard permutation.* This permutation then gives a tree, as in the first construction. This tree, once completed, gives the tree of the Lyndon word, by writing the letter on the leaves. The whole construction is illustrated in Figure 2.

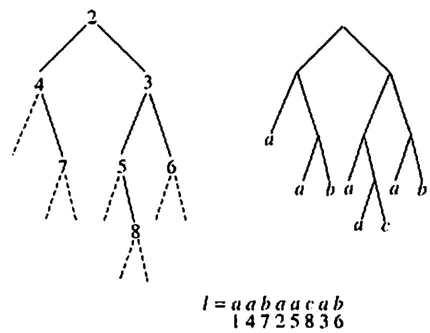

Figure 2

Note that, for the tree associated to a Lyndon word, or equivalently for the standard bracketting of this word, there is an algorithm in [12], extending Schützenberger's for Hall sets. It seems that this algorithm is quadradic in time (but it computes more than the bracketting of a given Lyndon word: it computes the decomposition of any word in the Poincaré-Birkhoff-Witt basis of the free associative algebra), and that the algorithm of the present paper is better: indeed, suffix standardization can be done time $0(n \log n)$, see [10], [2] p.155; moreover, the tree associated to a permutation is constructed in linear time; hence the whole construction is in $0(n \log n)$.

As an application, we derive an algorithm to factorize any word into a decreasing product of Lyndon words see the corollary. In the course of the paper, we also show that the suffix standard permutation of a Lyndon word corresponds to the ordering of the infinite words obtained by iterating its conjugates, see Prop. 2.1; this result is motivated by the bijection of [7] between words and multisets of primitive circular words.

2 Suffix standardization

Let l be a word on a totally ordered alphabet, and $l = s_1, s_2, \dots, s_n$ its nonempty right factors, of decreasing length. The *suffix standard permutation* of l is the unique $\sigma \in S_n$ defined by $s_{\sigma^{-1}(1)} < s_{\sigma^{-1}(2)} < \cdots < s_{\sigma^{-1}(n)}$, where $<$ is the alphabetical ordering. In other words, label the n letters of l from 1 to n, 1 being the label of the first letter of the smallest right factor, 2 the label of the second smallest, and so on. See Figure 2 for an example. Note that σ is not the standard permutation of [13].

When l is a Lyndon word, l is the smallest of all its right factors [9], so that $\sigma(1) = 1$. A more subtle property of Lyndon words is the following: if w is a word, let w^∞ denote the right infinite word obtained by repeating w infinitely often. Then ordering the right factors of l amounts to (lexicographically) order the infinite words obtained by iterating the corresponding conjugates of l. In other words, we have the following result.

Proposition 2.1 *Let l be a Lyndon word. For each right factor s_i of l, as above, let p_i the corresponding left factor, so that $l = p_i s_i$. Let σ be the suffix standard permutation of l. Then $i < j \Leftrightarrow (s_{\sigma^{-1}(i)} p_{\sigma^{-1}(i)})^\infty < (s_{\sigma^{-1}(j)} p_{\sigma^{-1}(j)})^\infty$.*

In other words, with evident notations: $s < s' \Leftrightarrow (sp)^\infty < (s'p')^\infty$.

As an example, take $l = aabaacab$ as in Figure 2. Then we have $\sigma = 14725836$ and indeed $l^\infty < (aacabaab)^\infty < (abaabaac)^\infty < (abaacaba)^\infty < (acabaaba)^\infty < (baabaaca)^\infty < (baacabaa)^\infty < (cabaabaa)^\infty$.

Note that the proposition is not true if l is not a Lyndon word. Indeed, for $l = baa$, we have $\sigma = 321$, but $(aab)^\infty < (aba)^\infty < (baa)^\infty$.

Proof

It is enough to show that if $s_i < s_j$, then $(s_i p_i)^\infty < (s_j p_j)^\infty$. This is clear if s_i is not a left factor of s_j. In the opposite case, we have $s_j = s_i x$ and then $(s_i p_i)^\infty = s_i p_i s_i \ldots = s_i l \ldots$, which is smaller than $s_i x \ldots$ (since x is a right factor of l Lyndon word, so that $l < x$, see [9]), hence $(s_i p_i)^\infty < s_i x l^\infty = s_j l^\infty = (s_j p_j)^\infty$. $\square$

3 Statement of results

Given a complete binary planary tree t with leaves labelled in an alphabet, its *projection* is the word $p(t)$ defined recursively by $p(t) = p(t_1)p(t_2)$, if $t = (t_1, t_2)$. For example, the projection of the tree at the right of Figure 2 is *aabaacab*. Clearly, if t has n leaves, their labels are known once $p(t)$ is known.

Given a Lyndon word l, let $\sigma = 1\ \pi$ be its standard suffix permutation; associate to π (viewed as a permutation of $\{2, \ldots, n\}$) the binary increasing tree obtained by the bijection described in the introduction; complete this tree into a tree t, with leaves labelled so that $p(t) = l$.

On the other hand, if l is written $l = l'l''$, with l'' its smallest proper right factor, then it is known that l', l'' are Lyndon words (see [9]); then the *Lyndon tree* t_1 of l is defined recursively by $t_1 = (t', t'')$, if t', t'' are the Lyndon trees of l', l'' respectively (if l is a letter, $t_1 = l$ is a tree with only one node, labelled l).

With the previous notations, one has the following result.

Theorem 3.1 $t = t_1$.

As an application, we may easily factorize a word into Lyndon words.

Corollary 3.1 *Let w be any word and $w = l_1 \dots l_n$ its unique decreasing factorization into Lyndon words: $l_1 \geq l_2 \geq \cdots \geq l_n$. Let σ be the suffix standard permutation of w and $1 = i_1 < i_2 < \cdots < i_k$ the positions of the left-to-right minima of σ. Then $k = n$ and l_j starts at the i_j − th letter of w.*

Note that this gives another factorization proceduce as the one in [4]; the latter is however better in time, since it is linear, and since the present algorithm is in $0(n \log n)$, as noted at the end of the introduction.

4 Proofs

4.1 Let t_1 be a binary, complete, planary tree with leaves labelled in an alphabet. We have defined the word $p(t_1)$, the projection of t_1. Now, let α be a leaf of t_1: denote by $p(\alpha, t_1)$ the corresponding right factor of $p(t_1)$. Recursively: if $t_1 = (t_2, t_3)$ and α is in t_2 (resp. t_3), then $p(\alpha, t_1) = p(\alpha, t_2)\, p(t_3)$ (resp. $p(\alpha, t_3)$). For example, for α the leftmost leaf labelled b in the tree t_1 at the right of Figure 2, we have $p(\alpha, t_1) = baacab$.

4.2 There is a natural bijection φ between the set of internal nodes of a binary complete planary tree t_1 and the set of its leaves, excluding the leftmost leaf: if x is an internal node, let $l(x)$ (resp. $r(x)$) be its left (resp. right) successor in the tree. Then $\varphi(x) = l^i r(x)$ with i maximum. For example, in Figure 2, internal node 2 is mapped onto the third leaf (from the left) labelled a.

4.3 Let t_1 be the Lyndon tree of the Lyndon word l. Let t' be the tree obtained by suppressing in t_1 the leaves, with each node x in t' labelled by the word $p(\varphi(x), t_1)$ (which is a right factor of l). Since binary increasing trees are in one-to-one correspondence with permutations, and since t_1 is clearly the tree obtained by completing t', in order to prove the theorem, it is enough to show that t' is an increasing tree. See Figure 3, which should be compared to Figure 2.

This amounts to show that for any two internal nodes x, y of t_1, with x preceding y (that is, y is in the subtree rooted at x), one has $p(\varphi(x), t_1) < p(\varphi(y), t_1)$.

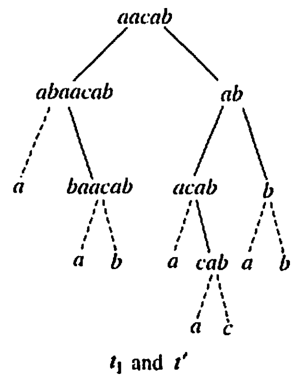

Figure 3

It is enough to prove this when y is a successor of x in t_1; so we may distinguish two cases: $y = l(x), y = r(x)$.

4.4 Note that if x is an internal node of t_1, then $p(\varphi(x), t_1)$ is equal to the product $p(t_1) \dots p(t_n)$, with $p(t_1) \geq \dots \geq p(t_n)$, where $t_1, \dots, t_n$ are the subtrees hanging at the right of the unique path form x to the root, in this order. Indeed, this is seen by geometric inspection of the tree, and the inequalities follow from the fundamental lemma of Melançon (see [11] Lemma 2.1, noting that the set of Lyndon words is a particular Hall set).

4.5 Suppose that $y = l(x)$. Then, by 4.4, we have $p(\varphi(x), t_1) = l_2 \dots l_n$, $p(\varphi(y), t_1) = l_1 l_2 \dots l_n$ for some Lyndon words with $l_1 \geq l_2 \geq \dots \geq l_n$; see Figure 4.

This implies that, in the set of sequences of Lyndon words, ordered alphabetically, the sequence $(l_2, \dots, l_n)$ is smaller than $(l_1, \dots, l_n)$. But another result of Melançon ([11] p. 299-300), is that the alphabetical order on words coincides with the alphabetical order of sequences of Lyndon words, each word being identified with the unique decreasing sequence whose product is this word.

Thus $p(\varphi(x)(, t_1) < p(\varphi(y), t_1)$, what was to be shown.

4.6 Suppose now that $y = r(x)$. Then by 4.4, we have $p(\varphi(x), t_1) = l_1 l_2 \dots l_n$ and $p(\varphi(y), t_1) = l_1'' l_2 \dots l_n$ for some Lyndon words $l_1'', l_1, \cdots, l_n$ with $l_1'' > l_1 \geq \dots \geq l_n$ (the first inequality since l_1'' is a right factor of l_1); see Figure 4.

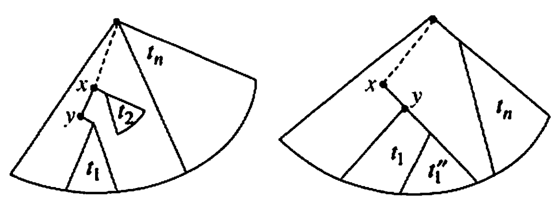

Figure 4

This implies that one has the inequality of sequences: $(l_1, \ldots, l_n) < (l_1'', l_2, \ldots, l_n)$, and, by the same result as in 4.5, we have $p(\varphi(x), t_1) < p(\varphi(y), t_1)$. This proves the theorem.

4.7 We now prove the corollary. Note that if $l = aw$ (a is the first letter of l) is a Lyndon word, and if $t_1, \ldots, t_n$ are the trees hanging at the right of the path from the extreme left leaf x of the Lyndon tree t_1 of l to the root, in this order, then the $p(t_i)$ are Lyndon words and $p(t_1) \geq \cdots \geq p(t_n)$. This is the "décomposition normale gauche" of [14] (extended by Melançon in his lemma quoted in 4.4). See Figure 5.

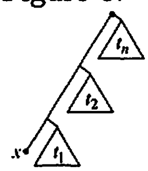

Figure 5

Now, if σ is a permutation and t its associated increasing tree, then the left-to-right minima of σ lie on the extreme left branch of t (see [16] p. 24).

To conclude, it is enough to apply all this, given an arbitrary word w, to the Lyndon word $l = aw$ where a is a letter smaller than any letter in w.

Remark

Note that Melançon's lemma is valid for each Hall set H. Moreover, he has an order $<_H$ which plays the same role as alphabetical order for Lyndon words. So it seems likely that all our results extend to arbitrary Hall sets.

Aknowledgments: thanks to J. Berstel, M. Crochemore, P. Flajolet and X. Viennot, for useful references.

References

[1] Chen, K.-T., Fox, R., and Lyndon, R. (1958). Free differential calculus IV. The quotient group of the lower central series. *Ann. Math,* 68:81-95.

[2] Crochemore, M., Hancart, T., and Lecroq, T. (2001). *Algorithmique du texte.* Vuibert.

[3] Donaghey, R. (1975). Alternating permutations and binary increasing trees. *J. Combin. Th. A*, 18:141-148.

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

[5] Foata, D. and Schützenberger, M.-P. (1971). Nombres d'Euler et permutations alternantes. University of Florida, Gainesville. Preprint dep. of mathematics.

[6] Françon, J. (1976). Arbres binaires de recherche: propriétés combinatoires et applications. *RAIRO Inform. Théorique*, 10:37-50.

[7] Gessel, I. and Reutenauer, C. (1993). Counting permutations with given cycle structure and descent set. *J. Combin. Th. A*, 64:189-215.

[8] Kundu, S. (1977). Sorting tree, nestling tree and inverse permutation. *Information and Processing Letter*, 6:94-96.

[9] Lothaire, M. (1983). *Combinatorics on words.* Addison Wesley; second edition Cambridge University Press, 1997.

[10] Mc Creight, E. (1976). A space-economical suffix tree construction algorithm. *J. Assoc. Comput. Mach.*, 23:262-272.

[11] Melançon, G. (1992). Combinatorics of Hall trees and Hall words. *J. Combin. Th. A*, 59:285-308.

[12] Melançon, G. and Reutenauer, C. (1989). Lyndon words, free algebras and shuffles. *Canad. J. Math*, 41:577-591.

[13] Schensted, C. (1961). Longest increasing and decreasing subsequences. *Canad. J. Math*, 13:179-191.

[14] Schützenberger, M.-P. (1958). Sur une propriété combinatoire des algèbres de lie libres pouvant être utilisée dans un problème de mathématiques appliquées. Séminaire P. Dubreil. Paris. Faculté des Sciences.

[15] Sedgewick, R. and Flajolet, P. (1996). *An introduction to the analysis of algorithms.* Addison Wesley.

[16] Stanley, R. (1986). *Enumerative Combinatorics*, volume A. Wadsworth and Brooks/Cole.

[17] Viennot, G. (1976). Quelques algorithmes de permutation. *in: Journées algorithmiques, Paris, Soc. Math. France, Astérisque*, 38-39:275-293.

[18] Vuillemin, J. (1980). A unifying look at data structures. pages 229-239. Comm. of the ACM 23.