Shuffle of Parenthesis Systems and Baxter Permutations

ROBERT CORI, SERGE DULUCQ, AND GÉRARD VIENNOT

Laboratoire de Mathématiques et d'Informatique,
Laboratoire associé au C.N.R.S. n° 226, Université de Bordeaux I,
351, Cours de la Libération, 33405 Talence Cedex, France

Communicated by the Managing Editors
Received April 12, 1984

A formula for the number alternating Baxter permutations is given. The proof of this formula is given by constructing bijection between permutations, trees, and words. This gives also a combinatorial proof of a formula appearing in the enumerative theory of planar maps.

1. INTRODUCTION

The Catalan number \( C_n = \frac{(2n)!}{n!(n+1)!} \) play an important role in enumerative theory. They appear as counting a various class of combinatorial objects: binary trees, parenthesis systems, dissections of \( n \)-gons into triangles etc. We are interested here in objects enumerated by the product of two consecutive Catalan numbers \( C_n C_{n+1} \) or by the square of a Catalan number \( C_n^2 \). These numbers appear in the enumeration of certain families of planar maps (Mullin [10], Tutte [11]); our main result here is that they enumerate also alternating Baxter permutations.

Baxter permutations (introduced by Baxter [3] in the study of the composition of two commuting functions) are permutations \( \sigma_1, \sigma_2, \ldots, \sigma_n \) of 1, 2, \ldots, \( n \) satisfying for any \( 1 \leq i < j < k < l \leq n \):

\[
\begin{align*}
\sigma_i + 1 &= \sigma_j, \\
\sigma_j > \sigma_i &\Rightarrow \sigma_k > \sigma_i \\
\sigma_i + 1 &= \sigma_i, \\
\sigma_k > \sigma_i &\Rightarrow \sigma_j > \sigma_i
\end{align*}
\]

Their number was given by Chung, Graham, Hoggatt, and Kleiman [4] using Macsyma (see also Mallows [9]), a combinatorial proof of this result was found by Viennot [13]. A permutation is alternating if for any \( i \):

\[
\begin{align*}
0 < 2i &\leq n \Rightarrow \sigma_{2i} > \sigma_{2i-1} \\
1 \leq 2i + 1 &\leq n \Rightarrow \sigma_{2i} > \sigma_{2i+1}
\end{align*}
\]

\((\text{Alt})\)
It is known since Andre (1879) [1, 2], that alternating permutations are counted by the tangent and secant numbers (see also Foata & Schützenberger [5], Foata & Strehl [6], Viennot [12] for a combinatorial development). We prove here the result:

**Theorem 1.1.** The number of permutations of \{1, 2, ..., 2n\} satisfying \((B_1, B_2)\) and \((A_l)\) is \(C_n \cdot C_n\). The number of such permutations of \{1, 2, ..., 2n + 1\} is \(C_n C_{n+1}\).

As we mentioned these numbers appear also in the enumerative theory of planar maps. A planar map with a hamiltonian circuit and in which each face is a triangle, can be divided into two “triangulations” of the hamiltonian polygon (one inside the polygon, the other one outside). This remark gives an intuitive reason for the fact that these “hamiltonian rooted triangular maps” are enumerated by a product of two Catalan numbers. Dually, hamiltonian cubic maps are planar maps with a hamiltonian circuit in which all vertices have degree three. In such a map any vertex is incident with only one edge not in the hamiltonian polygon, this edge may be inside the polygon or outside. Thus to build a “Hamiltonian rooted cubic map” one has to choose \(2k\) vertices among the \(2n\) (those incident with inside edges) then draw a planar map inside the polygon (it is easy to see that this can be done in \(C_k\) ways) and a planar map outside. We have thus also an intuitive proof of the fact that the number of “hamiltonian rooted cubic maps with \(2n\) vertices” is

\[ H_n = \sum_{k=0}^{n} \binom{2n}{2k} C_k C_{n-k}. \]

A straightforward computation gives:

\[ H_n = \frac{2n!}{(n+1)! (n+1)!} \sum_{k=0}^{n} \binom{n+1}{k} (n+1)(k+1), \]

\[ = \frac{2n!}{(n+1)! (n+1)!} \binom{2n+2}{n+2} = C_n C_{n+1}. \]

And we have the combinatorial identity:

\[ C_n C_{n+1} = \sum_{k=0}^{n} \binom{2n}{2k} C_k C_{n-k}. \]  

(I)

Duality does not give a combinatorial interpretation of this identity: if the dual of a cubic map is a triangular one, unfortunately the dual of hamiltonian cubic map may not be hamiltonian as remarked by Mullin.
who asked in [10] for a combinatorial interpretation of (I). As a by-product of our considerations we will give here a bijective proof of (I).

Our main tool is what we call Baxter trees. They are introduced (in Sect. 2) as all the labeled binary complete trees obtained from the elementary tree

\[
\begin{array}{c}
1 \\
0 \quad 0
\end{array}
\]

by a sequence of operations called "graft" and "desactivation." The relationship between Baxter trees and Baxter permutations is given in Part 3: traversing a Baxter tree "in order" yields an alternating Baxter permutation, conversely any such permutation can be obtained in that way. It is thus equivalent to enumerate Baxter trees or alternating Baxter permutations. A use of the combinatorial identity (I) could have led us directly to an enumerative theorem. We prefer to investigate the combinatorial properties of Baxter trees obtaining thus a bijective proof of (I).

In Part 4, we look for the "code" of a Baxter tree. This gives a "context free" language \( L \) related to the Dyck language (set of well formed parenthesis systems). Syntactic analysis of a word of the language \( L \) gives the way for reconstructing the Baxter tree from its code. In Part 5 we prove that the number of labellings of a binary complete tree in order to obtain a Baxter tree is precisely a Catalan number. In Part 6 all the combinatorial consequences of the investigations of Parts 4 and 5 are obtained.

Let us recall first a few definitions that will be of constant use in the sequel that concern essentially words (see, for instance, [8] for further details):

We shall use sets which are called alphabets whose elements are called letters. The main alphabets we shall use are \( \{l, r\}, A = \{a, b, \bar{a}, \bar{b}\}, X = \{x, y, u, v\}, Y = \{u, v, y, \bar{y}\} \), and also alphabets containing the integers \( 1, 2, \ldots, n \) as letters. A word is a finite sequence of letters which we often denote by \( f = a_1a_2\cdots a_n \). The empty sequence (or the empty word) will be denoted by \( \emptyset \). The set \( A^* \) of words on the alphabet \( A \), or the free monoid generated by \( A \), is defined by the binary operation of the concatenation of two words or sequences; thus a word can be considered as the concatenation of its letters. Of course, \( \emptyset \) is a neutral element for this operation.

The length of a word \( f \) denoted by \( |f| \) is the number of letters of \( f \). For a letter \( x \) in the alphabet, \( |f|_x \) denotes the number of letters of \( f \) that are equal to \( x \).

A word \( f' \) is a left factor of a word \( f \) if there exists a word \( f'' \) such that \( f = f'f'' \); it is a proper left factor if \( f' \neq f \).

Given the alphabet \( \{a, \bar{a}\} \), a word \( f \) is a parenthesis system on \( \{a, \bar{a}\} \) if
$|f|_a = |f|_a$ and $|f''|_a \geq |f'|_a$ for any left factor $f'$ of $f$. The set of parenthesis systems on \( \{a, \bar{a}\} \) will be denoted by $P_{a, \bar{a}}$. It is often called the Dyck language. It is well known that the number of parenthesis systems of length $2n$ is the Catalan number $C_n$.

The shuffle of two words $f$ and $g$ of a free monoid $A^*$ is the subset denoted by $f \shuffle g$ of all words $h$ such that $h = f_1 g_1 f_2 \cdots f_n g_n$, $f_i, g_i \in A^*$, $f = f_1 f_2 \cdots f_n$, $g = g_1 g_2 \cdots g_n$. The shuffle of two subsets $L$ and $M$ of $A^*$ is the union of the shuffle of any element of $L$ by each element of $M$. It will be denoted by $L \shuffle M$. In the sequel we will often consider the set $P_{a, \bar{a}} \shuffle P_{\bar{b}, b}$ of shuffles of two parenthesis systems.

2. Baxter Trees

2.1. Complete Binary Trees

In what follows we consider subsets of the free monoid generated by the alphabet containing two letters $l$ and $r$.

**Definition 2.1.** A complete binary tree is a nonempty subset $B$ of the free monoid $\{l, r\}^*$ satisfying the following conditions

1. $f g \in B \Rightarrow f \in B$ \hspace{1cm} (A1)
2. $fl \in B \Rightarrow fr \in B$. \hspace{1cm} (A2)

Condition (A1) implies that the empty word $\dagger$ is always an element of a binary tree $B$. The tree consisting only of the element $\dagger$ is denoted by $B_0$ in the sequel.

**Example 2.2.** A complete binary tree.

$$B = \{\dagger, l, ll, lr, lrl, lrlr, lrlrl, lrlrr, lrr, r, rl, rr, rlr, rll\}.$$  

Generally a complete binary tree $B$ is drawn in the plane in the following way: to each element of $B$ corresponds a point in the plane, the point corresponding to the empty word $\dagger$ is in the top of the drawing; each $f$ is joined by a straight line going down left to $fl$ and by a straight line going down right to $fr$, if these words belong to $B$. The drawing corresponding to Example 2.2 is given in Fig. 1.

An element $f$ in $B$ such that $fl$ (and also $fr$ by A2) is not in $B$ is a *leaf*. If the last letter of $f$ is an $l$ then $f$ is a *left leaf*, if the last letter is $r$ then it is a *right leaf*. The elements of $B$ are also called the nodes of $B$. 

DEFINITION 2.3. An increasing binary tree is a complete binary tree with a mapping \( \varepsilon \) (label) of \( B \) in \{0, 1, 2,..., n\} such that

C1 \( \varepsilon(f) = 0 \Rightarrow f \) is a leaf

C2 \( \varepsilon \) restricted to the set of elements such that \( \varepsilon(f) \neq 0 \) is one to one

C3 \( \varepsilon(fg) \neq 0, \ g \neq \emptyset \Rightarrow \varepsilon(f) < \varepsilon(fg) \).

The leaves of an increasing binary tree such that \( \varepsilon(f) = 0 \) will be called active. An increasing binary tree is given in Fig. 2.

DEFINITION 2.4. The symmetric ordering \( <_s \) on the elements of \{l, r\} is the total ordering given by:

\[ f <_s f' <_s f'' \text{ for any } f, f', f'' \in \{l, r\}. \]

For any subset of \{l, r\} the symmetric ordering induces a permutation of its elements.

EXAMPLE 2.5. The ordering of the elements of the complete binary tree of Example 2.2 is:

\[ ll, l, lrl, lrl, lrl, lrl, lrr, lr, lrr, l, rll, rl, rlr, r, rr. \]

In the sequel \{l, r\} will be implicitly ordered by \( <_s \), thus the least element of \( B \), will mean the least element of \( B \) with respect to \( <_s \).
2.2 The Graft and Desactivation Operations

We introduce four operations associating an increasing tree on \( \{1, 2, \ldots, n+1\} \) to any increasing tree on \( \{1, 2, \ldots, n\} \); let us give their intuitive definition:

**Left graft** consists in giving the label \( n+1 \) to the greater left active leaf and adding the elementary tree

\[
\begin{array}{c}
\circ \\
\circ \\
0 \quad 0
\end{array}
\]

below it.

**Right graft** consists in the same operation as the left graft but done on the least right active leaf.

**Right desactivation** consists in giving the label \( n+1 \) to the least right active leaf.

**Left desactivation** consists in giving the label \( n+1 \) to the greater left active leaf.

These operations are respectively denoted by \( a, b, \overline{a}, \) and \( \overline{b} \). Operation \( a \) increases by one the number of right active leaves while \( \overline{a} \) decreases it by one. Similarly \( b \) increases the number of left active leaves while \( \overline{b} \) decreases it. Clearly \( a, \overline{b} \) are defined only for trees having an active left leaf and symmetrically \( b \) and \( \overline{a} \) act only on trees having an active right leaf.
After this intuitive description of the operations $a, \bar{a}, b, \bar{b}$ we can give the precise definitions. Let $(B, \varepsilon)$ be an increasing binary tree, with at least one active left leaf and right leaf. Let $fl$ be the greater active left leave ($fll \notin B; \varepsilon(fl) = 0$; $gl \notin B \Rightarrow gl \leq_S fl$) and $gr$ the least active right leaf.

**Definition 2.6.** The trees $(B, \varepsilon) \cdot a = (B_a, \varepsilon_a)$; $(B, \varepsilon) \cdot b = (B_b, \varepsilon_b)$; $(B, \varepsilon) \cdot \bar{a} = (B, \varepsilon_{\bar{a}})$; $(B, \varepsilon) \cdot \bar{b} = (B, \varepsilon_{\bar{b}})$ are given by

$$
B_a = B \cup \{fll, flr\} \quad B_b = B \cup \{grl, gr\r
$$

$\varepsilon_a$ and $\varepsilon_b$ are equal to $\varepsilon$ for all elements except for $fl$, $fll$, and $flr$, $\varepsilon_a(fl) = \varepsilon_b(fl) = n + 1$; $\varepsilon_a(fll) = \varepsilon_b(fll) = 0$;

$\varepsilon_{\bar{a}}$ and $\varepsilon_{\bar{b}}$ are equal to $\varepsilon$ for all elements except for $gr$, $grl$, and $grr$, $\varepsilon_{\bar{a}}(gr) = \varepsilon_{\bar{b}}(gr) = n + 1$; $\varepsilon_{\bar{a}}(grl) = \varepsilon_{\bar{b}}(grl) = 0$.

The action of a word $\alpha$ of $\{a, b, \bar{a}, \bar{b}\}^*$ on an increasing binary tree $(B, \varepsilon)$ is defined as the composition of the actions of the letters of the word read from left to right.

Of course, this composition of actions is not always defined. However, if one supposes that the tree $(B, \varepsilon)$ has at least one active left and one active right leaf, and if the word $\alpha$ satisfies the following condition for any left factor $\alpha'$ of $\alpha$:

$$
|\alpha'|_a \geq |\alpha'|_b, \quad |\alpha'|_b \geq |\alpha'|_{\bar{a}}
$$

then $(B, \varepsilon) \cdot \alpha$ is well-defined.

**Definition 2.7.** A Baxter tree is any increasing binary tree obtained from the tree $(B_1, \varepsilon_1) = \begin{array}{c}
1 \\
0 \\
0
\end{array}$

$B_1 = \{1, l, r\}$, $\varepsilon_1(1) = 1$, $\varepsilon_1(l) = \varepsilon_1(r) = 0$; by the action of a word $\alpha$ of $\{a, \bar{a}, b, \bar{b}\}^*$.

**Example 2.8.** The increasing tree of Fig. 2 is a Baxter tree with $(B, \varepsilon) = (B_1, \varepsilon_1) \cdot abu\bar{a}abu$.

**Remark 2.9.** Given a Baxter tree $(B, \varepsilon)$ we can determine the unique tree $(B', \varepsilon')$ and letter $\beta(\beta = a, b, \bar{a}, \bar{b})$ such that $(B', \varepsilon') \cdot \beta = (B, \varepsilon)$ by examining the node of highest integer label. Thus repeating this operation it is also possible to determine the unique $\alpha$ such that $(B_1, \varepsilon_1) \cdot \alpha = (B, \varepsilon)$. 
2.3. Permutation Associated to a Baxter Tree

Let \((B, \varepsilon)\) be an increasing binary tree with \(k\) elements, and let \(f_1 <_s f_2 <_s f_3 \cdots <_s f_k\) be the permutation induced on the elements of \(B\) by the symmetric order \(<_s\).

**Definition 2.10.** The word \(\mathcal{P}(B, \varepsilon)\) associated to an increasing binary tree is \(\mathcal{P}(B, \varepsilon) = \eta(f_1) \cdot \eta(f_2) \cdots \eta(f_k)\), where \(\eta(f_i) = \varepsilon(f_i)\) if \(\varepsilon(f_i) \neq 0\); \(\eta(f_i) = u\) if \(\varepsilon(f_i) = 0\) and \(f_i\) is a left leaf, \(\eta(f_i) = v\) if \(\varepsilon(f_i) = 0\) and \(f_i\) is a right leaf.

---

Fig. 3. The four operators \(a, b, \bar{a}, \bar{b}\) on the increasing binary tree of Fig. 2.
Fig. 4. The four operations on the word associated to a Baxter tree.

The operations \( a, b, \tilde{a}, \tilde{b} \) on the tree \((B, \varepsilon)\) induce operations on the words of the free monoid \(\{u, v, 1, 2, \ldots, n\}^*\) containing exactly one letter \(i\) for \(1 \leq i \leq n\). \(\mathcal{S}((B, \varepsilon) \cdot a)\) and \(\mathcal{S}((B, \varepsilon) \cdot b)\) are obtained from \(\mathcal{S}(B, \varepsilon)\) replacing the rightest \(u\) by \(u_{n+1} v\) and \(n+1\) respectively. Similarly \(\mathcal{S}((B, \varepsilon) \cdot b)\) and \(\mathcal{S}((B, \varepsilon) \cdot \tilde{a})\) are obtained replacing the leftmost \(v\) in \(\mathcal{S}(B, \varepsilon)\) by \(u_{n+1} v\) and \(n+1\), respectively. From this we have;

**Proposition 2.11.** If \((B, \varepsilon)\) is a Baxter tree, then in \(\mathcal{S}(B, \varepsilon)\) all the letters \(v \) appear after the letters \(u\), moreover all the \(u\) are before the greatest number \(n\) in the world and all the letters \(v\) after this \(n\).

The operations given in Fig. 3 for the tree \((B, \varepsilon)\) are displayed in Fig. 4 for the word \(\mathcal{S}(B, \varepsilon)\).

3. **Alternating Baxter Permutations and Their Trees**

The definition of a Baxter permutation is given in the introduction, we extend this definition to the words of \(\{1, 2, \ldots, n, u, v\}^*\) containing exactly one letter equal to \(i\) for any \(i \leq i \leq n\) by considering the following order of the alphabet \(1 < 2 \cdots < n < u, v\).

**Definition 3.1.** A word \(f\) of length \(m\) containing exactly one letter equal to \(i\) for any \(1 \leq i \leq n\) is a Baxter word if for any \(1 \leq i < j < k < l \leq m\):
(B1) $f_i = f_i + 1$ and $f_j > f_i \Rightarrow f_k > f_i$.

(B2) $f_i = f_i + 1$ and $f_k > f_i \Rightarrow f_j > f_i$.

Remark that since "u + 1" and "v + 1" are not defined, $f_i$ and $f_i$ are different from $u$ and $v$ in (B1) and (B2). Moreover a word $f$ equal to $f_1 if_2 xf_3 if_4 i + 1 f_5$ is not a Baxter word if $x = u$ or $x = v$ and if $j < i$.

**DEFINITION 3.2.** A word $f = f_1 \cdots f_m$ is alternating if $f_1 > f_2$, $f_2 < f_3$, $f_3 > f_4 \cdots f_{2i-1} > f_{2i}, f_{2i} < f_{2i+1}$.

**PROPOSITION 3.3.** Let $(B, \varepsilon)$ be a Baxter tree then $\mathcal{S}(B, \varepsilon)$ is an alternating Baxter word.

**Proof.** We prove this proposition by induction on the length of the word $\alpha$ such that $(B_1, \varepsilon_1) \cdot \alpha = (B, \varepsilon)$. Clearly as $\mathcal{S}(B_1, \varepsilon_1) = uv$ the result is true for $|\alpha| = 0$.

Assume the proposition satisfied by $(B, \varepsilon)$ and let us prove it for $(B, \varepsilon) \cdot \alpha$ where $\alpha = a, b$ or $\overline{a}$ or $\overline{b}$. That $\mathcal{S}(B, \varepsilon)$ is alternating is clear, we are only going to prove that (B1) is satisfied, the proof of (B2) is very similar. Let us first establish the following claim: (which is condition (B1) for $f_j = u$ or $v$).

“If $\mathcal{S}((B, \varepsilon) \cdot \alpha)$ is equal to $f_1 if_2 xf_3 i + 1 f_4$ where $x$ is either $u$ or $v$ then $f_3$ does not contain any $j$ with $j < i$.”

We distinguish two cases:

(a) $i \neq n$. In which case $\mathcal{S}(B, \varepsilon)$ can be written in the same way: $\mathcal{S}(B, \varepsilon) = f_1 if_2 xf_3 i + 1 f_4$ and $f_2 xf_3$ is either equal to $f_2 xf_3$ or obtained from it replacing a “$u$” or a “$v$” by $n + 1$ or by $n+ 1 v$. In either cases applying the inductive hypothesis $f_3$ does not contain a letter “$f$” with $j < i$, and the same property holds for $f_3$.

(b) $i = n$. In that case $\mathcal{S}(B, \varepsilon)$ can be written: $\mathcal{S}(B, \varepsilon) = f_1 yf_2 f_3 yf_4$, where $y = u$ or $v$, and $\mathcal{S}(B, \varepsilon)$. $\alpha$ is obtained from $\mathcal{S}(B, \varepsilon)$ replacing $y$ by $n + 1$ or by $n+ 1 v$. From Proposition 2.11 $y$ is necessarily equal to $v$ as $n$ is the larger number appearing in $\mathcal{S}(B, \varepsilon)$. Also $x$ cannot appear in $f_3$ as it would be necessarily equal to $v$ (it is at the right of $n$) and then the replacement would take place at the leftmost $v$. Thus, as $f_3$ cannot contain any $i (i < n)$ by the inductive hypothesis, $f_3 = 1$ and the claim is proved.

Let us now consider the rest of Proposition 3.3 and let us examine $\mathcal{S}((B, \varepsilon) \cdot \alpha) = f_1 if_2 f_3 i + 1 f_4$ where $j > i$. We have to prove that any $k$ appearing in $f_3$ is greater than $i$. Two cases are also considered:

(a) $j \neq n + 1$. In that case $\mathcal{S}(B, \varepsilon)$ can be written in the same way: $f_1 if_2 f_3 i + 1 f_4$ and by the inductive hypothesis $f_3$ does not contain any $k$
with \( k < i \); this is also true for \( f_3 \) as it is equal to \( f'_3 \) or obtained from it adding \( n + 1 \) or \( u \ n + 1 \ v \);

(b) \( j = n + 1 \). In that case \( \mathcal{S}(B, \varepsilon) = f_1 i f'_2 x f'_3 i + 1 f_4 \) where \( x \) is equal to \( u \) or \( v \), and \( f'_3 \) contains the same \( k \) as \( f_3 \). The result then follows from the claim.

**Proposition 3.4.** Let \( f \) be an alternating Baxter word of \( \{1, 2, \ldots, n, u, v\}^* \) with \( f = f'nf'' \) where \( |f'|_u = |f''|_u = 0 \). Then there exists a unique Baxter tree \((B, \varepsilon)\) such that \( f = \mathcal{S}(B, \varepsilon) \).

**Proof.** We also prove this result by induction on \( n \). If \( n = 1 \) then \( f = uv \) and \((B, \varepsilon) = (B_1, \varepsilon_1)\). Let now \( n > 1 \) and \( f = f'nf'' \), assume that \( f' \) ends with the letter \( u \) and \( f'' \) begins with the letter \( v \). If not a similar proof holds.

We thus have \( f = f'_1 u f''_1 \) let us denote by \( g \) the word \( f'_i u f''_i \) or \( f'_i v f''_i \) according to the position of \( n - 1 \) (in \( f'_i \) or in \( f''_i \)). Let us show that \( g \) satisfies the conditions of the proposition:

- \( g \) is alternating as \( f \) is,
- \( g \) is a Baxter word as a subword of a Baxter word is also a Baxter word.

If \( f'_i \) contains \( n - 1 \) then we have to prove that there is no letter \( u \) at the right of \( n - 1 \) in \( f'_i \).

If such is the case then: \( f = f'_2 n - 1 f'_3 u f'_4 v f''_1 \), where \( f'_4 \) contains a number \( j \) less than \( n - 1 \) (\( f \) being alternating no two \( u \) are consecutive in \( f \)), and we would have

\[
 f = f'_2 n - 1 f'_3 u f'_4 v f''_1 = f,
\]

contradicting the fact that \( f \) is a Baxter word.

Thus \( g \) satisfies the inductive hypothesis and there exists a Baxter tree \((B', \varepsilon')\) such that \( g = \mathcal{S}(B', \varepsilon') = f'_i u f''_i \) (or \( f'_i v f''_i \)). There is a leaf \( f_u \) (or \( f_v \)) in \( B' \) corresponding to the letter \( u \) (or \( v \)) of \( f'_i u f''_i \) (or \( f'_i v f''_i \)); moreover as \( f''_i \) does not contain any letter \( u \) (or \( f'_i \) does not contain any letter \( v \)) this leaf is the greater left active leaf (or the least right active leaf) then \((B', \varepsilon') \cdot a \) (or \((B', \varepsilon') \cdot b\)) is the Baxter tree such that

\[
 \mathcal{S}(B, \varepsilon) = f'_i u f''_1 = f,
\]

and Proposition 3.4 is proved.

From Propositions 3.4 and 3.3 it is easy to obtain the result:

**Theorem 3.5.** For \( n \geq 0 \) there exist bijections between:

1. The set of alternating Baxter permutations on \( \{1, 2, \ldots, 2n + 1\} \).
(2) The set of Baxter trees with $2n + 3$ nodes in which only the least left leaf and the greater right one are active.

(3) The set of words $\alpha$ of $\{a, b, \bar{a}, \bar{b}\}$* of length $2n$ such that $|\alpha|_a = |\alpha|_{\bar{a}}$, $|\alpha|_b = |\alpha|_{\bar{b}}$ and $|\alpha'\alpha|_a > |\alpha'|_{\bar{a}}$, $|\alpha'\alpha|_b > |\alpha'|_{\bar{b}}$ for any left factor $\alpha'$ of $\alpha$. (It is the set of words of $P_{a,\bar{a}} \sqcup P_{b,\bar{b}}$ of length $2n$).

There exist also bijections between:

(1') The set of alternating Baxter permutations on $\{1, 2, \ldots, 2n\}$.

(2') The set of Baxter trees with $2n + 2$ nodes in which only the least left leaf is active.

(3') The set of words $\alpha$ of $\{a, b, \bar{a}, \bar{b}\}$* of length $2n - 1$ such that $|\alpha|_a = |\alpha|_{\bar{a}} - 1$, $|\alpha|_b = |\alpha|_{\bar{b}}$, $|\alpha'|_a \geq |\alpha'|_{\bar{a}} - 1$, $|\alpha'|_b \geq |\alpha'|_{\bar{b}}$ for any left factor $\alpha'$ of $\alpha$ and $|\alpha'|_a \geq |\alpha'|_{\bar{a}}$ for any $\alpha'$ such that $\alpha = \alpha'b\alpha''$.

Let $\sigma$ be an alternating Baxter permutation on $\{1, 2, \ldots, 2n + 1\}$ and let us consider the word $u\sigma v$. Clearly it is a Baxter word, and by Proposition 3.3, there is a unique Baxter tree $(B, \varepsilon)$ such that $\mathcal{S}(B, \varepsilon) = uv$. In this tree only the least left leaf and greater right one are active (because of the position of $u$ and $v$ in $uv$). Thus $\mathcal{S}$ is the bijection between (1) and (2). It is also the bijection between (1') and (2').

Now if $\alpha$ is such that $(B_1, \varepsilon_1) \alpha$ has only one left active and one right active leaf then $|\alpha|_a = |\alpha|_{\bar{a}}$ and $|\alpha|_b = |\alpha|_{\bar{b}}$, moreover if this left active leaf is the least one then $|\alpha'|_b \geq |\alpha'|_{\bar{b}}$ for any left factor $\alpha'$ of $\alpha$. Similarly $|\alpha'|_a \geq |\alpha'|_{\bar{a}}$. To verify that the mapping associating to any word $\alpha$, satisfying these conditions, the Baxter tree $(B_1, \varepsilon_1) \alpha$ is a bijection, we can use Remark 2.9.

A very similar proof holds for the bijection between (1'), (2'), and (3').

**Notation.** In the sequel we will denote by $P_{a,\bar{a}} \sqcup P_{b,\bar{b}}$ the set of words satisfying Condition 3 (it is an abreviation of $P_{a,\bar{a}} \sqcup \sqcup P_{b,\bar{b}}$) and by $P_{a,\bar{a}} \sqcup P_{b,\bar{b}}$ the set of words satisfying condition (3').

## 4. CODING OF BAXTER TREES

### 4.1. Coding Binary Complete Trees

Given a binary complete tree $B$ different from $B_0$, the elements of $B$ can be divided into two subsets according to their first letter ($l$ or $r$). We thus have the decomposition:

$$B = \{\emptyset\} \cup lB' \cup rB''.$$
Moreover, $B'$ and $B''$ are complete binary trees, the left and the right subtree of $B$ respectively.

**Example 4.1.** For the tree given in Example 2.2 these subtrees are $B' = \{ 1, l, r, rl, rr, rll, rlr, rlrl, rlrr \}$, $B'' = \{ 1, l, r, ll, lr \}$.

**Definition 4.2.** The code $C(B)$ of the binary tree $B$ is recursively defined as a word of $\{x, y\}$* by:

1. If $B = B_0$ then $C(B_0) = y$,
2. If $B \neq B_0$, $B'$ and $B''$ being the left and right subtrees of $f$ then $C(B) = xC(B') C(B'')$.

**Example 4.3.** For the tree $B$ of Example 2.2, 

$$C(B) = xC(B') C(B'') = xxyxyxyyyyyxxyyy.$$ 

An intuitive way to obtain the code of a complete binary tree is to traverse the tree in preorder (visit first the root, then the left subtree and finish with the right subtree), write $y$ when visiting a leaf, and $x$ when visiting a node which is not a leaf. The order $f_1, \ldots, f_k$ in which the leaves are visited is such that $f_1 <_s f_2 \cdots <_s f_k$, that is the same as symmetric order. This result is well known:

**Proposition 4.4.** A word $f$ of $\{x, y\}$* is the coding of a binary complete tree if and only if:

$$|f|_x = |f|_y - 1 \text{ and } |f'|_x \geq |f'|_y \text{ for any proper left factor } f' \text{ of } f.$$ 

Moreover the number of such words of length $2n + 1$ is the Catalan number $C_n$.

**Definition 4.5.** The tree-code of a binary increasing tree $(B, \varepsilon)$ is the word of $\{x, y, u, v\}$* obtained from $C(B)$ replacing the "y's" corresponding to left active leaves by $u$ and those corresponding to right active ones by $v$. We denote it by $C(B, \varepsilon)$.

Of course, two distinct binary increasing trees have the same tree-code if and only if they differ only by the labels of the non-active nodes.

**Example 4.6.** $xxyyxyy$ is the code of the two following trees $(B, \varepsilon_1)$ and $(B, \varepsilon_2)$, where $B = \{ 1, l, ll, lr, r, rl, rr \}$, $\varepsilon_1(1) = \varepsilon_2(1) = 1$, $\varepsilon_1(l) = 2$, $\varepsilon_1(r) = 4$, $\varepsilon_2(l) = 2$, $\varepsilon_2(r) = 4$. 


\[ \varepsilon_1(ll) = \varepsilon_1(rr) = 0, \quad \varepsilon_1(lr) = 3, \quad \varepsilon_1(rl) = 5, \quad \varepsilon_2(l) = 4, \quad \varepsilon_2(r) = 2, \quad \varepsilon_2(ll) = 0, \quad \varepsilon_2(lr) = 5, \quad \varepsilon_2(rl) = 4, \quad \varepsilon_2(rr) = 0. \]

By the following diagram, the operations of graft and desactivation induce operations on the free monoid \( \{x, y, u, v\}^* \):

\[
\begin{array}{ccc}
(B, \varepsilon) & \xrightarrow{\text{graft or desactivation}} & (B, \varepsilon) \cdot \alpha \\
\downarrow & & \downarrow \\
C(B, \varepsilon) & \xrightarrow{\text{induced operation}} & C(B, \varepsilon) \cdot \alpha
\end{array}
\]

**Definition 4.7.** Let \( f \) be a word of \( \{x, y, u, v\}^* \) in which \( u \) appears at least once, let \( f = f'uf'' \), where \( |f''|_u = 0 \). Then \( f \cdot a = f'xuwf'' \), \( fh = f'yf'' \).

Let \( f \) be a word in which \( v \) appears at least once, let \( f = g'vg'' \), where \( |g'|_v = 0 \). Then \( fh = g'xuvwg'' \), \( f\tilde{a} = g'yvg'' \).

As in section 2, we can define the action of a word \( \alpha \) of \( \{a, \tilde{a}, b, \tilde{b}\}^* \) on the word \( f \) of \( \{x, y, u, v\}^* \). It is then clear that \( C(B, \varepsilon) \cdot \alpha = C((B, \varepsilon) \cdot \alpha) \).

The set of tree-codes of Baxter trees which is also the set of all words \( xuv \cdot \alpha (\alpha \in \{a, b, \tilde{a}, \tilde{b}\}^*) \) is denoted by \( L \) in the sequel.

Remark that if \( \alpha \) is a word in \( P_{a \cup b} \) or \( P'_{a \cup b} \) (as in Conditions (3) or (3') of Theorem 3.4) then \( xuv \cdot \alpha \) is well defined.

### 4.2. Combinatorial Properties of the Words in \( L \)

**Proposition 4.8.** If \( f \) is a word of \( L \) then

\[
\begin{align*}
(L_1) & \quad |f|_x + 1 = |f|_y + |f|_u + |f|_v, \\
(L_2) & \quad |f'|_x + 1 > |f'|_y + |f'|_u + |f'|_v \quad \text{for any proper left factor } f' \text{ of } f, \\
(L_3) & \quad f \in \{x, y, u\}^* \cdot \{x, y, v\}^*, \\
(L_4) & \quad \text{any } u \text{ in } f \text{ is immediately preceded by an } x, \\
(L_5) & \quad \text{any } v \text{ in } f \text{ is preceded by either } u \text{ or } v \text{ or } y.
\end{align*}
\]

**Proof.** (\( L_1 \)) and (\( L_2 \)) are immediate consequences of Proposition 4.4; as the \( u \) (or the \( v \)) replaced by \( y \) or by \( xuv \) is always the rightmost (resp. the leftmost) yields (\( L_3 \)): no \( "v" \) can precede a \( "u" \) in \( f \). (\( L_4 \)) and (\( L_5 \)) can be obtained using induction on the length of \( f \) (or of \( \alpha \) such that \( xuv \cdot \alpha = f \)).

We wish now to prove that any word \( f \) satisfying \( L_1, \ldots, L_5 \) is an element of \( L \) to do this we have to "parse" \( f \) and guess the way it is obtained from \( xuv \).
DEFINITION 4.9. Let \( f \) be a word satisfying \( L_3, L_4, L_5 \) the \( L \)-decomposition of \( f \) is a triple of words \( f_1, f_2, f_3 \) such that: \( f = f_1 f_2 f_3 \); \( f_1 \) is either the empty word or ends with \( u \); \( |f_2|_u = |f_2|_v = 0 \); \( f_3 \) is either the empty word or begins with \( v \).

PROPOSITION 4.10. Let \( f \) be a word satisfying \( L_3, L_4, L_5 \) and let \( f_1, f_2, f_3 \) be the \( L \)-decomposition of \( f \) then:

(A) If \( f_2 = \emptyset, f_3 \neq \emptyset \), and \( f_1 \) ends with \( xxu \) then there exists \( g \) such that \( g \cdot a = f \).

(B) If \( f_2 = \emptyset, f_3 \neq \emptyset \), and \( f_1 \) ends with either \( yxu \) or \( uxu \) then there exists \( g \) such that \( g \cdot b = f \).

(C) If \( |f_2| \neq 0 \), then there exists \( g \) such that either \( g \cdot \overline{a} = f \) or \( g \cdot \overline{a} = f \).

Moreover such a \( g \) satisfies \( L_3, L_4, L_5 \).

Proof. If (A) holds, then \( f = f'xxu f'' \) with \( |f''|_u = 0 \). Consider \( g = f'xuf'' \) then \( g \cdot a = f \). If (B) holds, then \( f = f'yxu f'' \) or \( f = f'uxuf'' \). Then \( g = f'yuf'' \) or \( g = f'uxuf'' \) is such that \( g \cdot b = f \). If (C) holds, then \( f_2 = f_2' y f_2'' \); let \( g \) be \( f_1 f_2' u f_2'' f_3 \) or \( f_1 f_2'' v f_2' f_3 \) according to the last letter of \( f_1 f_2 \) is \( x \) or not, then in the first case \( g \cdot b = f \) and in the second case, \( g \cdot \overline{a} = f \). Clearly as \( f \) satisfies \( L_3, L_4, L_5 \) the \( g \) defined satisfies also \( L_3, L_4, \) and \( L_5 \).

THEOREM 4.11. A word \( f, |f| > 1 \) is in \( L \) if and only if it satisfies \( L_1, L_2, L_3, L_4, L_5 \).

The only if part is Proposition 4.8. To prove the if part we proceed by induction on \( |f| + |f|_y \).

The shortest words satisfying these conditions are \( xuv \) and \( xyv, xyv \) trivially they are in \( L \) as equal to \( xuv \cdot \mathbb{1}, xu \cdot \overline{a}, xu \cdot \overline{b}, xu \cdot \overline{a} \overline{b} \).

Let \( f \) be a word satisfying \( (L_1), ..., (L_5) \) such that \( |f| + |f|_y > 3 \) let \( f_1, f_2, f_3 \) be its \( L \)-decomposition.

If \( f_2 = \emptyset \) as \( f \) begins with an \( x \) by \( (L_2) \) then \( f_1 \neq \emptyset \), moreover \( f_3 \) cannot be empty as by \( L_4, f_1 = f'_1 xu \) and \( |f'_1 xu|_x = |f'_1|_x + 1 > |f'_1|_u + |f'_1|_y + 1 \), and this would contradict \( L_1 \). If \( f_1 = xu \), then as \( f_3 \) begins with \( v \), \( (L_2) \) would imply \( f_3 = v \) and \( |f|_x + |f|_y = 3 \), thus \( f_1 = f'_1 xu \) with \( |f'_1| \neq 0 \). Then either \( (A) \) or \( (B) \) holds for \( f \). By Proposition 4.10 there exists \( g \) satisfying \( L_3, L_4, L_5 \) such that \( g \cdot a \) or \( g \cdot b = f \); this \( g \) also satisfies \( L_1 \) and \( L_2 \). By induction there exists \( \alpha \) such that \( xu \cdot \alpha = g \) and \( xa \) or \( x \overline{a} \) will work for \( f \).

If \( f_2 \neq \emptyset \) then \( |f_2|_y + |f_3|_y + |f_3|_x \neq 0 \) by \( L_2 \), thus either \( |f_2|_y \neq 0 \) or \( f_3 \) is not empty; in the later case \( f_3 \) begins with \( v \) and \( f_2 \) ends with \( y \) giving also \( |f_2|_y \neq 0 \). Thus by Proposition 4.10 there exists \( g \) such that \( g \cdot \overline{a} = f \) or \( g \cdot \overline{b} = f \) and we can conclude in the same way as above.
COROLLARY 4.12. The number of words of $L \cap x^* u \{x, y\}^* v$ of length $2n + 1$ is the Catalan number $C_n$; it is also the number of words of $L \cap xu(x, y)^*$ of length $2n + 1$.

A word of $L \cap (x)^* u \{x, y\}^* v$ is obtained from a word of $\{x, y\}^*$ satisfying $(L_1)$ and $(L_2)$ replacing the first $y$ by $u$ and the last one by $v$. The corollary is then a consequence of Proposition 4.4.

For a word $f$ satisfying $L_1, \ldots, L_5$ there may be more than one $x$ such that $xuv \cdot x = f$, as many as the number of Baxter trees $(B, \varepsilon)$ such that $C(B, \varepsilon) = f$. We determine this number in the next part.

5. THE NUMBER OF BAXTER TREES HAVING THE SAME TREE-CODE: 
THE LEAF CODE

5.1. The Leaf Code of a Baxter Tree

Let $(B, \varepsilon)$ be a Baxter tree, $g_1, g_2, \ldots, g_k$ be the leaves of $B$ ordered by $\prec_s$, let us denote by $\Phi(B, \varepsilon) = \eta(g_1) \eta(g_2) \cdots \eta(g_k)$ (where $\eta$ is defined in 2.10). The word $\Phi(B, \varepsilon)$ is a subword of $\mathcal{F}(B, \varepsilon)$, it is an element of $\{1, 2, \ldots, n, u, v\}^*$ in which each $i$ ($1 \leq i \leq n$) appears at most once.

Let $f = \Phi(B, \varepsilon)$ be such a word, then $D(f)$ the leaf-code of $(B, \varepsilon)$ is defined recursively by:

**DEFINITION 5.1.**

- If $f$ is the empty word $D(\emptyset) = 1$.
- If $f$ does not contain any $u$ or $v$, let $f = f'jf''$, where $j$ is the greatest $i$ ($1 \leq i \leq n$) appearing in $f$ then
  $$D(f) = yD(f') \bar{y}D(f'')$$
- If $f$ contains some $u$ or $v$'s let $f = g_1f_1g_2f_2 \cdots f_pg_{p+1}$ where $g_i \in \{u, v\}^*$ and $f_i \{1, 2, \ldots, n\}^*$ then
  $$D(f) = g_1D(f_1) g_2D(f_2) \cdots D(f_p) g_{p+1}.$$  

**EXAMPLE 5.2.** For the Baxter tree in Fig. 2 the word $\Phi(B, \varepsilon)$ is $uu56uvv$ and its leaf code is $uuyy\bar{y}yuv$.  

**Remark 5.3.** If $f$ is a permutation then $D(f)$ is the parenthesis code of the decreasing binary tree associated to $f$ (see Remarks 6.5 and 6.6).

As for tree-codes the mapping $D \cdot \Phi$ induces operations in $\{y, \bar{y}, u, v\}^*$ from the graft and desactivation operations by the diagram:
More precisely we have:

**Definition 5.4.** Let $g$ be a word of $\{y, \tilde{y}, u, v\}^*$

- If $|g|_u \neq 0$ then $g \circ a$ is obtained from $g$ replacing the rightmost $u$ by $u$.
- If $|g|_u > 1$ then $g \circ b$ is obtained from $g$ replacing the rightmost $u$ by $\tilde{y}$ and adding $y$ after the penultimate,
- If $|g|_u \neq 0$ then $g \circ b$ is obtained from $g$ replacing the leftmost $v$ by $u$.
- If $|g|_u \neq 0$ and $|g|_v \neq 0$ then $g \circ a$ is obtained from $g$ replacing the leftmost $v$ by $\tilde{y}$ and adding $y$ after the rightmost $u$.

Remark that for a word $\alpha$ of $P_{a,b} \cup P_{b,\tilde{y}}$ then $uv \circ \alpha$ is well defined it is the same for the words satisfying the Condition (3') of Theorem 3.5.

Let us denote by $M$ the set of words $uw \circ \alpha$ the set of leaf-codes of Baxter trees. For $M$ we have a result of the same type as that for $L$:

**Proposition 5.5.** A word $g$ is an element of $M$ if and only if

1. $g = uh_1uh_2\cdots uh_pv_{p+1}\cdots v_{p+q}p > 0; q > 0$
2. The $h_i$ are parenthesis systems on $y, \tilde{y}$.

As for Proposition 4.8 it is easy to prove the only if part by induction. The only thing we have really to verify is that adding a "$\tilde{y}$" at the end of a parenthesis system and a "$y$" at the begin we obtain also a parenthesis system.

In order to prove the if part we have to introduce the $M$-decomposition of a word $g$ of $\{y, \tilde{y}, u, v\}^*$, such a decomposition $(g_1, g_2, g_3)$ verifies $g = g_1g_2g_3$, $g_1$ ends with $u$, $g_3$ begins with $v$ (or is empty) and $|g_2|_u = |g_2|_v = 0$. If $g$ verifies (1) and (2) then $g_2$ is $h_p$ of condition $M_1$. We proceed by induction on the length of $g$. Remark that $g = g'_1ug_2vg'_3$, if $g_3$ is not empty. If $g_2 = 1$ then $g'_1vg'_3$ and $g'_1vg'_3$ satisfies $M_1$ and $M_2$, moreover $g'_1vg'_3 \cdot a = g'_1vg'_3 \cdot b = g$.

Then the inductive hypothesis applied to $g'_1ug'_3$ or $g'_1vg'_3$ gives the existence of $\alpha_1$ and $\alpha_2$ such that $g'_1ug'_3 = uw \circ \alpha_1$ and $g'_1vg'_3 = uv \circ \alpha_2$. We then have $g = uw \circ \alpha_1a$ and $g = uv \circ \alpha_2b$. If $g_2 = 1$, $g'_3 = 1$ then only $g = uv \circ \alpha_1a$
CORI, DULUCQ, AND VIENNOT

holds. If \( g_2 \neq 1 \) then \( g_2 = yg'_2yg''_2 \) and the same construction is valid for \( g_1g'_2yg''_2g_3 \) and \( g_1g'_2yg''_2g_3 \) giving the result.

In order to compare \( xuw \cdot \alpha \) and \( xuv \cdot x \) we introduce the two morphisms \( \varphi \) and \( \psi \) of \{ x, y, u, v \} and \{ y, y, u, v \}, respectively, onto \{ y, u, v \} ; they are given by:

\[
\begin{align*}
\varphi(y) &= \psi(y) = y, \\
\varphi(u) &= \psi(u) = u, \\
\varphi(v) &= \psi(v) = v
\end{align*}
\]

Remark 5.6. If \( f \) and \( g \) are words such that \( \varphi(f) = \psi(g) \) then \( \varphi(f \cdot \alpha) = \psi(g \cdot x) \) for any \( x \) in \{ a, b, \bar{a}, \bar{b} \} \) (immediate from the Definitions 4.7 and 5.4).

Remark 5.7. If \( f \) and \( g \) are such that \( g(f) = \psi(g) \); \( (f_1, f_2, f_3) \) and \( (g_1, g_2, g_3) \) are the \( L \) and \( M \)-decompositions of \( f \) and \( g \), respectively, then \( \varphi(f_i) = \psi(g_i) \) for \( i = 1, 2, 3 \).

We are now able to prove the main result of Parts 4 and 5.

**Theorem 5.8.** A Baxter tree is uniquely determined by its tree-code and its leaf-code. Moreover let \( f \) and \( g \) be words of \{ x, y, u, v \} * and \{ y, y, u, v \}, respectively, then \( f \) is the tree-code of a Baxter tree \( (B, \epsilon) \) and \( g \) the leaf-code of \( (B, \epsilon) \) if and only if \( f \) is in \( L \), \( g \) is in \( M \) and \( \varphi(f) = \psi(g) \).

**Proof.** The definitions of \( L \) and \( M \) and Remark 5.6 give the only if part of the second assertion of this theorem. In order to prove the if part and the first assertion we will consider two words \( f \) and \( g \) satisfying \( (L_1) \cdots (L_5) \) for \( f \), \( (M_1)(M_2) \) for \( g \) and \( \varphi(f) = \psi(g) \) and we prove that there exists a unique \( \alpha \) such that \( xuw \cdot \alpha = f \) and \( uv \cdot x = g \). This will be done by induction on the length of \( g \). For \( |g| = 2 \) the only possibility is \( f = xuv \) and \( \alpha = \bar{1} \).

Let \( g \) be of length \( k + 1 \) and let \( (f_1, f_2, f_3) \) and \( (g_1, g_2, g_3) \) be the \( L \) and \( M \)-decompositions of \( f \) and \( g \). We will distinguish two cases according to \( f_2 \) is empty or not.

If \(|f_2| = 0\) then by Remark 5.7, \( g_2 \) is also empty, examining the proof of Theorem 4.11 and Proposition 4.10, we can conclude that there exists a unique \( f' \) and a unique \( \beta (\beta \in \{ a, b \}) \) such that \( f' \cdot \beta = f \). On the other hand, there may exist two words \( g_1 \) and \( g_2 \) such that \( g_1 \cdot a = g_2 \cdot b \). But \( |g_1|_u = |g|_u = |g_2|_u - 1 = |f|_u \), and as a consequence of these relations only one of \( g_1 \) and \( g_2 \) is such that \( |g|_u = |f'|_u \). Thus by construction of \( g', f' \) is such that \( g' \cdot \beta = g \) and \( \varphi(f') = \psi(g') \). By induction there exists a unique \( \alpha' \) such that \( xuw \cdot \alpha' = f' \) and \( uv \cdot x = g' \); thus \( \alpha = \alpha' \beta \) is the unique word such that \( xuw \cdot \alpha = f \) and \( uv \cdot x = g \).

If \( f_2 \) is not empty then \( g_2 \) is not empty and \( g_2 = yg'_2yg''_2 \) this decomposition of \( g_2 \) determines uniquely a decomposition of \( f_2 : f_2 = f'_2yf''_2 \) such
that $\varphi(f_2') = \psi(g_2')$ (or equivalently $|f_2'|_y = |g_2'|_y$). Denoting by $(f', g')$ one of the following couple $(f_1 f_2' u f_3', g_1 g_2' u g_3')$ or $(f_1 f_2' v f_3', g_1 g_2' v g_3')$ according to the end of $f_1 f_2'$ ($x$ or another letter), we verify that $(f', g')$ is the unique couple of words such that there exists a letter $c$ from $\{a, b, \bar{a}, \bar{b}\}$ verifying $(f' \cdot c, g' \cdot c) = (f, g)$. That letter is $\bar{b}$ in the first case, $\bar{a}$ in the second. As above considering $(f', g')$ and applying the inductive hypothesis give the result.

6. The Enumerative Results

**Definition 6.1.** Let $\alpha$ be a word in $P_{a \cup b}$ or in $P_{a \cup \bar{b}}$ (see notations after Theorem 3.5); let $(B, e) = (B_1, e_1) \cdot \alpha$ be the Baxter tree associated to $\alpha$ by the graft and desactivation operations of Section 3. Let $f = C(B, e)$ be the tree-code of $(B, e)$ and $g$ its leaf-code. Then $C(\alpha)$ denotes the word obtained from $f$ by deleting the last letter and replacing the $u$ appearing in this word by $y$. For $\alpha$ in $P_{a \cup b}$, $D(\alpha)$ denotes the word obtained from $g$ by deleting the first and last letter. For $\alpha$ in $P_{a \cup \bar{b}}$, $D(\alpha)$ denotes the word obtained from $g$ by deleting the first letter.

Remark that $C(\alpha)$ is in fact obtained from $C(B)$ by deleting the last letter it verifies $|C(\alpha)|_x = |C(\alpha)|_y$ and $|w'|_x \geq |w'|_y$ for any left factor $w'$ of $C(\alpha)$. Clearly $|C(\alpha)| = |\alpha| + 2$ if $\alpha \in P_{a \cup b}$ and $|C(\alpha)| = |\alpha| + 1$ if $\alpha \in P_{a \cup \bar{b}}$.

**Theorem 6.2.** The mapping $\alpha \rightarrow (C(\alpha), D(\alpha))$ is a bijection:

1. From $P_{a \cup b}$ onto the set of pairs $(w_1, w_2)$ of words of $P_{x, y} \times P_{y, \bar{y}}$ such that $|w_1| = |w_2| + 2$.
2. From $P_{a \cup \bar{b}}$ onto the set of couples of words $(w_1, w_2)$ of $P_{x, y} \times P_{y, \bar{y}}$ such that $|w_1| = |w_2|$.

**Proof.** That the mapping $\alpha \rightarrow (C(\alpha), D(\alpha))$ maps $P_{a \cup b}$ and $P_{a \cup \bar{b}}$ in the set described is an immediate consequence of Propositions 4.8 and 5.5. Conversely, let for instance $(w_1, w_2)$ be an element of $P_{x, y} \times P_{y, \bar{y}}$ such that $|w_1| = |w_2| + 2 = 2n$; consider the words $f = w_1' v$, where $w_1'$ is obtained from $w_1$ replacing the first $y$ by $u$ and $g = uw_1' v$ then $\varphi(f) = uy_1 u y_2 u \cdots y_n u y_{n+1} v y_{n+2} \cdots v y_1$. Thus by Theorem 5.8 there exists a unique Baxter tree $(B, e)$ having $f$ as a tree-code and $g$ as a leaf-code. Applying Theorem 4.3 gives (1). A similar proof holds for (2).

From Theorem 5.8 we obtain also:

**Proposition 6.3.** Given a word $f$ of $L$. The number of Baxter trees having $f$ as a tree-code is the product of the Catalan numbers $C_{n_1}, C_{n_2}, \ldots, C_{n_k}$, where $\varphi(f) = y_{n_1} y_{n_2} u \cdots y_{n_{k-1}} y_{n_k} + 1 v y_{n_{k+2}} \cdots v y_{n_k}$. 
Proof. It is easy to see that this is the number of words $g$ of $\{y, \bar{y}, u, v\}$ satisfying $M_1$ and $M_3$ and such that $\varphi(f) = \psi(g)$.

**Proposition 6.4.** The number $B_{2n+1}$ of elements of the sets (1), (2), (3) of Theorem 3.5 is the product of the two Catalan numbers $C_n$ and $C_{n+1}$. The number of elements $B_{2n}$ of the sets (1'), (2'), and (3') is the square of the Catalan number $C_n$.

**Proof.** By Theorem 3.5 $B_{2n+1}$ is the number of $P_{a,b}$ of length $2n$; by Theorem 6.2 it is also $C_n \cdot C_{n+1}$. Using these two theorems we obtain also $B_{2n} = C_n \cdot C_n$. Thus, Theorem 1.1 is proved: the number of alternating Baxter permutations on $2n+1$ (resp. $2n$) is $C_n C_n+1$ (resp. $C_n^2$), as announced in the Introduction.

**Remark 6.5.** Some of the constructions of this paper come from the (now classical) bijection between permutations and the so-called increasing binary trees.

A binary tree is a subset of $\{l, r\}^*$ satisfying the first condition $(A_1)$ of Definition 2.1. An increasing binary tree is a binary tree $b$ with a labeling $\varepsilon: b \to \mathbb{N}$ with distinct integers, satisfying conditions $(C_2)$ and $(C_3)$ of Definition 2.3. When the labels are the integers $1, 2, \ldots, n$, then the map $\mathcal{S}$ of Definition 2.10 is a bijection between increasing binary trees on $\{1, 2, \ldots, n\}$ and the $n!$ permutations on $\{1, 2, \ldots, n\}$ This bijection has been introduced by Foata and Schützenberger [5] and explicitly studied by Françon [7] and Viennot [11] (see also the survey paper Viennot [13]). For $n$ odd, the permutation $\mathcal{S}(b, \varepsilon)$ is alternating iff $b$ is a complete binary tree.

**Remark 6.6.** There exists a (very classical) bijection $\Delta: b \to B$, between binary trees $b$ with $n$ vertices and complete binary trees $B$ with $2n+1$ vertices. The tree $\Delta^{-1}(B)$ is obtained by deleting all the leaves from $B$.

Now, let $\alpha$ be a word of $P_{a, \omega} b$ and $(B, \varepsilon) = (B_1, \varepsilon_1) \alpha$ the associated Baxter tree. Let $\tau = \varepsilon(g_1) \cdots \varepsilon(g_k)$ be the word obtained by reading (in the order $<_s$) the leaves of $(B, \varepsilon)$ having a label different from $u$ or $v$. (Remark that the word $\Phi(B, \varepsilon)$ defined in V.1 is here simply $\Phi(B, \varepsilon) = utv$). By reversing the order of the integer labels in condition $(C_3)$ of Definition 2.3, one can define dually the notion of decreasing binary tree. Let $(b, \varphi)$ be the decreasing binary tree associated to the word $\varphi$. We consider the complete binary tree $\Delta(b)$ and its code $\beta = C(\Delta(b))$ associated by Definition 4.2. This word $\beta$ is nothing, but the word $D(\alpha)$ of Definition 6.1. Remark that the leaf-code of the Baxter tree $(B, \varepsilon)$ defined in 5.1 is the word $u\beta v$.

**Remark 6.7.** Let $\alpha$ be a word of $P_{a, \omega} b$ and $b$ and $B$ the binary trees defined in Remark 6.6.

An analog of Theorem 6.2 is to say that the map $\alpha \mapsto (B, b)$ is a bijection...
between words \( x \) of length \( 2n \) of \( P_{a \downarrow b} \) and pairs \((B, B')\), where \( B \) is a complete binary tree with \( 2n + 3 \) vertices and \( B' \) a binary tree with \( n \) vertices.

Example 6.8. Let \( x = ab\bar{a}b\bar{b}ba\bar{b} \) be a word of length \( 2n = 10 \) of \( P_{ab} \) (shuffle of the two parenthesis words \( a\bar{a}a\bar{a} \) and \( b\bar{b}b\bar{b}b\bar{b} \)). The Baxter tree \((B, \varepsilon) = (B_1, \varepsilon_1)\) obtained from \((B_1, \varepsilon_1) = (B, \varepsilon)\) by the graft and desactivation operations of Section 2 is displayed on Fig. 5.

The word \( \mathscr{S}(B, \varepsilon) \) associated by Definition 2.10 is \( \mathscr{S}(B, \varepsilon) = u2834176(11)9(10)5v. \)
The corresponding alternating (Baxter) permutation (on \(2n + 1 = 11\) elements) of Theorem 3.5 is \(\sigma = 2834176(11)9(10)5\).

The tree-code \(f = C(B, \varepsilon)\) of the Baxter tree \((B, \varepsilon)\) is \(f = xxuxyxyxyxyyv\). The word \(w_1 = C(\alpha)\) (of \(P_{x,y}\) of length \(2n + 2 = 12\)) obtained by Definition 6.1 is \(w_1 = xxuussyxxyyxxy\).

The word \(h = \Phi(B, \varepsilon)\) defined in 5.1 is \(h = u847(11)(10)v\). The leaf-code of \((B, \varepsilon)\) is \(g = D(h)\) (Definition 5.1) with \(g = uyyyyyyyuyuyu\). Then the word \(w_2 = D(\alpha)\) (of \(P_{y,\bar{y}}\) of length \(2n = 10\)) obtained by Definition 6.1 is \(w_2 = yyyyuyuyuyu\).

From Remark 6.6, the word \(\tau = 847(11)(10)\) and the corresponding decreasing binary tree \((b, \varphi)\) is displayed on Fig. 6.

REFERENCES