1 An example: binary trees and generating function

We begin with a classical combinatorial object, called binary tree, and displayed on Figure 1. A binary tree is, else reduced to a single vertex, or else is a triple formed by a root and a pair of binary trees (left and right subtree). Each vertex has else two sons (a left and a right son), else no sons. In the first case the vertex is called internal (resp. leaf or external) vertex. A binary tree has \( n \) internal vertices and \( n + 1 \) external vertices.

The number of binary trees with \( n \) (internal) vertices is the classical Catalan number \( C_n \). The sequence of Catalan numbers appears everywhere in combinatorics, and also in some parts of pure mathematics and in theoretical physics. The first values (for \( n = 0, 1, ... \)) are 1, 2, 5, 14, 42, 132, ... Historically, these numbers appear as the number of triangulations of regular polygons with \( n + 2 \) vertices, that is maximal configurations of two by two disjoints diagonals, as displayed on Figure 2. The problem of counting such triangulations go back to Euler, Segner in the 18th century, and around 1830-1840 to Binet, Lamé, Rodrigues, Catalan, and others.
A nice, simple and explicit formula for Catalan numbers is:

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

(1)

The variety of different proofs is typical of the evolution of combinatorics through the last two centuries. Classically, from the very definition of a binary tree, and the fact that the choice of the left and right subtree are independent, one would write the following recurrence relation for Catalan numbers, and which one should deduce formula (1).

\[ C_{n+1} = \sum_{i+j=n} C_i C_j, \quad C_0 = 1. \]  

(2)

Enumeration of binary trees is the typical situation in enumerative combinatorics: we have a class of "combinatorial objects of size n", that is a set \( A \), and a "size" function \( |\alpha| \) sending \( A \) on the integers \( N \), such that the set \( A_n \) of objects of size \( n \) is finite. The problem is to find a "formula" for \( a_n \), the number of elements of \( A_n \). A powerful tool in combinatorics is the notion of generating function (which is neither generating, neither a function), it is simply the formal power series whose coefficients are the numbers \( a_n \) of objects.

\[ f(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n + \cdots \]  

(3)

As for polynomials, we can define the sum and the product of two formal power series. In the case of Catalan numbers, recurrence (2) is equivalent to the following algebraic equation

\[ y = 1 + ty^2. \]  

(4)

In modern combinatorics, a standard lemma enables to go directly from the definition of binary trees to the equation (4). The philosophy is to define a kind of operations on combinatorial objects such as the sum and product, and consider equation (4) as the "projection" on the algebra of generating power series of the analog equation in the space of binary trees.

Historically, it took some times before generating functions in combinatorics where considered as formal power series, without consideration of convergence
in the real or complex domain. Nevertheless, there subsist a kind of formal convergence for power series. For example, infinite sum such as $1 + 1 + 1 + ...$ or $t + t + t + ...$ are nonsense. Formal power series in one variable, with coefficient in a ring $K$ (in practice, $K = \mathbb{Z}$ or $\mathbb{Q}$), form an algebra. Extensions are immediate with several variables. There is the notion of quasi-inverse such as $1/(1 - t) = 1 + t + t^2 + t^3 + ...$ and of a substitution $f(u(t))$ where $f$ and $u$ are two formal power series with $u(0) = 0$, i.e. $u(t)$ has no constant term. For example, the reader will check, by substituting $u = t + t^2$ in the power series $1/(1 - t)$, that the power series $1/(1 - t - t^2)$ is the generating function of the Fibonacci numbers $F_n$ defined by the following linear recurrence

$$F_{n+1} = F_n + F_{n-1}; \quad F_0 = F_1 = 1.$$  

Going back to Catalan numbers, from the algebraic equation (4), we get an explicit expression for their generating function

$$y = \frac{1 - (1 - 4t)^{1/2}}{2t}.$$  

Using the "binomial" formula $(1+u)^m = 1 + \frac{m}{1!}u + \frac{m(m-1)}{2!}u^2 + \frac{m(m-1)(m-2)}{3!}u^3 + ...$ for $m = 1/2$ and $u = -4t$, one get the formula (1) for Catalan numbers.

Generating function for Catalan numbers is the archetype of algebraic generating function, that is formal power series $y$ such that $P(y, t) = 0$, where $P$ is a polynomial in $y$ and $t$.

Going from the recursive definition of binary trees to the algebraic equation (4) is typical of modern enumerative combinatorics. But during the process, at the end, in the computation following relation (6), we are far from the combinatorics of binary trees. Another approach in the spirit of "bijective combinatorics" is to explain identities by the construction of bijections relating the objects interpreting each member of the equality. For example relation (1) is equivalent to the following identity

$$(n + 1)C_n = \binom{2n}{n}.$$  

A bijective proof of (7) can be obtained by constructing a bijection between binary trees having $n$ internal vertices with one of its leaf being distinguished, and an $n$ elements subset of a set having $2n$ elements. Another identity equivalent to (1) or (7) is

$$2(2n + 1)C_n = (n + 2)C_{n+1}$$  

and a completely different combinatorial construction will give a bijective proof. Such construction has been given by Rémy in term of binary trees and (surprisingly for the context of that times) by Rodrigues in term of triangulations. An interest of this bijection is to give a linear time algorithm constructing a random binary tree (with uniform distribution over all binary trees having a given number of vertices).
2 Algebricity and decomposable structures

Binary trees are the standard example of decomposable structure leading to an algebraic equation for the generating function. From the definition, the decomposition of binary into smaller binary trees is immediate. There are many examples of classes of combinatorial objects, having a nice algebraic generating function, but where the explanation of this algebricity by a recursive decomposition is far from evident. We give a typical and well known example with planar maps.

A planar map is an embedding on the sphere of a planar graph, up to homeomorphim. It may have loops and multiple edges. Counting planar maps is easier when one edge is selected, with an orientation on it. These are the so-called rooted planar maps (see Figure 3). The generating function $y$ for such objects counted according to the number of edges is algebraic and is solution of the following algebraic system of equations, as given by W.Tutte in the 60’s:

$$y = A - tA^3$$
$$A = 1 + 3tA^2$$

From equations (9), using standard tools such as Lagrange inversion formula, one get the following formula for the number $a_m$ of rooted planar maps with $m$ edges:

$$a_m = \frac{2 \cdot 3^m}{(n + 2)} C_m$$

Tutte method for proving (9) is indirect, with the use of some so-called "catalytic" extra variables and "kernel method". Many efforts have been given in order to "explain" directly the algebraic equations (9) and the formula (10), in the 70’s by Cori and Vauquelin with the introduction of "well labeled trees", in the 80’s by Arquès, until the "final" explanation by Schaeffer [2] in 1998, with the introduction of the "balanced blossoming trees". A blossoming tree is a binary tree, with the choice for each internal vertex of a bud in each of the three
possible regions delimited by the edges incident to that vertex (we have added an extra edge to the root of the binary tree, see Figure 4). The number of such trees is $3^n C_n$, satisfying obviously the second equation of the system (9). By a process connecting these $n$ buds with $n$ of the $n + 2$ external edges of the binary tree one get a planar map, where two external edges are left not connected (see Figure 5). The external root edge is not connected if and only if the blossoming tree is balanced. Then by connecting this root edge with the other external edge not connected, one get a rooted planar map where every vertex has degree 4. Such rooted maps with $n$ vertices are in bijection with planar rooted maps having $n$ edges. Finally Figure 6 explains visually the first algebraic equation of system (9) (following a proof by Bouttier, Di Francesco and Guitter, 2002).

Figure 4: Binary tree and blossoming tree.

Figure 5: Balanced and unbalanced blossoming tree.
3 Substitution in generating function

We have seen above how decompositions of combinatorial structures are related to some operations on generating functions such as sum and product. This is the philosophy of modern enumerative theory, considering some "operations" on combinatorial objects (sum, product, ..) which are the "lifting" of the analog operation in the algebra of formal power series. It would be possible to write standard abstract lemma for each operation. But at this level of exposition we prefer to work with examples. We will see in section 5 more examples related to the operation product and also the operation "quasi-inverse". In this section, we give an example of the operation "substitution" in power series and in combinatorial objects.

We define the Strahler number of a binary tree by the following recursive procedure. The leaves (external vertices) are labeled 0. If an internal vertex has two sons labeled $k$ and $k'$, then the label of that vertex is $\max(k, k')$ if $k \neq k'$, and $k + 1$ if $k = k'$ (see Figure 8). In a recursive way, every vertex is labeled and the label of the root is called the Strahler number of the binary tree (see Figure 7). This parameter was introduced in hydrogeology by Strahler, following Horton and has a long history in various sciences including fractal physics of ramified patterns, computer graphics, molecular biology and theoretical computer science (see the survey paper [6]).

We are interested in the enumeration of binary trees according to the parameter "Strahler number". Let $S_{n,k}$ be the number of binary trees having $n$
(internal) vertices and Strahler number $k$. Let $S_k(t)$ and $S(t, x)$ be the corresponding generating functions

$$S_k(t) = \sum_{n \geq 0} S_{n,k} t^n; \quad S(t, x) = \sum_{k \geq 0} S_k(t) x^k.$$  \hfill (11)

We are going to give an idea of the fact that this double generating function satisfies the following functional equation:

$$S(t, x) = 1 + \frac{xt}{1-2t} S \left( \frac{t}{1-2t}, x \right).$$  \hfill (12)

First we recall the (very classical) bijection between binary trees and Dyck paths. A Dyck path is displayed on Figure 9 and the bijection is obtained by following the vertices of the binary tree in the so-called "prefix order" and associating a North-East (resp. South-East) step in the Dyck path when one reaches an internal (resp. external) vertex of the binary tree.

The height $H(\omega)$ of a Dyck path is the maximum level of its vertices (here 4 on Figure 9). We define the "logarithmic height" $LH(\omega)$ of the Dyck path $\omega$ as the integer part of the logarithm in base 2 of the height, augmented by 1, of the Dyck path $\omega$. In other words, we have the following characterization

$$LH(\omega) = k \text{ iff } 2^{k-1} - 2 < H(\omega) \leq 2^k - 2.$$  \hfill (13)

A remarkable fact is that the distribution of the parameter $LH$ among Dyck paths of length $2n$ is exactly the same as the one of the Strahler number among
binary trees having \( n \) internal vertices. In other words, the corresponding double generating function for Dyck paths also satisfies the functional equation (12).

The proof relies on a bijective interpretation of this functional equation on both binary trees and Dyck paths (due to Françon [3]). We briefly resume the key ideas of this bijective proof. We need to introduce another family of binary trees (general binary trees), i.e. binary trees having four kind of vertices: no son, one left son, one right son, two sons. The family introduced at the beginning of this paper will be called complete binary trees to avoid confusion.

The second member of the equation (12) is obtained by replacing \( S(u, x) \) by \( uS(u^2, x) \) and then substituting \( u \) by \( t/(1-2t) \). The first substitution is interpreted by a bijection between general binary trees having \( n \) vertices and complete binary trees having a total of \( 2n+1 \) vertices (\( n \) internal and \( n+1 \) external). Such bijection is shown on Figure 10. This first substitution is also interpreted on Dyck paths by the bijection displayed on Figure 12. A Dyck path of length \( 2n \) is put in bijection with a "2-colored Motzkin path" of length \( n-1 \), that is a path having 4 kinds of elementary steps: North-East and South-East as for Dyck paths, together with elementary East step, colored blue or red.

Figure 9: Bijection between binary trees and Dyck paths.
The generating function \( t/(1 - 2t) \) corresponds to "zig-zag filaments" in binary trees (the pieces shown on Figure 11), or to sequences of level steps colored blue or red in the paths of Figure 12. The substitution of \( u \) into \( t/(1 - 2t) \)
in the double generating function $S(u, x)$ corresponds in the class of (complete) binary trees to "substitute" each of the $2n + 1$ vertices by a "zig-zag filaments" as shown on Figure 11. The complete binary tree becomes a bigger binary tree. The same substitution is also interpreted on Dyck paths: each of the $2n + 1$ vertices of a Dyck paths is "substituted" by a sequence of blue or red East steps staying at the same level as the level of the vertex of the Dyck path. In these constructions, each of the parameter Strahler number of the binary tree and logarithmic height of the Dyck path is increased by one.

These ideas, involving the idea of "substitution" inside combinatorial objects are at the basis of the proof that the parameters "Strahler number" and "logarithmic height" have the same distribution. A recursive bijection between binary trees and Dyck paths, preserving these parameters, could be deduced from these considerations (Françon [3]). A direct and deep bijection has just been obtained by D.Knuth [1]. The problem of computing the generating function $S_k(t)$ for binary trees having Strahler number $k$ is thus reduced to the one of finding the generating function for Dyck paths having a given height. It is a rational power series and will be given in the next section as a consequence of a general inversion property.

## 4 Rational generating function

A formal power series is rational if it has the following form:

$$\sum_{n \geq 0} a_n t^n = \frac{N(t)}{D(t)}$$

where $N(t)$ and $D(t)$ are polynomials in the variable $t$ with $D(0) \neq 0$.

We give here a general inversion theorem, which takes many forms in different domains: inversion of matrices in linear algebra, transition matrix in physics, finite automata generating function in theoretical computer science.

We define a path (or walk) in an arbitrary set $S$ as to be a sequence $\omega = (s_0, s_1, ..., s_n)$ where $s_i \in S$. The vertex $s_0$ (resp. $s_n$) is the starting (resp. ending) point, $n$ is the length of the path and $(s_i, s_{i+1})$ its elementary steps. We suppose that a function $v : S \times S \rightarrow K[X]$ is given, where $K[X]$ is the algebra of polynomials with coefficients in the ring $K$. The weight of the path $\omega$ is defined as to be the product of the weight of its elementary steps.

The following proposition gives the generating function for weighted paths in a finite set $S$.

**Proposition.** Let $i$ and $j$ be elements of the finite set $S$. The generating function for weighted paths $\omega$ starting in $i$ and ending in $j$ is rational and given by

$$\sum_{\omega, i \rightarrow j} v(\omega) = \frac{N_{ij}}{D}$$
\[ D = \sum_{\{\gamma_1, \ldots, \gamma_r\}, \text{2 by 2 disjoint cycles}} (-1)^r v(\gamma_1) \cdots v(\gamma_r) \]  

(16)

\[ N_{ij} = \sum_{\{\gamma_1, \ldots, \gamma_r\}} (-1)^r v(\eta) v(\gamma_1) \cdots v(\gamma_r). \]  

(17)

In \( D \) and in \( N_{ij} \), a cycle means a circular sequence of distinct vertices (as in the decomposition of a permutation into disjoint cycles). The weight of a cycle is the product of the weight of its oriented edges. In \( D \) the cycles \( \gamma_1, \ldots, \gamma_r \) are two by two disjoints; in \( N_{ij} \) the path \( \eta \) is a self-avoiding path and the cycles \( \gamma_1, \ldots, \gamma_r \) are two by two disjoints and disjoints of the path \( \eta \) (see Figure 13).

In fact, formula (15) is nothing but another form of the classical inversion formula of a matrix in linear algebra. If we define the matrix

\[ A = (A_{ij})_{1 \leq i, j \leq n} \quad \text{with} \quad v(i, j) = a_{ij} \]  

(18)

then the term \( ij \) of the inverse of the matrix \((I - A)\) is the sum of the weight of the paths \( \omega \) going from \( i \) to \( j \). The denominator \( D \) is the determinant of the matrix \((I - A)\), while \( N_{ij} \) is the cofactor \( ji \) of the same matrix.

**Example 1. Fibonacci numbers.**

We consider the segment graph of length \( n \). A matching is a collection of two by two disjoints edges \((i, i + 1)\) (see Figure 14). Such matchings are in bijection with paths on the set \( S = \{1, 2\} \) with weighted edges as displayed on Figure 15, and starting and ending at the vertex 1. Applying formula (15) gives the
generating function for Fibonacci numbers:

\[
\sum_{n \geq 0} F_n t^n = \frac{1}{1 - t - t^2}.
\]  (19)

Figure 14: Matching and Fibonacci numbers.

**Example 2. Bounded Dyck paths.**

We define \(F_k(t)\) as to be the alternating generating polynomial for matchings of the segment of the length \(n\), that is

\[
F_n(x) = \sum_{k=0}^{n} a_{n,k} (-x)^k.
\]  (20)

where \(a_{n,k}\) is the number of matchings of the segment graph of length \(n\) having \(k\) edges. The generating function for Dyck paths bounded by the height \(k\) is given by the following rational function:

\[
\sum_{\omega \text{ Dyck paths } H(\omega) \leq k} t^{|\omega|/2} = \frac{F_k(t)}{F_{k+1}(t)}.
\]  (21)

Figure 15: Bounded Dyck path.

A Dyck path is in bijection with a path on the segment \([0, k]\) starting and ending at the vertex 0, with elementary step \((i, i+1)\) and \((i, i-1)\). Equation (21) is just a consequence of the above proposition. The corresponding matrix \(A = (a_{ij})\) is a tridiagonal matrix with 0 as entries on the diagonal and \(t\) as entries above and under the diagonal.
The polynomials \( F_k(t) \) are usually called Fibonacci polynomials. Up to a change of variable, they are Tchebycheff polynomials of the second kind. Figure 16 gives the Fibonacci polynomial of order 4.

Combining equation (21) with section 3 gives the generating function \( S_{\leq k}(t) \) for binary trees with Strahler number \( \leq k \). Subtracting \( S_{\leq k}(t) \) and \( S_{\leq k-1}(t) \), and applying some trigonometric formula about modified Tchebycheff polynomials gives the following generating functions for binary trees with given Strahler number \( k \) (enumerated by number of vertices).

\[
S_k(t) = \frac{t^{(2^k-1)}}{R_{2^k-1}(t)} = S_{\leq k}(t) - S_{\leq k-1}(t) \tag{22}
\]

\[
S_{\leq k}(t) = \frac{R_{2^k-2}(t)}{R_{2^k-1}(t)}
\]

5 \textbf{q-series and q-analogues}

For various reasons, some generating functions are denoted with variable "q". In this section, we are in the garden of \( q \)-series, \( q \)-calculus and \( q \)-analogues, sometimes also called quantum combinatorics. A typical example is the generating function for partitions of integers. As in previous sections 1, 2, 3, this section will show the relationship between elementary operations on power series and the corresponding operations on combinatorial structures. Here we will illustrate a new operation: the operation "quasi-inverse" \( \frac{1}{1-u} \), corresponding to sequence of combinatorial objects.

A partition of the integer \( n \) is nothing but a decreasing sequence of non-zero integers (\( \lambda_1 \geq ... \geq \lambda_k \)) such that the sum \( \lambda_1 + ... + \lambda_k = n \). Such partition can be visualized with the so-called \textit{Ferrers diagram}, as shown on Figure 17. The \( i \)th row (from bottom to top) has \( \lambda_i \) cells.

We explicit the generating function for partitions of integers, or Ferrers diagrams. First, the generating function for a single row of length \( i \) is reduced to a single monomial \( q^i \). A rectangle with \( i \) columns is nothing but a sequence of rows of length \( i \). Now a general lemma says that if a combinatorial object
A has generating function \( u \), then a "sequence" of combinatorial object \( A \) has generating function \( \frac{1}{1-u} \). Thus rectangle with \( i \) columns has generating function \( \frac{1}{1-q^i} \). Any Ferrers diagram can be decomposed in a unique way into a family of rectangles (see Figure 18). Applying the "product" lemma as in sections 1 and 2, we get the generating function for Ferrers diagrams having at most \( m \) columns: \( \frac{1}{(1-q)(1-q^2)...(1-q^m)} \). Going to the limit, we get the generating function for partitions of \( n \) (or Ferrers diagram with \( n \) cells):

\[
\sum_{n \geq 0} a_n q^n = \prod_{i \geq 1} \frac{1}{(1-q^i)}
\]  

(23)

\textbf{q-analogues}

The generating function of Ferrers diagrams included in a \( m \times n \) rectangle is given by the following expression

\[
\frac{(1-q)(1-q^2)...(1-q^{m+n})}{(1-q)...(1-q^m)(1-q)...(1-q^n)}
\]  

(24)

If we take the formal variable \( q \) as to be a real number, in the limit \( q \to 1 \), the expression (24) tends to the binomial coefficient \( \binom{m+n}{m} \). The polynomial (24), also called Gaussian polynomial, is a \( q \)-analogue of the binomial coefficient. Ferrers diagrams included in a rectangle are defined by a path of length \( m+n \) with \( m \) elementary steps East and \( n \) elementary step South, and are thus enumerated by that binomial coefficient. The parameter \( q \) is the counting parameter for the area below the path. This is a typical situation of a combinatorial \( q \)-analogue. Another example is counting permutations on \( n \) elements according to the number of \textit{inversions}, which is given by the following polynomial

\[
(1+q)(1+q+q^2)...(1+q+...+q^{n-1}) = \frac{(1-q)(1-q^2)...(1-q^n)}{(1-q)^n}.
\]  

(25)
the philosophy of bijective proofs of identities.

Bijective proof of an identity

Already in the ancient greek times such proofs were given, as for example a "bijective" or "visual" proof of the identity $n^2 = 1 + 3 + \ldots + (2n - 1)$, displayed on Figure 19.

For $q = 1$, this polynomial gives back $n!$, the number of permutations.

For $q = 1$, this polynomial gives back $n!$, the number of permutations.

Bijective proof of an identity

Already in the ancient greek times such proofs were given, as for example a "bijective" or "visual" proof of the identity $n^2 = 1 + 3 + \ldots + (2n - 1)$, displayed on Figure 19.

In the same spirit as proving equation (23), we illustrate on Ferrers diagrams the philosophy of bijective proofs of identities.
We will prove the following identity

\[
\sum_{m \geq 1} \frac{q^{m^2}}{(1-q)(1-q^2) \cdots (1-q^m)}^2 = \prod_{i \geq 1} \frac{1}{1 - q^i}.
\]  

(26)

A bijective proof is displayed on Figure 20.

The left hand-side of identity (26) is a product of 3 generating functions. The numerator \(q^{m^2}\) is the generating for a Ferrers diagram having a square shape \(m \times m\). The denominator \(1/(1-q)\cdots(1-q^m)\) is the generating function for Ferrers diagrams having at most \(m\) columns (or by symmetry at most \(m\) rows). The right hand side of identity (26) is the generating function for (general) Ferrers diagrams. An arbitrary Ferrers \(F\) diagram can be decomposed in a unique way into a triple of Ferrers diagrams of the type described by the numerator and denominator in the left hand side of (24): it suffices to consider the largest square contained in the Ferrers diagram \(F\).

Such bijective proof, relating calculus and manipulation of combinatorial objects is typical of the so-called bijective paradigm. Identities coming from various part of mathematics can be treated this way. First one must find combinatorial interpretation of both sides of the identity which will appear as the sum of certain weighted objects. Then the identity will be seen as a consequence of the construction of a weight preserving bijection between the combinatorial objects interpreting both side of the identity. One possible interest is to give a better understanding of the identity. In the last years, many work has been done, in particular putting at the combinatorial level domain of mathematics such as special functions, orthogonal polynomials and continued fractions theory, elliptic functions, symmetric functions, representation theory of group and algebras,
Combinatorics is "plural", with various attributes such as enumerative, bijective or algebraic. We finish this lesson with a typical chapter of algebraic combinatorics by introducing heaps of pieces and an inversion lemma. The theory of heaps of pieces has been very useful for interaction with theoretical physics and will be explained in more details in our second lesson.

6 Algebraic combinatorics: an example with heaps of pieces

The theory of heaps of pieces (in French: "empilements de pièces") has been introduced by the author in 1985 [5] as a geometric and combinatorial interpretation of the so-called commutation monoids defined by Cartier and Foata in 1969 [4]. Commutation monoids have been used in computer science as models for parallelism and concurrency access to data structures. They are also called trace monoids after the pioneer work of Mazurkiewicz.

![Figure 21: Heaps of dimers on a chessboard.](image)

The intuitive idea of a heap of pieces can be visualized on Figure 21. Here the pieces are "dimers on a chessboard". They are put one by one on the chessboard, such that the projection of each dimer is the union of two consecutive cells. Then, intuitively, the heap is symbolized by what can be seen when one forgets in which order the dimers has been put. Here we will develop only the particular case of "heaps of dimers on the integers $\mathbb{N}$", as shown on Figure 22. A basic dimer is a pair $(i, i+1)$ of consecutive integers and a heap of dimers is formed by dimers, lying at a certain level $k \geq 0$ and having as projection on the horizontal
axis a basic dimer \((i, i+1)\). In that case, the commutation monoid is the monoid generated by the variables \(\sigma_0, \sigma_1, \ldots, \sigma_n\) with commutations of the form:

\[
\sigma_i \sigma_j = \sigma_j \sigma_i \quad \text{iff} \quad |i - j| \geq 2.
\] (27)

We briefly resume the basic idea relating commutation monoid and heaps of pieces with the example of Figure 22. We start with the word \(w = \sigma_2 \sigma_3 \sigma_5 \sigma_1 \sigma_4 \sigma_1 \sigma_3\). The columns above the horizontal axis are labeled \(\sigma_0, \ldots, \sigma_5\), corresponding to basic dimers \((0, 1), \ldots, (5, 6)\). Reading the word \(w\) from left to right, each letter \(\sigma_i\) produces a dimer falling down in the column labeled \(\sigma_i\) above the basic dimer \((i, i+1)\). The dimers fall down one by one, else on the “floor” (the horizontal axis at level 0), or onto another dimer which is in “concurrency” (i.e. the corresponding column are labeled \(\sigma_i\) and \(\sigma_j\) with \(|i - j| \leq 1\)). Two different words can leads to the same heap, as for example the words \(w\) and \(w' = \sigma_5 \sigma_2 \sigma_1 \sigma_3 \sigma_5 \sigma_4 \sigma_3\) giving the heap displayed on Figure 22. The fundamental lemma is that two words give the same heap if and only if they are in the same commutation class, that is if they can be transform from one to another by a sequence of elementary commutations of the form (27). Thus, we get a bijection between commutation class of the commutation monoid and the set of heaps of dimers over \(N\).

We suppose that each basic dimer \(\alpha = (i, i+1)\) is weighted by a certain polynomial \(v(\alpha)\) of the polynomial algebra \(K[X]\). We suppose that \(v(\alpha)\) has no constant term. In general, \(v(\alpha)\) will be a monomial. We define the weight \(v(E)\) of a heap of dimers \(E\) as to be the product of the weight of each pieces of the heap, the weight of a piece being the weight of its projection on the horizontal axis. We suppose that the generating fuction for weighted heaps, that is the infinite sum \(\sum_E v(E)\) (over all heaps \(E\)) makes sense.

A heap is called trivial when all its pieces are at level 0. We also suppose that the generating function for trivial heaps is a well defined formal power series. It will be a polynomial when the set of basic pieces is finite, as for example if we restrict the heaps of dimers on \(N\) to be on the segment \([0, n-1]\) of length \(n\). A
fundamental lemma, which would be considered in physics as a *boson-fermion identity*, is the following:

**Inversion lemma**

*The generating function for weighted heaps is the inverse of the alternating generating function for trivial heaps, that is*

\[
\sum_{E \text{ heap}} v(E) = \frac{1}{D} \quad \text{with} \quad D = \sum_{T \text{ trivial heap}} (-1)^{|T|} v(T) \quad (28)
\]

A typical example is given with heaps of dimers on the segment \([0, n - 1]\). If each basic dimer is weighted by the variable \(t\), then the generating function for weighted heaps is simply the generating function enumerating heaps of dimers according to the number of dimers. Form section 4, it is the inverse of the Fibonacci polynomial \(F_n(t)\).

![Figure 23: Inversion lemma.](image)

It will be very useful to use an extension of the inversion lemma. We define the *maximal pieces* of a heap as to be the pieces which can be "remove" from the heap by sliding them up without bumping another piece (dually *minimal pieces* would be the pieces lying at level 0). Let \(M\) be a set of basic pieces. We state the following extension of the inversion lemma:

**Extended inversion lemma**

*Let \(M\) be a set of basic pieces. The generating function for weighted heaps of pieces such that the projection of the maximal pieces are contained in \(M\) is*
given by the ratio:

\[
\sum_{E \text{ heap } = \{\text{max} \ E\} \subseteq M} v(E) = \frac{N}{D}.
\]  

(29)

where \(D\) (resp. \(N\)) is the alternating generating function of trivial heaps of pieces (resp. trivial heaps with pieces which are not in \(M\)).

If the set of basic pieces is finite, then this heaps generating function is rational. In particular, we will apply in the next lesson this extended inversion lemma for pyramids, that is heaps having a unique maximal piece.

The theory of heaps of pieces is particularly useful for the interaction between combinatorics and theoretical physics, in particular for solving in a pure combinatorial way some models from statistical physics such as: the directed animals model, hard gas model, stair-case polygons enumerated by perimeter and area, SOS (Solid-on-Solid) model and path with neighbour interactions. Heaps of pieces has also reappeared in 2D Lorentzian quantum gravity after work of Ambjørn, Loll, Di Francesco, Guitter, Kristjansen, James and the author.

Selected further reading


Another reference, with emphasis on the so called analytic combinatorics and analysis of algorithms in computer science is the book in preparation of P.Flajolet (717p. in April 2006) available on his web site (pauillac.inria.fr/algo/flajolet/Publications/book.html). The first part (PartA: Symbolic Methods, final version) with its 3 chapters is in the spirit of this Cargèse lesson. Finally, in this lesson we have completely put aside the vast domain of combinatorics dealing with exponential generating functions. A good exposition of exponential structures is the book of F.Bergeron, G.labelle and P.Leroux, “Combinatorial species and tree-like structures”, in Encyclopaedia of Mathematics and its applications, Cambridge University Press, 1997, 479p. A French version is available at the “Publication du LACIM”, LACIM, UQAM, Montral, 1996.

References


