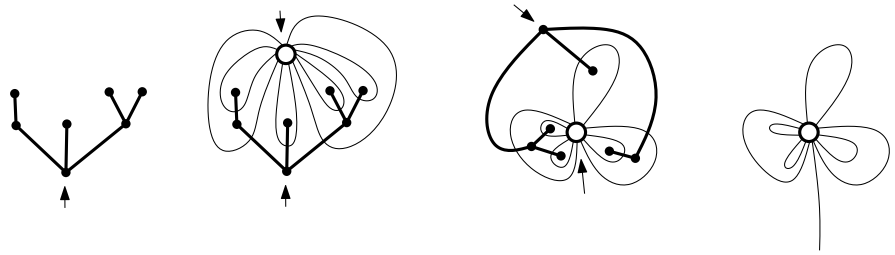

Figure 2.7: A rooted plane tree, its dual map, the same rotated, and the associated planted flower

The clockwise walk around a rooted plane tree. Consider a rooted planar tree (T, c) with n edges. The *clockwise walk* around (T, c) induces a total order $c_1 \prec c_2 \prec \cdots \prec c_{2n}$ on the $2n$ corners of T given by the order in which these corner are visited by a 2d little ant, starting at the root corner and travelling in clockwise direction on the border of the tree T.

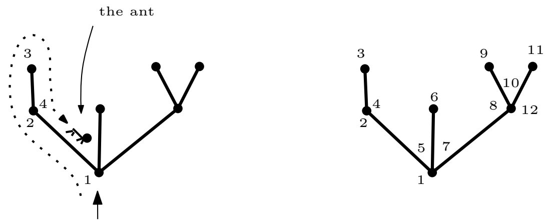

Figure 2.8: The first few corners visited by the clockwise walk, and the complete numbering according to the $\prec$ order.

More formally, the order $\prec$ is defined as follows: $c_1 = c$ is the root corner of T; for all $i \geq 2$, let e_i denote the edge that borders c in the clockwise direction around the vertex incident to c; then c_{i+1} is the opposite corner on the same side of e_i.

Claim 22 *The clockwise walk around (T, c) visits each corner once, and each edge twice: first away from the root, and then toward the root.*

The contour of a rooted plane tree The contour encoding of a rooted plane tree with n edges is the word of length $2n$ obtained along the clockwise walk around (T, c) upon:

- writing a letter u when an edge is visited for the first time
- writing a letter d when an edge is visited for the second time.

The contour walk is the one-dimensional walk associated to this word, as in Chapter 1.

Assuming that the clockwise walk is perfomed by an ant travelling at constant unit speed, the contour walk simply gives the distance to the root of the ant as a function of time. The reconstruction of a plane tree from its contour walk is also nicely describe graphically: cover the lower part of the Dyck path with glue and smash the path horizontally. Opposite sides get glued togeter and when the smashing is relaxed the result is a plane tree...

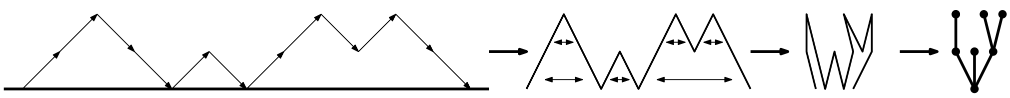

Figure 2.9: The contour of the tree of Figure 2.8 and the inverse "smashing"

Proposition 23 *The contour encoding is a bijection between*

- *rooted planar trees with n edges, and*
- *Dyck words on $\{u, d\}$ with $2n$ letters.*

Flowers, arch diagrams, chord diagrams and trees again Expand the only vertex of a flower into a small circle, cut at the root corner, and straighten the circle into a line. The result is a non-intersecting arch diagram with n arches.

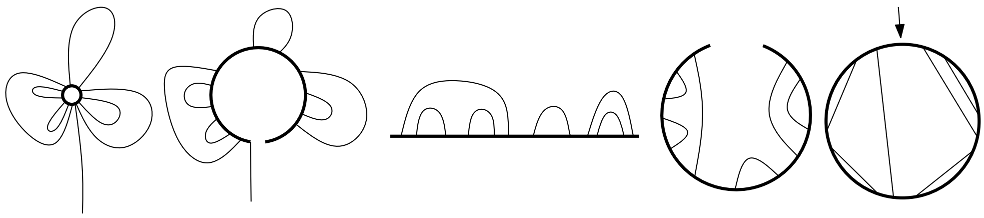

Figure 2.10: The arch and chord diagrams associated with a flower.

Close the arch diagram in a circle with arch inside, and straighten arches so that they become chords. The result is a rooted chord diagram with n chords. Observe that the flower and the chord diagram are the same when considered on the sphere. The only difference is the point at infinity used for the drawing in the plane: in the root face for the flower, inside the vertex for the chord diagram.

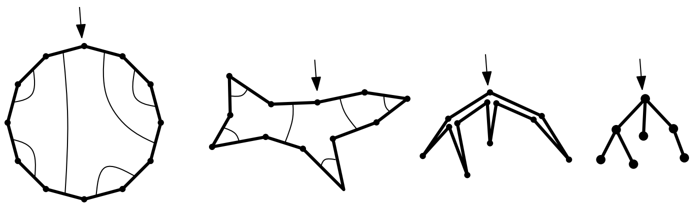

Figure 2.11: Folding a chord diagram into a tree.

From the chord diagram, one recover the tree by interpreting chords as strings connected opposite sides of a $2n$-gon. Upon smashing each string, the $2n$-gon folds into a tree.

Arch diagrams and balanced parenthesis words Reading from left to right the inital part of an arch can naturally be interpreted as an opening parenthesis and the end of the arch a closing one. Upon writing u for opening parenthesis and d for closing ones we recover a correspondance with Dyck words. However in order to have this construction commute with the previous ones, the arch diagram should be read from right to left. Equivalently, the third diagram of Figure 2.7 can be used to illustrate the fact that the clockwise walk around the tree crosses dual edges in counterclockwise order around the dual vertex.

Parentheses and algebraic expressions Upon giving names to vertices, the rightmost tree of Figure 2.11 is naturally associated to an expression

$$a(b(c(), d()), e(), f(g())).$$

Upon forgetting the name of symbols, we recover (the mirror image of) the Dyck code of the tree $((()())()(()))$. A classical alternative to the encoding of such an expression is the polnish notation:

Figure 2.12: Dfs degree encoding and bfs degree encoding of the same tree

if the arity of each symbol is known, parentheses are useless, and the expression can be recovered from the word

$$a_3 b_2 c_0 d_0 e_0 f_1 g_0$$

Because the arity of a vertex is its degree minus one, *except for the root vertex*, it is more convenient to state this encoding result for *planted* plane trees (*i.e.* with a root vertex of degree 1), rather that rooted ones.

This suggest the following procedure to encode a planted plane tree by its *depth-first-search degree code*:

- perform the cw walk around T and each time a non-root vertex is visited for the first time, write its degree.

The degree code of a planted plane tree is a word w on the alphabet $\mathcal{N} = \{1, 2, \ldots\}$. With δ defined by $\delta(i) = i - 2$ and extended additively on concatenations, this word satisfies $\delta(w) = 0$ (according to Proposition 18), and the Łukasiewicz property (as defined Section 1.8): any strict prefix w' of w is such that $\delta(w') > \delta(w)$. Indeed during the clockwise walk $\delta(w') + 1$ is always equal to the number of non-visited neighbors of already visited vertices: this quantity is positive until the visit of the last leaf.

Proposition 24 *For any finite sequence $(n_i)_{i \geq 1}$ such that $\sum_{i \geq 2}(i-2)n_i = n_1 - 2$, dfs degree encoding is a bijection between planted plane tree with n_i vertices of degree i and words on the alphabet $\{1, 2, \ldots\}$ with n_i letters i that satisfy the Łukasiewicz property for $\delta : i \to i - 2$.*

It should be noted that the degree code can be performed with any (deterministic) traversal policy: one could use for instance breadth first search instead of depth first search. In this case the quantity $\delta(w')$ is closely related to the width of the tree (the number of vertices at distance i of the root), since unvisited neighbors of visited vertices all lie almost at the same distance of the root at any time during bfs.

From degree codes to Dyck words A bijection between degree codes of length n and Dyck words of lenght $2n$ consists in replacing each letter i by a factor $u^{i-1}d$ and then removing the final letter d.

The reverse bijection is simply by iteratively factorizing the word from left to right using at each step the leftmost factor $u^i d$ available; there is no ambiguity since the language $\{u^i d, i \geq 0\}$ is a code: it does not contains a word which is a prefix of another word of the langage.

Degree codes on the alphabet $\{1, 3\}$ and binary trees Restricting our attention to the alphabet $\{1, 3\}$ is equivalent to considering trees with vertices of degree 1 and 3 only, that is, *planted binary trees.*

Proposition 25 *The degree encoding yields a bijection between planted binary trees with n inner nodes and words of length $2n + 1$ of the Dyck-Łukasiewicz language $\mathcal{D}d$.*

Dyck words therefore also encode binary trees. The direct relation between planted binary trees and rooted plane trees is given by the *left rotation correspondance*, illustrated by Figure 2.13. This correspondance can also be nicely described in terms of computer data structures for trees: a rooted plane tree is naturally described by attaching to each vertex a list of its children from left to right; but another natural data structure consists in giving to each non-root vertex of the tree, a (possibly `null`) *left* pointer to its leftmost child and a (possibly `null`) *right* pointer to the next sibbling in the list of its father: this second data structure is reinterpreted as encoding a binary tree, the left and right pointers giving left and right children of inner nodes and `null` pointers corresponding to leaves.

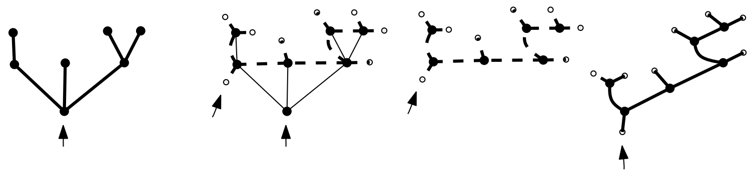

Figure 2.13: The rotation correspondance.

Binary trees and triangulations of a n-gon Duality applied to binary trees with n inner nodes and $n+2$ leaves yields a map with one vertex, n inner faces of degree 3 and $n+2$ faces of degree 1. This correspondance can be made nicer upon expanding each corner incident to faces of degree 1 into an edge, so as to form a $(n+2)$-gon, and redrawing the resulting map with a point inside the $(n+2)$-gon at infinity. The result is a triangulation of the $(n+2)$-gon with n triangles.

This transformation can now be extended to general trees with triangles replaced by general polygons.

Proposition 26 *There is a bijection between rooted plane trees with n edges, n_i vertices of degree $i \geq 2$ and k leaves, and dissections of a k-gon with n chords defining n_i faces of degree i.*

Observe that in this last statement, chords cannot be assumed to be straight, unless there is no vertex of degree 2 ($n_2 = 0$). Observe also that, as opposed to chords of chord diagrams, several chords of dissections may share the same endpoint.

Bipartite trees, bicolored chord diagrams and non-crossing partitions The vertices of a tree can be bicolored in a unique way such that the root is white. The above duality with chord diagrams allows to transfert this bicoloration into a bicoloration of the faces of a chord diagram. Observe that the $2n$ edges of the polygon alternatively border black and white face. Opening the circle at the middle of the white bordered edge incident to the root corner we get a bicolored arch diagram. Contracting now black bordered edges of the line this arch diagram is turned into a bicolored multi-arch diagram (arch are now allowed to share their extremity two by two). Upon numbering the vertices from 1 to n the black regions define a set partition of $\{1, 2, \ldots, n\}$. This partition is non-crossing, meaning that if $i \leq j \leq k \leq \ell$ then the pairs (i, k) and (j, ℓ) can be simultaneously be pairs of elements of a same part.

Proposition 27 *The above construction is a bijection between rooted plane trees with n edges and non-crossing partitions of $\{1, \ldots, n\}$.*

2.2.2 Enumeration in the garden

Enumeration, 1st try, Catalan The first properties we get is that all these combinatorial objects are counted by the same Catalan numbers as Dyck words.

Corollary 28 *The number of rooted plane trees with n edges is the Catalan number*

$$\frac{1}{n+1}\binom{2n}{n}$$

Mutandis mutandis, the same statement holds for all the other families we have introduced.