Notes on the Catalan problem

Daniele Paolo Scarpazza

Politecnico di Milano

Dipartimento di Elettronica e Informazione

Dottorato in Ingegneria dell'Informazione

XVIII Ciclo

Anno Accademico 2003-2004

Notes on the Catalan problem

Daniele Paolo Scarpazza
Politecnico di Milano
scarpazz@elet.polimi.it

March 16th, 2004

Abstract

This paper is about the Catalan numbers. The paper is organized as follows: section 1 presents a wide variety of problems which all have Catalan numbers as solution; section 2 presents traditional ways to derive expressions of the Catalan numbers in closed form; section 3 presents a novel interpretation of the problem, based upon formal languages considerations.

Materials of sections 1 and 2 are freely taken and adapted from [1]. Contents of section 3 are original. Considerations in section 4 present topics for future developments, and were suggested by Emanuale Munarini.

1 An overview of Catalan problems

The Catalan numbers appear as the solution of a variety of problems. They were first described in the 18th century by Leonhard Euler, when he was attempting to find a general formula to express the number of ways to divide a polygon with n sides into triangles using non-intersecting diagonals. The sequence is named after Eugene Catalan, a belgian mathematician which found their expression as the solution of the problem of finding how many ways one can parenthesize a chain of $n + 1$ letters using n pairs of parentheses such that there are either two letters, a parenthesized expression and a letter, or two parenthesized expressions within each pair of parentheses. In this section, we present a brief overview over a number of counting problems which all lead to the definition of Catalan numbers. The problem statements are enclosed in frames, and additional definitions and clarifications follow where necessary.

1.1 Balanced Parentheses

Determine the number of balanced strings of parentheses of length $2n$.

A string of parentheses is an ordered collection of symbols "(" and ")". A string of parentheses is said to be *balanced* when one of the following (equivalent) conditions is met:

- it has the same number of open and close parentheses and every prefix of the string has at least as many open parentheses as close parentheses;

```
proc checkstring string {
  set level 0
  for {set i 0} {$i < [llength $string]} {incr i} {
    if { [lindex $string $i]=="(" }  {
      incr level
    } else {
      if { [lindex $string $i]==")" } {
        incr level -1
        if {$level < 0} { return 0}
      }
    }
  }
  if {$level==0} {
    puts $string
    return 1
  } else {
    return 0
  }
}

proc recurse { N string } {
  if {$N == 0} {
    return [checkstring $string]
  } else {
    return [expr [recurse [expr $N -1] [concat $string ) ]] + \
           [recurse [expr $N -1] [concat $string ( ]]]
  }
}

for {set N 0} {$N <= 12} {incr N 2} {
  puts "$N: [recurse $N {} ]"
}
```

Figure 1: A simple Tcl program which enumerates all the strings of balanced parentheses of length between 0 and 12. Please note that variable N in the listing takes the values of $2n$

- for each closed parenthesis there is a matching open one, preceding it in the string; and for each open parenthesis there is a matching closed one, following it in the string;
- the string can be generated by a *van Dyck* formal grammar, namely a grammar with the following productions: $S \rightarrow (S)S|\epsilon$.

For example, string `()(()())` is balanced, whereas strings `)(())` and `(()()` are not.

It is a rather easy task to write a simple program which enumerates all the balanced strings of parentheses of any given length. A simple, recursive implementation, written in the Tcl language, is reported in Figure 1.1. A complete enumeration of all the balanced parenthesis strings of length $2n$, for n between 0 and 6, obtained by running the cited program, is reported in Figure 1.1.

Figure 2: Balanced strings of parentheses of length $2n$, for n between 0 and 6

n					C(n)
0	empty string				1
1	()				1
2	()()	(())			2
3	()()()	()(())	(())()	(()())	5
	((()))				
4	()()()()	()()(())	()(())()	()(()())	14
	()((()))	(())()()	(())(())	(()())()	
	(()()())	(()(()))	((()))()	((())())	
	((()()))	(((())))			
5	()()()()()	()()()(())	()()(())()	()()(()())	42
	()()((()))	()(())()()	()(())(())	()(()())()	
	()(()()())	()(()(()))	()((()))()	()((())())	
	()((()()))	()(((())))	(())()()()	(())()(())	
	(())(())()	(())(()())	(())((()))	(()())()()	
	(()())(())	(()()())()	(()()()())	(()()(()))	
	(()(()))()	(()(())())	(()(()()))	(()((())))	
	((()))()()	((()))(())	((())())()	((())()())	
	((())(()))	((()()))()	((()())())	((()()()))	
	((()(())))	(((())))()	(((()))())	(((())()))	
	(((()())))	((((()))))			
6	()()()()()()	()()()()(())	()()()(())()	()()()(()())	132
	()()()((()))	()()(())()()	()()(())(())	()()(()())()	
	()()(()()())	()()(()(()))	()()((()))()	()()((())())	
	()()((()()))	()()(((())))	()(())()()()	()(())()(())	
	()(())(())()	()(())(()())	()(())((()))	()(()())()()	
	()(()())(())	()(()()())()	()(()()()())	()(()()(()))	
	()(()(()))()	()(()(())())	()(()(()()))	()(()((())))	
	()((()))()()	()((()))(())	()((())())()	()((())()())	
	()((())(()))	()((()()))()	()((()())())	()((()()()))	
	()((()(())))	()(((())))()	()(((()))())	()(((())()))	
	()(((()())))	()((((()))))	(())()()()()	(())()()(())	
	(())()(())()	(())()(()())	(())()((()))	(())(())()()	
	(())(())(())	(())(()())()	(())(()()())	(())(()(()))	
	(())((()))()	(())((())())	(())((()()))	(())(((())))	
	(()())()()()	(()())()(())	(()())(())()	(()())(()())	
	(()())((()))	(()()())()()	(()()())(())	(()()()())()	
	(()()()()())	(()()()(()))	(()()(()))()	(()()(())())	
	(()()(()()))	(()()((())))	(()(()))()()	(()(()))(())	
	(()(())())()	(()(())()())	(()(())(()))	(()(()()))()	
	(()(()())())	(()(()()()))	(()(()(())))	(()((())))()	
	(()((()))())	(()((())()))	(()((()())))	(()(((()))))	
	((()))()()()	((()))()(())	((()))(())()	((()))(()())	
	((()))((()))	((())())()()	((())())(())	((())()())()	
	((())()()())	((())()(()))	((())(()))()	((())(())())	
	((())(()()))	((())((())))	((()()))()()	((()()))(())	
	((()())())()	((()())()())	((()())(()))	((()()()))()	
	((()()())())	((()()()()))	((()()(())))	((()(())))()	
	((()(()))())	((()(())()))	((()(()())))	((()((()))))	
	(((())))()()	(((())))(())	(((()))())()	(((()))()())	
	(((()))(()))	(((())()))()	(((())())())	(((())()()))	
	(((())(())))	(((()())))()	(((()()))())	(((()())()))	
	(((()()())))	(((()(()))))	((((()))))()	((((())))())	
	((((()))()))	((((())())))	((((()()))))	(((((())))))	

1.2 Mountain ranges

Determine the number of "mountain landscapes" which can be formed with n upstrokes and n downstrokes.

A mountain range is a polyline composed by segments of two types (upstrokes "/"and downstrokes "\"), such that its extreme points lie on the same horizontal line, and no segments cross that line. The possible mountain ranges for $0 \leq n \leq 3$ are shown in Table 1.2

n						C(n)
0	*					1
1	/\					1
2	/\/\	/\ / \				2
3	/\/\/\	/\ /\/ \	/\ / \/\	/\/\ / \	/\ / \ / \	5

Figure 3: Possible mountain ranges of length $2n$, with $0 \leq n \leq 3$.

1.3 Diagonal-avoiding paths on a lattice

Given a $n \times n$ lattice, determine the number of paths of length $2n$ which do not cross the diagonal.

Given a finite lattice composed by points $(i, j) : 1 \leq i \leq n, 1 \leq j \leq n$, a path on that lattice is a connected sequence of horizontal and vertical oriented line segments that connects node $(1, 1)$ to node (n, n), such that each segment is either a "west" or "south" segment, more formally each segment is either of type $(i, j) - (i + 1, j)$ or $(i, j) - (i, j + 1)$.

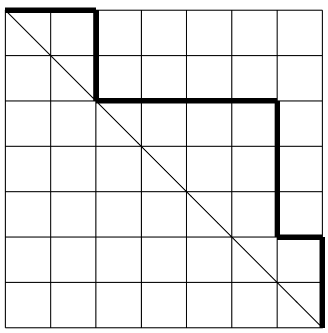

Figure 4: A sample diagonal-avoiding path in a 7×7 lattice, corresponding to the string of balanced parentheses (())((((()))()).

1.4 Multiplication precedence

Determine the number of ways in which $n+1$ factors can be multiplied together, according to the precedence of multiplications.

Examples with $0 \leq n \leq 4$ are given in Figure 1.4.

n		C(n)
0	a	1
1	$a \cdot b$	1
2	$(a \cdot b) \cdot c$ $a \cdot (b \cdot c)$	2
3	$((a \cdot b) \cdot c) \cdot d$ $(a \cdot b) \cdot (c \cdot d)$ $(a \cdot (b \cdot c)) \cdot d$ $a \cdot ((b \cdot c) \cdot d)$ $a \cdot (b \cdot (c \cdot d))$	5
4	$(((a \cdot b) \cdot c) \cdot d) \cdot e$ $((a \cdot b) \cdot c) \cdot (d \cdot e)$ $((a \cdot b) \cdot (c \cdot d)) \cdot e$ $(a \cdot b) \cdot ((c \cdot d) \cdot e)$ $(a \cdot b) \cdot (c \cdot (d \cdot e))$ $((a \cdot (b \cdot c)) \cdot d) \cdot e$ $(a \cdot (b \cdot c)) \cdot (d \cdot e)$ $(a \cdot ((b \cdot c) \cdot d)) \cdot e$ $(a \cdot (b \cdot (c \cdot d))) \cdot e$ $a \cdot (((b \cdot c) \cdot d) \cdot e)$ $a \cdot ((b \cdot c) \cdot (d \cdot e))$ $a \cdot ((b \cdot (c \cdot d)) \cdot e)$ $a \cdot (b \cdot ((c \cdot d) \cdot e))$ $a \cdot (b \cdot (c \cdot (d \cdot e)))$	14

Figure 5: Possible multiplication precedences in expressions of $n+1$ factors, with $0 \leq n \leq 4$.

1.5 Regular polygon triangulation

Determine the number of ways in which a regular polygon with $n+2$ edges can be triangulated.

Triangulating a polygon consists in dividing it in triangles by connecting couples of non-adjacent vertices via segments which do not cross each other. Triangulations of regular polygons with $3, 4, 5$ and 6 vertices are illustrated in Figure 6.

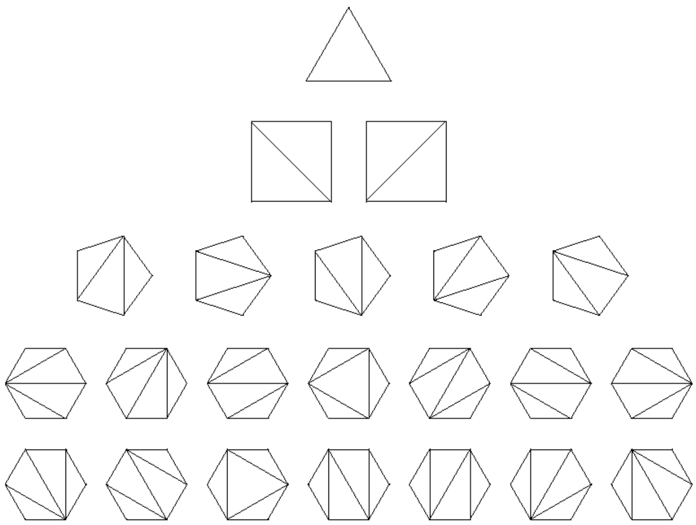

Figure 6: Possible polygon triangulations, with $1 \leq n \leq 4$.

1.6 Handshakes across a table

Determine the number of ways in which $2n$ people sitting around a table can shake hands without crossing their arms.

The possible hand-shaking configurations for $n = 1, 2, 3, 4$ are shown in Figure 7.

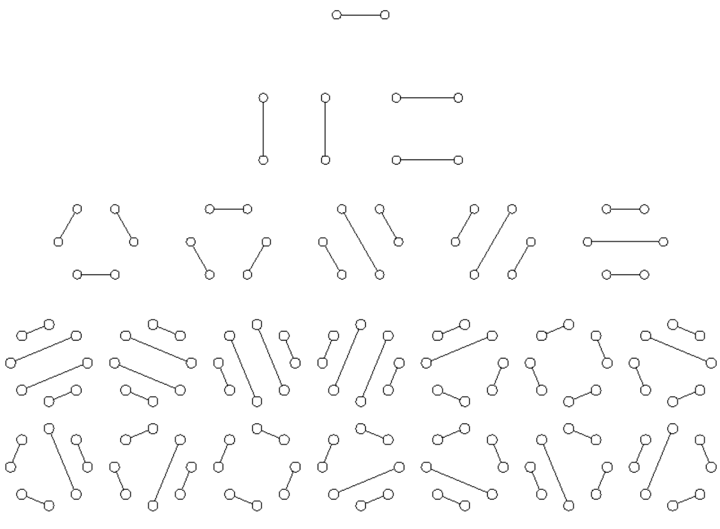

Figure 7: Possible handshakes across a table, with $1 \leq n \leq 4$.

1.7 Binary rooted trees

Determine the number of binary rooted trees with n internal nodes.

Given a rooted tree, each non-leaf node is defined as an *internal node*. Binary rooted trees with n internal nodes and n ranging from 0 to 3 are shown in Figure 8.

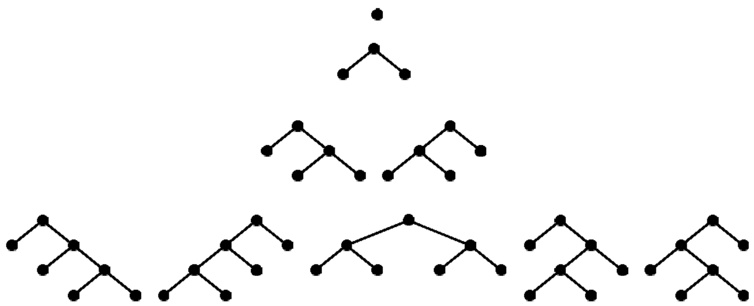

Figure 8: All the binary rooted trees with n internal nodes, with $0 \leq n \leq 3$.

1.8 Plane trees

Determine the number of plane trees with n edges.

A plane tree is such that it is possible to draw it on a plane without having edges crossing each other.

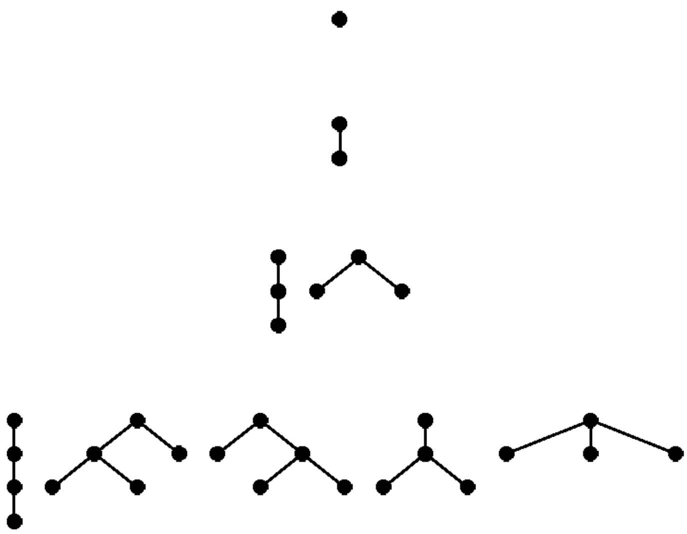

Figure 9: Plane trees with n edges, with $0 \leq n \leq 3$.

1.9 Skew Polyominos

Determine the number of skew polyominos with perimeter $2n + 2$.

A polyomino is a figure composed by squares connected by their edges. A

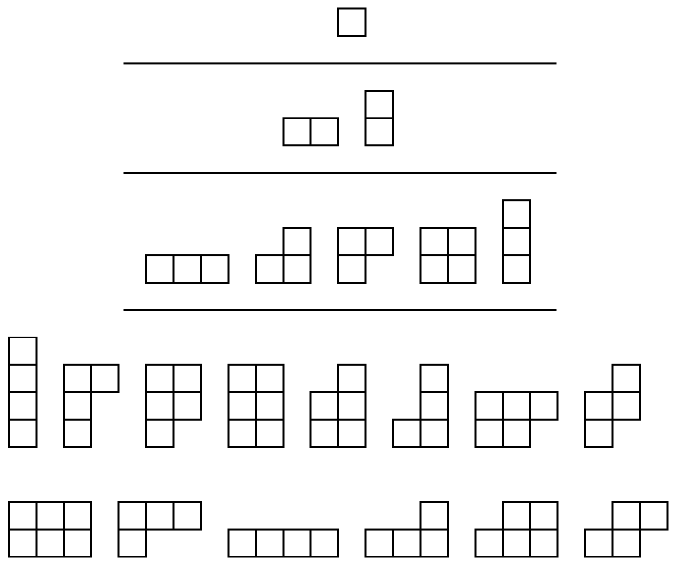

Figure 10: Skew polyominos with perimeter $2n + 2$, with $1 \leq n \leq 4$.

skew polyomino is a polyomino such that every vertical and horizontal line hits a connected set of squares and such that the successive columns of squares from left to right increase in height–the bottom of the column to the left is always lower or equal to the bottom of the column to the right. Similarly, the top of the column to the left is always lower than or equal to the top of the column to the right. Figure 10 shows all skew polyominos with perimeter $2n + 2, 1 \leq n \leq 4$.

2 Derivations of the Catalan numbers

2.1 Catalan numbers derived with generating functions

In this section we report a traditional derivation of the closed-form expression of the Catalan numbers, obtained by setting recurrence relations and solving them with generating functions.

Recalling the first example of Catalan problems (i.e., balanced strings of parentheses) we observe that any such string of length $n > 0$ begins with an open parentesis, and this open parenthesis matches a closed one, located somewhere in the remainder of the string. It is therefore possible to state that the two cited parentheses partition the rest of the string in two substrings A and B:

$$(A)B$$

and if A contains k pairs of parentheses, B must contain exactly $n-k-1$, such that the number of parenthesis pairs in $(A)B$ is $k+(n-k-1)+1 = n$. Therefore the number of balanced strings having n parenthesis pairs is the count of all the configurations in which A is empty and B contains $n-1$ pairs, plus the number of configurations in which A contains 1 pair and B contains $n-2$, and so on, as expressed by:

$$\begin{aligned}
C_1 &= C_0C_0 \\
C_2 &= C_0C_1 + C_1C_0 \\
C_3 &= C_0C_2 + C_1C_1 + C_2C_0 \\
C_4 &= C_0C_3 + C_1C_2 + C_2C_1 + C_3C_0 \\
... &= ...
\end{aligned}$$

which can be rewritten in the form of the following recurrence relations:

$$\begin{aligned}
C_0 &= 1 \\
C_1 &= 1 \\
C_n &= \sum_{i=0}^{n-1} C_i C_{n-1-i}
\end{aligned}$$

We will now solve the above recurrences with the use of generating functions. Let's start by writing the generating function $C(x)$ corresponding to the Catalan number succession:

$$C(x) = C_0 + C_1 \cdot x + C_2 \cdot x^2 + ... = \sum_{i=0}^{+\infty} C_i \cdot x^i$$

Let's now examine the expression of $[C(x)]^2 = C(x)C(x)$, as follows:

$$\begin{array}{ccccccc}
C_0C_0 & + & (C_0C_1 + C_1C_0) & x & +(C_0C_2 + C_1C_1 + C_2C_0) & x^2 + ... = \\
|| & & || & & || & \\
C_1 & + & C_2 & x\ + & C_3 & x^2 + ...
\end{array}$$

That is, the square of the Catalan generating function is still a generating function with Catalan coefficients, shifted one position left:

$$[C(x)]^2 = C_1 + C_2 x + C_3 x^2 = \sum_{i=0}^{+\infty} C_{i+1} x^i \quad .$$

Therefore if we multiply the whole series by x and add C_0, the original Catalan series is obtained:

$$C(x) = C_0 + x[C(x)]^2. \tag{1}$$

The above formula is a quadratic equation, which could be put into the more familiar form:

$$xC^2 - C + C_0 = 0,$$

where C (was $C(x)$) is the unknown and x, C_0 are constant coefficients. Replacing C_0 with its value (i.e., 1), the solution is trivially given by:

$$C = \frac{1 \pm \sqrt{1-4x}}{2x}.$$

Of the two solutions given by the $\pm$ sign, only the $-$ is acceptable, being $C_0 = 1$:

$$C = \frac{1 - \sqrt{1-4x}}{2x}. \tag{2}$$

The solution contains the power of a binomial with fractional exponent: $\sqrt{1-4x} = (1-4x)^{1/2} = \sum_{n\geq 0} \binom{1/2}{n}(-4x)^n$, which can be expanded as:

$$\begin{aligned}(1-4x)^{1/2} &= 1 - \frac{1/2}{1}4x + \frac{(1/2)(-1/2)}{2\cdot 1}(4x)^2 + \\ &+ \frac{(1/2)(-1/2)(-3/2)}{3\cdot 2\cdot 1}(4x)^3 + \\ &+ \frac{(1/2)(-1/2)(-3/2)(-5/2)}{4\cdot 3\cdot 2\cdot 1}(4x)^4 + ...\end{aligned}$$

which can be simplified as follows:

$$\begin{aligned}(1-4x)^{1/2} &= 1 - \frac{1}{1!}2x + \frac{1}{2!}4x^2 + \\ &- \frac{3\cdot 1}{3!}8x^3 + \frac{5\cdot 3\cdot 1}{4!}16x^4 + ...\end{aligned} \tag{3}$$

Now, substituting equation 3 in 2, we obtain:

$$\begin{aligned}C(x) &= 1 - \frac{1}{2!}2x + \frac{3\cdot 1}{3!}4x^2 + \\ &+ \frac{5\cdot 3\cdot 1}{4!}8x^3 + \frac{7\cdot 5\cdot 3\cdot 1}{5!}16x^4 + ...\end{aligned} \tag{4}$$

We can get rid of terms like $7 \cdot 5 \cdot 3 \cdot 1$ (factorials missing the even factors), by considering that:

$$\begin{aligned} 2^2 \cdot 2! &= 4 \cdot 2 \\ 2^3 \cdot 3! &= 6 \cdot 4 \cdot 2 \\ 2^4 \cdot 4! &= 8 \cdot 6 \cdot 4 \cdot 2 \\ ... &= ... \\ 2^n \cdot n! &= \prod_{i=1}^{n} 2i \end{aligned}$$

Consequently:

$$\begin{aligned} C(x) &= 1 + \frac{1}{2}(\frac{2!}{1!1!})x + \frac{1}{3}(\frac{4!}{2!2!})x^2 + \frac{1}{4}(\frac{6!}{3!3!})x^2 = \\ &= \sum_{i=0}^{+\infty} \frac{1}{1+i}\binom{2i}{i} z^i \end{aligned}$$

Therefore, the i^{th} Catalan number is:

$$C_i = \frac{1}{1+i}\binom{2i}{i}.$$

2.2 The simplest proof

The simplest derivation of a closed form for Catalan numbers known to the author is suggested by considerations on the problem of diagonal-avoiding paths on a lattice, described in § 1.3. Determining the n-th Catalan number equals counting the total number of paths through the grid and then subtracting the number of invalid paths, namely the ones which cross the diagonal.

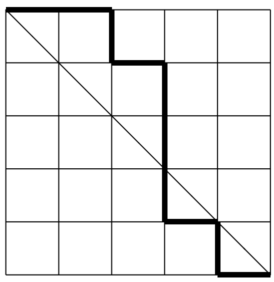

Figure 11: An invalid (diagonal-crossing) path in a 5×5 lattice.

Figure 11 illustrates an example of an invalid path, which crosses the diagonal. Invalid paths may cross the diagonal multiple times, but for each invalid path there always exist a first time in which the diagonal is crossed, more formally a down-oriented segment of coordinates $(i, i) - (i, i + 1)$, for some i. The point of coordinates $(i, i + 1)$ is the first illegal reached point, and will be referred as P from now on. Now, let's transform each invalid path according to the following rule: starting from point P on, we will replace each "south" segment with a "west" segment and vice versa. Figure 12 illustrates the result of transforming the invalid path depicted in Figure 11 according to the above rules.

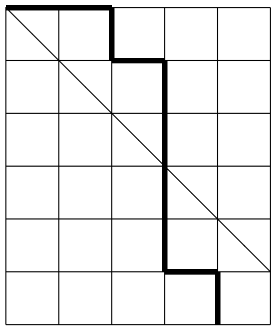

Figure 12: The transformed path.

Please note that, since the transformation starts at point $(i, i+1)$ (i.e. after i "west" and $i+1$ "south" segments), and it causes the remaining $n-i$ "west" segments to be replaced by "south" segments and the remaining $n-i-1$ "south" segments to be replaced by "west" segments, it can be proved that the new ending coordinates will be $(i+(n-i-1),(i+1)+(n-i))$, that is, $(n-1, n+1)$.

Therefore it is correct to say that the above transformation turns a $n \times n$ illegal path into a $(n-1) \times (n+1)$ path. By construction of the transformation rule, every illegal path in the $n \times n$ lattice corresponds to exactly one non-constrained path in $(n-1) \times (n+1)$ lattice.

Since the number of paths in a $a \times b$ lattice is $\binom{a+b}{a}$, the total number of non-constrained paths through the $n \times n$ lattice is $\binom{2n}{n}$, whereas the total number of invalid paths in the same $n \times n$ lattice is equal to the number of paths in a $(n-1) \times (n+1)$, that is $\binom{2n}{n+1}$, therefore the total number of non diagonal-crossing paths is given by:

$$C_n = \binom{2n}{n} - \binom{2n}{n+1} = \binom{2n}{n} - \frac{n}{n+1}\binom{2n}{n} = \frac{1}{n+1}\binom{2n}{n}.$$

3 A novel interpretation

In this section we will give a novel calculation of the Catalan numbers, inspired by formal language considerations. The contents of this section are, to the best of our knowledge, original.

The language of balanced parentheses is a well-known and studied example in the field of formal languages, under the name of Van Dyck languages. In the case of a single type of parentheses (i.e., `(`, `)`, no square, curly or angular brackets), the grammar which generates the above language is given as follows:

$$
\begin{aligned}
G &= (\Sigma, N, S, R) \\
\Sigma &= \{(,)\} \\
N &= \{S\} \\
R &= \{r_1, r_2\} \\
r_1 &: S \to \varepsilon \\
r_2 &: S \to (S)S
\end{aligned}
$$

The above formulae define a grammar with an alphabet composed by `(` and `)`, a single non-terminal symbol, called S, which is also the start symbol (a.k.a. the *axiom*) for the grammar, and two rules, r_1 and r_2.

Given a grammar, a *sentential form* is a sequence of terminal and nonterminal symbols which can be derived from the start symbol S. Strings belonging to the language generated by a given grammar are special sentential forms, composed of terminal symbols only.

Each sentential form contains a number n of nonterminal symbols and t terminal symbols. We will label such a sentential form with a (n, t) label. Consequently, strings $\in \mathcal{L}(G)$ will be labeled $(0, 2i), i \in \mathbf{N}$. It is easy to prove that the number of terminal symbols is even: the axiom, S contains no terminal symbol (0 is even), and both r_1 and r_2 rules preserve parity.

Let's now define a *derivation step* as a substitution which replaces a single S symbol in a sentential form with the right-hand side of rule r_1 or r_2. If a sentential form containing a certain number of terminal and non-terminal symbols (thus labelled (n, t)) is rewritten, its derived form will:

- have one nonterminal symbol less than the original, and the same number of terminals, if rule r_1 was applied;
- have one more non-terminal symbols more and two terminal symbols more than the original form, if rule r_2 was applied.

This is graphically expressed in figure 13.

Let's now consider a given sentential form, with label (n, t), and determine all the possible labels of sentential forms which could have derived it. There are only two:

- an $(n + 1, t)$ form, from which our form derived through rule r_1; and
- an $(n - 1, t - 2)$ form, from which our form derived through rule r_2.

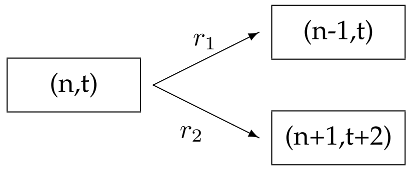

Figure 13: A balanced $(n, t)-$sentential form can derive an $(n-1, t)-$sentential form via rule r_1, or a $(n+1, t+2)-$sentential form through rule r_2.

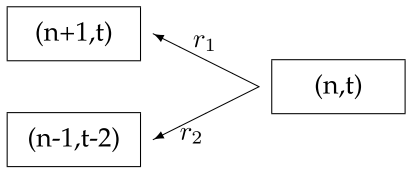

Figure 14: An $(n, t)-$sentential form could have been derived either from an $(n+1, t)-$form via rule r_1 or from an $(n-1, t-2)-$form via rule r_2.

The above considerations are summarized in Figure 14.

Therefore, given the label of a sentential form, figure 13 indicates how to find the labels of the sentential forms of its derivation-predecessors. Applying recursively the same algorithm on them, it is possible to trace back up to the axiom, which has obviously label $(1, 0)$. Please note that, due to the simplicity of our grammar (more precisely to the fact that N contains the only element S), it is trivial to prove that the axiom can only have label $(1, 0)$ and, conversely, label $(1, 0)$ can correspond to the axiom only.

Figure 14 seems to suggests that each sentential form has exactly two derivation predecessors. This is not true, in particular:

- a $(1, 0)$ form has no predecessor by definition, being the axiom;
- $(0, t)$ and $(1, t)$ forms can only have a $(n+1, t)$ predecessor. (Proof: by contradiction, the $(n-1, t-2)$ predecessor would have zero or less non-terminals, therefore it could have no successors.)
- (n, t) forms with $n > t$ do not exist, apart from the axiom $(1, 0)$. (Proof: by induction. For each form (n, t) let's consider the $t - n$ difference. Given a form f having this difference $= \delta$, both after the application of rule r_1 and r_2, the new value of this difference is $\delta + 1$. Therefore, the axiom has $\delta = -1$,but after the first derivation $\delta = 0$, and at each derivation, δ is incremented.)

According to Figure 14 and the above constraints, the whole derivation tree for any given sentential form can be drawn. Derivation trees for the strings $(0, 2)$, $(0, 4)$, $(0, 6)$ and $(0, 8)$ are shown in Figures 15 and following. Axiom nodes are marked with an exclamation mark "!", whereas invalid nodes are marked with a "$\times$" sign.

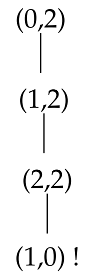

Figure 15: Derivation tree for $(0, 2)$, i.e., $2n = 2$

Please note that each derivation tree, built according to the above rules, starts with a label corresponding to a language string, $(0, 2i)$, and reaches leaf nodes which are either axioms or invalid nodes. The number of axioms contained in the tree of a given $(0, 2i)$-string is the number of different ways in which the axiom can derive a $(0, 2i)$-string, therefore the number of different strings of balanced parentheses of length $2i$ since each derivation is unique.

The reader is kindly asked to count axiom nodes in the each of the reported $(0, 2i)$-derivation trees, and verify that the count is the i^{th} Catalan number. According to the above rules, the i^{th} Catalan number can also be derived by counting the number of axiom nodes in the derivation tree of a $(0, t)$ form, with $t = 2i$, as expressed by the following recurrence relation:

$$R(n,t) = \begin{cases} 1 & \text{if } (n = 1 \wedge t = 0) \\ 0 & \text{if } n > t \\ R(n+1,t) & \text{if } n <= 1 \\ R(n-1,t-2) + R(n+1,t) & \text{otherwise.} \end{cases}$$

By construction,

$$C_i = R(0, 2i).$$

The above relation R was implemented as recursive Tcl function (shown in Figure 3) and tested for correctness.

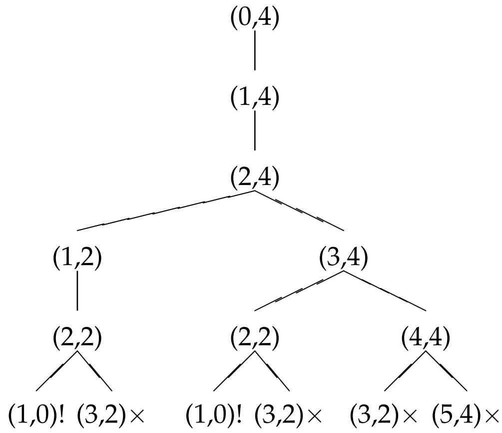

Figure 16: Derivation tree for $(0,4)$, i.e., $2n = 4$

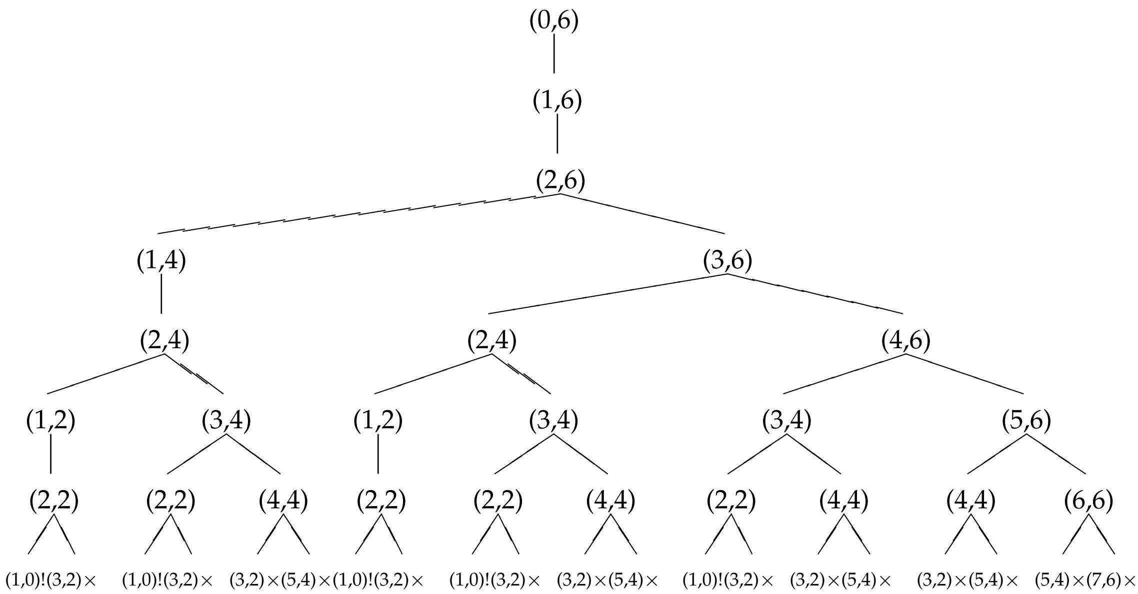

Figure 17: Derivation tree for $(0,6)$, i.e., $2n = 6$

```
(0,8
 +--(1,8
 |   +--(2,8
 |   |   +--(1,6
 |   |   |   +--(2,6
 |   |   |   |   +--(1,4
 |   |   |   |   |   +--(2,4
 |   |   |   |   |   |   +--(1,2
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   +--) 2
 |   |   |   |   +--) 2
 |   |   |   |   +--(3,6
 |   |   |   |   |   +--(2,4
 |   |   |   |   |   |   +--(1,2
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   +--) 2
 |   |   |   |   |   +--(4,6
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(5,6
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   |   +--(6,6
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 0
 |   |   |   |   |   +--) 1
 |   |   |   |   +--) 3
 |   |   |   +--) 5
 |   |   +--) 5
 |   |   +--(3,8
 |   |   |   +--(2,6
 |   |   |   |   +--(1,4
 |   |   |   |   |   +--(2,4
 |   |   |   |   |   |   +--(1,2
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   +--) 2
 |   |   |   |   +--) 2
 |   |   |   |   +--(3,6
 |   |   |   |   |   +--(2,4
 |   |   |   |   |   |   +--(1,2
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   +--) 2
 |   |   |   |   |   +--(4,6
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(5,6
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   |   +--(6,6
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 0
 |   |   |   |   |   +--) 1
 |   |   |   |   +--) 3
 |   |   |   +--) 5
 |   |   |   +--(4,8
 |   |   |   |   +--(3,6
 |   |   |   |   |   +--(2,4
 |   |   |   |   |   |   +--(1,2
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   +--) 2
 |   |   |   |   |   +--(4,6
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(5,6
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   |   +--(6,6
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 0
 |   |   |   |   |   +--) 1
 |   |   |   |   +--) 3
 |   |   |   |   +--(5,8
 |   |   |   |   |   +--(4,6
 |   |   |   |   |   |   +--(3,4
 |   |   |   |   |   |   |   +--(2,2
 |   |   |   |   |   |   |   |   +--(1,0) 1
 |   |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 1
 |   |   |   |   |   |   +--(5,6
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   |   +--(6,6
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 0
 |   |   |   |   |   +--) 1
 |   |   |   |   |   +--(6,8
 |   |   |   |   |   |   +--(5,6
 |   |   |   |   |   |   |   +--(4,4
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   |   +--(6,6
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--(7,8
 |   |   |   |   |   |   |   +--(6,6
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   |   +--(8,8
 |   |   |   |   |   |   |   +--) 0
 |   |   |   |   |   |   +--) 0
 |   |   |   |   |   +--) 0
 |   |   |   |   +--) 1
 |   |   |   +--) 4
 |   |   +--) 9
 |   +--) 14
 +--) 14
) 14
```

Figure 18: Derivation tree for $(0, 8)$, i.e., $2n = 8$. For lack of space, the tree structure was represented in a column form and illegal nodes were not reported.

Figure 19: A simple Tcl program which implements the recursive function $R(n, t)$.

```
proc R { n t } {
  if {$n == 1 && $t == 0} { return 1 }
  if {$n > $t }           { return 0 }
  if {$n <= 1 }           { return [R [expr $n+1] $t] }

  return [expr [R [expr $n-1] [expr $t-2] ] \
             + [R [expr $n+1] $t          ] ]
}
```

Figure 20: The first values of $R(n, t)$

t/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	5	5	5	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	14	14	14	9	4	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	42	42	42	28	14	5	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	132	132	132	90	48	20	6	1	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	429	429	429	297	165	75	27	7	1	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1430	1430	1430	1001	572	275	110	35	8	1	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	4862	4862	4862	3432	2002	1001	429	154	44	9	1	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	16796	16796	16796	11934	7072	3640	1638	637	208	54	10	1	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	58786	58786	58786	41990	25194	13260	6188	2548	910	273	65	11	1	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	208012	208012	208012	149226	90440	48450	23256	9996	3808	1260	350	77	12	1	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	742900	742900	742900	534888	326876	177650	87210	38760	15504	5508	1700	440	90	13	1	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	2674440	2674440	2674440	1931540	1188640	653752	326876	149226	62016	23256	7752	2244	544	104	14	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	9694845	9694845	9694845	7020405	4345965	2414425	1225785	572033	245157	95931	33915	10659	2907	663	119	15	1	0

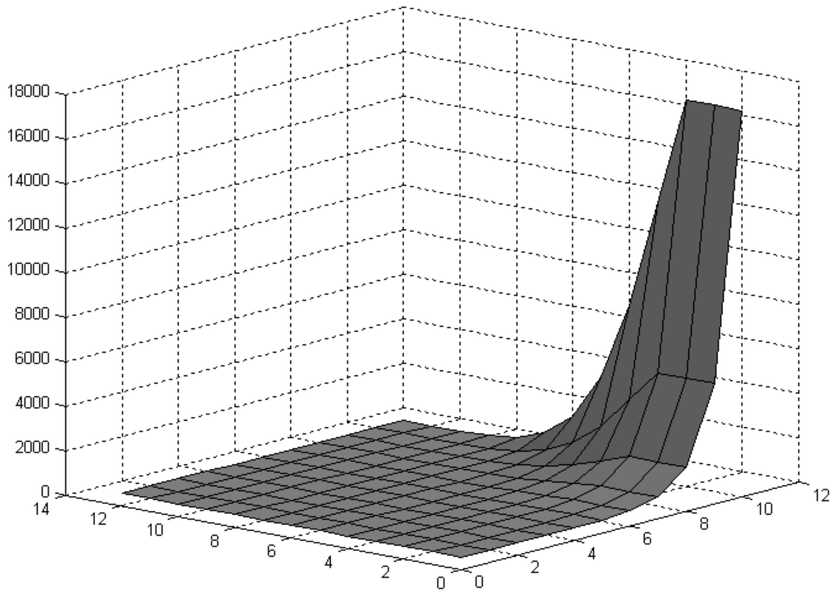

Figure 21: Plot of the surface $R(n,t)$. Points with odd values of t were not plotted.

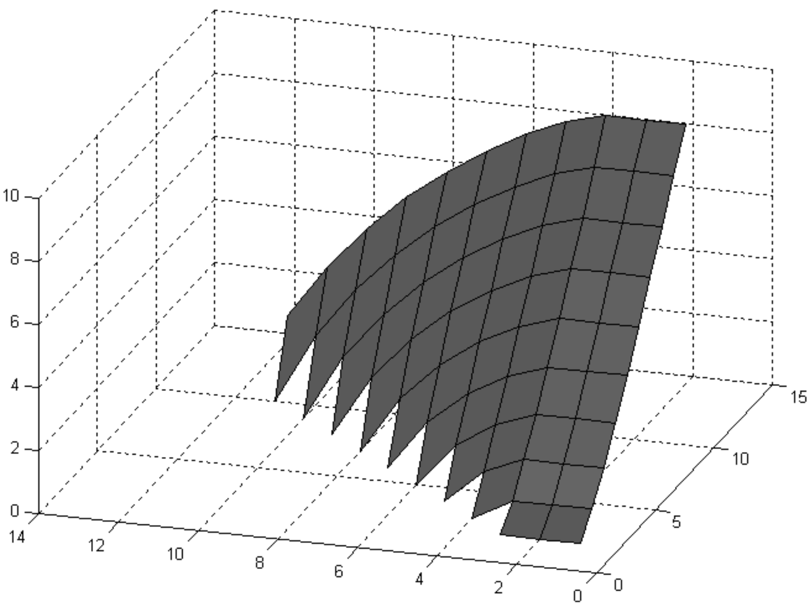

Figure 22: Plot of the surface $\log R(n,t)$. Points with odd values of t were not plotted.

4 Future Developments

This section contains a number of considerations based upon language enumeration techniques, which pose a first basis for the achievement of a closed form for R.

If D is the language defined by the production:

$$S \rightarrow \varepsilon \mid (S)S,$$

then D can be recursively written as:

$$D = \{\varepsilon\} \ + \ (D)D \tag{5}$$

where the $+$ symbol denotes the operation of disjoint set union.

If an alphabet $A = \{a_1, a_2, ..., a_m\}$ is considered, A^* denotes the language of all the strings over alphabet A, and if $\alpha \in A^*$ is a string over A, $|\alpha|$ denotes the length of α, i.e. the count of symbols which compose α. For a language A^*, a function $w : A^* \rightarrow \mathbb{Z}[\![x]\!]$ can be defined in the following manner:

$$w(\alpha) = x^{|\alpha|}, \text{with } \alpha \in A^*$$

and we set by convention that $w(\varepsilon) = 1$.

It is trivial to prove that function w exhibits the following property:

$$\forall \alpha, \beta \in A^*, w(\alpha \cdot \beta) = w(\alpha)w(\beta),$$

where the $\cdot$ symbol denotes the operation of string concatenation. w can be extended on languages, by defining $w(L) = \sum_{\alpha \in L} w(\alpha)$. Therefore

$$w(A^*) = \sum_{\alpha \in A^*} w(\alpha) = \sum_{\alpha \in A^*} x^{|\alpha|} = \sum_{n \geq 0} \left(\sum_{|\alpha|=n} 1 \right) x^n = \sum_{n \geq 0} m^n x^n = \frac{1}{1 - mx}.$$

As a consequence of 5, the following equation can be set for language D:

$$w(D) = 1 + x^2 w(D)^2$$

which is closely related to equation 1 and can be solved similarly, by replacing $y = w(D)$, thus obtaining $y = 1 + x^2y^2$, which can be put into a more familiar form as $x^2y^2 - y + 1 = 0$. The solution is given by:

$$y = \frac{1 - \sqrt{1 - 4x^2}}{2x^2} \tag{6}$$

By applying the same techniques shown in §2.1, it can be shown that:

$$D(x) = w(D) = D_0 + D_2x^2 + D_4x^4 + D_6x^6 + ...$$

where $\forall i \in \mathbb{N}, D_{2i} = C_i$ and $D_{2i+1} = 0$, therefore

$$D(x) = w(D) = C_0 + C_1x^2 + C_2x^4 + C_3x^6 + ...$$

Our aim is now to extend the above considerations to the language of all the *sentential forms* of grammar G, introduced in §3, that we will call E from now on. As we showed, each string in D corresponds to exactly one mountain range (as described in §1.2); it is worth noting that each string in E corresponds to exactly one *extended* mountain range, where not only upstrokes and downstrokes are allowed, but also horizontal strokes. For example, the following string is a string of N: $(S)(S)((S)(S(S)))S$, and its corresponding extended mountain range representation follows:

```
             _
        _  _/ \
 _  _  / \/    \
/ \/ \/         \_
```

The language E of sentential forms of grammar G is a new language, can be defined by the a new grammar H, given as follows:

$$
\begin{aligned}
H &= (\Sigma', N, B, R') \\
\Sigma' &= \{(,), \mathtt{S}\} \\
N &= \{B\} \\
R &= \{r_1, r_2, r_3\} \\
r_1 &: B \to \varepsilon \\
r_2 &: B \to \mathtt{S}B \\
r_3 &: B \to (B)B
\end{aligned}
$$

Please note that S is a terminal symbol for grammar H. The above language is called the *Motzkin* language [3]. As we did with equation 5, considering the production:

$$B \to \varepsilon \mid SB \mid (B)B,$$

we give now a recursive definition of E:

$$E = \{\varepsilon\} \ + \ SE \ + \ (E)E. \tag{7}$$

It is then time to introduce a newer, more useful definition of $w(\alpha)$:

$$w(\alpha) = x^{p(\alpha)} y^{o(\alpha)} z^{|\alpha|}$$

where $p(\alpha) = |\alpha|_(+ |\alpha|_)$, and $o(\alpha) = |\alpha|_S$, therefore $p(\alpha) + o(\alpha) = |\alpha|$. From the recursive definition of E given in 7, as done for 7, it is possible to set the following equation:

$$w(E) = 1 + yzw(E) + x^2 z^2 w(E)^2,$$

which, replacing $e = w(E)$, is:

$$x^2 z^2 e^2 + (yz - 1)e + 1 = 0,$$

which, solved by e yields:

$$e = \frac{1 - yz - \sqrt{1 - 2yz + y^2 z^2 - 4x^2 z^2}}{2x^2 z^2}$$

Thus $e(x,y,z)$ can be written as a formal power series with coefficients $E_{i,j,k}$:

$$\begin{aligned} e(x,y,z) &= E_{0,0,0} + \\ &+ E_{1,0,0}x + E_{0,1,0}y + E_{0,0,1}z + \\ &+ E_{2,0,0}x^2 + E_{0,2,0}y^2 + E_{0,0,2}z^2 + E_{1,1,0}xy + E_{0,1,1}yz + E_{1,0,1}xz + \\ &+ \ldots \end{aligned}$$

It is now evident that the number of sentential forms with n nonterminals and t terminals, previously called $R(n,t)$ is given by :

$$R(n,t) = E_{n,t,n+t} = [x^t y^n]e(x,y,1),$$

where the notation [...] has the following meaning $[x^n]f(x) = f_n \Leftrightarrow f(x) = \sum_{n\geq 0} f_n x^n$, in particular $[x^i y^j z^k]e(x,y,z) = E_{i,j,k}$. Furthermore, an expression of $e(x,y,1)$ can be obtained by restriction:

$$e(x,y,1) = e(x,y,z)|_{z=1} = \frac{1-y-\sqrt{1-2y+y^2-4x^2}}{2x^2} \tag{8}$$

Incidentally, the i-th Catalan number, which was equal to $R(0,2i)$ can be obtained by setting $y=0$, thus:

$$e(x,0,1) = e(x,y,1)|_{y=0} = \frac{1-\sqrt{1-4x^2}}{2x^2},$$

which is identical to equation 6 and admits the same solutions.

To obtain an expression of $R(i,j)$, in equation 8 we can collect $(1-y)$ in the numerator and $(1-y)^2$ in the denominator, thus obtaining:

$$e = \frac{1-y}{(1-y)^2}\frac{1-\sqrt{1-4\frac{x^2}{(1-y)^2}}}{\frac{2x^2}{(1-y)^2}} = \frac{1}{1-y}\left.\frac{1-\sqrt{1-4q^2}}{2q^2}\right|_{q=\frac{x}{1-y}},$$

which can be solved by comparison with equation 6, thus:

$$e = \frac{1}{1-y}\,D\left(\frac{x}{1-y}\right)$$

but since

$$D(q) = \frac{1-\sqrt{1-4q^2}}{2q^2} = \sum_{k\geq 0} D_k q^k$$

then

$$\begin{aligned} e = \frac{1}{1-y}\sum_{k\geq 0} D_k \frac{x^k}{(1-y)^k} &= \sum_{k\geq 0} D_k \frac{x^k}{(1-y)^{k+1}} \\ &= \sum_{k\geq 0} D_k x^k \sum_{n\geq 0}\binom{n+k}{k} y^n \\ &= \sum_{n,k\geq 0}\binom{n+k}{k} D_k x^k y^n \end{aligned}$$

therefore it should be true that:

$$R(n,k) = \begin{cases} 0 & \text{if } k \text{ odd} \\ \binom{n+k}{k} C_{k/2} & \text{if } k \text{ even} \end{cases}$$

Unfortunately the above equation was sperimentally verified to be incorrect. The cause is due to the fact that the real language of the sentential forms of E is smaller than $\mathcal{L}(H)$; proof: $S, SS, SSS, ...S(), S(S), ... \in \mathcal{L}(H) - E$, thus the above equations need to be rewritten.

Our current efforts are devoted to finding a new correct and unambiguous grammar for language E, obtaining an appropriate recursive definition of E and a corresponding equation for $w(E)$, which, solved, would yield a formal power series for $w(E)$, thus, a closed form for $R(n,t)$ numbers.

Bibliography

1. Tom Davis, *Catalan Numbers*, `http://www.geometer.org/mathcircles`, 2001.

2. Kenneth P. Bogart, *Combinatorics through Guided Discovery*, 2002.

3. Robert Stoyan, Volker Strehl, *Enumeration of Hamiltonian Circuits in Rectangular Grids*, 1994.

4. Eric W. Weisstein, *Catalan Number*, From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/CatalanNumber.html`