New York Journal of Mathematics

New York J. Math. **7** (2001) 223–231.

The Umbral Transfer-Matrix Method, III: Counting Animals

Doron Zeilberger

Abstract. This is the third part of the five-part saga on the umbral transfer-matrix method, based on Gian-Carlo Rota's seminal notion of the umbra. In this article we describe the Maple package ZOO that for any specific k, automatically constructs an umbral scheme for enumerating "k-board" lattice animals (polyominoes) on the two-dimensional square lattice. Such umbral schemes enable counting these important classes of animals in polynomial time as opposed to the exponential time that is required for counting all animals.

Contents

Received December 27, 2000. Revised November 7, 2001.
Mathematics Subject Classification. 05A15, 05A40, 82-04.
Key words and phrases. Lattice animals, polyominoes, umbral calculus, functional equations.
Supported in part by the NSF. I thank the referee for numerous corrections and improvements.

ISSN 1076-9803/01

Required reading

The reader is assumed to be familiar with [Z1]. It would also help to read the part of [Z0] concerned with finding generating functions for counting lattice animals with vertical cross-sections of bounded width. The present treatment is inspired by the finite case described in detail in [Z0], but with the umbral twist, getting matrices whose entries are *operators* rather than mere polynomials, as *transfer matrices.*

The alphabet

Recall that a *square-lattice animal* (from now on *animal*) is a *connected* subset of lattice points in the two-dimensional square lattice, up to translation-equivalence. For example,

$$A := \{(0,0),(0,1),(1,1),(1,2),(1,3),(0,3),(2,2)\}$$

is (a representative) of an animal, but $\{(0,0),(0,1),(1,2),(1,3),(0,3)\}$ is not, since the latter has two connected components.

We first need a convenient way to represent finite sets of integers, up to translation-equivalence. If we have a 'continuous segment' $\{i, i+1, \ldots, i+j-1\}$, we will denote it by $[j]$, and say that this is a 1-board set. Otherwise, we can partition the set into a union of such maximal boards, interspersed with gaps. If there are two boards (and hence one gap), we will represent this by a triple $[A_1, B_1, A_2]$ which represents a finite set of integers consisting of A_1 consecutive integers, followed by B_1 consecutive integers that *do not* belong to it, followed again by A_2 consecutive integers that *do* belong to it. If the smallest member is 0 then $[A_1, B_1, A_2]$ is the set

$$\{0,1,2,3,\ldots,A_1-1, A_1+B_1, A_1+B_1+1,\ldots,A_1+B_1+A_2-1\}.$$

In general, the set of integers of type $[A_1, B_1, A_2, \ldots, B_{k-1}, A_k]$ whose smallest member is 0, is the set

$$\begin{aligned}\{0,1,2,\ldots,A_1-1, A_1+B_1, A_1+B_1+1,\ldots,A_1+B_1+A_2-1,\ldots,\\ A_1+B_1+A_2+B_2+\cdots+A_{k-1}+B_{k-1}, A_1+B_1+A_2+B_2+\cdots+A_{k-1}+B_{k-1}+1,\ldots,\\ A_1+B_1+A_2+B_2+\cdots+A_{k-1}+B_{k-1}+A_k-1\}.\end{aligned}$$

For example the set of integers $\{0,1,2,5,6,7,10,11,14,15,16\}$ is represented, up to translation, by $[3,2,3,2,2,2,3]$, and has 4 boards (of lengths $3,3,2,3$) interspersed by 3 gaps (of lengths $2,2,2$). For a k-board set of integers, we will call the $k-1$ gaps between the boards *genuine* gaps. In addition we have two additional 'infinite gaps', the bottom one, from $-\infty$ to -1, and the top one, from $A_1+B_1+A_2+B_2+\cdots+A_{k-1}+B_{k-1}+A_k$ to ∞.

Each animal induces a *word*, that can be read from right to left, obtained by looking at the set-types induced by its vertical cross sections. For example, for the animal A above: we first have the word

$$\{(0,0),(0,1),(0,3)\},\{(1,1),(1,2),(1,3)\},\{(2,2)\}\}.$$

We then look at the set-types of these vertical cross-sections (ignoring the x-coordinate that is the same at any given vertical cross-section). In this example we would have $[[2,1,1],[3],[1]]$.

So the alphabet consists of lists of positive integers, $[A_1, B_1, A_2, \ldots, B_{k-1}, A_k]$, $k \geq 1$, of odd size. Such a letter-type means that at the given cross section, scanning from bottom up, we first have a board of A_1 dots belonging to our animal, followed by a gap of B_1 dots, followed again by a board of length A_2, and so on, the top board being of length A_k.

Each animal uniquely defines a word in the above alphabet. But this resulting word is not enough to reconstruct the animal, since we also have to describe how each 'letter' gets interfaced to the next one. Also, for the sake of Markovity, we have to know how the k boards (of lengths $A_1, A_2, \ldots, A_k$) that feature in $[A_1, B_1, A_2, \ldots, B_{k-1}, A_k]$ relate to each other under the relation "reachability from the right". This is clearly an equivalence relation, and hence the k boards of length $A_1, A_2, \ldots, A_k$ form a *set partition* amongst themselves. It is easy to see ([Z0]) that they must form a *non-crossing set partition.* It is well-known, and not too hard to see, that the number of non-crossing set partitions of a k-element set $\{1, 2, \ldots, k\}$ equals the Catalan number $C_k = (2k)!/(k!(k+1)!)$. For example when $k = 1$ there is obviously only one non-crossing set partition: $\{\{1\}\}$, when $k = 2$, there are two: $\{\{1\}, \{2\}\}$ and $\{\{1, 2\}\}$; When $k = 3$, we have 5, and when $k = 4$ we have 14: all the $B_4 = 15$ set partitions except the *crossing* set-partition $\{\{1, 3\}, \{2, 4\}\}$.

The umbral letters

The beauty of the umbral approach is that we can consider families of *infinitely many letters* at once. What we do is look at all the letters with the same number of boards, and inducing the same non-crossing set partition. We will call each such family an *umbral* letter. Thus, there is only one umbral letter for 1-board letters, namely $\{\{1\}\}$, there are two umbral letters for 2-board letters:$\{\{1\}, \{2\}\}$, and $\{\{1, 2\}\}$, etc. The natural classes to consider are, for a specified k, the set $A(k)$, of all animals all of whose vertical cross-sections have *at most* k boards. Hence the umbral alphabet for $A(k)$ will consist of $C_1 + C_2 + \cdots + C_k$ letters.

The weight

To the concrete pre-letter $[A_1, B_1, A_2, B_2, \ldots, B_{k-1}, A_k]$ with k boards and $k-1$ gaps, we assign the *weight*

$$(qx_1)^{A_1} y_1^{B_1} (qx_2)^{A_2} y_2^{B_2} \ldots y_{k-1}^{B_{k-1}} (qx_k)^{A_k} .$$

Note that the variable q keeps track of the number of dots, which is our main concern, while the variables $x_1, y_1, x_2, \ldots, y_{k-1}, x_k$ are *catalytic variables*, ultimately to be all set equal to 1.

Interfaces

How can a pre-letter, let's call it M, consisting of m boards and $m-1$ gaps be placed to the left of an n-board letter, N? Let's number the bottom (infinite) 'gap' (i.e. the region under the first board) of N, by 1, the bottom board of N by 2, the gap between the first and second boards of N, by 3, the second board by 4, and so on, until we reach the top board of N, that gets to be numbered $2n$, and the infinite (last) 'gap' above it, that is called 'region $2n+1$'. Let's call these $2n+1$ intervals of N 'regions'. Now we look, for each of the m boards of the tentative letter where it begins and where it ends. Suppose that the first (bottom) board of

M starts at region a_1 and ends at region b_1 of N, the second board of M starts at region a_2 and ends at region b_2 of N, ..., the m^{th} board of M starts at region a_m and ends at region b_m. All this information is encoded into a list of m pairs:

$$[[a_1, b_1], [a_2, b_2], \dots, [a_m, b_m]].$$

Of course we must have $a_1 \le b_1 \le a_2 \le b_2 \le \dots \le a_m \le b_m$.

Not every interface is *legal*. We need that every component of the letter N will touch at least one board of M or else the constructed animal will die (destined to be disconnected). We will denote by Interfaces(m, n) the set of all the interfaces between the m-boards on the left and the n boards and $n+1$ gaps (including the infinite gaps on the bottom and top). Given the letter N, the subset of Interfaces(m, n) that consists of legal continuation to the left, of the umbral letter N, will be denoted by LegalInterfaces(m, n, N). For example:

$$\text{Interfaces}(1,1) = \{[[1,1]], [[1,2]], [[1,3]], [[2,2]], [[2,3]], [[3,3]]\},$$

but

$$\text{LegalInterfaces}(1,1,\{\{1\}\}) = \{[[1,2]], [[1,3]], [[2,2]], [[2,3]]\}.$$

The interface $[[1,1]]$ is illegal since then the first (and only) board of M starts and ends in region 1 of N (which is the infinite bottom gap), leaving the sole board of N untouched. For a similar reason $[[3,3]]$ is illegal, since now the only board of M is totally immersed inside the top infinite gap.

Induced letters

Given the letter N that stands on the right, and a legal interface of a pre-letter M with m boards, we turn the pre-letter M into a genuine letter: a non-crossing set-partition of $\{1, 2, \dots, m\}$. Board i and board j of M belong to the same component iff they touch the same component of N, since then it is possible to walk, from board i to board j, without using the dots to the left of M.

Given a letter N (a non-crossing set partition), and one of its legal interfaces, I, let's denote by LeftLetter(I, N) the induced letter M, that the boards of I turn into.

For example: LeftLetter$([[2,3],[4,5]], \{\{1,3\},\{2\}\})$ equals $\{\{1\},\{2\}\}$, since the first board, $[2,3]$ touches the component $\{1,3\}$ while the second board, $[4,5]$ touches the component $\{2\}$. On the other hand LeftLetter$([[2,3],[4,6]], \{\{1,3\},\{2\}\})$ equals $\{\{1,2\}\}$ since both boards $[2,3]$ and $[4,6]$ touch the same component of $N : \{1,3\}$.

Gap interfaces

Remember from [Z1] that we need a completely algorithmic way to find the umbral evolution, i.e., starting with a generic member of a fixed k-board letter, with weight $x_1^{A_1} y_1^{B_1} \dots y_{k-1}^{B_{k-1}} x_k^{A_k}$, to find all the ways of continuing it. Hence we had to break it up into cases according to interfaces. But we have to break these up even further, before we are ready to compute the umbral scheme.

We need to know not only how the m boards of the left letter M interface with the letter to its right, N, but also how the $m+1$ gaps do so. Here we also include the two infinite 'gaps': the one below the bottom board and the one above the top board. "But this is implied!" you would shout. If we have the interface $[[a_1, b_1], \dots, [a_m, b_m]]$ of letter M in relation to letter N, which means that the first board of M starts at region a_1 and ends at region b_1, the second board starts at region a_2 and ends at region b_2, etc., then necessarily, the bottom infinite gap must go from region 1 to region a_1, the first genuine gap (between the first and second board) must go between region b_1 and a_2, the second gap must extend between b_2 and a_3 and so on. Well, there are other possibilities. For example the bottom infinite gap may end at region $a_1 - 1$ (if $a_1 > 1$). Also the gap between the first and second boards may start at region $b_1 + 1$, or it may end in region $a_2 - 1$, or both! (if feasible), and so on.

Given an interface I, and a positive integer n, denoting the number of boards of the right letter N, we will denote by GapInterfaces(I, n) all the possible gap interfaces compatible with I. For example GapInterfaces([[1,2],[3,4]],2) equals

$$\{[[1,1],[2,3],[4,5]],[[1,1],[2,3],[5,5]],[[1,1],[2,2],[4,5]],[[1,1],[2,2],[5,5]],\\ [[1,1],[3,3],[4,5]],[[1,1],[3,3],[5,5]]\}.$$

In this case the first member of each gap-interface must be $[1,1]$, since $a_1 = 1$. The last gap-interface above: $[[1,1],[3,3],[5,5]]$, together with its parent interface $[[1,2],[3,4]]$ represents the following scenario. The letter M has two boards and three gaps. The bottom (infinite) gap extends from region 1 and ends at region 1. The bottom board starts at region 1 and ends at region 2. The middle gap starts at region 3 and ends at region 3; the top board starts at region 3 and ends at region 4, and the infinite gap above the top board starts at region 5 and ends at region 5.

A very important polynomial in this algorithm

Let A be a *symbol* denoting an integer, and let $z_1, z_2, \ldots, z_m$ be variables. We let

$$P_A(z_1, \ldots, z_m) = \sum z_1^{i_1} z_2^{i_2} \cdots z_m^{i_m},$$

where the sum extends over all m-tuples of *positive* integers $(i_1, \ldots, i_m)$ satisfying $i_1 + \cdots + i_m = A$. Note that $P_A(z_1, \ldots, z_m) = z_1 \cdots z_m h_{A-m}(z_1, \ldots, z_m)$, where h_i is the usual *complete homogeneous symmetric polynomial* of degree i.

If $Z = \{z_1, \ldots, z_m\}$ is a set of variables, we will sometimes write the above as $P_A(Z)$, which is legitimate, since P_A is a symmetric function of its arguments.

We have, of course,

$$P_A(z_1, \ldots, z_m) = \sum_{i=1}^{A-1} P_{A-i}(z_1, \ldots, z_{m-1}) z_m^i.$$

Since $P_A(z_1) = z_1^A$, we can repeatedly use the above inductive formula to get $P_A(z_1, \ldots, z_m)$ for any m. All that is involved is summing geometric series, that Maple does very well. For example

$$P_A(z_1, z_2) = \frac{z_1^A z_2 - z_1 z_2^A}{z_1 - z_2}.$$

Note that the evaluated expression of $P_A(z_1, \ldots, z_m)$ is a linear combination of $z_1^A, z_2^A, \ldots, z_m^A$ with coefficients that are rational functions of $z_1, z_2, \ldots, z_m$.

How can each individual board and gap of the right letter be continued to the left?

So now we have the input letter with n boards, N, with its generic weight $x_1^{A_1}y_1^{B_1}x_2^{A_2}\dots y_{n-1}^{B_{n-1}}x_n^{A_n}$, and we also have a specific interface and a specific gap-interface amongst its compatible ones. Let $[[a_1,b_1],[a_2,b_2],\dots,[a_m,b_m]]$ be that specific interface, and let $[[c_1,d_1],[c_2,d_2],\dots,[c_{m+1},d_{m+1}]]$ be the specific gap-interface. This means that the bottom board of N is of length A_1. (Note that, e.g., A_1 is symbolic, not numeric—that's the beauty of the present approach!) Now let's see which parts of the continued-to-the-left letter, M, reside on that bottom board. Recall that we call the first (bottom) board of N *region 2*. So we look at all the intervals of the interface $[[a_1,b_1],[a_2,b_2],\dots,[a_m,b_m]]$, that contain 2, i.e. all those $[a_i,b_i]$ that satisfy $a_i \le 2 \le b_i$. Whenever this is the case this means that the variable x_i (that lives on the i^{th} board of M) 'occupies space' on the first (A_1) board. Let the set of those x_i that made it be X. Similarly, we do the same for the gap-interface, with their associated variables 1 (corresponding to the infinite bottom gap), y_1 (corresponding to the gap between the bottom board to the one right above it), etc. Let the set of y's that thus made it be called Y. So the M-occupants of the bottom board of N consists of the variables $X \cup Y$, and each of them shows up at least once. So the generating function of *all* such possibilities is $P_{A_1}(X\cup Y)$. Now we do the same for each of the (genuine) gaps (of length B_1, $B_2,\dots$) and the rest of the boards (of length $A_2, A_3,\dots$). Since each of the decisions of the occupants are independent, to get the generating function for *all* possible continuations, with that specific interface and gap interface, we multiply $P_{A_1}(.)P_{A_2}(.)\cdots P_{A_n}(.)$ times $P_{B_1}(.)P_{B_2}(.)\cdots P_{B_{n-1}}(.)$, where the arguments of the P_{A_i} and P_{B_i} are the variables that "reside" on the appropriate board and gap respectively.

But, we also have to consider the bottom and top infinite gaps of the input letter N, since they also may have the boards and gaps of M on top of them. Remember that they are called region 1 and $2n+1$ respectively. As before we look at the appropriate variables that correspond to the boards and gaps of M that reside there, and multiply the previous product by $P_\infty(.)P_\infty(.)$.

Finally we replace each of the x_i's $(i=1,\dots,m)$ by qx_i, because the new boards generate more area.

An example

Suppose the (input) letter, N, on the right is the 3-board letter $\{\{1,3\},\{2\}\}$. Suppose that letter to the left (with 3 boards) has interface $[[1,1],[1,3],[4,7]]$, and has gap-interface $[[1,1],[1,1],[3,4],[7,7]]$.

Who intersects with the bottom infinite gap of N? Of course the bottom infinite gap of M, but this does not count. Then we have the first board (since $1 \le 1 \le 1$), the first gap (since $1 \le 1 \le 1$), and second board (since $1 \le 1 \le 3$). This contributes $P_\infty(x_1,y_1,x_2)$.

Who intersects with the first (bottom) board of N? Only the second board of M (since $1 \le 2 \le 3$). This gives a contribution of $P_{A_1}(x_2)=x_2^{A_1}$ to the product.

Who resides on the second gap of N, between the first and second boards of N? First x_2, (since $1 \le 3 \le 3$) and then y_2, (since $3 \le 3 \le 4$). So this gives a contribution of $P_{B_1}(x_2,y_2)$.

Who resides on the second board of N, of length A_2? y_2 (since $3 \le 4 \le 4$) and x_3. So this contributes $P_{A_2}(y_2, x_3)$ to the product. Similarly, only x_3 lives on the gap between the second and third board of N, contributing $P_{B_2}(x_3)$. Also, only x_3 lives on the third board of N, of length A_3, giving a contribution of $P_{A_3}(x_3)$, and finally, only x_3 lives on the top infinite gap of N, contributing $P_\infty(x_3) = x_3/(1-x_3)$.

Note that the resulting letter M is $\{\{1\},\{2,3\}\}$. So we finally multiply it by $Z[\{\{1\},\{2,3\}\}]$. Hence the pre-umbra corresponding ONLY to this particular pair of (interface, gap-interface) is

$$x_1^{A_1} y_1^{B_1} x_2^{A_2} y_2^{B_2} x_3^{A_3} Z[\{\{1,3\},\{2\}\}] \to$$
$$P_\infty(qx_1, y_1, qx_2) P_{A_1}(qx_2) P_{B_1}(qx_2, y_2) P_{A_2}(y_2, qx_3) P_{B_2}(qx_3) P_{A_3}(qx_3) P_\infty(qx_3)$$
$$\cdot Z[\{\{1\},\{2,3\}\}].$$

The pre-umbras

Now we (or rather our computers) do this for each and every one of the interfaces and each and every one of its accompanying gap-interfaces, and add all these expressions up. This will give us the action of our umbral scheme on a generic monomial $x_1^{A_1} y_1^{B_1} x_2^{A_2} y_2^{B_2} x_3^{A_3} Z[\{\{1,3\},\{2\}\}]$. That's what we called pre-umbra in [Z1].

The pre-umbral matrix

Now we form a square matrix whose dimension is the size of the animal alphabet (i.e. $C_1 + C_2 + \cdots + C_k$), both of whose rows and columns are labelled by letters (non-crossing partitions of sets of $\le k$ elements). The entry corresponding to the row labelled by L_1 and column labelled by L_2 is the coefficient of $Z[L_2]$ in the pre-umbra constructed above, which is our umbra applied to the generic monomial $x_1^{A_1} y_1^{B_1} x_2^{A_2} \ldots y_{m-1}^{B_{m-1}} x_m^{A_m} Z[L_1]$, where m is the number of boards of L_1.

The umbral matrix

We now convert, as described in [Z1], each of the entries of the pre-umbral matrix into a full-fledged umbra. (this is accomplished by procedure `ToUmbra` in `ROTA`, that has been transported to `ZOO`).

The umbral scheme

Recall that ([Z1]) in addition to the umbral matrix, an umbral scheme also needs a subset of *starting letters*, *ending letters*, and the *initial generating functions* for the starting letters. Since the letters correspond to the (necessarily non-crossing) set partitions of the set of boards, and two boards belong to the same member-set of the set-partition iff they can be reached from each other by only using points *to the right*, and we go (as in Arabic) from right to left, the rightmost (first) letter can only be the one corresponding to all singletons. Hence there are k right-letters altogether: $\{\{1\}\}, \{\{1\},\{2\}\}, \ldots \{\{1\},\{2\},\ldots,\{k\}\}$. Who are the terminal letters? Here we have the other extreme, since the boards may not reach each other from the left (since they are leftmost), and since animals are connected, all the boards must belong to the same component, and the only eligible set-partitions are those consisting of *one* member set, namely *everything*. So the terminal (leftmost) letters are : $\{\{1\}\}, \{\{1,2\}\}, \dots \{\{1,2,\dots,k\}\}$.

Finally the initial generating function for one-letter (pre-) animals, for the i-board letter ($1 \le i \le k$), $\{\{1\},\{2\},\dots,\{i\}\}$, is obviously

$$\prod_{j=1}^{i} \frac{qx_j}{1-qx_j} \prod_{j=1}^{i-1} \frac{y_j}{1-y_j},$$

since the initial i boards and initial $i-1$ gaps between them may be of any positive length independent of each other.

A user's manual for the maple package ZOO

We strongly advise the reader to study carefully the source code of the Maple package `ZOO` that automatically finds umbral schemes for $\le k$-board animals for *arbitrary k*. It does not require `ROTA`, since all the necessary procedures of the latter were transported.

The main procedures of `ZOO` are `AniUmSc` and `ApplyUmSc`. If you want to get the umbral scheme for enumerating animals with $\le k$ boards in each vertical cross-section, type `AUSk:=AniUmSc(k,x,y,q);`. Here `k` is a specific positive integer (`k`$=1,2,\dots$), while `x` and `y` are the indexed variable-names for the board- and gap-lengths as described above, and q is the variable of interest, that carries the number of dots.

Once Maple gives you the umbral scheme (that we chose to call `AUSk` above), to get the first n terms in the enumerating series (what physicists call "series expansions") type `ApplyUmSc(AUSk,q,n,var):`, where `var` is the *set* of catalytic variables: $\{x[1], y[1], x[2], \dots y[k-1], x[k]\}$. E.g., try `ApplyUmSc(AUS1,q,n,{x[1]}):` and `ApplyUmSc(AUS2,q,n,{x[1],y[1],x[2]}):` .

Since outputting the umbral scheme, `AUS2`, for $\le$ 2-board animals takes a while, we pre-computed `AUS2`, and for the sake of completeness, also `AUS1`.

Typing `ApplyUmSc(AUS2,q,54,{x[1],y[1],x[2]}):` and waiting for a few days, extends sequence **A001170** (Formerly **M1638** and **N0640**) ("Board-pair-pile polyominoes with n cells", first computed in [L]) of Neil Sloane and Simon Plouffe's *Encyclopedia* and *database* to 54 terms. You would get:

```
   [1, 2, 6, 19, 63, 216, 760, 2723, 9880, 36168, 133237, 492993, 1829670, 6804267,
25336611, 94416842, 351989967, 1312471879, 4894023222, 18248301701, 68036380665,
253638655582, 945464013411, 3523978989671, 13133649924269, 48944841261703,
182390886053785, 679639952406737, 2532435605836553, 9435940029787771,
35157829654829347, 130993992060546335, 488061959593417980, 1818420122974985383,
6775015368226385755, 25242011759461486461, 94045041136136802213,
350385854676874166894, 1305438172166371546918, 4863684396241586531678,
18120658979728427379998, 67512198042235076760287, 251530274898487735498572,
937126264995917017697537, 3491450036706654367545738, 13008087968822268294707203,
48464198629752538664862046, 180562916626733292681832702, 672722655136932068675036320,
2506360408024259482601919315, 9337937683903257909568706940,
34790318720202928318378465769, 129618154246548101581553316346,
482917840548933948578460002791].
```

The big disappointment

`ZOO` can, *in principle*, find an umbral scheme for any given, fixed, number of boards. But the number of interfaces grows so fast that Maple ran out of memory even when our computer tried to compute the umbral scheme for 3-board animals, i.e., it refused to perform the command: `AUS3:=AniUmSc(3,x,y,q);`.

We are almost sure that with a more careful implementation, and/or by transporting to numeric programming languages such as `C`, `AUS3:=AniUmSc(3,x,y,q);` is still feasible, but frankly, we doubt whether it is worth the effort.

The most important point is that there *exists* a program that can generate (at least in principle) a polynomial-time (in n), scheme for enumerating k-board animals with $\leq n$ cells, for any fixed given k. So what if the 'polynomial' is so big as to make it impractical? I will be very impressed if you can *just* prove the *existence* of an $O(n^{1000000000000000})$ algorithm for *Satisfiability*!

Many Thanks are due to the dedicated referee for his or her very careful reading, that corrected many minor errors.

References

[L] W. F. Lunnon, *Counting polyominoes*, Computers in Number Theory (A. O. L. Atkin and B. J. Birch, eds.), Academic Press, NY, 1971, pp. 347-372, Zbl 0214.51604.

[Z0] Doron Zeilberger, *Symbol-crunching with the transfer-matrix method in order to count skinny physical creatures*, Integers (`http://www.integers-ejcnt.org`) **0** (2000), paper A9 (34 pages), MR 2001e:05010, Zbl 0954.82006.

[Z1] Doron Zeilberger, *The umbral transfer-matrix method. I. Foundations*, J. Comb. Theory Ser. A **91** (2000), 451–463, MR 2001g:05018, Zbl 0961.05003.

[Z2] Doron Zeilberger, *The umbral transfer-matrix method. II. Counting plane partitions*, Personal Journal of Ekhad and Zeilberger, `http://www.math.rutgers.edu/~zeilberg/pj.html`.

[Z4] Doron Zeilberger, *The umbral transfer-matrix method. IV. Counting self-avoiding polygons and walks*, Elec. J. Comb. **8(1)** (2001), R28 (17 pages).

[Z5] Doron Zeilberger, *The umbral transfer-matrix method. V. The Goulden-Jackson cluster method for infinitely many mistakes*, submitted. Available from the author's website.

Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA
zeilberg@math.rutgers.edu http://www.math.rutgers.edu/~zeilberg/

This paper is available via http://nyjm.albany.edu:8000/j/2001/7-14.html.