Automatic Counting of Restricted Dyck Paths
via (Numeric and Symbolic) Dynamic Programming

Shalosh B. EKHAD and Doron ZEILBERGER

In memory of our hero Richard Guy (1916-2020)

"On some fields it is difficult to tell whether they are sound or phony. Perhaps they are both. Perhaps the decision depends on the circumstances, and it changes with time .... Some subjects start out with impeccable credentials, catastrophe theory for instance, and then turn out to resemble a three-dollar bill. Others, like dynamic programming, have to overcome a questionable background before they are reluctantly recognized to be substantial and useful."
– Gian-Carlo Rota ("Discrete Thoughts", Birkhäuser, 1992, p. 1)

Preface

In his classic essays "The Strong Law of Small Numbers" [G1][G2], Richard Guy gave numerous *cautionary tales* where one can't 'jump to conclusions' from the first few terms of a sequence. But if you are cautious enough you can find many families of enumeration problems where it is very safe to deduce the general pattern from the first few cases, obviating the need for either the human or the computer to think too hard, and by using the **'Keep It Simple Stupid'** principle (**KISS** for short), one can easily derive many deep enumeration theorems by doing exactly what Richard Guy told us **not** to do, compute the first few terms of the sequence and deduce the formula for the general term. We admit that often 'few' should be replaced by 'quite a few', but it is still much less painful than trying to figure out the intricate combinatorial structure by 'conceptual' means.

The way we generate sufficiently many terms of the studied sequence that enables guessing the pattern is via *numeric* **dynamic programming**. While it is very fast to generate many terms of the sequence, the guessing part, for larger problems, would eventually become impractical. For these larger problems, both the human programmer and the machine need to think a bit, and use **symbolic** dynamic programming. Note that the human only has to think **once**, writing a general program that teaches the computer how to 'think' in each specific case of a large class of enumeration problems.

The Maple packages

This article is accompanied by two Maple packages

• `Dyck.txt`, that uses **Numeric** Dynamic programming and *number-crunching* to generate many terms of the desired sequence, and then uses the KISS method to guess algebraic equations and linear recurrences that can be rigorously justified *a posteriori*.

• `DyckClever.txt`, that uses **Symbolic** Dynamic programming and *symbol-crunching* to directly get the desired equations for the generating functions of interest, and by using further symbol-crunching, derives linear recurrences for the sequences of interest.

Both packages are available, along with input and output files with ample data, from the front of the present article

`http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/dyck.html` .

Part I: Using Numeric Dynamic Programming to Enumerate Restricted Dyck Paths

Enumerating Dyck Paths

Recall that a **Dyck path** of **semi-length** n is a walk in the plane, from the origin $(0,0)$ to $(2n,0)$ with **atomic steps** $U := (1,1)$ and $D := (1,-1)$ that **never** goes below the $x-$axis, i.e. that always stays in $y \geq 0$.

For example, the five Dyck paths of semi-length 3 are

$$UUUDDD \quad , \quad UUDUDD \quad , \quad UUDDUD \quad , \quad UDUUDD \quad , \quad UDUDUD \quad .$$

The number of Dyck paths of semi-length n is famously the **Catalan number**, $\frac{(2n)!}{n!(n+1)!}$, the most popular, and **important** sequence in enumerative combinatorics (with no offense to Fibonacci), the subject of a whole book by Guru Richard Stanley [St].

When we searched (on June 2, 2020) our favorite website, the OEIS [Sl] for the phrase "Dyck paths" we got back 1100 hits. Of course we did not have the patience to read all of them, but a random browsing revealed that they enumerate Dyck paths with various **restrictions**, and they refered to the interesting papers [ELY], [PW], and [M]. These were all (admittedly clever) human-generated efforts. We will soon see how to quickly *automatically* enumerate 'infinitely' many such classes, but first let's recall one of the many proofs of the fact that the number of Dyck paths of semi-length n is indeed the venerable OEIS sequence A108, $\frac{(2n)!}{n!(n+1)!}$.

Let $a(n)$ be the desired number, i.e. the number of Dyck paths of semi-length n, and consider the **ordinary generating function**

$$f(x) := \sum_{n=0}^{\infty} a(n)\, x^n \quad ,$$

which is the **weight-enumerator** of the set of all Dyck paths, with $weight(P) := x^{SemiLength(P)}$.

It is readily seen that any Dyck path P is either empty or can be written **uniquely** (i.e. **unambiguously**) as $P = U\,P_1\,D\,P_2$, where P_1 and P_2 are shorter Dyck paths, and vica versa, for any Dyck paths P_1, P_2, $U\,P_1\,D\,P_2$ is a Dyck path on its own right. Let $\mathcal{P}$ be the **totality** of all Dyck paths, then we have the **grammar**

$$\mathcal{P} = \{EmptyPath\} \cup U\,\mathcal{P}\,D\,\mathcal{P} \quad .$$

Applying the **weight** functional we get

$$f(x) = 1 + x f(x)^2 \quad .$$

To deduce that $a(n) = (2n)!/(n!(n+1)!)$, you can, *inter alia*

• (i) Solve the quadratic and use Newton's binomial theorem .

• (ii) Differentiate both sides getting a differential equation for $f(x)$ that translates to a first-order recurrence for $a(n)$.

• (iii) Use Lagrange Inversion (see [Z1] for a brief and lucid exposition).

How it All Started: Vladimir Retakh's Question

Volodia Retakh asked whether there is a proof of the fact that the number of Dyck paths of semi-length n such that the height of all peaks is either 1 or even is given by the also famous Motzkin numbers (OEIS sequence A1006), whose generating function satisfies

$$f(x) = 1 \, + \, x f(x) \, + \, x^2 f(x)^2 \quad .$$

We first tried to find a 'conceptual' proof generalizing the above proof, and indeed we found one, by adapting the above classical proof enumerating all Dyck paths.

Let $\mathcal{P}_1$ be the set of Dyck paths whose peak-heights are never in $\{3, 5, 7, \dots\}$, and let $f_1 = f_1(x)$ be its weight enumerator.

Let P_1 be any member of $\mathcal{P}_1$ then, it is either empty, or we can write

$$P_1 \, = \, U \, P_2 \, D \, P_1' \quad ,$$

where $P_1' \in \mathcal{P}_1$ but P_2 has the property that none of its peak-heights is in $\{2, 4, 6, \dots\}$. Let $\mathcal{P}_2$ be the set of such Dyck paths.

Hence, we can write the 'grammar'

$$\mathcal{P}_1 \, = \, \{EmptyPath\} \cup U \, \mathcal{P}_2 \, D \, \mathcal{P}_1 \quad .$$

Let $f_2 = f_2(x)$ be the weight-enumerator of $\mathcal{P}_2$.

Taking weights above, we have the equation

$$f_1 = 1 + x f_2 f_1 \quad .$$

Alas, now we have to put-up with $\mathcal{P}_2$ and f_2. Let P_2 be any member of $\mathcal{P}_2$. Then either it is empty, or it can be written as

$$U \, P_3 \, D \, P_2' \quad ,$$

where $P_2' \in \mathcal{P}_2$ but P_3 is a Dyck path whose peak-heights are never in $\{1,3,5,7,\dots\}$.

Let $\mathcal{P}_3$ be the set of such Dyck paths. We have the grammar

$$\mathcal{P}_2 \,=\, \{EmptyPath\} \cup U\,\mathcal{P}_3\,D\,\mathcal{P}_2 \quad .$$

Let $f_3 = f_3(x)$ be the weight-enumerator of $\mathcal{P}_3$.

Taking weights we have another equation

$$f_2 = 1 + x f_3 f_2 \quad .$$

It looks like we are doomed to have **infinite regress**, but let's try one more time.

Let P_3 be any member of $\mathcal{P}_3$. It is either empty, or we can write

$$P_3 \,=\, U\,P_4\,DP_3'$$

where $P_3' \in \mathcal{P}_3$ and P_4 is a **non-empty** path avoiding peak-heights in $\{2,4,6,8,\dots\}$. But this looks familiar, so the set of P_4-paths is really $\mathcal{P}_2 \backslash \{EmptyPath\}$, and we have the grammar

$$\mathcal{P}_3 \,=\, \{EmptyPath\} \cup U\,(\mathcal{P}_2 \backslash \{EmptyPath\})\,D\,\mathcal{P}_3 \quad .$$

Taking weight, we get

$$f_3 = 1 + x(f_2 - 1) f_3 \quad .$$

We have a system of three algebraic equations

$$\{ f_1 = 1 + x f_2 f_1 \quad , \quad f_2 = 1 + x f_3 f_2 \quad , \quad f_3 = 1 + x(f_2 - 1) f_3 \quad \} \quad ,$$

in the unknowns

$$\{f_1, f_2, f_3\} \quad .$$

Eliminating f_2, f_3 yields the following algebraic equation for our object of desire f_1,

$$f_1(x) = 1 \,+\, x f_1(x) \,+\, x^2 f_1(x)^2 \quad ,$$

proving Volodia Retakh's claim.

In Part II we will see how to teach the computer to do these reasonings, but if neither human nor machine feel like thinking too hard, for many problems one can use the **KISS** method.

The KISS way

Now that we know that such an argument **exists**, and that the desired generating function $f(x)$, satisfies an algebraic equation of the form $P(x, f(x)) = 0$ for *some* bivariate polynomial $P(x, y)$, why not **keep it simple**, and rather than wrecking our brains (both those of humans and those of machines), let's collect sufficiently many terms of the desired sequence, and then use Maple's command `gfun[listtoalgeq]` (or our own home-made version) to **guess** the desired polynomial equation $P(x, f(x)) \,=\, 0$.

Numerical Dynamic Programming to the rescue

Suppose that we don't know anything, and want to compute the number of Dyck paths of semi-length n, i.e. the number of **all** walks using the fundamental steps $U = (1, 1)$ and $D = (1, -1)$, that start at $(0, 0)$, end at $(2n, 0)$ and never visit $y < 0$. A natural approach is to consider the more general quantity $d(m, k)$, the number of walks from $(0, 0)$ to (m, k) staying weakly above the x-axis and **ending at a down step**. If the length of that downward run is r, then the previous peak was at $(m - r, k + r)$, and we need to introduce the auxiliary function $u(m, k)$ the number of such paths that end at (m, k) and end at an up step.

We have

$$d(m,k) \; = \; \sum_{r=1}^{m} u(m-r, k+r) \quad .$$

Analogously,

$$u(m,k) \; = \; \sum_{r=1}^{m} d(m-r, k-r) \quad .$$

Of course we have the obvious **initial condition** $d(0, 0) = 1$, and the **boundary conditions** $d(m, k) = 0$ and $u(m, k) = 0$ if $k > m$.

Here is the short Maple code that does it

```
u:=proc(m,k) local r:  option remember:  if m=0 then 0:  else add(d(m-r,k-r),r=1..k):
fi:  end:
```

```
d:=proc(m,k) local r:  option remember:
if m=0 then if k=0 then RETURN(1):  else RETURN(0):  fi:  fi:  add(u(m-r,k+r),r=1..m):
end:
```

To get the desired sequence enumerating all Dyck paths of semi-length n for n from 1 to N, in other words $\{d(2n, 0)\}_{n=0}^{N}$ for any desired N, we type

```
seq(d(2*n,0),n=0..N);   .
```

Now, recall that we had to work much harder, *logically* and *conceptually*, to find the algebraic equation for the Dyck paths considered by Volodia Retakh. To get the analogous sequence we only need to change the program by **one line**. Let's call the analogous quantities $u_1(m, k)$ and $d_1(m, k)$.

```
u1:=proc(m,k) local r:  option remember:  if (k>1 and k mod 2=1) then RETURN(0):  fi:
if m=0 then 0:  else add(d1(m-r,k-r),r=1..k):  fi:  end:
```

```
d1:=proc(m,k) local r:  option remember:  if m=0 then if k=0 then RETURN(1):  else
RETURN(0):  fi:  fi:  add(u1(m-r,k+r),r=1..m):  end:
```

In other words, just declaring that $u_1(m, k) = 0$ if the elevation k is an odd integer larger than 1.

Typing

```
seq(d1(2*n,0),n=0..N);
```

will let us get, very fast, the first $N+1$ terms, that would enable us to guess the algebraic equation satisfied by the generating function, that we can justify *a posteriori*, since we know that it **exists**, saving us the mental agony of doing it logically, by figuring out the intricate 'grammar'.

The general case

Since it is so easy to tweak the numerical dynamic programming procedure, why not be as general as can be? Let A, B, C, D be *arbitrary* sets of positive integers, either finite sets, or infinite sets (like in Retakh's case) that are arithmetical progressions (or unions thereof). We are interested in counting Dyck paths that obey the following restrictions

• No peak can be of a height that belongs to A.

• No valley can be of a height that belongs to B.

• No upward run can be of a length that belongs to C.

• No downward run can be of a length that belongs to D.

Then we declare that $u(m,k)=0$ if $k \in A$ and $d(m,k)=0$ if $k \in B$ and otherwise

$$d(m,k) \;=\; \sum_{\substack{1 \le r \le m \\ r \notin C}} u(m-r,k+r) \quad .$$

Analogously,

$$u(m,k) \;=\; \sum_{\substack{1 \le r \le m \\ r \notin D}} d(m-r,k-r) \quad .$$

Then we get, very fast, sufficiently many terms to guess an algebraic equation, by finding $\{d(2n,0)\}_{n=0}^{N}$, for a sufficiently large N.

Guessing linear recurrences

It is well-known (see [KP], Theorem 6.1) that if $f(x)$ is an algebraic formal power series (like in our case), then it satisfies a linear differential equation with polynomial coefficients, i.e. it is D-finite, and hence its sequence of coefficients, $\{a(n)\}_{n=0}^{\infty}$, satisfies a linear recurrence equation with polynomial coefficients, i.e. is P-recursive. While there are easy algorithms for finding these, they do not always give the minimal recurrence, and once again, let's keep it simple! Just guess such a recurrence using *undetermined coefficients*, and we are guaranteed by the background 'general nonsense' that everything is rigorously proved, and we don't have to worry about Richard Guy's Strong Law of Small Numbers.

The Maple package `Dyck.txt`

Everything here is implemented in the Maple package `Dyck.txt` available from the front of this article

`https://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/dyck.html` .

There you would also find long web-books with many deep enumeration theorems. Let us present just one **random** example.

Typing

`Theorem({1},{},{2},{1},60,P,x,n,a,20,1000);`

gives

Sample Theorem: Let $a(n)$ be the number of Dyck paths of semi-length n obeying the following restrictions. The height of a peak can not belong to $\{1\}$, no upward-run can have a length belonging to $\{2\}$, and no downward-run can have a length that belongs to $\{1\}$, then the generating function

$$f(x) := \sum_{n=0}^{\infty} a(n)x^n \quad ,$$

satisfies the algebraic equation

$$1 + \left(-x^2 - x - 1\right) f(x) + \left(x^4 + x^3 + x^2 + x\right) f(x)^2 = 0 \quad ,$$

and the sequence $a(n)$ satisfies the following linear recurrence

$$a(n) = \frac{(n-2)\,a(n-1)}{n+1} + 2\,\frac{(n-2)\,a(n-2)}{n+1} + \frac{(4n-11)\,a(n-3)}{n+1}$$
$$+\frac{(8n-25)\,a(n-4)}{n+1} + 6\,\frac{(n-4)\,a(n-5)}{n+1} + \frac{(5n-22)\,a(n-6)}{n+1} + 3\,\frac{(n-5)\,a(n-7)}{n+1} \quad ,$$

subject to the initial conditions $a(1) = 0, a(2) = 0, a(3) = 1, a(4) = 2, a(5) = 3, a(6) = 7, a(7) = 1$.

Part II: Symbolic Dynamic Programming

We will now briefly describe the **clever** way, implemented in the Maple package `DyckClever.txt`. Note that the "thinking" and "research" is done by the computer *all by itself*, using *symbolic* dynamic programming, to generate a **system of algebraic equations**, and then it solves that system. What happens is that in order to study the generating function for the original set, we are **forced** to consider other sets, that in turn, necessitate yet more sets. Sooner or later, if all goes well, there would be no more new 'uninvited guests', and the computer would have a **finite** system of algebraic equations with **as many equations as unknowns**. Using the Buchberger algorithm *under the hood*, it solves that system, giving us much more than we asked for, not just the desired generating function, but lots of other ones that we had to introduce, and that we may not care about.

Avoiding Peak-Heights and Valley Heights with Finite Sets to Avoid

Suppose that we have two **arbitrary finite** sets of non-negative integers A and B, and we are interested in $f_{A,B}(x)$, the ordinary generating function enumerating Dyck paths such that

• None of the peak-heights is in A .

• None of the valley-heights is in B .

Assume, for now, that $0 \notin A$ and $0 \notin B$, and define

$$A_1 = \{a-1 \,:\, a \in A\} \quad ,$$

$$B_1 = \{b-1 \,:\, b \in B\} \quad .$$

Recall that every Dyck P is either empty or else can be written as

$$UP_1DP_2 \quad ,$$

where P_1 and P_2 are Dyck paths on their own right. If P is counted by $f_{A,B}(x)$ then P_1 is counted by $f_{A_1,B_1}(x)$, but P_2 is counted again by $f_{A,B}(x)$. This leads to the quadratic equation

$$f_{A,B}(x) \;=\; 1 + x f_{A_1,B_1}(x) f_{A,B}(x) \quad .$$

Alas now we have to set-up an equation for $f_{A_1,B_1}(x)$ and keep going. Sooner or later we will get an $f_{A',B'}(x)$ where either A' or B' contain 0 (or both). It is readily seen that if $0 \in A$, then writing $A_1 := A \backslash \{0\}$, we get the equation

$$f_{A,B}(x) \;=\; f_{A_1,B}(x) - 1 \quad .$$

If $0 \notin A$ but $0 \in B$ then define

$$A_1 = \{a-1 \,:\, a \in A\} \quad ,$$

as above and

$$B_1 = \{b-1 \,:\, b \in B\} \backslash \{-1\} \quad ,$$

and the corresponding equation is

$$f_{A,B}(x) = 1 + x f_{A_1,B_1}(x) \quad .$$

Sooner or later there would be no more new 'states' $[A, B]$ and we would have a finite set of quadratic equations with as many equations as unknowns. Solving the resulting system we would get, *inter-alia*, our original object of desire, $f_{A,B}(x)$.

This is implemented in procedure `fAB(A,B,x,P)` in the Maple package `DyckClever.txt` available from

`https://sites.math.rutgers.edu/~zeilberg/tokhniot/DyckClever.txt` .

Just to take a random example, in order to get the quadratic equation satisfied by $P = \sum_{n=0}^{\infty} a(n)x^n$, where $a(n)$ is the number of Dyck paths with no peak-heights in the set $\{2, 5\}$ and no valley-heights in the set $\{1, 4\}$, type

```
fAB({2,5},{1,4},x,P);
```

getting, in a fraction of a second,

$$\left(4\,x^6 - 13\,x^5 + 24\,x^4 - 27\,x^3 + 19\,x^2 - 7\,x + 1\right) P^2$$
$$+\left(4\,x^5 - 15\,x^4 + 26\,x^3 - 26\,x^2 + 12\,x - 2\right) P + x^4 - 4\,x^3 + 8\,x^2 - 5\,x + 1 \,=\, 0 \quad .$$

The closely related procedure `fABcat(A,B,x,C)` expresses P in terms of the Catalan generating function $C := (1 - \sqrt{1-4x})/(2x)$. Typing

```
fABcat( { 2,5 } , { 1,4 },x,C);
```

would give

$$\frac{-Cx^2 + x^2 - 2\,x + 1}{(x^3 - x^2)\,C - 2\,x^3 + 3\,x^2 - 3\,x + 1} \quad .$$

Avoiding Peak-Heights and Valley Heights with infinite Sets to Avoid

The original motivating problem, asked by Volodia Retakh (see above) asked for the number of Dyck paths whose peak-heights is either 1 or even, in other words the set of Dyck paths none of whose peak-heights is in the range of the arithmetical progression $2r + 3$ for $r \geq 0$. The above procedure $fAB(A, B, x, P)$ can be easily modified to procedure

`fABr(A,B,r,x,P)` ,

where A and B are sets of arithmetical progressions written in the form $ar + b$ for a and b non-negative integers and r is a **symbol** ranging over the non-negative integers. For example, to get Retakh's result type

`fABr({2*r+3 },{},r,x,P);` ,

getting

$$x^2P^2 + Px - P + 1 \,=\, 0 \quad .$$

For the equation satisfied by the generating function of the sequence enumerating Dyck paths where neither peak-heights nor valley-height is a member of the arithmetical progression $5r+1$ type

```
fABr( { 5*r+1 } , { 5*r+1 } ,r,x,P);
```

getting

$$\left(x^3+x^2-x\right)P^2+\left(x^3-2\,x^2-x+1\right)P+2\,x-1\,=\,0 \quad .$$

Avoiding Ascending Run-Lengths and Descending Run-Lengths with Finite Sets to Avoid

An **irreducible** Dyck path of *semi-length* n is one who never touches the x-axis except, of course, at the beginning $(0,0)$ and the end, when it is at $(2n,0)$.

Let C and D be **arbitrary** finite sets of positive integers. We are interested in finding the algebraic equation satisfied by the generating function of the enumerating sequence of Dyck paths that never have an **ascending run-length** in C and never have a **descending run-length** in D. Let's call it $h_{C,D}(x)$.

Alas, in order to get the *symbolic dynamic programming* (recursive) procedure rolling, we are **forced** to consider more general classes.

Given sets of positive integers C, C_1, D, D_1, let $h_{C,C_1,D,D_1}(x)$ be the weight-enumerators with $weight(P) := x^{SemiLength(P)}$ of **all** Dyck paths P such that

- The length of a starting upward-run is **not** in C_1 .

- All other upward run-lengths are **not** in C.

- The length of an ending downward-run is **not** in D_1.

- All other downward run-lengths are **not** in D.

Let $H_{C,C_1,D,D_1}(x)$ be the analogous quantity enumerating **irreducible** Dyck paths with these restrictions.

We have the following equation

$$h_{C,C_1,D,D_1}(x)\,=\,1\,+\,H_{C,C_1,D,D_1}(x)\,+\,H_{C,C_1,D,D}(x)\,h_{C,C,D,D}(x)\,H_{C,C,D,D_1}(x) \quad .$$

This follows from the fact that every such Dyck path is either the empty path (weight 1), or is irreducible, or else can be viewed as concatenation of three paths, such that the first is weight-counted by $H_{C,C_1,D,D}(x)$, the second (possibly empty) path counted by $h_{C,C,D,D}(x)$ and the third path by $H_{C,C,D,D_1}(x)$.

We must now find an equation for H_{C,C_1,D,D_1}.

If $1 \notin C_1$ and $1 \notin D_1$ then, let

$$C_2 := \{c-1 \,:\, c \in C_1\} \quad ,$$

$$D_2 := \{d-1 \,:\, d \in D_1\} \quad ,$$

Obviously, we have

$$H_{C,C_1,D,D_1}(x) \,=\, x\, h_{C,C_2,D,D_2}(x) \quad .$$

If $1 \in C_1$ and $1 \notin D_1$ then

$$H_{C,C_1,D,D_1}(x) \,=\, H_{C,C_1\backslash\{1\},D,D_1}(x) - x \quad ,$$

since the set of Dyck paths counted by $H_{C,C_1\backslash\{1\},D,D_1}(x)$ and $H_{C,C_1,D,D_1}(x)$ are almost the same, the only exception is UD that belongs to the former, but not to the latter, and whose weight is 1.

Similarly, if $1 \notin C_1$ and $1 \in D_1$ then

$$H_{C,C_1,D,D_1}(x) \,=\, H_{C,C_1,D,D_1\backslash\{1\}}(x) - x \quad ,$$

Finally, if $1 \in C_1$ and $1 \in D_1$ then

$$H_{C,C_1,D,D_1}(x) \,=\, H_{C,C_1\backslash\{1\},D,D_1\backslash\{1\}}(x) - x \quad .$$

This is implemented in procedure `fCD(C,D,x,P)`. Just to take a random example, in order to get the algebraic equation satisfied by $P = \sum_{n=0}^{\infty} a(n)x^n$ where $a(n)$ is the number of Dyck paths with no ascending run of length 2 and no descending run of length 3 just type

`fCD({2},{ 3 },x,P);`

getting

$$1-(x+1)\left(x^2-x+1\right)P+x\left(x^3+x^2-x+1\right)P^2+x^4\left(x^5-x^3+2\,x^2-1\right)P^3$$
$$+x^4\left(x^6-3\,x^5+x^4-x+1\right)P^4+x^6\left(x^7+2\,x^4+x^3+x^2+1\right)P^5$$
$$+x^9\left(2\,x^6+x^5+2\,x^3+3\,x^2-x+1\right)P^6+x^{11}\left(x^6+2\,x^5+x^3+3\,x^2-2\,x+1\right)P^7$$
$$+x^{15}\left(x^4+x^3+x^2+1\right)P^8+x^{18}\left(x^3+x^2+1\right)P^9+x^{22}P^{10} \,=\, 0 \quad .$$

Avoiding Ascending Run-Lengths and Descending Run-Lengths with infinite Sets to Avoid

The above procedure $fCD(C, D, x, P)$ can be easily modified to procedure

`fCDr(C,D,r,x,P)`

where C and D are sets of arithmetical progressions written in the form $ar+b$ for a and b non-negative integers and r is a symbol ranging over the non-negative integers. For example, to get the generating function for the sequence of Dyck paths where there are no ascending run-lengths in the infinite set $\{3,5,7,9,11,\dots\}$, type:

```
fCDr({2*r+3 },{},r,x,P);
```

getting

$$-1+(-x+1)\,P+x^2\,(x-1)\,P^3 \;=\; 0 \quad .$$

For the equation satisfied by the generating function of the sequence enumerating Dyck paths where there are no ascending run-lengths in the infinite set $\{2,4,6,8,10,\dots\}$, and no descending run-lengths in the infinite set $\{1,3,5,7,9,\dots\}$, type:

```
fABr( { 2*r+2 } , { 2*r+1 } ,r,x,P);
```

getting

$$1+\left(-x^2-1\right)P-P^2x^2+x^2\left(3\,x^2+2\right)P^3-P^4x^4-x^4\left(x^2+1\right)P^5+x^6P^6 \;=\; 0 \quad .$$

Homework Assignments

As in most of our papers, our main goal was to demonstrate an *approach* rather than discover new theorems. It would be relatively straightforward to combine procedures `fAB(A,B,x,P)` and `fCD(C,D,x,P)` to produce procedure `fABCD(A,B,C,D,x,P)` using the 'clever' approach, thereby giving a 'clever analog' of the general problem discussed in Part I. We leave this to the interested reader.

Avoiding an ascending run-length of length a is the same thing as saying that U^aD can't occur at the start of the word, and DU^aD can't occur later, and analogously for avoiding descending run-lengths. This naturally generalizes to the problem of counting Dyck paths avoiding as *consecutive subwords* (i.e. 'factors' in the language of formal languages) an arbitrary finite set of **forbidden subwords**. Both the 'dumb' approach and the 'clever' approach can be easily adapted to this problem, and even when the subwords to avoid are 'infinite set' consisting of regular expressions. We also leave this to the interested reader, as well as *interfacing* it with restricted peak-heights and valley-heights.

Conclusion

For many enumeration problems the '*dumb*' approach, *keeping it simple*, suffices, notwithstanding Richard Guy's cautionary tales, since by *mumbling* a few words, one can easily shut-up the traditional, machinophobic 'rigorist', who would have to accept it, at least reluctantly (he would still say that such a proof 'gives no insight'). This KISS approach is based on *number-crunching*, using *numeric* dynamic programming as the engine.

But for larger problems, we may need, unfortunately, to do some 'thinking' by examining the combinatorial-recursive structure of the combintorial sets involved. Luckily, this approach can be automated as well, requiring us to only meta-think **once** by writing a **general** program that *teaches* the computer to do the thinking for us. This *clever* approach is based on *symbol-crunching*, using *symbolic* dynamic programming as the engine.

References

[ELY] Sen-Peng Eu, Shu-Chung Liu, and Yeong-Nan Yeh, *Dyck Paths with Peaks Avoiding or Restricted to a Given Set*, Studies in Applied Mathematics **111** (2003), 453-465.

[G1] Richard K. Guy, *The strong law of small numbers*, Amer. Math. Monthly, **95** (1988), 697–712.

[G2] Richard K. Guy, *The second strong law of small numbers*, Math. Mag. **63** (1990), 3–20.

[KP] Manuel Kauers and Peter Paule, *"The Concrete Tetrahedron"*, Springer, 2011.

[M] Toufik Mansour, *Counting Peaks at Height k in a Dyck Path*, Journal of Integer Sequences, **5** (2002), Article 02.1.1

[PW] P. Peart and W.-J. Woan, *Dyck Paths With No Peaks at Height k*, J. Integer Sequences **4** (2001), #01.1.3.

[Sl] Neil Sloane, *The On-Line Encyclopedia of Integer Sequences*, `https://oeis.org` .

[St] Richard Stanley, *"Catalan Numbers"*, Cambridge University Press, 2015.

[Z1] Doron Zeilberger, *Lagrange Inversion Without Tears (Analysis) (based on Henrici)*, The Personal Journal of Shalosh B. Ekhad and Doron Zeilberger,
`https://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/lag.html` .

Shalosh B. Ekhad, c/o D. Zeilberger, Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA.
Email: `ShaloshBEkhad at gmail dot com` .

Doron Zeilberger, Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA.
Email: `DoronZeil at gmail dot com` .

Exclusively published in the Personal Journal of Shalosh B. Ekhad and Doron Zeilberger and arxiv.org .

First Written: June 3, 2020. This version: June 12, 2020.