COUNTING UP-DOWN WALKS

NOAM BUCKMAN

Abstract. This paper analyzes two types of up-down walks, Type I consisting of steps (1,0), (1,1), and (1,-1), and Type II consisting of steps (1/2,1), (1/2,-1), (1,0). Using recurrence equations and generating functions, we derive counting functions for both walks to enumerate the number of possible walks of length l.

1. Background

Analyzing walks has been of interest to mathematicians for centuries. Changing various characteristics of the walk, such as step size, direction, or dimension, leads to various patterns and results. One such walk, Dyck walks restricts walks to above the y-axis, such that the number of up steps are equal to down steps. As seen in Figure 1, Dyck walks can be interpreted as various configurations of mountains of a fixed length. More interestingly, the enumeration of Dyck paths leads to the special number sequence, Catalan numbers, which can also be used to count binary trees, word strings, and plane trees.

In this paper, we will study up-down walks, which like Dyck walks, consists of discrete steps that must remain above the y-axis. However, two significant differences make the up-down walks unique. First, up-down walks consists of horizontal/side steps in addition to the usual up and down steps, which do not increase the height of the walk. In addition, the first down step can only come after the last up step. As such, the walk will consist of a sequence of up steps followed by a sequence of down steps that will return back to y = 0 or ground level. Interspersed, side steps will be allowed since it does not change the height of the walk. In this paper, we will explore two types of up-down walks, Type I up-down walks, which consists of up, down, and side steps that all have width one, and Type II up-down walks where the up and down steps have a width of 1/2. A convenient method for representing

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{dyck_path.png}
\caption{A Dyck Path consisting of 10 steps.}
\end{figure}

\textit{Date: December 11, 2015.}
Figure 2. This is an example up-walk of length $l = 10$. Acceptable steps include up steps $(1,1)$, side steps $(1,0)$, and down steps $(1,-1)$.

The allowed steps is to use vector notation, where a $(1,1)$ step represents an up-step which ends a distance 1 to the right and 1 step upwards. For Type I, the allowed steps are: $(1,1)$, $(1,-1)$, $(1,0)$, representing the up, down, and side steps, respectively. In Type II walks, the allowed steps are: $(1,1), (1,-1), (1,0)$. For both of these walks, the goal is to formulate a counting function for the number of different walks for a given length walk, $l$.

In Figure 2 a running example of a Type I up-down walk of length $l = 10$ is shown. Notice that the up-down walk meets the characteristics of a generic up-down walk: it consists of up, down, and side steps, the first down step (second step) occurs after the last up step (sixth step), and no step goes below the y-axis.

The remainder of the paper is the follows: in Section 2 we define counting functions and in Section 3 we prove a few lemmas about the characteristics of an up-down walk. Then in Section 4 and Section 5, we derive a recursion for the Type I up-down walks and solve the recursion using generating functions. In Section 6, we introduce Type II up-down walks and then the recursion for the counting function in Section 7. Finally in Section 8 we solve the recursion to derive the Type II counting function.

2. Counting Up-Down Walks

Before proceeding in calculating our desired counting function, we will define a few functions and notation that will be used in this paper.

Definition 2.1. $\mathcal{U}_I^l$ is the set that consists of all possible Type I up-down walk of length $l$. Likewise, $\mathcal{U}_II^l$ is the set that consists of all possible Type II up-down walks of length $l$. 
Definition 2.2. \( f(l) \) is a counting function that counts the number of distinct up-down walks of length \( l \). In other words,

\[
f(l) = |U^I_l|
\]

Likewise, \( g(l) \) is the counting function for Type II walks and

\[
g(l) = |U^II_l|
\]

Example 2.3. Figure 3 shows the various Type I up-down walks for \( l = 1, 2, \) and 3. Since the sets \( U^I_1, U^I_2, \) and \( U^I_3 \) each contain 1, 2, and 4 distinct up-down walks, respectively then \( f(1) = 1, f(2) = 2, f(3) = 4. \)

The overall strategy for obtaining the counting function, \( f(l) \), for both Type I walks and Type II walks is to divide the walk into a sub-walk, and formulate a recursion for \( f(l) \). We will then use the technique of generating functions to solve these recursions, and obtain a close-formed formulation for \( f(l) \). We will apply this method to the first type of walk in more detail and then reapply this technique with the slightly modified Type II walks.

3. Type I Up-Down Walk

The primary observation that will motivate a recurrence is that an up-down walk of length \( l \) contains a smaller up-down subwalk, \( U_{sub} \) (not to be confused with \( U \)) that begins and ends at \( y = 1 \). This is because whatever walk exists above \( y = 1 \) must still meet the conditions of an up-down walk, namely that the first down step will occur after the last up step, and the walk does not go below the x-axis (in this case the axis is at \( y = 1 \)). This can also be confirmed in Figure 4 where the steps above \( y = 1 \) are boxed and form a subwalk of length \( l_{sub} \). Note that any subwalk will be preceded by an up-step and succeeded by a down-step. This must occur since the sub-walk is defined as being above \( y = 1 \), and since the original up-down walk must start and end at \( y = 0 \), an up and down step must occur before and
after the subwalk. Before proceeding with the recursion, a few characteristics of
the up-down walk and its sub-walk are shown.

\[ \text{Lemma 3.1.} \]
\[ l_{\text{sub}} \leq l - 2 \]

\textit{Proof.} As mentioned before, the sub-walk must be preceded by an up-step and
succeeded by a down-step. Since there are only \( l \) steps in the entire up-down walk
and two of them are an up-step and down step (each with width 1, there can only
be up to \( l - 2 \) steps in the subwalk itself. \( \square \)

\[ \text{Lemma 3.2.} \quad \text{Let} \ h \ \text{be the number of horizontal steps} \ \text{that lie on the} \ x-axis, \ \text{then:} \]
\[ h = l - l_{\text{sub}} - 2 \]

\textit{Proof.} The subwalk must be preceded by an up-step and succeeded by a down-step,
denoted \( u \) and \( d \), respectively. Since an up-down walk may not go beneath \( y = 0 \)
there can not be an up-step before \( u \). In addition, since no down-steps can precede
an up-step, there can not be a down-step before \( u \). Thus the only steps allowed
before \( u \) are horizontal steps. By the same logic, namely no up-steps or down-steps
can occur after \( d \), the only steps allowed after \( d \) are horizontal steps. Since the
only possible steps allowed are horizontal steps, the number of horizontal steps are
\( l - l_{\text{sub}} - 2 \). \( \square \)

\[ \text{Example 3.3.} \quad \text{Figure 5 shows two different up-down walks with length 10. The}
\text{top up-down walk length contains the smallest possible subwalk, with} \ l_{\text{sub}} = 0.
\text{The bottom up-down walk in contrast is the largest subwalk which has length}
\ l_{\text{sub}} = l - 2 = 8 \ \text{as described by (3.1). In addition, the number of horizontal side}
\text{steps for both up-down walks can be described by (3.2).} \]

\[ \text{Lemma 3.4.} \quad \text{For a given number of horizontal segments,} \ h, \ \text{and sub-walk} \ U_{\text{sub}},
\text{there will be} \ h + 1 \ \text{different possible permutations or walks.} \]

\textit{Proof.} Proof by bijection. We can represent the entire subwalks as a dot and the
placements of the horizontal steps as dashes.. There is a bijection between the
number of ways of up-down walks with a given \( U_{\text{sub}} \) and the number of ways to
arrange a dot and \( h \) dashes. This bijection is shown in Figure 6 where both the
inputs and outputs of the bijection are shown. Since there are \( h + 1 \) different ways to organize a line and \( h \) dashes, then the there are \( h + 1 \) subwalks for a given \( U_{sub} \).

## 4. Recursion

Motivated by the sub-walks above \( y = 1 \), we will look for a recursion of the form:

\[
(4.1) \quad f(l) = c_1 f(l - 1) + c_2 f(l - 2) \ldots c_n
\]

We will utilize the first three lemmas to formulate a recursion of the form above. By 3.1, walks of length \( l \) can contain subwalks of length \( l_{sub} = 0, 1, \ldots l - 2 \) and there will be a total of \( f(0), f(1), \ldots f(l - 2) \) unique subwalks of lengths \( l_{sub} \). In addition, by 3.3 and 3.2, each subwalk corresponds to \( l - l_{sub} - 1 \) unique up-down walks. Summing over the possible lengths of subwalks we arrive at the recursion:

\[
(4.2) \quad f(l) = \sum_{k=0}^{l-2} f(k) \cdot (l - k - 1)
\]

Not included in this recursion is the up-down walk with no subwalk at all. There is conveniently only one such walk which is the up-down walk with only horizontal steps. Accounting for this up-down walk yields the recursion equation

\[
(4.3) \quad f(l) = 1 + \sum_{k=0}^{l-2} f(k) \cdot (l - k - 1)
\]

Expanding the recursion we get:

\[
(4.4) \quad f(l) = f(l - 2) + 2f(l - 3) + 3f(l - 4) \ldots (l - 2)f(0) + 1
\]
To solve the recurrence, we simplify the recurrence to fewer terms by manipulating (4.4) until we get a recursion of only two or three terms.

We express \( f(l - 1) \) using (4.4) and subtract from \( f(l) \),

\[
\begin{align*}
\quad f(l) - f(l - 1) &= f(l - 2) + 2f(l - 3) + 3f(l - 4) + \cdots + (l - 2)f(0) + 1 \\
&\quad - (f(l - 3) + 2f(l - 4) + \cdots + (l - 3)f(0) + 1) \\
&\quad = f(l - 2) + f(l - 3) + f(l - 4) + \cdots + f(0) \\
&\quad - (f(l - 1) - f(l - 2)) \\
&\quad = f(l - 2) + f(l - 3) + f(l - 4) + \cdots + f(0) \\
&\quad - f(l - 1) + f(l - 2)
\end{align*}
\]

This expression can be further simplified by expressing \( f(l - 1) - f(l - 2) \) using (4.5) and subtracting,

\[
\begin{align*}
\quad f(l) - 2f(l - 1) + f(l - 2) &= f(l - 2) \\
&\quad - f(l - 1) + f(l - 2)
\end{align*}
\]

Simplifying, we get the final recursion equation for Type I walks as

\[
\quad f(l) = 2f(l - 1)
\]

Note that this recursion does not work if we include \( f(0) = 1 \) (i.e. there is only one walk of size \( l = 1 \)) since \( f(1) \neq 2f(0) \), however, it does hold true if we only consider \( l > 0 \). Thus we will first solve the recurrence

\[
\quad \tilde{f}(l) = 2\tilde{f}(l)
\]
where $\tilde{f}(0) = 1$, $\tilde{f}(1) = 2$ and then solve the recursion for $f(l)$. (Note that in the original recurrence, $f(0) = f(1) = 1$ and $f(2) = 2$.)

5. Solving Type I Recurrence with Generating Function

In order to solve the recursion, we utilize generating functions. By converting the recursion into an equation of generating functions, we can solve for the generating function, and then derive from the generating function the counting function $\tilde{f}(l)$ and subsequently the original counting function $f(l)$.

**Definition 5.1.** Let $f(l)$ be the counting function of up-down walks, then the generating function $F(x)$ is

$$F(x) = \sum_{l=0}^{\infty} f(l)x^l$$

**Lemma 5.2.** The generating function $F(x)$ for a recurrence $f_n$ of the form:

$$f_n = \alpha f_{n-1} + \beta f_{n-2} + \gamma f_{n-3}$$

corresponds to the equation:

$$F(x) = \alpha x F(x) + \beta x^2 F(x) + \gamma x^3 F(x) + p(x)$$

where $p(x)$ is a polynomial of degree at most the number of terms in the recurrence, that makes the equation work for the first few elements.\(^1\)

Applying Lemma 5.2 to the modified Type I recurrence, we get that

$$\tilde{F}(x) = 2x\tilde{F}(x) + c$$

$$(1 - 2x)\tilde{F}(x) = c$$

$$\tilde{F}(x) = \frac{c}{1 - 2x}$$

where $c$ is a constant that allows the recursion to work the first terms of the sequence. The next step is to write the generating function in terms of a sequence

$$\tilde{F}(x) = \sum_{l=0}^{\infty} c(2x)^l = \sum_{l=0}^{\infty} c2^l x^l = \sum_{l=0}^{\infty} \tilde{f}(l)x^l$$

The solution of the recurrence, must thus be in the form $\tilde{f}(l) = c2^l$. Using the initial value $f(0) = 1$ as our initial value we get

$$\tilde{f}(l) = 2^l$$

and the generating function is

$$\tilde{F}(x) = \frac{1}{1 - 2x}$$

---

Note that the only difference between $f(l)$ and $\tilde{f}$ is a right shift, i.e. $\tilde{f}(l) = \tilde{f}(l+1)$ and $f(0) = 1$. The counting function can be easily modified

\begin{equation}
(5.4) \quad f(l) = \begin{cases} 
1, & l = 0, \\
2^{l-1}, & l > 0
\end{cases}
\end{equation}

To account for the right shift we can just write $F(x) = x\tilde{F}(x)$ and then add $f(0) = 1$, so that the final Type I generating function is

\begin{equation}
(5.5) \quad F(x) = \frac{x}{1-2x} + 1 = \frac{1-x}{1-2x}
\end{equation}

Finally, we can verify the first four terms of the counting function and generating function: $f(0) = 1, f(1) = 2^{1-1} = 1, f(2) = 2^{2-1} = 2, f(3) = 2^{3-1} = 4, f(4) = 2^{4-1} = 8$. and

$$\frac{1-x}{1-2x} = 1 + x + 2x^2 + 4x^3 + 8x^4 \ldots$$

6. Type II Walks

As mentioned before, Type II up-down walks are identical except that the walk consists of the following steps: side steps with a width of one $(1,0)$, up steps with width of one-half $(\frac{1}{2}, 1)$, and down steps with width of one-half $(\frac{1}{2}, -1)$. The normal restrictions on the walk will still apply: all walks must remain above the y-axis and the first down-step must occur after the last up-step.

In Figure 7 we show a few examples of Type II up-down walks, displaying walks of length 1 and walks of length 2. Since Type I and Type II walks are almost identical, we will analyze them in a similar fashion as Type I walks, using a recursive strategy and solving the recursion with generating functions.

Like before, a Type II up-down walk can be broken up into a sub-walk that begins after the first up-step and ends at the last down-step. As shown in Figure 8 above the line $y = 0$ is a subwalk that is also a Type II up-down walk. The goal then is to find $g(l)$, the counting function for the number of Type II up-down walks of length $l$. For a given subwalk of length $l_{sub}$, the number of of total up-down walks are $l - l_{sub}$. Notice that in Type I walks, there were $l - l_{sub} - 1$ up-down walks.
Figure 8. A subwalk of length $l - 2 - h$ is boxed starting about $y = 1$

walks which is one less up-down walk, which makes sense since the first up-step and last down-step only take up a total of 1 (half for each) in the horizontal direction instead of 2 as before, allowing for more possible placements of $U_{sub}$. Now that we know the function, we can iterate over all possible configurations of the first level and obtain the following formulation:

(6.1) \[ g(l) = \sum_{k=0}^{l-1} g(k) \cdot (l - k) \]

Finally, we must take into the account that the up-down walk never goes above $y = 0$, which is the single up-walk that consists of only horizontal steps and no ups or downs. Accounting for this walk, we get the following recursion for Type II walks:

(6.2) \[ g(l) = 1 + \sum_{k=0}^{l-1} g(k) \cdot (l - k) \]

7. Solving the Recurrence

Expanding the recurrence above we get that:

(7.1) \[ g(l) = g(l - 1) + 2g(l - 2) + 3g(l - 3) \cdots + l \cdot g(0) + 1 \]

In order to simplify the recurrence above, we manipulate the terms with some simple arithmetic and substitutions.

(7.2) \[ g(l) - g(l - 1) = g(l - 1) + 2g(l - 2) + 3g(l - 3) \cdots + l \cdot g(0) + 1 \]
\[ - \left( g(l - 2) + 2g(l - 3) + \cdots + (l - 1)g(0) + 1 \right) \]
By expressing \( g(l-1) - g(l-2) \) using (7.2) and subtracting it from \( g(l) - g(l-1) \) we obtain:

\[
\begin{align*}
g(l) - g(l-1) &= g(l-1) + g(l-2) + g(l-3) + \cdots + g(0) \\
= (g(l-1) - g(l-2)) &+ (g(l-2) + g(l-3) + \cdots + g(0)),
\end{align*}
\]

(7.3) \( g(l) - 2g(l-1) + g(l-2) = g(l-1) \)

Simplifying, we get the final recursion for Type II up-down walks: and simplifying, we get the final recursion for Type II walks

(7.4) \( g(l) = 3g(l-1) - g(l-2) \)

8. Generating Function of Type II

We can now apply Lemma 5.2 to (7.4) to get the following generating function

(8.1) \( G(x) = \frac{p(x)}{1 - 3x + x^2} \)

where \( p(x) \) guarantees that the recursion matches the initial conditions \( g(0) = 1 \) and \( g(1) = 2 \).

In order to determine \( p(x) \) and the counting function \( g(l) \) we utilize partial fractions to rewrite \( G(x) \) as

\[
G(x) = \frac{c_1}{1 - r_1x} + \frac{c_2}{1 - r_2x}
\]

where \( r_1 = \frac{3 + \sqrt{5}}{2}, \quad r_2 = \frac{3 - \sqrt{5}}{2} \), and \( c_1 \) and \( c_2 \) ensure that the recurrence fit the initial conditions of the recurrence. Both fractions can then be expanded using its power series to obtain

(8.2) \( g(n) = c_1 \left( \frac{3 + \sqrt{5}}{2} \right)^n - c_2 \left( \frac{3 - \sqrt{5}}{2} \right)^n \)

Since the first two terms of the counting functions are \( g(1) = 1 \) and \( g(2) = 2 \), we can solve the system of equations and obtain the final counting function:

(8.3) \( g(l) = \left( \frac{1}{2} - \frac{1}{2\sqrt{5}} \right) \left( \frac{3 + \sqrt{5}}{2} \right)^l + \left( \frac{1}{2} + \frac{1}{2\sqrt{5}} \right) \left( \frac{3 - \sqrt{5}}{2} \right)^l \)

and generating function

(8.4) \( G(x) = \frac{\frac{1}{2} - \frac{1}{2\sqrt{5}}}{1 - (3 + \sqrt{5})x} + \frac{\frac{1}{2} + \frac{1}{2\sqrt{5}}}{1 - (3 - \sqrt{5})x} = \frac{1 - 2x}{1 - 3x + x^3} \)

If we use our counting functions to enumerate the first few Type II up-down walks, we see yet another famous sequence, the odd-th Fibonacci sequence (i.e. every other number): 1, 2, 5, 13, 34, ...