A generalized Goulden–Jackson cluster method and lattice path enumeration

Yan Zhuang
Department of Mathematics
Brandeis University
zhuangy@brandeis.edu

December 29, 2015

Abstract

The Goulden–Jackson cluster method is a powerful tool for obtaining generating functions for counting words in a free monoid by occurrences of a set of subwords. We introduce a generalization of the cluster method for monoid networks, which generalize the combinatorial framework of free monoids. As a sample application of the generalized cluster method, we compute bivariate and multivariate generating functions counting Motzkin paths—both with height bounded and unbounded—by statistics corresponding to the number of occurrences of various subwords, yielding both closed-form and continued fraction formulae.

1. Introduction

Given a finite or countably infinite set A, let A^* be the set of all finite sequences of elements of A, including the empty sequence. We call A an *alphabet*, the elements of A *letters*, and the elements of A^* *words*. By defining an associative binary operation on two words by concatenating them, we see that A^* is a monoid under the operation of concatenation (where the empty word is the identity element), and we call A^* the *free monoid* on A. The length $l(\alpha)$ of a word $\alpha \in A^*$ is the number of letters in α. For $\alpha, \beta \in A^*$, we say that β is a *subword* of α if $\alpha = \gamma_1\beta\gamma_2$ for some $\gamma_1, \gamma_2 \in A^*$, and in this case we also say α *contains* β.

More generally, a *free monoid* is a monoid isomorphic to a free monoid on some alphabet. The combinatorial framework of free monoids is useful for the study of combinatorial objects that can be uniquely decomposed into sequences of "prime elements", corresponding to letters in an alphabet. This framework can furthermore be generalized using what are called "monoid networks", which were first introduced by Gessel [8, Chapter 6] in a slightly different yet equivalent form called "G-systems".[1] Roughly speaking, a monoid network consists of a digraph G with each arc assigned a set of letters from an alphabet A, in which

[1] The term "G-system" was dropped at the request of Ira Gessel, who prefers the name "monoid network" given by the author.

the set of sequences of arcs in G is given a monoid structure and is equipped with a monoid homomorphism.

The Goulden–Jackson cluster method allows one to determine the generating function for words in a free monoid A^* by occurrences of words in a set $B \subseteq A^*$ as subwords in terms of the generating function for what are called "clusters" formed by words in B, which is easier to compute. As its name suggests, this celebrated result was first given by Goulden and Jackson in [9]. The cluster method has seen a number of extensions and generalizations [14, 4, 19, 10, 18, 12, 1], and the cluster method itself can be viewed as a generalization of the Carlitz–Scoville–Vaughan theorem, which allows one to count words in a free monoid avoiding a specified set of length 2 subwords.

In this paper, we give a new generalization of the Goulden–Jackson cluster method of a different flavor: we generalize the cluster method to monoid networks, which gives a way of counting words in A^* corresponding to walks between two specified vertices in G (that is, words in a regular language if the alphabet A is finite) by occurrences of subwords in a set B. Then the original version of the cluster method corresponds to the special case in which G consists of a single vertex with a loop to which the entire alphabet A is assigned.

The organization of this paper is as follows. In Section 2, we give an expository account of the original Goulden–Jackson cluster method. In Section 3, we introduce the combinatorial framework of monoid networks and present our generalization of the cluster method for monoid networks. Finally, in Section 4, we demonstrate how our monoid network version of the cluster method can be used to tackle problems in lattice path enumeration.

Although many types of lattice paths can be represented as walks in certain digraphs, in this paper we focus on Motzkin paths, which are paths in $\mathbb{Z}$ beginning and ending at 0 with steps -1, 0, and 1 (also called "down steps", "flat steps", and "up steps", respectively). We consider both regular Motzkin paths and Motzkin paths bounded by height, and our results include bivariate and multivariate generating functions counting these paths by ascents, plateaus, peaks, and valleys—all of which are statistics that are determined by occurrences of various subwords in the underlying word of the Motzkin path—as well as generating functions for Motzkin paths with restrictions on the heights at which these subwords can occur, yielding both closed-form and continued fraction formulae. Several interesting identities are uncovered along the way.

2. The Goulden–Jackson cluster method

We begin this section with a motivating problem: let A be a finite or countably infinite alphabet and suppose that we want to count words in A^* that do not contain a specified set B of forbidden subwords of length at least 2. The Goulden–Jackson cluster method allows us to count this restricted set of words by counting "clusters" formed by words in B, which we shall define shortly.

Given a word $\alpha = a_1 a_2 \cdots a_n \in A^*$ (where the a_i are letters) and a set $B \subseteq A^*$, we say that (i, β) is a *marked subword* of α if $\beta \in B$ and

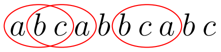
$$\beta = a_i a_{i+1} \cdots a_{i+l(\beta)-1},$$

that is, β is a subword of α starting at position i. Moreover, we say that (α, S) is a *marked word* (on α) if $\alpha \in A^*$ and S is any set of marked subwords of α.

For example, suppose that $A = \{a, b, c\}$ and $B = \{abc, bca\}$. Then

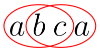
$$\{abcabbcabc, \{(1, abc), (2, bca), (6, bca)\}\}\,, \tag{1}$$

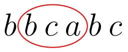
is a marked word which can also be displayed as



.

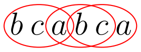
The concatenation of two marked words is defined in the obvious way. For example, (1) can be obtained by concatenating $\{abca, \{(1, abc), (2, bca)\}\}$ and $\{bbcabc, \{(2, bca)\}\}$, i.e.,



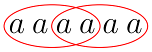
and


.

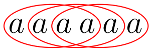
A marked word (on α) is called a *cluster* (on α) if it is not a concatenation of two nonempty marked words. So, (1) is not a cluster, but



is a cluster. Two additional examples of clusters, using $A = \{a\}$ and $B = \{aaaa\}$, are



and



,

which we include to emphasize the fact that a cluster is not required to be "maximal" in the sense that every possible marked subword must be included. If a word α has only one possible cluster, then there is no need to indicate the positions of the marked subwords and we say (by abuse of language) that the only cluster on α is itself.

Before formally presenting the cluster method, we introduce some additional notation. For a word $\alpha \in A^*$, let $\text{bad}(\alpha)$ be the number of occurrences in α of words in B and let C_α be the set of all clusters on the word α. Given a cluster c, let $\text{mk}(c)$ be the number of marked subwords in c. Given an indeterminate t that commutes with all of the letters in A, define

$$F(t) = \sum_{\alpha \in A^*} \alpha t^{\text{bad}(\alpha)}$$

and

$$L(t) = \sum_{\alpha \in A^*} \alpha \sum_{c \in C_\alpha} t^{\text{mk}(c)},$$

so that $F(t)$ is the generating function for words in A^* by the number of occurrences of words in B, and $L(t)$ is the generating function for clusters by the number of marked subwords. Both $F(t)$ and $L(t)$ are elements of the formal power series algebra $K\langle\langle A^*\rangle\rangle[[t]]$, where K is a field of characteristic zero (which we can take to be $\mathbb{C}$) and $K\langle\langle A^*\rangle\rangle$—called the *total algebra* of A^* over K—is the algebra of formal sums of words in A^* with coefficients in K.

Theorem 1 (Goulden–Jackson cluster method, version 1)**.** *Let A be an alphabet and let $B \subseteq A^*$ be a set of words of length at least 2. Then,*

$$F(t) = \left(1 - \sum_{a \in A} a - L(t-1)\right)^{-1}.$$

Proof. We prove the equivalent statement

$$F(1+t) = \left(1 - \sum_{a \in A} a - L(t)\right)^{-1}.$$

We have

$$\begin{aligned} F(1+t) &= \sum_{\alpha \in A^*} \alpha (1+t)^{\mathrm{bad}(\alpha)} \\ &= \sum_{\alpha \in A^*} \alpha \sum_{k=0}^{\infty} \binom{\mathrm{bad}(\alpha)}{k} t^k \\ &= \sum_{\alpha \in A^*} \alpha \sum_{S \subseteq B_\alpha} t^{|S|}, \end{aligned} \tag{2}$$

where B_α is the set of occurrences of words in B in α. Note that (2) counts marked words weighted by the number of marked subwords that it contains, and from here it is easy to see that

$$\begin{aligned} F(1+t) &= \sum_{\alpha \in A^*} \alpha \sum_{S \subseteq B_\alpha} t^{|S|} \\ &= \left(1 - \sum_{a \in A} a - L(t)\right)^{-1} \end{aligned}$$

since every marked word is uniquely built from a sequence of letters in A and clusters. □

We indicate three specializations of Theorem 1 that are of particular importance:

- By setting $t = 0$, we obtain

  $$\left(1 - \sum_{a \in A} a - L(-1)\right)^{-1}$$

  as the generating function for words in A^* that do not contain any words in B, which solves the problem posed at the beginning of this subsection, assuming that we can compute the cluster generating function $L(t)$.

- If every word in B has length exactly 2, then setting $t = 0$ yields the Carlitz–Scoville–Vaughan theorem, which was independently discovered by Fröberg [7, Section 4], Carlitz et al. [2, Theorem 7.3], and Gessel [8, Theorem 4.1]. In fact, Chapters 4 and 5 of Gessel's doctoral thesis [8] are devoted to the Carlitz–Scoville–Vaughan theorem and its many enumerative applications.

- By setting $t = 1$, we obtain the free monoid identity

$$\sum_{\alpha \in A^*} \alpha = \left(1 - \sum_{a \in A} a\right)^{-1}. \tag{3}$$

More generally, we can assign each word in B its own indeterminate. Write $B = \{\beta_1, \beta_2, \dots\}$ so that the words in B are ordered. (Here, B is presented as countably infinite although in most applications it is finite.) Given a word $\alpha \in A^*$, let $\mathrm{bad}_i(\alpha)$ be the number of occurrences of β_i in α, and given a cluster c, let $\mathrm{mk}_i(c)$ be the number of marked subwords in c of the form (j, β_i). Let $t_1, t_2, \dots$ be indeterminates that commute with each other and with the letters of A, and define the generating functions

$$F(t_1, t_2 \dots) = \sum_{\alpha \in A^*} \alpha \prod_{k=1}^{\infty} t_k^{\mathrm{bad}_k(\alpha)}$$

and

$$L(t_1, t_2, \dots) = \sum_{\alpha \in A^*} \alpha \sum_{c \in C_\alpha} \prod_{k=1}^{\infty} t_k^{\mathrm{mk}_k(c)}.$$

Then we have a refinement of Theorem 1, which follows by the same reasoning as before:

Theorem 2 (Goulden–Jackson cluster method, version 2)**.** *Let A be an alphabet and let $B = \{\beta_1, \beta_2, \dots\} \subseteq A^*$ be a set of words of length at least 2. Then,*

$$F(t_1, t_2 \dots) = \left(1 - \sum_{a \in A} a - L(t_1 - 1, t_2 - 1, \dots)\right)^{-1}.$$

The statement of Theorem 2 uses an infinite set B and infinitely many indeterminates t_i, but it is clear that the finite case works as well. The number of indeterminates also does not need to equal the number of words in B; for example, we can have $B = \{\beta_1, \dots, \beta_k\}$ along with two indeterminates t_1 and t_2, and attach t_1 to all β_i with i odd and attach t_2 to all β_i with i even.

As an example, let $A = \{a, b, c\}$ and suppose that we want to count words in A^* by occurrences of $\beta_1 = acb$ and $\beta_2 = bc$. Then the only clusters are acb, bc, and $acbc$, so

$$L(t_1, t_2) = acbt_1 + bct_2 + acbct_1t_2$$

and by Theorem 2, we obtain

$$F(t_1, t_2) = (1 - a - b - c - acb(t_1 - 1) - bc(t_2 - 1) - acbc(t_1 - 1)(t_2 - 1))^{-1} \tag{4}$$

as the generating function for words in A^* by occurrences of acb and bc. By setting $t_1 = t_2 = 0$, we obtain

$$(1 - a - b - c + acb + bc - acbc)^{-1} \tag{5}$$

as the generating function for words in A^* which contain neither acb nor bc.

Now, let x be an indeterminate that commutes with t_1 and t_2. If we apply the homomorphism sending each of the letters to x, we obtain the generating functions

$$\frac{1}{1-3x-x^2(t_2-1)-x^3(t_1-1)-x^4(t_1-1)(t_2-1)}$$

and

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

from (4) and (5), respectively, which keep track of these words by length.

We say that the set B is *reduced* if no word $\beta \in B$ is a subword of another word β' in B. Although the cluster method as presented above works regardless of whether B is reduced, Goulden and Jackson gave a formula in their original paper [9] for the cluster generating function when A and B are finite sets with B reduced. A set B of forbidden subwords can always be replaced by a reduced set and still yield the same restricted set of words; if $\beta \in B$ is a subword of $\beta' \in B$, then we can remove β' from B because containing β' implies containing β. However, the criterion of having a reduced set can be an issue if we want to count words by occurrences of subwords (that is, without setting $t = 0$). For instance, we would not be able to use Goulden and Jackson's formula to compute the cluster generating function given $B = \{aba, abab\}$ since aba is a subword of $abab$.

As part of [14], Noonan and Zeilberger wrote a Maple package that handles the case where B is arbitrary (i.e., not necessarily reduced), but without a detailed explanation of their algorithms. Bassino et al. [1] later gave an explicit expression for the cluster generating function in the non-reduced case. We omit these formulae of Goulden and Jackson and Bassino et al. because the cluster generating functions in Section 4 of this paper will require essentially no computation.

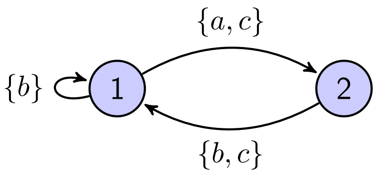
3. Our generalization of the cluster method

3.1. Monoid networks

Throughout this section, fix a field K of characteristic zero and let A be a finite or countably infinite alphabet. As in the previous section, $K\langle\langle A^*\rangle\rangle$ is the total algebra of A^* over K. We also let $\mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$ denote the algebra of $m \times m$ matrices with entries in $K\langle\langle A^*\rangle\rangle$.

Let G be a digraph on the vertex set $[m]$ such that each arc (i, j) of G is assigned a set of letters $P_{i,j}$ in A, and let P be the set of all pairs (a, e) where $e = (i, j)$ is an arc of G and $a \in P_{i,j}$. Define $\overrightarrow{P^*} \subseteq P^*$ to be the subset of all sequences $\alpha = (a_1, e_1)(a_2, e_2)\cdots(a_n, e_n)$ where $e_1e_2\cdots e_n$ is a walk in G. Given $\alpha = (a_1, e_1)(a_2, e_2)\cdots(a_n, e_n)$ in $\overrightarrow{P^*}$, we define $\rho(\alpha) = a_1a_2\cdots a_n$ to be the word obtained by projecting onto A^* and let $E(\alpha) = (i, j)$ where i and j are the initial and terminal vertices, respectively, of the walk $e_1e_2\cdots e_n$.

For example, consider the following:



Here $P = \{(b,(1,1)),(a,(1,2)),(c,(1,2)),(b,(2,1)),(c,(2,1))\}$. One element of $\overrightarrow{P^*}$ is $\alpha = (b,(2,1))(b,(1,1))(a,(1,2))$, and so $\rho(\alpha) = bba$ and $E(\alpha) = (2,2)$.

We say that (G,P) a *monoid network* on A^* if for all nonempty $\alpha, \beta \in \overrightarrow{P^*}$, if $\rho(\alpha) = \rho(\beta)$ and $E(\alpha) = E(\beta)$ then $\alpha = \beta$. That is, the same word in A^* cannot be obtained by traversing two different walks with the same initial and terminal vertices. It is easy to see that (G,P) in the example given above is a monoid network.

We can very naturally represent words in $\overrightarrow{P^*}$ using matrices. For each element $p = (a,(i,j)) \in P$, we associate p with the $m \times m$ matrix M_p with a in the (i,j) entry and 0 everywhere else, which defines a monoid homomorphism $\lambda : P^* \to \mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$, where we consider the codomain as the multiplicative monoid of the algebra $\mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$. Applying λ to the empty word 1 gives the $m \times m$ identity matrix I_m.

If $\alpha \in \overrightarrow{P^*}$ and $E(\alpha) = (i,j)$, then $\lambda(\alpha)$ is the $m \times m$ matrix with $\rho(\alpha)$ in the (i,j) entry and 0 everywhere else; we denote this matrix M_α. If $\alpha \notin \overrightarrow{P^*}$, then $M_\alpha = \lambda(\alpha) = 0_m$, the $m \times m$ zero matrix.

Returning to the example above, the matrices M_p are

$$\begin{bmatrix} b & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & c \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ b & 0 \end{bmatrix}, \text{ and } \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix},$$

and for $\alpha = (b,(2,1))(b,(1,1))(a,(1,2))$, we have

$$\lambda(\alpha) = \begin{bmatrix} 0 & 0 \\ 0 & bba \end{bmatrix}.$$

We then extend λ by linearity to an algebra homomorphism $K\langle\langle P^*\rangle\rangle \to \mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$, which we also call λ by a slight abuse of notation. Given a monoid network (G,P) and a subset $S \subseteq A^*$, let $\overrightarrow{\Gamma}_G(S) \in \mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$ be the matrix whose (i,j) entry is the generating function for words in S that can be obtained by traversing a walk from i to j in G. It is clear that

$$\overrightarrow{\Gamma}_G(S) = \sum_{\alpha \in V} M_\alpha$$

where V is the set of all words $\alpha \in P^*$ such that $\rho(\alpha) \in S$.

If the alphabet A is finite, then the idea of monoid networks may seem too similar to finite-state automata to warrant its own definition, but our approach is novel and is based on the monoid structure of P^* and the power of the homomorphism λ. Moreover, our construction generalizes the combinatorial framework of free monoids, hence the name "monoid network". For example, the following is an elementary result traditionally proven using the transfer-matrix method (see [16, Section 4.7] or [6, Section V.6]), but we can give a very simple proof using the homomorphism λ:

Theorem 3. *Suppose that (G,P) is a monoid network on A^*. Then*

$$\overrightarrow{\Gamma}_G(A^*) = \Big(I_m - \sum_{p \in P} M_p\Big)^{-1}.$$

Proof. Take

$$\sum_{\alpha \in P^*} \alpha = \left(1 - \sum_{p \in P} p\right)^{-1},$$

which is (3) applied to the free monoid P^*, and then apply λ to both sides of the equation. □

Our proof of the generalized Goulden–Jackson cluster method presented later in this section is of a similar flavor.

Continuing with the example above, we have

$$\overrightarrow{\Gamma}_G(A^*) \quad = \quad \begin{bmatrix} 1-b & -a-c \\ -b-c & 1 \end{bmatrix}^{-1}$$

by Theorem 3. If we want the generating function for words by length that can be obtained by traversing a walk from 1 to 2 in (G, P), then we apply to $\overrightarrow{\Gamma}_G(A^*)$ the homomorphism sending each of the letters to x to obtain the matrix

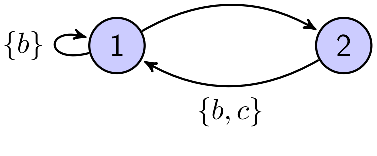
$$\begin{bmatrix} 1-x & -2x \\ -2x & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \dfrac{1}{1-x-4x^2} & \dfrac{2x}{1-x-4x^2} \\ \dfrac{2x}{1-x-4x^2} & \dfrac{1-x}{1-x-4x^2} \end{bmatrix}$$

and then take the $(1, 2)$ entry.

We give one last remark before presenting our generalization of the cluster method. In the paper [20], the present author generalizes a theorem by Gessel which enables one to count words and permutations with restrictions on the lengths of their increasing runs. This generalized run theorem allows for a much wider variety of restrictions on increasing run lengths; specifically, these restrictions are those which can be encoded by a special type of digraph called a "run network". Run networks are in fact monoid networks where the alphabet A is $\mathbb{P}$ (the positive integers), but the language of monoid networks was not used in the exposition of [20] because the proof of the generalized run theorem presented in that paper did not make use of the monoid structure of $\mathbb{P}^*$ or the homomorphism λ. However, it is possible to prove the generalized run theorem—albeit in a more complicated way—using a monoid network version of the Carlitz–Scoville–Vaughan theorem.

3.2. The Goulden–Jackson cluster method for monoid networks

To motivate our generalization of the Goulden–Jackson cluster method, let us combine two previous examples and suppose that we want to count words on the alphabet $A = \{a, b, c\}$ that satisfy two conditions. First, these words cannot contain any occurrences of $\beta_1 = acb$ and $\beta_2 = bc$, and second, these words must be obtainable by traversing a walk from vertex 1 to vertex 2 in the following monoid network (G, P):



We can do this using our monoid network version of the Goulden–Jackson cluster method, which we now present in full generality. Let A be an alphabet and let $B = \{\beta_1, \beta_2, \dots\} \subseteq A^*$ be a set of words. Moreover, let (G, P) be a monoid network with m vertices, and for each u, let $\overrightarrow{B}_u$ be the set of all words α in $\overrightarrow{P^*}$ with $\rho(\alpha) = \beta_u$, and let $\overrightarrow{B} = \bigcup_{u=1}^{\infty} \overrightarrow{B}_u$.

Define $\overrightarrow{F}_G(t_1, t_2, \dots)$ to be the $m \times m$ matrix whose (i, j) entry is the sum

$$\sum_{\alpha} \rho(\alpha) \prod_{k=1}^{\infty} t_k^{\mathrm{bad}_k(\rho(\alpha))}$$

over all $\alpha \in \overrightarrow{P^*}$ with $E(\alpha) = (i, j)$, which is the same as the sum

$$\sum_{\alpha} \alpha \prod_{k=1}^{\infty} t_k^{\mathrm{bad}_k(\alpha)}$$

over all $\alpha \in A^*$ that can be obtained by traversing a walk from vertex i to vertex j in the monoid network (G, P). Furthermore, define

$$\overrightarrow{L}_G(t_1, t_2, \dots) = \sum_{\alpha \in \overrightarrow{P^*}} M_\alpha \sum_{c \in C_\alpha} \prod_{k=1}^{\infty} t_k^{\mathrm{mk}_k(c)},$$

where C_α is the set of all clusters (formed by words in $\overrightarrow{B}$) on the word α, and $\mathrm{mk}_u(c)$ is the number of marked subwords in c of the form (v, γ) with $\gamma \in \overrightarrow{B}_u$. We will refer to $\overrightarrow{L}_G(t_1, t_2, \dots)$ as the *cluster matrix*.

Theorem 4 (Goulden–Jackson cluster method for monoid networks)**.** *Let A be an alphabet and let $B = \{\beta_1, \beta_2, \dots\} \subseteq A^*$ be a set of words of length at least 2. Also, let G be a digraph on $[m]$ and let (G, P) be a monoid network on A^*. Then,*

$$\overrightarrow{F}_G(t_1, t_2 \dots) = \left(I_m - \sum_{p \in P} M_p - \overrightarrow{L}_G(t_1 - 1, t_2 - 1, \dots) \right)^{-1}.$$

Proof. First, apply the original Goulden–Jackson cluster method (Theorem 2) for the alphabet P and the set $\overrightarrow{B}$, where we attach the indeterminate t_u to each word in $\overrightarrow{B}_u$. Then applying the homomorphism λ yields the desired result. □

As before, the set of words in B need not be infinite, and the number of indeterminates can be less than the number of words in B. It is also possible to alter the cluster matrix to only include clusters occurring at specified positions in the monoid network, which we do so in Section 4 to count Motzkin paths with no occurrences of subwords at specified heights.

We mention three specializations which are completely analogous to those given after Theorem 1:

- By setting each indeterminate equal to 0, we obtain

$$\left(I_m - \sum_{p \in P} M_p - \overrightarrow{L}_G(-1, -1, \dots) \right)^{-1}$$

as the $m \times m$ matrix whose (i,j) entry is the sum $\sum_\alpha \alpha$ over all $\alpha \in A^*$ that can be obtained by traversing a walk from vertex i to vertex j in the monoid network (G,P) and contain no occurrences of words in B.

- If every word in B has length exactly 2, then setting each indeterminate equal to 0 yields a monoid network version of the Carlitz–Scoville–Vaughan theorem.
- Setting each indeterminate equal to 1 gives an alternative proof for Theorem 3.

Observe that the original Goulden–Jackson cluster method corresponds to the special case in which the monoid network consists of a single vertex with a loop to which the entire alphabet A is assigned. Thus Theorem 4 can accurately be characterized as a generalization of the Goulden–Jackson cluster method.

Finally, we note that if the alphabet A is finite, then a monoid network gives the transition diagram of a nondeterministic finite automaton. Nondeterministic finite automaton are equivalent to deterministic finite automaton, and the transition diagram of a deterministic finite automaton is a monoid network. Therefore, Theorem 4 can be used to count words in a regular language by occurrences of a specified set of subwords. (For relevant definitions regarding formal languages and automata, see any introductory text on the theory of computation.)

Let us now complete the example from earlier. We have

$$\begin{aligned}\overrightarrow{L}_G(t_1,t_2) &= \begin{bmatrix} acb & 0 \\ 0 & 0 \end{bmatrix} t_1 + \begin{bmatrix} 0 & 0 \\ 0 & bc \end{bmatrix} t_2 + \begin{bmatrix} 0 & bc \\ 0 & 0 \end{bmatrix} t_2 + \begin{bmatrix} 0 & acbc \\ 0 & 0 \end{bmatrix} t_1t_2 \\ &= \begin{bmatrix} acbt_1 & bct_2 + acbct_1t_2 \\ 0 & bct_2 \end{bmatrix};\end{aligned}$$

indeed, recall that the only three clusters formed by the words *acb* and *bc* are *acb*, *bc*, and *acbc*, which can be obtained in the given monoid network by traversing walks with initial and terminal vertices indicated in the matrices above. Thus,

$$\begin{aligned}\overrightarrow{F}_G(t_1,t_2) &= \left(I_2 - \sum_{p \in P} M_p - \overrightarrow{L}_G(t_1-1,t_2-1) \right)^{-1} \\ &= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} b & a+c \\ b+c & 0 \end{bmatrix} - \begin{bmatrix} acb(t_1-1) & bc(t_2-1) + acbc(t_1-1)(t_2-1) \\ 0 & bc(t_2-1) \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} 1-b-acb(t_1-1) & -a-c-bc(t_2-1)-acbc(t_1-1)(t_2-1) \\ -b-c & 1-bc(t_2-1) \end{bmatrix}^{-1}.\end{aligned}$$

Now we apply the homomorphism sending each of the letters to x, yielding the matrix

$$\begin{bmatrix} 1-x-x^3(t_1-1) & -2x-x^2(t_2-1)-x^4(t_1-1)(t_2-1) \\ -2x & 1-x^2(t_2-1) \end{bmatrix}^{-1}$$

whose $(1,2)$ entry is

$$\frac{2x-(1-t_2)x^2+(1-t_1-t_2+t_1t_2)x^4}{1-x-(3+t_2)x^2+(2-t_1-t_2)x^3-(1-t_1-t_2+t_1t_2)x^5},$$

which is the generating function for words obtained by traversing a walk from vertex 1 to vertex 2 in the given monoid network, weighted by length, occurrences of acb, and occurrences of bc. Setting $t_1 = t_2 = 0$ gives the generating function

$$\frac{2x - x^2 + x^4}{1 - x - 3x^2 + 2x^3 - x^5}$$

for those words that do not contain any occurrences of acb or bc.

We also state a weighted version of Theorem 4. Let $\{w_a^{(i.j)} : (a,(i,j)) \in P\}$ be a set of weights that commute with each other, the indeterminates $t_1, t_2, \ldots$, and the letters in A. Set $w_a^{(i,j)} = 0$ if $(a,(i,j)) \notin P$. Given $\alpha = a_1 a_2 \cdots a_k \in A^*$ and $1 \le i,j \le m$, let $w^{(i,j)}(\alpha) = w_{a_1}^{e_1} \cdots w_{a_k}^{e_k}$ if there exists $\beta = (a_1,e_1)\cdots(a_k,e_k) \in \overrightarrow{P^*}$ such that $E(\beta) = (i,j)$ and $\rho(\beta) = \alpha$.

Define the map $\hat{\lambda} : P^* \to \mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$ by sending $p = (a,(i,j))$ to the matrix $\hat{M}_p$ with $w_a^{(i.j)}a$ in the (i,j) entry and 0 everywhere else. If $\alpha = (a_1,e_1)\cdots(a_n,e_n) \in \overrightarrow{P^*}$ and $E(\alpha) = (i,j)$, then $\hat{\lambda}(\alpha)$—which we also denote $\hat{M}_\alpha$—has $w_{a_1}^{e_1} \cdots w_{a_n}^{e_n}\rho(\alpha)$ in the (i,j) entry and 0 everywhere else, and if $\alpha \notin \overrightarrow{P^*}$ then $\hat{M}_\alpha = 0_m$. Again, $\hat{\lambda}$ extends to a homomorphism $K\langle\langle P^*\rangle\rangle \to \mathrm{Mat}_m(K\langle\langle A^*\rangle\rangle)$, which we also call $\hat{\lambda}$. Note that setting all of the weights equal to 1 gives $\hat{\lambda} = \lambda$.

Theorem 5 (Goulden–Jackson cluster method for monoid networks, weighted version)**.** *Let A be an alphabet and let $B = \{\beta_1, \beta_2, \ldots\} \subseteq A^*$ be a set of words of length at least 2; let G be a digraph on $[m]$ and let (G,P) be a monoid network on A^*; let $\hat{F}_G(t_1, t_2, \ldots)$ be the $m \times m$ matrix whose (i,j) entry is the sum*

$$\sum_{\alpha} w^{(i,j)}(\alpha)\rho(\alpha)\prod_{k=1}^{\infty} t_k^{\mathrm{bad}_k(\rho(\alpha))}$$

over all $\alpha \in \overrightarrow{P^*}$ with $E(\alpha) = (i,j)$; and let

$$\hat{L}_G(t_1, \ldots, t_k) = \sum_{\alpha \in \overrightarrow{P^*}} \hat{M}_\alpha \sum_{c \in C_\alpha} \prod_{k=1}^{\infty} t_k^{\mathrm{mk}_k(c)}.$$

Then,

$$\hat{F}_G(t_1, t_2 \ldots) = \left(I_m - \sum_{p \in P} \hat{M}_p - \hat{L}_G(t_1 - 1, t_2 - 1, \ldots)\right)^{-1}.$$

The proof is the same as that of Theorem 4, except that we apply $\hat{\lambda}$ instead of λ.

Although we will not use the weighted version of our main theorem in subsequent sections, we note that it can be used with the monoid network framework to examine time-homogeneous Markov chains, which are probabilistic analogues of finite-state automata. Specifically, let (G,P) be a monoid network with m vertices, and for every $a \in A$ and $i,j \in [m]$, let $w_a^{(i,j)} \in [0,1]$ such that $w_a^{(i.j)} = 0$ if $(a,(i,j)) \notin P$ and

$$\sum_{j=1}^{m} \sum_{a \in A} w_a^{(i,j)} = 1$$

for each fixed $1 \leq i \leq m$. With a choice of initial vertex and terminal vertex, we can think of this monoid network as a random word model, where a word is given by traversing a random walk in G from the initial vertex to the terminal vertex with $w_a^{(i,j)}$ being the probability that at vertex i, the next letter in the word will be a and the next arc (i, j). Using Theorem 5, we can then compute probabilities associated with this random process, such as the probability that a length n word obtained from traversing a walk between two specified vertices avoids a specified set of forbidden subwords.

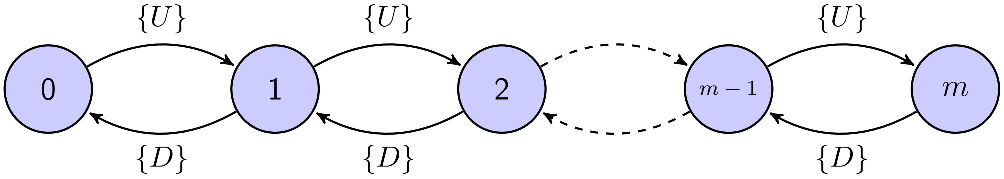
4. An application to lattice path enumeration

4.1. Representing lattice paths using monoid networks

A *path* on $\mathbb{Z}^k$ with *steps* in $S \subseteq \mathbb{Z}^k$ is an ordered tuple $(a_0, a_1, a_2, \ldots, a_n)$ of values in $\mathbb{Z}^k$ such that $a_{i+1} - a_i \in S$ for every $0 \leq i < n$. Equivalently, it is an ordered tuple $(s_1, s_2, \ldots, s_n)$ of values in S. Each step $s \in S$ is assigned a length in $\mathbb{Z}$—which we take to be 1 unless otherwise noted—and the *length* of a path is the sum of the lengths of all of its steps s_i.

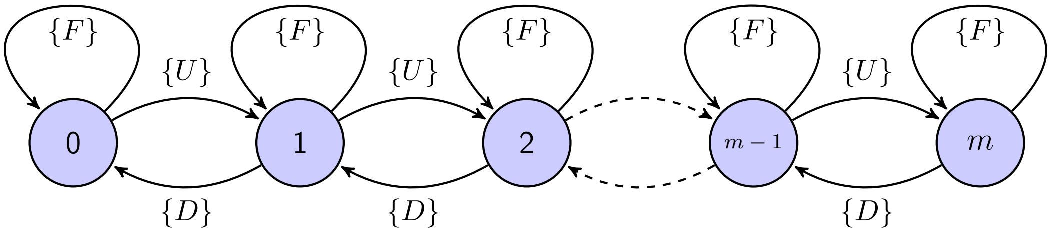
These paths are collectively known as lattice paths. In particular, lattice paths on $\mathbb{Z}$ have been widely studied in the literature, usually with the conditions $a_0 = a_k = 0$ and $a_i \geq 0$ for every i. Examples of these paths include *Dyck paths*, which have steps in $\{-1, 1\}$; *Motzkin paths*, which have steps in $\{-1, 0, 1\}$; and *Schröder paths*, which are Motzkin paths but with '0' steps having length 2 instead of 1. These paths are often illustrated as paths in the plane starting at the origin, ending on the x-axis, and never going below the x-axis, with up steps $(1, 1)$ corresponding to 1, down steps $(1, -1)$ corresponding to -1, and in the case of Motzkin or Schröder paths, flat steps $(1, 0)$ or $(2, 0)$, respectively, corresponding to 0.

We say that a lattice path on $\mathbb{Z}$ has *height bounded by* m if we add the condition that $a_i \leq m$ for every i. Lattice paths with bounded heights correspond to walks in certain monoid networks. For example, a Dyck path with height bounded by m is a walk from vertex 0 to itself in the following monoid network:



Here the alphabet is $\{U, D\}$, with U corresponding to an up step and D corresponding to a down step. The vertices represent the possible heights at each step of the path; indeed, a Dyck path with height bounded by m must begin and end at height 0, and its height must stay between 0 and m.

We can also add a letter F for flat steps, and so we can represent Motzkin paths and Schröder paths using the following, which we call the *Motzkin monoid network of order* m:



Using monoid networks, we can model a wide variety of bounded lattice paths with different types of steps and various restrictions, so we may use the tools that we have for monoid networks to obtain generating functions for counting lattice paths of bounded height. Taking the formal power series limit as $m \to \infty$ yields analogous results for lattice paths of unbounded height.

The idea of representing lattice paths as walks in digraphs and the transfer-matrix method are standard techniques in lattice path enumeration; see [11] for a recent survey of the literature. Such an approach has not yet been combined with the Goulden–Jackson cluster method to count lattice paths by occurrences of subwords, which we shall do here.

However, the original version of the cluster method was applied by Wang [17] to count Dyck paths by occurrences of various subwords. His approach is fundamentally different in that it relies on recursive decompositions of paths and does not use the correspondence to walks in digraphs, whereas our method reduces almost all of the computations to matrix algebra. Because Wang conducted his investigation on Dyck paths, we shall instead focus on Motzkin paths in this paper.

4.2. A note on continued fractions

A finite continued fraction is an expression of the form

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{\ddots + \cfrac{b_m}{a_m}}},$$

which we will write as

$$a_0 + \frac{b_1}{a_1+}\,\frac{b_2}{a_2+}\cdots\frac{b_m}{a_m}$$

for compactness. We say that a finite continued fraction has *depth* m if it is written with m fraction bars when completely written out in this notation, so the continued fraction above has depth m. We write an infinite continued fraction

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \ddots}}$$

as

$$a_0 + \frac{b_1}{a_1+}\,\frac{b_2}{a_2+}\cdots$$

and say that it has *infinite depth*.

Continued fractions arise naturally in combinatorics and especially in lattice path enumeration; e.g., see Flajolet's landmark paper [5]. Many of our results in this section are continued fraction formulae.

4.3. Counting Motzkin paths by ascents

Let $\mathcal{M}_n^m$ be the set of Motzkin paths of length n with height bounded by m and $\mathcal{M}_n$ the set of all Motzkin paths of length n. An *ascent* of a Motzkin path μ is a maximal consecutive sequence of up steps in μ, and let us define $\operatorname{asc}\mu$ to be the number of ascents in μ. We also define

$$F_m^{\mathrm{asc}}(x,t) = \sum_{n=0}^{\infty} \sum_{\mu \in \mathcal{M}_n^m} t^{\operatorname{asc}\mu} x^n \qquad \text{and} \qquad F^{\mathrm{asc}}(x,t) = \sum_{n=0}^{\infty} \sum_{\mu \in \mathcal{M}_n} t^{\operatorname{asc}\mu} x^n$$

to be bivariate generating functions for Motzkin paths with height bounded by m and regular Motzkin paths, respectively, weighted by length and number of ascents. Our main result here is the following:

Theorem 6. *Let $\{P_m^{\mathrm{asc}}(x,t)\}_{m\geq 0}$ be a sequence of polynomials recursively defined by*

$$P_m^{\mathrm{asc}}(x,t) = (1-x-x^2(t-1))P_{m-1}^{\mathrm{asc}}(x,t) - (x^2+x^3(t-1))P_{m-2}^{\mathrm{asc}}(x,t)$$

for $m \geq 2$ and $P_0^{\mathrm{asc}}(x,t) = 1$ and $P_1^{\mathrm{asc}}(x,t) = 1-x$. Then

$$\begin{aligned} F_m^{\mathrm{asc}}(x,t) &= \frac{P_m^{\mathrm{asc}}(x,t)}{P_{m+1}^{\mathrm{asc}}(x,t)} \\ &= \underbrace{\frac{1}{1-x-x^2(t-1)-}\,\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\cdots\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\,\frac{x^2+x^3(t-1)}{1-x}}_{\text{depth } m+1} \end{aligned}$$

for $m \geq 1$ and

$$\begin{aligned} F^{\mathrm{asc}}(x,t) &= \frac{1}{1-x-x^2(t-1)-}\,\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\,\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\cdots \\ &= \frac{1-x-x^2(t-1)-\sqrt{1-2x-x^2(2t+1)-2x^3(t-1)+x^4(t-1)^2}}{2(x^2+x^3(t-1))}. \end{aligned}$$

Proof. We apply the cluster method to the Motzkin monoid network of order m with $B = \{UD, UF\}$, since the number of occurrences of the subwords UD and UF in a Motzkin path is equal to its number of ascents. We weight both UD and UF by t. The only clusters formed by UD and UF are themselves, and so we have the $(m+1)\times(m+1)$ cluster matrix

$$\overrightarrow{L}_G(t) = \begin{bmatrix} UDt & UFt & & & & \\ & UDt & UFt & & & \\ & & UDt & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & UDt & UFt \\ & & & & & 0 \end{bmatrix}.$$

Then, by Theorem 4, $\overrightarrow{F}_G(t)$ is the inverse matrix of $A_m - \overrightarrow{L}_G(t-1)$, where A_m is the $(m+1)\times(m+1)$ matrix given by

$$A_m = \begin{bmatrix} 1-F & -U & & & & \\ -D & 1-F & -U & & & \\ & -D & 1-F & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1-F & -U \\ & & & & -D & 1-F \end{bmatrix}.$$

Thus, $F_m^{\text{asc}}(x,t)$ is the $(1,1)$ entry of M_m^{-1} where M_m is the $(m+1)\times(m+1)$ matrix

$$\begin{bmatrix} 1-x-x^2(t-1) & -x-x^2(t-1) & & & & \\ -x & 1-x-x^2(t-1) & -x-x^2(t-1) & & & \\ & -x & 1-x-x^2(t-1) & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1-x-x^2(t-1) & -x-x^2(t-1) \\ & & & & -x & 1-x \end{bmatrix}$$

obtained by applying the homomorphism sending $U, D, F \mapsto x$ to $A_m - \overrightarrow{L}_G(t-1)$. By Cramer's rule, we can compute this generating function as the quotient of two determinants

$$F_m^{\text{asc}}(x,t) = \frac{\det M_{m-1}}{\det M_m}.$$

Using column-addition matrix operations, which preserve the determinant, we can then transform M_m into an upper-triangular matrix with diagonal entries

$$u_{i,i} = \begin{cases} 1-x-x^2(t-1) - \dfrac{x^2+x^3(t-1)}{u_{i+1,i+1}}, & \text{if } 1 \le i \le m \\ 1-x, & \text{if } i = m+1. \end{cases}$$

From here we deduce the recursive expression

$$\begin{aligned} \det M_m &= \prod_{i=1}^{m+1} u_{i,i} \\ &= \left(1-x-x^2(t-1) - \frac{x^2+x^3(t-1)}{\left(\frac{\det M_{m-1}}{\det M_{m-2}}\right)}\right)\det M_{m-1} \\ &= (1-x-x^2(t-1))\det M_{m-1} - (x^2+x^3(t-1))\det M_{m-2} \end{aligned}$$

with initial conditions $\det M_{-1} = 1$ and $\det M_0 = 1-x$. Hence, these determinants are polynomials, and we write $P_m^{\text{asc}} = \det M_{m-1}$. Moreover,

$$\frac{\det M_m}{\det M_{m-1}} = \underbrace{1-x-x^2(t-1) - \frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\cdots\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\,\frac{x^2+x^3(t-1)}{1-x}}_{\text{depth } m},$$

so

$$F_m^{\mathrm{asc}}(x,t) = \underbrace{\frac{1}{1-x-x^2(t-1)-}\;\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\cdots\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\;\frac{x^2+x^3(t-1)}{1-x}}_{\text{depth } m+1}. \tag{6}$$

We now proceed to Motzkin paths unbounded by height. By taking the limit of (6) as $m \to \infty$, this sequence of formal power series converges to the infinite continued fraction

$$F^{\mathrm{asc}}(x,t) = \frac{1}{1-x-x^2(t-1)-}\;\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\;\frac{x^2+x^3(t-1)}{1-x-x^2(t-1)-}\cdots. \tag{7}$$

Equation (7) gives the recursive expression

$$F^{\mathrm{asc}}(x,t) = \frac{1}{1-x-x^2(t-1)-(x^2+x^3(t-1))F^{\mathrm{asc}}(x,t)}$$

or

$$(x^2+x^3(t-1))F^{\mathrm{asc}}(x,t)^2 - (1-x-x^2(t-1))F^{\mathrm{asc}}(x,t) + 1 = 0,$$

and solving this functional equation gives

$$F^{\mathrm{asc}}(x,t) = \frac{1-x-x^2(t-1) \pm \sqrt{1-2x-x^2(2t+1)-2x^3(t-1)+x^4(t-1)^2}}{2(x^2+x^3(t-1))}$$

but one can easily check that the subtraction solution is the correct one. □

The first several terms of $F^{\mathrm{asc}}(x,t)$ are in the following table:

n	$[x^n]\,F^{\mathrm{asc}}(x,t)$	n	$[x^n]\,F^{\mathrm{asc}}(x,t)$
0	1	5	$1+14t+6t^2$
1	1	6	$1+26t+23t^2+t^3$
2	$1+t$	7	$1+46t+70t^2+10t^3$
3	$1+3t$	8	$1+79t+186t^2+56t^3+t^4$
4	$1+7t+t^2$	9	$1+133t+451t^2+235t^3+15t^4$

These numbers are in the OEIS [15, A114580]. Notice that the constant terms of these polynomials are all 1; the only Motzkin paths with no ascents consist of all flat steps, and there is exactly one of each length. We also obtain an expression for the linear coefficients, which count Motzkin paths with exactly one ascent.

Corollary 7. *Let* $\mathrm{Fib}(i)$ *denote the ith Fibonacci number defined by* $\mathrm{Fib}(0) = 0$, $\mathrm{Fib}(1) = 1$, *and* $\mathrm{Fib}(n) = \mathrm{Fib}(n-1)+\mathrm{Fib}(n-2)$ *for* $n \geq 2$. *Then the number of Motzkin paths of length* $n \geq 1$ *with exactly one ascent is equal to* $\mathrm{Fib}(n+3) - n - 2$.

Proof. Using Maple, one may verify that

$$\left[\frac{\partial}{\partial t}F^{\mathrm{asc}}(x,t)\right]_{t=0} = \frac{x^2}{(1-x-x^2)(1-x)^2}.$$

It is known that

$$\frac{x}{(1-x-x^2)(1-x)^2}$$

is the generating function for the sequence $\mathrm{Fib}(n+4)-n-3$ (see [15, A001924]). Then,

$$\begin{aligned}[x^n t]\, F^{\mathrm{plt}}(x,t) &= [x^n]\, \left[\frac{\partial}{\partial t} F^{\mathrm{asc}}(x,t)\right]_{t=0} \\ &= [x^{n-1}]\, \frac{x}{(1-x-x^2)(1-x)^2} \\ &= \mathrm{Fib}(n+3)-n-2.\end{aligned}$$

□

The leading coefficients of the even-degree polynomials are 1; a Motzkin path of length $2n$ has at most n ascents, and only when the path is $(UD)^n$. A Motzkin path of length $2n+1$ also has at most n ascents, and we show that the leading coefficients of the odd-degree polynomials are the triangular numbers (see [15, A000217]):

Proposition 8. *The number of Motzkin paths of length $2n+1$ with n ascents is $\binom{n+2}{2}$.*

Proof. The maximum number of ascents that a Motzkin path of length $2n+1$ can have is n. Fix such a path μ, and let k be the number of subwords UD that occur at height 0 in μ.

- If $k=n$, then the remaining step (which must be a flat step) can be in $k+1$ possible positions: at the beginning, at the end, or between two consecutive occurrences of UD.
- If $k<n$, then it is easy to see that in order for μ to have n ascents, the remaining steps must form the subword $UF(UD)^{n-k-1}D$ beginning at height 0. Again, there are $k+1$ possible positions for this subword: at the beginning, at the end, or between two consecutive occurrences of UD.

Summing over all k, we conclude that the number of Motzkin paths of length $2n+1$ with n ascents is equal to

$$\sum_{k=0}^{n}(k+1) = \binom{n+2}{2}.$$

□

We can also use the generalized cluster method to count Motzkin paths with ascents ending only at specified heights. Let $\mathbb{P}$ be the set of positive integers, $\mathbb{N}$ the set of non-negative integers, $\mathbb{E}$ the set of positive even integers, $\mathbb{O}$ the set of positive odd integers, and $\mathbb{E}_{\geq 0}$ the set of non-negative even integers.

Theorem 9. *Let $A \subseteq \mathbb{P}$ and let*

$$F^{\mathrm{asc}}(A;x) = \sum_{n=0}^{\infty} c_n x^n$$

where c_n is the number of Motzkin paths of length n with every ascent ending at a height in A. Then,

$$F^{\text{asc}}(A;x) = \frac{1}{1-x+C_1-}\,\frac{x^2-xC_1}{1-x+C_2-}\,\frac{x^2-xC_2}{1-x+C_3-}\cdots$$

where

$$C_i = \begin{cases} x^2, & \textit{if } i \notin A \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. We weight both UD and UF by t, but we only wish to consider instances of these subwords occuring at impermissible heights as we will be setting $t = 0$ afterward. The impermissible heights are $i-1$ where $i \notin A$, so that the corresponding ascents end at height i. Thus, following the proof of Theorem 6, we take the cluster matrix $\overrightarrow{L_G}(t)$ but delete all entries in rows $i-1$ with $i \in A$. We obtain the result by applying the cluster method, using matrix operations to obtain a continued fraction formula, and then taking the limit as $m \to \infty$—all in the same way as before—and finally by setting $t = 0$. □

For example, taking $A = \mathbb{E}$ and $A = \mathbb{O}$, we obtain

$$\begin{aligned} F^{\text{asc}}(\mathbb{E};x) &= \frac{1}{1-x+x^2-}\,\frac{x^2-x^3}{1-x-}\,\frac{x^2}{1-x+x^2-}\,\frac{x^2-x^3}{1-x-}\,\frac{x^2}{1-x+x^2-}\cdots \\ &= \frac{1-2x+2x^2-\sqrt{1-4x+4x^2-4x^4+4x^5}}{2(x^2-x^3+x^4)} \\ &= 1+x+x^2+x^3+2x^4+5x^5+12x^6+27x^7+60x^8+135x^9+309x^{10}+\cdots \end{aligned}$$

and

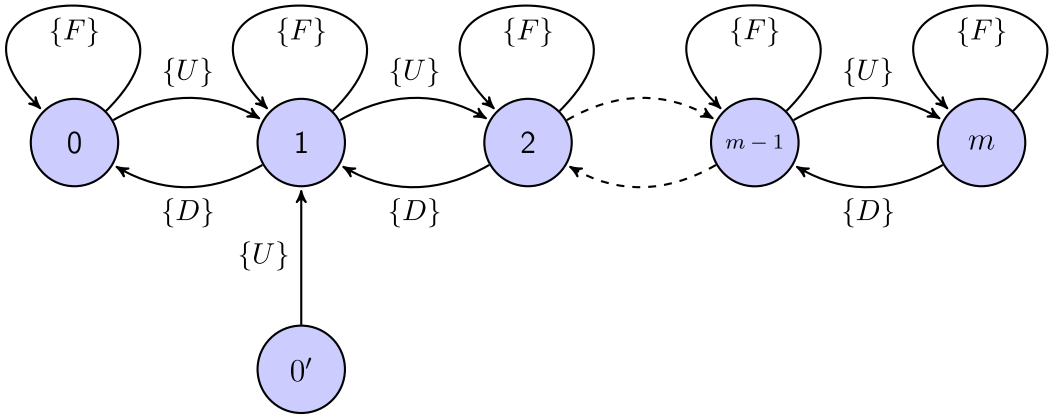
$$\begin{aligned} F^{\text{asc}}(\mathbb{O};x) &= \frac{1}{1-x-}\,\frac{x^2}{1-x+x^2-}\,\frac{x^2-x^3}{1-x-}\,\frac{x^2}{1-x+x^2-}\,\frac{x^2-x^3}{1-x-}\cdots \\ &= \frac{1-2x+2x^2-2x^3-\sqrt{1-4x+4x^2-4x^4+4x^5}}{2(x^2-2x^3+x^4)} \\ &= 1+x+2x^2+4x^3+8x^4+16x^5+33x^6+70x^7+152x^8+336x^9+754x^{10}+\cdots \end{aligned}$$

as the generating functions for Motzkin paths with all ascents ending at even heights and odd heights, respectively.[2]

One can produce a refinement of Theorem 9 that also keeps track of the number of ascents. Rather than deleting rows in the cluster matrix, assign each UD and UF in those rows a weight of u. After setting $t = 0$, the remaining indeterminates x and u would keep track of length and number of ascents, respectively.

It is also possible to count paths with restrictions on the heights at which ascents begin, but the analysis is slightly more complicated. Here we would want to set $B = \{DU, FU\}$, which suffices for Motzkin paths that do not begin with an ascent. However, Motzkin paths that begin with an ascent can be counted by considering walks in the monoid network

[2] We note that the coefficients of $F^{\text{asc}}(\mathbb{E};x)$ match OEIS sequence [15, A190171] up to x^{10} and the coefficients of $F^{\text{asc}}(\mathbb{O};x)$ match OEIS sequence [15, A110334] up to x^{12}, but begin to deviate afterward.



from vertex $0'$ to vertex 0, and we would multiply the result by t at the end to take into account the first ascent.

4.4. Counting Motzkin paths by plateaus

We now count Motzkin paths by occurrences of UF^kD, which we call a *k-plateau*.[3] For a fixed k, let $\mathrm{plt}_k(\mu)$ be the number of k-plateaus of a Motzkin path μ, and let

$$F_m^{\mathrm{plt}_k}(x,t)=\sum_{n=0}^{\infty}\sum_{\mu\in\mathcal{M}_n^m} t^{\mathrm{plt}_k(\mu)}x^n \qquad \text{and} \qquad F^{\mathrm{plt}_k}(x,t)=\sum_{n=0}^{\infty}\sum_{\mu\in\mathcal{M}_n} t^{\mathrm{plt}_k(\mu)}x^n.$$

Then we have the following formulae:

Theorem 10. *Let $\{P_m^{\mathrm{plt}_k}(x,t)\}_{m\geq 0}$ be a sequence of polynomials recursively defined by*

$$P_m^{\mathrm{plt}_k}(x,t)=(1-x-x^{k+2}(t-1))P_{m-1}^{\mathrm{plt}_k}(x,t)-x^2P_{m-2}^{\mathrm{plt}_k}(x,t)$$

for $m\geq 2$ and $P_0^{\mathrm{plt}_k}(x,t)=1$ and $P_1^{\mathrm{plt}_k}(x,t)=1-x$. Then

$$\begin{aligned}F_m^{\mathrm{plt}_k}(x,t)&=\frac{P_m^{\mathrm{plt}_k}(x,t)}{P_{m+1}^{\mathrm{plt}_k}(x,t)}\\&=\underbrace{\frac{1}{1-x-x^{k+2}(t-1)-}\,\frac{x^2}{1-x-x^{k+2}(t-1)-}\cdots\frac{x^2}{1-x-x^{k+2}(t-1)-}\,\frac{x^2}{1-x}}_{\text{depth } m+1}\end{aligned}$$

for $m\geq 1$ and

$$\begin{aligned}F^{\mathrm{plt}_k}(x,t)&=\frac{1}{1-x-x^{k+2}(t-1)-}\,\frac{x^2}{1-x-x^{k+2}(t-1)-}\,\frac{x^2}{1-x-x^{k+2}(t-1)-}\cdots\\&=\frac{1-x-x^{k+2}(t-1)-\sqrt{(1-x-x^{k+2}(t-1))^2-4x^2}}{2x^2}.\end{aligned}$$

[3] These are sometimes also called *k-humps* in the literature.

The two formulae for $F^{\mathrm{plt}_k}(x,t)$ were found earlier by Drake and Gantner [3, Proposition 3.4 and Theorem 4.2] using a different method; here we give a proof using our generalization of the cluster method.

Proof. Set $B = \{UF^kD\}$, and once again consider the Motzkin monoid network of order m. The only cluster formed by UF^kD is itself, and so the $(m+1) \times (m+1)$ cluster matrix is

$$\overrightarrow{L}_G(t) = \begin{bmatrix} UF^kDt & & & & & \\ & UF^kDt & & & & \\ & & UF^kDt & & & \\ & & & \ddots & & \\ & & & & UF^kDt & \\ & & & & & 0 \end{bmatrix}.$$

By Theorem 4, we have $\overrightarrow{F}_G(t) = (A_m - \overrightarrow{L}_G(t))^{-1}$ (where A_m is defined in the proof of Theorem 6), and so $F_m^{\mathrm{plt}_k}(x,t)$ is the $(1,1)$ entry of M_m^{-1} where M_m is the matrix

$$\begin{bmatrix} 1-x-x^{k+2}(t-1) & -x & & & & \\ -x & 1-x-x^{k+2}(t-1) & -x & & & \\ & -x & 1-x-x^{k+2}(t-1) & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1-x-x^{k+2}(t-1) & -x \\ & & & & -x & 1-x \end{bmatrix}$$

obtained by applying to $A_m - \overrightarrow{L}_G(t)$ the homomorphism sending each of U, F, and D to x.

It follows that

$$F_m^{\mathrm{plt}_k}(x,t) = \frac{\det M_{m-1}}{\det M_m},$$

and the determinant of M_m is equal to that of an upper-triangular matrix with diagonal entries

$$u_{i,i} = \begin{cases} 1-x-x^{k+2}(t-1) - \dfrac{x^2}{u_{i+1,i+1}}, & \text{if } 1 \le i \le m \\ 1-x. & \text{if } i = m+1. \end{cases}$$

Thus we have the recursion

$$\begin{aligned} \det M_m &= \prod_{i=1}^{m+1} u_{i,i} \\ &= \left(1-x-x^{k+2}(t-1) - \frac{x^2}{\left(\frac{\det M_{m-1}}{\det M_{m-2}}\right)}\right) \det M_{m-1} \\ &= (1-x-x^{k+2}(t-1)) \det M_{m-1} - x^2 \det M_{m-2} \end{aligned}$$

with initial conditions $\det M_{-1} = 1$ and $\det M_0 = 1 - x$. These are polynomials, and we write $P_m^{\text{plt}_k} = \det M_{m-1}$. Moreover,

$$\frac{\det M_m}{\det M_{m-1}} = \underbrace{1 - x - x^{k+2}(t-1) - \frac{x^2}{1 - x - x^{k+2}(t-1)-} \cdots \frac{x^2}{1 - x - x^{k+2}(t-1)-} \; \frac{x^2}{1-x}}_{\text{depth } m},$$

so

$$F_m^{\text{plt}_k}(x,t) = \underbrace{\frac{1}{1 - x - x^{k+2}(t-1)-} \; \frac{x^2}{1 - x - x^{k+2}(t-1)-} \cdots \frac{x^2}{1 - x - x^{k+2}(t-1)-} \; \frac{x^2}{1-x}}_{\text{depth } m+1}.$$

Taking the limit as $m \to \infty$, we obtain

$$\begin{aligned} F^{\text{plt}_k}(x,t) &= \frac{1}{1 - x - x^{k+2}(t-1)-} \; \frac{x^2}{1 - x - x^{k+2}(t-1)-} \; \frac{x^2}{1 - x - x^{k+2}(t-1)-} \cdots \\ &= \frac{1}{1 - x - x^{k+2}(t-1) - x^2 F^{\text{plt}_k}(x,t)} \end{aligned}$$

which can be rewritten as

$$x^2 F^{\text{plt}_k}(x,t)^2 - (1 - x - x^{k+2}(t-1)) F^{\text{plt}_k}(x,t) + 1 = 0. \tag{8}$$

Solving (8) gives

$$F^{\text{plt}_k}(x,t) = \frac{1 - x - x^{k+2}(t-1) \pm \sqrt{(1 - x - x^{k+2}(t-1))^2 - 4x^2}}{2x^2},$$

but one can check that the subtraction solution is the correct one. □

By specializing to $k = 0$ and defining peak $= \text{plt}_0$, we obtain the bivariate generating function

$$F^{\text{peak}}(x,t) = \frac{1 - x - x^2(t-1) - \sqrt{(1 - x - x^2(t-1))^2 - 4x^2}}{2x^2}$$

for Motzkin paths by *peaks*, which are occurrences of UD. The first several terms of $F^{\text{peak}}(x,t)$ are in the following table:

n	$[x^n]\, F^{\text{peak}}(x,t)$	n	$[x^n]\, F^{\text{peak}}(x,t)$
0	1	5	$8 + 10t + 3t^2$
1	1	6	$17 + 24t + 9t^2 + t^3$
2	$1 + t$	7	$37 + 58t + 28t^2 + 4t^3$
3	$2 + 2t$	8	$82 + 143t + 81t^2 + 16t^3 + t^4$
4	$4 + 4 + t^2$	9	$185 + 354t + 231t^2 + 60t^3 + 5t^4$

See [15, A097860] for its OEIS entry. Also see [15, A004148] for the constant coefficients of these polynomials, which count Motzkin paths with no peaks. The generating function for the linear coefficients of these polynomials can be verified to be

$$\left[\frac{\partial}{\partial t} F^{\text{peak}}(x,t)\right]_{t=0} = \frac{1 - x + x^2 - \sqrt{1 - 2x - x^2 - 2x^3 + x^4}}{2\sqrt{1 - 2x - x^2 - 2x^3 + x^4}},$$

and interestingly enough, dividing this generating function by x (i.e., shifting the indices of the underlying sequence) yields the generating function for the number of flat steps in all peakless Motzkin paths of length n (see [15, A110236]). These numbers are given by a binomial coefficient sum, which in turn gives us the following:

Corollary 11. *The number of Motzkin paths of length $n \geq 2$ with exactly one peak is equal to $\sum_{k=0}^{n-2} \binom{k+1}{n-k+1}\binom{k}{n-k}$.*

Now let us consider 1-plateaus, or occurrences of UFD. The bivariate generating function

$$F^{\mathrm{plt}_1}(x,t) = \frac{1 - x - x^3(t-1) - \sqrt{(1 - x - x^3(t-1))^2 - 4x^2}}{2x^2}$$

counts Motzkin paths by 1-plateaus, and its first several terms are:

n	$[x^n]\,F^{\mathrm{plt}_1}(x,t)$	n	$[x^n]\,F^{\mathrm{plt}_1}(x,t)$
0	1	5	$15+6t$
1	1	6	$36+14t+t^2$
2	2	7	$85+39t+3t^2$
3	$3+t$	8	$209+102t+12t^2$
4	$7+2t$	9	$517+280t+37t^2+t^3$

These are also in the OEIS [15, A114583], along with the constant coefficients [15, A114584], which count Motzkin paths with no occurrences of UFD.

We can also count Motzkin paths by all plateaus, without a fixed k. Let $\mathrm{plt}(\mu)$ be the number of plateaus in a Motzkin path μ, that is, the number of occurrences of subwords in $B = \{UD, UFD, UFFD, \ldots\}$. We define the bivariate generating functions $F_m^{\mathrm{plt}}(x,t)$ and $F^{\mathrm{plt}}(x,t)$ in the same way as before, and to determine these generating functions, we would change each nonzero entry in the cluster matrix from UF^kDt (for a fixed k) to

$$\sum_{k=0}^{\infty} UF^kDt = U(1-F)^{-1}Dt.$$

Then the computation would follow in the same way, yielding the following result:

Theorem 12. *Let $\{R_m^{\mathrm{plt}}(x,t)\}_{m\geq 0}$ be a sequence of rational functions recursively defined by*

$$R_m^{\mathrm{plt}}(x,t) = \left(1 - x - \frac{x^2}{1-x}(t-1)\right) R_{m-1}^{\mathrm{plt}}(x,t) - x^2 R_{m-2}^{\mathrm{plt}}(x,t)$$

for $m \geq 2$ and $R_0^{\mathrm{plt}}(x,t) = 1$ and $R_1^{\mathrm{plt}}(x,t) = 1 - x$. Then

$$\begin{aligned} F_m^{\mathrm{plt}}(x,t) &= \frac{R_m^{\mathrm{plt}}(x,t)}{R_{m+1}^{\mathrm{plt}}(x,t)} \\ &= \underbrace{\frac{1}{1 - x - \frac{x^2}{1-x}(t-1) -}\;\frac{x^2}{1 - x - \frac{x^2}{1-x}(t-1) -}\cdots\frac{x^2}{1 - x - \frac{x^2}{1-x}(t-1) -}\;\frac{x^2}{1-x}}_{\text{depth } m+1} \end{aligned}$$

for $m \geq 1$ *and*

$$\begin{aligned} F^{\mathrm{plt}}(x,t) &= \frac{1}{1-x-\frac{x^2}{1-x}(t-1)-}\,\frac{x^2}{1-x-\frac{x^2}{1-x}(t-1)-}\,\frac{x^2}{1-x-\frac{x^2}{1-x}(t-1)-}\cdots \\ &= \frac{1-2x-x^2(t-2)-\sqrt{1-4x-2x^2(t-2)+4x^3t+x^4t(t-4)}}{2(x^2-x^3)}. \end{aligned}$$

The first several terms of $F^{\mathrm{plt}}(x,t)$ are below, which can also be found on the OEIS [15, A097229]:

n	$[x^n]\,F^{\mathrm{plt}}(x,t)$	n	$[x^n]\,F^{\mathrm{plt}}(x,t)$
0	1	5	$1+15t+5t^2$
1	1	6	$1+31t+18t^2+t^3$
2	$1+t$	7	$1+63t+56t^2+7t^3$
3	$1+3t$	8	$1+127t+160t^2+34t^3+t^4$
4	$1+7t+t^2$	9	$1+255t+432t^2+138t^3+9t^4$

We now give expressions for the linear and quadratic coefficients of these polynomials.

Corollary 13. *The number of Motzkin paths of length* $n \geq 1$ *with exactly one plateau is equal to* $2^{n-1}-1$.

Proof. Using Maple, one may verify that

$$\left[\frac{\partial}{\partial t}F^{\mathrm{plt}}(x,t)\right]_{t=0} = \frac{x^2}{(1-2x)(1-x)}.$$

Then,

$$\begin{aligned} [x^nt]\,F^{\mathrm{plt}}(x,t) &= [x^n]\,\left[\frac{\partial}{\partial t}F^{\mathrm{plt}}(x,t)\right]_{t=0} \\ &= [x^{n-2}]\,\frac{1}{(1-2x)(1-x)} \\ &= [x^{n-2}]\,\left(\frac{2}{1-2x}-\frac{1}{1-x}\right) \\ &= [x^{n-2}]\Big(\sum_{n=0}^{\infty}(2^{n+1}-1)x^n\Big) \\ &= 2^{n-1}-1. \end{aligned}$$

□

Corollary 14. *The number of Motzkin paths of length* $n \geq 3$ *with exactly two plateaus is equal to* $(n-3)n2^{n-6}$.

Proof. Using Maple, one may verify that

$$\left[\frac{\partial^2}{\partial t^2}F^{\mathrm{plt}}(x,t)\right]_{t=0} = \frac{2(1-x)x^4}{(1-2x)^3}$$

and

$$\frac{(1-x)x}{(1-2x)^3}$$

is known to be the generating function for the sequence $(n(n+3)2^{n-3})_{n\geq 1}$ (see [15, A001793]). Then,

$$\begin{aligned}[x^nt^2]\,F^{\mathrm{plt}}(x,t) &= [x^n]\,\frac{1}{2}\left[\frac{\partial^2}{\partial t^2}F^{\mathrm{plt}}(x,t)\right]_{t=0}\\ &= [x^{n-3}]\,\frac{(1-x)x}{(1-2x)^3}\\ &= (n-3)n2^{n-6}.\end{aligned}$$

□

Hence, Motzkin paths with exactly 1 plateau and those with exactly 2 plateaus are equinumerous with many other combinatorial objects (see [15, A000225 and A001793]).

Drake and Gantner [3, Section 4] showed how one can find continued fraction formulae for variations of these results, including bivariate generating functions for counting Motzkin paths by plateaus occurring only at certain heights, and with restrictions on the lengths of plateaus. Their approach involved inserting appropriate "correction terms" at each level of the continued fraction formulae that encode the types of plateaus that they wish to count.

All of these variations can also be computed using our method. To disregard plateaus occurring at certain heights, we would delete the corresponding rows from the cluster matrix, which is completely analogous to Theorem 9 for ascents. To place restrictions on the lengths of plateaus, we would alter the "forbidden set" B appropriately and set the appropriate indeterminates to 0. We leave the details to the reader.

Our method also allows for an interpretation of Drake and Gantner's correction terms in terms of clusters. Their correction terms are of the form $x^k(t-1)$ for various k and are then multiplied by x^2, and these precisely correspond to the terms contributed by the cluster matrix in our computations. This is a relatively simple case because the only clusters formed by the words in $B = \{UD, UFD, UFFD, \dots\}$ are the words in B themselves. Counting paths by subwords having additional clusters would require more complicated correction terms when working through the lens of Drake and Gantner.

4.5. Counting Motzkin paths by peaks and valleys

Peaks, or occurrences of UD, were introduced in the previous subsection. Similarly, we define a *valley* to be an occurrence of DU, and $\operatorname{val}\mu$ the number of valleys of a Motzkin path μ. Here find the joint distribution of peaks and valleys in Motzkin paths. Let

$$F_m^{\mathrm{p,v}}(x,t_1,t_2) = \sum_{n=0}^{\infty}\sum_{\mu\in\mathcal{M}_n^m} t_1^{\operatorname{peak}\mu}t_2^{\operatorname{val}\mu}x^n \qquad \text{and} \qquad F^{\mathrm{p,v}}(x,t_1,t_2) = \sum_{n=0}^{\infty}\sum_{\mu\in\mathcal{M}_n} t_1^{\operatorname{peak}\mu}t_2^{\operatorname{val}\mu}x^n.$$

Then we have the following:

Theorem 15. *Let $\{R_m^{\mathrm{p,v}}(x,t_1,t_2)\}_{m\geq 0}$ be a sequence of rational functions recursively defined by*

$$R_m^{\mathrm{p,v}}(x,t_1,t_2) = (1-x-C_1-C_2)R_{m-1}^{\mathrm{p,v}}(x,t_1,t_2) - (x+C_3)^2 R_{m-2}^{\mathrm{p,v}}(x,t_1,t_2)$$

for $m \geq 2$ and $R_0^{\mathrm{p,v}}(x,t_1,t_2) = 1$ and $R_1^{\mathrm{p,v}}(x,t_1,t_2) = 1-x-C_2$, where

$$C_1 = \frac{x^2(t_1-1)}{1-x^2(t_1-1)(t_2-1)},\ C_2 = \frac{x^2(t_2-1)}{1-x^2(t_1-1)(t_2-1)},\ C_3 = \frac{x^3(t_1-1)(t_2-1)}{1-x^2(t_1-1)(t_2-1)}.$$

Then

$$\begin{aligned} F_m^{\mathrm{p,v}}(x,t_1,t_2) &= \frac{R_m^{\mathrm{p,v}}(x,t_1,t_2)}{(1-x-C_1)R_m^{\mathrm{p,v}}(x,t_1,t_2) - (x+C_3)^2 R_{m-1}^{\mathrm{p,v}}(x,t_1,t_2)} \\ &= \underbrace{\frac{1}{1-x-C_1-}\,\frac{(x+C_3)^2}{1-x-C_1-C_2-}\cdots\frac{(x+C_3)^2}{1-x-C_1-C_2-}\,\frac{(x+C_3)^2}{1-x-C_2}}_{\text{depth } m+1} \end{aligned}$$

for $m \geq 1$ and

$$\begin{aligned} F^{\mathrm{p,v}}(x,t_1,t_2) &= \frac{1}{1-x-C_1-}\,\frac{(x+C_3)^2}{1-x-C_1-C_2-}\,\frac{(x+C_3)^2}{1-x-C_1-C_2-}\cdots \\ &= \frac{2}{1-x-C_1+C_2+\sqrt{(1-x-C_1-C_2)^2-4(x+C_3)^2}}. \end{aligned}$$

Proof. Set $B = \{UD, DU\}$. This time, we weight occurrences of UD by t_1 and occurrences of DU by t_2. However, finding the cluster matrix is no longer a trivial task. We make the following observations:

- Clusters starting and ending at height 0 are of the form $UDUD\cdots UD$, since a path cannot go down from height 0. We can decompose these words into a sequence of UDs, where the first UD contributes a t_1 and each subsequent UD contributes a t_1 and a t_2.

- Clusters starting and ending at height m are of the form $DUDU\cdots DU$, since a path cannot go up from height m. We can decompose these words into a sequence of DUs, where the first DU contributes a t_2 and each subsequent DU contributes a t_1 and a t_2.

- Clusters starting and ending at height k with $0 < k < m$ are of the above two forms, since a path can go either up or down from height k.

- Clusters starting at height k and ending at height $k+1$ are of the form $UDUDU\cdots DU$, which can be decomposed into an initial subword UDU—contributing a t_1 and a t_2—and a sequence of DUs, each contributing a t_1 and a t_2.

- Clusters starting at height k and ending at height $k-1$ are of the form $DUDUD\cdots UD$, which can be decomposed into an initial subword DUD—contributing a t_1 and a t_2—and a sequence of UDs, each contributing a t_1 and a t_2.

Thus, the $(m+1)\times(m+1)$ cluster matrix is

$$\overrightarrow{L}_G(t_1,t_2) = \begin{bmatrix} \hat{C}_1 & \hat{C}_3 & & & & \\ \hat{C}_4 & \hat{C}_1+\hat{C}_2 & \hat{C}_3 & & & \\ & \hat{C}_4 & \hat{C}_1+\hat{C}_2 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \hat{C}_1+\hat{C}_2 & \hat{C}_3 \\ & & & & \hat{C}_4 & \hat{C}_2 \end{bmatrix}$$

where

$$\hat{C}_1 = \frac{UDt_1}{1-UDt_1t_2},\ \hat{C}_2 = \frac{DUt_2}{1-DUt_1t_2},\ \hat{C}_3 = \frac{UDUt_1t_2}{1-DUt_1t_2},\ \hat{C}_4 = \frac{DUDt_1t_2}{1-UDt_1t_2}.$$

By applying Theorem 4, we see that $F_m^{\mathrm{p,v}}(x,t_1,t_2)$ is the $(1,1)$ entry of M_m^{-1} where M_m is the $(m+1)\times(m+1)$ matrix

$$M_m = \begin{bmatrix} 1-x-C_1 & -x-C_3 & & & & \\ -x-C_3 & 1-x-C_1-C_2 & -x-C_3 & & & \\ & -x-C_3 & 1-x-C_1-C_2 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & 1-x-C_1-C_2 & -x-C_3 \\ & & & & -x-C_3 & 1-x-C_2 \end{bmatrix}$$

and C_1, C_2, C_3 defined in the statement of this theorem. Then,

$$\begin{aligned} F_m^{\mathrm{p,v}}(x,t_1,t_2) &= \frac{\det M'_m}{\det M_m} \\ &= \frac{\det M'_m}{\left(1-x-C_1-\frac{(x+C_3)^2}{\left(\frac{\det M'_m}{\det M'_{m-1}}\right)}\right)\det M'_m} \\ &= \frac{\det M'_m}{(1-x-C_1)\det M'_m-(x+C_3)^2\det M'_{m-1}} \end{aligned}$$

where M'_m is the matrix obtained from M_m by deleting the first row and the first column. The determinant of M'_m is equal to that of an upper-triangular matrix with diagonal entries

$$u_{i,i} = \begin{cases} 1-x-C_1-C_2-\dfrac{(x+C_3)^2}{u_{i+1,i+1}}, & \text{if } 1\le i\le m \\ 1-x-C_2, & \text{if } i=m+1, \end{cases}$$

so

$$\begin{aligned}\det M'_m &= \prod_{i=1}^{m+1} u_{i,i} \\ &= \left(1 - x - C_1 - C_2 - \frac{(x+C_3)^2}{\left(\frac{\det M'_{m-1}}{\det M'_{m-2}}\right)}\right) \det M'_{m-1} \\ &= (1 - x - C_1 - C_2) \det M'_{m-1} - (x+C_3)^2 \det M'_{m-2}\end{aligned}$$

with initial conditions $\det M'_0 = 1$ and $\det M'_1 = 1 - x - C_2$. These are rational functions, and we write $R^{\mathrm{p,v}}_m = \det M'_m$. Furthermore,

$$\frac{\det M_m}{\det M'_m} = \underbrace{1 - x - C_1 - \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \cdots \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \, \frac{(x+C_3)^2}{1 - x - C_2}}_{\text{depth } m},$$

so

$$F^{\mathrm{p,v}}_m(x, t_1, t_2) = \underbrace{\frac{1}{1 - x - C_1 -} \, \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \cdots \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \, \frac{(x+C_3)^2}{1 - x - C_2}}_{\text{depth } m+1}.$$

By taking the limit as $m \to \infty$, we have that

$$\begin{aligned}F^{\mathrm{p,v}}(x, t_1, t_2) &= \frac{1}{1 - x - C_1 -} \, \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \, \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \cdots \\ &= \frac{1}{1 - x - C_1 - (x+C_3)^2 G(x, t_1, t_2)}\end{aligned}$$

where

$$\begin{aligned}G(x, t_1, t_2) &= \frac{1}{1 - x - C_1 - C_2} \, \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \, \frac{(x+C_3)^2}{1 - x - C_1 - C_2 -} \cdots \\ &= \frac{1}{1 - x - C_1 - C_2 - (x+C_3)^2 G(x, t_1, t_2)}.\end{aligned}$$

Thus we have the functional equation

$$(x+C_3)^2 G(x, t_1, t_2)^2 - (1 - x - C_1 - C_2) G(x, t_1, t_2) + 1 = 0,$$

and solving it gives

$$G(x, t_1, t_2) = \frac{1 - x - C_1 - C_2 \pm \sqrt{(1 - x - C_1 - C_2)^2 - 4(x+C_3)^2}}{2(x+C_3)^2}.$$

As before, one can verify that the subtraction solution is the correct one, and we conclude that

$$\begin{aligned} F^{\mathrm{p,v}}(x,t_1,t_2) &= \frac{1}{1-x-C_1-\frac{1}{2}\Big(1-x-C_1-C_2-\sqrt{(1-x-C_1-C_2)^2-4(x+C_3)^2}\Big)} \\ &= \frac{2}{1-x-C_1+C_2+\sqrt{(1-x-C_1-C_2)^2-4(x+C_3)^2}}. \end{aligned}$$

□

The first several terms of $F^{\mathrm{p,v}}(x,t_1,t_2)$ are the following:

n	$[x^n]\, F^{\mathrm{p,v}}(x,t_1,t_2)$
0	1
1	1
2	$1+t_1$
3	$2+2t_1$
4	$4+4t_1+t_1^2t_2$
5	$8+8t_1+2t_1t_2+t_1^2+2t^2t_2$
6	$16+t_2+18t_1+6t_1t_2+3t_1^2+6t_1^2t_2+t_1^3t_2^2$
7	$33+4t_2+40t_1+18t_1t_2+9t_1^2+16t_1^2t_2+3t_1^2t_2^2+2t_1^3t_2+2t_1^3t_2^2$
8	$69+13t_2+90t_1+50t_1t_2+25t_1^2+3t_1t_2^2+47t_1^2t_2+t_1^3+9t_1^2t_2^2+6t_1^3t_2+9t_1^3t_2^2+t_1^4t_2^3$

The constant coefficients, which count Motzkin paths with no peaks and valleys, are in the OEIS [15, A004149].

Liu et al. [13] gave recursive and continued fraction formulae for counting Dyck paths with peaks avoiding a specified set of heights and valleys avoiding another specified set of heights. We can do the same thing by applying our cluster method to the monoid network for Dyck paths, but here we give the analogous results for Motzkin paths. Note that Liu et al. defined the height of a peak (respectively, valley) to be the height at which its down step (respectively, up step) occurs, but we use the convention that the height of a peak or valley is the height at which the corresponding subword (UD or DU) begins.

Theorem 16. *Let*

$$F^{\mathrm{p,v}}(P,V;x)=\sum_{n=0}^{\infty}c_nx^n$$

where c_n is the number of Motzkin paths of length n with every peak occuring at a height in $P\subseteq\mathbb{N}$ and every valley occuring at a height in $V\subseteq\mathbb{P}$. Then,

$$F^{\mathrm{p,v}}(P,V;x)=\frac{1}{1-x+C_{1,0}-}\,\frac{(x+C_{3,0})^2}{1-x+C_{1,1}+C_{2,1}-}\,\frac{(x+C_{3,1})^2}{1-x+C_{1,2}+C_{2,2}-}\cdots$$

where

$$C_{1,i}=\begin{cases}\frac{x^2}{1-x^2}, & \textit{if } i\notin P \textit{ and } i+1\notin V\\ x^2, & \textit{if } i\notin P \textit{ and } i+1\in V\\ 0, & \textit{otherwise,}\end{cases}\qquad C_{2,i}=\begin{cases}\frac{x^2}{1-x^2}, & \textit{if } i\notin V \textit{ and } i-1\notin P\\ x^2, & \textit{if } i\notin V \textit{ and } i-1\in P\\ 0, & \textit{otherwise,}\end{cases}$$

and

$$C_{3,i} = \begin{cases} \frac{x^3}{1-x^2}, & \textit{if } i \notin P \textit{ and } i+1 \notin V \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. We weight both UD and DU by t, but we only wish to consider instances of UD at heights $i \notin P$ and instances of DU at heights $i \notin V$. We claim that the cluster matrix is

$$\overrightarrow{L}_G(t) = \begin{bmatrix} \hat{C}_{1,0} & \hat{C}_{3,0} & & & & \\ \hat{C}_{4,1} & \hat{C}_{1,1}+\hat{C}_{2,1} & \hat{C}_{3,1} & & & \\ & \hat{C}_{4,2} & \hat{C}_{1,2}+\hat{C}_{2,2} & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \hat{C}_{1,m-1}+\hat{C}_{2,m-1} & \hat{C}_{3,m-1} \\ & & & & \hat{C}_{4,m} & \hat{C}_{2,m} \end{bmatrix}$$

where

$$\hat{C}_{1,i} = \begin{cases} \frac{UD(t-1)}{1-UD(t-1)^2}, & \text{if } i \notin P,\ i+1 \notin V \\ UD(t-1), & \text{if } i \notin P,\ i+1 \in V \\ 0, & \text{otherwise,} \end{cases} \qquad \hat{C}_{2,i} = \begin{cases} \frac{DU(t-1)}{1-UD(t-1)^2}, & \text{if } i \notin V,\ i-1 \notin P \\ DU(t-1), & \text{if } i \notin V,\ i-1 \in P \\ 0, & \text{otherwise,} \end{cases}$$

$$\hat{C}_{3,i} = \begin{cases} \frac{UDU(t-1)}{1-DU(t-1)^2}, & \text{if } i \notin P,\ i+1 \notin V \\ 0, & \text{otherwise,} \end{cases} \qquad \hat{C}_{4,i} = \begin{cases} \frac{DUD(t-1)}{1-UD(t-1)^2}, & \text{if } i \notin V,\ i-1 \notin P \\ 0, & \text{otherwise.} \end{cases}$$

For example, $\hat{C}_{1,i}$ gives clusters starting and ending at height i and beginning with an up step. Every such cluster begins with a peak, so if $i \in P$, then $\hat{C}_{1,i} = 0$. Otherwise, $i \notin P$, and if $i+1 \in V$, then the only possible such cluster is UD because all other possible clusters begin with UD and are followed by a valley at height $i+1$. However, if $i \notin P$ and $i+1 \notin V$, then every subword of the form $UDUD\cdots$ is a valid cluster. One can verify the formulae for $\hat{C}_{2,i}, \hat{C}_{3,i}, \hat{C}_{4,i}$ using similar reasoning, and the result follows from the same process as before. □

Below are the generating functions for Motzkin paths with parity restrictions on the heights of peaks and valleys:

$$\begin{aligned} F^{\mathrm{p,v}}(\mathbb{O}, \mathbb{E}_{\geq 0}; x) &= \frac{1}{1-x+\frac{x^2}{1-x^2}-}\,\frac{(x+\frac{x^3}{1-x^2})^2}{1-x+\frac{x^2}{1-x^2}-}\,\frac{x^2}{1-x+\frac{x^2}{1-x^2}-} \\ &\qquad\qquad\qquad \frac{(x+\frac{x^3}{1-x^2})^2}{1-x+\frac{x^2}{1-x^2}-}\,\frac{x^2}{1-x+\frac{x^2}{1-x^2}-}\cdots \\ &= \frac{1-2x+2x^2-2x^4-\sqrt{1-4x+4x^2-4x^4}}{2x^2(1-x+x^3)} \\ &= 1+x+x^2+2x^3+5x^4+12x^5+27x^6+60x^7+136x^8+\cdots \end{aligned}$$

$$F^{\mathrm{p,v}}(\mathbb{E}_{\geq 0}, \mathbb{O}; x) = \cfrac{1}{1-x-}\,\cfrac{x^2}{1-x+\frac{x^2}{1-x^2}-}\,\cfrac{(x+\frac{x^3}{1-x^2})^2}{1-x+\frac{x^2}{1-x^2}-}\,\cfrac{x^2}{1-x+\frac{x^2}{1-x^2}-}\,\cfrac{(x+\frac{x^3}{1-x^2})^2}{1-x+\frac{x^2}{1-x^2}-}\cdots$$
$$= \frac{2(1-x+x^3)}{1-2x+2x^3+\sqrt{1-4x+4x^2-4x^4}}$$
$$= 1+x+2x^2+4x^3+8x^4+17x^5+38x^6+88x^7+208x^8+\cdots$$

$$F^{\mathrm{p,v}}(\mathbb{O}, \mathbb{O}; x) = \cfrac{1}{1-x+x^2-}\,\cfrac{x^2}{1-x-}\,\cfrac{x^2}{1-x+2x^2-}\,\cfrac{x^2}{1-x-}\,\cfrac{x^2}{1-x+2x^2-}\cdots$$
$$= \frac{2(1-x)}{1-2x+x^2+\sqrt{1-4x+6x^2-8x^3+5x^4-4x^5+4x^6}}$$
$$= 1+x+x^2+2x^3+5x^4+12x^5+27x^6+60x^7+137x^8+\cdots$$

$$F^{\mathrm{p,v}}(\mathbb{E}_{\geq 0}, \mathbb{E}_{\geq 0}; x) = \cfrac{1}{1-x-}\,\cfrac{x^2}{1-x+2x^2-}\,\cfrac{x^2}{1-x-}\,\cfrac{x^2}{1-x+2x^2-}\,\cfrac{x^2}{1-x-}\cdots$$
$$= \frac{1-2x+3x^2-2x^3-\sqrt{1-4x+6x^2-8x^3+5x^4-4x^5+4x^6}}{2x^2(1-x)}$$
$$= 1+x+2x^2+4x^3+7x^4+13x^5+27x^6+59x^7+131x^8+\cdots$$

We note that the list of coefficients of $F^{\mathrm{p,v}}(\mathbb{E}_{\geq 0}, \mathbb{O}; x)$ in particular is a shifted version of the OEIS sequence [15, A025276], which can be verified by comparing generating functions.

Acknowledgements. The author thanks Ira Gessel and Jordan Tirrell for reading earlier versions of the manuscript and providing helpful suggestions, as well as Cyril Banderier for several generous discussions pertaining to this project at the 8th International Conference on Lattice Path Combinatorics & Applications.

References

[1] Frédérique Bassino, Julien Clément, and Pierre Nicodème. Counting occurrences for a finite set of words: combinatorial methods. *ACM Trans. Algorithms*, 8(3):Article 31, 28, 2012.

[2] L. Carlitz, Richard Scoville, and Theresa Vaughan. Enumeration of pairs of sequences by rises, falls and levels. *Manuscripta Math.*, 19(3):211–243, 1976.

[3] Dan Drake and Ryan Gantner. Generating functions for plateaus in Motzkin paths. *J. Chungcheong Math. Soc.*, 25(3):475–489, 2012.

[4] Anne E. Edlin and Doron Zeilberger. The Goulden-Jackson cluster method for cyclic words. *Adv. in Appl. Math.*, 25(2):228–232, 2000.

[5] P. Flajolet. Combinatorial aspects of continued fractions. *Discrete Math.*, 32(2):125–161, 1980.

[6] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[7] Ralph Fröberg. Determination of a class of Poincaré series. *Math. Scand.*, 37(1):29–39, 1975.

[8] Ira Martin Gessel. *Generating Functions and Enumeration of Sequences*. PhD thesis, Massachusetts Institute of Technology, 1977.

[9] I. P. Goulden and D. M. Jackson. An inversion theorem for cluster decompositions of sequences with distinguished subsequences. *J. London Math. Soc. (2)*, 20(3):567–576, 1979.

[10] Yong Kong. Extension of Goulden-Jackson cluster method on pattern occurrences in random sequences and comparison with Régnier-Szpankowski method. *J. Difference Equ. Appl.*, 11(15):1265–1271, 2005.

[11] Christian Krattenthaler. Lattice Path Enumeration. In Miklós Bóna, editor, *Handbook of Enumerative Combinatorics*, chapter 10, pages 589–678. CRC Press, 2015.

[12] Elizabeth J. Kupin and Debbie S. Yuster. Generalizations of the Goulden-Jackson cluster method. *J. Difference Equ. Appl.*, 16(12):1463–1480, 2010.

[13] Shu-Chung Liu, Jun Ma, and Yeong-Nan Yeh. Dyck paths and peak- and valley-avoiding sets. *Stud. Appl. Math.*, 121(3):263–289, 2008.

[14] John Noonan and Doron Zeilberger. The Goulden-Jackson cluster method: extensions, applications and implementations. *J. Differ. Equations Appl.*, 5(4-5):355–377, 1999.

[15] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. Published electronically at `http://oeis.org`, 2015.

[16] Richard P. Stanley. *Enumerative Combinatorics, Vol. 1*. Cambridge University Press, 2nd edition, 2011.

[17] Chao-Jen Wang. *Applications of the Goulden-Jackson cluster method to counting Dyck paths by occurrences of subwords*. PhD thesis, Brandeis University, 2011.

[18] Xiangdong Wen. The symbolic Goulden-Jackson cluster method. *J. Difference Equ. Appl.*, 11(2):173–179, 2005.

[19] Doron Zeilberger. The umbral transfer-matrix method. V. The Goulden-Jackson cluster method for infinitely many mistakes. *Integers*, 2:Paper A5, 12, 2002.

[20] Yan Zhuang. Counting permutations by runs. Preprint, 2015.