Enumeration of Hamiltonian Circuits in Rectangular Grids

Robert Stoyan*
Volker Strehl†
Computer Science Department (Informatik)
University of Erlangen-Nürnberg
D-91058 Erlangen, Germany

[Note: this article has been accepted for publication by the *Journal of Combinatorial Mathematics and Combinatorial Computing*]

Abstract

We describe an algorithm for computing the number $h_{m,n}$ of Hamiltonian circuits in a rectangular grid graph consisting of $m \times n$ squares. For any fixed m, the set of Hamiltonian circuits on such graphs (for varying n) can be identified via an appropriate coding with the words of a certain regular language $L_m \subset (\{0,1\}^m)^*$. We show how to systematically construct a finite automaton $\mathcal{A}_m$ recognizing L_m. This allows, in principle, the computation of the (rational) generating function $h_m(z)$ of L_m. We exhibit a bijection between the states of $\mathcal{A}_m$ and the words of length $m+1$ of the familiar Motzkin language. This yields an upper bound on the degree of the denominator polynomial of $h_m(z)$, hence of the order of the linear recurrence satisfied by the coefficients $h_{m,n}$ for fixed m.

The method described here has been implemented. Numerical data resulting from this implementation are presented at the end of this article.

*rtstoyan@cip.informatik.uni-erlangen.de
†strehl@immd1.informatik.uni-erlangen.de

1 Introduction

We consider the problem of enumerating Hamiltonian circuits on rectangular grid graphs.

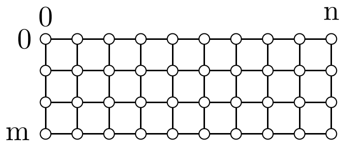


Figure 1. The grid graph $G^*_{3,9}$

Let $h_{m,n}$ denote the number of Hamiltonian circuits on a grid graph with $(m+1) \times (n+1)$ vertices as in Fig. 1, and let

$$h_m(z) = \sum_{n \geq 1} h_{m,n} z^n$$

denote the associated generating function for fixed m. The main goal of this article is to outline an algorithm which allows to systematically compute — in principle — $h_m(z)$ for any m. It turns out that $h_m(z)$ is always a rational function, a fact that has been observed by authors who studied this enumeration problem for small values of m by *ad hoc* methods ([3],[4],[5]). This result is an immediate consequence of the transfer-matrix method, which we employ here for the general approach. See Sec. 4.7 in [7] for a presentation of this method. Indeed, we show how Hamiltonian circuits on grid graphs can be encoded by the words of a suitable language that is recognized by a finite automaton. Note that Hamiltonicity of a graph has both a local (every vertex is visited exactly once) and a global (the subgraph is connected) aspect. It is quite obvious how to code the local aspect in a way that can be checked by a finite automaton. It is less obvious how the same can be done for the global aspect in a systematical way. This is the main contribution of the present article.

Once a coding in terms of a regular language has been given and a recognizing finite automaton has been constructed, the computation of the generating function $h_m(z)$ is a routine matter — in principle! In practice there are severe limits due to the exponentially increasing number of states (as a function of m). Indeed, we can give precise information about the number of states of our automata (prior to minimization) and thus an upper bound for the degree of the denominator polynomial of $h_m(z)$. Interestingly, the states can be put into bijection with the words of the familiar Motzkin language. Even though minimization may cut down the number of states considerably (for m odd about half of the original states turn out to be non-reachable), we conjecture that the growth of the degrees of the denominator polynomials of $h_m(z)$ is of the same order as that of the Motzkin numbers.

We refer the reader to [2] for the basic notions of automata theory needed here, and [1] for the relation between regular languages and rational generating functions.

The algorithm outlined here has been implemented by the first author ([8]). For efficiency reasons, this implementation uses a slightly different way of representing the automaton in question, which we will not discuss here. The program, written in the C++ language, and a complete description of its functionality are available on request from the first author. At the end of this article we present some results obtained by this program.

2 Representation and characterization

We begin by introducing some notation:

For positive integers m, n the grid graph $G_{m,n}$ is given by its vertex set $\{\,(x,y)\,;\,1 \le x \le n, 1 \le y \le m\,\}$ and the usual nearest-neighbour-edges of a rectangular grid.

It is convenient for our purposes to introduce also the extended grid graph $G^*_{m,n}$ with vertex set $\{\,(x,y)\,;\,0 \le x \le n, 0 \le y \le m\,\}$. See Fig. 1 for an example.

A "cell" of $G^*_{m,n}$ is a quadrangle of points

$$[x,y] = \{(x,y),(x-1,y),(x,y-1),(x-1,y-1)\}$$

where $1 \le x \le n, 1 \le y \le m$. Think of $G_{m,n}$ as a graph whose vertices are the cells of $G^*_{m,n}$, and edges in $G_{m,n}$ joining neighbouring cells in $G^*_{m,n}$ (i.e. cells which have an edge of $G^*_{m,n}$ in common).

A *Hamiltonian circuit* of $G^*_{m,n}$ is a subgraph H^* of $G^*_{m,n}$ with the following properties:

- every vertex (x,y) has degree 2 w.r.t. H^*
- H^* is connected

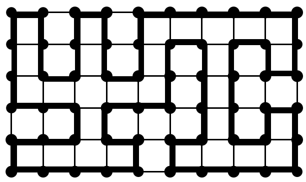
By a discrete version of the Jordan curve theorem it is clear that each cell $[x,y]$ of $G^*_{m,n}$ lies either "inside" or "outside" such a Hamiltonian circuit

H^*. This gives rise to a mapping

$$H : G_{m,n} \to \{0,1\} : (x,y) \mapsto \begin{cases} 1 & \text{if } [x,y] \text{ is an "inside" cell w.r.t. } H^* \\ 0 & \text{if } [x,y] \text{ is an "outside" cell w.r.t. } H^* \end{cases}$$

Let now $F : G_{m,n} \to \{0,1\}$ be any mapping. We will use the same notation F for different presentations of the same object:

- a mapping from (the vertex set of) $G_{m,n}$ to $\{0,1\}$, as indicated;
- the corresponding $(m \times n)$-matrix which has entry $F(x,y)$ in position (x,y);
- the *induced* subgraph of $G_{m,n}$ with vertex set $F^{-1}(1)$;
- the word $F^{(1)}F^{(2)} \dots F^{(n)}$ over the alphabet $\Sigma_m := \{0,1\}^m$ where $F^{(k)}$ denotes the k-th column of the matrix F, written as a word over Σ_m, for convenience.



1	0	1	0	1	1	1	1	1
1	0	1	0	1	0	1	0	1
1	1	1	1	1	0	1	0	0
0	0	1	0	0	0	1	0	1
1	1	1	1	0	1	1	1	1

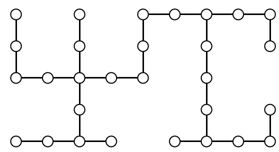


Figure 2. Three representations : conventional, matrix, induced subgraph.

Our first goal is to give a handy characterization of those mappings F that correspond to the Hamiltonian circuits of $G^*_{m,n}$. For this purpose we need a concept which allows us to represent the degree constraint of circuits.

Two vectors (or words) $u = u_1 \dots u_p$, $v = v_1 \dots v_p \in \{0,1\}^p$ (for some p) are *compatible*, if for all k $(1 \le k < p)$

$$\begin{pmatrix} u_k & v_k \\ u_{k+1} & v_{k+1} \end{pmatrix} \notin \left\{ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\}$$

Think of the matrix on the left as representing the "inside"-"outside" situation of four neighbouring cells of $G^*_{m,n}$ with respect to a Hamiltonian circuit H^*. It is clear that neither of the matrices on the r.h.s is possible because they represent the situation at vertices of degree 0 or 4 w.r.t. the subgraph H^*. Note that all the other 12 (2×2)-matrices over $\{0,1\}$ may occur because they represent the situation occuring at vertices of degree 2.

For any vector (word) $w \in \{0,1\}^m$ let $\overline{w}$ denote the *augmented* vector $0 \cdot w \cdot 0 \in \{0,1\}^{m+2}$, constructed by prepending and appending a 0 to w. Furthermore, let $\mathbf{0} = 0^m \in \{0,1\}^m$ denote the all-zero vector and $\overline{\mathbf{0}} = 0^{m+2}$ the corresponding augmented vector.

It is easy to see, that the following holds:

- if H^* is a Hamiltonian circuit of $G^*_{m,n}$, then
  - the sequence of vectors
    $$\overline{\mathbf{0}}\ ,\ \overline{H}^{(1)}\ ,\ \overline{H}^{(2)}\ ,\ \ldots\ ,\ \overline{H}^{(n)}\ ,\ \overline{\mathbf{0}}$$
    is a compatible sequence, i.e. $\overline{H}^{(i)}$ is compatible with $\overline{H}^{(i+1)}$ for $0 \leq i \leq n$, where we put $\overline{H}^{(0)} = \overline{\mathbf{0}} = \overline{H}^{(n+1)}$;
  - the induced subgraph H of $G_{m,n}$ is a tree.

It is a little less obvious that the converse also holds: any mapping $F : G_{m,n} \to \{0,1\}$ such that the sequence $\overline{\mathbf{0}}, \overline{F}^{(a)}, \ldots, \overline{F}^{(n)}, \overline{\mathbf{0}}$ is compatible and such that F, viewed as an induced subgraph of $G_{m,n}$, is a tree, actually gives rise to a Hamiltonian circuit of $G^*_{m,n}$. We give a short outline of *proof*:

F will be used to define a subgraph F^* of $G^*_{m,n}$ in the obvious way:

- an edge $(x-1,y) \longleftrightarrow (x,y)$, where $1 \leq x \leq n$, $0 \leq y \leq m$ belongs to F^* (together with its endpoints) iff $\overline{F}(x,y) \neq \overline{F}(x,y+1)$
- an edge $(x,y-1) \longleftrightarrow (x,y)$, where $0 \leq x \leq n$, $1 \leq y \leq m$ belongs to F^* iff $\overline{F}(x,y) \neq \overline{F}(x+1,y)$

We have used here the augmented $(m+2) \times (n+2)$-matrix $\overline{F}$:

$$\overline{F}(x,y) = \begin{cases} F(x,y) & \text{if } 1 \leq x \leq n, 1 \leq y \leq m \\ 0 & \text{if } x \in \{0, n+1\} \text{ or } y \in \{0, m+1\} \end{cases}$$

is obtained by bordering F with zeros.

The compatibility condition guarantees that F^* has degree 2 at each vertex. The tree condition guarantees that F^* is connected.

3 Constructing Hamiltonian circuits

Let us now consider the problem of systematically *constructing* Hamiltonian circuits $H^* \subseteq G^*_{m,n}$ for fixed m and arbitrary $n \in \mathbf{N}$. We have seen that these circuits correspond to mappings $H : G_{m,n} \to \{0,1\}$ such that the sequence $\overline{\mathbf{0}}, \overline{H}^{(1)}, \overline{H}^{(2)}, \ldots, \overline{H}^{(n)}, \overline{\mathbf{0}}$ of augmented column vectors is a compatible one, and that the induced subgraph $H \subseteq G_{m,n}$ is a tree. This implies that each initial seqment $\overline{\mathbf{0}}, \overline{H}^{(1)}, \ldots, \overline{H}^{(k)}$ is a compatible sequence and that the corresponding subgraph $H^{(1\ldots k)} \subseteq G_{m,k}$ is a forest. This forest is "special" in that all its trees have at least one vertex belonging to the last, the k-th column. We will now consider two aspects: *continuation* and *termination*.

- Continuation: if we want to extend such an initial segment in a way that may ultimately lead to a Hamiltonian circuit $H^* \subseteq G^*_{m,n}$ for some $n > k$, we have to examine as candidates for the $(k+1)$-st column $H^{(k+1)}$ all nonzero vectors $K \in \{0,1\}^m$ such that the following holds:

  - $\overline{H}^{(k)}$ and $\overline{K}$ are compatible;
  - the subgraph induced by $H^{(1)}, \ldots, H^{(k)}, K$ in $G_{m,k+1}$ is again a special forest.

  The first of these conditions can be easily checked by a finite automaton. What is less obvious is that for checking the second condition we do not need to know the whole "history" $H^{(1)}, \ldots, H^{(k)}$, but only a limited (i.e. bounded for m fixed) amount of information in addition to knowing $H^{(k)}$:

  - we need to know which of the 1-cells of $H^{(k)}$ (i.e. the y such that $H^{(k)}_y = H(k,y) = 1$) belong to the same tree in the special forest induced by $H^{(1)}, \ldots, H^{(k)}$.

  We can then check whether the addition of K as $(k+1)$-st column leads to a cycle or not in the new induced subgraph, and if not, whether all the previously existing trees are now merged into trees which all have at least one vertex belonging to in the $(k+1)$-st column. Note that the addition of K may also create new trees consisting of just a single vertex or a string of vertices in the $(k+1)$-st column. (In the example given in Fig. 4, this happens in the third column.)

- Termination: If all the vertices of $H^{(k)}$ belong to the same tree, then by maintaining the property of being a "special" forest by continuation, it is clear that the forest induced by $H^{(1)}, \ldots, H^{(k)}$ is in fact a single tree. If $\overline{H}^{(k)}$ turns out to be compatible with $\overline{\mathbf{0}}$, we may terminate the construction and a Hamiltonian cycle of $G^*_{m,k}$ is constructed.

4 Some technical details

It should be evident from the previous section that for each fixed $m > 0$ a finite automaton can be constructed which works over the alphabet $\Sigma_m = \{0,1\}^m$ and which accepts a word $H = H^{(1)}H^{(2)}\dots H^{(n)} \in \Sigma_m^*$ if and only if the corresponding matrix H represents a Hamiltonian circuit of $G^*_{m,n}$. In this section we look a little closer on the problem how to construct such an automaton. The main point is, of course, how to integrate the knowledge about the forest induced by initial segments $H^{(1)}H^{(2)}\dots H^{(k)}$ into the states of such an automaton.

Let us begin with recalling some familiar concepts from combinatorics. A *partition* of the set $[1..n] := \{1,2,\dots,n\}$ can be specified by a function $\pi : [1..n] \to [1..n]$ such that

$$\pi(1) = 1\ ,\ 1 \leq \pi(j) \leq \max\{\pi(i); i < j\} + 1 \text{ for } 2 \leq j \leq n$$

The idea of this coding is that element j belongs to block number k if $\pi(j) = k$, and the sequence of smallest elements in blocks numbered $1,2,3,\dots$ is increasing.

A partition π of $[1..n]$ is *non-crossing* (*ncp*) if for each quadruple $1 \leq i < j < k < l \leq n$

$$\pi(i) = \pi(k) \wedge \pi(j) = \pi(l) \text{ implies } \pi(i) = \pi(j)(= \pi(k) = \pi(l))$$

For later use we introduce the notation $\mathcal{NCP}_n$ to denote the set of non-crossing partitions of $[1..n]$.

If we look at the situation discussed above, namely that of a sequence $H^{(1)}, H^{(2)}, \dots, H^{(k)}$ in Σ_m inducing a “special” forest in $G_{m,k}$, we notice that the partition π of the vertices belonging to $H^{(k)}$ according to the membership in the trees of the forest is necessarily an *ncp*. More precisely, let

$$\overline{H}^{(k)} = 0^{j_0}1^{i_1}0^{j_1}1^{i_2}0^{j_2}\dots 1^{i_r}0^{j_r} \text{ with } i_1,\dots,i_r,j_0,j_1,\dots,j_r > 0$$

be the unique factorization of $\overline{H}^{(k)}$ into maximal 0-blocks and maximal 1-blocks. Vertices belonging to the same 1-block obviously belong to the same tree induced by $H^{(1)}, \ldots, H^{(k)}$. Thus a partition π of $[1..r]$ indicates to which tree the vertices of each 1-block belong. An example is given in Fig. 4, where a Hamiltonian circuit on $G^*_{9,5}$, written in matrix form, is given, together with the *ncp*s associated to the five column vectors and coding the backward tree structure. These partitions π must belong to $\mathcal{NCP}$ for obvious topological reasons. Otherwise we would have a contradictory situation as indicated in Fig. 3. These objects, pairs (u, π) consisting of a vector $u \in \{0,1\}^m$ together with an *ncp* π on its set of maximal 1-blocks, are actually the *states* of the automaton we are going to construct.

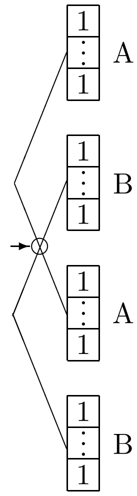


Figure 3.

$$\begin{array}{|c|c|c|c|c|}\hline 1&1&1&0&1\\\hline 1&0&0&0&1\\\hline 1&0&1&1&1\\\hline 1&0&0&0&1\\\hline 1&1&1&0&1\\\hline 0&0&1&0&1\\\hline 1&1&1&1&1\\\hline 0&0&1&0&0\\\hline 1&1&1&1&1\\\hline\end{array} \longrightarrow \begin{pmatrix}1\\2\\3\end{pmatrix}, \begin{pmatrix}1\\1\\2\\3\end{pmatrix}, \begin{pmatrix}1\\2\\1\end{pmatrix}, \begin{pmatrix}1\\2\\2\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}$$

Figure 4. Example : The partitions π corresponding to the columns.

We now define the state set:

$$Q_m = \{\, q = (u, \pi)\,;\, u \in \{0,1\}^m, \pi \ ncp \text{ on maximal 1-blocks of } u \,\}$$

5 Construction of the automaton $\mathcal{A}_m$

Let $L_m \subset \Sigma_m^*$ denote the set of matrices corresponding to Hamiltonian circuits on $G^*_{m,n}$ for some $n \geq 1$. L_m will be considered as a language over the alphabet $\Sigma_m = \{0,1\}^m$, written as column vectors, and with horizontal concatenation of columns as operation. In this section we will construct a finite automaton $\mathcal{A}_m$ recognizing this language L_m.

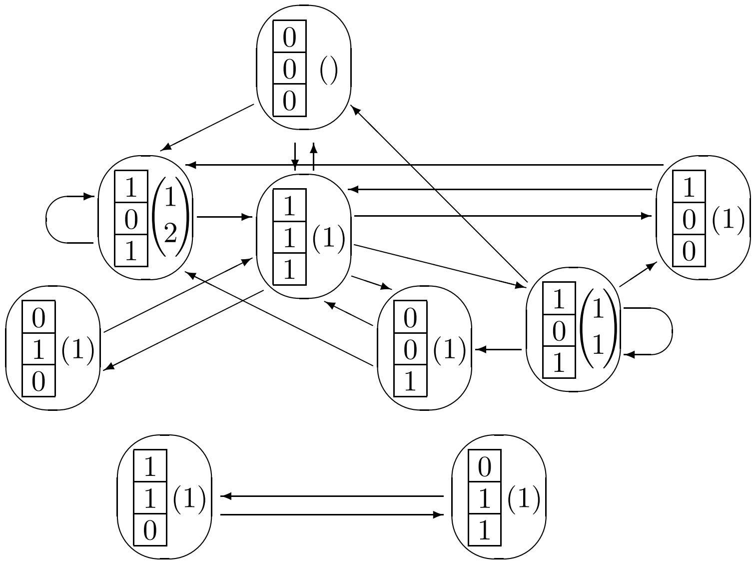


Figure 5. The transition system $\mathcal{T}_3$

We first construct a transition system $\mathcal{T}_m = (Q_m, \to)$ as follows:

> for $q, q' \in Q_m$ with $q = (u, \pi)$, $q' = (v, \sigma)$ we put $q \to q'$ if and only if the following holds
>
> – $\overline{u}$ and $\overline{v} \in \{0,1\}^{m+2}$ are compatible vectors
>
> – extending the (type of) special forest structure encoded in $q = (u, \pi)$ via the *ncp* π by column v again leads to a (type of) special forest structure, which is encoded by (v, σ).

It must be admitted that checking the second item is rather intricate to implement. No further details are given here, see [8]. We point out, however, that given a state $q = (u, \pi)$ and $v \in \Sigma_m$ such that $\overline{u}$ and $\overline{v}$ are compatible, there is *at most one* σ such that $q' = (v, \sigma) \in Q_m$ and $q \to q'$ holds. The paths of length $n+1$ in $\mathcal{T}_m$, starting at $q_0 := (\mathbf{0}, \emptyset)$ and ending in q_0, without returning to q_0 in between, precisely correspond to the Hamiltonian circuits of $G^*_{m,n}$.

This leads to the construction of the automaton $\mathcal{A}_m := \langle Q'_m, \Sigma_m, \delta, \alpha, \Omega \rangle$. Again, we take $\Sigma_m = \{0,1\}^m$ as alphabet and define a *complete deterministic* automaton over the state set

$$Q'_m = \{Q_m \setminus q_0\} \cup \{\alpha, \rho\}$$

where the new states α (ρ resp.) serve as *initial* (*dead* resp.) states. The set Ω of terminal states is given by

$$\Omega = \{ (u, \pi_\omega) \,;\, (u, \pi_\omega) \to q_0 \text{ in } \mathcal{T}_m \}$$

Here $\pi_\omega = (1)$ denotes the *ncp* where all 1-blocks belong to the same class.

The transition function $\delta : Q'_m \times \Sigma_m \to Q'_m$ is defined as follows:

– for $q = (u, \pi)$, $q' = (v, \sigma) \in Q_m \setminus \{q_0\}$:

$$\delta(q, v) := \begin{cases} q' & \text{iff } q \to q' \text{ in } \mathcal{T}_m \\ \rho & \text{iff } q \not\to q' \text{ in } \mathcal{T}_m \end{cases}$$

– for each $u \in \Sigma_m$ such that $q_0 \to (u, \pi_\alpha)$ in $\mathcal{T}_m$:

$$\delta(\alpha, u) = (u, \pi_\alpha)$$

Here π_α is the *ncp* where each 1-block of u is in a class by itself.

The language L_m of Hamiltonian circuits on $G^*_{m,n}$ (for some $n \geq 1$) is the language accepted by $\mathcal{A}_m$. In particular, we have thus shown that the language L_m is a regular (rational) language for any m, and that its generating function is rational.

6 A review of the Motzkin language

In the next section we will present another way of writing the states of our transition systems $\mathcal{T}_m$, hence of the automata $\mathcal{A}_m$. This has the double advantage of being closer to the actual implementation of the automata, and of giving precise information about the size (number of states) of $\mathcal{T}_m$ and $\mathcal{A}_m$. Quite surprisingly, it turns out that the states of $\mathcal{T}_m$ can be put into bijection with the words of length $m+1$ of the familiar Motzkin language.

In the present section we recall some (familiar) facts about the Motzkin language $\mathcal{M}$ over the ternary alphabet $\{x, \overline{x}, y\}$. This language can be defined as the unique solution in $\{x, \overline{x}, y\}^*$ of the fixed point equation

$$Z = y^* \cdot (\lambda + x \cdot Z \cdot \overline{x} \cdot Z) \quad ,$$

where λ denotes the empty word. This equation reflects the fact that $\mathcal{M}$ is a context-free language, generated by the (unambiguous) grammar

$$Z \to Y + Y \cdot x \cdot Z \cdot \overline{x} \cdot Z \quad , \quad Y \to \lambda + y \cdot Y$$

Hence, a word $w \in \{x, \overline{x}, y\}^*$ belongs to $\mathcal{M}$ if and only if one of the two cases holds:

– $w = y^k$ for some $k \geq 0$

– w has a factorization $w = y^k \cdot x \cdot u \cdot \overline{x} \cdot v$ with $u, v \in \mathcal{M}$ and $k \geq 0$ (recursion!)

Another way of looking at the Motzkin language is by taking the Dyck language over the alphabet $\{x, \overline{x}\}$ and "shuffling" it with the set of words $y^* = \{y^k\,;\, k \geq 0\}$.

We note in passing that the set $\mathcal{NCP}_n$ of non-crossing partitions of $[1..n]$ is in bijection with the words of length $2n$ of the Dyck-language over $\{x, \overline{x}\}$, hence the cardinality $\sharp\mathcal{NCP}_n$ is the Catalan number $cat_n = \frac{1}{n+1}\binom{2n}{n}$.

We now list some known facts about $mot_n := \sharp\mathcal{M}_n$, the number of Motzkin words of length n:

– The sum representation

$$mot_n = \sum_{k=0}^{\lfloor n/2 \rfloor} cat_k \binom{n}{2k} = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{k+1} \binom{2k}{k} \binom{n}{2k}$$

– The generating function

$$\begin{aligned} \sum_n mot_n\, z^n &= \frac{1 - z - \sqrt{1 - 2z - 3z^2}}{2z^2} \\ &= 1 + z + 2z^2 + 4z^3 + 9z^4 + 21z^5 + 51z^6 + \ldots \end{aligned}$$

– The asymptotic behaviour

$$mot_n = \sqrt{\frac{3}{4\,\pi\, n^3}}\, 3^n - \frac{45}{32} \frac{1}{\sqrt{3\,\pi\, n^5}}\, 3^n + O\left(\frac{3^n}{n^{7/2}}\right)$$

7 A bijection

We will now define a mapping Ψ which maps the state set Q_m of the transition system $\mathcal{T}_m$ bijectively onto the set $\mathcal{M}_{m+1}$ of words of length $m+1$ of the Motzkin language $\mathcal{M}$ over the alphabet $\{x, \overline{x}, y\}$.

The definition is by induction on m.

– For $m = 0$, we formally introduce $Q_0 = \{(\lambda, \emptyset)\}$, where λ is the empty word and $\emptyset$ denotes the empty partition, the unique element of $\mathcal{NCP}_0$. This "state" is mapped by Ψ onto the word $y \in \mathcal{M}_1$.

– More generally: for any $m \geq 0$, the state set Q_m contains the element $(0^m, \emptyset)$, and this particular state will be mapped by Ψ onto the word $y^{m+1} \in \mathcal{M}_{m+1}$.

– Let $q = (u, \pi) \in Q_m, q \neq (0^m, \emptyset)$. Then $u \in \{0,1\}^m$ has r maximal 1-blocks for some r with $1 \leq r \leq \lfloor m/2 \rfloor$, and π is an element of $\mathcal{NCP}_r$. As in Section 4, write

$$\overline{u} = 0 \cdot u \cdot 0 = 0^{j_0} 1^{i_1} 0^{j_1} 1^{i_2} 0^{j_2} \ldots 1^{i_r} 0^{j_r}$$

with $i_1, \ldots, i_r, j_0, \ldots, j_r > 0$ and

$$i_1 + \ldots + i_r + j_0 + \ldots + j_r = m + 2$$

Now factorize π according to the last position in $\pi(1)\pi(2)\ldots\pi(r)$ where "1" appears. Let

$$\begin{aligned} k &= \max\{\, \nu \,;\, \pi(\nu) = 1 \,\} \\ h &= \max\{\, \pi(\nu) \,;\, 1 \leq \nu \leq k \,\} \end{aligned}$$

Thus k is this last position and h is the maximum block number that appears up to this position. By the properties of non-crossing partitions it is clear that after position k only block numbers bigger than h appear in π. More precisely:

$$\begin{aligned} \pi' &:= \pi(1)\pi(2)\ldots\pi(k-1) \in \mathcal{NCP}_{k-1} \\ \pi'' &:= (\pi(k+1) - h)(\pi(k+2) - h)\ldots(\pi(r) - h) \in \mathcal{NCP}_{r-k} \end{aligned}$$

(This decomposition $\pi \mapsto (\pi', \pi'')$ can actually be used to produce a bijection between $\mathcal{NCP}_r$ and the set of Dyck words of length $2r$.)

Now let

$$\begin{aligned} q' &:= \Big(0^{j_0} 1^{i_1} 0^{j_1} \ldots 0^{j_{k-2}} 1^{i_{k-1}} 0^{j_{k-1}} \,,\, \pi'\Big) \in Q_{m'} \\ q'' &:= \Big(0^{j_k} 1^{i_{k+1}} 0^{j_{k+1}} \ldots 0^{j_{r-1}} 1^{i_r} 0^{j_r} \,,\, \pi''\Big) \in Q_{m''} \end{aligned}$$

where

$$\begin{aligned} i_1 + \cdots + i_{k-1} + j_0 + \cdots + j_{k-1} &= m' + 2 \\ i_{k+1} + \cdots + i_r + j_k + \cdots + j_r &= m'' + 2 \end{aligned}$$

and define

$$\Psi(q) := y^{i_k - 1} \cdot x \cdot \Psi(q') \cdot \overline{x} \cdot \Psi(q'')$$

If we assume (by induction) that

$$\Psi(q') \in \mathcal{M}_{m'+1} \quad \text{and} \quad \Psi(q'') \in \mathcal{M}_{m''+1}$$

then — in view of the characterization on $\mathcal{M}$ given in the previous section — we have

$$\Psi(q) \in \mathcal{M}_t \quad \text{where} \quad t = i_k + 1 + m' + 1 + m'' + 1 = m + 1 \quad ,$$

as desired. It is a routine task to check that this decomposition and the mapping Ψ can be perfectly reversed, i.e. Ψ maps Q_m bijectively onto $\mathcal{M}_{m+1}$ for each $m \geq 0$.

Let us illustrate this construction of Ψ by an example. For this purpose we take states $q = (u, \pi) \in Q_m$ with u as above and write it as

$$q \equiv 0^{j_0} \pi(1)^{i_1} 0^{j_1} \pi(2)^{i_2} 0^{j_2} \ldots \pi(r)^{i_r} 0^{j_r}$$

i.e. we replace the ones in the 1-blocks of $\overline{u} = 0 \cdot u \cdot 0$ by the corresponding π-values of the blocks. We let Ψ operate on words of this type. (Note that in this coding we have $\Psi(0^n) = y^{n-1}$ for $n > 0$.)

Let $m = 30$ and $q = (u, \pi)$ with

$$u = 0^1 1^2 0^3 1^1 0^2 1^3 0^1 1^4 0^2 1^1 0^1 1^2 0^2 1^1 0^3 1^1 \in \{0,1\}^{30}$$

and

$$\pi = 1\,2\,1\,1\,3\,4\,3\,5 \in \mathcal{NCP}_8$$

We have

$$k = 4 \ , \ h = 2 \ , \ i_k = 4 \ , \ \pi' = 1\,2\,1 \ , \ \pi'' = 1\,2\,1\,3$$

so that

$$q \equiv 0^2 1^2 0^3 2^1 0^2 1^3 0^1 1^4 0^2 3^1 0^1 4^2 0^2 3^1 0^3 5^1 0^1$$

will be mapped onto

$$\Psi(q) = y^3 \cdot x \cdot \Psi(0^2 1^2 0^3 2^1 0^2 1^3 0^1) \cdot \overline{x} \cdot \Psi(0^2 1^1 0^1 2^2 0^2 1^1 0^3 3^1 0^1)$$

Proceeding inductively we arrive at

$$\begin{aligned}
\Psi(0^2 1^2 0^3 2^1 0^2 1^3 0^1) &= y^2 \cdot x \cdot \Psi(0^2 1^2 0^3 2^1 0^2) \cdot \overline{x} \cdot \Psi(0) \\
&= y^2 \cdot x \cdot y \cdot x \cdot \Psi(0^2) \cdot \overline{x} \cdot \Psi(0^3 1^1 0^2) \cdot \overline{x} \\
&= y^2 \cdot x \cdot y \cdot x \cdot y \cdot \overline{x} \cdot x \cdot \Psi(0^3) \cdot \overline{x} \cdot \Psi(0^2) \cdot \overline{x} \\
&= y^2 \cdot x \cdot y \cdot x \cdot y \cdot \overline{x} \cdot x \cdot y^2 \cdot \overline{x} \cdot y \cdot \overline{x}
\end{aligned}$$

and

$$\begin{aligned}
\Psi(0^2 1^1 0^1 2^2 0^2 1^1 0^3 3^1 0^1) &= x \cdot \Psi(0^2 1^1 0^1 2^2 0^2) \cdot \overline{x} \cdot \Psi(0^3 1^1 0^1) \\
&= x \cdot x \cdot \Psi(0^2) \cdot \overline{x} \cdot \Psi(0^1 1^2 0^2) \cdot \overline{x} \cdot x \cdot \Psi(0^3) \cdot \overline{x} \cdot \Psi(0) \\
&= x^2 \cdot y \cdot \overline{x} \cdot y \cdot x \cdot \Psi(0) \cdot \overline{x} \cdot \Psi(0^2) \cdot \overline{x} \cdot x \cdot y^2 \cdot \overline{x} \\
&= x^2 \cdot y \cdot \overline{x} \cdot y \cdot x \cdot \overline{x} \cdot y \cdot \overline{x} \cdot x \cdot y^2 \cdot \overline{x}
\end{aligned}$$

To conclude this section, we take the example of a Hamiltonian circuit (in matrix form) given at the end of Section 4 and show how it translates into a sequence of Motzkin words (as columns).

$$\begin{array}{ccccc}
1 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
 & & & &
\end{array}
\simeq
\begin{array}{ccccc}
y & x & y & x & x \\
y & x & y & y & y \\
y & \bar{x} & y & y & y \\
y & y & y & \bar{x} & y \\
x & y & x & x & y \\
\bar{x} & \bar{x} & x & x & y \\
x & x & \bar{x} & y & y \\
\bar{x} & \bar{x} & x & y & x \\
x & x & \bar{x} & \bar{x} & \bar{x} \\
\bar{x} & \bar{x} & \bar{x} & \bar{x} & \bar{x}
\end{array}$$

Figure 6. A (matrix) Hamiltonian circuit and its Motzkin translation

8 Compatible vectors again

In this section we point out a simplification that is possible in both constructing the automata $\mathcal{A}_m$ and computing with them. This is due to a simple parity argument that is inherent in the concept of compatible vectors (or words), as introduced in Section 2.

Consider the following simple transition system on four states, named 00, 01, 10, and 11.

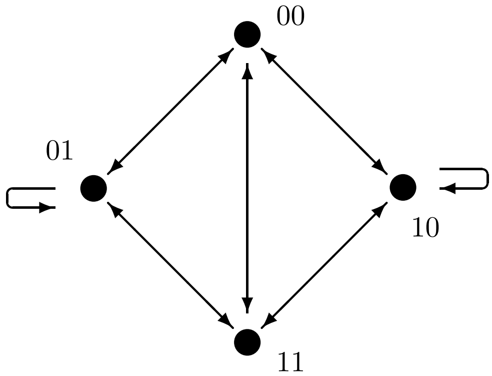


Figure 7. The transition system for compatible vectors

To each walk w of length m in this transition graph, given by the sequence $w = w_0, w_1, \ldots, w_n$ of states it visits, we associate two words of length $m+1$ over $\{0,1\}$:

$$\begin{array}{lcl} u_w & := & \text{the concatenation of the first components of } w_0, w_1, \ldots, w_m \\ v_w & := & \text{the concatenation of the second components of } w_0, w_1, \ldots, w_m \end{array}$$

It is easy to see that the pairs (u_w, v_w) of words of the same length thus generated are precisely the pairs of compatible vectors as defined in Section 2.

For $\mathcal{T}_m$ and $\mathcal{A}_m$ we need to consider (and to produce) all compatible pairs $(\overline{u}, \overline{v}) = (0 \cdot u \cdot 0, 0 \cdot v \cdot 0)$ with $u, v \in \{0,1\}^m$ — i.e. we have to consider walks of length $m+1$ starting and ending in state 00. Note that each walk of that kind has an *even* number of transitions to or from state 11. Legal transitions between the other three states, however, have the property that the generate a factor 00 either in u_w or in v_w, but not in both words.

This last observation has the following consequence: call $u \in \{0,1\}^m$ an *even* vector if the word $\overline{u} = 0 \cdot u \cdot 0$ contains an even number of factors 00, otherwise u is *odd*. Now, by the previous remark:

> let $u, v \in \{0,1\}^m$ be vectors such that $\overline{u}$ and $\overline{v}$ are compatible, then
>
> – if m is odd, then either both of u, v are even or both are odd
>
> – if m is even, then one of u, v is even and the other one is odd

We draw the following consequences:

- if m is odd, then 0^m is even and only states $q = (u, \pi) \in Q_m$ with even u are reachable from 0^m in $\mathcal{T}_m$. Hence all the states $q = (u, \pi)$ with u odd can be eliminated from the construction.

- if m is even, then 0^m is odd and we need an even number of transitions for a walk that leads from 0^m back to 0^m, since successive states must alternate in "parity".

The last remark reflects the well known fact that $G^*_{m,n}$ has a Hamiltonian circuit if and only if not both m and n are even.

The next to last remark is illustrated by our example for $m = 3$: there are $mot_4 = 9$ states in $\mathcal{T}_3$, namely

$$(000, \emptyset)\ ,\ (001, 1)\ ,\ (010, 1)\ ,\ (100, 1)\ ,\ (101, 11)\ ,\ (101, 12)\ ,\ (111, 1)$$
$$(011, 1)\ ,\ (110, 1)$$

of which the last two are "odd", hence unreachable from any of the seven "even" states, see Fig. 5.

The precise evaluation of the size of the set of unreachable states in the case of odd m will be given elsewhere.

We conclude this section with a *conjecture* that has been verified by our computations up to $m = 12$, but for which we are unable to give a proof in general:

- in the automaton $\mathcal{A}_m$, all accessible states are also coaccessible, i.e. from every state, which is accessible from the initial state α, there is at least one path that leads to the final state ω.

If this conjecture were true in general, the minimization procedure à la Nerode for the automaton $\mathcal{A}_m$ that we employ prior to computing the generating function (see the end of the following section) could be simplified considerably by just identifying states that have the same set of successor states.

9 Computing the generating function $h_m(z)$

From Section 5 we know that the generating function $h_m(z) = \sum_{n\geq 1} h_{m,n}\, z^n$ is rational. In this section we informally discuss a way of actually computing $h_m(z)$, at least for small values of m. As mentioned in the introduction, a program has been written which carries out these computations for arbitrary m — in principle. We have explicitly computed the automata $\mathcal{A}_m$, recognizing the Hamiltonian language L_m, for $m \leq 12$. Some of our results are presented in the concluding section. Needless to say that the exponential growth of $\mathcal{A}_m$ (in terms of the number of states, as made precise by the Motzkin-coding in Section 7) puts severe bounds on practical computations.

A standard way of computing $h_m(z)$ proceeds as follows: one takes the automaton $\mathcal{A}_m$ and represents it by the transition matrix

$$\mathbf{A}_m = \left(a^{(m)}_{p,q}\right)_{p,q \in Q''} \quad \text{where} \quad a^{(m)}_{p,q} = \begin{cases} 1 & \text{if } \delta(p,v) = q \text{ for some } v \in \Sigma_m \\ 0 & \text{else} \end{cases}$$

with $Q''_m = Q'_m \setminus \{\alpha, \rho\}$. Note that, by the definition of $\mathcal{A}_m$, there is at most one $v \in \Sigma_m$ such that $\delta(p,v) = q$, for any $p, q, \in Q''_m$.

Let $\mathbf{s}$ denote the vector of states immediately acessible from the initial state α of $\mathcal{A}_m$:

$$\mathbf{s} = (s_p)_{p \in Q''_m} \quad \text{where} \quad s_p = \begin{cases} 1 & \text{if } \delta(\alpha, u) = p = (u, \pi_\alpha) \text{ for some } u \in \Sigma^* \\ 0 & \text{else} \end{cases}$$

and, correspondingly, let $\mathbf{t}$ denote the vector of terminal states

$$\mathbf{t} = (t_q)_{q \in Q''_m} \quad \text{where} \quad t_q = \begin{cases} 1 & \text{if } q \in \Omega \\ 0 & \text{else} \end{cases}$$

Then, for any $n \geq 0$,

$$h_{m,n+1} = \mathbf{s} \cdot \mathbf{A}_m^n \cdot \mathbf{t}^{\mathrm{T}}$$

and thus

$$h_m(z) = \sum_{n \geq 1} h_{m,n}\, z^n = z \cdot \mathbf{s} \cdot (\mathbf{I} - z\, \mathbf{A_m})^{-1} \cdot \mathbf{t}^{\mathrm{T}}$$

where $\mathbf{I}$ denotes an identity matrix of the same format as $\mathbf{A}_m$.

Inverting the matrix $\mathbf{I} - z\,\mathbf{A}_m$ is feasible only for very moderate values of m. One way to compute $h_m(z)$ for larger values of m is by producing sufficiently many initial coefficients $h_{m,1}$, $h_{m,2}$, $h_{m,3}$, ... using the matrix $\mathbf{A}_m$ and its powers, and then employing the techniques of Padé approximation. In particular, the size of the matrix $\mathbf{A}_m$ gives us an upper bound on the degree of the denominator polynomial of $h_m(z)$, so that we know in advance when to stop the approximation procedure.

A slightly more sophisticated approach that we have experimented successfully with for the purpose of computing $h_m(z)$ is a combination of modular arithmetic and the Berlekamp-Massey algorithm (see e.g. Section 8.6 in

[6]): we compute the residues of the sequence $h_{m,1}$, $h_{m,2}$, $h_{m,3}$, ... modulo various (not too big) primes p and apply the Berlekamp-Massey algorithm over the finite fields $GF(p)$ in parallel. This gives a very fast way of determining the true degree of the denominator polynomial of $h_m(z)$. The numerator and denominator polynomial themselves can then be obtained by Chinese remaindering techniques. A more detailed description of this approach will be given elsewhere.

One more aspect must be mentioned: the techniques just sketched will not be applied to the automaton $\mathcal{A}_m$ and its transition matrix $\mathbf{A}_m$ themselves. We will, of course, first apply standard minimization techniques from automata theory in order to cut down the size of the matrices to be handled, by eliminating inaccessible states (remember Section 8) and identifying equivalent states. The minimal automaton $\tilde{\mathcal{A}}_m$ obtained from $\mathcal{A}_m$ gives us a transition matrix $\tilde{\mathbf{A}}_m$ considerably smaller in size, thus leading to a much better *à priori* bound on the degree of the numerator polynomial of the $h_m(z)$. Nonetheless, our numerical results seem to indicate that the growth rates of the number of states after minimization is of the same order as the one for the automata $\mathcal{A}_m$ themselves.

10 Numerical results

10.1 The generating functions $h_m(z)/z$ up to $m = 8$

We give the rational generating functions $h_m(z)/z$ for $1 \leq m \leq 6$ and the degrees of the numerator and denominator polynomials for $m = 7, 8$. For these two latter cases the polynomials have been computed explicitly and are available on request, as are (very large, with n in the thousands) initial segments of the coefficient sequences $h_{m,n}$ $(n \geq 1)$ up to $n = 12$.

Note that for m even the generating function $h_m(z)/z$ is actually "even" in the sense of being a function of z^2. This follows from the parity argument given in Section 8.

$m = 1$:

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

$m = 2$:

$$\frac{1}{-2z^2+1} = 1 + 2z^2 + 4z^4 + 8z^6 + 16z^8 + \ldots$$

$m = 3$:

$$\frac{1}{-z^4+2z^3-2z^2-2z+1} = 1 + 2z + 6z^2 + 14z^3 + 37z^4 + 92z^5 + 236z^6 + \ldots$$

$m = 4:$

$$\frac{3z^2+1}{-2z^6-11z^2+1} = 1 + 14z^2 + 154z^4 + 1696z^6 + 18684z^8 + \dots$$

$m = 5:$

$$\frac{(z-1)\,(z^{11} - z^{10} + 3\,z^9 + 12\,z^8 - 3\,z^7}{-2z^{14} + 4z^{13} - 28z^{12} - 42z^{11} + 82z^{10} + 8z^9 - 118z^8 + 66z^7}$$

$$\dots \frac{-\ 3\,z^4 + 21\,z^3 - 3\,z^2 - 1)}{+\ 35z^6 - 90z5 - 12z^4 + 63z^3 - 14z^2 - 5z + 1}$$

$$= 1+4z+37z^2+154z^3+1072z^4+5320z^5+32675z^6+175294z^7+1024028z^8+\dots$$

$m = 6$:

$$\frac{96z^{32} - 48z^{30} - 4592z^{28} - 9162z^{26} + 64012z^{24} - 252197z^{22}}{-144z^{36}-1728z^{34}+5972z^{32}-17732z^{30}-92790z^{28}+178842z^{26}+1036420z^{24}}$$

$$\dots \frac{+\ 643288z^{20} - 797154z^{18} + 453054z^{16} - 40229z^{14}}{-\ 3390877z^{22} + 4008954z^{20} - 2681994z^{18} + 1690670z^{16} - 1251439z^{14}}$$

$$\dots \frac{-\ 111603z^{12} + 87046z^{10} - 33250z^8 + 6525z^6 - 568z^4 + 7z^2 + 1}{+\ 815141z^{12} - 386724z^{10} + 116734z^8 - 20403z^6 + 1932z^4 - 85z^2 + 1}$$

$$= 1 + 92z^2 + 5320z^4 + 301384z^6 + 17066492z^8 + 966656134z^{10} + \dots$$

$m = 7:$ degree of numerator $:$ 64, degree of denominator $:$ 66.

$$1+8z+236z^2+1696z^3+32675z^4+301384z^5+4638576z^6+49483138z^7+\dots$$

$m = 8:$ degree of numerator $:$ 206, degree of denominator $:$ 208.

$$1+596z^2+175294z^4+49483138z^6+13916993782z^8+3913787773536z^{10}+\dots$$

The reader may check the obvious property $h_{m,n} = h_{n,m}$ from these data.

10.2 Data of the computed automata (matrices)

	num. of states		num. of transitions	
m	$\mathcal{A}_m$	$\tilde{\mathcal{A}}_m$	$\mathcal{A}_m$	$\tilde{\mathcal{A}}_m$
1	3	3	3	3
2	5	4	6	5
3	10	6	20	14
4	22	13	64	44
5	52	22	224	121
6	128	74	803	543
7	324	117	2966	1396
8	836	461	11133	7349
9	2189	728	42409	18285
10	5799	3094	163295	105154
11	15512	4828	634700	255196
12	41836	21552	2486247	1556317

The number $\sharp Q'_m$ of states of $\mathcal{A}_m$ equals $mot_m + 1$. The size $\sharp Q''_m$ of $\mathbf{A}_m$ equals $mot_m - 1$. Similarly, $\tilde{\mathcal{A}}_m$ is bigger by 2 than $\tilde{\mathbf{A}}$.

10.3 Computation times and memory use

m	cpusec	bytes
1	.0394	24
2	.0448	46
3	.0656	150
4	.0946	254
5	.9802	2166
6	20.4996	21920
7	2189.6090	350512
8	2251385.5	26895646

These are the average computation times for the generating functions. The bytes column shows the maximum memory need of the main datastructures of the program. The computations reported here have been done on a SPARC 2 (30MHz). The computation for $m = 8$ was performed on a SPARC 10.

References

[1] Berstel, Jean and Reutenauer, Christophe : *Rational series and their languages*. Springer, Berlin 1988.

[2] Hopcroft, John E. and Ullman, Jeffrey D. : *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley 1979.

[3] Kreweras, Germain : Dénombrement des Cycles Hamiltoniens dans un Rectangle Quadrillé. *European Journal of Combinatorics* **13** (1992), 473-467.

[4] Kwong, Harris : Enumeration of Hamiltonian Cycles in $P_4 \times P_n$ and $P_5 \times P_n$. *Ars Combinatoria* **33** (1992), 87-96.

[5] Kwong, Harris and Rogers, D. G. : A Matrix Method for Counting Hamiltonian Cycles on Grid Graphs. To appear in *European Journal of Combinatorics*.

[6] Lidl, Rudolf and Niederreiter, Harald : *Finite Fields* (Encyclopedia of Mathematics and its Applications, vol. 20), Cambridge University Press, 1984.

[7] Stanley, Richard P. : *Enumerative Combinatorics*. Wadsworth & Brooks/Cole, Monterey, California 1986.

[8] Stoyan, Robert : *Anzahl der Hamilton-Kreise in rechteckigen Gittern*. Studienarbeit im Fach Informatik an der Universität Erlangen-Nürnberg, 1993.