Max van Delft

Spanning Trees on Lattices
Combinatorics and Probability

Bachelor thesis

9/04/2019

Thesis supervisors: dr. E.A. Verbitskiy

Leiden University
Mathematical Institute

Abstract

In 1993 Häggström [3] derived a characterisation of the uniform spanning tree on the $\{0,1\} \times \mathbb{Z}$ lattice, or shorthand 2-lattice. He proposed a way to extend to the $\{0, ... m-1\} \times \mathbb{Z}$ lattice, or shorthand m-lattice. We propose a characterisation for the m-lattice that is slightly more compact, but the main ideas are the same. We use a different representation of spanning trees, or rather it is a representation for so called special forests, namely a sequence of letters and partitions. A special forest may be viewed as a spanning tree in the making. We have found a characterisation of the uniform spanning tree on the m-lattice. The results may be extended to so called repetitive graphs. Others have used the same representation of special forests to find a recurrence relation for the number of spanning trees and of Hamilton cycles of various (finite) repetitive graphs. Finally we discuss the matrix tree theorem and the Markov chain tree theorem. Combining these two theorems immediately proves a formula for the stationary distribution of an irreducible, aperiodic Markov chain.

Contents

1 Introduction

We will discuss various topics concerning spanning trees on the lattice. A characterisation of the uniform spanning tree on the $\{0, ... m-1\} \times \mathbb{Z}$ lattice, or shorthand m-lattice, has been derived. But first section earlier work by Häggström on the uniform spanning tree on the 2-lattice is discussed. Thereafter we introduce a representation of spanning trees on the m-lattice through so called special forests.

What follows are our main results on the uniform spanning tree on the m-lattice. Then, following work by Desjarlais and Molina, we will show how recursion formulas for the number of spanning trees of various graphs can be found using the same representation of spanning trees. With a similar approach the recursion formulas for the number of Hamilton cycles can be obtained. Most results can be extended to what we call repetitive graphs. Finally we give a proof of the matrix tree theorem and derive a well-known formula for the stationary distribution of an aperiodic Markov chain.

2 Preliminaries

2.1 Finite graphs

Given a finite connected undirected graph $G = (V, E)$ with vertex set V and edge set E. Define $n := |V|$.

Definition 2.1. A *spanning tree* is a subgraph of G that has $n-1$ edges and such that there is a path between any pair of vertices.

It follows that a spanning tree has no loops.

Definition 2.2. A *spanning forest* is a subgraph of G without loops and with all vertices in its vertex list.

2.2 Random Walk

Later on we want to uniformly walk over a multigraph. For irreducible aperiodic multigraphs there exists a random walk such that each walk has the same probability. First we have to define a random walk on a multigraph.

Let G be a multigraph with edge probability matrix $\tilde{P}$. The edge probabilities are denoted as $\tilde{p}(v_1, v_2, e)$, and they must satisfy $\sum_{v_2,e} \tilde{p}(v_1, v_2, e) = 1$. $\tilde{P}$ can be viewed as a two-dimensional matrix over nodes, its elements of are sequences of edge probabilities.

Definition 2.3. The sequence $(E_n)_{n \in \mathbb{N}_1}$ of random variables with initial distribution μ on the vertices, i.e. $\mu_i = P(X_0 = i)$, is called a *unilateral random walk* on G, and is defined as follows:

1. Let P be the probability matrix such that $p(v_1, v_2) = \sum_e \tilde{p}(v_1, v_2, e)$.
2. Let $(X_n)_{n\in\mathbb{N}}$ be a Markov chain with transition matrix P, and initial distribution μ.
3. Let $P(E_k = e) = \dfrac{\tilde{p}(X_{k-1}, X_k, e)}{\sum_{e'} \tilde{p}(X_{k-1}, X_k, e')}$ for all $k \in \mathbb{N}_1$.

It is not difficult to prove that $(E_n)_{n\in\mathbb{N}_1}$ as defined above has the Markov property. Also we have:

$$P(E_1 = e_1, \dots E_k = e_k) = \mu_{x_0} \cdot \prod_{i=0}^{k-1} \tilde{p}(x_i, x_{i+1}, e_{i+1}) \tag{1}$$

where $(x_i)_{0\leq i\leq k}$ is the the sequence of nodes induced by $(e_i)_{1\leq i\leq k}$.

Analogously a bi-infinite random walk on G may be defined by letting $n \in \mathbb{Z}$. The advantage of this definition of a random walk compared to taking a walk over for instance the edges of G (or equivalently, on the line graph of G) or over the expansion (putting a node on the middle of each edge) of G is that the number of nodes is not increased. The random walk part is performed by the Markov chain $(X_n)_{n\in\mathbb{N}}$ over the graph G', where the multi-edges in G are replaced by a single edge. After such a random walk on G' is performed, the only thing left to do is to correctly sample the edges to obtain the random walk on G.

2.3 Perron-Frobenius theory

Definition 2.4. A matrix $A \in \mathcal{M}_d(\mathbb{R}_{\geq 0})$ is *primitive* if there exists a natural number k such that the k^{th} power of A is positive.

Remark 2.1. Every non-negative irreducible aperiodic matrix is primitive. The converse is also true.

Theorem 2.2 (Perron-Frobenius Theorem)**.** *Let A be a non-negative irreducible matrix. Then:*

1. *There exists a real eigenvalue λ, called the Perron-Frobenius eigenvalue, such that $|\lambda| \geq |\lambda'|$ for all other eigenvalues λ' of A, the inequality being strict if and only if A is primitive.*
2. *The number of eigenvalues with the same largest absolute value is equal to the period of A.*
3. *The left and right eigenspaces of λ are one-dimensional.*
4. *The left and right eigenvectors are real, and can be chosen to be strictly positive.*

The above theorem lists some facts about the largest eigenvalue of a non-negative irreducible matrix. However, they may be generalised to arbitrary non-negative matrices. In this case statements 2 and 3 and the second part of statement 1 are no longer true. Also in statement 4 the eigenvectors are non-negative, not necessarily strictly positive, see [1].

2.4 Parry matrix

Let $A \in \mathcal{M}_d(\mathbb{R}_{\geq 0})$ be a primitive matrix. Let λ be its Perron-Frobenius eigenvalue, with associated left and right eigenvectors u and v.

Definition 2.5. The Parry matrix P associated to A is defined as:

$$p_{ij} = a_{ij} \frac{v_j}{v_i \lambda} \tag{2}$$

Remark 2.3. The Parry matrix is a probability matrix. It induces a Markov chain over G in which edge ij is present if and only if $a_{ij} > 0$. Its stationary distribution π satisfies: $\pi_i = \frac{u_i v_i}{uv}$.

Remark 2.4. The notion of Markov chains may be extended to graphs with multi-edges, i.e. with adjacency matrix satisfying $A \in \mathcal{M}_d(\mathbb{N})$. We call such walks Markovian random walks. In this case we define edge probability matrix $\tilde{P}$ by $\tilde{p}(i,j,e) = \frac{v_j}{v_i \lambda} \mathbb{1}_{a_{ij}>0}$. For more details, see [12].

2.5 Parry measure

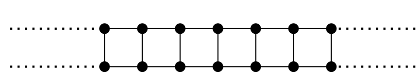
The uniform measure on bi-infinite paths[1] of an irreducible aperiodic graph without multi-edges is better known as the *Parry measure* or *measure of maximal entropy*. It has maximal entropy (information), since for this measure it is most difficult to guess a bi-infinite path right. For more details we refer to [6].

3 Spanning trees on the 2-lattice

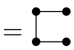
In this section we will discuss how Häggström represented spanning trees on the bi-infinite 2-lattice and how he characterised the uniform spanning tree.

3.1 Introduction

We present Häggström's [3] main ideas, developed for his thesis, on spanning trees on the $\{0,1\} \times \mathbb{Z}$ lattice, or shorthand 2-lattice, which is defined as follows:

$L_2 =$ 

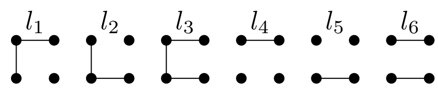
[1]for any path $x_0 ... x_k$, the probability on cilinder set $\pi[x_0, ... x_k]$ is the limiting fraction $2n + k$-step paths with x in the middle

We start with dividing the graph into small blocks: $B =$  .

Pasting infinitely many blocks B together yields L_2.[2] The objective in this chapter will be to paste infinitely many subgraphs of this block together in such a way that they form a spanning tree.

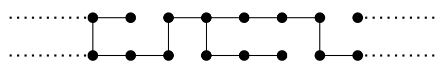
3.2 Letters

In spanning trees the following subgraphs of B can occur:



The above subgraphs of B are called *letters*. We define the *alphabet* Σ to be the set of letters, that is $\Sigma = \{l_1, \ldots, l_6\}$.

Every spanning tree on L_2 can be written as a sequence of letters. For instance:

 $= \ldots l_3 l_5 l_1 l_3 l_6 l_4 l_2 \ldots$

The letter l_6 is called the identity letter: It can follow any letter and when generating a spanning tree from left to right (see 3.5) adding it does not affect the letters that can be appended to the subtree.

3.3 Combining two letters

It is clear that the sequence $l_3 l_1$ cannot occur in a spanning tree, since it causes a loop. Also the combinations $l_1 l_4$ and $l_1 l_5$ are not possible, since in this case the spanning tree would not be connected, whatever the other letters may be.

It is useful to determine which letters can follow one another. For example, l_1 can be succeeded by l_1, l_2, l_3 and l_6. The complete behaviour is captured by a matrix A, where $a_{ij} = \begin{cases} 1 & \text{if } l_i l_j \text{ could be part of a spanning tree} \\ 0 & \text{otherwise} \end{cases}$.

$$A = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{array} \begin{array}{c} \begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

[2]We can write $L_2 = B^{\infty}$. Here the multiplication operator $(\cdot)$ stands for the so called 'paste operation' on graphs, which we will define later on.

3.4 Generating a spanning tree, naïve approach

We would like to generate spanning trees through this matrix A. Therefore

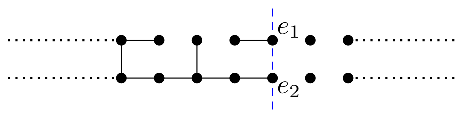


Figure 1: Example of a part of a spanning tree on L_2

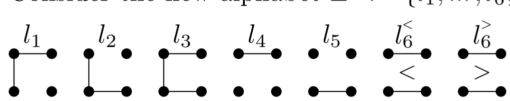
suppose we have a previously generated part of a spanning tree (subtree) starting from -∞ (not necessarily connected yet[3]) and assume we only know its last letter. Can we use A to choose a successive letter? Which letters can follow in figure 1 and can it be derived from A?

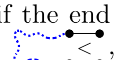
The matrix A shows all 2-letters combinations that have the *potential* to occur at some point in a spanning tree.
If $l_1, ..., l_4$ or l_5 is found at the end of the partially generated (sub)tree, then the letters which A suggests to be candidate successors are always correctly appendable to the subtree.[4]
But if the last generated letter was l_6 (as in figure 1), then whether it can be succeeded by l_1, l_2, l_3 or l_4, l_5 depends on whether the end vertices e_1, e_2 of the generated part are connected.

There is a simple unique reason why this naïve approach does not work: loops over a larger distance must be avoided and to do so the previous letter alone does not provide enough information. However, we are close, since using A to find the next possible letters is completely sound apart from when the last letter was l_6.

3.5 Generating a spanning tree

Consider the new alphabet $\Sigma^* := \{l_1, ..., l_6^<, l_6^>\}$ with letters:



The letter $l_6^<$ can be appended if the end vertices in the previously generated part of the tree is connected, i.e. , and if not then $l_6^>$ can be appended.
We now get the following adjacency matrix:

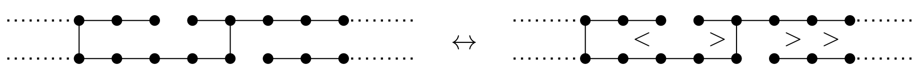
[3]If the generated part is not connected, then it is technically not a tree but rather a special kind of forest. For simplicity we avoid defining this for now.

[4]These letters determine whether the vertices e_1, e_2 are connected or not, independent of the past. The connectivity of e_1, e_2 in turn determines which letters can follow. This is the key observation, which will be explained in more detail later on.

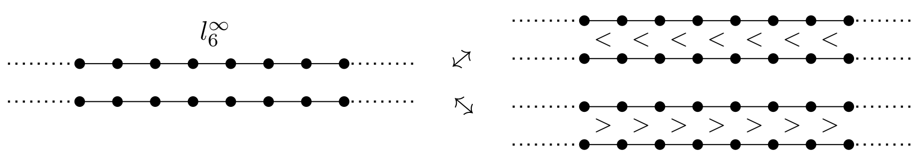
$$
A^* = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6^< \\ l_6^> \end{array}
\begin{array}{c} \begin{array}{ccccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6^< & l_6^> \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1
\end{pmatrix} \end{array} \tag{3}
$$

From this matrix no loops can be created and we can generate all spanning trees. This is because any combination of letters that can occur in a normal spanning tree can be associated with a combination in A^*. The correspondence is unique, except for $l_6 l_6$, because $l_6^< l_6^<$ and $l_6^> l_6^>$ both represent this piece of tree.

Equivalently, we can convert any sequence of normal letters into one with the new letters, for example:

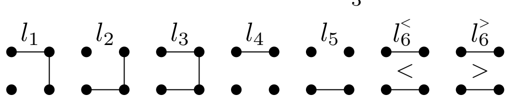


apart from $l_6^\infty := \prod_{-\infty}^{\infty} l_6$, because it has two associated trees (forests):



3.6 Mirrored Letters (3-letters)

We can do the same for letters with a vertical edge on the right, which we call *3-letters*. The new letterset Σ_3^* becomes:



And the adjacency matrix is:

$$
A_3^* = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6^< \\ l_6^> \end{array}
\begin{array}{c} \begin{array}{ccccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6^< & l_6^> \end{array} \\
\begin{pmatrix}
1 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix} \end{array} \tag{4}
$$

3.7 Random Walk

Let G be the graph with adjacency matrix A_3^* and with nodes being the letters in Σ_3^*.[5] By putting transition probabilities in A_3^* we obtain a Markov chain $(X_n)_{n\in\mathbb{Z}}$ on G, where $X_n \in \Sigma^*$. Let T be the graph defined by recording the letters X_n. By properties of A_3^* this graph will not contain a loop, hence it will be a forest. But it will not necessarily be connected. However, from the Markov chain stationary distribution we know for instance that l_3 will occur infinitely many times, so T will be connected infinitely many times (with probability 1). Informally the Markov chain induces a probability measure (concentrated) on spanning trees.

3.8 Uniform Spanning Tree

The Parry Markov chain on an irreducible aperiodic graph induces a uniform measure on bi-infinite paths. Clearly the graph G with adjacency matrix A_3^* is aperiodic, since it contains a self-loop. Hence the Parry matrix is well defined. Using the Parry measure Häggström determined the probabilities for which the random walk on G would be a uniform spanning tree:

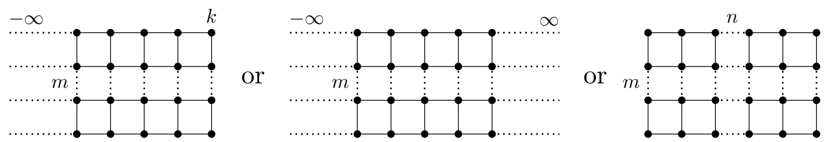
$$P = \begin{pmatrix} \frac{1}{3}p_1 & \frac{1}{3}p_1 & 0 & \frac{1}{2}(1-p_1) & \frac{1}{2}(1-p_1) & \frac{1}{3}p_1 & 0 \\ \frac{1}{3}p_1 & \frac{1}{3}p_1 & 0 & \frac{1}{2}(1-p_1) & \frac{1}{2}(1-p_1) & \frac{1}{3}p_1 & 0 \\ \frac{1}{3}p_1 & \frac{1}{3}p_1 & 0 & \frac{1}{2}(1-p_1) & \frac{1}{2}(1-p_1) & \frac{1}{3}p_1 & 0 \\ 0 & 0 & 1-p_2 & 0 & 0 & 0 & p_2 \\ 0 & 0 & 1-p_2 & 0 & 0 & 0 & p_2 \\ \frac{1}{3}p_1 & \frac{1}{3}p_1 & 0 & \frac{1}{2}(1-p_1) & \frac{1}{2}(1-p_1) & \frac{1}{3}p_1 & 0 \\ 0 & 0 & 1-p_2 & 0 & 0 & 0 & p_2 \end{pmatrix}$$

with $p_1 = 3(2-\sqrt{3}) \approx 0.8038$ and $p_2 = 2-\sqrt{3} \approx 0.2679$.

[5] We could just as well taken A^* here. However, in this section our intension is to give an overview of Häggström's results, and he worked with 3-letters

4 Special Forest

Let L be one of the following the lattices



Definition 4.1. A forest of G is *special* or *forest with paths to the right* if:

1. All vertices except those on the very right end have a path to the right.

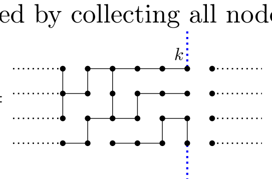
or equivalently if:

2. Every component reaches the very right end (or $+\infty$).

It is not difficult to see that the above definitions are equivalent.

4.1 Part of a spanning tree

Let t be a spanning tree of L defined above, and let t_k be part of the tree obtained by collecting all nodes and edges up to index k. For example:[6]

$t_k =$ 

A graph t_k that can be formed this way is called a *spanning tree generated up to index* k.

Theorem 4.1. *A subgraph of* L *is a spanning tree generated up to index* k *if and only if it is a special forests of* $L \cap \{1, \dots, m\} \times \{-\infty, \dots, k\}$.

Before we derive this, we list some properties of special forests.

4.2 Properties

Special forests have the following properties:

Property 1. A spanning tree, cut of at some index k, is a special forest.

Property 2. A special forest f can become a spanning tree, in the sense that there exists a subgraph G of the $m \times k$-lattice such that appending G to the right hand side of f results in a spanning tree.

[6]The nodes to the right of k are not part of t_k.

Lemma 4.2. *Let L_1, L_2 be lattices of height m, and let G_1 be a subgraph of L_1. Then there exists subgraph G_2 such that $G_1 \cdot G_2$ is a spanning tree if and only if G_1 is a special forest.*[7]

Proof.
'$\Rightarrow$'
Let G_1 be a subgraph of L_1 that is **not** a special forest. Then there exists a component that does not reach the right end. After appending any subgraph G_2 of L_2 to G_1, this component will not get connected to any of the other components. Hence $G_1 \cdot G_2$ will not be connected. This implies that $G_1 \cdot G_2$ is not a spanning tree for any subgraph G_2 of L_2. This proves this direction by rule of transposition (logic).

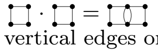
'$\Leftarrow$'
Let G_1 be a special forest of L_1. We will construct G_2, which must be a subgraph of L_2, such that appending it to G_1 yields a spanning tree.
First let G_1 be the subgraph of L_1 with all nodes, but no edges. To make $G_1 \cdot G_2$ a special forest, add all horizontal edges to G_2. Now that $G_1 \cdot G_2$ is a special forest, all components reach the right hand side, but they are not necessarily connected. However there exists a vertical edge that connects two components. Adding this edge does not cause a loop. Now repeatedly add a vertical edge to G_2 that connects two components in $G_1 \cdot G_2$ until the graph is connected. The vertical edges may be added at any index.[8] The resulting graph will be connected, has no loops and contains all vertices. Hence it is a spanning tree. $\square$

Proof of theorem 4.1.
Let L_1 be the subgraph of L with edges and nodes up to index k. Let L_2 be the subgraph of L with edges and nodes starting from index k.

'$\Rightarrow$'
Let t_k be a spanning tree generated up to index k. It is a subgraph of L_1. By definition it is the first part of a spanning tree of L, so there exists a subgraph G_2 of L_2 such that $t_k \cdot G_2$ is a spanning tree. By lemma 4.2 t_k is a special forest.

'$\Leftarrow$'
Let t_k be a special forest. Then by lemma 4.2 there exists a subgraph G_2 of L_2 such that $t_k \cdot G_2$ is a spanning tree of L. Hence t_k is a spanning tree generated up to index k. $\square$

Concluding remarks
Forests with paths to the right are interesting objects of study, because of their relation to spanning trees on the lattice. The term 'special forest' can also be

[7] The product $(\cdot)$ denotes the paste operation on graphs, which is defined in 12. For example, for $m = 1$ we have: 

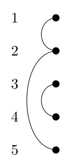
[8] To actually obtain these vertical edges one can keep track of the component structure at the right end of $G_1 \times G_2$, which may be viewed as a partition over the m nodes. The reader should be able to derive the further details after reading section 6.

found in [10], but there it has a different meaning. Next we will construct all special forests, and with it all spanning trees since they are special forests, using the component structure of a previously generated part to find the next possible letters to add.

5 Partitions

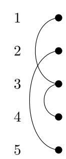
A component structure can be represented as a *partition*. A partition of a set is a collection of subsets that are non-empty and such that every element of the original set lies in exactly one subset. We will look at partitions of the natural numbers $\{1, ..., n\}$. Two elements a, b are said to be equivalent ($a \sim b$) if they lie in the same subset of the partition. A partition on natural numbers (or on any other ordered set) is called *non-crossing* if $a < b < c < d$ and $a \sim c$ and $b \sim d$ implies that $a \sim b \sim c \sim d$.

Non-crossing



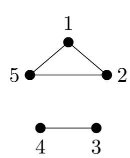
125,34[9]

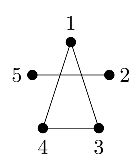
Crossing



134,25

A partition can also be presented in a circle.





A partition is crossing if and only if there are at least 2 intersecting polygons in the circular presentation. The number of non-crossing partitions (ncp's) of $\{1, ..., n\}$ is the n^{th} Catalan number C_n. [14]

[9] 125,34 is shorthand for {{1,2,5},{3,4}}

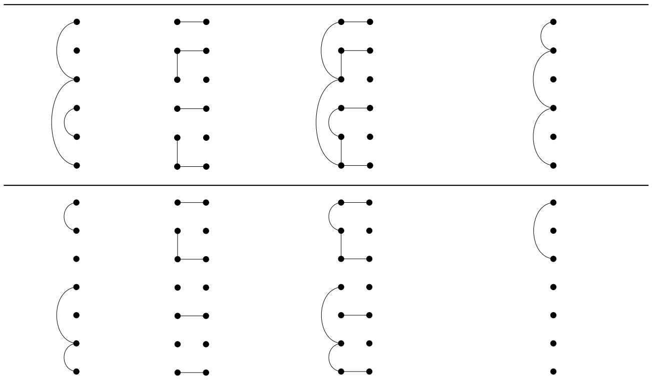
6 Partition-letter transitions

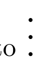
We are going to construct special forests by adding letters to partitions. The partition, as mentioned before, represents the component structure of vertices on the right hand side of a generated part of a special forest. So suppose we have a special forest up to index k and its component structure at the right end. Then a letter can be added to form a special forest up to index $k+1$ if the addition of the letter:
1. does not cause a loop.
2. every node on index k must have a path to the right, just like the other nodes have.

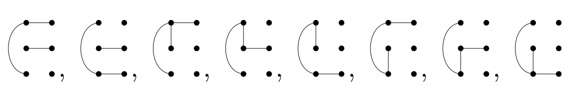
For any partition we can check which letters can follow it, and obtain the end partition, i.e. the component structure on the right hand side that has now shifted one spot.

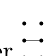
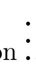
Example 1.

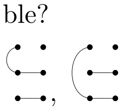
Partition	Letter	Partition, letter	End partition



Example 2.
Which letters can cause a transition from  to  ?

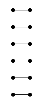


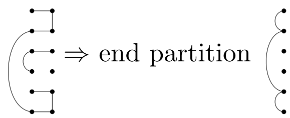
Example 3.
Given the letter  and end partition . Which start partition(s) is (are) possible?



There are 2! Consequently a combination (letter, end partition) does not uniquely determine the starting partition.

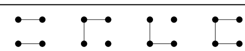
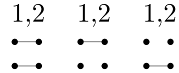
Mirrored letters (3-letters) Instead of letters with vertical edges on the left hand side, we could also consider 3-letters with vertical edges on the right hand side. Lets look at another example to show everything works about the same.

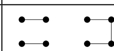
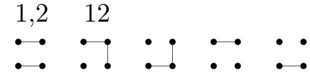
Example 4.
Given partition 1,26,34,5 and a letter . What will be the end partition?



6.1 Partition Graph, Partition Matrix and Count Matrix

Now for $m = 2$ we will derive all possible transitions for both E-letters and 3-letters.

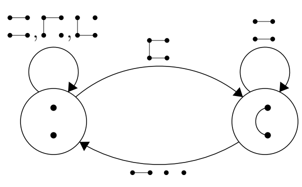
Partition	E-letters			
1,2				
end partition	1,2	1,2	1,2	12
12				
end partition	12	1,2	1,2	

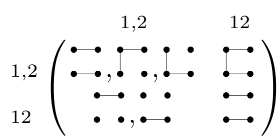
Partition	3-letters				
1,2					
end partition	1,2	12			
12					
end partition	12	12	12	1,2	1,2

This can be presented as a graph or matrix.

Partition Graph **Partition Matrix** **Count Matrix**

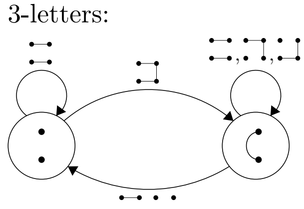
E-letters:

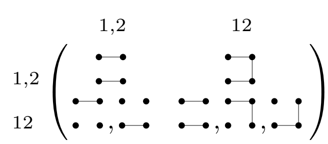




$$A_E = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix}$$

3-letters:



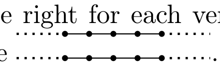
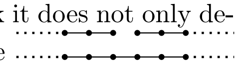


$$A_3 = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$$

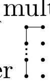
Remark 6.1. The partitions that can occur inside a spanning tree are exactly all non-crossing partitions. The partition graph includes only these partitions.

7 Uniform Spanning Tree

7.1 Walking on the partition graph

A walk on the partition graph corresponds to a special forest by recording the sequence of traversed edges. For all special forests there exists a walk corresponding to it. This is because we exactly allow all transitions that do not cause a loop and such that there is a path to the right for each vertex. This walk does not have to be unique. Take for example . There are 2 walks on the partition graph for $m = 2$ that correspond to this special forest. For a special forest to have more than 1 corresponding walk it does not only depend on whether it has multiple components. For example  has a unique corresponding walk. For doubly infinite special forests with a pair of vertices that could be connected at $+\infty$ or $-\infty$ the corresponding walk on the partition graph is not unique. The probability that a uniformly chosen special forest has this property is 0.

7.1.1 Connectedness of special forests

Consider the doubly infinite random walk over the partition graph by putting strictly positive probabilities on the (multi) edges. Then with probability 1 we get a special forest where the letter  occurs infinitely many times in the direction of $-\infty$ as well as $+\infty$. The special forest will be connected with probability 1. This is because any two vertices both have a path leading to the first occurrence of this letter at the right hand side of both vertices, and hence they are connected. Thus the special forest is with probability 1 a spanning tree.

The random walk over the partition graph is not a Markov chain, since there are multi-edges. However, we derived that the edge probability matrix $\tilde{P}$ such that $\tilde{p}(i, j, e) = \frac{v_j}{v_i \lambda} \mathbb{1}_{\{a_{ij} > 0\}}$ induces a random walk on a multigraph that chooses its overall path uniformly.[10] The Partition graph, together with the matrix $\tilde{P}$, where $\tilde{P}$ may be derived from the adjacency matrix (which we called the count matrix) yields a characterisation of the uniform spanning tree on the bi-infinite lattice.

8 Transformation multigraph to normal graph

We did not know about this matrix $\tilde{P}$ beforehand. In this section we show how we got to the conclusion that the probability on a finite section of the bi-infinite random walk induced by $\tilde{P}$, which we call the *Parry random walk*, is equal to the limiting uniform measure on the finite middle section of $(2n+k)$-step paths.

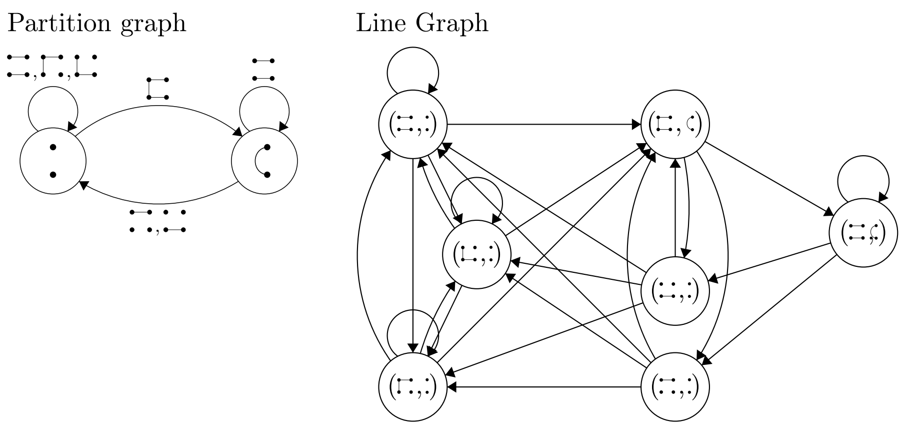
[10] See also remark 2.4

Our objective was to somehow obtain uniform bi-infinite walks on the partition graph, in particular obtaining transition probabilities. The problem was, or so we thought, that the partition graph had multi-edges and hence a Markovian random walk on the partition graph is not a Markov chain. We initially solved this problem as follows.

8.1 Walking on the letter-partition graph

To make the random walk on the partition graph a Markov chain, we set the states to be (letter, end partition). We add the logical edges between these states: there is an edge between (l_1, p_1) and (l_2, p_2) if and only if l_1 can proceed p_1 and l_2 causes partition transition $p_1 p_2$. We call this graph the letter-end-partition graph or simply letter-partition graph.

A walk on the letter-partition graph is similar to a walk on the so called *line graph* of the partition graph. The line graph is the graph where the edges become the nodes and we have the logical edges between them (namely there is an edge if one edge could follow the other). For $m = 2$ this holds, but for $m \geq 3$ there is a difference: some edges get the same label due to the fact that a combination of (letter, end partition) does not always uniquely determine the starting partition (see example 3).



The above line graph is the same as the graph used by Häggström, except for the fact that he worked with 3-letters instead of E-letters, see 3.8.

Not all combinations of (letter, end partition) are possible. For example, if the end partition is 12, the previous letter could not have been .

8.2 Parry Markov chain

Recall that for an irreducible, aperiodic graph G with adjacency matrix A, associated Parry matrix P and bi-infinite Parry Markov chain $(X_n)_{n \in \mathbb{Z}}$, the probability $P(X_0 = x_0, ..., X_k = x_k)$ is equal to that of the limiting uniform measure on $2n + k$-step paths, where x represents a finite middle section of the path. The letter-partition graph is clearly aperiodic, since the states (p, l) where l is the neutral letter have a self loop. Let A be the adjacency matrix of the letter-partition graph. Then the Parry matrix P of A is well-defined. The bi-infinite Markov chain on the letter-partition graph induced by P now describes the uniform special forest.[11] With an argument similar to that in 7.1.1, the uniform measure on special forests is concentrated on spanning trees. We will use the terms uniform spanning tree and uniform special forest interchangeably, even though they are a bit different.

8.2.1 Initial characterisation of the uniform spanning tree

We computed the Parry matrix of the Letter partition graph, and thus obtained a numerical characterisation of the uniform spanning tree. But we observed that many of the values in the Parry matrix occured multiple times, see also 3.8, and so we asked ourselves whether the states (letter, end partition) really required the information of the letter. We already gave the answer. Indeed, they do not. Let us show how we numerically derived this.

8.2.2 Partition matrix with letter-probabilities

First we introduce some new notions:

A letter-partition combination (l, q) is called *meaningful* if there exists a partition p such that l causes transition pq.
Let $(X_n)_{n\in\mathbb{Z}}$ be the Markov chain over meaningful letter-partition combinations, with probabilities conform to the Parry measure.
Let l_1pl_2q be an abbreviation for the transition from (l_1, p) to (l_2, q).
Let $P(l_1pl_2q) = P\left(X_n = (l_2, q) \mid X_{n-1} = (l_1, p)\right)$.

We asked ourselves which of the values in the Parry Matrix coincided, and thought it would be logical that the transition probability $P(l_1pl_2q)$ is independent of l_1 (as long as l_1 can proceed partition p) and equal for all l_2 that cause the transition pq, so we tested it. Indeed numerically this turned out to be the case.[12]

Under this assumption we defined a matrix B such that:

$$b_{pq} = \begin{cases} P(l_1pl_2q) & \text{for any } l_1 \text{ that can occur before } p \text{ and any } l_2 \text{ that causes} \\ & \text{transition } pq\text{, assuming such a letter } l_2 \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

Theorem 8.1. *The partition graph with edge probabilities in B is another way to characterise the uniform spanning tree.*

[11] Again, in hindsight the same probabilities occur in $\tilde{P}$, which may be obtained directly from the count matrix.
[12] We have proven it theoretically.

Proof.
Consider the random walk on the partition graph with the transition probabilities $\tilde{P}(i, j, e) = b_{ij}$. Let us compare it to the Parry Markov chain of the letter-partition graph. In both graphs, starting from any partition p, the exact same letter sequences $\prod_i l_i$ are allowed. Each letter sequence induces the same partition sequence $\prod_i p_i$. The probability of $\prod_i l_i$ is in both cases $\prod_i b_{p_i p_{i+1}}$. In both graphs the same probability is assigned to all bi-infinite walks, and these in turn correspond to spanning trees. Hence the partition graph with edge probabilities in B describes the uniform spanning tree as well. □

8.2.3 Partition graph probability matrix

Let M be the partition matrix, where elements are sets of letters. Let A be the count matrix (adjacency matrix) corresponding to M. Define the matrix C such that $c_{pq} = a_{pq} b_{pq}$ for all partitions p, q. Then C is a probability matrix, c_{pq} is the probability to go from partition p to q and $\frac{c_{pq}}{a_{pq}}$ is the probability to go from p to q with letter l for any l that causes transition pq in a uniform spanning tree. The pair (M, C) characterises the uniform spanning tree.

8.2.4 States with start partition

Analogous to before, we may define the start-partition-letter graph with nodes (start partition, letter), the logical edges between them.[13] This graph, together with the probabilities obtained from its Parry matrix also yields a characterisation of the uniform spanning tree. Numerically we saw that $P(p_1 l_1 p_2 l_2)$ did not depend on p_1 and l_1, which was quite surprising to us at first. Then we realised that the probability to go from e_1 to e_2 in the line graph is equal to the probability to go from the **end** node of e_1 to the **end** node of e_2. Hence it is not very clever to describe the edges as (start node,edge). We defined the matrix B with: $b_{pq} = P(x l_1 p l_2)$ for any (x, l_1) that can proceed p and any l_2 that causes transition pq.[14] We saw numerically that this matrix B coincided with the one we defined earlier.

This concludes how we numerically conjectured that we could put probabilities $\tilde{p}(i, j, e)$ on the edges of a multigraph such that the bi-infinite paths are uniform. Much later we observed that the matrix C defined above was equal to the Parry matrix of the count matrix. This led us to prove formally that $\tilde{p}(i, j, e) = \frac{v_j}{v_i \lambda} \mathbb{1}_{\{a_{ij} > 0\}}$.[15]

[13] The start-partition-letter graph is isomorphic to the line graph of the partition graph: the edges in the partition graph become the nodes, but unlike in the letter-partition graph no edges are contracted to form a single node.

[14] Any sequence $x l_1 p l_2$ induces the sequence $x l_1 p l_2 q$ for q such that l_2 causes transition pq. If you think about it, then it is very logical that the probability to go from one edge to the other, the probability of this transition is equal to the probability to go from the end node of the first edge to the end node of the second edge. These end nodes are p and q.

[15] We realised that the Parry matrix was a probability matrix for all non-negative primitive matrices $A \in \mathcal{M}(\mathbb{R}_{\geq 0})$

8.3 Numerical results

Summing up, we have obtained six equivalent characterisations of the uniform spanning tree. The following tables show the number of nodes and edges of the corresponding graphs:

	Partition graph			
	E-letters		3-letters	
m	$\|V\|$	$\|E\|$	$\|V\|$	$\|E\|$
2	2	7	2	7
3	5	58	5	55
4	14	523	14	484
5	42	4984	42	4563

	Letter-partition graph				Partition-letter graph			
	E-letters		3-letters		E-letters		3-letters	
m	$\|V\|$	$\|E\|$	$\|V\|$	$\|E\|$	$\|V\|$	$\|E\|$	$\|V\|$	$\|E\|$
2	7	26	7	26	7	26	7	26
3	53	671	50	630	58	727	55	682
4	495	20751	456	18713	523	21729	484	19635
5	4858	666364	4437	588755	4984	680232	4563	601697

Let $A \in \mathcal{M}_d(\mathbb{N})$ be the adjacency matrix of the partition graph. Let $\tilde{P}$ be an edge probability matrix such that $\tilde{p}(i,j,e) = \frac{v_j}{v_i \lambda} \mathbb{1}_{\{a_{ij}>0\}}$.

Now, it is interesting to compute $\tilde{P}$, since for any letter l we know that $\tilde{p}(p,q,l)$ is the probability of the transition pq through l.

For $m = 2$ we have:

$$\tilde{P}$$

E-lettes

$$\begin{array}{c} \\ 1{,}2 \\ 12 \end{array} \begin{array}{cc} 1{,}2 & 12 \\ \left(\begin{matrix} 0.268 & 0.196 \\ 0.366 & 0.268 \end{matrix}\right. & \left.\right) \end{array}$$

3-letters

$$\begin{array}{c} \\ 1{,}2 \\ 12 \end{array} \begin{array}{cc} 1{,}2 & 12 \\ \left(\begin{matrix} 0.268 & 0.732 \\ 0.098 & 0.268 \end{matrix}\right. & \left.\right) \end{array}$$

For $m = 3$:

$\tilde{P}$

E-letters:

	1,2,3	1,23	13,2	12,3	123
1,2,3	0.080	0.056	0.099	0.056	0.039
1,23	0.113	0.080	0.140	0.080	0.055
13,2	0.064	0.045	0.080	0.045	0.032
12,3	0.113	0.080	0.140	0.080	0.055
123	0.163	0.115	0.201	0.115	0.080

3-letters:

	1,2,3	1,23	13,2	12,3	123
1,2,3	0.080	0.216	0	0.216	0.489
1,23	0.029	0.080	0	0.080	0.181
13,2	0.025	0.067	0.080	0.067	0.151
12,3	0.029	0.080	0	0.080	0.181
123	0.013	0.035	0.042	0.035	0.080

There exist algorithms to pick a spanning tree uniformly. We have generated a uniform spanning tree for an $m \times n$ lattice for large n using Wilson's algorithm. We repeated the algorithm many times and stored the partition-letter transition that was observed in the middle of the graph.[16] This allows us to derive an estimate for the probability of each transition. We will do this in the next section and compare the observed probabilities with those in $\tilde{P}$.

9 Wilson's Algorithm

Wilson's Algorithm is an algorithm for picking a spanning tree of a graph uniformly at random. It uses the concept of a loop-erased random walk (LRW) on a graph. The loop erased random walk is a simple random walk on a graph (the successor of the current node is chosen uniformly from the neighbours at each time step), but with the loops being erased as soon as they occur.

Pseudocode for Wilson's Algorithm

Order vertices in vertex list V.
Pick end vertex $e \in V$. Let $Q := \{e\}$.
while($Q \neq V$){
  Let $v \in V$ be the first vertex such that $v \notin Q$.
  Perform Loop-erased Random Walk starting from v, stop when Q is hit.
  - Store the edge list obtained from the LRW

[16] Wilson's algorithm even allows us to generate a finite part of an infinite spanning tree on the m-lattice, but then the walk may get too far off the middle

```
07      - Add the not-erased vertices from the LRW to Q.
08   }
```

The vertices $e, v \in V$ in the second and fourth line do not have to be chosen randomly. One is free to pick an end vertex and may choose the order of the vertices where the LRW should make its restart. Wilson's algorithm is therefore very powerful.

9.1 Implementation

We have implemented Wilson's Algorithm for the $m \times n$ lattice in C++, with a slight modification. Instead of drawing a full spanning tree, we stop when the middle letter is finished. The middle letter is finished precisely when all vertices in the middle letter are connected. For the end- and start- vertices we choose those that lie in the middle letter. Then we run Wilson's algorithm until there are no more starting vertices, i.e. until all vertices in the middle letter lie in the tree and hence are connected. This way we obtain statistics for occurrences of the middle letter. Also after each run of the algorithm the connectivity classes, which rely only on the edges lying to the left of the letter, are obtained. For n chosen sufficiently large, results are similar to those of the $\{0, ..., m-1\} \times \mathbb{Z}$ lattice.

9.2 Edge probabilities

Inside the following tables are the transition probabilities as we found them in the spanning trees obtained doing Wilson's algorithm, with $n = 100$, repeating the algorithm 100.000 times.

	1,2	12
1,2	 0.269  0.268  0.269	 0.195
12	0.368 0.366	0.266

	1,2	12
1,2	 0.268	 0.732
12	0.099 0.101	0.266 0.267 0.266

There is only minimal difference with Häggström's theoretical values in section 3.8.

Also the results on $m = 3$ are shown on the following page. Comparing them with the theoretical values in 2 we see that they are consistent with $\tilde{P}$.

E-Letters:

	1,2,3	1,23	13,2	12,3	123
1,2,3	0.082 0.080 0.080 0.082 0.080 0.078 0.080 0.079	0.056 0.056	0.095	0.057 0.057	0.039
1,23	0.116 0.116 0.111 0.115 0.113	0.076 0.080	0.143	0.078	0.053
13,2	0.066 0.064 0.068 0.065 0.065 0.061 0.063 0.062	0.044 0.048	0.083 0.082 0.080	0.044 0.042	0.032 0.031
12,3	0.116 0.113 0.116 0.116 0.114	0.073	0.137	0.078 0.082	0.056
123	0.166 0.167 0.156	0.115	0.199	0.117	0.080

3-Letters:

	1,2,3	1,23	13,2	12,3	123
1,2,3	0.083	0.225		0.222	0.470
1,23	0.031 0.027	0.078 0.080 0.079		0.080 0.082	0.178 0.184 0.181
13,2	0.030 0.023	0.064 0.063	0.087	0.064 0.065	0.140 0.160 0.152 0.151
12,3	0.031 0.030	0.082 0.076		0.079 0.076 0.077	0.183 0.182 0.183
123	0.014 0.014 0.013	0.034 0.035 0.034 0.035	0.041	0.035 0.035 0.036 0.036	0.080 0.078 0.080 0.081 0.081 0.079 0.081 0.079

10 Partition Matrix Computation

10.1 Introduction

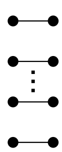
In the context of partition transitions, some letters are more simple than others, and often the effect of a letter may be expressed in the effects of simpler letters. It is like some letters are a kind of prime elements, and others may be decomposed into prime factors.

It is relatively easy to compute a partition transition induced by a prime letter. Now for each letter there is a prime factorisation, and a partition transition is also easily computed as a kind of multiplication of the partition transitions of the prime factors. That enables us to generate partition matrices in a simple and efficient way.

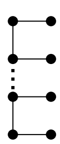
In this chapter we explain this in detail. We will use a convenient ad hoc notation.

10.2 Horizontal and vertical letters

It will be beneficial to distinguish between horizontal and vertical letters. Horizontal letters consist of only horizontal edges: one or more. It is a subgraph of:



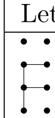
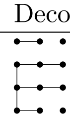
Vertical letters may be presented as follows:



In a vertical letter all horizontal edges must be present. Any combination of vertical edges is allowed. The partition change is only caused by the vertical edges.

10.3 Horizontal and vertical components

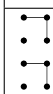
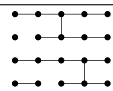
All letters may be split up into a horizontal and vertical letter. For example:

Letter	Decomposition
	

.

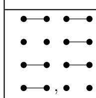
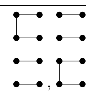
These are equivalent in the sense that they cause the same partition change when appended to a special forest. This is not that useful, yet.

However, letters may be decomposed into even simpler building blocks. For example:

Letter	Decomposition
	

.

The decomposition consists of the two horizontal letters and the two vertical letters:

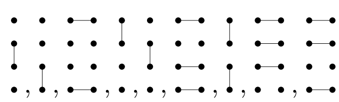
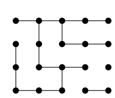
Horizontal primes	Vertical primes
	

.

The letters in the decomposition are the simplest in terms of partition change. This will be discussed in more detail later. But first we introduce a different representation for the vertical letters.

10.4 Short hand form for vertical letters

The horizontal edges in the above representation of vertical letters are redundant. If we leave them out the partition transition behaviour will remain the same.

Sequencing the letters:  now results in: .

This notation is:

- Simple
- In line with the horizontal letters as only vertical edges are utilized
- Clearly applicable in the domain of 3-letters [17]

But it has a downside:

- A multi-edge arises when the same vertical edge is added twice in a row. Hence the resulting graphs are not necessarily part of the usual grid.

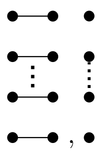
[17] The old representation is also applicable in the domain of 3-letters.

This downside is something to be aware of, but it will not have any impact on the results in this chapter.

10.5 The two neutral letters and primes

Neutral letters
The *neutral letter* is defined as a letter that does not change the component structure and is allowed to be appended to any special forest.

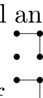
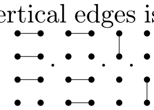
There is a horizontal and a vertical neutral letter:



Prime letters
A horizontal letter with precisely one absent horizontal edge is called a *horizontal prime*. A vertical letter with precisely one vertical edge is called a *vertical prime*. Horizontal and vertical primes differ the least from their respective neutral letter. This makes their partition transitions the simplest.[18]

10.6 Prime factorisation

In general an E-letter, and likewise for 3-letters, is characterised by its horizontal and vertical edges, or equivalently by the present vertical edges and the absent horizontal edges. The corresponding prime elements, first the vertical and then the horizontal primes, form the prime factorization, for E-Letters that is.

For 3-letters the order of horizontal and vertical edges is reversed. For example: one possible prime factorisation of  is: . See section 10.3. The order of the horizontal primes on the left may be changed, as for the vertical primes on the right.

10.7 Count matrix computation

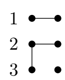
There is a method to compute the count matrix using only the partition transition matrices of prime letters. Recall that in the count matrix each element counts the number of letters that cause a certain partition transition.

[18]Note that in the previous representation of the vertical letters there would have been only one neutral letter. The primes are exactly those that differ the least from the neutral letter.

10.7.1 Adjacency matrix

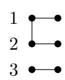
Letters, and subgraphs of the $m \times n$ grid in general, have a certain effect on each partition. This behaviour is captured by a *partition transition matrix* or *adjacency matrix*.

For example:

The letter  has adjacency matrix:

	1,2,3	12,3	1,23	13,2	123
1,2,3	1	0	0	0	0
12,3	0	1	0	0	0
1,23	0	0	0	0	0
13,2	0	0	0	1	0
123	0	0	0	0	0

.

The letter  has adjacency matrix:

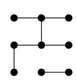
	1,2,3	12,3	1,23	13,2	123
1,2,3	0	1	0	0	0
12,3	0	0	0	0	0
1,23	0	0	0	0	1
13,2	0	0	0	0	1
123	0	0	0	0	0

.

Every row contains at most one non-zero element.

10.7.2 Matrix multiplication

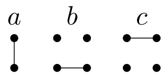
The multiplication of two adjacency matrices results in a new adjacency matrix, namely the adjacency matrix corresponding to the first and second graph pasted together.

Example: The adjacency matrix of:  may be obtained by multiplying the adjacency matrices of the individual letters shown in the previous section:

	1,2,3	12,3	1,23	13,2	123
1,2,3	0	1	0	0	0
12,3	0	0	0	0	0
1,23	0	0	0	0	0
13,2	0	0	0	0	1
123	0	0	0	0	0

10.7.3 Count matrix m=2

There are three prime letters: a, b and c.



Let A, B, C denote their respective adjacency matrices. Let $\tilde{A} = A+I, \tilde{B} = B+I$ and $\tilde{C} = C + I$. Then the count matrix for E-letters is $\tilde{A}\tilde{B}\tilde{C}$, and for 3-letters is $\tilde{B}\tilde{C}\tilde{A}$.[19]

Why do we add the identity matrix? One could interpret the matrix $\tilde{A}$ as: either we add prime a, or not. Let us explain this more briefly.

In ABC every path corresponds to the behaviour of the letter abc[20]. In $\tilde{A}\tilde{B}\tilde{C}$ still every path corresponds to a letter. However, the diagonals are not 0. If a path contains a certain diagonal element x_{ii} then the prime letter x is simply disregarded by this path. The path corresponds to the behaviour of a letter that does not have x as a prime. By going through all the diagonals in a path, we see which primes are absent in the letter that corresponds to the path. Therefore every path still corresponds to a certain letter.

We see that $\tilde{a}_{ij}\tilde{b}_{jk}\tilde{c}_{kl} = \begin{cases} 1 \text{ if the corresponding letter causes the transition } il \\ 0 \text{ otherwise} \end{cases}$.
Since $\tilde{A}\tilde{B}\tilde{C}_{ij}$ sums over all paths from partition i to j, it counts all letter transitions between them. This is because every letter that causes transition ij has a prime factorisation, so it must have a corresponding path of partitions from i to j and hence it must be contained in $\tilde{A}\tilde{B}\tilde{C}_{ij}$ since it sums over all paths.

10.7.4 Count matrix

The above findings hold for general m, the only difference being that there are more primes and more partitions. The prime letters are: $v_1, ..., v_{m-1}, h_1, ..., h_m$. Let $V_1, ..., V_{m-1}, H_1, ..., H_m$ be their corresponding adjacency matrices. Let $\tilde{V}_i = V_i + I$ and $\tilde{H}_i = H_i + I$ for all i. Let $\tilde{V} = \prod_{i=1}^{m-1} \tilde{V}_i$ and $\tilde{H} = \prod_{i=1}^{m} \tilde{H}_i$. Then the count matrix for E-letters is $\tilde{V}\tilde{H}$ and for 3-letters it is $\tilde{H}\tilde{V}$.

10.8 Partition matrix computation

Recall that in the partition matrix each element is the set of letters that all cause a particular partition transition. Instead of counting letters (paths of primes) between partitions we are now interested in the letters (paths) themselves. The paths can be revealed through a modification of the prime letter's adjacency matrices. This allows us to compute the partition matrix.

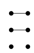
[19] $\tilde{B}$ and $\tilde{C}$ commute, since in terms of partition change it doesn't matter which horizontal edge is removed first.

[20] Though $ABC = 0$ since $BC = 0$

10.8.1 Prime letter matrices

In a prime letter matrix each 1 in the prime's adjacency matrix is replaced with the letter itself.

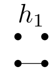
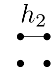
For example:

Prime $h_3 =$  has prime letter matrix:
$$\begin{array}{c|ccccc} & 1,2,3 & 12,3 & 1,23 & 13,2 & 123 \\ \hline 1,2,3 & 0 & 0 & 0 & 0 & 0 \\ 12,3 & 0 & 0 & 0 & 0 & 0 \\ 1,23 & h_3 & 0 & 0 & 0 & 0 \\ 13,2 & 0 & 0 & 0 & 0 & 0 \\ 123 & 0 & h_3 & 0 & 0 & 0 \end{array}.$$

The multiplication of the new matrices requires a kind of multiplication and addition for the elements. Before we discuss this in more detail, we show how the partition matrix is computed for $m = 2$.

10.8.2 Partition matrix m=2

We have the following neutral and prime letter matrices:

	Neutral letter	Prime letters		
Letter	e 	v 	h_1 	h_2 
Matrix	I: $\begin{array}{c\|cc} & 1,2 & 12 \\ \hline 1,2 & e & 0 \\ 12 & 0 & e \end{array}$	V: $\begin{array}{c\|cc} & 1,2 & 12 \\ \hline 1,2 & 0 & v \\ 12 & 0 & 0 \end{array}$	H_1: $\begin{array}{c\|cc} & 1,2 & 12 \\ \hline 1,2 & 0 & 0 \\ 12 & h_1 & 0 \end{array}$	H_2: $\begin{array}{c\|cc} & 1,2 & 12 \\ \hline 1,2 & 0 & 0 \\ 12 & h_2 & 0 \end{array}$

Let $\tilde{V} = V + I, \tilde{H}_1 = H_1 + I$ and $\tilde{H}_2 = H_2 + I$. Note that e is the neutral element for multiplication: $xe = ex = x$ for all graphs x.

We have: $\tilde{V} = \begin{array}{c|cc} & 1,2 & 12 \\ \hline 1,2 & e & v \\ 12 & 0 & e \end{array}$ and $\tilde{H}_1\tilde{H}_2 = \begin{array}{c|cc} & 1,2 & 12 \\ \hline 1,2 & e & 0 \\ 12 & h_1 + h_2 & e \end{array}$.

Now the partition matrices are:

E-Letters	3-Letters
$\tilde{V}\tilde{H}_1\tilde{H}_2$	$\tilde{H}_1\tilde{H}_2\tilde{V}$
$\begin{array}{c\|cc} & 1,2 & 12 \\ \hline 1,2 & e + vh_1 + vh_2 & v \\ 12 & h_1 + h_2 & e \end{array}$	$\begin{array}{c\|cc} & 1,2 & 12 \\ \hline 1,2 & e & v \\ 12 & h_1 + h_2 & e + h_1v + h_2v \end{array}$

Some letters in our partition matrix have width $n = 1$. The resulting matrix shows the prime factorisations, from which the actual letters may be derived.

10.8.3 Facts

Here follow some facts about the prime letter matrices. For general m we have:

$V_iV_j = V_jV_i$	The order of the placement of the vertical and
$H_iH_j = H_jH_i$	horizontal edges does not affect the partition transitions.
$V_i^2 = 0$	A loop is created.
$H_i^2 = 0$	An isolated vertex occurs.
$\prod_{i=1}^{m} H_i = 0$	All horizontal edges have been removed.

10.8.4 Semiring

In order to perform matrix multiplication the elements must be part of a semiring. A semiring R is a set R equipped with two operations: multiplication and addition.

We will formally define the semiring, but warn the reader that it is not very insightful. This is because the set R of the semiring on which the operations are defined is quite complex. Namely it is the set of all sets of subgraphs of all $m \times n$ grids with infinitely many multi-edges between any two vertical nodes, where m is fixed and $n \geq 1$. Multiplication and addition will be defined on these sets of subgraphs.

Addition
$(R, +)$ is a commutative monoid, where $+$ stands for set union, and with neutral element $\emptyset$, which we will denote by 0.

Multiplication
The multiplication of two subgraphs is defined as pasting one to the other in the logical order. For $(R, \cdot)$ we define: $XY = \{xy \mid x \in X, y \in Y\}$ for all $X, Y \in R$.

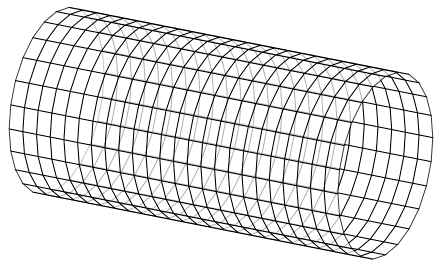
Distributive laws
The following laws are satisfied:
$a(b + c) = ab + ac$
$(a + b)c = ac + bc$

Multiplication by 0
We define $x0 = 0x = 0$ for all $x \in R$.

10.8.5 Partition matrix

The way the partition matrix is computed is similar to the count matrix. For the prime letters: $v_1, \dots, v_{m-1}, h_1, \dots, h_m$, let $V_1, \dots, V_{m-1}, H_1, \dots, H_m$ be their corresponding prime letter matrices. Let $\tilde{V}_i = V_i + I$ and $\tilde{H}_i = H_i + I$ for all i. Let $\tilde{V} = \prod_{i=1}^{m-1} \tilde{V}_i$ and $\tilde{H} = \prod_{i=1}^{m} \tilde{H}_i$. Then the partition matrix, with the letters being represented by their corresponding prime factorisation, for E-letters is $\tilde{V}\tilde{H}$ and for 3-letters it is $\tilde{H}\tilde{V}$.

10.8.6 Cylindrical lattice



Compared to the m-lattice the cylindrical lattice has an extra vertical edge between the top and bottom nodes. The partition matrix of the m-lattice can easily be derived from the partition matrix of the cylindrical lattice, namely by removing all letters that contain the (the prime corresponding to the) added edge. The advantage of computing the partition matrix for the cylindrical lattice is that this graph is more symmetric.

For example, suppose we know all partition transitions starting from partition $012, 3, \ldots, n-1$.[21] It corresponds to a row in the partition matrix. Then the transitions starting from $0, 123, \ldots, n-1$ or $0(n-2)(n-1), 2, 3, \ldots$ and so on can be derived from this row by translating the indexes of the primes. There is a rotational symmetry between these partitions. We can can also sense that $013, 2, 4, \ldots, n-1$ and $023, 1, 4, \ldots, n-1$ are symmetrical. Indeed we have: $i \to -i+3 \bmod n$. It is a combination of reflection, where each node i inside a partition is send to $-i \bmod n$, and a rotational symmetry.

This way the partitions may be grouped to form equivalence classes, with equivalence under rotation and reflection, and for each class we may choose a representative element. The partition matrix can be derived from the rows corresponding to the representative elements. Therefore only the rows of the representative elements have to be computed and stored after each prime letter matrix multiplication step. The compression factor is yet to be determined.

10.8.7 Relation to APSP

There are some similarities between our approach and a matrix implementation of the All Pairs Shortest Paths problem (APSP). In APSP the objective is to find the shortest path between all pairs of nodes. The algorithm is as follows.

Given a weighted graph G with weights in $\mathbb{R}_{\geq 0}$ and let A be the corresponding

[21]Here we labeled the nodes from 0 to $n-1$, to allow for a simple notation for symmetries. Note that the set of labels is now invariant under mod n

matrix. Let element-wise multiplication in A stand for the addition of the edge weights and let addition stand for taking the minimum. Now the identity matrix is added to the matrix A.[22] Finally $(A + I)^k$ can be computed to obtain shortest paths of length k or less between all pairs of nodes.

In this algorithm too it is not only the length of the shortest path that is of interest, but also the shortest path itself. The shortest paths can be obtained by a simple modification of A, by letting the elements be tuples of the shortest path length and the corresponding path.

10.8.8 Limitations

There are repetitive graphs, such as the triangular lattice, for which the presented approach to compute the partition matrix does not work. This is because letters can often not be decomposed into simple primes.

10.9 Implementation

We have implemented the algorithm for the computation of the partition graph of the m-lattice in a computer program and have computed the Parry matrix for the letter-partition graph with nodes corresponding to a pair (letter, end partition) and the natural edges between them. The code can be found here in [11].

10.10 Concluding remarks

There are fast matrix multiplication algorithms for dense matrices; generally they depend on an inverse for the addition operation. For our program these do not apply since there is no inverse for the set union.

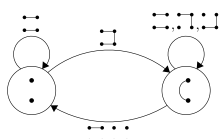
Also the prime letter matrices are sparse, so one may consider using a matrix representation that ignores zeroes.

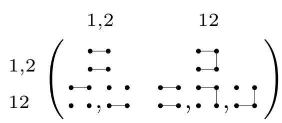
[22] The neutral element for multiplication is 0, for addition it is infinity and the addition operator in $A + I$ stands for taking the minimum.

11 Recursion formulas

Molina and Desjarlais found a way to find to express the number of spanning trees T_n of a $m \times n$ lattice in terms of smaller n. They did so by dividing the lattice into letters, the first letter being a vertical line followed by 3-letters. Later Paul Raff [9] extended their work.

$m = 2$
For $m = 2$ the method is as follows:
At each time step we increase the width of the lattice, denoted by n, by 1.
Let T_n be the number of spanning trees at time n.
Let F_n be the number of disconnected special forests at time n.





$$A = \begin{array}{c} \\ F_n \\ T_n \end{array} \begin{array}{cc} F_{n+1} & T_{n+1} \\ \left(\begin{array}{c} 1 \\ 2 \end{array} \right. & \left. \begin{array}{c} 1 \\ 3 \end{array} \right) \end{array}$$

We find: $F_{n+1} = F_n + 2T_n$ with $F_1 = 1$ (The number of unconnected special
$T_{n+1} = F_n + 3T_n$ with $T_1 = 1$ forests of is 1)

Let $v_n = \begin{bmatrix} F_n \\ T_n \end{bmatrix}$.

Then $v_{n+1}^{\intercal} = v_n^{\intercal} A$ with starting conditions $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

The characteristic polynomial of A is $\chi_\lambda(A) = \det(\lambda I - A) = \lambda^2 - 4\lambda + 1$.

By the Cayley-Hamilton Theorem we have $A^2 - 4A + I = 0$.

This can be rewritten to $A^2 = 4A - I$.

By multiplying both sides by $v_n^{\intercal}$ on the left we obtain $\begin{bmatrix} F_{n+2} \\ T_{n+2} \end{bmatrix} = 4 \begin{bmatrix} F_{n+1} \\ T_{n+1} \end{bmatrix} - \begin{bmatrix} F_n \\ T_n \end{bmatrix}$.

Hence we find a recursion for the number of spanning trees: $T_{n+2} = 4T_{n+1} - T_n$ with $T_1 = 1$ and $T_2 = 4$.

$m = 3$
Note that a similar approach works for $m = 3$:
Let T_n be the number of spanning trees at time n.
Let $F_n^{1,2,3}$ be the number of special forests at time n with rhs partition 1,2,3.
$F_n^{1,23}$ 1,23.
⋮ $F_n^{13,2}$ ⋮ 13,2.
$F_n^{12,3}$ 12,3.
Let F_n^{123} 123.

We have:

$$A = \begin{array}{c} \\ F_n^{1,2,3} \\ F_n^{1,23} \\ F_n^{13,2} \\ F_n^{12,3} \\ T_n \end{array} \begin{array}{c} \begin{array}{ccccc} F_{n+1}^{1,2,3} & F_{n+1}^{1,23} & F_{n+1}^{13,2} & F_{n+1}^{12,3} & T_{n+1} \end{array} \\ \left(\begin{array}{ccccc} 1 & 1 & 0 & 1 & 1 \\ 2 & 3 & 0 & 2 & 3 \\ 2 & 2 & 1 & 2 & 4 \\ 2 & 2 & 0 & 3 & 3 \\ 3 & 4 & 1 & 4 & 8 \end{array} \right) \end{array}$$

We find:

$$\begin{aligned} F_{n+1}^{1,2,3} &= F_n^{1,2,3} + 2F_n^{1,23} + 2F_n^{13,2} + 2F_n^{12,3} + 3T_n & \text{with } F_1^{1,2,3} &= 1 \\ F_{n+1}^{1,23} &= F_n^{1,2,3} + 3F_n^{1,23} + 2F_n^{13,2} + 2F_n^{12,3} + 4T_n & \text{with } F_1^{1,23} &= 1 \\ F_{n+1}^{13,2} &= F_n^{13,2} + T_n & \text{with } F_1^{13,2} &= 0 \\ F_{n+1}^{12,3} &= F_n^{1,2,3} + 2F_n^{1,23} + 2F_n^{13,2} + 3F_n^{12,3} + 4T_n & \text{with } F_1^{12,3} &= 1 \\ T_{n+1} &= F_n^{1,2,3} + 3F_n^{1,23} + 4F_n^{13,2} + 3F_n^{12,3} + 8T_n & \text{with } T_1 &= 1 \end{aligned}$$

Let $v_n = \begin{bmatrix} F_n^{1,2,3} \\ F_n^{1,23} \\ F_n^{13,2} \\ F_n^{12,3} \\ T_n \end{bmatrix}$. Then $v_{n+1}^{\intercal} = v_n^{\intercal} A$ with starting conditions $v_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$.

The characteristic polynomial of A is

$$\begin{aligned} \chi_\lambda(A) = \det(\lambda I - A) &= \lambda^5 - 16\lambda^4 + 47\lambda^3 - 47\lambda^2 + 16\lambda - 1 \\ &= (\lambda - 1)(\lambda^4 - 15\lambda^3 + 32\lambda^2 - 15\lambda + 1). \end{aligned}$$

By the Cayley-Hamilton Theorem and since $A \neq I$ we have $A^4 - 15A^3 + 32A^2 - 15A + I = 0$.

This can be rewritten to $A^4 = 15A^3 - 32A^2 + 15A - I$.

By multiplying both sides by $v_n^{\intercal}$ on the left we obtain a recursion for the number of spanning trees: $T_{n+4} = 15T_{n+3} - 32T_{n+2} + 15T_{n+1} - T_n$ with $T_1 = 1, T_2 = 15, T_3 = 192$ and $T_4 = 2415$.

Remark

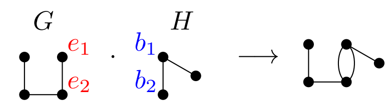
The characteristic polynomials of the count matrices of E-letters and 3-letters are equal for $m \leq 3$ and maybe also for larger m. If so, then there should be a combinatorial argument for deriving the recurrence relations using E-letters, but we do not know about it.

12 Generalisation to repetitive graphs

Possibly the reader has noticed that the ideas involving spanning trees on lattice graphs we have seen so far can be applied to any graph that has some kind of repetitive structure. We will define what kind of repetitivity the graph should have, and derive recurrence relations for some other graphs like the $2 \times n$ triangular lattice.

12.1 Pasting operation

We define the pasting operation $\cdot$ on two connected graphs, G, H. The graph G on the left hand side must have marked 'end' nodes and the graph H on the right must have as many 'begin' nodes. The new graph $G \cdot H$ is formed by identifying each end node with a unique begin node. For example:

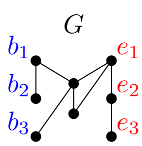


As seems natural the begin and end nodes with the same index are contracted. The labels should be chosen accordingly. Every label may occur only once, i.e. we don not allow one to identify more than two vertices. This is to keep the operation simple and because it is most intuitive.

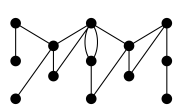
It is not yet clear why we use different terms: begin node and end node. This will show to be useful in the following section.

12.2 Pasting the same graph

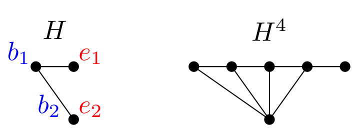
Many actively studied graphs satisfy the property that they can be formed by repeatedly pasting one and the same graph after another. To write this down formally, we can use the notation we just introduced. However, this time we define a graph with both begin and end nodes. For example:



By simply ignoring the superfluous labeling we can apply the pasting operation described above. It yields the following graph.

$G^2 =$ 

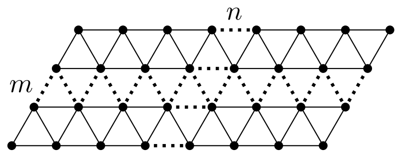
We can allow for a begin and an end node to coincide. For example:



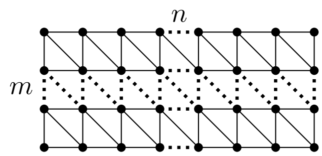
A graph is called *repetitive* if it is formed by a first graph with labeled end nodes, and pasting to it $n-1$ times the same second graph with as many labeled begin and end nodes.

13 Triangular Lattice

The $m \times n$ triangular lattice is:

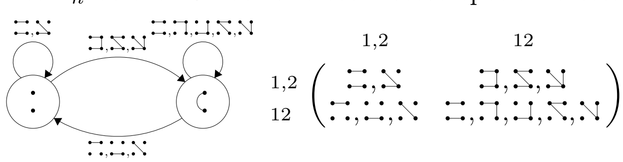


It has the following equivalent representation:



For this graph we can also determine the partition graph. The possible partitions that can occur in a spanning tree on the triangular lattice are again all of the non-crossing ones. Using the same method as before we can derive the recurrence relation for the number of spanning trees. We will do so for $m = 2$.

Recurrence relation for $m = 2$
Let T_n be the number of spanning trees at time n.
Let F_n be the number of disconnected special forests at time n.



$$A = \begin{array}{c} \\ F_n \\ T_n \end{array} \begin{array}{cc} F_{n+1} & T_{n+1} \\ \left(\begin{array}{c} 2 \\ 3 \end{array}\right. & \left.\begin{array}{c} 3 \\ 5 \end{array}\right) \end{array}$$

We find: $F_{n+1} = 2F_n + 3T_n$ with $F_1 = 1$
$T_{n+1} = 3F_n + 5T_n$ with $T_1 = 1$

Let $v_n = \begin{bmatrix} F_n \\ T_n \end{bmatrix}$. Then $v_{n+1}^\intercal = v_n^\intercal A$ with starting conditions $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

The characteristic polynomial of A is $\chi_\lambda(A) = \det(\lambda I - A) = \lambda^2 - 7\lambda + 1$.

By the Cayley-Hamilton Theorem we have $A^2 - 7A + I = 0$.

This can be rewritten to $A^2 = 7A - I$.

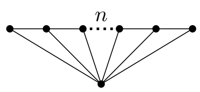
By multiplying both sides by $v_n^\intercal$ on the left we obtain $\begin{bmatrix} F_{n+2} \\ T_{n+2} \end{bmatrix} = 7 \begin{bmatrix} F_{n+1} \\ T_{n+1} \end{bmatrix} - \begin{bmatrix} F_n \\ T_n \end{bmatrix}$.

Hence we find a recursion for the number of spanning trees: $T_{n+2} = 7T_{n+1} - T_n$ with $T_1 = 1$ and $T_2 = 8$.

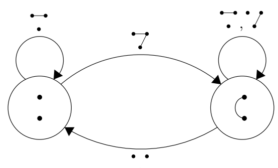
Uniform Spanning Tree
The uniform spanning tree can be characterised analogous to the m-lattice.

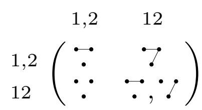
14 Fan graph

The fan of length n is:



Recurrence relation
Let T_n be the number of spanning trees at time n.
Let F_n be the number of disconnected special forests at time n.





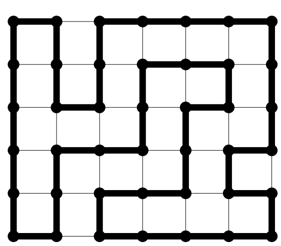
$$A = \begin{array}{c} \\ F_n \\ T_n \end{array} \begin{array}{c} \begin{array}{cc} F_{n+1} & T_{n+1} \end{array} \\ \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \end{array}$$

We find: $F_{n+1} = F_n + T_n$ with $F_1 = 1$
$T_{n+1} = F_n + 2T_n$ with $T_1 = 1$

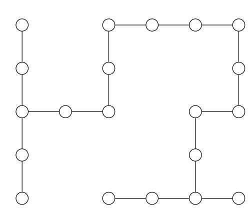
$\Rightarrow F_{n+1} - T_{n+1} = -T_n$
$\Rightarrow F_n = T_n - T_{n-1}$
$\Rightarrow T_{n+1} = 3T_n - T_{n-1}$ with $T_1 = 1$ and $T_2 = 3$

15 Hamilton cycles on the lattice

A hamilton cycle is a closed path that visits every vertex once. Hamilton cycles on the lattice graph were studied in [10]. In particular they obtained recurrence relations using a slightly more complicated concept of special forest. Hamilton cycles on the lattice are related to a special kind of tree, which is illustrated below.



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



Hamilton cycle, matrix notation, tree.

There is a different notion of a partition since the tree is different from a spanning tree.
In the above example the sequence of partitions is as follows:
12345 → 3 → 123,5 → 1,5 → 1,345 → 1235

Letters
The letters are matrix columns of ones and zeros. It turns out useful for the definition of possible partition-letter transitions to surround the whole matrix, except the right hand side, with zeros.

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

This means the first letter consists of only zeros. The other letters have a zero added to it at the bottom and at the top.

Partition-letter transitions
Now we must find rules for which partition-letter transitions are possible. They are:

1. No loops are created.
2. Every component has a path to the right.
3. There is no $\begin{smallmatrix}0&0\\0&0\end{smallmatrix}$, $\begin{smallmatrix}1&1\\1&1\end{smallmatrix}$ or $\begin{smallmatrix}1&0\\0&1\end{smallmatrix}$, $\begin{smallmatrix}0&1\\1&0\end{smallmatrix}$, where the letter is extended with a zero at the top and at the bottom.

Special forests
A matrix filled with ones and zeros is a special forest if every column satisfies the partition-letter transition rules.

Number of Hamilton cycles
The number of Hamilton cycles is equal to the number of special forests such that the final partition consists of one component and such that the third transition rule is satisfied when the extended all zeros letter (with an extra zero at the top and bottom) is added to the right most end.

Recurrence relation for $m = 4$
We will derive a recurrence relation for the number of hamilton cycles on the $4 \times n$ lattice.

The Partition matrix:

	–	1	2	3	12	23	13	1,3	123
–								1 0 1	1 1 1
1								1 0 1	1 1 1
2									1 1 1
3								1 0 1	1 1 1
12						0 1 1			
23					1 1 0				
13		1 0 0		0 0 1			1 0 1		
1,3								1 0 1	1 1 1
13		1 0 0	0 1 0	0 0 1			1 0 1		

The count matrix is:

$$
\begin{array}{c}
\\
\\
F_n^{-}\\ F_n^{1}\\ F_n^{2}\\ F_n^{3}\\ F_n^{12}\\ F_n^{23}\\ F_n^{13}\\ F_n^{1,3}\\ F_n^{123}
\end{array}
\begin{array}{c}
A \\
\begin{array}{ccccccccc}
F_{n+1}^{-} & F_{n+1}^{1} & F_{n+1}^{2} & F_{n+1}^{3} & F_{n+1}^{12} & F_{n+1}^{23} & F_{n+1}^{13} & F_{n+1}^{1,3} & F_{n+1}^{123}
\end{array}\\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\end{array}
\quad
\begin{array}{c}
v_n\\
\begin{bmatrix} F_n^{-}\\ F_n^{1}\\ F_n^{2}\\ F_n^{3}\\ F_n^{12}\\ F_n^{23}\\ F_n^{13}\\ F_n^{1,3}\\ F_n^{123} \end{bmatrix}
\end{array}
\quad
\begin{array}{c}
v_1\\
\begin{bmatrix} 1\\0\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix}
\end{array}
$$

Then $v_{n+1}^{\intercal} = v_n^{\intercal} A$ with starting conditions v_1.

The characteristic polynomial of A is
$\chi_\lambda(A) = \det(\lambda I - A) = \lambda^3(\lambda - 1)(\lambda + 1)(\lambda^4 - 2\lambda^3 - 2\lambda^2 + 2\lambda - \lambda)$

By the Cayley-Hamilton Theorem we have $A^4 - 2A^3 - 2A^2 + 2A - I = 0$.

This can be rewritten to $A^4 = 2A^3 + 2A^2 - 2A + I$.

By multiplying both sides by $v_n^{\intercal}$ on the left we obtain:
$F_{n+4}^{13} = 2F_{n+3}^{13} + 2F_{n+2}^{13} - 2F_{n+1}^{13} + F_n^{13}$ with $F_1^{13} = 0, F_2^{13} = 0, F_3^{13} = 1, F_4^{13} = 2$.
$F_{n+4}^{123} = 2F_{n+3}^{123} + 2F_{n+2}^{123} - 2F_{n+1}^{123} + F_n^{123}$ with $F_1^{123} = 0, F_2^{123} = 1, F_3^{123} = 1, F_4^{123} = 4$.

The number of Hamilton cycles is $H_n = F_n^{13} + F_n^{123}$. It satisfies the same recurrence:

$$H_{n+4} = 2H_{n+3} + 2H_{n+2} - 2H_{n+1} + H_n \quad \text{with } H_1 = 0, H_2 = 1, H_3 = 2, H_4 = 6.$$

In the next section we cover a well known theorem on the enumeration of spanning trees of any irreducible graph.

16 Matrix tree theorem

16.1 Introduction

The matrix tree theorem is due to Kirchhoff (1847). We will give an outline of a more recent, standard proof. For the complete proof we refer to [13].

16.2 Preliminaries

We will often need to make a distinction between graphs and directed graphs. When a graph is directed we will refer to it as a digraph. Let $G = (\mathcal{V}, \mathcal{E})$ be a (di)graph with n vertices and m edges.

Let $\mathcal{A}$ be the *adjacency matrix* of G and let $\mathcal{D}$ be the matrix with the degrees of the vertices on the diagonal. The *Laplacian* of G is defined as follows: $\mathcal{L} = \mathcal{D} - \mathcal{A}$

The *incidence matrix* $\mathcal{B}$ of a graph is an $(n \times m)$ matrix with:

$$b_{ij} = \begin{cases} 1 & \text{vertex } i \text{ is an end point of edge } j \\ 0 & \text{otherwise} \end{cases}$$

The *incidence matrix* $\mathcal{B}$ of a digraph is an $(n \times m)$ matrix with:

$$b_{ij} = \begin{cases} 1 & \text{edge } j \text{ enters vertex } i \\ -1 & \text{edge } j \text{ leaves vertex } i \\ 0 & \text{otherwise} \end{cases}$$

The *oriented incidence matrix* $\mathcal{B}$ of a graph is the same as the incidence matrix, except that the edges are given an arbitrary orientation.

16.3 Proof

We first present a couple of lemmas.

Lemma 16.1. *Let $\mathcal{B}$ be the oriented incidence matrix of a simple undirected G with arbitrary orientation. Then $\mathcal{L} = \mathcal{B} \cdot \mathcal{B}^\intercal$.*

Lemma 16.2. *Let G be a graph with n vertices and let $\mathcal{B}$ denote its incidence matrix. Then G is connected $\iff$ rank$(\mathcal{B}) = n - 1$.*

Definition 16.1. An $r \times s$ matrix M is called totally unimodular (TUM) if every $k \times k$ submatrix has determinant equal to $0, 1$ or -1 with $1 \leq k \leq \min\{r, s\}$.

It follows from the definition that the entries of a TUM matrix are $1, -1$ or 0.

Lemma 16.3. *Suppose M is a matrix with all entries $1, -1$ or 0 for which each column (row) contains at most one 1 and one -1. Then M is TUM.*

Consequently $\mathcal{B}$ is TUM.

Now follows the connection between $\mathcal{B}$ and spanning trees.

Lemma 16.4. *Let M be a $n-1 \times n-1$ submatrix of $\mathcal{B}$. Then M is invertible $\iff$ the subgraph of G induced by the edges (columns) in M is a spanning tree.*

Proof. Let H be the subgraph of G determined by M. Note that M contains $n-1$ columns which represent edges, so H consists of $n-1$ edges. Therefore H is a spanning tree $\iff$ H is connected. By lemma 3.1 we have: H is connected $\iff$ $\text{rank}(M) = n-1$. Note that $\text{rank}(M) = n-1 \iff M$ is invertible, since M is a $n-1 \times n-1$ matrix. Hence M is invertible $\iff$ H is a spanning tree. $\square$

Theorem 16.5 (Cauchy-Binet)**.** *Let A, B be a $m \times n$ matrix and $n \times m$ matrix respectively, where $1 \leq m \leq n$. Then:*

$$\det(AB) = \sum_{S \subset \{1,\ldots,n\}, |S|=m} \det(A_{mS}) \det(B_{mS})$$

where A_{mS} denotes the matrix obtained from A by only taking those columns that occur in S and analogously B_{mS} denotes the matrix obtained by only taking the rows that occur in S.

Theorem 16.6 (Matrix Tree Theorem)**.** *Let $G = (\mathcal{V}, \mathcal{E})$ be a connected graph with n vertices and let $v \in \mathcal{V}$ arbitrary. Then the number of spanning trees of G is given by:*

$$t(G) = \det(\mathcal{L}^{\{v\}})$$

where $\mathcal{L}^{\Delta}$ with $\Delta \subset \mathcal{V}$ is the matrix obtained from $\mathcal{L}$ by removing the rows and columns corresponding to the vertices in Δ.

Proof. Let $\mathcal{B}^{\{n\}}$ the submatrix of $\mathcal{B}$ obtained by removing row n, which corresponds to vertex n. Then:

$$\begin{aligned}
\det(\mathcal{L}^{\{n\}}) &= \det(\mathcal{B}^{\{n\}} \cdot (\mathcal{B}^{\{n\}})^{\intercal}) && \text{(similar to Lemma 2.1)} \\
&= \sum_{B, \text{the } n-1 \times n-1 \text{ submatrices of } \mathcal{B}^{\{n\}}} \det(B) \cdot \det(B^{\intercal}) && \text{(Cauchy-Binet)} \\
&= \sum_{B} \det(B)^2 \\
&= \sum_{B \text{ invertible}} \det(B)^2 + \sum_{B \text{ non-invertible}} \det(B)^2 \\
&= \sum_{B \text{ invertible}} \det(B)^2 \\
&= \sum_{B \text{ invertible}} 1 && \text{(Lemma 3.2)} \\
&= \text{number of invertible } n-1 \times n-1 \text{ submatrices of } \mathcal{B}^{\{n\}} \\
&= \text{t(G)} && \text{(Lemma 3.3)}
\end{aligned}$$

This completes the proof. $\square$

Alternative way to apply the matrix tree theorem

Remark 16.7. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with n vertices. Define G' by adding a vertex w and the edge (v_1, w) to G. Note that the number of spanning trees does not change. We have:

$$\mathcal{L}_{G'} = \begin{array}{c} \\ w \\ v_1 \\ \\ v_n \end{array} \begin{array}{c} \begin{array}{cccc} w & v_1 & & v_n \end{array} \\ \begin{pmatrix} 1 & -1 & \dots & 0 \\ -1 & \deg(v_1)+1 & & \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & \deg(v_n) \end{pmatrix} \end{array}$$

Hence $t(G) = t(G') = \det(\mathcal{L}_{G'}^{\{w\}})$

16.4 Matrix tree theorem for weighted directed graphs

The matrix tree theorem can be generalised to weighted digraphs. Let G be a digraph. Let us write w_{ij} for the edge weights from vertex i to j, and alternatively, if the two incident vertices are unknown, we will denote the weights by $W(e)$. Define the Laplacian as follows: $\mathcal{L}_{ij} = \begin{cases} \sum_{k \neq i} w_{ki} & \text{if } i = j \\ -w_{ij} & \text{otherwise} \end{cases}$

Let $\mathcal{T}$ be the collection of spanning ditrees oriented into the root, called *anti-arborescences*, in G, and for $t \in \mathcal{T}$ we define its weight as: $W(t) = \prod_{e \in t} W(e)$. In particular, if all edge weights are natural numbers, the weight of a spanning tree could be viewed as the number of trees in G of similar form. Define $\mathcal{T}_j$ to be the spanning trees rooted at j. Then the following theorem on the sum of the weighted anti-arborescences rooted at j is due to Tutte.

Theorem 16.8 (Tutte's Matrix Tree Theorem)**.** *Let $G = (\mathcal{V}, \mathcal{E}, W)$ be a weighted digraph. Then the number of weighted spanning trees is given by:*

$$\sum_{t \in \mathcal{T}_j} W(t) = \det(\mathcal{L}^{\{j\}})$$

See [8] for a proof. For an even further generalisation, see [7].

Remark 16.9. Consider $\mathcal{L}'$ equal to the laplacian $\mathcal{L}$ for directed graphs, but for each diagonal element we take the row sum instead of the column sum. Observe that $\det(\mathcal{L}'^{\intercal}) = \det(\mathcal{L}')$. Hence $\det(\mathcal{L}')$ is the number of weighted anti-arborescences in the socalled transpose graph of G, the graph in which the edge directions are reversed. Thus $\det(\mathcal{L}')$ equals the weighted number of arborescences in the original graph. To summarise, one can compute the total weighted anti-arborescences or arborescences by taking the determinant of the laplacian or its counterpart respectively.

Further reading

An article by Michael Kozdron [5] provides a proof of the matrix tree theorem simplifying Lawler's [4] proof. It uses the theory of random walks. During the process they rediscovered a result on stationary distribution of aperiodic Markov Chains, see 18.1.

17 Markov chain tree theorem

We will assume the reader has basic knowledge of Markov chain theory. From now on, by a Markov chain we mean a finite, time-homogenous Markov chain.

Recall that the stationary probability π_{ij} is the fraction of time that the Markov chain starting from i spends at vertex j in the long run.

Theorem 17.1 (Markov Chain Tree Theorem)**.** *Let M be an irreducible aperiodic Markov chain. Then the stationary distribution is given by:*

$$\pi_j = \frac{\sum_{t \in \mathcal{T}_j} W(t)}{\sum_{t \in \mathcal{T}} W(t)}$$

For a proof, see [2].

18 Stationary distribution of a Markov chain

There is a simple formula for the stationary distribution of a Markov chain.

Corollary 18.1 (Richards-Stroock Theorem)**.** *The stationary distribution of irreducible aperiodic Markov chain M can be written as:*

$$\pi_j = \frac{\det(\mathcal{L}^{\{j\}})}{\sum_{i=1}^{n} \det(\mathcal{L}^{\{i\}})}$$

where $\mathcal{L} = I - P$.

It immediately follows from the matrix tree theorem and the Markov chain tree theorem.

19 Further research

We studied the uniform spanning tree on the bi-infinite lattice, inspired by Häggström's paper for $m = 2$. Häggström considered his main result to be the relation between the uniform spanning tree and a Subshift of Finite Type.

Definition 19.1. Let $G = (V, E)$ be a connected, directed graph without multi-edges. Let Y be the power set of bi-infinite paths, i.e. sequences of edges, over G. Let T be the left shift-operator on these sequences, defined by $T : (a_k)_{k \in \mathbb{Z}} \mapsto (a_{k+1})_{k \in \mathbb{Z}}$. A Nearest Neighbour Subshift of Finite Type is then defined as the pair (Y, T).

The set Y is T-invariant. Häggström observed that the bi-infinite walks on the graph corresponding to the adjacency matrix in 3 or 4 induce a Subshift of finite type.

However, Häggström provided only one application of the subshift of finite type, which involved the Parry measure. This is why we did not introduce the reader to the topic of SFT's. However, if the reader is familiar with the subject or has become interested in SFT's, one is free to explore them in the context of infinite spanning trees. The bi-infinite paths on the start-partition-letter graph, or on the letter-end-partition graph, together with the left shift T, are a subshift of finite type. Since the graphs are aperiodic, the Parry measure on these SFT's is well-defined.

20 Conclusion

My thanks go to my supervisor Evgeny Verbitskiy and my father, André van Delft, for their valuable advice and patience.

References

[1] Perron-frobenius theory, 2008-09. URL https://stanford.edu/class/ee363/lectures/pf.pdf.

[2] V. Anantharam and P. Tsoucas. A proof of the markov chain tree theorem. *Statistics & Probability Letters*, 8(2):189 – 192, 1989. ISSN 0167-7152. URL http://www.sciencedirect.com/science/article/pii/0167715289900163.

[3] O. Häggström. Uniform Spanning Trees for the Infinite Ladder. September 1993.

[4] Harry Kesten, Maury Bramson, and Richard Durrett. *Perplexing problems in probability: festschrift in honor of Harry Kesten.* Birkhäuser Verlag, Basel; Cambridge, MA, 1999.

[5] Michael J. Kozdron. A random walk proof of kirchhoffs matrix tree theorem. 2015. URL https://www.newton.ac.uk/files/seminar/20150617113012301-395824.pdf.

[6] Jean Mairesse and Irène Marcovici. Uniform sampling of subshifts of finite type on grids and trees. *Int. J. Found. Comput. Sci.*, 28(3):263–287, 2017. ISSN 0129-0541; 1793-6373/e.

[7] J.W. Moon. Some determinant expansions and the matrix-tree theorem. *Discrete Mathematics*, 124(1):163 – 171, 1994. ISSN 0012-365X. URL http://www.sciencedirect.com/science/article/pii/0012365X9200059Z.

[8] James B Orlin. Line-digraphs, arborescences, and theorems of tutte and knuth. *Journal of Combinatorial Theory, Series B*, 25(2):187 – 198, 1978. ISSN 0095-8956. URL http://www.sciencedirect.com/science/article/pii/0095895678900382.

[9] P. Raff. Spanning Trees in Grid Graphs. *ArXiv e-prints*, September 2008.

[10] Robert Stoyan and Volker Strehl. Enumeration of Hamiltonian circuits in rectangular grids. *Sémin. Lothar. Comb.*, 34:21, 1995. ISSN 1286-4889/e.

[11] André van Delft and Max van Delft. Partition matrix computation, may 2017. URL https://github.com/MaxvanDelft/Spanning-Tree.

[12] Max van Delft. Parry matrix and perron frobenius theory. january 2019.

[13] Janneke van den Boomen. The matrix tree theorem. 2007. URL http://www.math.kun.nl/~bosma/Students/jannekebc3.pdf.

[14] Wikipedia. Noncrossing partition. URL https://en.wikipedia.org/wiki/Noncrossing_partition.