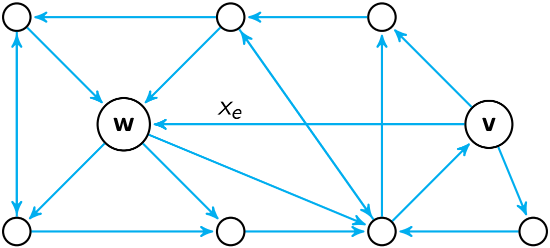
Spanning trees of tree graphs

Philippe Biane, CNRS-IGM-Université Paris-Est

Firenze, May 18 2015

joint work with Guillaume Chapuy, CNRS-LIAFA-Université Paris 7

V, E=directed graph



Q=Laplacian matrix, indexed by $V \times V$

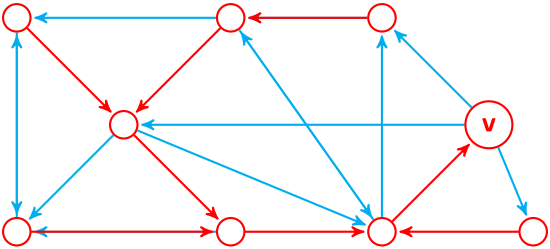
- $Q_{vw} = x_e$ if $e : v \to w$ is a directed edge of the graph
- $Q_{vv} = -\sum_w Q_{vw}$

Markov chains

If $x_e \geq 0$ this is the generator of a continuous time Markov chain on the graph, with transition probabilities e^{tQ}.

- $Q\mathbf{1} = 0$ where $\mathbf{1}$ is the constant vector.
- If the chain is *irreducible* or the graph is *strongly connected* the kernel is one dimensional (Perron Frobenius).
- $\mu Q = 0$ for a unique positive invariant measure μ.

Rooted spanning trees



A spanning tree rooted at v

Kirchhoff'theorem

$X \subset V$
$Q_X = Q$ with rows and columns in X

$$\det(Q_X) = \sum_{f \in F_X} \prod_{e \in f} x_e$$

The sum is over forests rooted in $V \setminus X$.

In particular the invariant measure is

$$\mu(v) = \sum_{t \in T_v} \prod_{e \in t} x_e$$

sum over oriented trees rooted at v

Tree-graph of a graph

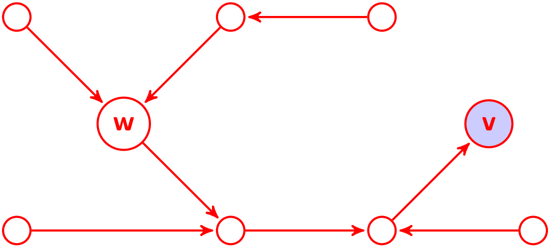
$$TG = (TV, TE)$$

TV=Vertices of the tree graph=spanning trees of the graph

s=spanning tree rooted at v

$e = v \to w$

edge from s to t:



The tree s

Tree-graph of a graph

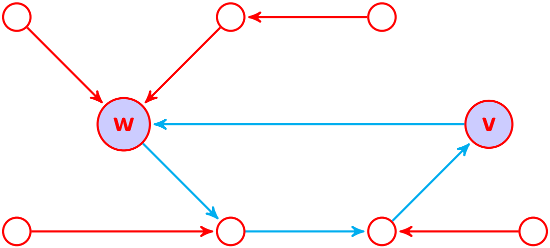
$$TG = (TV, TE)$$

TV=Vertices of the tree graph=spanning trees of the graph

s=spanning tree rooted at v

$e = v \to w$

edge from s to t:



Tree-graph of a graph

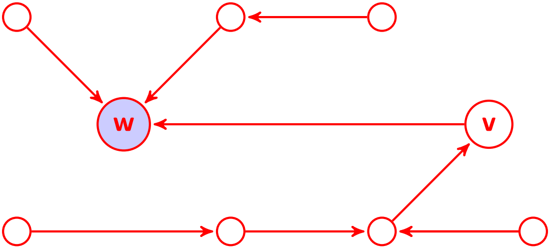
$$TG = (TV, TE)$$

TV=Vertices of the tree graph=spanning trees of the graph

s=spanning tree rooted at v

$e = v \to w$

edge from s to t:



The tree t

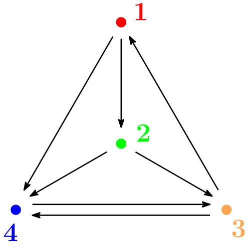
The tree-graph is a covering graph:

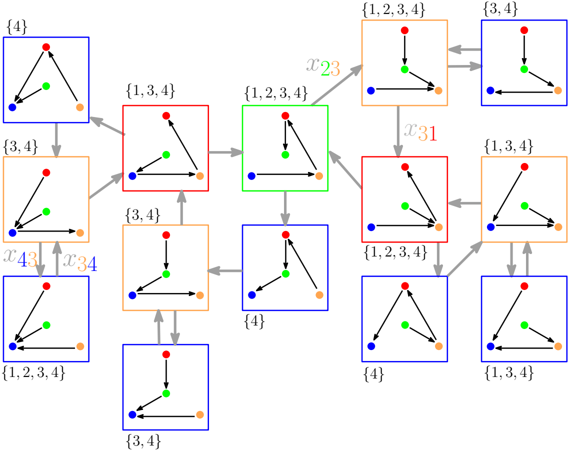
$$p : s \mapsto v$$

mapping each tree to its root.

Every path in V can be lifted to TV.

Example


1
2
3
4


{4}
{3, 4}
{1, 2, 3, 4}
{1, 3, 4}
{3, 4}
{3, 4}
{1, 2, 3, 4}
{4}
{1, 2, 3, 4}
{3, 4}
{1, 2, 3, 4}
{1, 3, 4}
{4}
{1, 3, 4}
x_{43}
x_{34}
x_{23}
x_{31}

Laplacian matrix of the tree-graph

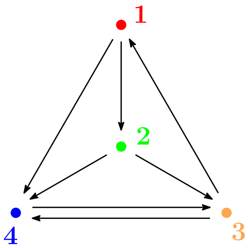
On each edge $s \to t$ above $v \to w$ put the weight x_e.
This defines the Laplacian matrix R of the tree-graph

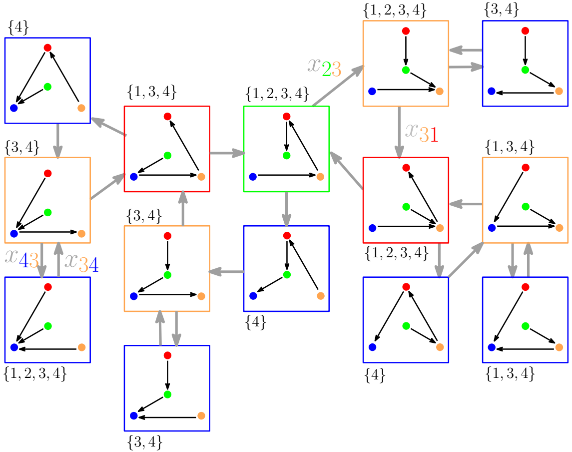
$$R_{st} = Q_{vw}; \qquad p(s) = v; p(t) = w$$

This is the generator of a continuous time Markov chain on the tree-graph.

Lifting of the Markov chain

The chain on TV projects to the chain on V by $p : TV \to V$: if TX is a R-Markov chain on TV then $p(TX)$ is a Q-Markov chain on V.





Lemma: the invariant measure of the chain on the tree graph is

$$T\mu(t) = \prod_{e \in t} x_e$$

This provides a combinatorial proof of Kirchhoff's theorem since

$$p(T\mu) = \mu$$

$$\mu(v) = \sum_{t \in p^{-1}(v)} T\mu(t) = \sum_{t \in T_v} \prod_{e \in t} x_e$$

Spanning trees of the tree-graph

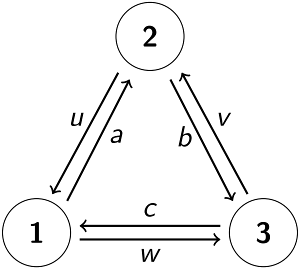
The invariant measure of the chain on the tree graph can also be computed using spanning trees of the tree graph.
The preceding result implies

$$\sum_{\mathbf{t}\in T_t}\prod_{\mathbf{e}\in\mathbf{t}} x_{\mathbf{e}} = P(x_e; e\in E)\prod_{e\in t} x_e$$

the sum is over spanning trees of TG rooted at t.
The polynomial P is independent of t,
it depends only on the graph V.

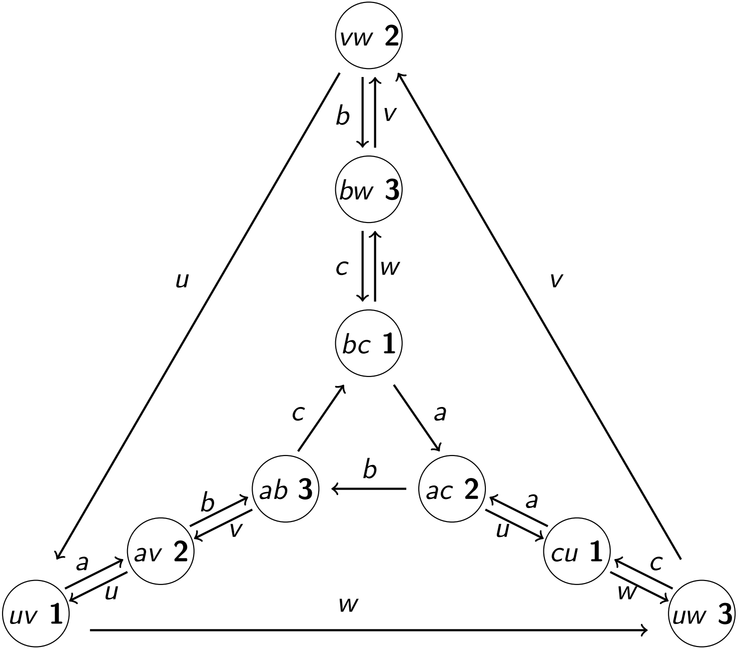
Example

The complete graph on $X = \{1, 2, 3\}$



$$Q = \begin{pmatrix} \lambda & a & w \\ u & \mu & b \\ c & v & \nu \end{pmatrix}$$

with $\lambda = -a - w$, $\mu = -b - u$, $\nu = -c - v$


vw 2
b
v
bw 3
c
w
bc 1
u
v
c
a
b
ab 3
ac 2
b
a
v
u
av 2
cu 1
a
c
u
w
uv 1
w
uw 3

The transition matrix for the lifted Markov chain is

$$R = \begin{pmatrix} \lambda & 0 & 0 & 0 & a & 0 & w & 0 & 0 \\ 0 & \lambda & 0 & a & 0 & 0 & w & 0 & 0 \\ 0 & 0 & \lambda & 0 & a & 0 & 0 & w & 0 \\ \\ 0 & u & 0 & \mu & 0 & 0 & 0 & 0 & b \\ u & 0 & 0 & 0 & \mu & 0 & 0 & 0 & b \\ 0 & u & 0 & 0 & 0 & \mu & 0 & b & 0 \\ \\ c & 0 & 0 & 0 & 0 & v & \nu & 0 & 0 \\ 0 & 0 & c & 0 & 0 & v & 0 & \nu & 0 \\ 0 & 0 & c & v & 0 & 0 & 0 & 0 & \nu \end{pmatrix}$$

The polynomial P can be computed

$$\begin{aligned} P(a,b,c,u,v,w) &= (bc+cu+uv)(av+ac+vw)(ab+bw+uw) \\ &= \prod_{i\in X}\left(\sum_{t\in T_i}\pi(t)\right) \end{aligned}$$

It is a product of the 2-minors of the matrix

$$Q = \begin{pmatrix} \lambda & a & w \\ u & \mu & b \\ c & v & \nu \end{pmatrix}$$

$$\lambda = -a-w, \quad \mu = -b-u, \quad \nu = -c-v$$

Theorem

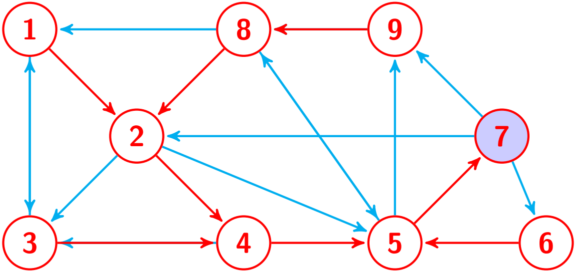
There exist integers $m(W); W \subset V$ such that

$$P(x_e; e \in E) = \prod_{W \subsetneq V} \det(Q_W)^{m(W)}$$

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

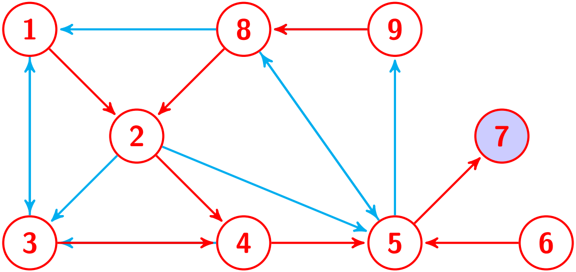


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

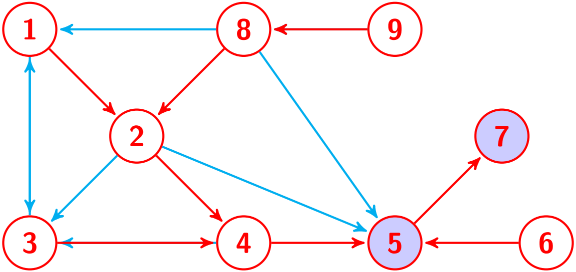


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

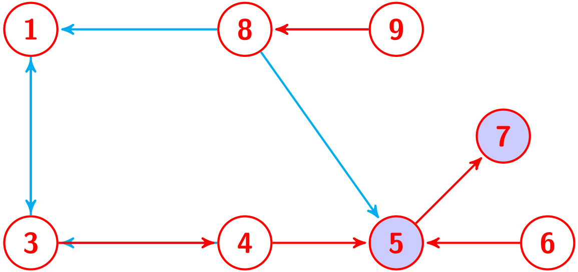


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

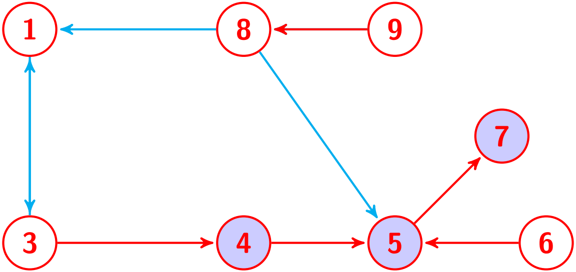


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

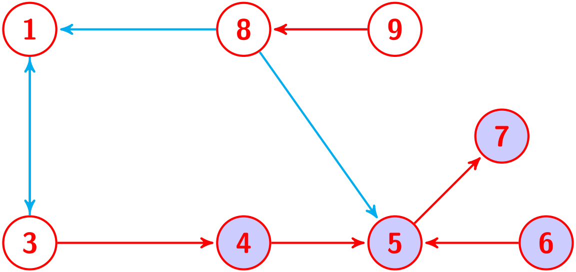


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

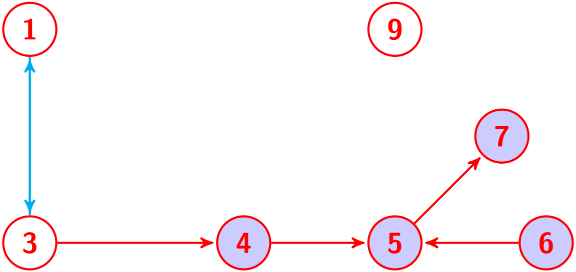


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

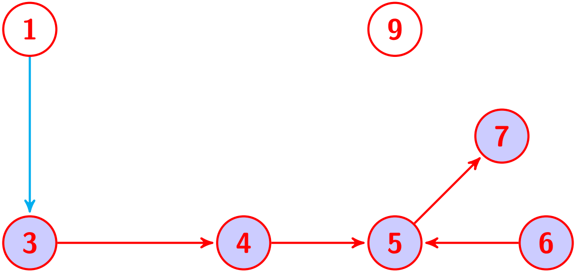


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

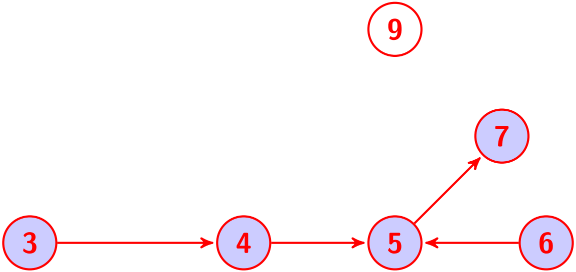


This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.



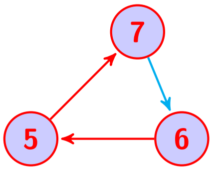
This yields a tree on vertex set X. The output is the strongly connected component of v in X.

Computation of the multiplicities $m(W)$

Fix a total ordering of the vertex set V of G.

Start with a vertex v, and a spanning tree t rooted at v. Perform *breadth first search* of the graph t and for each vertex obtained erase it if the edge is not in the tree t.

The set W



This yields a tree on vertex set X. The output is the strongly connected component of v in X.

The multiplicity $m(v, W)$ is equal to the number of spanning trees rooted at v such that the algorithm outputs W.

Proposition: For all $W \subset V$, the multiplicity $m(W) = m(v, W)$ does not depend on $v \in W$. Also it does not depend on the ordering of the vertices.

$m(W)$ is the multiplicity in the formula

$$P(x_e; e \in E) = \prod_{W \subsetneq V} \det(Q_W)^{m(W)}$$

Proof

The proof of the formula

$$P(x_e; e \in E) = \prod_{W \subsetneq V} \det(Q_W)^{m(W)}$$

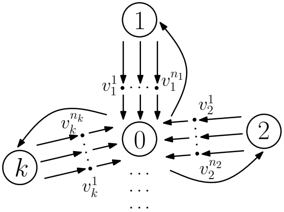
is algebraic, actually one has

$$\det(zI - TQ) = \prod_{W \subset V} \det(zI - Q_W)^{m(W)}$$

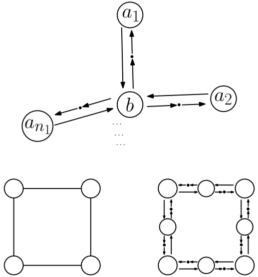
The proof of this formula consists in finding appropriate invariant subspaces for TQ.

Example

The bouquet graphs:



When $n_1 = \ldots = n_k = 1$ the tree graph of the bouquet graph is the hypercube $\{0,1\}^k$.



We recover then Stanley's formula, generalized by Bernardi:

Theorem

The generating function of spanning oriented forests of the hypercube $\{0,1\}^k$*, with a weight* z *per root and a weight* y_i^j *for each edge mutating the* i*-th coordinate to the value* j *is given by:*

$$\prod_{J\subset[1..k]}\left(z+\sum_{i\in J}(y_i^0+y_i^1)\right).$$