Greedoid polynomial, chip-firing, and G-parking function for directed graphs

Swee Hong Chan
Cornell University

Connections in Discrete Mathematics

June 15, 2015

Motivation

Tutte polynomial [Tut54] is a polynomial defined for undirected graphs that has applications to several branches of mathematics (graph coloring, knot theory, Ising model, etc.).

We are interested in a "generalization" of Tutte polynomial for directed graphs and its connections to other topics in discrete mathematics.

Known Tutte-like polynomial for directed graphs:

- Cover polynomial [CG95],
- Greedoid polynomial [BKL85].

In this talk, G is a directed graph that may have loops and multiple edges, and is strongly connected.

Motivation

Tutte polynomial [Tut54] is a polynomial defined for undirected graphs that has applications to several branches of mathematics (graph coloring, knot theory, Ising model, etc.).

We are interested in a "generalization" of Tutte polynomial for directed graphs and its connections to other topics in discrete mathematics.

Known Tutte-like polynomial for directed graphs:

- Cover polynomial [CG95],
- Greedoid polynomial [BKL85].

In this talk, G is a directed graph that may have loops and multiple edges, and is strongly connected.

Motivation

Tutte polynomial [Tut54] is a polynomial defined for undirected graphs that has applications to several branches of mathematics (graph coloring, knot theory, Ising model, etc.).

We are interested in a "generalization" of Tutte polynomial for directed graphs and its connections to other topics in discrete mathematics.

Known Tutte-like polynomial for directed graphs:

- Cover polynomial [CG95],
- Greedoid polynomial [BKL85].

In this talk, G is a directed graph that may have loops and multiple edges, and is strongly connected.

Motivation

Tutte polynomial [Tut54] is a polynomial defined for undirected graphs that has applications to several branches of mathematics (graph coloring, knot theory, Ising model, etc.).

We are interested in a "generalization" of Tutte polynomial for directed graphs and its connections to other topics in discrete mathematics.

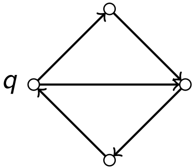
Known Tutte-like polynomial for directed graphs:

- Cover polynomial [CG95],
- Greedoid polynomial [BKL85].

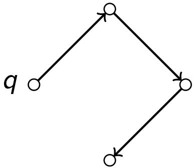
In this talk, G is a directed graph that may have loops and multiple edges, and is strongly connected.

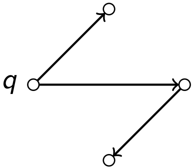
Rooted spanning tree

Choose a root vertex $q \in V(G)$.
In a **q-rooted spanning tree**, there is a unique directed path from q to v for each vertex $v \in V(G)$.



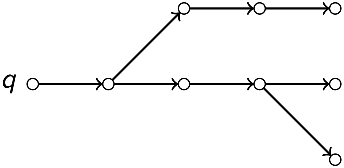
q-rooted spanning trees





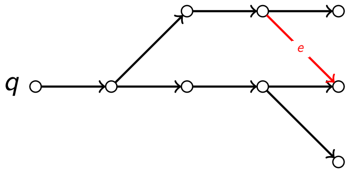
External activity

Fix a total order $<_e$ on $E(G)$.
Let T be a q-rooted spanning tree, and let $e \in E(G) \setminus E(T)$.
The graph $T \cup \{e\}$ has a unique circuit C.



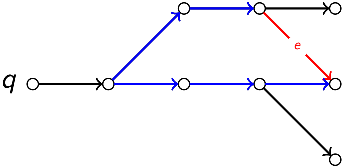
External activity

Fix a total order $<_e$ on $E(G)$.
Let T be a q-rooted spanning tree, and let $e \in E(G) \setminus E(T)$.
The graph $T \cup \{e\}$ has a unique circuit C.



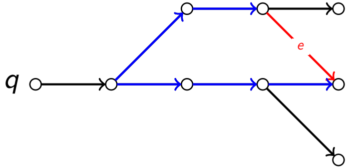
External activity

Fix a total order $<_e$ on $E(G)$.
Let T be a q-rooted spanning tree, and let $e \in E(G) \setminus E(T)$.
The graph $T \cup \{e\}$ has a unique circuit C.

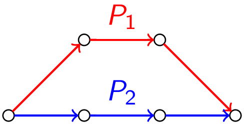


External activity

Fix a total order $<_e$ on $E(G)$.
Let T be a q-rooted spanning tree, and let $e \in E(G) \setminus E(T)$.
The graph $T \cup \{e\}$ has a unique circuit C.

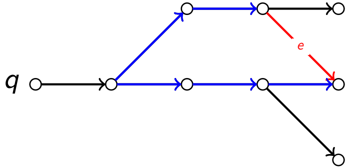


The circuit C decomposes to two directed paths P_1 and P_2. Let P_1 be the path that contains e.

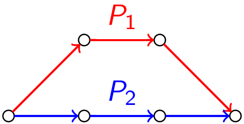


External activity

Fix a total order $<_e$ on $E(G)$.
Let T be a q-rooted spanning tree, and let $e \in E(G) \setminus E(T)$.
The graph $T \cup \{e\}$ has a unique circuit C.



The circuit C decomposes to two directed paths P_1 and P_2. Let P_1 be the path that contains e.



The edge e is **externally active w.r.t.T** if the smallest edge of $E(C)$ is contained in $E(P_1)$.

Greedoid polynomial

The **external activity** of T, denoted by $\text{ext}(T)$, is the number of edges that are externally active w.r.t T.
The **greedoid polynomial** of G w.r.t root vertex q is the polynomial

$$\lambda_q(G; y) = \sum_T y^{\text{ext}(T)}.$$

- The polynomial $\lambda_q(G; y)$ does not depend on the total ordering $<_e$.
- If G is undirected, then $\lambda_q(G; y) = y^{|\bar{E}(G)|} T(G; 1, y)$.

Greedoid polynomial

The **external activity** of T, denoted by $\text{ext}(T)$, is the number of edges that are externally active w.r.t T.
The **greedoid polynomial** of G w.r.t root vertex q is the polynomial

$$\lambda_q(G;y) = \sum_T y^{\text{ext}(T)}.$$

- The polynomial $\lambda_q(G;y)$ does not depend on the total ordering $<_e$.
- If G is undirected, then $\lambda_q(G;y) = y^{|\bar{E}(G)|} T(G;1,y)$.

Greedoid polynomial (ctd.)

Things that are different from Tutte polynomial:

- There is no notion of 'internal activity'.
- It satisfies a 'weak' deletion-contraction recurrence, i.e. if e is an edge rooted at q, then:

$$\lambda_q(G;y) = \begin{cases} y\lambda_q(G \setminus \{e\};y) & \text{if } e \text{ is a loop;} \\ \lambda_q(G \setminus \{e\};y) + \lambda_q(G/\{e\};y) & \text{otherwise.} \end{cases}$$

- The polynomial $\lambda_q(G;y)$ depends on the choice of root q.

G-parking function

A function $f : V(G) \setminus \{q\} \to \mathbb{N}$ is a **G-parking function** w.r.t. q [PS04] if for all nonempty $A \subseteq V(G) \setminus \{q\}$, there is $v \in A$ so that

$$f(v) < \text{number of edges from } V(G) \setminus A \text{ to } v.$$

Example.

The **level** of f, denoted by $\text{lvl}(f)$, is

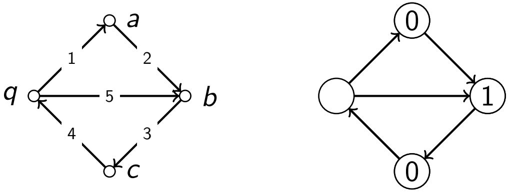
$$\text{lvl}(f) = |E(G)| - |V(G)| + 1 - \sum_{v \in V(G) \setminus \{q\}} f(v).$$

G-parking function

A function $f : V(G) \setminus \{q\} \to \mathbb{N}$ is a **G-parking function** w.r.t. q [PS04] if for all nonempty $A \subseteq V(G) \setminus \{q\}$, there is $v \in A$ so that

$$f(v) < \text{number of edges from } V(G) \setminus A \text{ to } v.$$

Example.



The **level** of f, denoted by $\text{lvl}(f)$, is

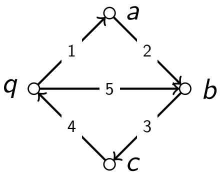
$$\text{lvl}(f) = |E(G)| - |V(G)| + 1 - \sum_{v \in V(G) \setminus \{q\}} f(v).$$

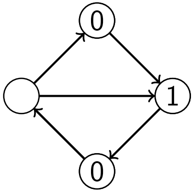
G-parking function

A function $f : V(G) \setminus \{q\} \to \mathbb{N}$ is a **G-parking function** w.r.t. q [PS04] if for all nonempty $A \subseteq V(G) \setminus \{q\}$, there is $v \in A$ so that

$$f(v) < \text{number of edges from } V(G) \setminus A \text{ to } v.$$

Example.





The **level** of f, denoted by $\mathrm{lvl}(f)$, is

$$\mathrm{lvl}(f) = |E(G)| - |V(G)| + 1 - \sum_{v \in V(G) \setminus \{q\}} f(v).$$

Connections to greedoid polynomial

Theorem 1 (C.,2015)

$$\sum_{f} y^{lvl(f)} = \sum_{T} y^{ext(T)} = \lambda_q(G; y).$$

Corollary 2

There is a bijection between G-parking functions w.r.t q and q-rooted spanning trees of G that translates level to external activity.

Corollary 2 is a generalization of Cori-Le Borgne bijection for undirected graphs [CLB03].

Connections to greedoid polynomial

Theorem 1 (C.,2015)

$$\sum_{f} y^{lvl(f)} = \sum_{T} y^{ext(T)} = \lambda_q(G; y).$$

Corollary 2

There is a bijection between G-parking functions w.r.t q and q-rooted spanning trees of G that translates level to external activity.

Corollary 2 is a generalization of Cori-Le Borgne bijection for undirected graphs [CLB03].

Connections to greedoid polynomial

Theorem 1 (C.,2015)

$$\sum_{f} y^{lvl(f)} = \sum_{T} y^{ext(T)} = \lambda_q(G; y).$$

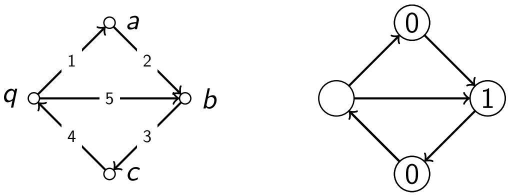
Corollary 2

There is a bijection between G-parking functions w.r.t q and q-rooted spanning trees of G that translates level to external activity.

Corollary 2 is a generalization of Cori-Le Borgne bijection for undirected graphs [CLB03].

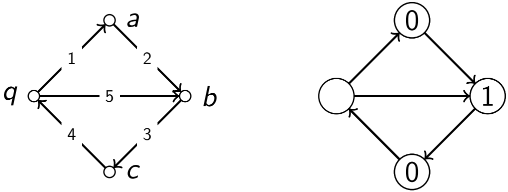
The bijection

The graph G and the G-parking function.



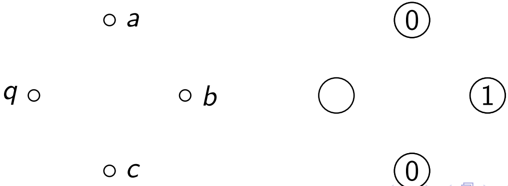
The bijection

The graph G and the G-parking function.



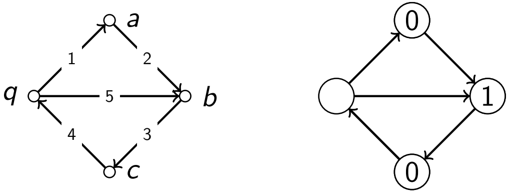
The algorithm.

$S = \{q\}$, $E = \{1, 2, 3, 4, 5\}$, $E_S = \{1, 5\}$.



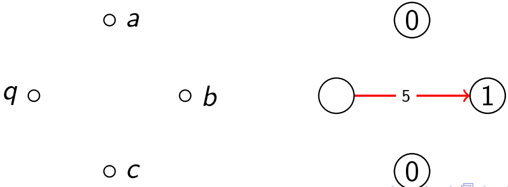
The bijection

The graph G and the G-parking function.



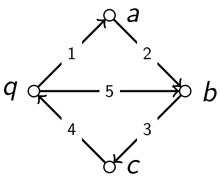
The algorithm.

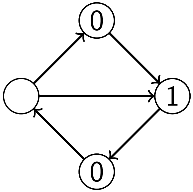
$S = \{q\}$, $E = \{1, 2, 3, 4, 5\}$, $E_S = \{1, 5\}$.



The bijection

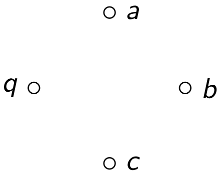
The graph G and the G-parking function.

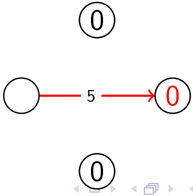




The algorithm.

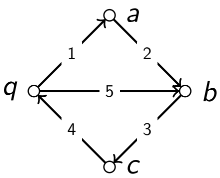
$S = \{q\}$, $E = \{1, 2, 3, 4, \not{5}\}$, $E_S = \{1, 5\}$.

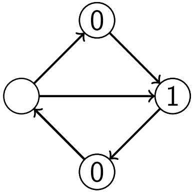




The bijection

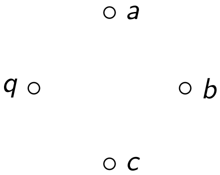
The graph G and the G-parking function.

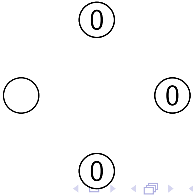




The algorithm.

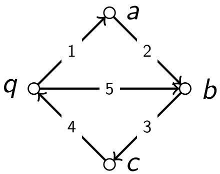
$S = \{q\}$, $E = \{1, 2, 3, 4\}$, $E_S=\{1\}$.

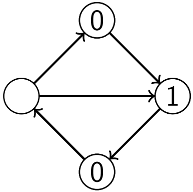




The bijection

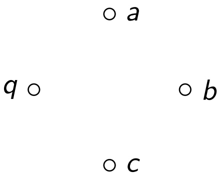
The graph G and the G-parking function.

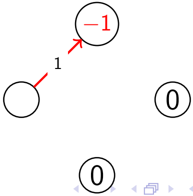




The algorithm.

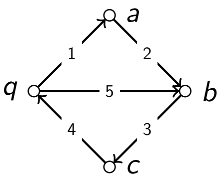
$S = \{q\},\ E = \{\not{1}, 2, 3, 4\},\ E_S = \{1\}.$

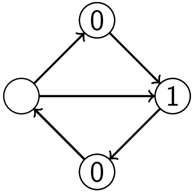




The bijection

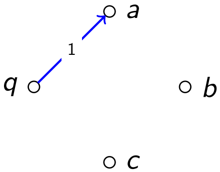
The graph G and the G-parking function.

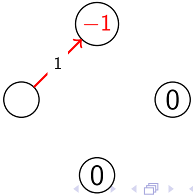




The algorithm.

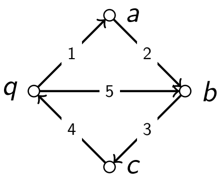
S={q, a}, $E = \{\not{1}, 2, 3, 4\}$, E_S={1}.

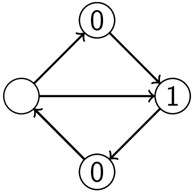




The bijection

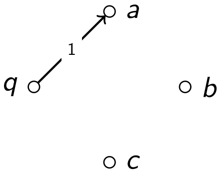
The graph G and the G-parking function.

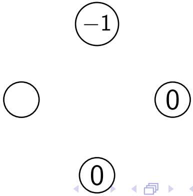




The algorithm.

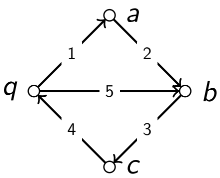
S={q, a}, $E = \{2, 3, 4\}$, E_S={2}.

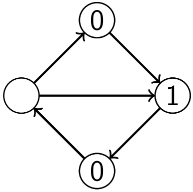




The bijection

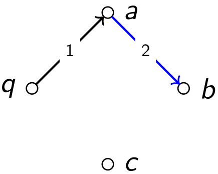
The graph G and the G-parking function.

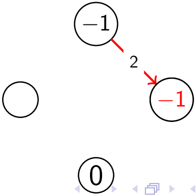




The algorithm.

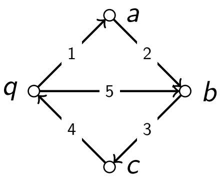
S={q, a, b}, $E = \{\not{2}, 3, 4\}$, E_S={2}.

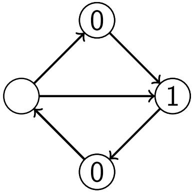




The bijection

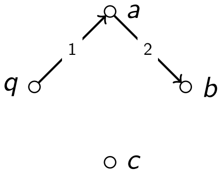
The graph G and the G-parking function.

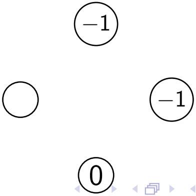




The algorithm.

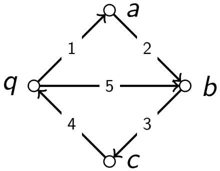
S={q, a, b}, $E = \{3, 4\}$, E_S={3}.

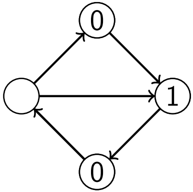




The bijection

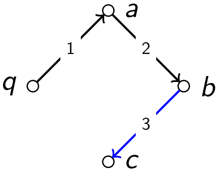
The graph G and the G-parking function.

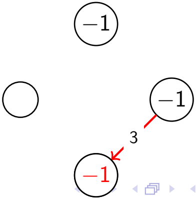




The algorithm.

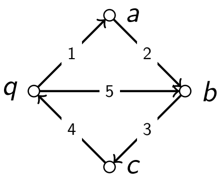
$S=\{q, a, b, c\}$, $E = \{\not{3}, 4\}$, $E_S = \{3\}$.

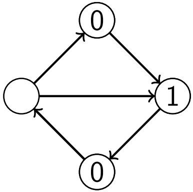




The bijection

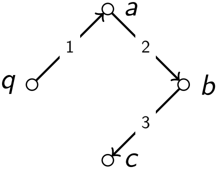
The graph G and the G-parking function.

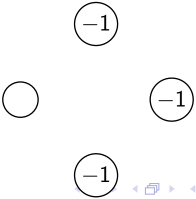




The algorithm.

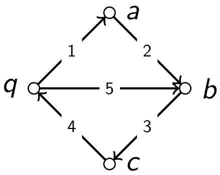
S={q, a, b, c}, $E = \{4\}$, E_S={4}.

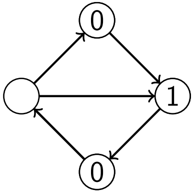




The bijection

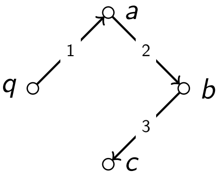
The graph G and the G-parking function.

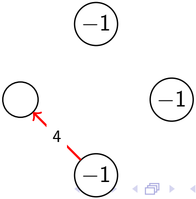




The algorithm.

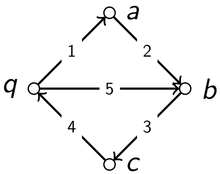
S={q, a, b, c}, $E = \{\not 4\}$, E_S={4}.

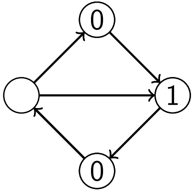




The bijection

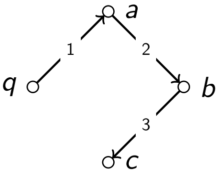
The graph G and the G-parking function.

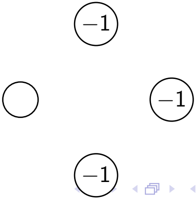




The algorithm.

S={q, a, b, c}, $E = \{\}$, E_S={}. Algorithm stops.





Chip-firing game

The sinkless **Chip-firing game** [BTW88], also known as **abelian sandpile model** and **Bak-Tang-Wiesenfield model**, is a dynamical system on a digraph G.

Each vertex $v \in V(G)$ is assigned $c(v)$ chips, and v is **unstable** if $c(v)$ is greater than or equal to the outdegree of v. If v is unstable, then we can **topple** the vertex by sending a chip through each of its outgoing edges to its neighbors.

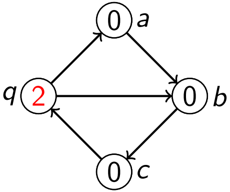
Example of toppling moves.

Chip-firing game

The sinkless **Chip-firing game** [BTW88], also known as **abelian sandpile model** and **Bak-Tang-Wiesenfield model**, is a dynamical system on a digraph G.

Each vertex $v \in V(G)$ is assigned $c(v)$ chips, and v is **unstable** if $c(v)$ is greater than or equal to the outdegree of v. If v is unstable, then we can **topple** the vertex by sending a chip through each of its outgoing edges to its neighbors.

Example of toppling moves.

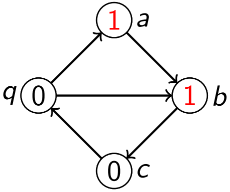


Chip-firing game

The sinkless **Chip-firing game** [BTW88], also known as **abelian sandpile model** and **Bak-Tang-Wiesenfield model**, is a dynamical system on a digraph G.

Each vertex $v \in V(G)$ is assigned $c(v)$ chips, and v is **unstable** if $c(v)$ is greater than or equal to the outdegree of v. If v is unstable, then we can **topple** the vertex by sending a chip through each of its outgoing edges to its neighbors.

Example of toppling moves.

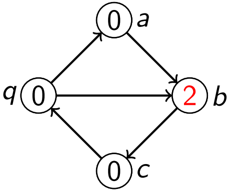


Chip-firing game

The sinkless **Chip-firing game** [BTW88], also known as **abelian sandpile model** and **Bak-Tang-Wiesenfield model**, is a dynamical system on a digraph G.

Each vertex $v \in V(G)$ is assigned $c(v)$ chips, and v is **unstable** if $c(v)$ is greater than or equal to the outdegree of v. If v is unstable, then we can **topple** the vertex by sending a chip through each of its outgoing edges to its neighbors.

Example of toppling moves.

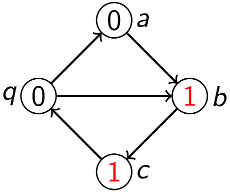


Chip-firing game

The sinkless **Chip-firing game** [BTW88], also known as **abelian sandpile model** and **Bak-Tang-Wiesenfield model**, is a dynamical system on a digraph G.

Each vertex $v \in V(G)$ is assigned $c(v)$ chips, and v is **unstable** if $c(v)$ is greater than or equal to the outdegree of v. If v is unstable, then we can **topple** the vertex by sending a chip through each of its outgoing edges to its neighbors.

Example of toppling moves.



Recurrent configuration

A **chip configuration** is a function $c : V(G) \to \mathbb{N}$, where $c(v)$ is the number of chips in vertex v. We write $c \to c'$ if there is a sequence of toppling moves from c to c'.

A chip configuration c is **recurrent** if:

- c is unstable;
- If $c \to c'$, then $c' \to c$.

For two recurrent configurations c and c', we write $c \sim c'$ if $c \to c'$ and $c' \to c$.

The **level** of a recurrent equivalence class $[c]$ is

$$\mathrm{lvl}([c]) = \sum_{v \in V(G)} c(v).$$

We use r_n to denote the number of recurrent equivalence classes with level n.

Recurrent configuration

A **chip configuration** is a function $c : V(G) \to \mathbb{N}$, where $c(v)$ is the number of chips in vertex v. We write $c \to c'$ if there is a sequence of toppling moves from c to c'.

A chip configuration c is **recurrent** if:

- c is unstable;
- If $c \to c'$, then $c' \to c$.

For two recurrent configurations c and c', we write $c \sim c'$ if $c \to c'$ and $c' \to c$.

The **level** of a recurrent equivalence class $[c]$ is

$$\mathrm{lvl}([c]) = \sum_{v \in V(G)} c(v).$$

We use r_n to denote the number of recurrent equivalence classes with level n.

Recurrent configuration

A **chip configuration** is a function $c : V(G) \to \mathbb{N}$, where $c(v)$ is the number of chips in vertex v. We write $c \to c'$ if there is a sequence of toppling moves from c to c'.
A chip configuration c is **recurrent** if:

- c is unstable;
- If $c \to c'$, then $c' \to c$.

For two recurrent configurations c and c', we write $c \sim c'$ if $c \to c'$ and $c' \to c$.
The **level** of a recurrent equivalence class $[c]$ is

$$\mathrm{lvl}([c]) = \sum_{v \in V(G)} c(v).$$

We use r_n to denote the number of recurrent equivalence classes with level n.

Connections to greedoid polynomial

Theorem 3 (C.,2015)

If G is an Eulerian digraph, then

$$\sum_{n \geq 0} r_n y^n = \frac{\lambda_q(G; y)}{(1-y)}.$$

Theorem 4 (C., 2015)

The greedoid polynomial $\lambda_q(G; y)$ does not depend on the choice of root vertex q if and only if G is an Eulerian digraph.

Theorem 5 (C., 2015)

Let $G^\top$ be the reverse of the digraph G. We have G is an Eulerian digraph if and only if $\lambda_q(G; y) = \lambda_q(G^\top; y)$ for all $q \in V(G)$.

Connections to greedoid polynomial

Theorem 3 (C.,2015)

If G is an Eulerian digraph, then

$$\sum_{n\geq 0} r_n y^n = \frac{\lambda_q(G;y)}{(1-y)}.$$

Theorem 4 (C., 2015)

The greedoid polynomial $\lambda_q(G;y)$ does not depend on the choice of root vertex q if and only if G is an Eulerian digraph.

Theorem 5 (C., 2015)

Let $G^{\top}$ be the reverse of the digraph G. We have G is an Eulerian digraph if and only if $\lambda_q(G;y) = \lambda_q(G^{\top};y)$ for all $q \in V(G)$.

Connections to greedoid polynomial

Theorem 3 (C.,2015)

If G is an Eulerian digraph, then

$$\sum_{n \geq 0} r_n y^n = \frac{\lambda_q(G;y)}{(1-y)}.$$

Theorem 4 (C., 2015)

The greedoid polynomial $\lambda_q(G;y)$ does not depend on the choice of root vertex q if and only if G is an Eulerian digraph.

Theorem 5 (C., 2015)

Let $G^\top$ be the reverse of the digraph G. We have G is an Eulerian digraph if and only if $\lambda_q(G;y) = \lambda_q(G^\top;y)$ for all $q \in V(G)$.

Further research

- Find a bijective proof for Theorem 4 and Theorem 5.
- Prove or disprove that $\lambda_q(G; y)$ is unimodal.

Questions?

Thank you!

Anders Björner, Bernhard Korte, and László Lovász.
Homotopy properties of greedoids.
Adv. in Appl. Math., 6(4):447–494, 1985.

Per Bak, Chao Tang, and Kurt Wiesenfeld.
Self-organized criticality.
Phys. Rev. A (3), 38(1):364–374, 1988.

F. R. K. Chung and R. L. Graham.
On the cover polynomial of a digraph.
J. Combin. Theory Ser. B, 65(2):273–290, 1995.

Robert Cori and Yvan Le Borgne.
The sand-pile model and Tutte polynomials.
Adv. in Appl. Math., 30(1-2):44–52, 2003.
Formal power series and algebraic combinatorics (Scottsdale, AZ, 2001).

Criel Merino López.
Chip firing and the Tutte polynomial.
Ann. Comb., 1(3):253–259, 1997.