The role of computer experiments in the theory of word-representable graphs

Sergey Kitaev

University of Strathclyde

25th October, 2018

ACiD seminar

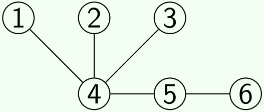
A quick introduction

Basic idea

A **motivation** to study various encodings of graphs by words is the hope, for a given (difficult) problem on graphs, to be able to find a **suitable encoding** that would allow to translate the problem on graphs to an **easier** problem on words, and solve it. Such an encoding does **not** have to be **optimal in size**.

Example: Prüfer codes (sequences) to encode labelled trees (1918)

Provides a proof of **Cayley's formula** (n^{n-2}) to enumerate labelled trees on n vertices.



Remove the leaf with the **smallest label** and record its neighbour: 4445 (the last neighbour does not need to be recorded)

Overview of the talk

Word-representable graphs

- Some history + motivation + literature + definitions
- Key results (incl. characterisation via certain orientations)

Impact of computer experiments to the theory

- Earlier computer experiments + available software
- Enumeration
- Finding forbidden subgraphs
  - Triangulations of grid-covered cylinder graphs
  - Split graphs

Some History

2004

Steve Seif

Definition 3.6.10. The six-element monoid $\mathbf{B_2^1} = \langle B_2^1; \cdot \rangle$, the Perkins semigroup, has the following elements with the usual matrix multiplication operation:

$$\mathbf{0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \ \mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ a = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \ a' = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

$$aa' = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \ a'a = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Some History

2004

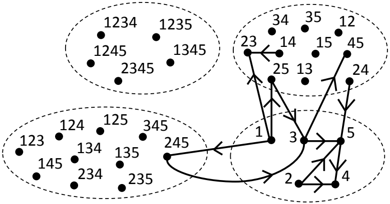
Steve Seif

Definition 3.6.10. The six-element monoid $\mathbf{B_2^1} = \langle B_2^1; \cdot \rangle$, the Perkins semigroup, has the following elements with the usual matrix multiplication operation:

$$\mathbf{0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},\ \mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ a = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},\ a' = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

$$aa' = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},\ a'a = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Alternation digraph



Some History

Major contributors to the theory of word-representable graphs

Magnus M. Halldorsson

Artem Pyatkin

Some History

Major contributors to the theory of word-representable graphs

Magnus M. Halldorsson

Artem Pyatkin

Other contributors: Özgür Akgün, Posper Akrobotu, Bas Broere, Herman Chen, Gi-Sang Cheon, Andrew Collins, Jessica Enright, Alice Gao, Ian Gent, Marc Glen, Christopher Jefferson, Miles Jones, Jinha Kim, Minki Kim, Sergey Kitaev, Alexander Konovalov, Vincent Limouzy, Steven Linton, Vadim Lozin, Yelena Mandelshtam, Zuzana Masárová, Jeff Remmel, Akira Saito, Pavel Salimov, Chris Severs, Brian Sun, Henning Úlfarsson, Hans Zantema, Philip Zhang, and several others.

Motivation

- Study of the Perkins semigroup (original motivation) — Algebra

Motivation

- Study of the Perkins semigroup (original motivation) — Algebra
- Generalisation of several classes of graphs — Graph Theory

Motivation

- Study of the Perkins semigroup (original motivation) — Algebra
- Generalisation of several classes of graphs — Graph Theory
- Relies on various properties of words — Combinatorics on Words

Motivation

- Study of the Perkins semigroup (original motivation) — Algebra
- Generalisation of several classes of graphs — Graph Theory
- Relies on various properties of words — Combinatorics on Words
- Solving algorithmic questions — Computer Science

Motivation

- Study of the Perkins semigroup (original motivation) — Algebra
- Generalisation of several classes of graphs — Graph Theory
- Relies on various properties of words — Combinatorics on Words
- Solving algorithmic questions — Computer Science
- Solving certain scheduling problems — Operations Research

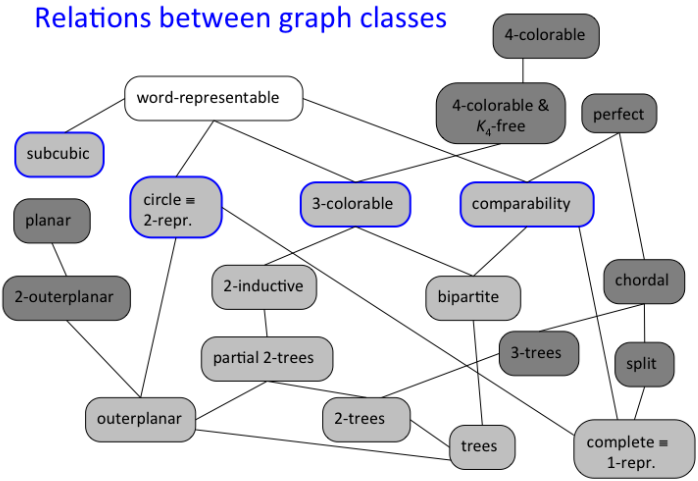
Motivation

- Study of the Perkins semigroup (original motivation) — Algebra
- Generalisation of several classes of graphs — Graph Theory
- Relies on various properties of words — Combinatorics on Words
- Solving algorithmic questions — Computer Science
- Solving certain scheduling problems — Operations Research
- Beautiful mathematics — Mathematics

Motivation

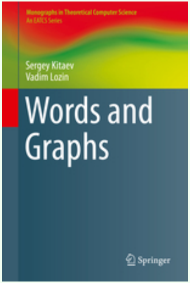
- Study of the Perkins semigroup (original motivation) — Algebra
- Generalisation of several classes of graphs — Graph Theory
- Relies on various properties of words — Combinatorics on Words
- Solving algorithmic questions — Computer Science
- Solving certain scheduling problems — Operations Research
- Beautiful mathematics — Mathematics
- Just fun — Human Science

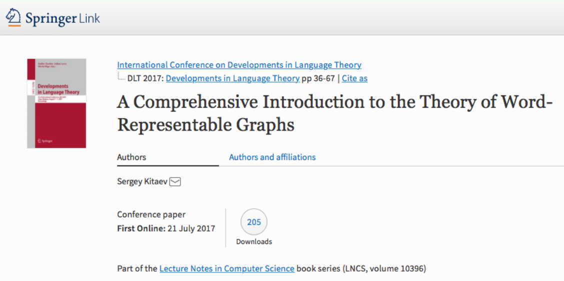
Relations between graph classes



Literature

The best way to learn about the subject





Literature

- Akgun, Gent, Kitaev, Zantema. Solving computational problems in the theory of word-representable graphs. arXiv:1808.01215 (2018)
- Akrobotu, Kitaev, Masárová. On word-representability of polyomino triangulations. *Siberian Advan. Math.* (2015)
- Broere. Word-representable graphs. Master Thesis (2018).
- Broere, Zantema. The k-cube is k-representable. arXiv:1808.01800 (2018)
- Chen, Kitaev, Sun. Word-representability of face subdivisions of triangular grid graphs. *Graphs and Combin.* (2016)
- Chen, Kitaev, Sun. Word-representability of triangulations of grid-covered cylinder graphs. *Discr. Appl. Math.* (2016)
- Cheon, Kim, Kim, Kitaev, Pyatkin. On k-11-representable graphs. Journal of Combinatorics. (2018)
- Collins, Kitaev, Lozin. New results on word-representable graphs. *Discr. Appl. Math.* (2017)
- Daigavane, Singh, George. 2-uniform words: cycle graphs, and an algorithm to verify specific word-representations of graphs. arXiv:1806.04673 (2018)
- Gao, Kitaev, Zhang. On 132-representable graphs. *Australasian Journal of Combinatorics* (2017)
- Glen. Colourability and word-representability of near-triangulations. arXiv:1605.01688 (2016)
- Glen, Kitaev. Word-representability of triangulations of rectangular polyomino with a single domino tile. *J. Combin. Math. Combin. Comput.* (2017)
- Halldórsson, Kitaev, Pyatkin. Graphs capturing alternations in words. *Lecture Notes Comp. Sci.* (2010)
- Halldórsson, Kitaev, Pyatkin. Alternation graphs. *Lecture Notes Comp. Sci.* (2011)
- Halldórsson, Kitaev, Pyatkin. Semi-transitive orientations and word-representable graphs. *Discr. Appl. Math.* (2016)
- Jones, Kitaev, Pyatkin, Remmel. Representing graphs via pattern avoiding words. *Electron. J. Combin.* (2015)
- Kitaev. On graphs with representation number 3. *J. Autom. Lang. Combin.* (2013)
- Kitaev. Existence of u-representation of graphs. *Journal of Graph Theory* (2017)
- Kitaev, Pyatkin. On representable graphs. *J. Autom. Lang. Combin.* (2008)
- Kitaev, Salimov, Severs, Úlfarsson. On the representability of line graphs. *Lecture Notes Comp. Sci.* (2011)
- Mandelshtam. On graphs representable by pattern-avoiding words. *Discussiones Mathematicae Graph Theory* (2018)

Basic definitions

Alternating letters in a word

In the word 23125413241362, the letters 2 and 3 alternate because removing all other letters we obtain 2323232 where 2 and 3 come in alternating order.

Basic definitions

Alternating letters in a word

In the word 23125413241362, the letters 2 and 3 alternate because removing all other letters we obtain 2323232 where 2 and 3 come in alternating order.

Also, 1 and 3 do not alternate because removing all other letters we obtain 311313 and the factor 11 breaks the alternating order.

Basic definitions

Alternating letters in a word

In the word 23125413241362, the letters 2 and 3 alternate because removing all other letters we obtain 2323232 where 2 and 3 come in alternating order.

Also, 1 and 3 do not alternate because removing all other letters we obtain 311313 and the factor 11 breaks the alternating order.

Note that removing all letters but 5 and 6 we obtain 56 showing that the letters 5 and 6 alternate (by definition).

Basic definitions

All graphs considered by us are simple (no loops, no multiple edges).

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Basic definitions

All graphs considered by us are simple (no loops, no multiple edges).

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Word-representant

w is a word-representant. We say that w represents G.

Basic definitions

All graphs considered by us are simple (no loops, no multiple edges).

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Word-representant

w is a word-representant. We say that w represents G.

Remark

We deal with **unlabelled graphs**. However, to apply the definition, we need to label graphs. Any labelling of a graph is **equivalent** to any other labelling because letters in w can always be renamed.

Basic definitions

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Remark

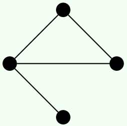
The class of word-representable graphs is hereditary. That is, removing a vertex v in a word-representable graph G results in a word-representable graph G'. Indeed, if w represents G then w with v removed represents G'.

Basic definitions

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Example

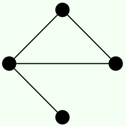
The graph  is word-representable.

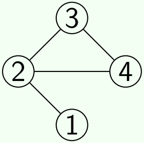
Basic definitions

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Example

The graph  is word-representable.

Indeed, 

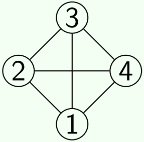
can be represented by 1213423.

Basic definitions

Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Example: representing complete graphs and empty graphs



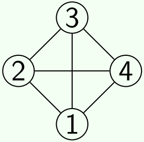
can be represented by 1234 or 12341234.
or by any permutation of $\{1, 2, 3, 4\}$.

Basic definitions

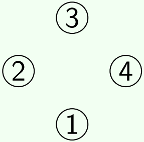
Word-representable graph

A graph $G = (V, E)$ is word-representable if there exists a word w over the alphabet V such that letters x and y, $x \neq y$, alternate in w **if and only if** $xy \in E$. (w **must** contain **each** letter in V)

Example: representing complete graphs and empty graphs



can be represented by 1234 or 12341234.
or by any permutation of $\{1, 2, 3, 4\}$.



can be represented by 12344321 or 11223344.

k-representability and graph's representation number

Uniform word

k-uniform word = **each** letter occurs k times

243321442311 is a 3-uniform word

23154 is a 1-uniform word or permutation

k-representability and graph's representation number

Uniform word

k-uniform word = **each** letter occurs k times
243321442311 is a 3-uniform word
23154 is a 1-uniform word or permutation

k-representable graph

A graph is k-representable if there exists a k-uniform word representing it.

k-representability and graph's representation number

Uniform word

k-uniform word = **each** letter occurs k times
243321442311 is a 3-uniform word
23154 is a 1-uniform word or permutation

k-representable graph

A graph is k-representable if there exists a k-uniform word representing it.

Theorem (SK, Pyatkin; 2008)

A graph is word-representable **iff** *it is k-representable for some k.*

k-representability and graph's representation number

Uniform word

k-uniform word = **each** letter occurs k times
243321442311 is a 3-uniform word
23154 is a 1-uniform word or permutation

k-representable graph

A graph is k-representable if there exists a k-uniform word representing it.

Theorem (SK, Pyatkin; 2008)

A graph is word-representable **iff** *it is k-representable for some k.*

Theorem (SK, Pyatkin; 2008)

k-representability implies $(k+1)$-representability.

k-representability and graph's representation number

Graph's representation number

Graph's representation number is the **least** k such that the graph is k-representable. This notion is well-defined for word-representable graphs. For non-word-representable graphs, we let $k = \infty$.

k-representability and graph's representation number

Graph's representation number

Graph's representation number is the **least** k such that the graph is k-representable. This notion is well-defined for word-representable graphs. For non-word-representable graphs, we let $k = \infty$.

Notation

$\mathcal{R}(G)$ denotes G's representation number & $\mathcal{R}_k = \{G : \mathcal{R}(G) = k\}$

k-representability and graph's representation number

Graph's representation number

Graph's representation number is the **least** k such that the graph is k-representable. This notion is well-defined for word-representable graphs. For non-word-representable graphs, we let $k = \infty$.

Notation

$\mathcal{R}(G)$ denotes G's representation number & $\mathcal{R}_k = \{G : \mathcal{R}(G) = k\}$

Observation

$\mathcal{R}_1 = \{G : G \text{ is a complete graph}\}$

k-representability and graph's representation number

Graph's representation number

Graph's representation number is the **least** k such that the graph is k-representable. This notion is well-defined for word-representable graphs. For non-word-representable graphs, we let $k = \infty$.

Notation

$\mathcal{R}(G)$ denotes G's representation number & $\mathcal{R}_k = \{G : \mathcal{R}(G) = k\}$

Observation

$\mathcal{R}_1 = \{G : G \text{ is a complete graph}\}$

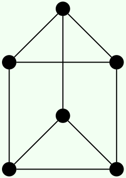
Theorem (Halldórsson, SK, Pyatkin; 2011)

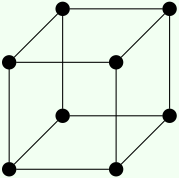
$\mathcal{R}_1 \cup \mathcal{R}_2 = \{G : G \textit{ is a circle graph}\}$

Graphs with representation number 3

No characterization is known, but a number of interesting results are obtained. Prisms are just one example.

Prisms





Theorem (SK, Pyatkin; 2008)

Every *prism is 3-representable.*

Theorem (SK; 2013)

None *of the prisms is 2-representable.*

Comparability graphs

Transitive orientation

An orientation of a graph is transitive if presence of edges $u \to v$ and $v \to z$ implies presence of the edge $u \to z$.

Comparability graphs

Transitive orientation

An orientation of a graph is transitive if presence of edges $u \to v$ and $v \to z$ implies presence of the edge $u \to z$.

Comparability graph

A non-oriented graph is a comparability graph if it admits a transitive orientation.

Comparability graphs

Transitive orientation

An orientation of a graph is transitive if presence of edges $u \to v$ and $v \to z$ implies presence of the edge $u \to z$.

Comparability graph

A non-oriented graph is a comparability graph if it admits a transitive orientation.

Theorem (SK, Pyatkin; 2008)

G is word-representable $\Rightarrow$ the neighbourhood of each vertex is a **comparability graph**.

Comparability graphs

Transitive orientation

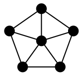
An orientation of a graph is transitive if presence of edges $u \to v$ and $v \to z$ implies presence of the edge $u \to z$.

Comparability graph

A non-oriented graph is a comparability graph if it admits a transitive orientation.

Theorem (SK, Pyatkin; 2008)

G is word-representable $\Rightarrow$ the neighbourhood of each vertex is a **comparability graph**.

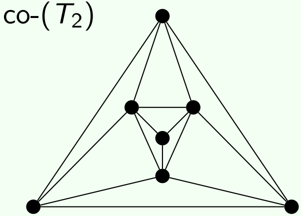
The **smallest** non-word-representable graph is the wheel $W_5 =$ 

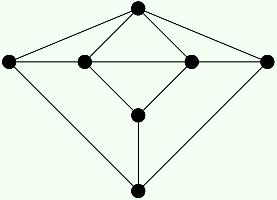
Converse to the last theorem is not true

Theorem (Halldórsson, SK, Pyatkin; 2010)

G is word-representable $\not\Leftarrow$ the neighbourhood of each vertex is **permutationally representable** *(is a* **comparability graph***).*

Minimal counterexamples





Semi-transitive orientations

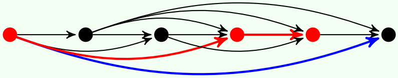
Shortcut

A shortcut is an oriented graph that

- is **acyclic** (that it, there are **no directed cycles**);
- has **at least 4 vertices**;
- has **exactly one** source (no edges coming in), **exactly one** sink (no edges coming out), and a **directed path** from the source to the sink that goes through **every** vertex in the graph;
- has an edge connecting the **source** to the **sink**;
- is **not** transitive (that it, there exist vertices u, v and z such that $u \to v$ and $v \to z$ are edges, but there is **no** edge $u \to z$).

Semi-transitive orientations

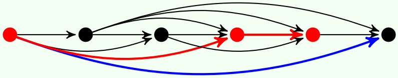
Example of a shortcut



The part of the graph in red shows **non-transitivity**. There are **two other violations** of transitivity.

Semi-transitive orientations

Example of a shortcut



The part of the graph in red shows **non-transitivity**. There are **two other violations** of transitivity.

The blue edge, from the **source** to the **sink**, justifies the name "short-cut" for this type of graphs. Indeed, **instead** of going through the **longest directed path** from the **source** to the **sink**, we can shortcut it by going directly through the single edge. The blue edge is called shortcutting edge.

Semi-transitive orientations

Semi-transitive orientation

An orientation of a graph is semi-transitive if it is

- **acyclic**, and
- **shortcut-free**.

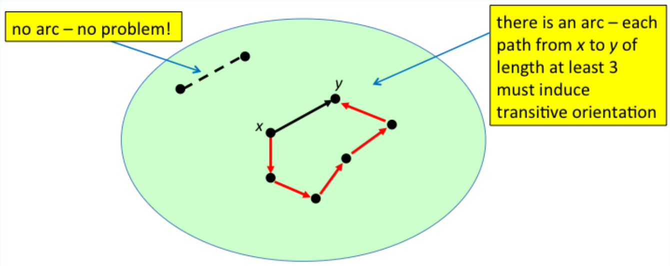
Semi-transitive orientations

Semi-transitive orientation

An orientation of a graph is semi-transitive if it is

- **acyclic**, and
- **shortcut-free**.

Checking if a given acyclic orientation is semi-transitive



A key result in the theory of word-representable graphs

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

A graph G is word-representable **iff** *G admits a semi-transitive orientation.*

A key result in the theory of word-representable graphs

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

A graph G is word-representable **iff** *G admits a semi-transitive orientation.*

Proof.

"$\Leftarrow$" Rather complicated and is **omitted**. An **algorithm** was created to turn a **semi-transitive orientation** of a graph into a **word-representant**.

$\square$

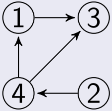
A key result in the theory of word-representable graphs

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

A graph G is word-representable **iff** *G admits a semi-transitive orientation.*

Proof.

"$\Leftarrow$" Rather complicated and is **omitted**. An **algorithm** was created to turn a **semi-transitive orientation** of a graph into a **word-representant**.
"$\Rightarrow$" **Proof idea:** Given a word, say, $w = 2421341$, orient the graph represented by w by letting $x \to y$ be an edge if the **leftmost** x is to the left of the **leftmost** y in w, to obtain a **semi-transitive orientation**:



□

The shortest length of a word-representant

An upper bound on the length of a word-representant

Any **complete graph** is 1-representable.

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

Each **non-complete** *word-representable graph G is $2(n-\kappa(G))$-representable, where $\kappa(G)$ is the size of the maximum clique in G.*

The shortest length of a word-representant

An upper bound on the length of a word-representant

Any **complete graph** is 1-representable.

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

Each **non-complete** *word-representable graph G is* $2(n-\kappa(G))$*-representable, where $\kappa(G)$ is the size of the maximum clique in G.*

A corollary to the last theorem

The recognition problem of word-representability is in **NP**.

The shortest length of a word-representant

An upper bound on the length of a word-representant

Any **complete graph** is 1-representable.

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

Each **non-complete** *word-representable graph G is* $2(n-\kappa(G))$*-representable, where $\kappa(G)$ is the size of the maximum clique in G.*

A corollary to the last theorem

The recognition problem of word-representability is in **NP**.

Theorem (Limouzy; 2014)

It is an **NP-complete problem** *to recognize whether a given graph is word-representable.*

3-colorable graphs

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

Any 3-colorable graph is word-representable.

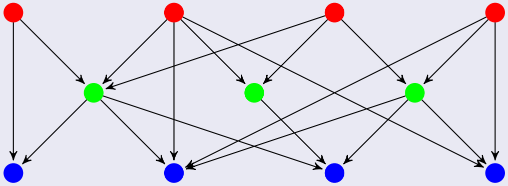
3-colorable graphs

Theorem (Halldórsson, Kitaev, Pyatkin; 2015)

Any 3-colorable graph is word-representable.

Proof.

Coloring a 3-**colorable graph** in three colors Red, Green and Blue, and orienting the edges as Red $\rightarrow$ Green $\rightarrow$ Blue, we obtain a **semi-transitive orientation**. Indeed, obviously there are **no cycles**, and because the longest directed path involves only three vertices, there are **no shortcuts**.

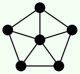


□

Earlier impact of computer experiments

Representation of graphs of up to 6 vertices

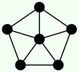
Artem Pyatkin has represented **all** graphs on up to 6 vertices **but**

$W_5 =$  which was then proved to be non-word-representable.

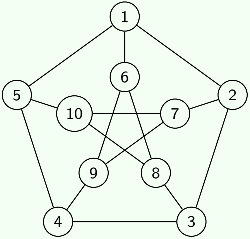
Earlier impact of computer experiments

Representation of graphs of up to 6 vertices

Artem Pyatkin has represented **all** graphs on up to 6 vertices **but**

$W_5 =$  which was then proved to be non-word-representable.

Petersen's graph – a turned down conjecture

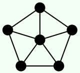


Two **non-equivalent** 3-representations (by Alexander Konovalov and Steven Linton):
1387296(10)7493541283(10)7685(10)194562
134(10)58679(10)273412835(10)6819726495

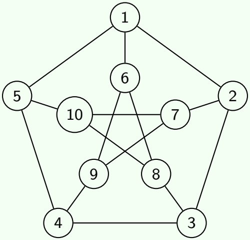
Earlier impact of computer experiments

Representation of graphs of up to 6 vertices

Artem Pyatkin has represented **all** graphs on up to 6 vertices **but**

$W_5 =$  which was then proved to be non-word-representable.

Petersen's graph – a turned down conjecture



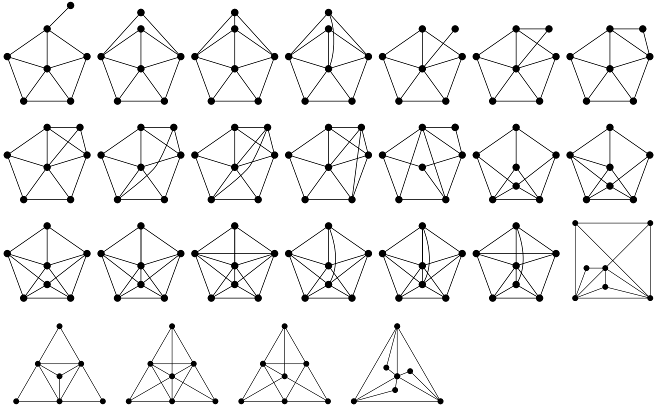
Two **non-equivalent** 3-representations (by Alexander Konovalov and Steven Linton):
1387296(10)7493541283(10)7685(10)194562
134(10)58679(10)273412835(10)6819726495

Theorem (Halldórsson, SK, Pyatkin; 2010)

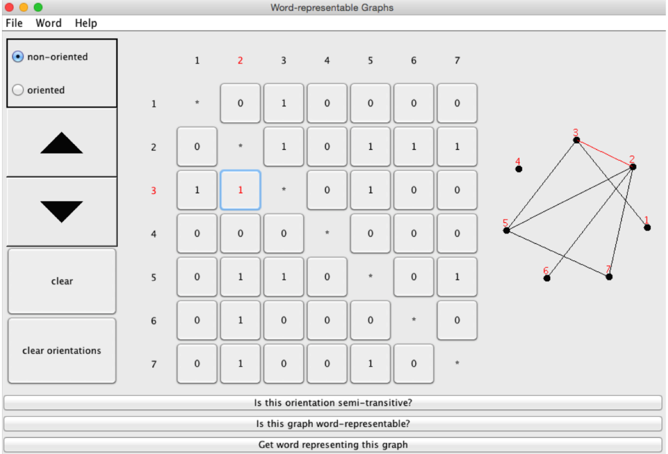
Petersen graph is **not** *2-representable.*

All 25 non-word-representable graphs on 7 vertices

The following picture was created by Herman Chen. It was useful in (i) finding various **counter-examples** (ii) a **generalization** of word-representable graphs (iii) to support a **conjecture** saying that the **line graph** of a non-word-representable graph is always non-word-representable.



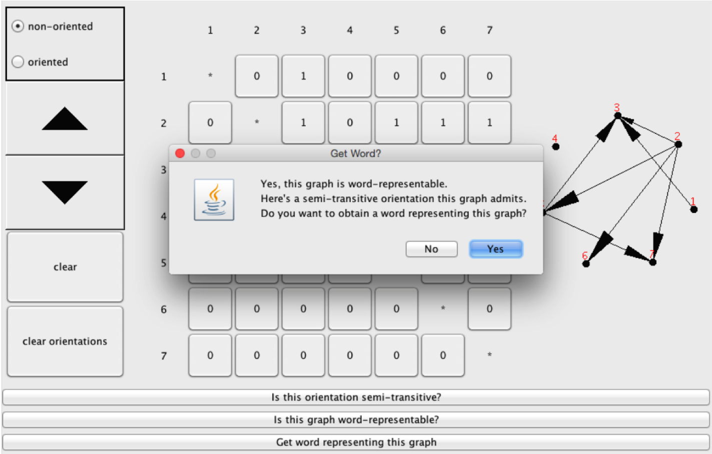
Software by Marc Glen to study word-representable graphs



Available at

https://personal.cis.strath.ac.uk/sergey.kitaev/word-representable-graphs.html

Software by Marc Glen to study word-representable graphs



Available at

https://personal.cis.strath.ac.uk/sergey.kitaev/word-representable-graphs.html

Software by Hans Zantema for word-representable graphs

Available at http://www.win.tue.nl/ hzantema/reprnr.html

The tool

The tool REPRNR computes the representation number of a graph by encoding its definition into a formula, and then calling the SMT solver Z3. It was developed by Hans Zantema after an inspiring invited lecture on this topic by Sergey Kitaev at the DLT conference in Liege in August 2017.
The notion semi-transitive has been proven to be equivalent to having a finite representation number; for details we refer to the above mentioned book. We also provide a tool SEMITR checking for being semi-transitive, and if so, presenting a corresponing orientation. Again the implementation builds a formula reflecting the definition, and then calls the SMT solver. For running in Windows a zip file is provided containing

- the executable tool REPRNR to be run in Windows in command line;
- the executable tool SEMITR to be run in Windows in command line;
- some auxiliary files like the SMT solver Z3;
- some example graphs.

For running in Linux a zip file is provided, using the SMT solver yices instead of Z3. It contains

- the executable tool REPRNR to be run in Linux in command line;
- the executable tool SEMITR to be run in Linux in command line;
- some auxiliary files like the SMT solver yices;
- some example graphs.

The format

The input consists of of the number n of nodes, followed by summing up all edges. Every edge is indicated by its two node numbers, from 1 to n. To mark the end, the list of edges should end by 0 0. So for the complete graph on three nodes, the input reads:

3
1 2
1 3
2 3
0 0

Some examples are included:
cubegr.txt : the cube
petgr.txt: Peterson's graph
w5gr.txt: the 5-wheel, having no word representation, so the computation should be forced to stop
g4.txt: the graph G4
g5.txt: the graph G5

and without '.txt' extension in Linux. So by running

reprnr < petgr.txt

in the command line mode in Windows, or

./reprnr < petgr

in Linux, it is first established that the Peterson graph is not 2-representable, and then a 3-representing word is computed and shown.

Distribution of k-representable graphs

Hans Zantema produced the following results:

# of vertices	# of conn. graphs	representation number				
		1	2	3	4	> 4
3	2	1	1	0	0	0
4	6	1	5	0	0	0
5	21	1	20	0	0	0
6	112	1	109	1	0	1
7	853	1	788	39	0	25
8	11,117	1	8335	1852	0	929
9	261,080	1	117,282	88,838	2	54,957

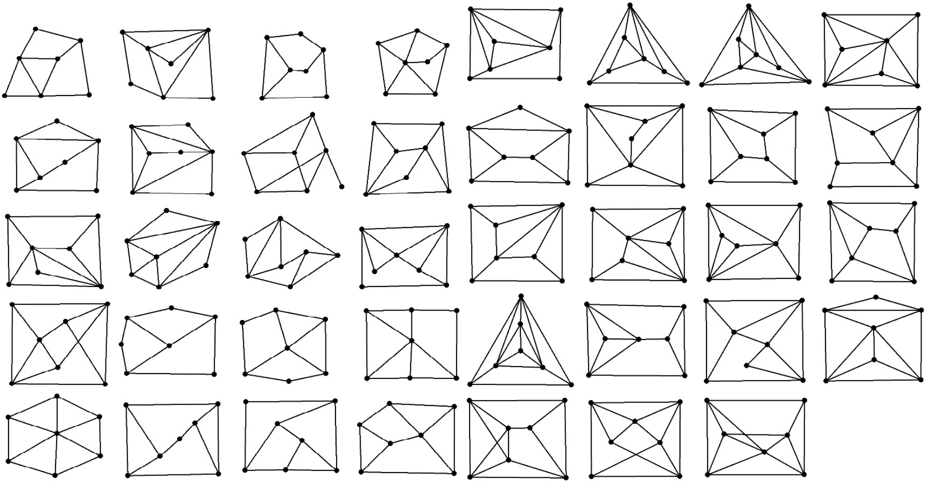
Distribution of k-representable graphs

Hans Zantema produced the following results:

# of vertices	# of conn. graphs	representation number				
		1	2	3	4	> 4
3	2	1	1	0	0	0
4	6	1	5	0	0	0
5	21	1	20	0	0	0
6	112	1	109	1	0	1
7	853	1	788	39	0	25
8	11,117	1	8335	1852	0	929
9	261,080	1	117,282	88,838	2	54,957

One of the major surprises was the 2 in the last row – our prediction was 1 in that place! This made us to question our **conjecture** that a particular graph on $2n+1$ vertices requires a **longest word-representant**.

Hans Zantema produced the following picture.



Enumeration of non-word-representable graphs

Ozgur Akgun and Ian Gent produced the following results:

	# of conn. graphs	All non-word-representable graphs				
		Total	%	Time	Min.	Non-Min.
6	112	1	0.89%	3.0s	1	0
7	853	25	2.93%	4.0s	10	15
8	11,117	929	8.36%	26s	47	882
9	261,080	54,957	21.05%	29m	179	54,778
10	11,716,571	4,880,093	41.65%	74h	-	-
11	1,006,690,565	650,856,040	64.65%	1,100d	-	-

Word-representation of split graphs

Split graph

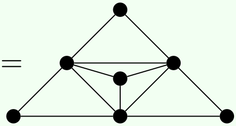
A split graph is a graph in which the vertices can be partitioned into a **clique** and an **independent set**.

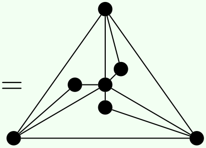
Word-representation of split graphs

Split graph

A split graph is a graph in which the vertices can be partitioned into a **clique** and an **independent set**.

Examples of split graphs

$T_1 =$ 

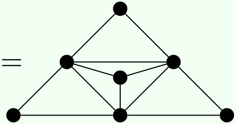
$T_2 =$ 

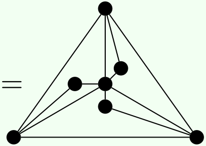
Word-representation of split graphs

Split graph

A split graph is a graph in which the vertices can be partitioned into a **clique** and an **independent set**.

Examples of split graphs

$T_1 =$ 

$T_2 =$ 

Notation

A split graph on n vertices is denoted by $S_n = (E_{n-m}, K_m)$, where K_m is a **maximal clique**, that is, vertices in the **independent set** E_{n-m} are of degree at most $m-1$.

Useful assumptions (for split graphs)

When studying word-representability of **any** graph G, we can assume that

- **each vertex** in G is of **degree at least 2**;
- **no two vertices** in G have the **same set of neighbours**.

Useful assumptions (for split graphs)

When studying word-representability of **any** graph G, we can assume that

- **each vertex** in G is of **degree at least 2**;
- **no two vertices** in G have the **same set of neighbours**.

For a split graph S_n, we can assume that

- a **maximal clique** in S_n is of size ≥ 4 (otherwise S_n is 3-colorable and thus is word-representable);

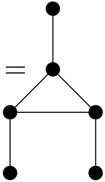
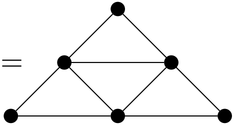
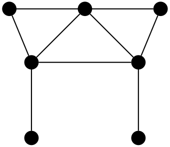
Useful assumptions (for split graphs)

When studying word-representability of **any** graph G, we can assume that

- **each vertex** in G is of **degree at least 2**;
- **no two vertices** in G have the **same set of neighbours**.

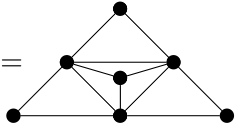
For a split graph S_n, we can assume that

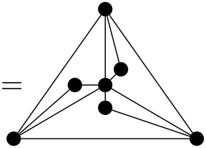
- a **maximal clique** in S_n is of size ≥ 4 (otherwise S_n is 3-colorable and thus is word-representable);
- [Never used so far!] S_n contains **at least** one of

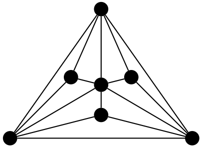
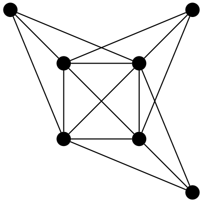
$M_1 =$  $M_2 =$  $M_3 =$ 

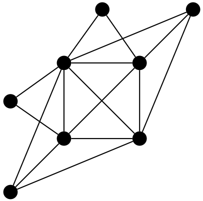
because otherwise S_n is a **comparability graph** by Golumbic's 1980 theorem and thus is word-representable.

Minimal non-word-representable split graphs

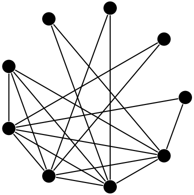
$T_1 =$ 

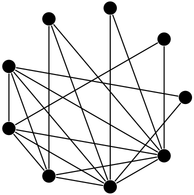
$T_2 =$ 

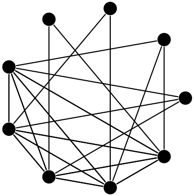
$T_3 =$  $=$ 

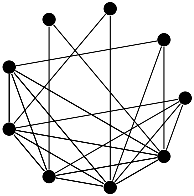
$T_4 =$ 

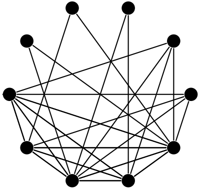
More minimal non-word-representable split graphs

$T_5 =$ 

$T_6 =$ 

$T_7 =$ 

$T_8 =$ 

$T_9 =$ 

Three classification results for split graphs

Computer was **not** used to prove the following theorem.

Theorem (Kitaev, Long, Ma, Wu; 2017)

Let $m \geq 1$ *and* $S_n = (E_{n-m}, K_m)$ *be a split graph. Also, let the degree of* **any** *vertex in* E_{n-m} *be* **at most** 2. *Then* S_n *is word-representable* **iff** S_n *does* **not** *contain the graph* T_2 *and* A_ℓ *as induced subgraphs.*

Three classification results for split graphs

Computer was **not** used to prove the following theorem.

Theorem (Kitaev, Long, Ma, Wu; 2017)

Let $m \geq 1$ and $S_n = (E_{n-m}, K_m)$ be a split graph. Also, let the degree of **any** *vertex in E_{n-m} be* **at most** *2. Then S_n is word-representable* **iff** *S_n does* **not** *contain the graph T_2 and A_ℓ as induced subgraphs.*

Essentially, computer was **not** used to prove the following theorem.

Theorem (Kitaev, Long, Ma, Wu; 2017)

$S_n = (E_{n-4}, K_4)$ is w.-r. **iff** *it does* **not** *contain T_1–T_4 as ind. subgraphs.*

Three classification results for split graphs

Computer was **not** used to prove the following theorem.

Theorem (Kitaev, Long, Ma, Wu; 2017)

Let $m \geq 1$ and $S_n = (E_{n-m}, K_m)$ be a split graph. Also, let the degree of **any** *vertex in E_{n-m} be* **at most** *2. Then S_n is word-representable* **iff** *S_n does* **not** *contain the graph T_2 and A_ℓ as induced subgraphs.*

Essentially, computer was **not** used to prove the following theorem.

Theorem (Kitaev, Long, Ma, Wu; 2017)

$S_n = (E_{n-4}, K_4)$ is w.-r. **iff** *it does* **not** *contain T_1–T_4 as ind. subgraphs.*

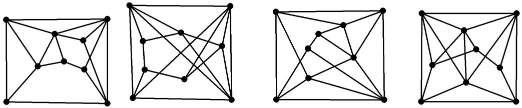
There is **only a computer-based proof** of the following theorem that still uses some theorems:

Theorem (Chen, Kitaev, Saito; 2018+)

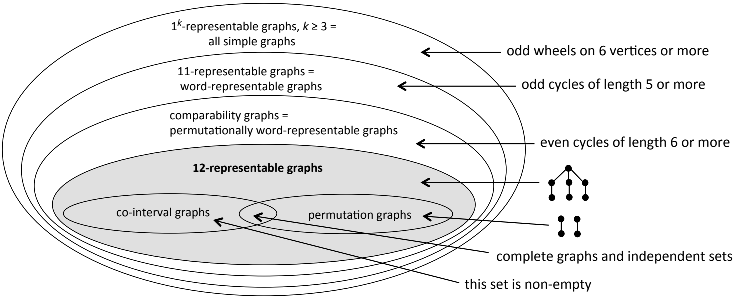
$S_n = (E_{n-5}, K_5)$ is w.-r. **iff** *S_n does* **not** *contain T_1–T_9 as ind. subgraphs.*

Stuff that should be included into this talk, but was not ...

- *k*-semi-transitive orientations; it was shown by computer that 3-**semi-transitively orientable**, but **non-semi-transitively orientable** graphs on 9 vertices exist;



- using computer to study 12-representable graphs.



Acknowledgment

Thank you for your attention!