Twin-width V: Linear Minors, Modular Counting, and Matrix Multiplication

Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Stéphan Thomassé

Université Grenoble Alpes, Laboratoire G-SCOP

STACS 2023, Hamburg

Twin-width of unordered graphs

Definition (Contraction sequence)

Contraction sequence of $G = (V, E)$: sequence of trigraphs $(G = G_n, G_{n-1}, \dots, G_1)$ where G_{i-1} is obtained by identifying two vertices of G_i.

Twin-width of unordered graphs

Definition (Contraction sequence)

Contraction sequence of $G = (V, E)$: sequence of trigraphs $(G = G_n, G_{n-1}, \dots, G_1)$ where G_{i-1} is obtained by identifying two vertices of G_i.
$V(G_i) \leftrightarrow$ partition of $V(G)$.

Twin-width of unordered graphs

Definition (Contraction sequence)

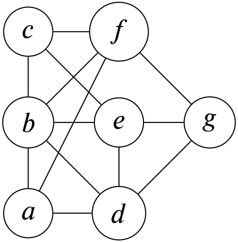
Contraction sequence of $G = (V, E)$: sequence of trigraphs $(G = G_n, G_{n-1}, \dots, G_1)$ where G_{i-1} is obtained by identifying two vertices of G_i.

$V(G_i) \leftrightarrow$ partition of $V(G)$.

For every $X, Y \in V(G_i)$ put:

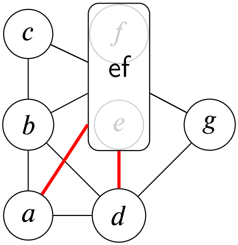
- An edge $XY \in E(G_i)$ if $G[X, Y]$ is a biclique;
- A nonedge in G_i if $G[X, Y]$ has no edge;
- A red edge $XY \in R(G_i)$ otherwise.

Twin-width of unordered graphs



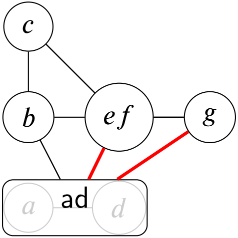
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width of unordered graphs

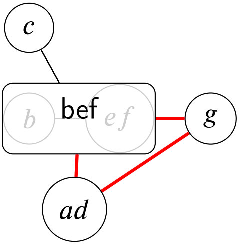


A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width of unordered graphs

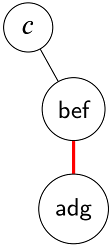


A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.



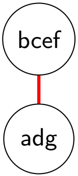
A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \dots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width of unordered graphs



A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width of unordered graphs



A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \dots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width of unordered graphs

A contraction sequence of G:
Sequence of trigraphs $G = G_n, G_{n-1}, \dots, G_2, G_1$ such that G_i is obtained by performing one contraction in G_{i+1}.

Twin-width of unordered graphs

Definition (Contraction sequence,twin-width)

Contraction sequence of $G = (V, E)$: sequence of trigraphs $(G = G_n, G_{n-1}, \ldots, G_1)$ where G_{i-1} is obtained by identifying two vertices of G_i.
$V(G_i) \leftrightarrow$ partition of $V(G)$.
For every $X, Y \in V(G_i)$ put:

- An edge if $G[X, Y]$ is a biclique;
- A nonedge if $G[X, Y]$ has no edge;
- A red edge otherwise.

$(G_i)_i$ has width at most d if every G_i has red degree at most d.
The twin-width of G is the minimum width a contraction sequence of G could have.

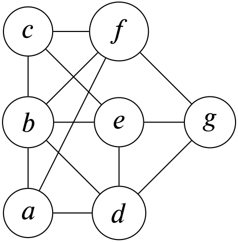
Examples and properties

- Cographs $\Leftrightarrow$ Graphs with twin-width 0;
- Trees have twin-width at most 2;
- [JP 22] Graphs of treewidth t have twin-width at most $3 \cdot 2^{t-1}$;
- [HJ 22] Planar graphs have twin-width at most 8;
- K_t-minor free graphs have twin-width $2^{2^{2^{\mathcal{O}(t)}}}$;
- Graphs with clique-width t have twin-width $\mathcal{O}(t)$;
- Permutation graphs G_σ such that σ avoids a pattern τ have twin-width $2^{\mathcal{O}(|\tau|)}$;
- ...

Twin-width of ordered structures

Graphs are given together with a total order on their vertices.
Rows and columns indices of ordered matrices are totally ordered.

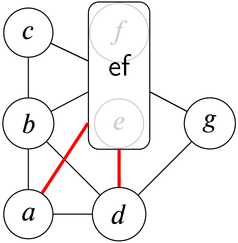
Twin-width of ordered structures



g	0	0	0	1	1	1	0
f	1	1	1	0	0	0	1
e	0	1	1	1	0	0	1
d	1	1	0	0	1	0	1
c	0	1	0	0	1	1	0
b	1	0	1	1	1	1	0
a	0	1	0	1	0	1	0
	a	b	c	d	e	f	g

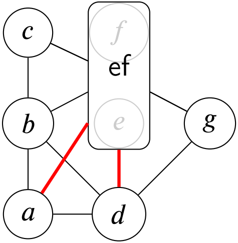
Left: Total order on $V(G)$: $a < b < c < d < e < f < g$. Right: the associated ordered adjacency matrix.

Twin-width of ordered structures



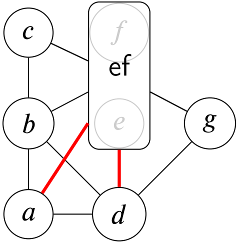
	a	b	c	d	e	f	g
g	0	0	0	1	1	1	0
f	1	1	1	0	0	0	1
e	0	1	1	1	0	0	1
d	1	1	0	0	1	0	1
c	0	1	0	0	1	1	0
b	1	0	1	1	1	1	0
a	0	1	0	1	0	1	0

Twin-width of ordered structures



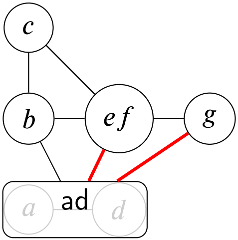
g	0	0	0	1	1	1	0
f	1	1	1	0	0	0	1
e	0	1	1	1	0	0	1
d	1	1	0	0	*r*	0	1
c	0	1	0	0	1	1	0
b	1	0	1	1	1	1	0
a	0	1	0	1	*r*	1	0
	a	b	c	d	e	f	g

Twin-width of ordered structures



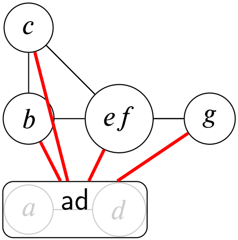
g	0	0	0	1	1	1	0
f	1	1	1	0	0	0	1
e	*r*	1	1	*r*	0	0	1
d	1	1	0	0	*r*	0	1
c	0	1	0	0	1	1	0
b	1	0	1	1	1	1	0
a	0	1	0	1	*r*	1	0
	a	b	c	d	e	f	g

Twin-width of ordered structures



g	r	0	0	1	1	1	0
f	1	1	1	0	0	0	1
e	r	1	1	r	0	0	1
d	r	1	0	0	r	0	1
c	0	1	0	0	1	1	0
b	1	0	1	1	1	1	0
a	0	1	0	1	r	1	0
	a	b	c	d	e	f	g

Twin-width of ordered structures



	a	b	c	d	e	f	g
g	r	0	0	1	1	1	0
f	1	1	1	0	0	0	1
e	r	1	1	r	0	0	1
d	r	1	0	0	r	0	1
c	0	1	0	0	1	1	0
b	1	0	1	1	1	1	0
a	0	1	0	1	r	1	0

Twin-width of ordered structures

Remark

A graph G has twin-width at most d if and only if there is a total ordering $<$ of $V(G)$ such that $(G,<)$ has twin-width at most d.

Algorithmic aspect of twin-width

Question (Fundamental question of twin-width)

Can we approximate twin-width?

Algorithmic aspect of twin-width

Question (Fundamental question of twin-width)

Can we approximate twin-width? i.e. Is there an algorithm taking d, G as input and returning in time $f(d) \cdot n^{\mathcal{O}(1)}$ either a "No" answer if G has twin-width more than d, or an $f(d)$-sequence otherwise?

Algorithmic aspect of twin-width

Question (Fundamental question of twin-width)

Can we approximate twin-width? i.e. Is there an algorithm taking d, G *as input and returning in time* $f(d) \cdot n^{\mathcal{O}(1)}$ *either a "No" answer if* G *has twin-width more than* d*, or an* $f(d)$*-sequence otherwise?*

Positive answer for every known "interesting family" of bounded twin-width.

Algorithmic aspect of twin-width

Question (Fundamental question of twin-width)

Can we approximate twin-width? i.e. Is there an algorithm taking d, G as input and returning in time $f(d) \cdot n^{\mathcal{O}(1)}$ either a "No" answer if G has twin-width more than d, or an $f(d)$-sequence otherwise?

Positive answer for every known "interesting family" of bounded twin-width.
→ True for classes of ordered graphs/matrices!

Theorem (BGOSTT '22 and BGOT '22)

There is an algorithm that, given an ordered graph $(G, <)$ and an integer d, returns in time $\mathcal{O}(f(d)n^2 \log(n))$:

- *"No" if $tww(G) > d$;*
- *a $g(d)$-sequence otherwise.*

Algorithmic aspect of twin-width

Question (Fundamental question of twin-width)

Can we approximate twin-width? i.e. Is there an algorithm taking d, G as input and returning in time $f(d) \cdot n^{\mathcal{O}(1)}$ either a "No" answer if G has twin-width more than d, or an $f(d)$-sequence otherwise?

Positive answer for every known "interesting family" of bounded twin-width.
→ True for classes of ordered graphs/matrices!

Theorem (BGOSTT '22 and BGOT '22)

There is an algorithm that, given an ordered graph $(G, <)$ and an integer d, returns in time $2^{2^{2^{2^{\mathcal{O}\left(d^2 \log(d)\right)}}}} n^2 \log(n)$:

- *"No" if $tww(G) > d$;*
- *a $2^{2^{2^{\mathcal{O}\left(d^4\right)}}}$-sequence otherwise.*

Linear Minors

Theorem (RS '86)

A class of graphs has bounded treewidth if and only its minor closure avoids the $k \times k$ grid for some $k \in \mathbb{N}$.

Linear Minors

Theorem (RS '86)

A class of graphs has bounded treewidth if and only its minor closure avoids the $k \times k$ grid for some $k \in \mathbb{N}$.

Definition (Linear Minor)

A matrix A is a Linear Minor of a matrix B if it can be obtained from B after the removal of some rows and replacing some pairs of consecutive rows or columns by a linear combination of them.

Linear Minors

Theorem (RS '86)

A class of graphs has bounded treewidth if and only its minor closure avoids the $k \times k$ grid for some $k \in \mathbb{N}$.

$$
B = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}
\qquad
\left[\begin{array}{cc|cccc|cc} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]
\qquad
A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}
$$

Linear Minors

Theorem (RS '86)

A class of graphs has bounded treewidth if and only its minor closure avoids the $k \times k$ grid for some $k \in \mathbb{N}$.

Definition (Linear Minor)

A matrix A is a Linear Minor of a matrix B if it can be obtained from B after the removal of some rows and replacing some pairs of consecutive rows or columns by a linear combination of them.

Theorem (Ordered case)

A class of matrices has bounded twin-width if and only if its linear minor closure avoids some matrix.

FO model checking on graphs

$\varphi \in \mathrm{FO}(E^{(2)})$: first order formula describing a graph problem.

FO model checking on graphs

$\varphi \in \mathrm{FO}(E^{(2)})$: first order formula describing a graph problem.

Example

$$\varphi := \exists x_1, \exists x_2, \dots, \exists x_k, \forall x, \left(\bigvee_{i=1}^{k} x = x_i \right) \vee \left(\bigvee_{i=1}^{k} E(x, x_i) \right)$$

corresponds to k-Dominating Set problem.

FO model checking on graphs

$\varphi \in \mathrm{FO}(E^{(2)})$: first order formula describing a graph problem.

Theorem (Bonnet, Kim, Thomassé, Watrigant '20)

There exists an algorithm that, given a graph G, a certificate that $tww(G) \leq d$ and a formula φ, decides whether $G \vDash \varphi$ in time $\mathcal{O}(f(d, |\varphi|) \cdot n)$.

Modular Counting

$\varphi \in \mathrm{FO} + \mathrm{MOD}(E^{(2)})$: first order formula describing a graph problem where we also allow existential quantifiers $\exists^{i[p]} x, \phi(x)$ expressing "there exists i mod p witnesses x for ϕ".

Modular Counting

$\varphi \in \mathrm{FO} + \mathrm{MOD}(E^{(2)})$: first order formula describing a graph problem where we also allow existential quantifiers $\exists^{i[p]} x, \phi(x)$ expressing "there exists i mod p witnesses x for ϕ".

Definition

G: graph. $G^{[2]}$: modular square of G, with same vertices and:

$$E(G^{[2]}) := \{uv \ : \ |N(u) \cap N(v)| = 1 \pmod 2\}.$$

Modular Counting

$\varphi \in \mathrm{FO} + \mathrm{MOD}(E^{(2)})$: first order formula describing a graph problem where we also allow existential quantifiers $\exists^{i[p]} x, \phi(x)$ expressing "there exists i mod p witnesses x for ϕ".

Example

$$\varphi := \exists x_1, \exists x_2, \ldots, \exists x_k, \forall x, \left(\bigvee_{i=1}^{k} x = x_i \right)$$

$$\vee \left(\bigvee_{i=1}^{k} E_{G^{[2]}}(x, x_i) \right)$$

"There exists a dominating set of size k in $G^{[2]}$".

Modular Counting

$\varphi \in \mathrm{FO} + \mathrm{MOD}(E^{(2)})$: first order formula describing a graph problem where we also allow existential quantifiers $\exists^{i[p]} x, \phi(x)$ expressing "there exists i mod p witnesses x for ϕ".

Example

$$\varphi := \exists x_1, \exists x_2, \dots, \exists x_k, \forall x, \left(\bigvee_{i=1}^{k} x = x_i \right) \vee \left(\bigvee_{i=1}^{k} \exists^{1[2]} y, E(x, y) \wedge E(y, x_i) \right)$$

"There exists a dominating set of size k in $G^{[2]}$".

Modular Counting

$\varphi \in \mathrm{FO} + \mathrm{MOD}(E^{(2)})$: first order formula describing a graph problem where we also allow existential quantifiers $\exists^{i[p]} x, \phi(x)$ expressing "there exists i mod p witnesses x for ϕ".

Theorem (BKTW 20, BGOT 22)

There exists an algorithm that, given a graph G, a certificate that $tww(G) \leq d$ and a FO+MOD formula φ, decides whether $G \models \varphi$ in time $\mathcal{O}(f(d, |\varphi|) \cdot n)$.

Matrix Multiplication

"Consequence" of Modular Counting+ Approximation algorithm:

Theorem

A, B $n \times n$ matrices over $\mathbb{F}_2$ of twin-width d.

- *Then AB has twin-width $f(d)$.*
- *There is a $\mathcal{O}_d(n^2 \log(n))$-time algorithm taking A, B as input and returning AB.*

Matrix Multiplication

"Consequence" of Modular Counting+ Approximation algorithm:

Theorem

A, B $n \times n$ matrices over $\mathbb{F}_2$ of twin-width d.

- *Then AB has twin-width $f(d)$.*
- *There is a $\mathcal{O}_d(n^2 \log(n))$-time algorithm taking A, B as input and returning AB.*

Completely unpractical

Matrix Multiplication

"Consequence" of Modular Counting+ Approximation algorithm:

Theorem

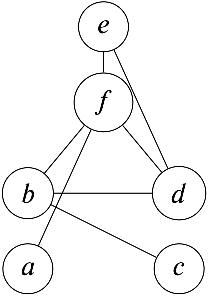
A, B $n \times n$ matrices over $\mathbb{F}_2$ of twin-width d.

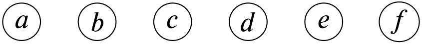
- *Then AB has twin-width $f(d)$.*
- *There is a $\mathcal{O}_d(n^2 \log(n))$-time algorithm taking A, B as input and returning AB.*

Completely unpractical

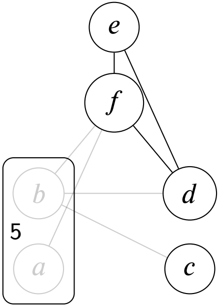
Our contribution: an ad-hoc algorithm for matrix multiplication.

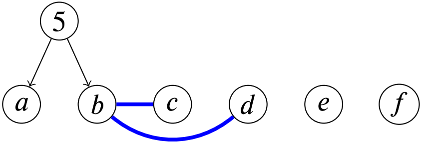
Twin-decompositions



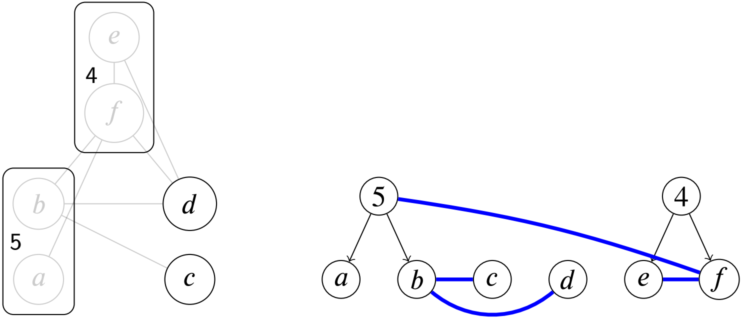


Twin-decompositions

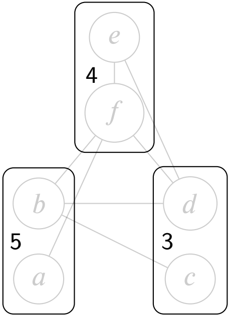


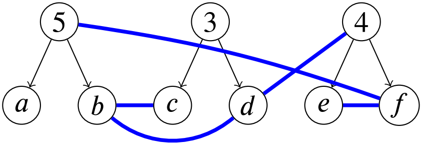


Twin-decompositions

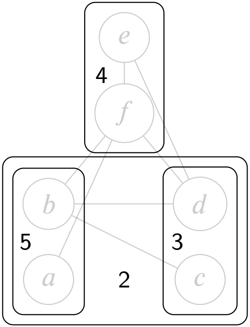


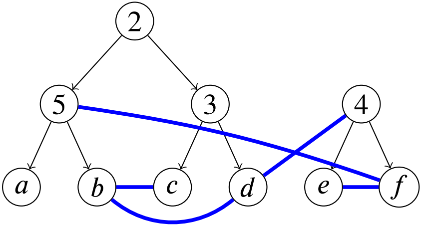
Twin-decompositions



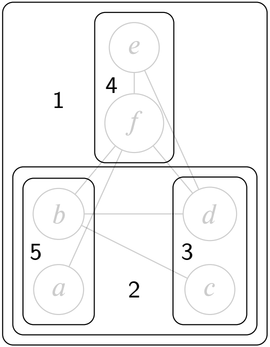


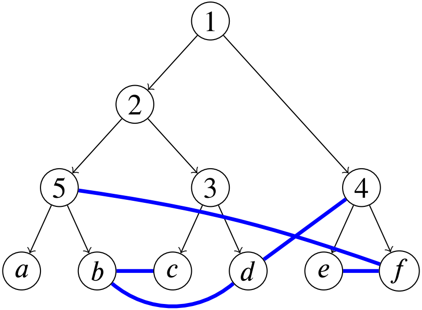
Twin-decompositions





Twin-decompositions





Twin-decompositions

Remark

- *If* $tww(G) \leq d$ *then* $|(\mathcal{T}, \mathcal{B})| = \mathcal{O}(nd)$.

Remark

- *If* $tww(G) \leq d$ *then* $|(\mathcal{T}, \mathcal{B})| = \mathcal{O}(nd)$.
- [*BGKTW '21*] *One can choose* $\mathcal{T}$ *with depth* $\mathcal{O}_d(\log(n))$.

Twin-decompositions

Remark

- *If* $tww(G) \leq d$ *then* $|(\mathcal{T}, \mathcal{B})| = \mathcal{O}(nd)$.
- [*BGKTW '21*] *One can choose* $\mathcal{T}$ *with depth* $\mathcal{O}_d(\log(n))$.
- [*PSZ '22*] *gain in query time; lose in space.*

Matrix Multiplication

$$\begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix} = \begin{pmatrix} AB & 0 \\ 0 & BA \end{pmatrix}$$

allows to reduce to the problem of squaring a matrix.

Matrix Multiplication

Theorem

There exists a $\mathcal{O}(d^2 4^d n)$-time algorithm that, given a twin-decomposition $(\mathcal{T}, \mathcal{B})$ of width d of A, outputs a twin-decomposition of width $\mathcal{O}(d^2 2^d)$ of A^2.

→ Extends to a FPT-algorithm for matrix multiplication over $\mathbb{F}_2$ with same complexity.

Matrix Multiplication

Theorem

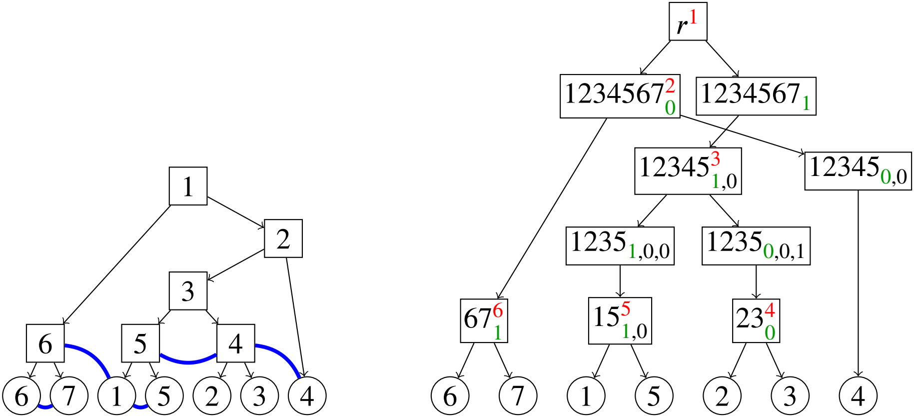
There exists a $\mathcal{O}(d^2 4^d n)$-time algorithm that, given a twin-decomposition $(\mathcal{T}, \mathcal{B})$ of width d of A, outputs a twin-decomposition of width $\mathcal{O}(d^2 2^d)$ of A^2.

→ Extends to a FPT-algorithm for matrix multiplication over $\mathbb{F}_2$ with same complexity.

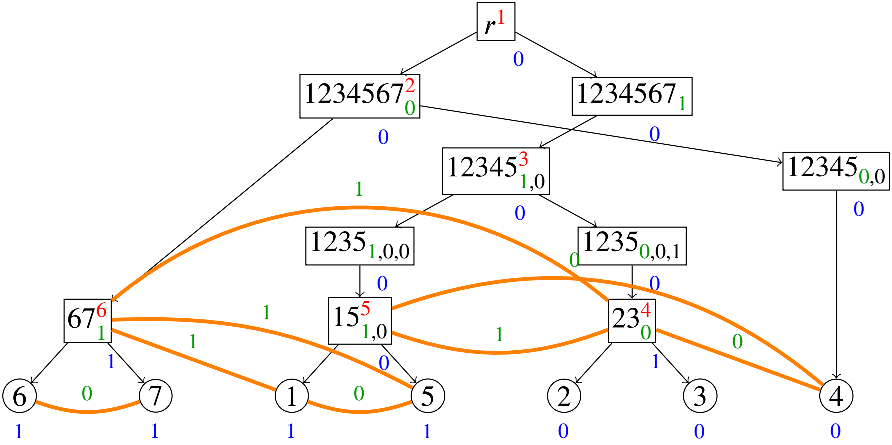
→ Extends over $\mathbb{F}_q$ for q: prime power.

Danke

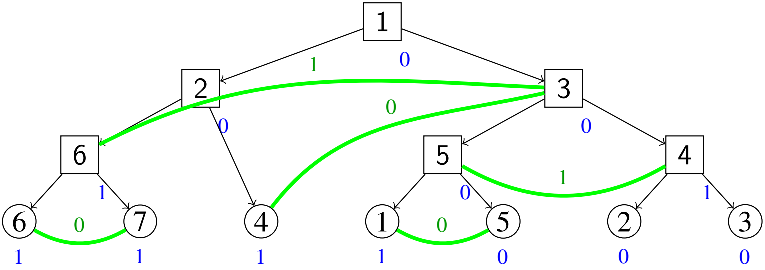
First step: the shape of the tree



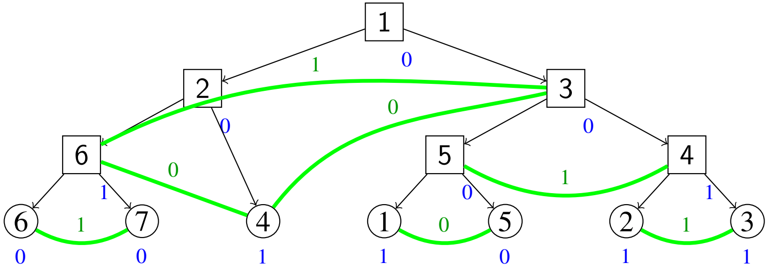
Second step: computation of labelled edges and labelling vertices



Last step: from orange to green edges



Last step: from orange to green edges



Last step: from orange to green edges