

MONOIDAL WIDTH

Elena Di Lavoro

joint work with Paweł Sobociński

Tallinn University of Technology

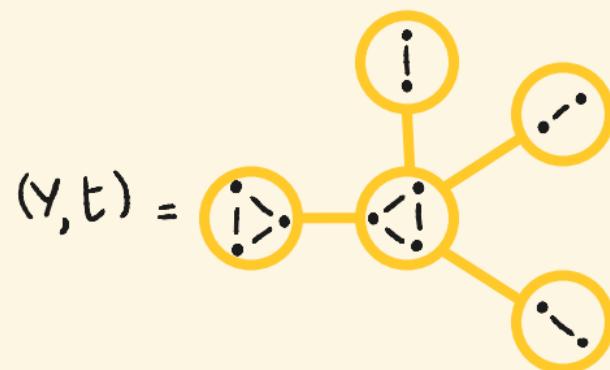
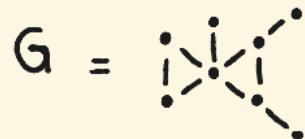


FIXED-PARAMETER TRACTABILITY

Some problems might be easier to solve on structurally "simple" inputs.

THEOREM (Courcelle 1990)

Every property expressible in the monadic second order logic of graphs can be verified in linear time on graphs of bounded tree width.



OVERVIEW

- Monoidal categories give process theories.
- Study fixed-parameter tractability of problems on morphisms in monoidal categories.
- Introduce monoidal width to measure structural complexity in monoidal categories.
- Capture tree width and rank width.

STRING DIAGRAMS

\mathcal{C} symmetric monoidal category

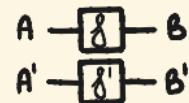
$f: A \rightarrow B, g: B \rightarrow C$ in \mathcal{C}

- composition $f; g: A \rightarrow C$

$f: A \rightarrow B, f': A' \rightarrow B'$ in \mathcal{C}

- monoidal product $f \otimes f': A \otimes A' \rightarrow B \otimes B'$

- symmetry $\sigma_{A,B}: A \otimes B \rightarrow B \otimes A$



$$A \xrightarrow{f} B \quad A' \xrightarrow{f'} B' = A \xrightarrow{\sigma} B \quad A' \xrightarrow{\sigma} B'$$

(naturality)

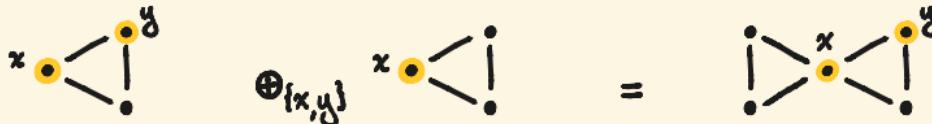
OUTLINE

- [• Monoidal decompositions]
- Matrices
- Rank width
- Branch width
- Fixed-parameter tractability

TREE DECOMPOSITIONS

Operation \oplus_x on graphs with sources:

$G \oplus_x H$ glues G and H along the sources in X .



A tree decomposition of G is a term for G , where operations are gluing along sources \oplus_x , deletion of sources ε_x and an edge generator e_{xy} .

$$\begin{aligned} \text{Graph with sources } x, y &= \varepsilon_{\{x\}}((e_{xy} \oplus_{\{x,y,2\}} e_{yz}) \oplus_{\{x,2\}} e_{xz}) \\ &\quad \oplus_{\{x,y\}} \varepsilon_{\{y,2\}}((e_{xy} \oplus_{\{x,y,2\}} e_{yz}) \oplus_{\{x,2\}} e_{xz}) \end{aligned}$$

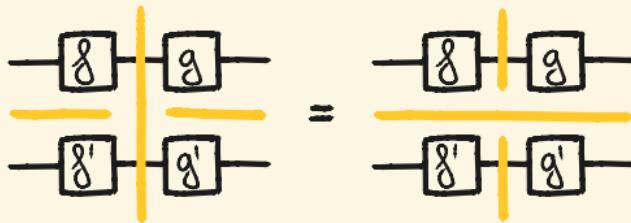
[Robertson & Seymour 1983, Courcelle 1990]

DECOMPOSING MORPHISMS IN MONOIDAL CATEGORIES

There are two operations in monoidal categories :

- composition ; \circ \rightsquigarrow resource sharing, synchronisation
 \Rightarrow COSTLY
- monoidal product \otimes \rightsquigarrow processes side-by-side
 \Rightarrow CHEAP

$$(\delta \otimes \delta') ; (g \otimes g') = (\delta; g) \otimes (\delta'; g')$$

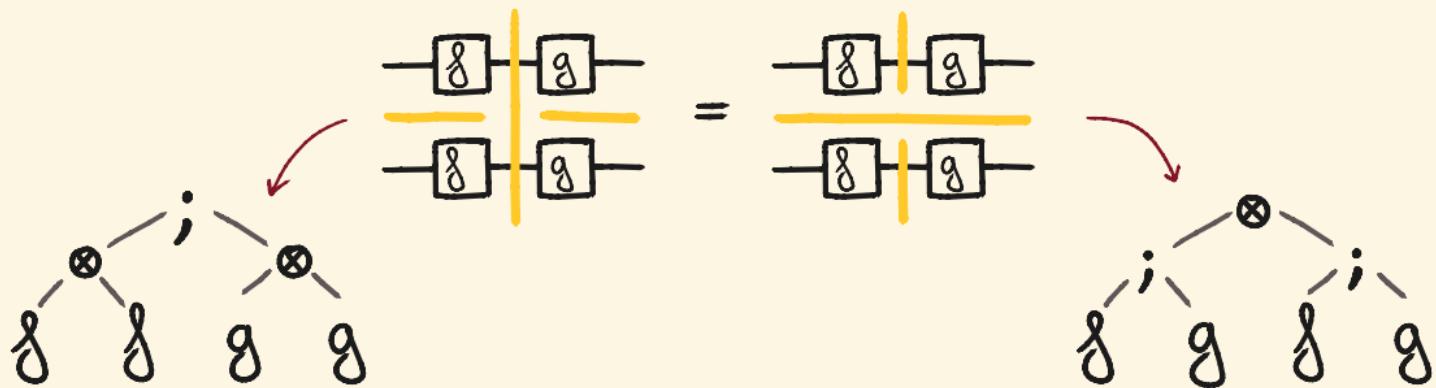


MONOIDAL DECOMPOSITIONS

A monoidal decomposition $d \in \mathcal{D}_g$ of $f: X \rightarrow Y$ is

$$d ::= (f)$$

$$\begin{cases} d_1 \text{ jc } d_2 & \text{if } f = f_1 \text{ jc } f_2, d_1 \in \mathcal{D}_{f_1}, d_2 \in \mathcal{D}_{f_2} \\ d_1 \otimes d_2 & \text{if } f = f_1 \otimes f_2, d_1 \in \mathcal{D}_{f_1}, d_2 \in \mathcal{D}_{f_2} \end{cases}$$



MONOIDAL WIDTH

WEIGHT FUNCTION

$w: \text{morphel} \rightarrow \mathbb{N}$ such that

- $w(f;_y g) + w(\mathbb{1}_y) \geq w(f) + w(g)$
- $w(f \otimes g) = w(f) + w(g)$

WIDTH OF A DECOMPOSITION

\rightsquigarrow cost of the most expensive operation

$$wd(d) := w(f)$$

$$d = (f)$$

$$\mid \max\{wd(d_1), w(\mathbb{1}_y), wd(d_2)\}$$

$$d = d'_1 ;_y d_2$$

$$\mid \max\{wd(d_1), wd(d_2)\}$$

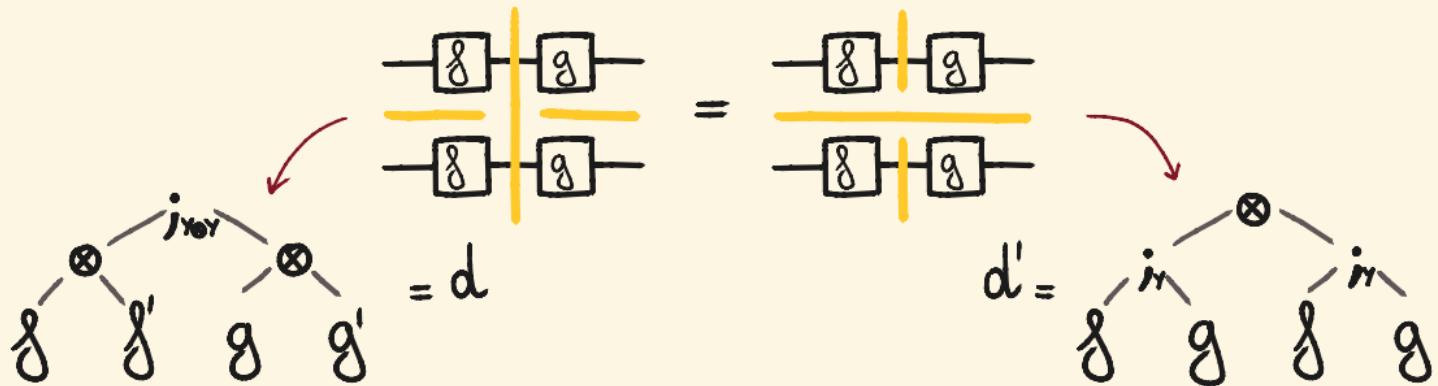
$$d = d'_1 \otimes d_2$$

MONOIDAL WIDTH

$$mwd(f) := \min_{d \in \mathcal{D}_f} wd(d)$$

\rightsquigarrow cost of a cheapest decomposition

MONOIDAL WIDTH INCENTIVISES PARALLELISM



$$\text{wd}(d) = \max\{w(f), w(g), 2 \cdot w(\mathbb{1}_Y)\} \geq \max\{w(f), w(g), w(\mathbb{1}_Y)\} = \text{wd}(d')$$

OUTLINE

- Monoidal decompositions
- Matrices
- Rank width
- Branch width
- Fixed-parameter tractability

]

BIALGEBRA : THE PROP OF N-MATRICES



COCOMMUTATIVE COMONOID

Three graphical equations for a cocommutative comonoid:

- $\text{loop with dot} = \text{loop with dot}$
- $\text{line with dot} = \text{line with dot}$
- $\text{loop with dot and line} = \text{line with dot}$

COMMUTATIVE MONOID

Three graphical equations for a commutative monoid:

- $\text{loop with dot and line} = \text{loop with line and dot}$
- $\text{line with dot} = \text{line with dot}$
- $\text{loop with dot and line} = \text{line with dot}$

BIALGEBRA

A graphical equation for a bialgebra:

$$\text{loop with dot and line} = \text{line with dot}$$

A graphical equation for a bialgebra:

$$\text{loop with two dots and line} = \text{line with two dots}$$

A graphical equation for a bialgebra:

$$\text{dot with line} = \text{dot}$$

A graphical equation for a bialgebra:

$$\text{line with dot} = \square$$

[Zanasi 2015]

PROP OF MATRICES - EXAMPLE

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \quad \begin{array}{c} 2 \\ 3 \\ 3 \\ 4 \end{array}$$

Annotations: Red arrows point to the first two columns of the matrix. A red circle highlights the value 1 at row 3, column 2. A blue circle highlights the value 2 at row 4, column 3. To the right, the matrix is shown as a directed graph with four nodes labeled 2, 3, 3, and 4. Directed edges connect node 2 to 3, 3 to 3, 3 to 4, and 4 to 3.

FACT : the minimal vertical cut in a matrix
is its rank : $\min \{ k \in \mathbb{N} \mid A = B_{jk} C \} = \text{rank } A$

$$\text{rank } A = 2 \rightsquigarrow \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \quad \begin{array}{c} 2 \\ \text{---} \\ \text{---} \\ \text{---} \end{array}$$

Annotation: A yellow vertical line is drawn through the second column of the matrix, indicating the minimal vertical cut.

PROP OF MATRICES - EXAMPLE

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} 2 \\ 3 \\ 3 \\ 4 \end{array}$$

Diagram illustrating the matrix A as a directed graph. The columns are labeled 2, 3, and 4 from left to right. The rows are labeled 1, 2, 3, and 4 from top to bottom. Edges connect nodes between adjacent columns. Red arrows point from row 1 to column 2 and from row 3 to column 3. Blue arrows point from row 4 to column 3 and from row 4 to column 4. Nodes are represented by small circles with outgoing edges.

FACT : the minimal vertical cut in a matrix
is its rank : $\min \{ k \in \mathbb{N} \mid A = B_{jk} C \} = \text{rank } A$

$$\text{rank } A = 2 \rightsquigarrow \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} 2 \\ 3 \\ 3 \\ 4 \end{array}$$

Diagram illustrating the rank of matrix A as a directed graph. A vertical yellow line labeled '2' passes through the first two columns. Nodes are represented by small circles with outgoing edges. The graph shows that the first two columns are linearly independent, while the third and fourth columns are dependent on them.

THE WIDTH OF NATURAL NUMBERS

$$\begin{aligned} w : \text{morph}(\text{Bialg}) &\rightarrow \mathbb{N} \\ f : m \rightarrow m &\mapsto \max\{m, n\} \end{aligned}$$

→ $w(\square) = 2$

→ $w(\circ) = 2$

LEMMA

$$\text{mwod}((n)) \leq 2$$

ex



$$\text{wd}\left(\text{---} \bullet \text{---} \square \text{---} \bullet \text{---} \right) = 4$$

$$\text{wd}\left(\text{---} \bullet \text{---} \square \text{---} \bullet \text{---} \right) = 2$$

THE WIDTH OF MATRICES

Matrices can be written in blocks:

$$A = \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & A_b \end{pmatrix} = A_1 \oplus A_2 \oplus \cdots \oplus A_b$$

ex $A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \text{[Diagram]} = j \oplus \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \oplus (2)$

THEOREM

$$\max_i \text{rank } A_i \leq \text{mwd } A \leq \max_i \text{rank } A_i + 1$$

ex $\text{wd} \left(\text{[Diagram]} \right) = \max \{ n(j), n(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}), n(2) \} + 1 = 2$

OUTLINE

- Monoidal decompositions
- Matrices
- Rank width
- Branch width
- Fixed-parameter tractability

]

A PROP OF GRAPHS

vertex
generator



bialgebra equations +

$$\text{cup} = \text{G}$$

$$\text{cap} = \text{o}$$

~> the cup transposes

$$[G] = [G^T]$$

and captures equivalence of adjacency matrices

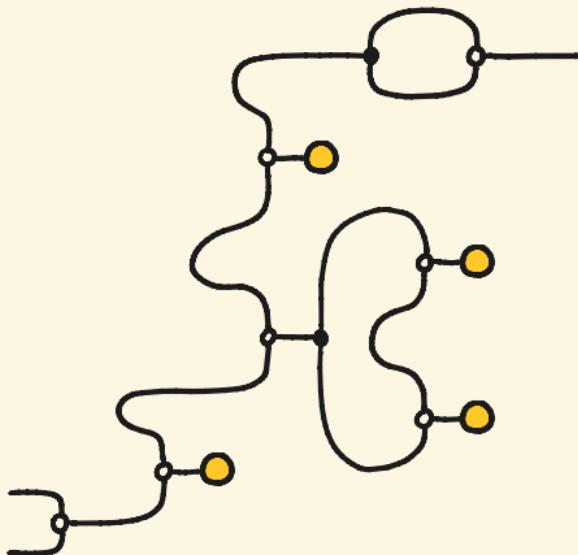
$$[G] = [H] \Leftrightarrow [G] = [H]$$

[Di Stefano, Hedges & Sobociński 2021]

GRAPHS AS MORPHISMS - EXAMPLE



graph on k vertices
given by the adjacency
matrix $[G]$



RANK WIDTH [Oum & Seymour, 2006]

G undirected graph

RANK DECOMPOSITION

(Y, π) where

- Y is a subcubic tree ($=$ any node has at most 3 neighbours)
- $\pi : \text{leaves}(Y) \xrightarrow{\cong} \text{vertices}(G)$ labelling bijection

WIDTH OF (Y, π)

$$\text{wd}(Y, \pi) := \max_{e \in \text{edges } Y} \text{rank}(X_e)$$

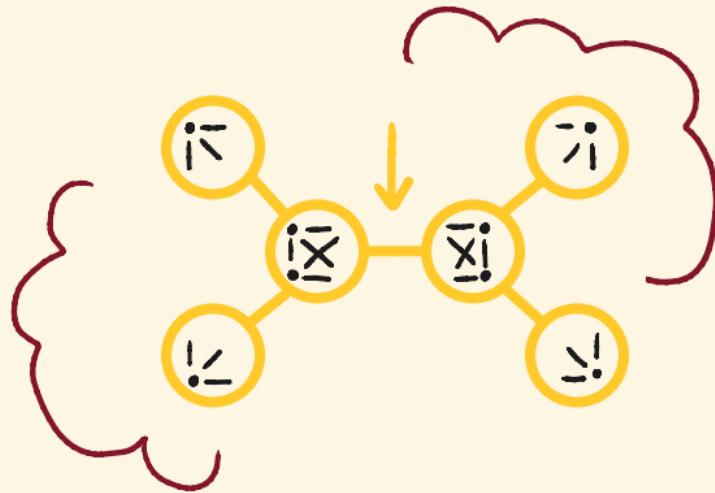
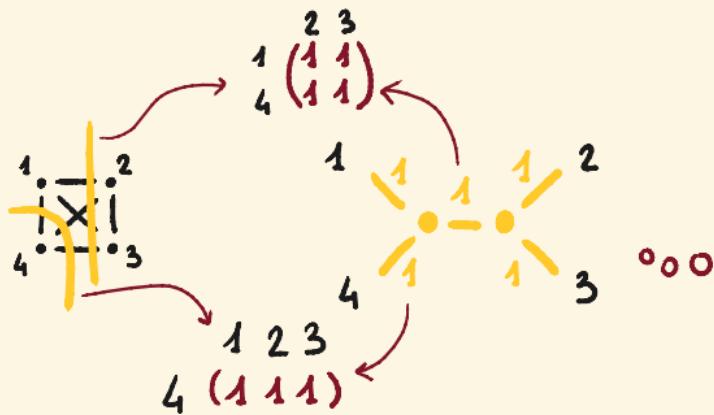
$\curvearrowleft X_e$ adjacency matrix
of the cut given
by e through π

RANK WIDTH

$$\text{rwd}(G) := \min_{(Y, \pi)} \text{wd}(Y, \pi) \quad \Rightarrow \text{cost of a cheapest decomposition}$$

RANK WIDTH - EXAMPLE

$$G = \begin{array}{c} 1 & & 2 \\ | & \diagup & | \\ 4 & X & 3 \\ | & \diagdown & | \\ 4 & & 3 \end{array}$$



$$\text{rwd}(G) = 1$$

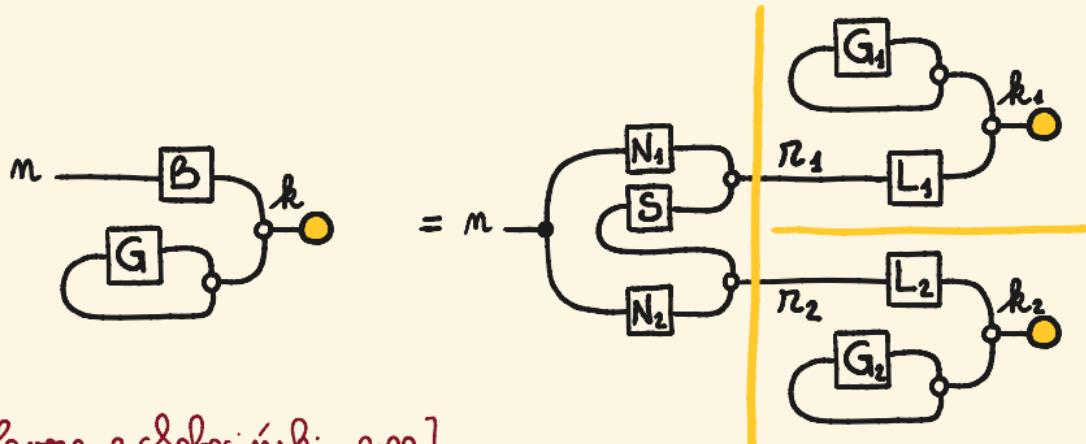
RANK WIDTH & MONOIDAL WIDTH

[G] undirected graph

$$g = \text{graph } G : 0 \rightarrow 0 \text{ in clgraph}$$

THEOREM

$$\frac{1}{2} \text{rwd}(G) \leq \text{mwd}(g) \leq 2 \text{rwd}(G)$$



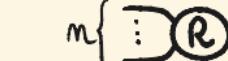
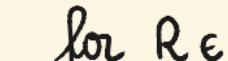
[Di Stefano & Sobociński 2022]

OUTLINE

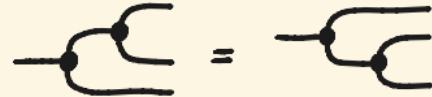
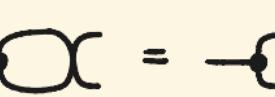
- Monoidal decompositions
- Matrices
- Rank width
- Branch width
- Fixed-parameter tractability

]

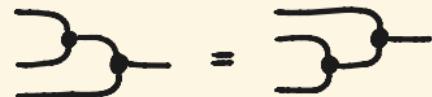
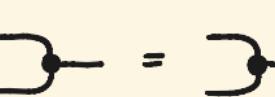
FROBENIUS : A PROP OF τ -STRUCTURES

 \rightarrow  \rightarrow  $\vdash \{ \text{DR} \}$ for $R \in \tau$ of arity n

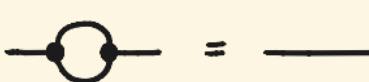
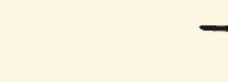
COCOMMUTATIVE COMONOID

 $=$   $=$  

COMMUTATIVE MONOID

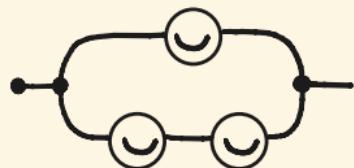
 $=$   $=$  

FROBENIUS

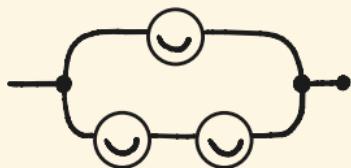
 $=$   $=$ 

GRAPHS AS MORPHISMS - EXAMPLE

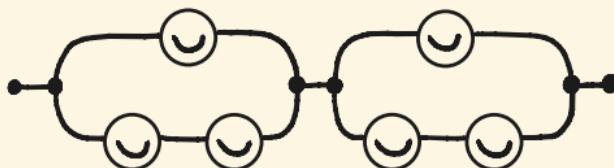
graphs with sources are τ -structures with $\tau = \{-\circlearrowleft\}$



\rightsquigarrow



\rightsquigarrow



\rightsquigarrow



BRANCH WIDTH [Robertson & Seymour, 1991]

G undirected graph

BRANCH DECOMPOSITION

(Y, b) where

- Y : is a subcubic tree ($=$ any node has at most 3 neighbours)
- b : $\text{leaves}(Y) \xrightarrow{\cong} \text{edges}(G)$ labelling bijection

WIDTH OF (Y, b)

$$\text{wd}(Y, b) := \max_{e \in \text{edges } Y} |\text{ends } A_e \cap \text{ends } B_e| \quad \begin{matrix} \curvearrowright \\ \{A_e, B_e\} \text{ partition} \\ \text{of } E \text{ given by} \\ e \text{ through } b \end{matrix}$$

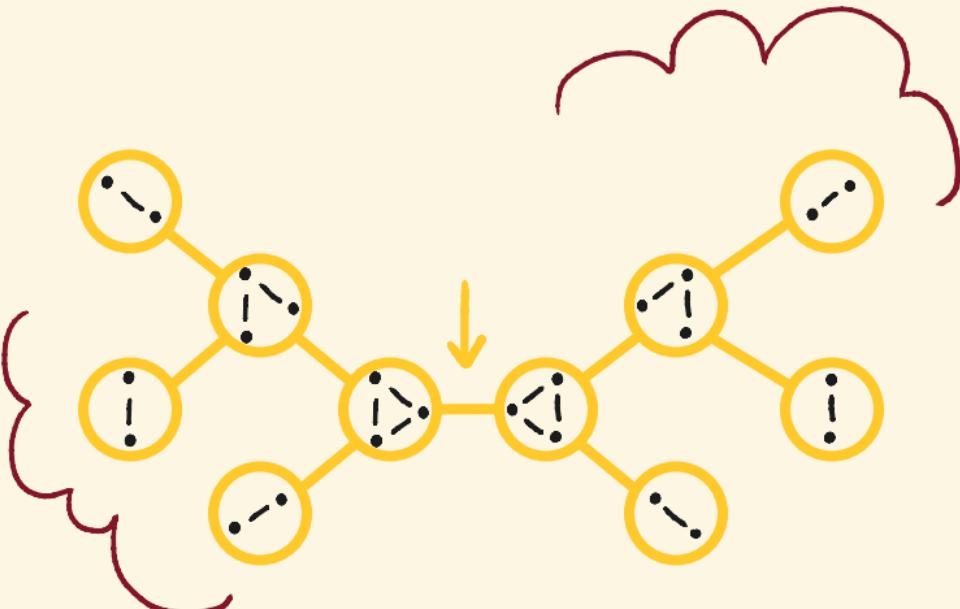
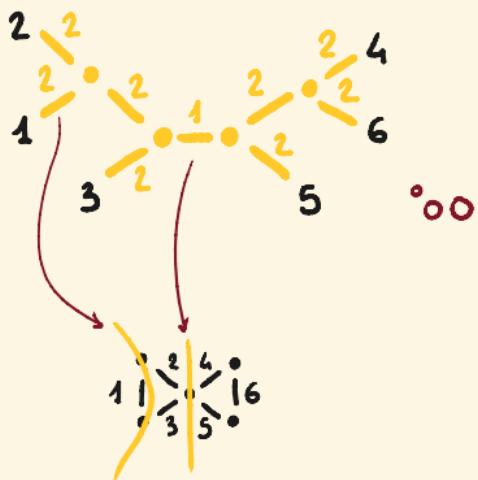
BRANCH WIDTH

$$\text{bw}(G) := \min_{(Y, b)} \text{wd}(Y, b) \rightsquigarrow \text{cost of a cheapest decomposition}$$

BRANCH WIDTH - EXAMPLE

$$G = \begin{array}{c} 1 \\ | \\ 1 \end{array} \begin{array}{c} 2 \\ | \\ 2 \end{array} \begin{array}{c} 4 \\ | \\ 4 \end{array} \begin{array}{c} 6 \\ | \\ 6 \end{array}$$

\begin{array}{c} \bullet \\ | \\ \bullet \\ 3 \\ | \\ 5 \\ | \\ \bullet \end{array}



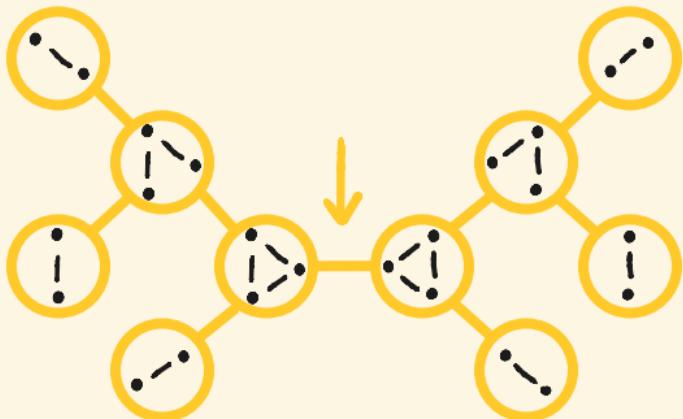
BRANCH WIDTH & MONOIDAL WIDTH

$G = (V, E)$ undirected graph

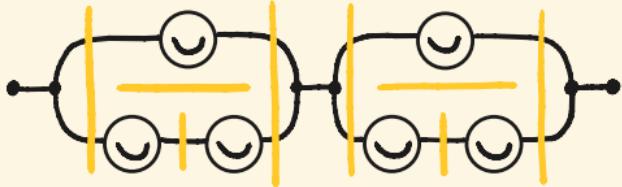
$g = \bigoplus_{\emptyset \neq S \subseteq V} G \cap_S : \emptyset \rightarrow \emptyset$ in $\text{clospan}(\text{Ugraph})_\circ$

THEOREM

$$\frac{1}{2} \text{bwd}(G) \leq \text{mwd}(g) \leq \text{bwd}(G) + 1$$



\rightsquigarrow



[Di Stefano & Sobociński 2022]

OUTLINE

- Monoidal decompositions
- Matrices
- Rank width
- Branch width
- Fixed-parameter tractability

COMPOSITIONAL ALGORITHMS

\mathcal{C}, \mathcal{D} monoidal categories

$P: \mathcal{C} \rightarrow \mathcal{D}$ monoidal functor

space of solutions

$w: \text{morph } \mathcal{C} \rightarrow \mathbb{N}$ weight function

a compositional algorithm for P wrt. w computes

1. $P(g)$ in time $\Theta(c(w(g)) \cdot w(g))$

2. $P(g);_v P(g)$ in \mathcal{D} in time $\Theta(c(w(1_y)) \cdot (w(g) + w(g)))$

3. $P(g) \otimes P(g')$ in \mathcal{D} in time $\Theta(c(1) \cdot (w(g) + w(g')))$

for some function $c: \mathbb{N} \rightarrow \mathbb{N}$.

usually more than exponential

[cf. Courcelle & Makowsky 2002]

cf. msol-smooth operations and msol-inductive classes of τ -structures

cf. number of vertices

FEFERMAN-VAUGHT THEOREM

THEOREM

For τ -structures A, B, A', B' and a set X of sources,
if $A \equiv_{\text{MSO}(\tau)} A'$ and $B \equiv_{\text{MSO}(\tau)} B'$, then $A \oplus_X B \equiv_{\text{MSO}(\tau)} A' \oplus_X B'$.
Computing $A \oplus_X B \models \varphi$ given $\text{Th}_{\text{MSO}_q(\tau)}(A)$ and $\text{Th}_{\text{MSO}_q(\tau)}(B)$
does not depend on $A \oplus_X B$.

COROLLARY

There is a monoidal functor

$$\text{Th}: \text{Struct}(\tau) \rightarrow \text{Struct}(\tau)/_{\equiv_{\text{MSO}(\tau)}} \\ A \mapsto \text{Th}_{\text{MSO}_q(\tau)}(A)$$

that can be computed with a compositional algorithm.

[Feferman & Vaught 1959, Courcelle & Makowsky 2002]

MONOIDAL FIXED-PARAMETER TRACTABILITY

THEOREM

computing a functional problem $P: \mathcal{C} \rightarrow \mathcal{D}$ with a compositional algorithm w.r.t. $w: \text{morph } \mathcal{C} \rightarrow \mathbb{N}$ is fixed-parameter tractable with parameter monoidal width.

Explicitly, if $\text{mwd}(g) \leq k$, computing $P(g)$ takes $O(c(k) \cdot w(g))$, for some $c: \mathbb{N} \rightarrow \mathbb{N}$.

COROLLARY

Checking an MSO formula $\varphi \in \text{MSO}_q$ on τ -structures is fixed-parameter tractable with parameter tree width.

SUMMARY & FUTURE DIRECTIONS

- Monoidal width measures structural complexity of morphisms in monoidal categories.
- Monoidal width captures rank width and tree width.
- We would like to find other examples of fixed-parameter tractability more in the spirit of morphisms as processes.