Petr Hliněný

Approaching Tree-Width of Graphs from a Matroidal Perspective

Faculty of Informatics,
Masaryk University in Brno,
Botanická 68a, 602 00 Brno, Czech Rep.

e-mail: hlineny@fi.muni.cz
http://www.fi.muni.cz/~hlineny

Based on joint work with **Geoff Whittle**

Victoria University of Wellington

Contents

1 TREE-WIDTH - an Overview

- Introduced [Robertson & Seymour, 80's] — the Graph minors project.

1 TREE-WIDTH - an Overview

- Introduced [Robertson & Seymour, 80's] — the Graph minors project.

Definition: A *tree-decomposition* of a graph G is a tree with

- "bags" (subsets) of vertices at the tree nodes,
- each edge of G belongs to some bag, and
- the bags containing some vertex form a subtree (interpolation).

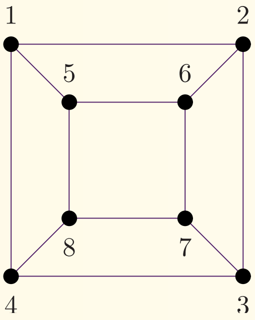
1 TREE-WIDTH - an Overview

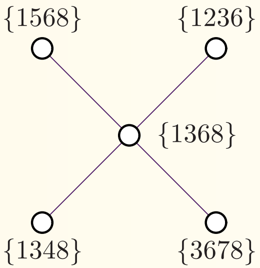
- Introduced [Robertson & Seymour, 80's] — the Graph minors project.

Definition: A *tree-decomposition* of a graph G is a tree with

- "bags" (subsets) of vertices at the tree nodes,
- each edge of G belongs to some bag, and
- the bags containing some vertex form a subtree (interpolation).

Tree-width $= \mathbf{min}_{\text{decompositions of } \boldsymbol{G}}\ \mathbf{max}\,\{|\boldsymbol{B}| - \mathbf{1} : \boldsymbol{B} \text{ bag in decomp.}\}$



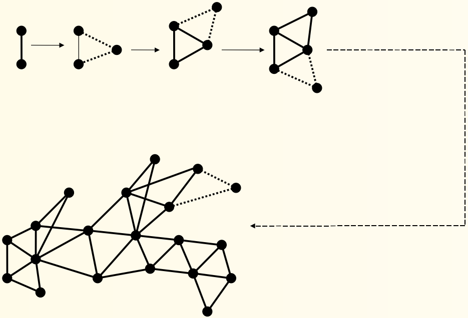


Alternative traditional definition

- The tree-width of G equals the smallest possible clique number minus one of a chordal supergraph of G.

Alternative traditional definition

- The tree-width of G equals the smallest possible clique number minus one of a chordal supergraph of G.
- This can be much easier understood via *k-trees*, see e.g. a 2-tree:



[Beineke & Pippert, 68 – 69], [Rose 74], [Arnborg & Proskurowski, 86].

- A graph G has tree-width $\leq k$ iff G is a partial (subgraph of a) k-tree.

Where is tree-width useful?

- Already the fact that independent approaches to tree-width evolved in time, suggests that it likely is an interesting and useful notion. . .

Where is tree-width useful?

- Already the fact that independent approaches to tree-width evolved in time, suggests that it likely is an interesting and useful notion. . .
- The profound Graph minors project makes an essential use of tree-width.

Where is tree-width useful?

- Already the fact that independent approaches to tree-width evolved in time, suggests that it likely is an interesting and useful notion. . .
- The profound Graph minors project makes an essential use of tree-width.
- *Parameterized* algorithmics:
  - Initial algorithmic attempts [Arnborg & Proskurowski, 86], [Arnborg, Corneil & Proskurowski, 87], [Bodlaender 88].

Where is tree-width useful?

- Already the fact that independent approaches to tree-width evolved in time, suggests that it likely is an interesting and useful notion...
- The profound Graph minors project makes an essential use of tree-width.
- *Parameterized* algorithmics:
  - Initial algorithmic attempts [Arnborg & Proskurowski, 86], [Arnborg, Corneil & Proskurowski, 87], [Bodlaender 88].
  - All graph properties expressible in *MSO logic* are efficiently solvable on the graphs of bounded tree-width (incl. many NP-hard ones). [Courcelle 88], [Arnborg, Lagergren & Seese, 88]

Where is tree-width useful?

- Already the fact that independent approaches to tree-width evolved in time, suggests that it likely is an interesting and useful notion...

- The profound Graph minors project makes an essential use of tree-width.

- *Parameterized* algorithmics:

  – Initial algorithmic attempts [Arnborg & Proskurowski, 86], [Arnborg, Corneil & Proskurowski, 87], [Bodlaender 88].

  – All graph properties expressible in *MSO logic* are efficiently solvable on the graphs of bounded tree-width (incl. many NP-hard ones). [Courcelle 88], [Arnborg, Lagergren & Seese, 88]

  – Linear-time parameterized algorithm for a tree-decomposition by [Bodlaender 96].

Where is tree-width useful?

- Already the fact that independent approaches to tree-width evolved in time, suggests that it likely is an interesting and useful notion. . .
- The profound Graph minors project makes an essential use of tree-width.
- *Parameterized* algorithmics:
  - Initial algorithmic attempts [Arnborg & Proskurowski, 86], [Arnborg, Corneil & Proskurowski, 87], [Bodlaender 88].
  - All graph properties expressible in *MSO logic* are efficiently solvable on the graphs of bounded tree-width (incl. many NP-hard ones). [Courcelle 88], [Arnborg, Lagergren & Seese, 88]
  - Linear-time parameterized algorithm for a tree-decomposition by [Bodlaender 96].
- Logic side:

  Decidability of *MSO theories* of the graphs of bounded tree-width [Courcelle 88]; a converse by [Seese 91].

2 "Vertex-free" Tree-Decompositions

Motivation: All the "traditional" definitions of tree-width make an essential use of graph vertices. Is this necessary?

2 “Vertex-free” Tree-Decompositions

Motivation: All the “traditional” definitions of tree-width make an essential use of graph vertices. Is this necessary?

- A new (matroidal) approach, proposed by [PH & Whittle, 03].

Definition: A *VF tree-decomposition* of a graph G is a tree T with

  – an arbitrary $\tau : E(G) \rightarrow V(T)$, without further restrictions.

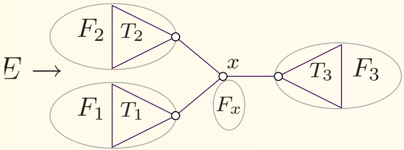
2 “Vertex-free” Tree-Decompositions

Motivation: All the “traditional” definitions of tree-width make an essential use of graph vertices. Is this necessary?

- A new (matroidal) approach, proposed by [PH & Whittle, 03].

Definition: A *VF tree-decomposition* of a graph G is a tree T with

– an arbitrary $\tau : E(G) \to V(T)$, without further restrictions.



– *Node with of* $x = |V(G)| + (d-1)\cdot c(G) - \sum_{i=1}^{d} c(G - F_i)$,
where F_i are the edges mapped to the subtrees $T - x$,
and $c()$ denotes the number of components.

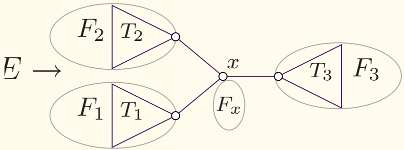
2 "Vertex-free" Tree-Decompositions

Motivation: All the "traditional" definitions of tree-width make an essential use of graph vertices. Is this necessary?

- A new (matroidal) approach, proposed by [PH & Whittle, 03].

Definition: A *VF tree-decomposition* of a graph G is a tree T with

– an arbitrary $\tau : E(G) \to V(T)$, without further restrictions.



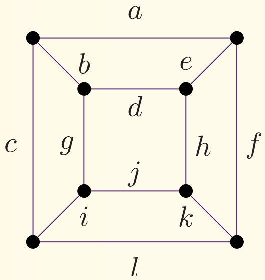
– *Node with of* $x = |V(G)| + (d-1) \cdot c(G) - \sum_{i=1}^{d} c(G - F_i)$,
where F_i are the edges mapped to the subtrees $T - x$,
and $c()$ denotes the number of components.

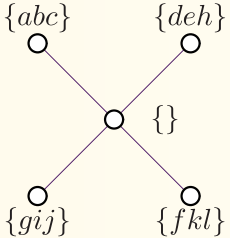
VF Tree-width $= \min_{\text{decompositions of } G} \max \{$ *node-width* in decomp. $\}$.

Are these two parameters really the same?

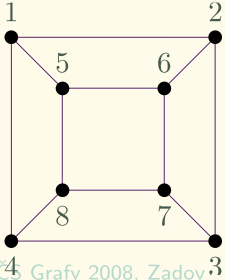
Are these two parameters really the same?

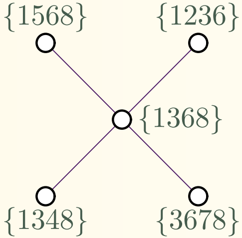
Check the following examples for an illustration...





node-with of $x = |V(G)| + (d-1)\cdot c(G) - \sum_{i=1}^{d} c(G - F_i)$





Where this idea comes from?

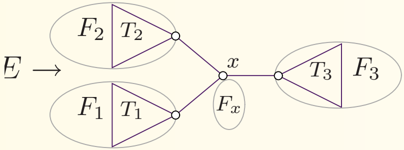
- A general definition of matroid tree-width proposed by [PH & Whittle, 03], following unpublished [Geelen].

Where this idea comes from?

- A general definition of matroid tree-width proposed by [PH & Whittle, 03], following unpublished [Geelen].

Definition: A *tree-decomposition* of a matroid M is a tree T with

– an arbitrary $\tau : E(M) \to V(T)$, without further restrictions.



– *Node with of* $x = \sum_{i=1}^{d} \mathrm{r}(M \setminus F_i) - (d-1) \cdot \mathrm{r}(M)$,

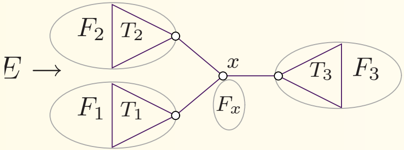
where $\mathrm{r}()$ denotes the matroid *rank* ("dimension").

Where this idea comes from?

- A general definition of matroid tree-width proposed by [PH & Whittle, 03], following unpublished [Geelen].

Definition: A *tree-decomposition* of a matroid M is a tree T with

– an arbitrary $\tau : E(M) \to V(T)$, without further restrictions.



– *Node with of* $x = \sum_{i=1}^{d} \mathrm{r}(M \setminus F_i) - (d-1) \cdot \mathrm{r}(M)$,

where $\mathrm{r}()$ denotes the matroid *rank* ("dimension").

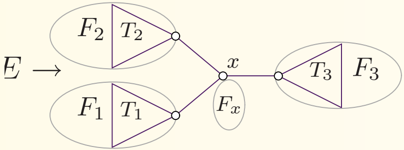
(M) Tree-width $=$ $\mathbf{min}_{\text{decomps. of } M}$ $\mathbf{max}\,\{$ *node-width* in decomp. $\}$.

Where this idea comes from?

- A general definition of matroid tree-width proposed by [PH & Whittle, 03], following unpublished [Geelen].

Definition: A *tree-decomposition* of a matroid M is a tree T with

– an arbitrary $\tau : E(M) \to V(T)$, without further restrictions.



– *Node with of x* $= \sum_{i=1}^{d} \mathrm{r}(M \setminus F_i) - (d-1) \cdot \mathrm{r}(M)$,

where $\mathrm{r}()$ denotes the matroid *rank* ("dimension").

(M) Tree-width $=$ $\mathbf{min}_{\text{decomps. of } M}$ $\mathbf{max}\,\{$ *node-width* in decomp. $\}$.

- BTW, if a matroid M has tree-width k and branch-width b (which readily extends to matroids), then $b-1 \leq k \leq \max(2b-1, 1)$ — that is nice...

Comparing the tree-width parameters

Theorem [PH & Whittle, 03]. Let a graph G has an edge, and M be the cycle matroid of G. Then the tree-width of G equals the tree-width of M.

Comparing the tree-width parameters

Theorem [PH & Whittle, 03]. Let a graph G has an edge, and M be the cycle matroid of G. Then the tree-width of G equals the tree-width of M.

Equivalently:

Theorem [PH & Whittle, 03]. Let a graph G has an edge. Then the VF tree-width of G equals the (ordinary) tree-width of G.

Comparing the tree-width parameters

Theorem [PH & Whittle, 03]. Let a graph G has an edge, and M be the cycle matroid of G. Then the tree-width of G equals the tree-width of M.

Equivalently:

Theorem [PH & Whittle, 03]. Let a graph G has an edge. Then the VF tree-width of G equals the (ordinary) tree-width of G.

Some thoughts on these parameters...

- An equality between the above node-width formulas for graphs and matroids is easy to show.

Comparing the tree-width parameters

Theorem [PH & Whittle, 03]. Let a graph G has an edge, and M be the cycle matroid of G. Then the tree-width of G equals the tree-width of M.

Equivalently:

Theorem [PH & Whittle, 03]. Let a graph G has an edge. Then the VF tree-width of G equals the (ordinary) tree-width of G.

Some thoughts on these parameters. . .

- An equality between the above node-width formulas for graphs and matroids is easy to show.
- For vector matroids, a tree-decomposition has a nice "visualization" with
  - affine *subspaces* modelling the traditional "bags",
  - with *dimension* in place of bag size, and an *interpolation* property.

Comparing the tree-width parameters

Theorem [PH & Whittle, 03]. Let a graph G has an edge, and M be the cycle matroid of G. Then the tree-width of G equals the tree-width of M.

Equivalently:

Theorem [PH & Whittle, 03]. Let a graph G has an edge. Then the VF tree-width of G equals the (ordinary) tree-width of G.

Some thoughts on these parameters...

- An equality between the above node-width formulas for graphs and matroids is easy to show.
- For vector matroids, a tree-decomposition has a nice "visualization" with
  - affine *subspaces* modelling the traditional "bags",
  - with *dimension* in place of bag size, and an *interpolation* property.
- An ordinary tree-decomposition can be readily translated into a VF tree-decomposition; just find a bag hosting each edge of G.

3 From one Decomposition to Another

- Where we stand?
  - The VF tree-width is at most the ordinary tree-width;
    since an ordinary tree-decomposition naturally translates to a VF tree-decomposition of at most the same width.

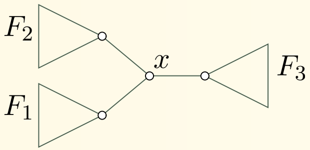
3 From one Decomposition to Another

- Where we stand?
  - The VF tree-width is at most the ordinary tree-width; since an ordinary tree-decomposition naturally translates to a VF tree-decomposition of at most the same width.
- What happens in the converse direction?
  - Again, any VF tree-decomposition naturally translates into an ordinary decomposition (just apply the interpolation property to the ends of mapped edges).

3 From one Decomposition to Another

- Where we stand?
  - The VF tree-width is at most the ordinary tree-width;
    since an ordinary tree-decomposition naturally translates to a VF tree-decomposition of at most the same width.
- What happens in the converse direction?
  - Again, any VF tree-decomposition naturally translates into an ordinary decomposition (just apply the interpolation property to the ends of mapped edges).
  - However, the width may increase (dramatically)!

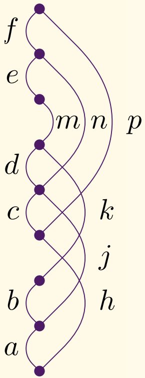
The problem is that edges mapped to a branch in the decomposition may induce a disconnected subgraph, hence further decreasing the node-width in the VF setting...

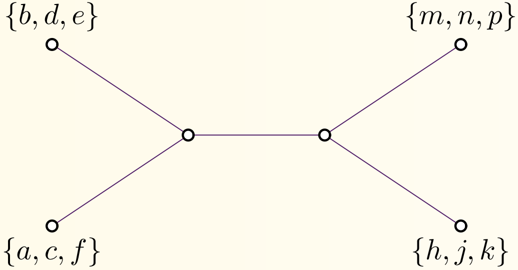


node-with of $x =$

$|V(G)| + (d-1) \cdot c(G) - \sum_{i=1}^{d} c(G - F_i)$

An example of a "disconnected" decomposition

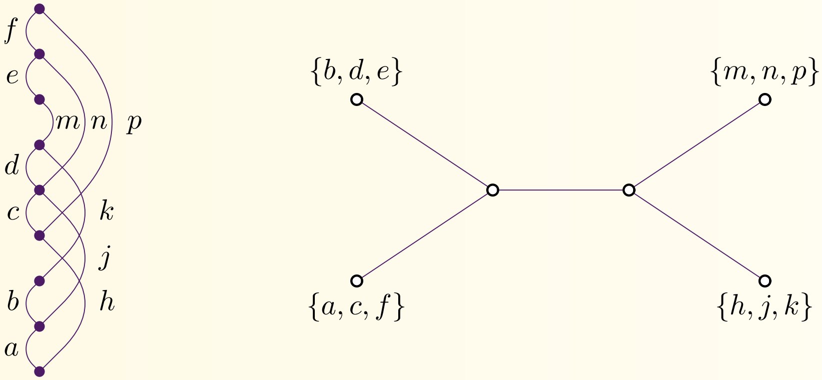




node-with formula $= |V(G)| + (d-1) \cdot c(G) - \sum_{i=1}^{d} c(G - F_i)$

Easy to check that all six nodes in this VF tree-decomposition have width 4.

An example of a "disconnected" decomposition



node-with formula $= |V(G)| + (d-1) \cdot c(G) - \sum_{i=1}^{d} c(G - F_i)$

Easy to check that all six nodes in this VF tree-decomposition have width $\mathbf{4}$.

However, the central two nodes induce bags of size $\mathbf{9}$ in an ordinary tree-decomposition! (tree-width up to 8)

Handling a "disconnected" decomposition

- If we want to get an ordinary tree-decomposition of the same width, we have to alter "disconnected" spots of a VF tree-decomposition. . .
- Actually, the proof complications appear similar to those emerging when proving equality of matroid branch-width to graph branch-width [Hicks & McMurray, 07], [Mazoit & Thomassé].
  (No short proof of this statement is known so far.)

Handling a “disconnected” decomposition

- If we want to get an ordinary tree-decomposition of the same width, we have to alter “disconnected” spots of a VF tree-decomposition. . .

- Actually, the proof complications appear similar to those emerging when proving equality of matroid branch-width to graph branch-width [Hicks & McMurray, 07], [Mazoit & Thomassé].
(No short proof of this statement is known so far.)

- The “easy” altering method published as a proof in [PH & Whittle, EJC 06] was, unfortunately, not correct (it did not cover all the cases);

  as pointed out by [Adler 07].

Handling a "disconnected" decomposition

- If we want to get an ordinary tree-decomposition of the same width, we have to alter "disconnected" spots of a VF tree-decomposition...

- Actually, the proof complications appear similar to those emerging when proving equality of matroid branch-width to graph branch-width [Hicks & McMurray, 07], [Mazoit & Thomassé].
(No short proof of this statement is known so far.)

- The "easy" altering method published as a proof in [PH & Whittle, EJC 06] was, unfortunately, not correct (it did not cover all the cases);

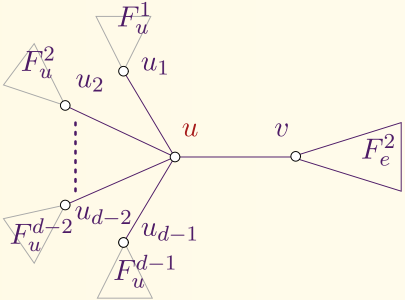
  as pointed out by [Adler 07].

- In response to that, [PH & Whittle, 08] have got an updated, though longer proof.

  We sketch its idea next...

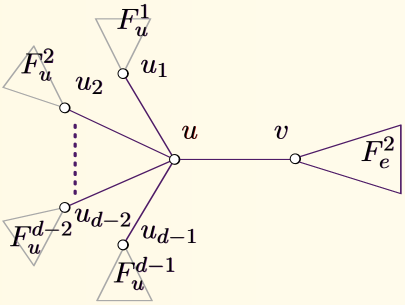
Proof (altering a "disconnected" edge of a VF tree-decomposition T of G).

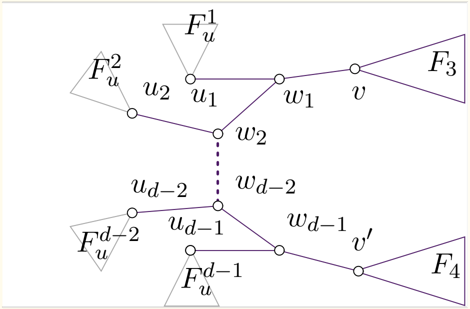
- We assume an edge $e = uv$ of T such that the G-edges mapped to the u-branch of T form a disconnected subgraph of G, and that the edges mapped to the branches of u-neighbours (not v) stay connected in G.



Proof (altering a "disconnected" edge of a VF tree-decomposition T of G).

- We assume an edge $e = uv$ of T such that the G-edges mapped to the u-branch of T form a disconnected subgraph of G, and that the edges mapped to the branches of u-neighbours (not v) stay connected in G.

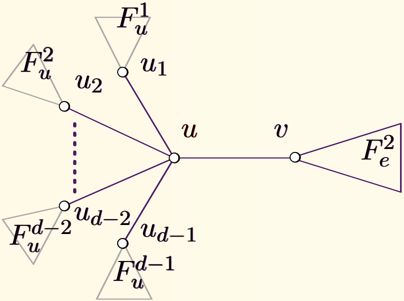


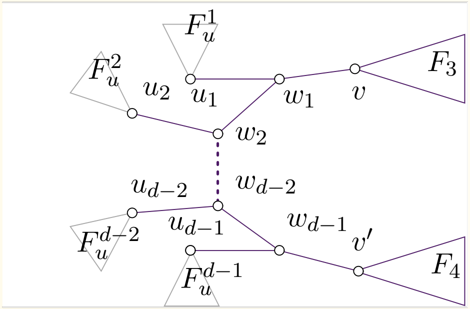


- If we find a disconnected partitioning (of the G-edges mapped to the v-branch) $F_e^2 = F_3 \cup F_4$, then we *"split" T* as above.

Proof (altering a "disconnected" edge of a VF tree-decomposition T of G).

- We assume an edge $e = uv$ of T such that the G-edges mapped to the u-branch of T form a disconnected subgraph of G, and that the edges mapped to the branches of u-neighbours (not v) stay connected in G.

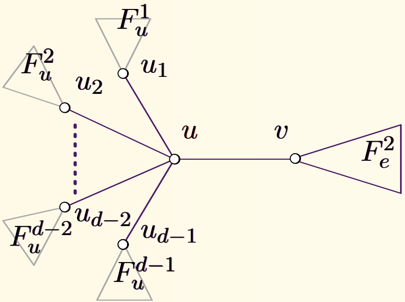


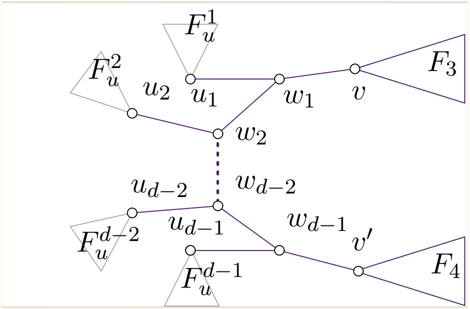


- If we find a disconnected partitioning (of the G-edges mapped to the v-branch) $F_e^2 = F_3 \cup F_4$, then we *"split"* T as above.
  The hard part is to prove that width does not increase (two subcases).

Proof (altering a "disconnected" edge of a VF tree-decomposition T of G).

- We assume an edge $e = uv$ of T such that the G-edges mapped to the u-branch of T form a disconnected subgraph of G, and that the edges mapped to the branches of u-neighbours (not v) stay connected in G.



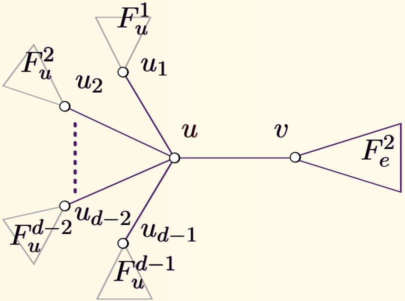


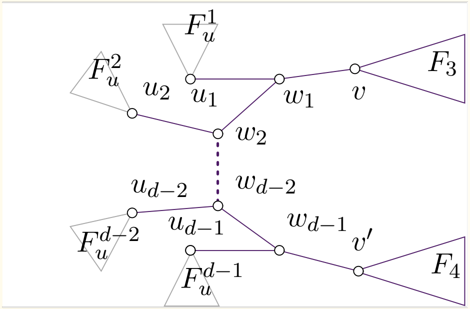
- If we find a disconnected partitioning (of the G-edges mapped to the v-branch) $F_e^2 = F_3 \cup F_4$, then we *"split" T* as above.
  The hard part is to prove that width does not increase (two subcases).

- If F_e^2 is connected in G, then we simply contract e in T (an easy case).

Proof (altering a "disconnected" edge of a VF tree-decomposition T of G).

- We assume an edge $e = uv$ of T such that the G-edges mapped to the u-branch of T form a disconnected subgraph of G, and that the edges mapped to the branches of u-neighbours (not v) stay connected in G.





- If we find a disconnected partitioning (of the G-edges mapped to the v-branch) $F_e^2 = F_3 \cup F_4$, then we *"split" T* as above.
  The hard part is to prove that width does not increase (two subcases).

- If F_e^2 is connected in G, then we simply contract e in T (an easy case).

- After all, there is a "strictly decreasing" sequence of alterations, leading to the connected case in which both tree-width measures are equal.

4 Conclusions

- Shown that a matroidal (geometric) view can bring new and interesting notions and properties of ordinary graphs – cf. the *VF tree-width*.

4 Conclusions

- Shown that a matroidal (geometric) view can bring new and interesting notions and properties of ordinary graphs – cf. the *VF tree-width*.

- A VF tree-decomposition seems to be a "stronger" notion:

  – We have seen that there exist VF tree-decompositions which do not easily translate to ordinary decompositions of the same width. . .

4 Conclusions

- Shown that a matroidal (geometric) view can bring new and interesting notions and properties of ordinary graphs – cf. the *VF tree-width*.

- A VF tree-decomposition seems to be a "stronger" notion:

  - We have seen that there exist VF tree-decompositions which do not easily translate to ordinary decompositions of the same width. . .

- Can VF tree-width notion be used to provide some easier proofs in structural graph theory?

4 Conclusions

- Shown that a matroidal (geometric) view can bring new and interesting notions and properties of ordinary graphs – cf. the *VF tree-width*.
- A VF tree-decomposition seems to be a "stronger" notion:
  - We have seen that there exist VF tree-decompositions which do not easily translate to ordinary decompositions of the same width. . .
- Can VF tree-width notion be used to provide some easier proofs in structural graph theory?

THANK YOU FOR ATTENTION