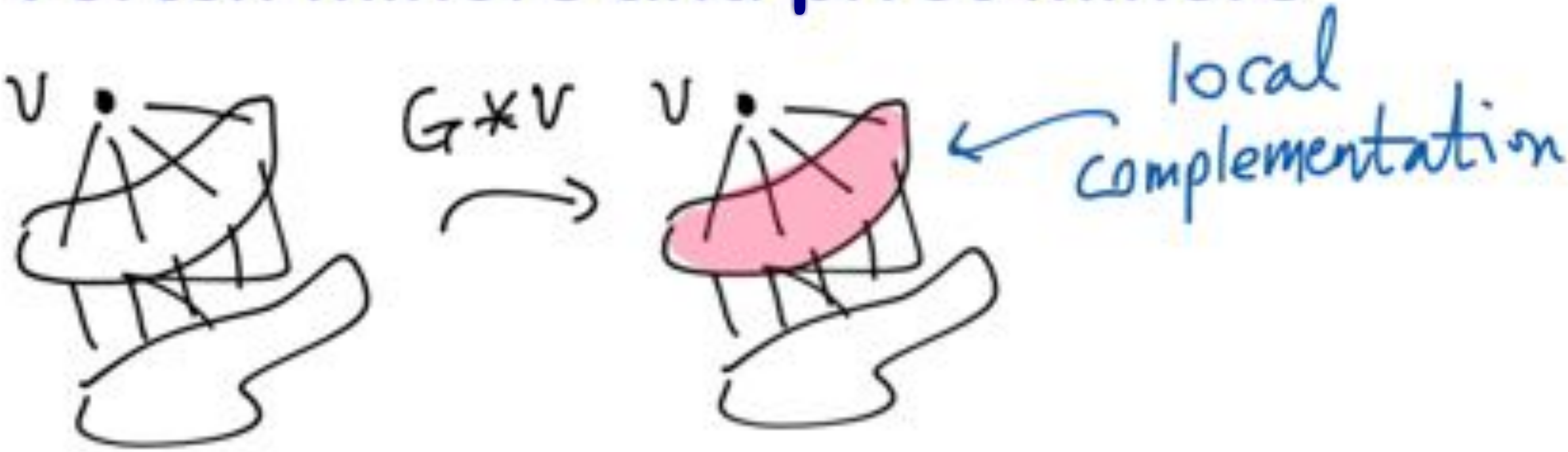
Vertex-minors and Pivot-minors

Sang-il Oum
KAIST
Daejeon, Korea

ISMP 2012 (Berlin)

Vertex-minors and pivot-minors



Vertex-minors and pivot-minors

H is a vertex-minor of G if $H = G * v_1 * v_2 * \cdots * v_k \setminus W$

for some vertices $v_1 \cdots v_k$ and $W \subseteq V(G)$.

ex :

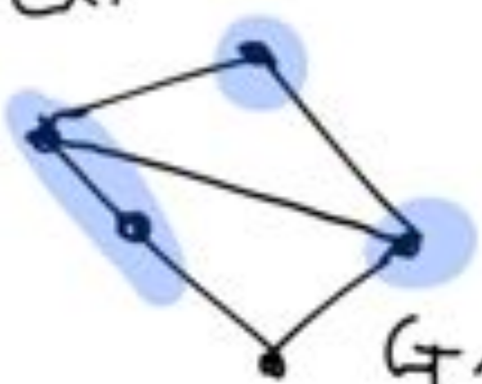
is a vertex-minor of

Vertex-minors and pivot-minors

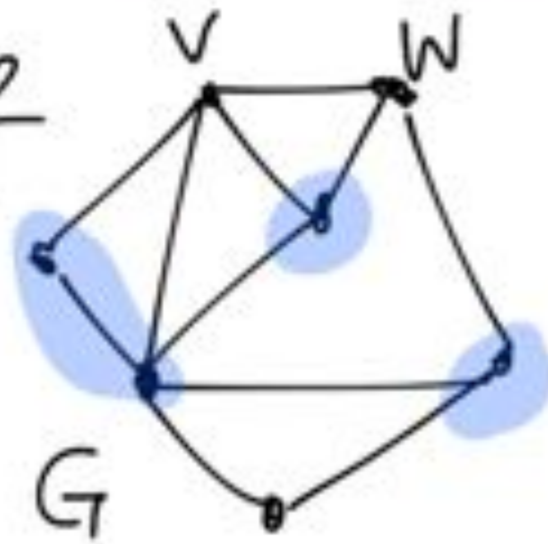
Vertex-minors and pivot-minors

H is a pivot-minor of G if $H = G \wedge e_1 \wedge e_2 \wedge \cdots \wedge e_k \setminus W$ for edges $e_i \in E(G \wedge e_1 \wedge \cdots \wedge e_{i-1})$ and $W \subseteq V(G)$.

ex)

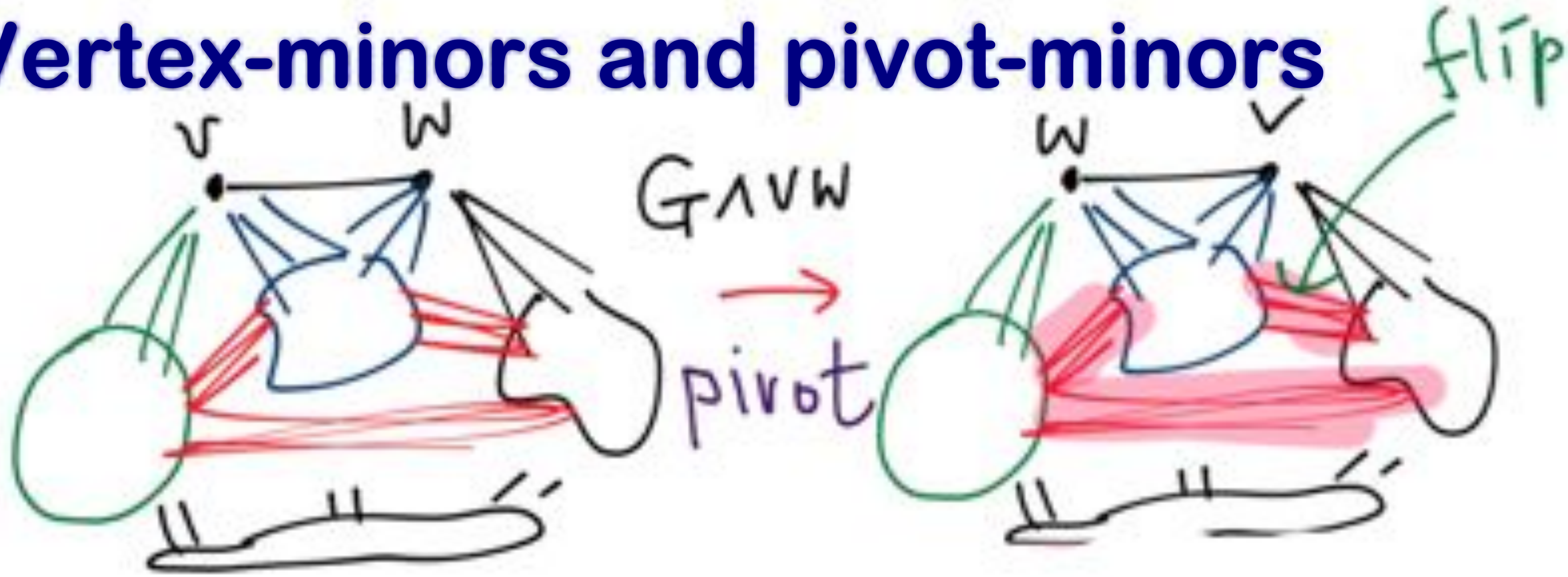


is a pivot-minor of



$G \wedge vw \setminus v \setminus w$ G

Vertex-minors and pivot-minors

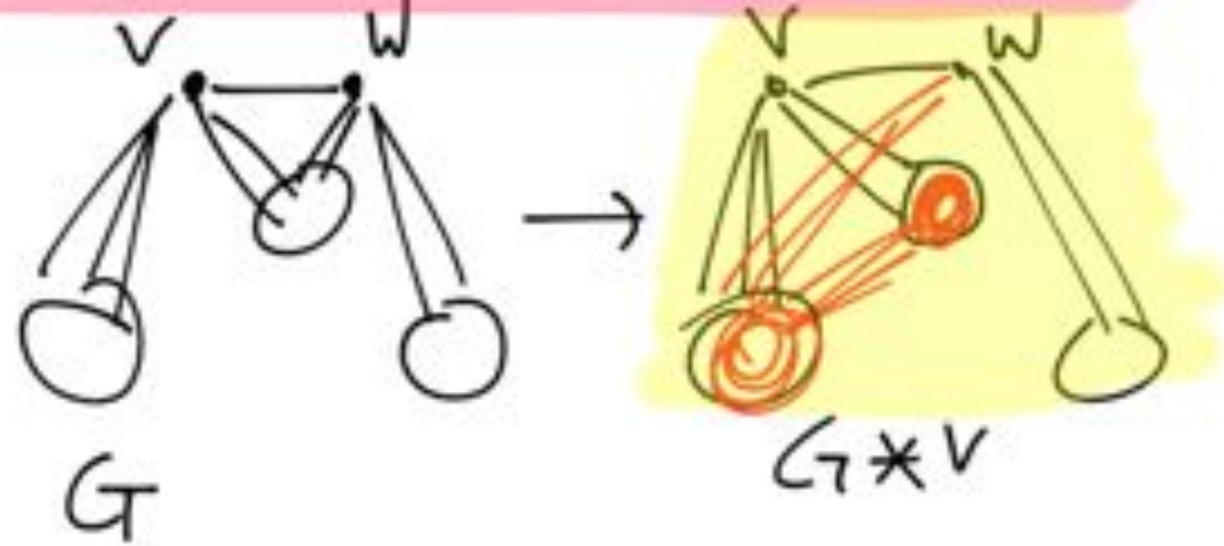


Every pivot-minor is a vertex-minor.

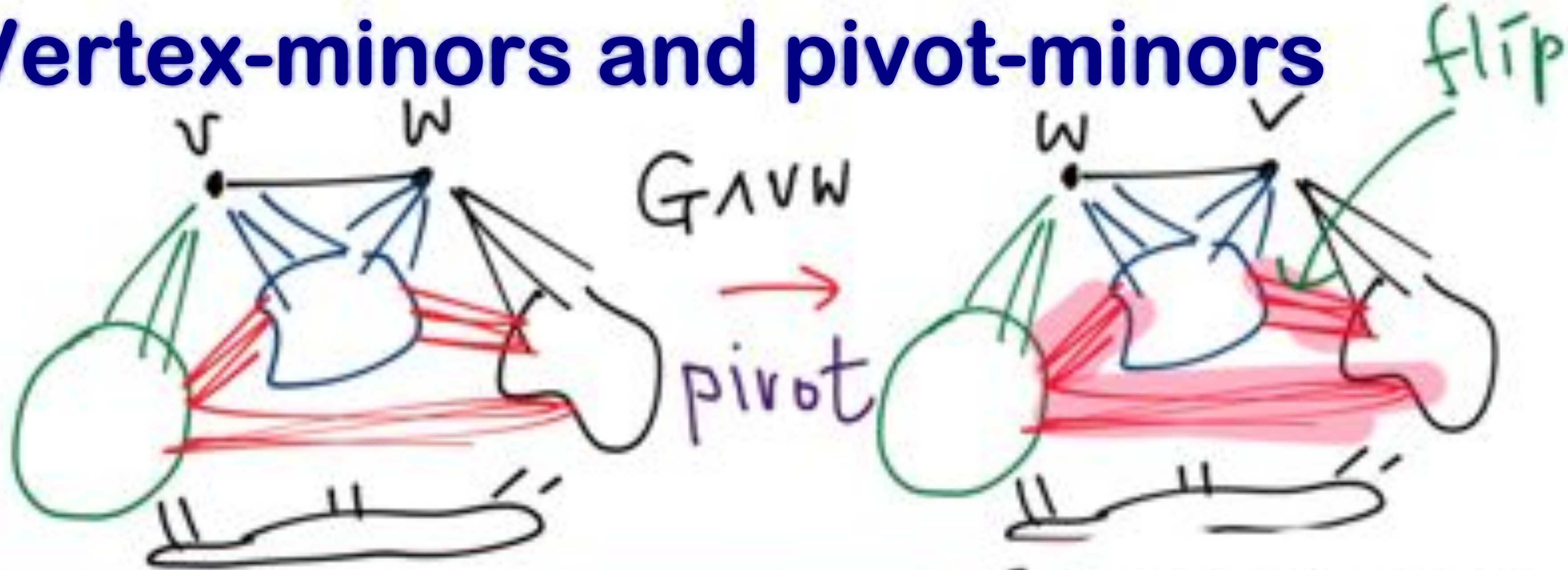
Vertex-minors and pivot-minors

$G \wedge vw = G * v * w * v$

Every pivot-minor is a vertex-minor.

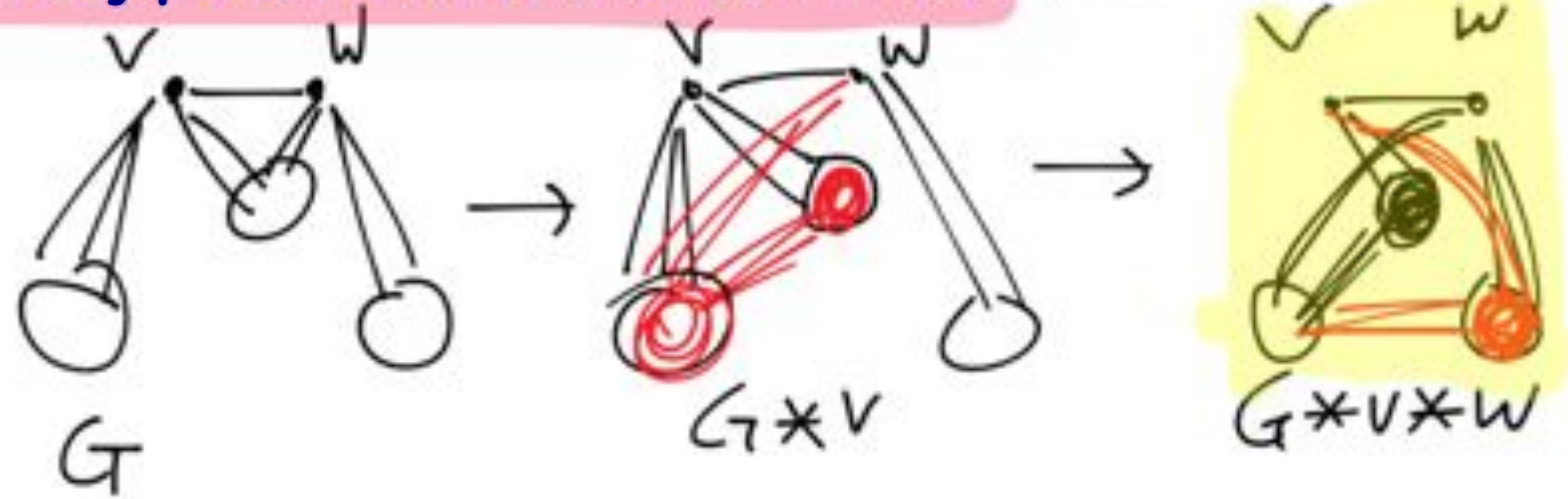


Vertex-minors and pivot-minors



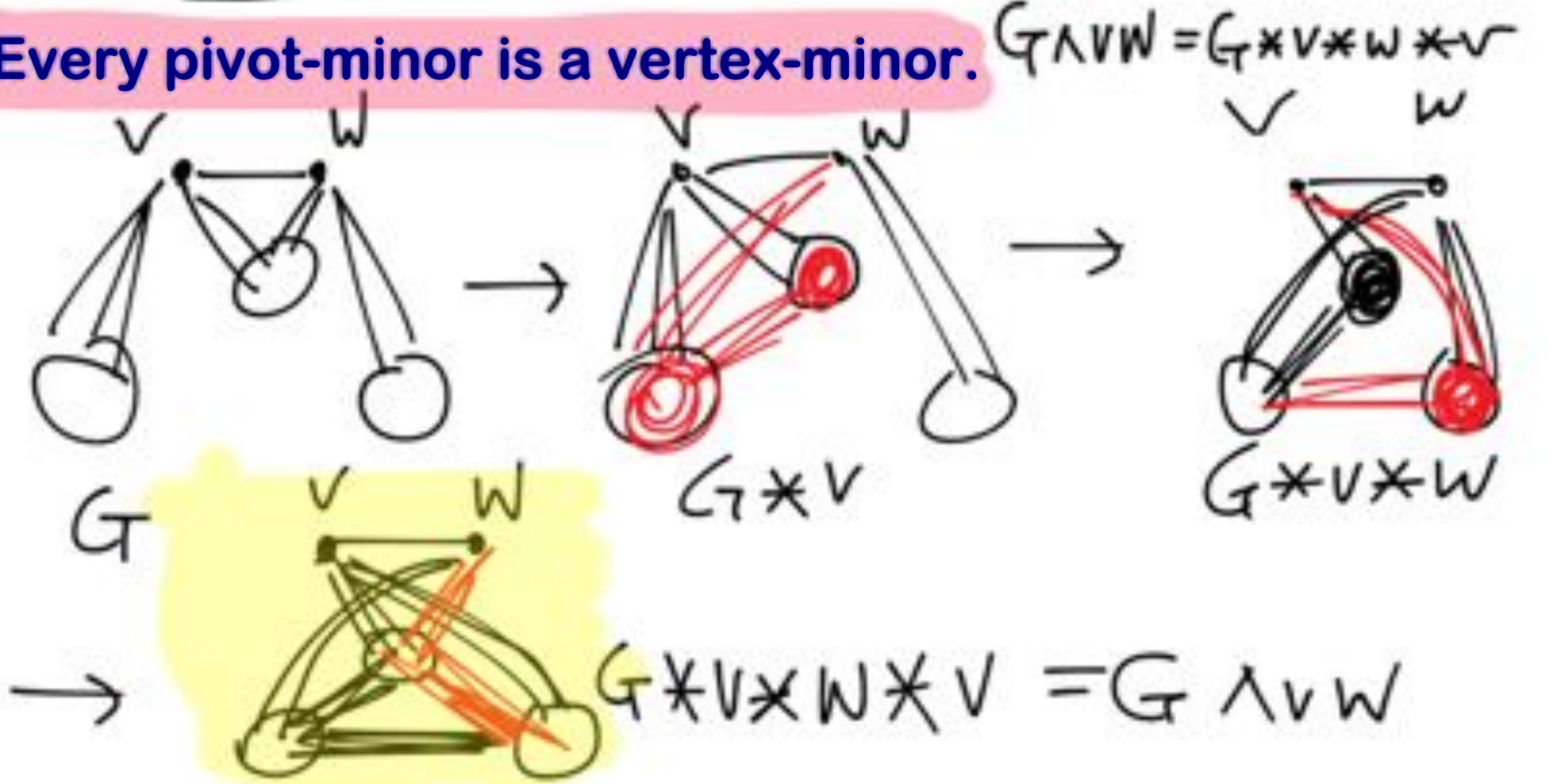
Every pivot-minor is a vertex-minor.

$G\wedge vw = G*v*w*v$



Vertex-minors and pivot-minors

Every pivot-minor is a vertex-minor.



Vertex-minors and pivot-minors

Every pivot-minor is a vertex-minor.

Not every vertex-minor is a pivot-minor.

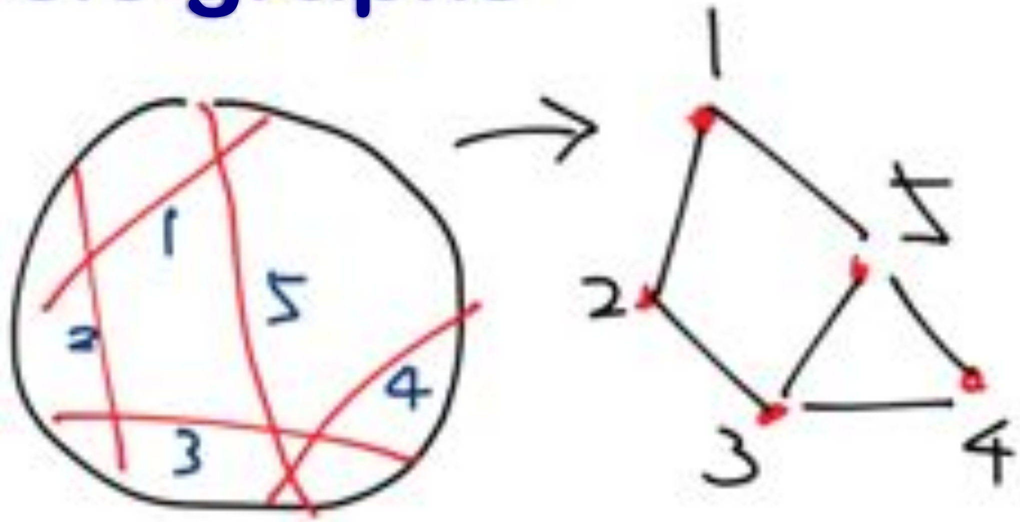
Every pivot-minor of a bipartite graph is bipartite.

G∗v (local complementation)

Motivation

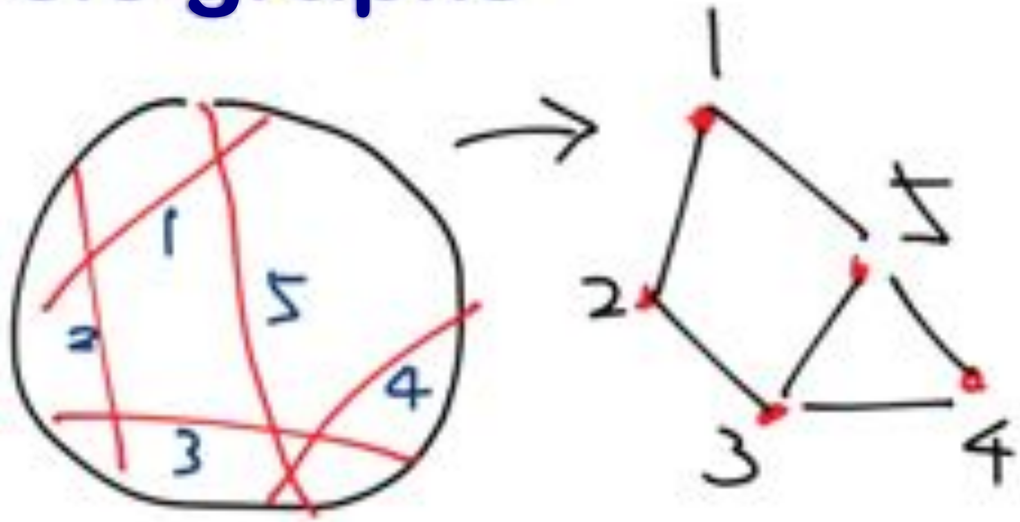
Motivation 1 - circle graphs

Circle graph: intersection graph of chords in a circle

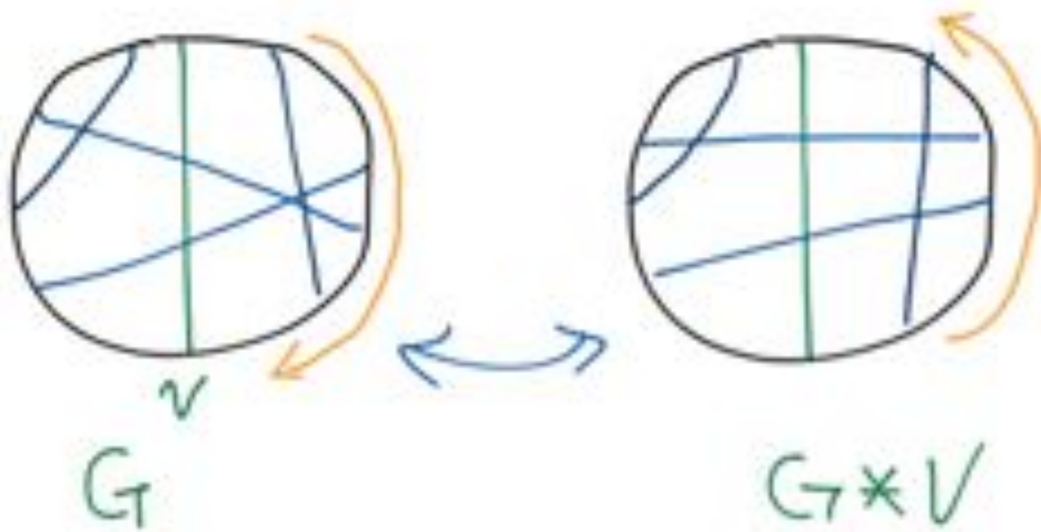


Motivation 1 - circle graphs

Circle graph: intersection graph of chords in a circle

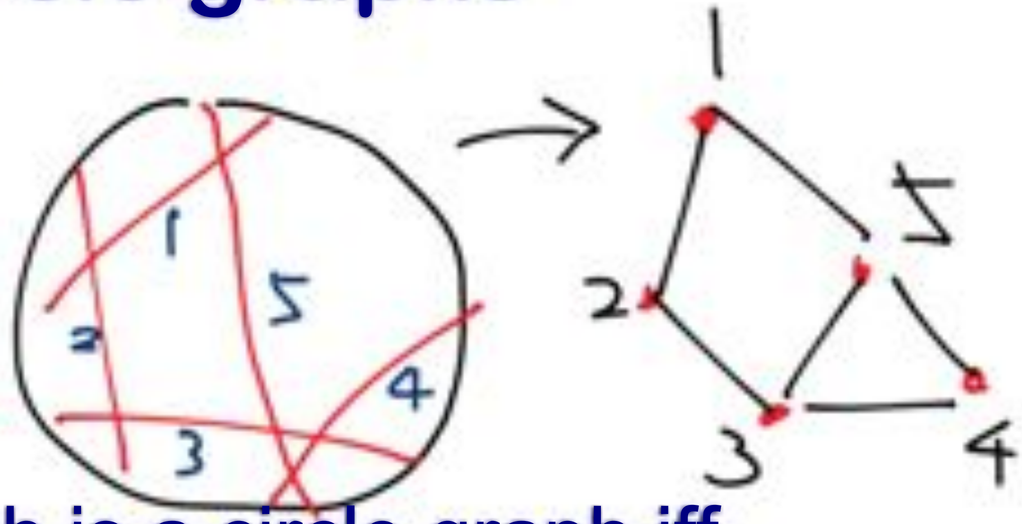


A vertex-minor of a circle graph is a circle graph.



Motivation 1 - circle graphs

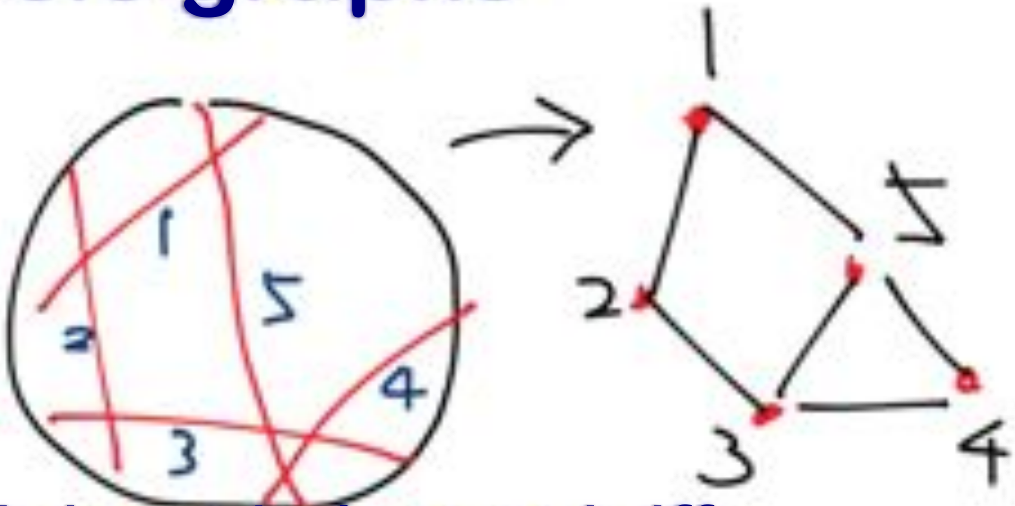
Circle graph: intersection graph of chords in a circle



Bouchet 1994: A graph is a circle graph iff it has no vertex-minor isomorphic to

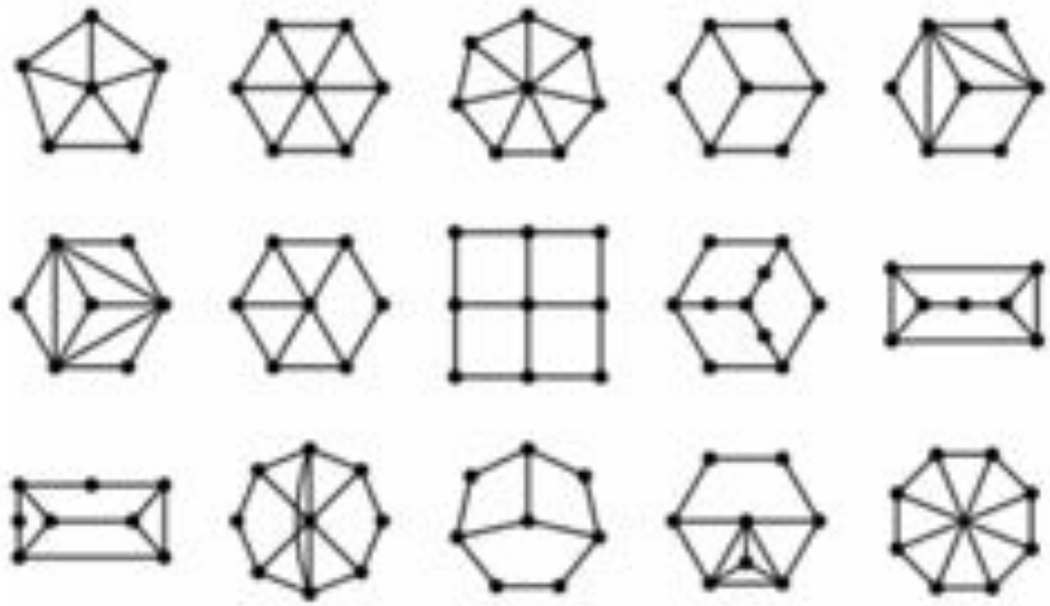
Motivation 1 - circle graphs

Circle graph: intersection graph of chords in a circle



Bouchet 1994: A graph is a circle graph iff it has no vertex-minor isomorphic to

Geelen, O. 2009: A graph is a circle graph iff it has no pivot-minor isomorphic to

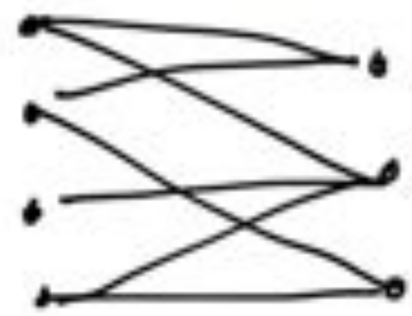


Motivation 2 - Binary matroids

Standard representation of a binary matroid

$$\left(\begin{array}{cccc|ccc} 1 & & & 0 & 1 & 1 & 0 \\ & 1 & & & 1 & 0 & 1 \\ & & 1 & & 0 & 1 & 0 \\ 0 & & & 1 & 0 & 1 & 1 \end{array}\right) \longrightarrow$$

Fundamental graph

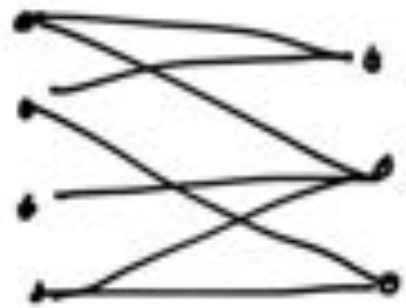


Motivation 2 - Binary matroids

Standard representation of a binary matroid

$$\left(\begin{array}{cccc|ccc} 1 & & & 0 & 1 & 1 & 0 \\ & 1 & & & 1 & 0 & 1 \\ & & 1 & & 0 & 1 & 0 \\ 0 & & & 1 & 0 & 1 & 1 \end{array}\right) \longrightarrow$$

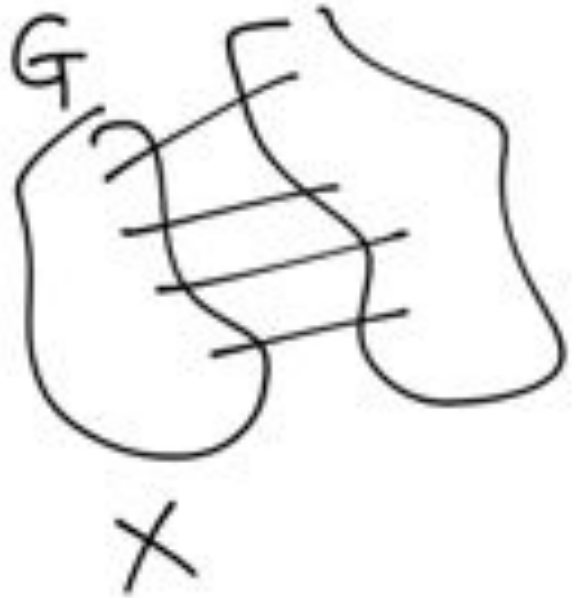
Fundamental graph



$\downarrow$ matroid minor (or its dual) $\rightleftarrows$ pivot-minor $\downarrow$

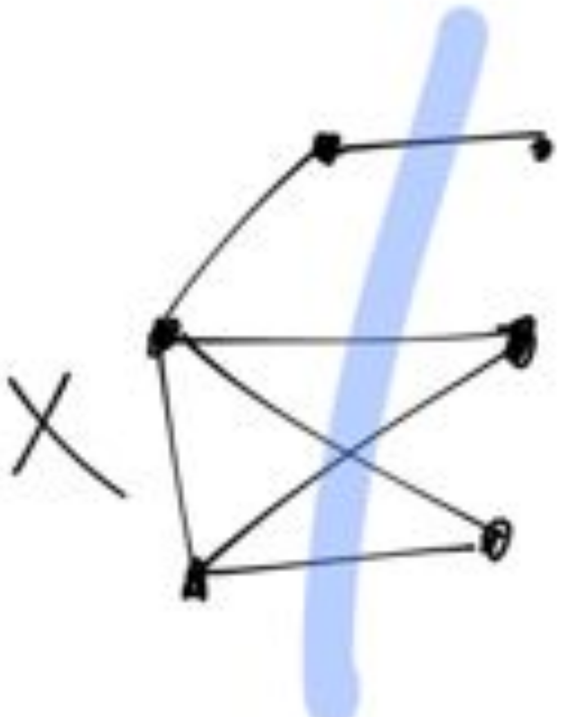
theory on minors of binary matroids $\sim$ pivot-minors of bipartite graphs

Motivation 3 - cut-rank function



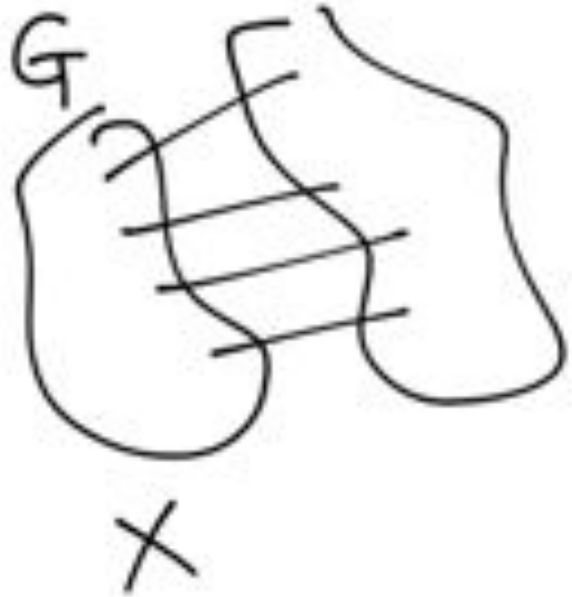
$$\mathrm{cutrk}_G(X) = \mathrm{rank}\begin{array}{c} \\ X \end{array}\!\!\begin{array}{c} V(G)-X \\ \left(\boxed{}\right) \end{array}$$

0-1 matrix over GF(2).



$$\mathrm{cutrk}(X) = \mathrm{rank}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} = 2$$

Motivation 3 - cut-rank function



$$\mathrm{cutrk}_G(X)$$

$$= \mathrm{rank}\begin{array}{cc} & V(G)-X \\ X & \left(\boxed{}\right) \end{array}$$

0-1 matrix over $GF(2)$.

Fact: Cut-rank function is invariant under taking local complementation and pivot.

Cut-rank function = natural connectivity measure in the context of vertex-minors and pivot-minors

Motivation 3 - rank-width

Rank-width: measuring how easy it is to decompose a graph into a tree-like structure where each cut has small cut-rank
(introduced by O., Seymour 2006)

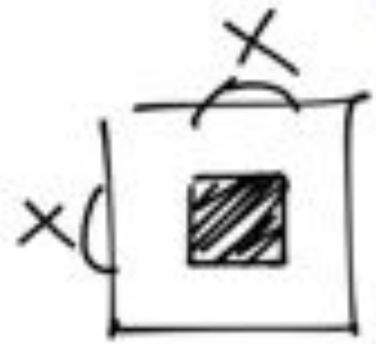
If H is a vertex-minor (or pivot-minor) of G, then rankwidth(H)<= rankwidth(G)

cf: If H is a minor of G, then treewidth(H)<= treewidth(G)

Motivation 4 - Principally unimodular

A square matrix A is principally unimodular (PU) if det(A[X])= +1, -1, 0 for all X.

↑ principal submatrix



A graph is PU-orientable if
∃ orientation so that
its directed adjacency matrix is PU.

{PU-orientable graphs} are closed under taking pivot-minors.

Problems

Well-quasi-ordering?

Conjecture If $\mathcal{C}$ is a set of graphs closed under taking pivot-minors (or vertex-minors), then $\exists$ finitely many graphs $H_1, \cdots, H_k$ s.t. $G \in \mathcal{C} \Leftrightarrow$ no H_i is a pivot-minor (vertex-minor) of G.

Equivalently.

$\boxed{\text{If } G_1, G_2, G_3, \cdots \text{ is an infinite sequence of graphs, then } \exists i<j \text{ s.t. } G_i \text{ is a pivot-minor (vertex-minor) of } G_j.}$

If $G_1, G_2, G_3, \cdots$ is an infinite sequence of graphs, then $\exists i<j$ s.t. G_i is a ~~pivot-minor~~ (vertex-minor) of G_j.

Known: True when G_i's are

(1) Graphs of small rank-width (O., 2008)

(2) Bipartite graphs (via binary matroids)

(3) Line graphs (via group-labelled graphs)

Geelen et al.

(4) Circle graphs [GM XXIII, immersion of 4-regular graphs)

If $G_1, G_2, G_3, \cdots$ is an infinite sequence of graphs, then $\exists i<j$ s.t. G_i is a pivot minor (~~vertex minor~~) of G_j.

Known: True when G_i's are

(1) Graphs of small rank-width (O., 2008)

(2) Bipartite graphs (via binary matroids)

(3) Line graphs (via group-labelled graphs)

Geelen et al.

pivot minors

?

(4) Circle graphs [GM XXIII, immersion of 4-regular graphs)

Corollaries of the conjecture

(1) Robertson-Seymour graph minors theorem
(2) Geelen et al.'s matroid minors theorem for binary matroids
(3) Finitely many forbidden pivot-minors for PU-orientable graphs

OPEN!

Thm (Bouchet) : Circle graphs are PU-orientable

Possible first step to the conjecture

Problem: For each bipartite circle graph H, there exists c(H) such that if G has no H pivot-minor, then

rankwidth(G)$<$c(H)

OPEN!

True for:

(1) Bipartite graphs → Geelen et al. matroids

(2) Circle graphs → Johnson 2002

(3) Line graphs → O. 2009

Algorithms

Problem: Can we find a poly-time algorithm to check whether an input graph has a pivot-minor isomorphic to a fixed graph H.

OPEN!

Yes for:

(1) Bipartite graphs → Geelen et al.

matroids

(2) Bounded rank-width → MSOL

(3) Line graphs → group-labelled graphs

Thank you for your attention!

Question?