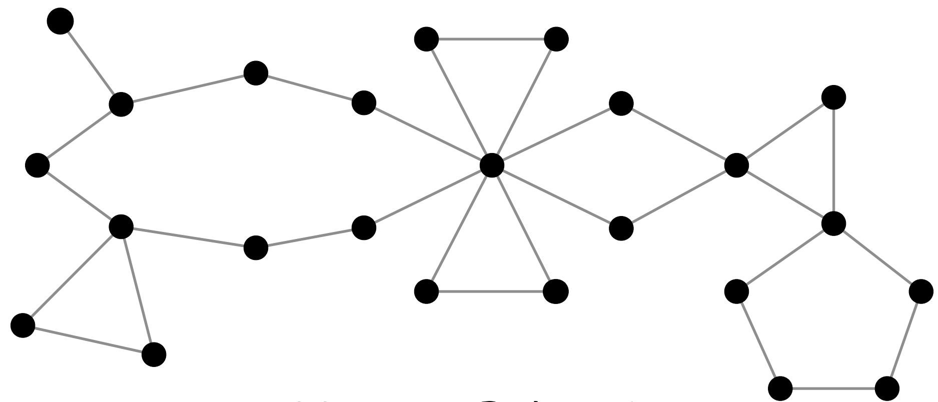
An Exact Enumeration of Unlabeled Cactus Graphs



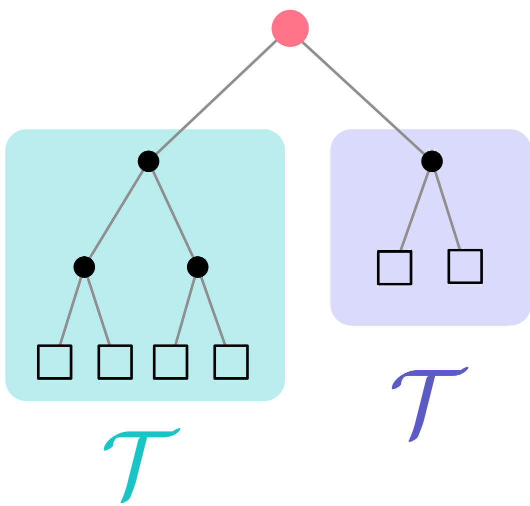
Maryam Bahrani

Under the Direction of Dr. Jérémie Lumbroso

PRINCETON UNIVERSITY

Analysis of Algorithms, June 2017

Symbolic Method on Trees



A **binary tree** is
— either a leaf □
— or an internal node,
and a left subtree,
and a right subtree

$$\mathcal{T} = \square \cup (\mathcal{T} \bullet \mathcal{T})$$

symbolic specification

$$T(z) = 1 + T(z) \times z \times T(z)$$

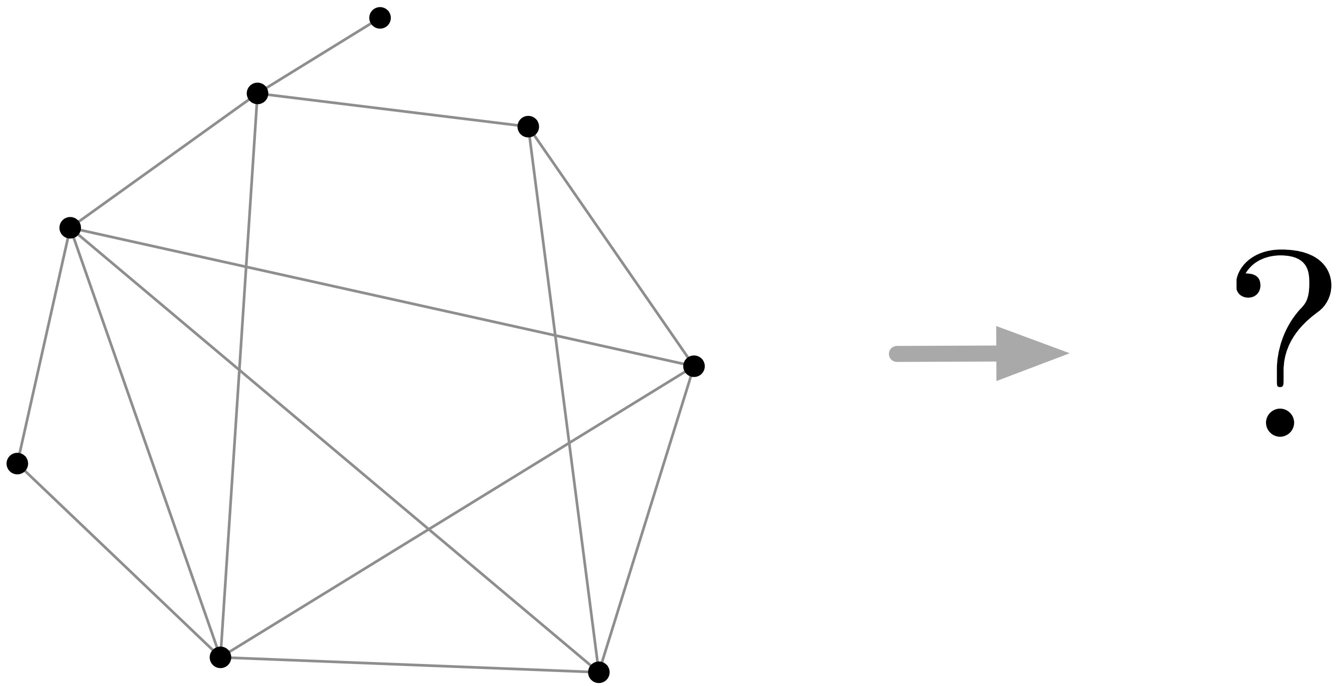
generating function

$$1, 2, 5, 14, 42, 132, 429, 1430, \ldots$$

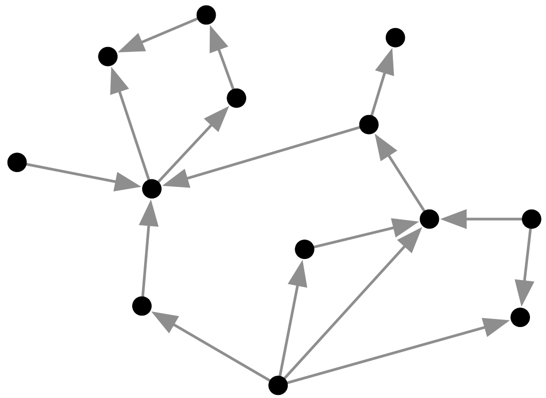
exact enumeration

Number of binary trees with 4 internal nodes

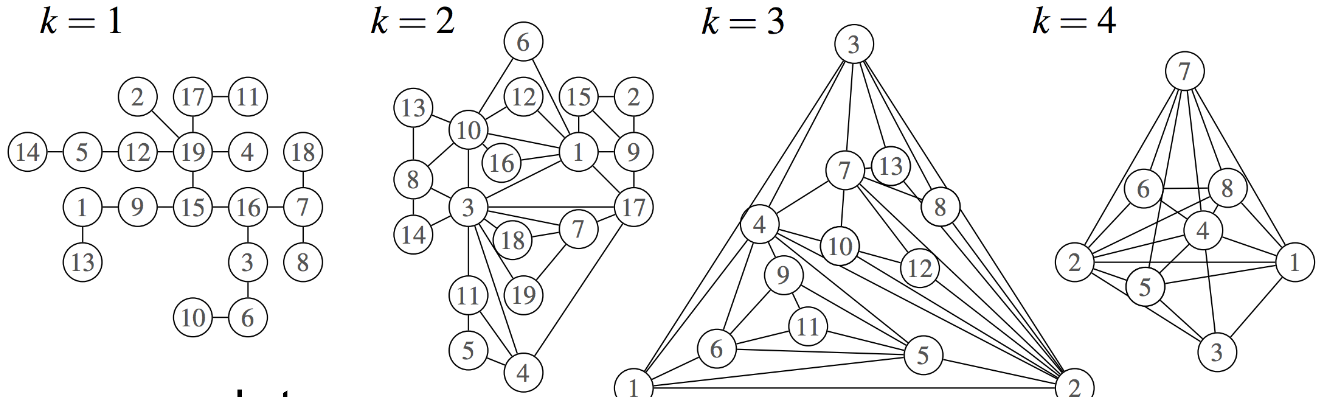
Decomposing General Graphs



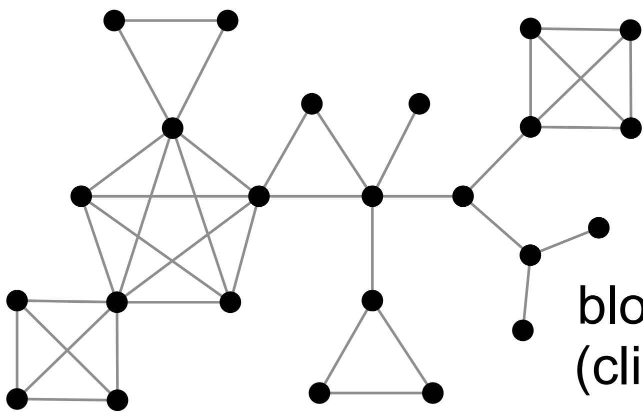
Generalizing Trees



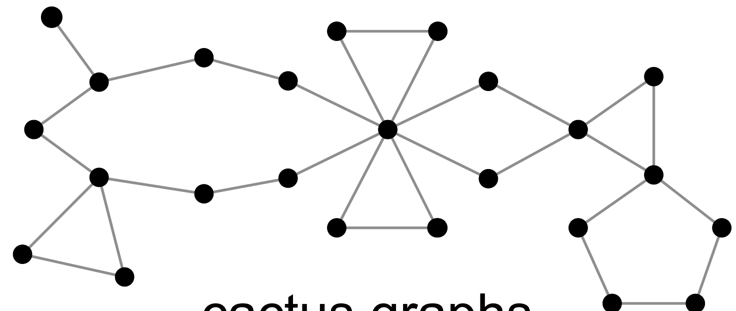
Directed Acyclic Graphs (DAGs)



k-trees

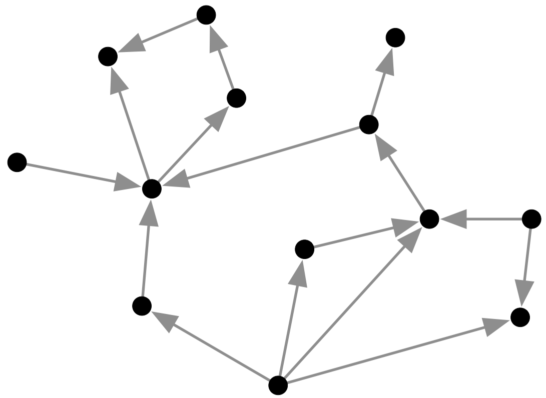


block graphs (clique trees)

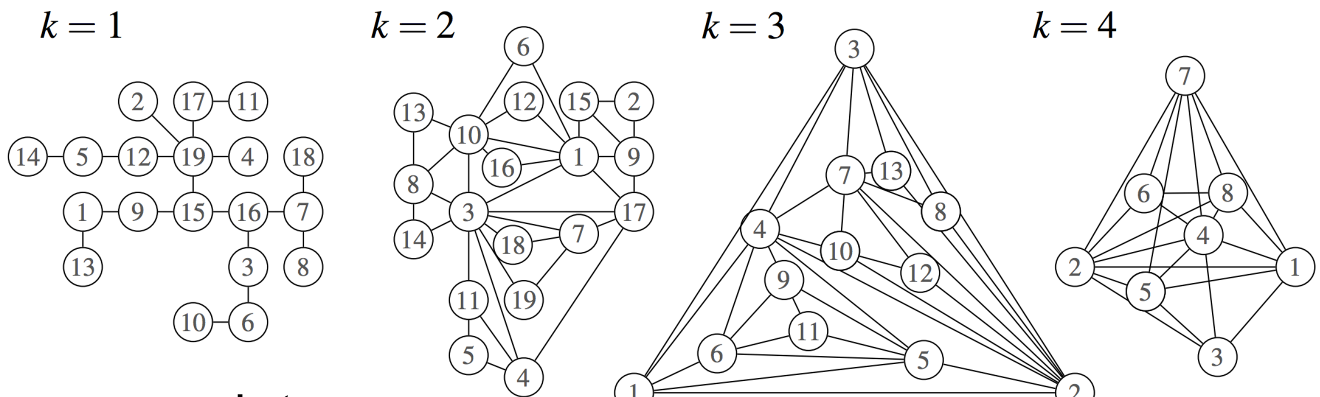


cactus graphs (cacti)

Generalizing Trees

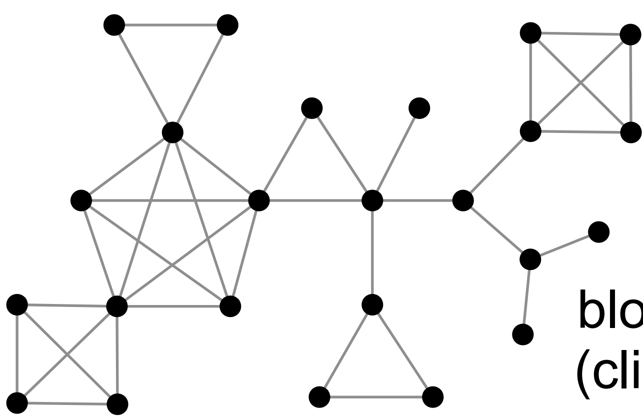


Directed Acyclic Graphs
(DAGs)

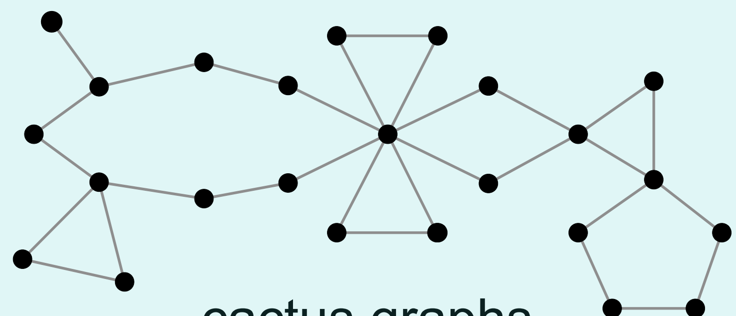


k-trees

Focus of this talk



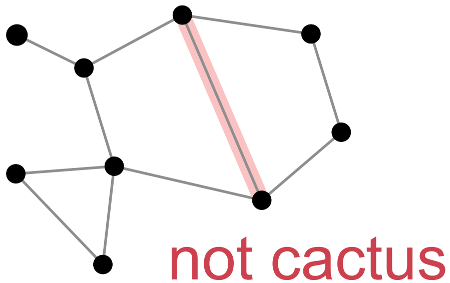
block graphs
(clique trees)

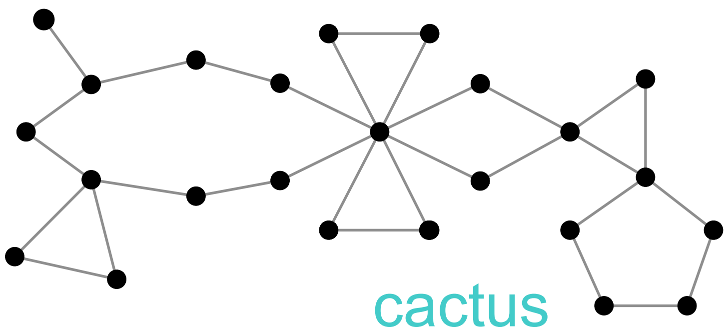


cactus graphs
(cacti)

Cactus Graphs

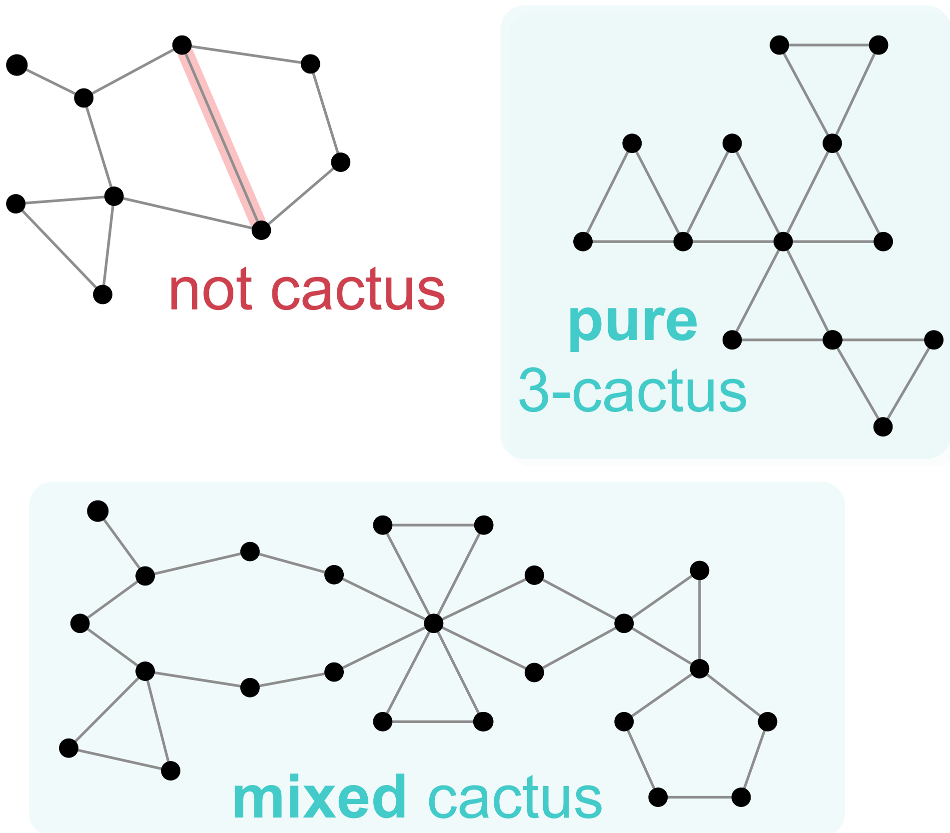
A graph is a **cactus** iff every edge is part of *at most* one cycle.





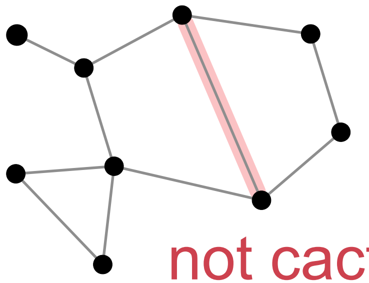
Cactus Graphs

A graph is a **cactus** iff every edge is part of *at most* one cycle.

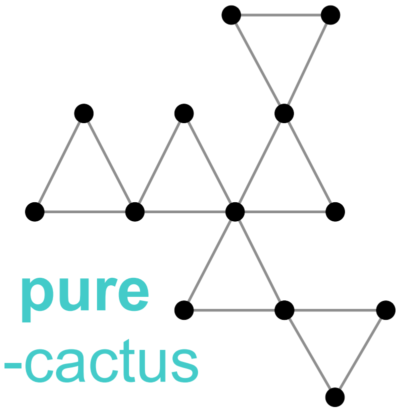


Cactus Graphs

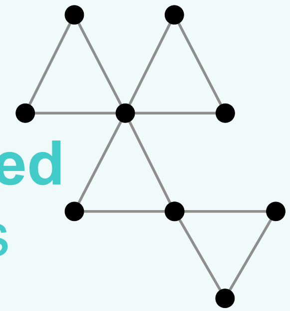
A graph is a **cactus** iff every edge is part of *at most* one cycle.



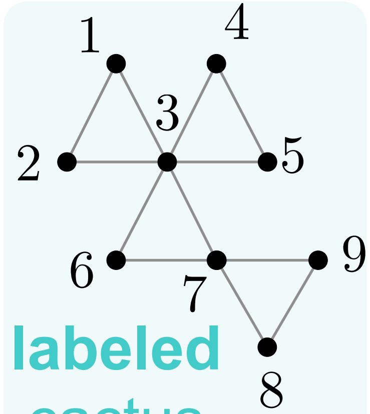
not cactus



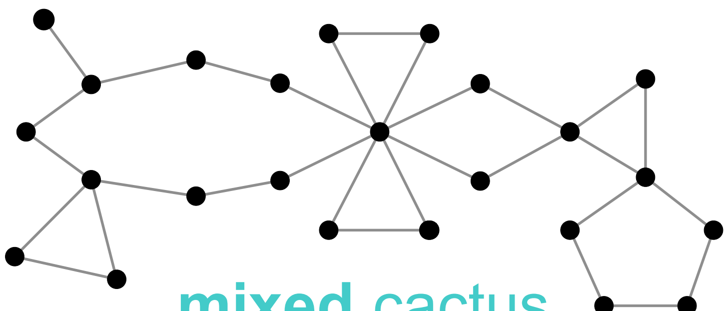
pure 3-cactus



unlabeled cactus



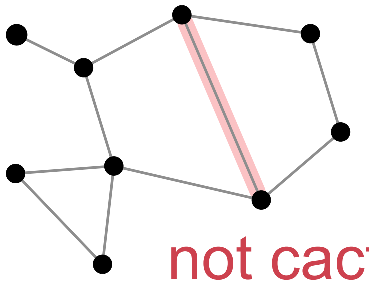
labeled cactus



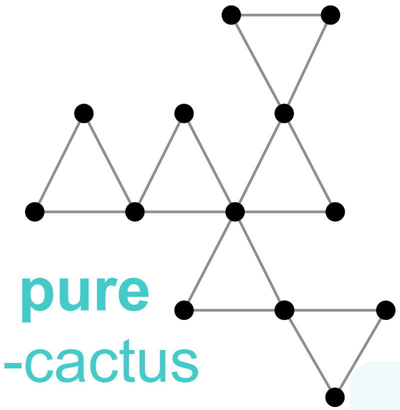
mixed cactus

Cactus Graphs

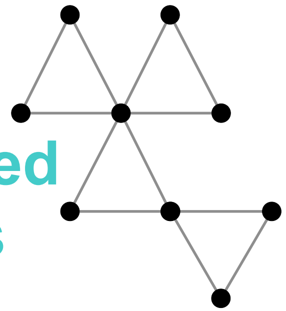
A graph is a **cactus** iff every edge is part of *at most* one cycle.



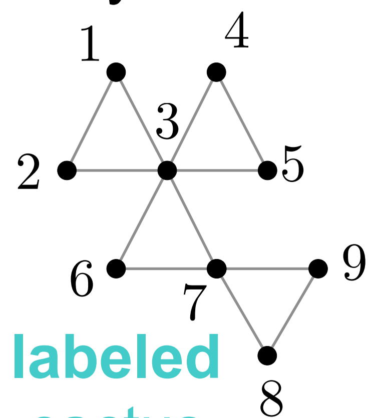
not cactus



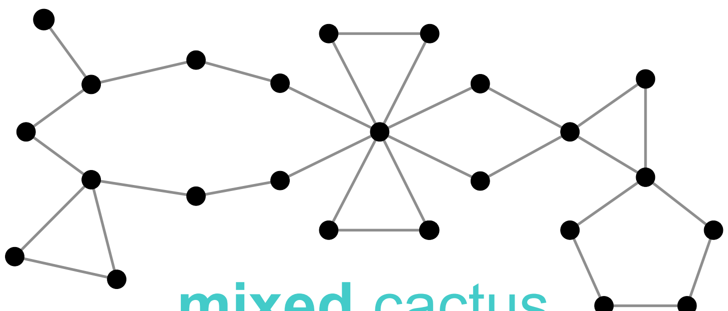
pure
3-cactus



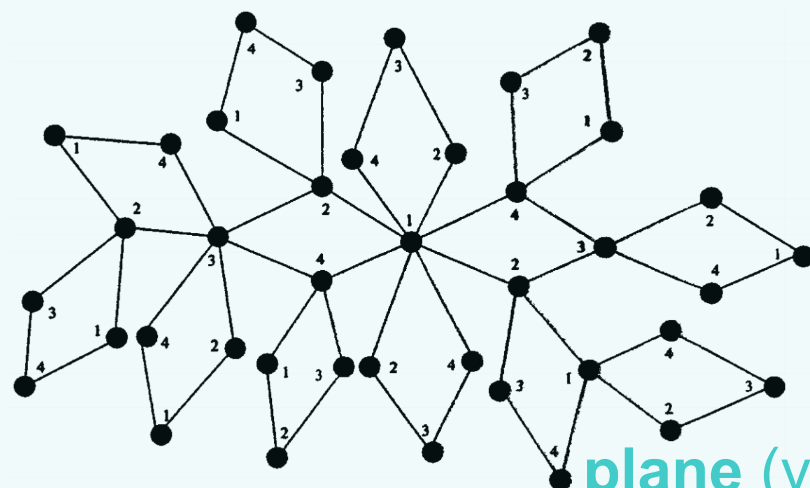
unlabeled
cactus



labeled
cactus



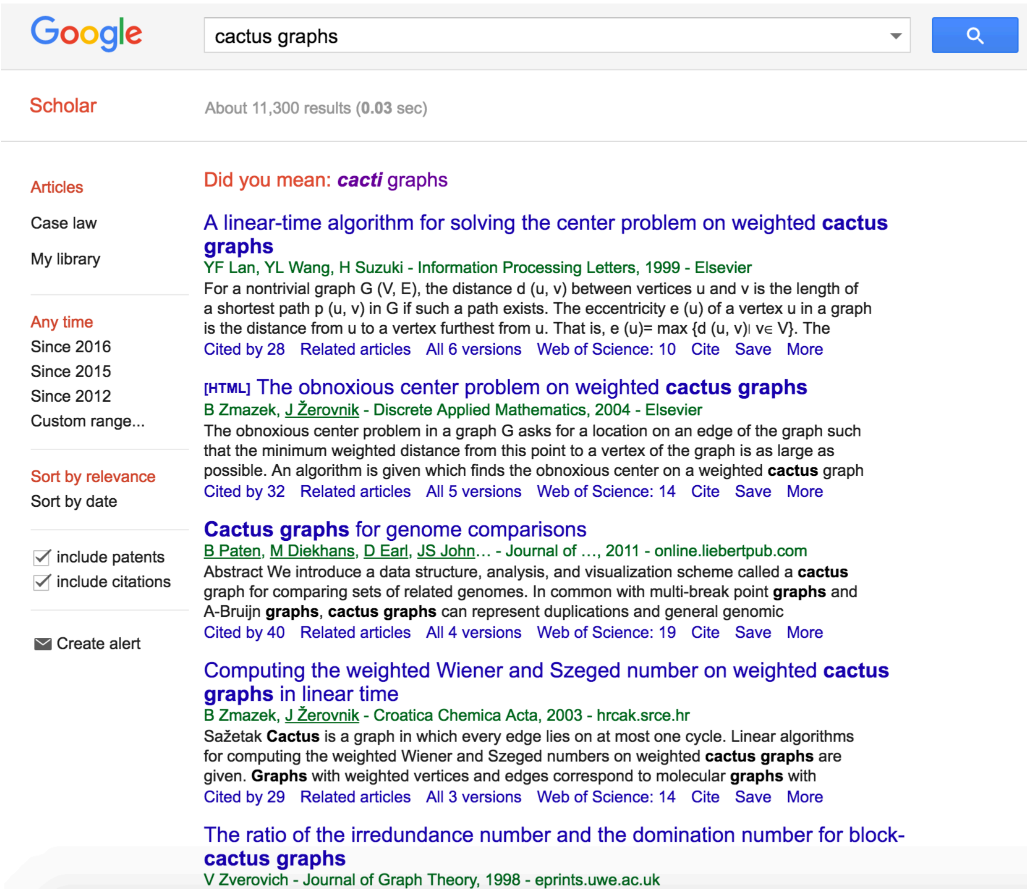
mixed cactus



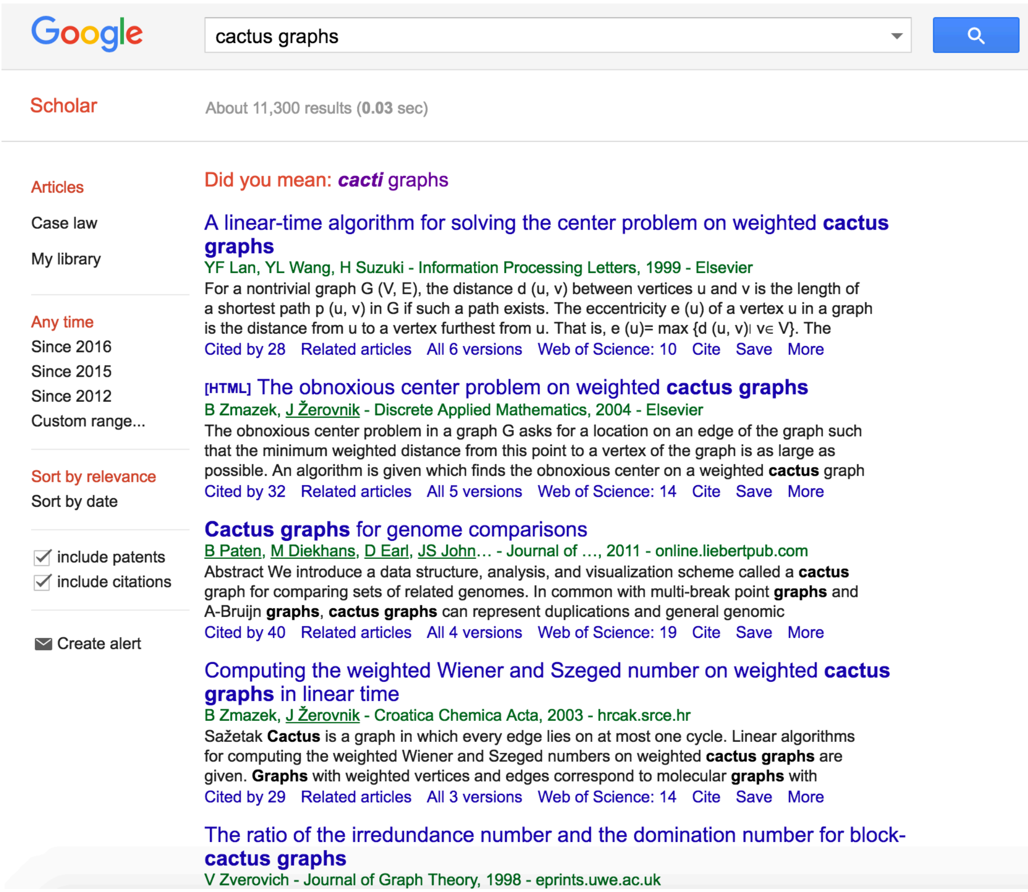
from *Enumeration of m-ary Cacti* (Bóna *et al.*)

plane (vs. **free**)
cactus

Prior Work


Google
cactus graphs
Scholar
About 11,300 results (0.03 sec)
Articles
Case law
My library
Any time
Since 2016
Since 2015
Since 2012
Custom range...
Sort by relevance
Sort by date
include patents
include citations
Create alert
Did you mean: cacti graphs
A linear-time algorithm for solving the center problem on weighted cactus graphs
YF Lan, YL Wang, H Suzuki - Information Processing Letters, 1999 - Elsevier
For a nontrivial graph G (V, E), the distance d (u, v) between vertices u and v is the length of a shortest path p (u, v) in G if such a path exists. The eccentricity e (u) of a vertex u in a graph is the distance from u to a vertex furthest from u. That is, e (u)= max {d (u, v)| v∈ V}. The
Cited by 28 Related articles All 6 versions Web of Science: 10 Cite Save More
[HTML] The obnoxious center problem on weighted cactus graphs
B Zmazek, J Žerovnik - Discrete Applied Mathematics, 2004 - Elsevier
The obnoxious center problem in a graph G asks for a location on an edge of the graph such that the minimum weighted distance from this point to a vertex of the graph is as large as possible. An algorithm is given which finds the obnoxious center on a weighted cactus graph
Cited by 32 Related articles All 5 versions Web of Science: 14 Cite Save More
Cactus graphs for genome comparisons
B Paten, M Diekhans, D Earl, JS John… - Journal of …, 2011 - online.liebertpub.com
Abstract We introduce a data structure, analysis, and visualization scheme called a cactus graph for comparing sets of related genomes. In common with multi-break point graphs and A-Bruijn graphs, cactus graphs can represent duplications and general genomic
Cited by 40 Related articles All 4 versions Web of Science: 19 Cite Save More
Computing the weighted Wiener and Szeged number on weighted cactus graphs in linear time
B Zmazek, J Žerovnik - Croatica Chemica Acta, 2003 - hrcak.srce.hr
Sažetak Cactus is a graph in which every edge lies on at most one cycle. Linear algorithms for computing the weighted Wiener and Szeged numbers on weighted cactus graphs are given. Graphs with weighted vertices and edges correspond to molecular graphs with
Cited by 29 Related articles All 3 versions Web of Science: 14 Cite Save More
The ratio of the irredundance number and the domination number for block-cactus graphs
V Zverovich - Journal of Graph Theory, 1998 - eprints.uwe.ac.uk

Prior Work



Google
cactus graphs

Scholar — About 11,300 results (**0.03** sec)

Articles
Case law
My library

Any time
Since 2016
Since 2015
Since 2012
Custom range...

Sort by relevance
Sort by date

include patents
include citations

Create alert

Did you mean: ***cacti*** graphs

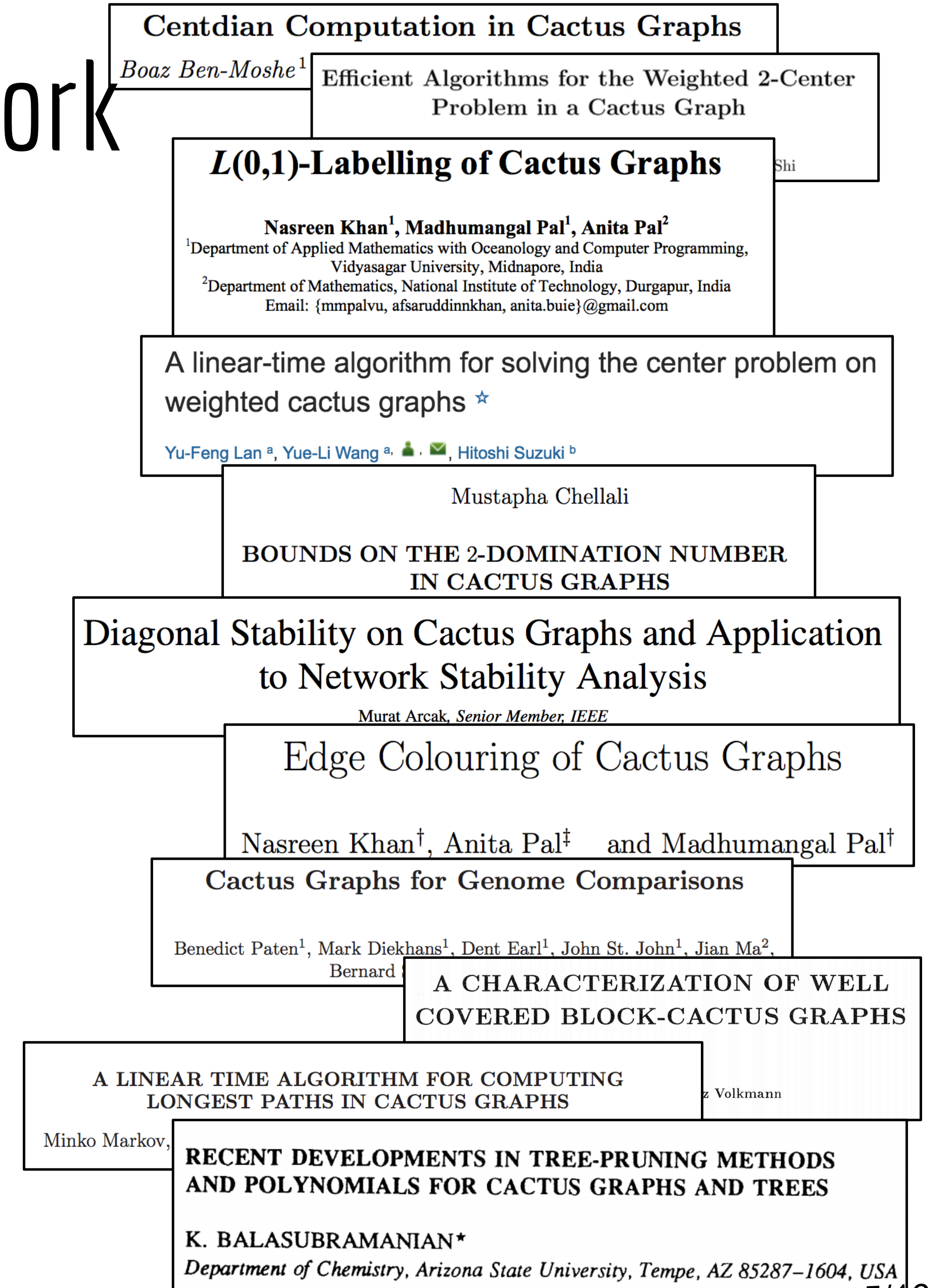
A linear-time algorithm for solving the center problem on weighted **cactus graphs**
YF Lan, YL Wang, H Suzuki - Information Processing Letters, 1999 - Elsevier
For a nontrivial graph G (V, E), the distance d (u, v) between vertices u and v is the length of a shortest path p (u, v) in G if such a path exists. The eccentricity e (u) of a vertex u in a graph is the distance from u to a vertex furthest from u. That is, e (u)= max {d (u, v)| v∈ V}. The
Cited by 28 Related articles All 6 versions Web of Science: 10 Cite Save More

[HTML] The obnoxious center problem on weighted **cactus graphs**
B Zmazek, J Žerovnik - Discrete Applied Mathematics, 2004 - Elsevier
The obnoxious center problem in a graph G asks for a location on an edge of the graph such that the minimum weighted distance from this point to a vertex of the graph is as large as possible. An algorithm is given which finds the obnoxious center on a weighted **cactus** graph
Cited by 32 Related articles All 5 versions Web of Science: 14 Cite Save More

Cactus graphs for genome comparisons
B Paten, M Diekhans, D Earl, JS John... - Journal of ..., 2011 - online.liebertpub.com
Abstract We introduce a data structure, analysis, and visualization scheme called a **cactus** graph for comparing sets of related genomes. In common with multi-break point **graphs** and A-Bruijn **graphs**, **cactus graphs** can represent duplications and general genomic
Cited by 40 Related articles All 4 versions Web of Science: 19 Cite Save More

Computing the weighted Wiener and Szeged number on weighted **cactus graphs** in linear time
B Zmazek, J Žerovnik - Croatica Chemica Acta, 2003 - hrcak.srce.hr
Sažetak **Cactus** is a graph in which every edge lies on at most one cycle. Linear algorithms for computing the weighted Wiener and Szeged numbers on weighted **cactus graphs** are given. **Graphs** with weighted vertices and edges correspond to molecular **graphs** with
Cited by 29 Related articles All 3 versions Web of Science: 14 Cite Save More

The ratio of the irredundance number and the domination number for block-**cactus graphs**
V Zverovich - Journal of Graph Theory, 1998 - eprints.uwe.ac.uk



Centdian Computation in Cactus Graphs
Boaz Ben-Moshe[1]

Efficient Algorithms for the Weighted 2-Center Problem in a Cactus Graph
Shi

***L*(0,1)-Labelling of Cactus Graphs**
Nasreen Khan[1], Madhumangal Pal[1], Anita Pal[2]
[1]Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore, India
[2]Department of Mathematics, National Institute of Technology, Durgapur, India
Email: {mmpalvu, afsaruddinnkhan, anita.buie}@gmail.com

A linear-time algorithm for solving the center problem on weighted cactus graphs ☆
Yu-Feng Lan [a], Yue-Li Wang [a], Hitoshi Suzuki [b]

Mustapha Chellali
BOUNDS ON THE 2-DOMINATION NUMBER IN CACTUS GRAPHS

Diagonal Stability on Cactus Graphs and Application to Network Stability Analysis
Murat Arcak, *Senior Member, IEEE*

Edge Colouring of Cactus Graphs
Nasreen Khan†, Anita Pal‡ and Madhumangal Pal†

Cactus Graphs for Genome Comparisons
Benedict Paten[1], Mark Diekhans[1], Dent Earl[1], John St. John[1], Jian Ma[2],
Bernard

A CHARACTERIZATION OF WELL COVERED BLOCK-CACTUS GRAPHS
Volkmann

A LINEAR TIME ALGORITHM FOR COMPUTING LONGEST PATHS IN CACTUS GRAPHS
Minko Markov,

RECENT DEVELOPMENTS IN TREE-PRUNING METHODS AND POLYNOMIALS FOR CACTUS GRAPHS AND TREES
K. BALASUBRAMANIAN*
Department of Chemistry, Arizona State University, Tempe, AZ 85287–1604, USA

Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— proposed method for enumerating **free, unlabeled** cacti
- derived functional equations for **3-** and **4-**cacti.

— promised to provide "a more systematic treatment of the general case of *pure k-cacti*" in a subsequent paper
- it appears they never published such a paper

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

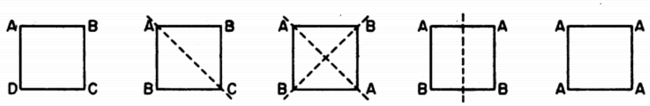


FIG. 2

If $c_1^*, \ldots, c_5^*$ denote again the number of dissimilar quadrilaterals of these symmetry types occurring in a given tree, then one has

$$\begin{aligned} l^* &= 4c_1^* + 2c_2^* + c_3^* + 3c_4^* + c_5^* \\ c^* &= c_1^* + c_2^* + c_3^* + c_4^* + c_5^* \\ c_{pp}^* &= c_2^* + c_3^* + c_5^* \\ c_u^* &= c_4^* + c_5^*. \end{aligned}$$

Thus one gets from (6), since a is of course again zero:

$$1 = p^* - 3c_1^* - 2c_2^* - c_3^* - c_4^*. \qquad (6c)$$

Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— proposed method for enumerating **free, unlabeled** cacti
- derived functional equations for **3-** and **4-**cacti.

— promised to provide "a more systematic treatment of the general case of *pure k-cacti*" in a subsequent paper
- it appears they never published such a paper

Enumeration of m-ary cacti
Miklós Bóna *et al.* (1999):

— enumerated **pure, plane, unlabeled** cacti.

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

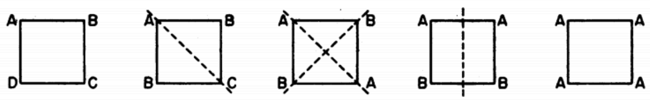


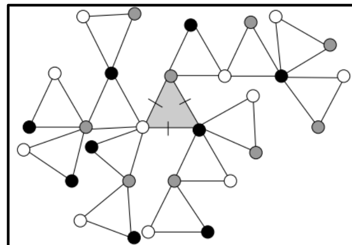
FIG. 2

If $c_1^*, \ldots, c_5^*$ denote again the number of dissimilar quadrilaterals of these symmetry types occurring in a given tree, then one has

$$\begin{aligned} l^* &= 4c_1^* + 2c_2^* + c_3^* + 3c_4^* + c_5^* \\ c^* &= c_1^* + c_2^* + c_3^* + c_4^* + c_5^* \\ c_{pp}^* &= c_2^* + c_3^* + c_5^* \\ c_u^* &= c_4^* + c_5^*. \end{aligned}$$

Thus one gets from (6), since a is of course again zero:

$$1 = p^* - 3c_1^* - 2c_2^* - c_3^* - c_4^*. \tag{6c}$$



$$\widetilde{\mathcal{K}}_{=s,n} = \frac{s}{p}\sum_{d|\frac{p}{s}} \mu(d)\binom{pm/sd}{p/sd} \tag{77}$$

and

$$\widetilde{\mathcal{K}}_{\geq s,n} = \frac{s}{p}\sum_{d|\frac{p}{s}} \phi(d)\binom{pm/sd}{p/sd}; \tag{78}$$

$$\widetilde{\mathcal{K}}_{=s,\vec{n}} = \sum_{i=1}^{m} \frac{s(p-n_i+1)}{p^2}\sum_{d} \mu(d/s)\binom{p/d}{(n_i-1)/d}\prod_{j\neq i}\binom{p/d}{n_j/d}, \tag{79}$$

and

$$\widetilde{\mathcal{K}}_{\geq s,\vec{n}} = \sum_{i=1}^{m} \frac{s(p-n_i+1)}{p^2}\sum_{d} \phi(d/s)\binom{p/d}{(n_i-1)/d}\prod_{j\neq i}\binom{p/d}{n_j/d}, \tag{80}$$

the second summations being taken over all integers d such that $s|d$ and d divides p and all components of $\vec{n}-\vec{e}_i$;

$$\widetilde{\mathcal{K}}_{=s,N} = \sum_{i=1}^{m} \frac{p^{m-2}s}{\prod_{j\neq i} n_j}\sum_{h,d} \mu(d/s)\binom{(n_i-1)/d}{(\mathbf{n}_i-\mathbf{e}_h)/d}\prod_{j\neq i}\binom{n_j/d}{\mathbf{n}_j/d}, \tag{81}$$

and

$$\widetilde{\mathcal{K}}_{\geq s,N} = \sum_{i=1}^{m} \frac{p^{m-2}s}{\prod_{j\neq i} n_j}\sum_{h,d} \phi(d/s)\binom{(n_1-1)/d}{(\mathbf{n}_i-\mathbf{e}_h)/d}\prod_{j\neq i}\binom{n_j/d}{\mathbf{n}_j/d}, \tag{82}$$

the second sommations being taken over all pairs of integers $h, d \geq 1$ such that $n_{ih} \neq 0, s|d$, and d divides h and all entries in $N - E_{ih}$. □

Prior Work

On the Number of Husimi Trees
Harary and Uhlenbeck (1952):

— proposed method for enumerating **free, unlabeled** cacti
- derived functional equations for **3-** and **4-**cacti.

— promised to provide "a more systematic treatment of the general case of *pure k-cacti*" in a subsequent paper
- it appears they never published such a paper

For a pure Husimi tree consisting of quadrilaterals one has five types of symmetry, illustrated by:

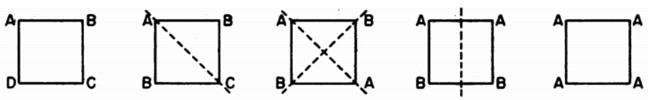


FIG. 2

If $c_1^*, \ldots, c_5^*$ denote again the number of dissimilar quadrilaterals of these symmetry types occurring in a given tree, then one has

not easily generalizable

$$\begin{aligned} l^* &= 4c_1^* + 2c_2^* + c_3^* + 3c_4^* + c_5^* \\ c^* &= c_1^* + c_2^* + c_3^* + c_4^* + c_5^* \\ c_{pp}^* &= c_2^* + c_3^* + c_5^* \\ c_u^* &= c_4^* + c_5^*. \end{aligned}$$

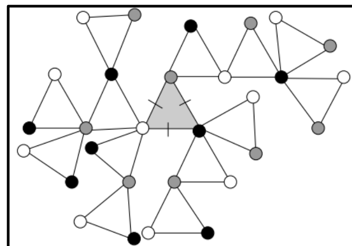
hard to extract

Thus one gets from (6), since a is of course again zero:

$$1 = p^* - 3c_1^* - 2c_2^* - c_3^* - c_4^*. \qquad (6c)$$

Enumeration of m-ary cacti
Miklós Bóna *et al.* (1999):

— enumerated **pure, plane, unlabeled** cacti.



only *plane* cacti

complicated methods

$$\widetilde{\mathcal{K}}_{=s,n} = \frac{s}{p}\sum_{d|\frac{p}{s}} \mu(d)\binom{pm/sd}{p/sd} \qquad (77)$$

and

$$\widetilde{\mathcal{K}}_{\geq s,n} = \frac{s}{p}\sum_{d|\frac{p}{s}} \phi(d)\binom{pm/sd}{p/sd}; \qquad (78)$$

$$\widetilde{\mathcal{K}}_{=s,\vec{n}} = \sum_{i=1}^{m} \frac{s(p-n_i+1)}{p^2}\sum_{d} \mu(d/s)\binom{p/d}{(n_i-1)/d}\prod_{j\neq i}\binom{p/d}{n_j/d}, \qquad (79)$$

and

$$\widetilde{\mathcal{K}}_{\geq s,\vec{n}} = \sum_{i=1}^{m} \frac{s(p-n_i+1)}{p^2}\sum_{d} \phi(d/s)\binom{p/d}{(n_i-1)/d}\prod_{j\neq i}\binom{p/d}{n_j/d}, \qquad (80)$$

the second summations being taken over all integers d such that $s|d$ and d divides p and all components of $\vec{n} - \vec{e}_i$;

$$\widetilde{\mathcal{K}}_{=s,N} = \sum_{i=1}^{m} \frac{p^{m-2}s}{\prod_{j\neq i} n_j}\sum_{h,d} \mu(d/s)\binom{(n_i-1)/d}{(\mathbf{n}_i-\mathbf{e}_h)/d}\prod_{j\neq i}\binom{n_j/d}{\mathbf{n}_j/d}, \qquad (81)$$

and

$$\widetilde{\mathcal{K}}_{\geq s,N} = \sum_{i=1}^{m} \frac{p^{m-2}s}{\prod_{j\neq i} n_j}\sum_{h,d} \phi(d/s)\binom{(n_1-1)/d}{(\mathbf{n}_i-\mathbf{e}_h)/d}\prod_{j\neq i}\binom{n_j/d}{\mathbf{n}_j/d}, \qquad (82)$$

the second sommations being taken over all pairs of integers $h,d \geq 1$ such that $n_{ih} \neq 0, s|d$, and d divides h and all entries in $N - E_{ih}$. □

New Result

Exact enumeration of **unlabeled, non-plane, pure** *n*-cacti.

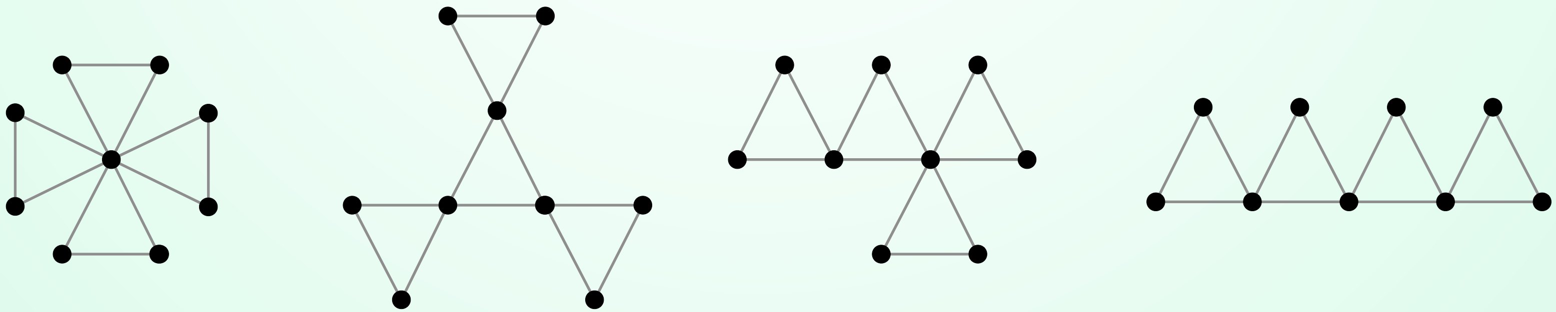
$n = 3$	$0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, \ldots$
$n = 4$	$0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, \ldots$
$n = 5$	$0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, \ldots$
$n = 6$	$0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, \ldots$

New Result

Exact enumeration of **unlabeled, non-plane, pure** *n*-cacti.

$n=3$	$0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, \ldots$
$n=4$	$0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, \ldots$
$n=5$	$0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, \ldots$
$n=6$	$0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, \ldots$

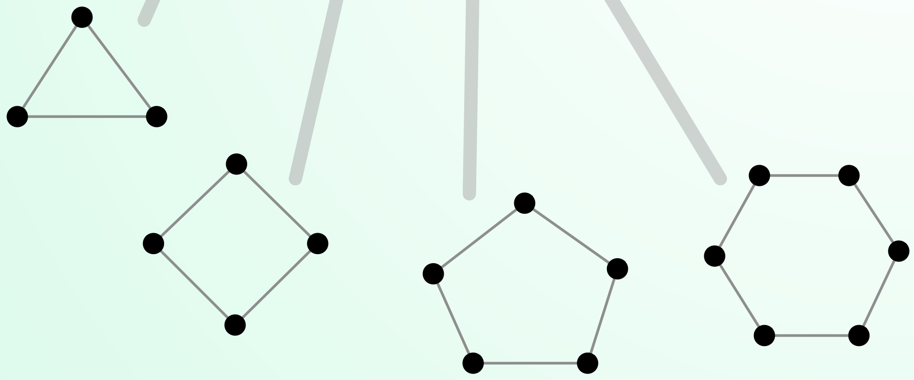
Number of pure 3-cacti with 9 vertices



New Result

Exact enumeration of **unlabeled, non-plane, pure** *n*-cacti.

$n = 3$	$0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, \ldots$
$n = 4$	$0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, \ldots$
$n = 5$	$0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, \ldots$
$n = 6$	$0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, \ldots$



The first non-zero term is always 1
(corresponding to polygon)

New Result

Exact enumeration of **unlabeled, non-plane, pure** *n*-cacti.

$n = 3$	$0, 0, 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 19, 0, 48, 0, 126, 0, 355, 0, 1037, \ldots$
$n = 4$	$0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 7, 0, 0, 25, 0, 0, 88, 0, 0, 366, 0, \ldots$
$n = 5$	$0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0, 0, 132, \ldots$
$n = 6$	$0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 13, 0, 0, 0, 0, 67, \ldots$

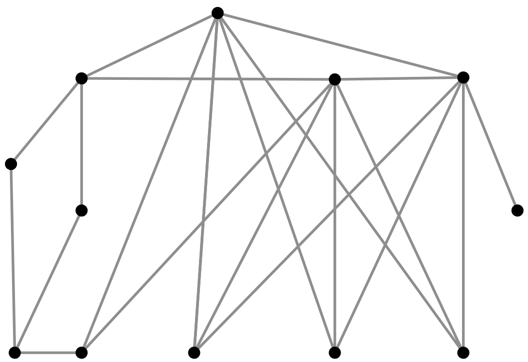
Our approach is simpler and more general than Bóna *et al.*:

— can easily be extended to derive their result (**plane** cacti)

e.g. plane 5-cacti: $0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 17, 0, 0, 0, 102, \ldots$

— methodology applicable to obtain many variations of cacti, including **mixed** cacti

Methodolgy: Overview

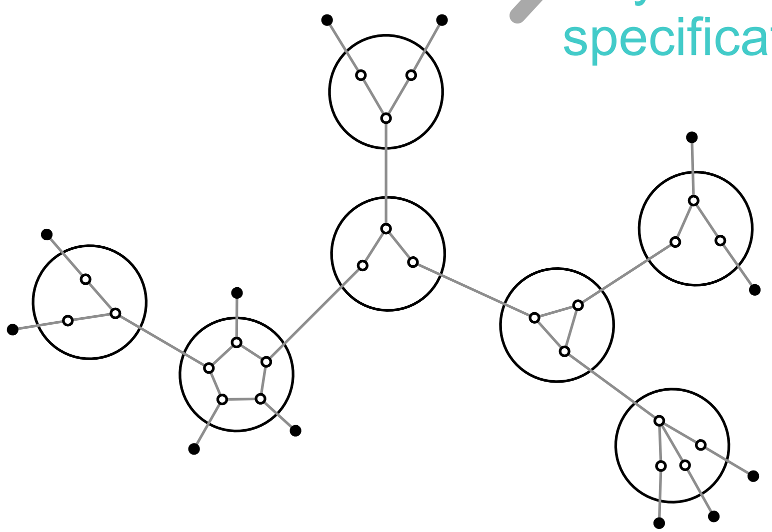


split
decomposition

$$\mathcal{G} = \mathcal{Z} \times (\mathcal{P} + \mathcal{S}_C)$$
$$\mathcal{P} = \textsc{Seq}_{=4}(\mathcal{Z} + \mathcal{S}_X)$$
$$\mathcal{S}_X = \mathcal{Z} \times \textsc{Seq}_{\geqslant 1}(\mathcal{P})$$
$$\mathcal{S}_C = \textsc{Cyc}_{\geqslant 2}(\mathcal{P})$$

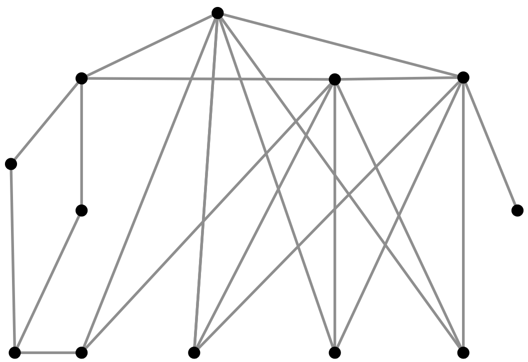
symbolic
specification

computer algebra
system (CAS)

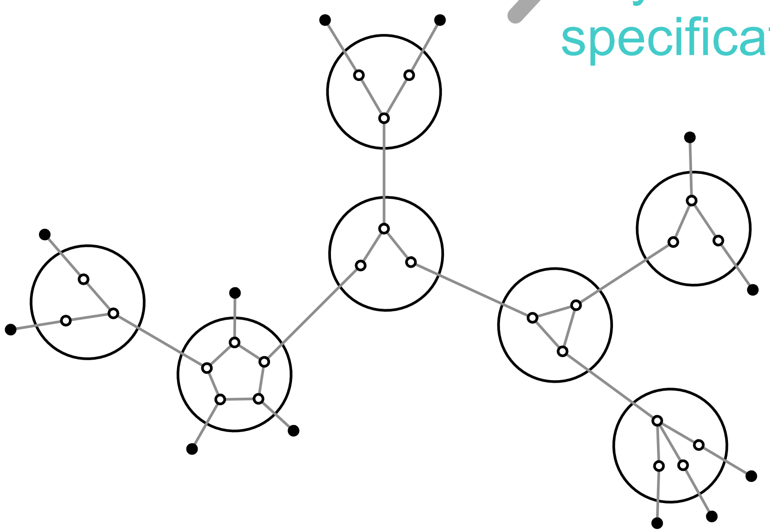


$0, 0, 1, 0, 1, 0, 2, 0,$
$4, 0, 8, 0, 19, 0, 48,$
$0, 126, 0, 355, 0,$
$1037, \ldots$

Methodolgy: Overview



split decomposition



symbolic specification

$$\mathcal{G} = \mathcal{Z} \times (\mathcal{P} + \mathcal{S}_C)$$
$$\mathcal{P} = \mathrm{S{\scriptstyle EQ}}_{=4}(\mathcal{Z} + \mathcal{S}_X)$$
$$\mathcal{S}_X = \mathcal{Z} \times \mathrm{S{\scriptstyle EQ}}_{\geqslant 1}(\mathcal{P})$$
$$\mathcal{S}_C = \mathrm{C{\scriptstyle YC}}_{\geqslant 2}(\mathcal{P})$$

computer algebra system (CAS)

$0, 0, 1, 0, 1, 0, 2, 0,$
$4, 0, 8, 0, 19, 0, 48,$
$0, 126, 0, 355, 0,$
$1037, \ldots$

An Exact Enumeration of Distance-Hereditary Graphs

Cédric Chauve* Éric Fusy† Jérémie Lumbroso‡

Abstract

Distance-hereditary graphs form an importan[t] ... graphs, from the theoretical point of view, ... fact that they are the totally decomposable grap[hs] ...

Theorem 4. *The class $\mathcal{DH}$ of unrooted distance-hereditary graphs is specified by*

$$\mathcal{DH} = \mathcal{T}_K + \mathcal{T}_S + \mathcal{T}_{S-S} - \mathcal{T}_{K-S} - \mathcal{T}_{S\to S} \quad (3.25)$$
$$\mathcal{T}_K = \mathrm{SET}_{\geqslant 3}(\mathcal{Z} + \mathcal{S}_C + \mathcal{S}_X) \quad (3.26)$$
$$\mathcal{T}_S = (\mathcal{Z} + \mathcal{K} + \mathcal{S}_C) \times \mathcal{S}_C \quad (3.27)$$
$$\mathcal{T}_{K-S} = \mathcal{K} \times (\mathcal{S}_C + \mathcal{S}_X) \quad (3.28)$$
$$\mathcal{T}_{S-S} = \mathrm{SET}_2(\mathcal{S}_C) + \mathrm{SET}_2(\mathcal{S}_X) \quad (3.29)$$
$$\mathcal{T}_{S\to S} = \mathcal{S}_C \times \mathcal{S}_C + \mathcal{S}_X \times \mathcal{S}_X \quad (3.30)$$
$$\mathcal{K} = \mathrm{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{S}_C + \mathcal{S}_X) \quad (3.31)$$
$$\mathcal{S}_C = \mathrm{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_X) \quad (3.32)$$
$$\mathcal{S}_X = \mathrm{SEQ}_{\geqslant 2}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_C). \quad (3.33)$$

Enumerations, Forbidden Subgraph Characterizations, and the Split-Decomposition

Maryam Bahrani* Jérémie Lumbroso*

Abstract

Forbidden characterizations may sometimes be the most natural way to describe families of graphs, and yet these characterizations are usually very hard to exploit for enumerative purposes.

By building on the work of Gioan and Paul (2012) and Chauve et al. (2014), we show a methodology by which we constrain a split-decomposition tree to avoid certain patterns, thereby avoiding the corresponding induced subgraphs in the ...

As far as we know, while these notions are part and parcel of the work of graph theorists, they are usually not exploited by analytic combinatorists. For forbidden minors, there is the penetrating article of Bousquet-Mélou and Weller [4]. For forbidden subgraphs or forbidden induced subgraphs, we know of few papers, except because of the simple nature of graphs [31], or because some other, alternate property is used instead [5], or only asymptotics are determined [32].

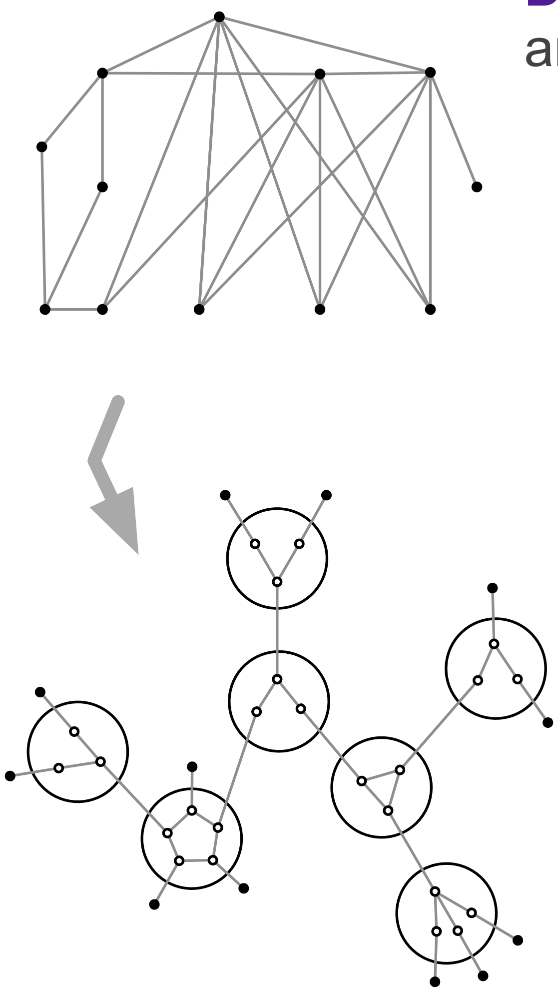
We are concerned, in this paper, with *forbidden induced* ...

... n induced subgraphs. ... ively well-known graph ... omposition, could be a ... n class called *distance-* ... meration had until then ... wn result was the bound ... d that there are at most ... y graphs on n vertices). ... version of this split- ... and Gioan, with inter- ... ved the legibility of the

Theorem 5. *The class $\mathcal{PG}_\bullet$ of ptolemaic graphs rooted at a vertex is specified by*

$$\mathcal{PG}_\bullet = \mathcal{Z}_\bullet \times (\mathcal{S}_C + \mathcal{S}_X + \mathcal{K}) \quad (4.15)$$
$$\mathcal{S}_C = \mathrm{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_X) \quad (4.16)$$
$$\mathcal{S}_X = (\mathcal{Z} + \overline{\mathcal{K}}) \times \mathrm{SET}_{\geqslant 1}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_X) \quad (4.17)$$
$$\mathcal{K} = \mathcal{S}_C \times \mathrm{SET}_{\geqslant 1}(\mathcal{Z} + \mathcal{S}_X) + \mathrm{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{S}_X) \quad (4.18)$$
$$\overline{\mathcal{K}} = \mathrm{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{S}_X) \quad (4.19)$$

Methodology: The Split Decomposition

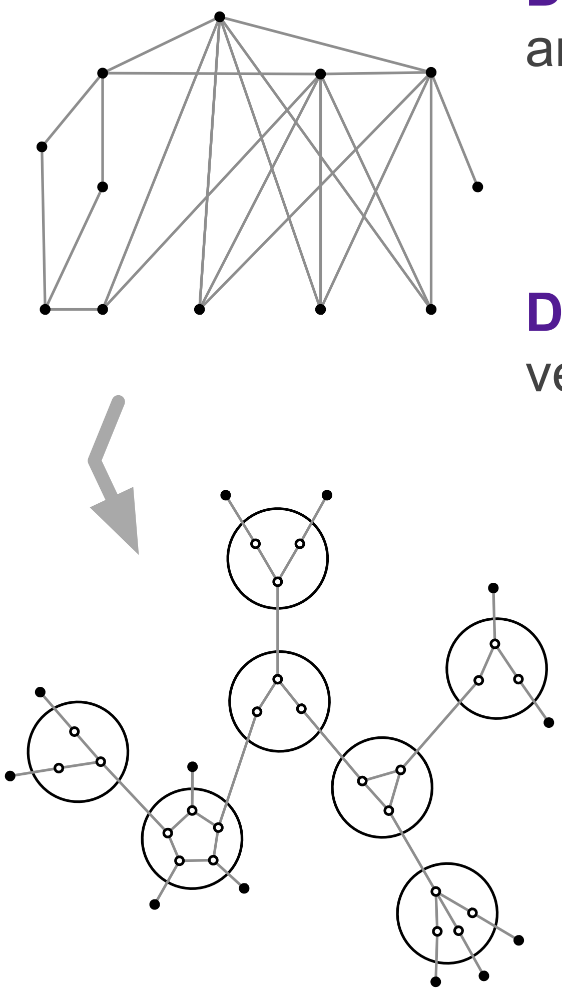


Def. A *graph-labeled tree* is a pair $(T, \mathcal{F})$, where T is a tree and $\mathcal{F}$ is a family of graphs, such that

— Every tree node $v \in V(T)$ is *labeled* with a graph $G_v \in \mathcal{F}$

— There is exactly one tree-edge for every vertex of G_v

Methodology: The Split Decomposition

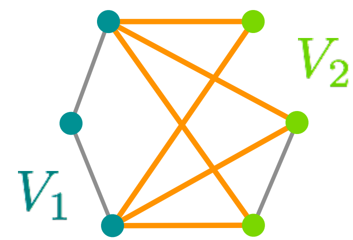


Def. A *graph-labeled tree* is a pair $(T, \mathcal{F})$, where T is a tree and $\mathcal{F}$ is a family of graphs, such that

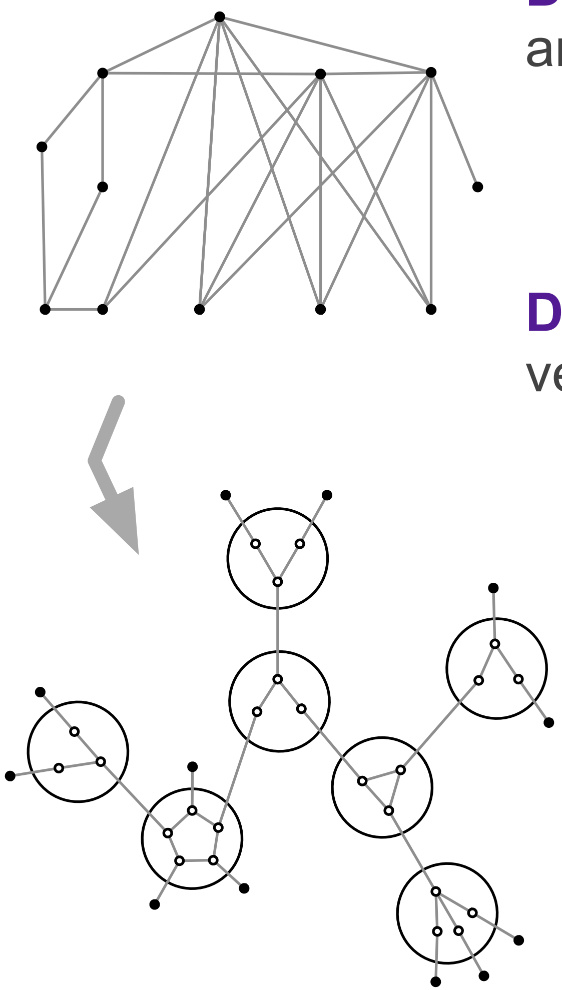
— Every tree node $v \in V(T)$ is *labeled* with a graph $G_v \in \mathcal{F}$
— There is exactly one tree-edge for every vertex of G_v

Def. A *split* in a graph is a bipartition of the vertices into two subsets V_1 and V_2 such that

— Each side has at least size 2
— The edges crossing the bipartition induce a complete bipartite graph.



Methodology: The Split Decomposition



Def. A *graph-labeled tree* is a pair $(T, \mathcal{F})$, where T is a tree and $\mathcal{F}$ is a family of graphs, such that

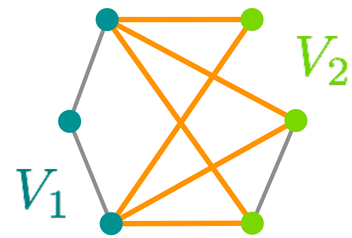
— Every tree node $v \in V(T)$ is *labeled* with a graph $G_v \in \mathcal{F}$

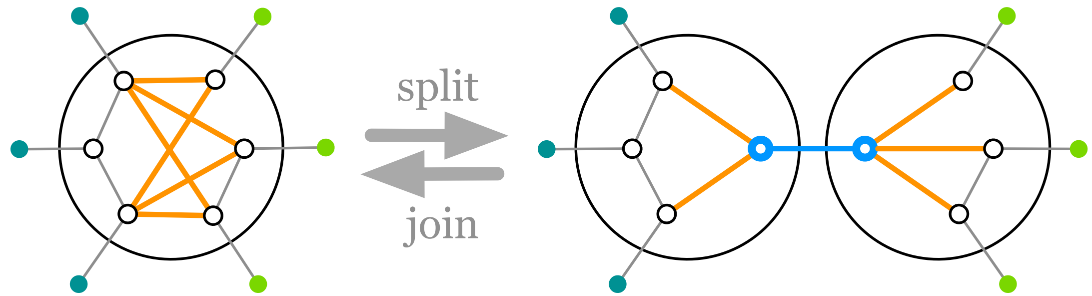
— There is exactly one tree-edge for every vertex of G_v

Def. A *split* in a graph is a bipartition of the vertices into two subsets V_1 and V_2 such that

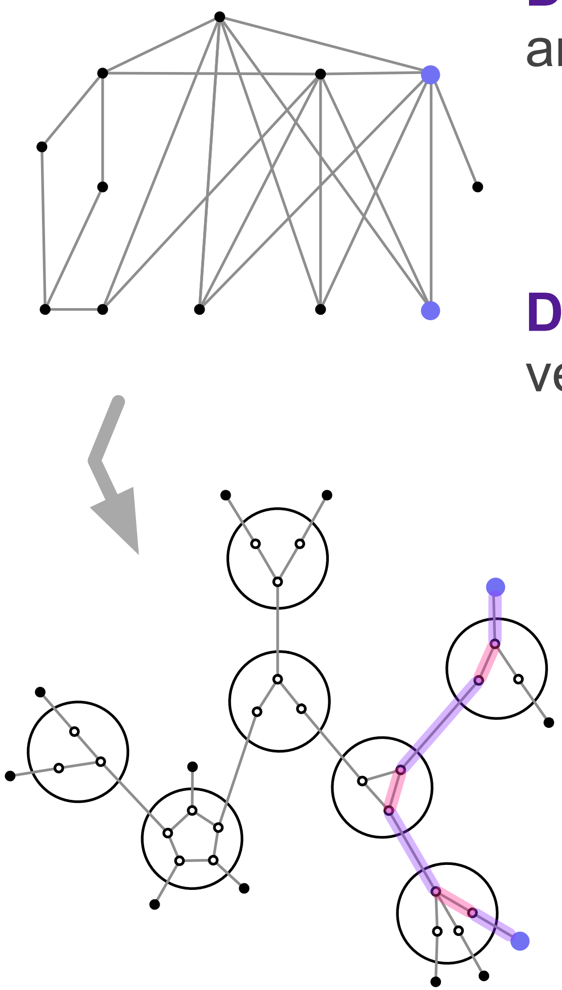
— Each side has at least size 2

— The edges crossing the bipartition induce a complete bipartite graph.





Methodology: The Split Decomposition



Def. A *graph-labeled tree* is a pair $(T, \mathcal{F})$, where T is a tree and $\mathcal{F}$ is a family of graphs, such that

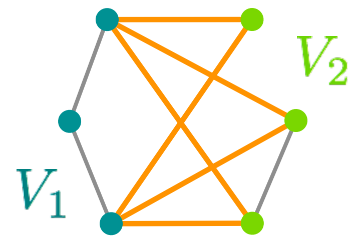
— Every tree node $v \in V(T)$ is *labeled* with a graph $G_v \in \mathcal{F}$

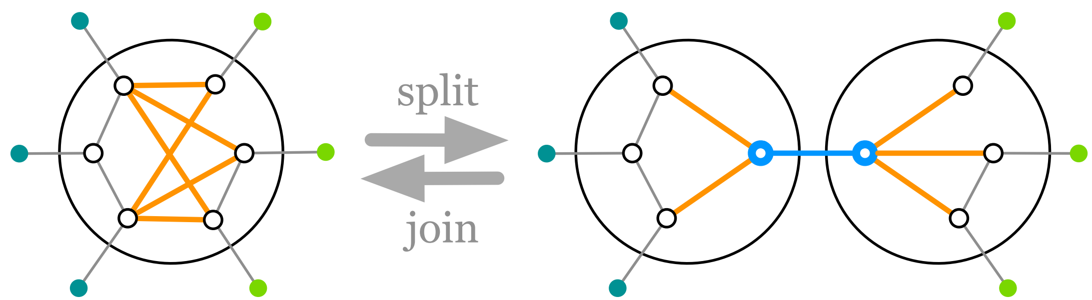
— There is exactly one tree-edge for every vertex of G_v

Def. A *split* in a graph is a bipartition of the vertices into two subsets V_1 and V_2 such that

— Each side has at least size 2

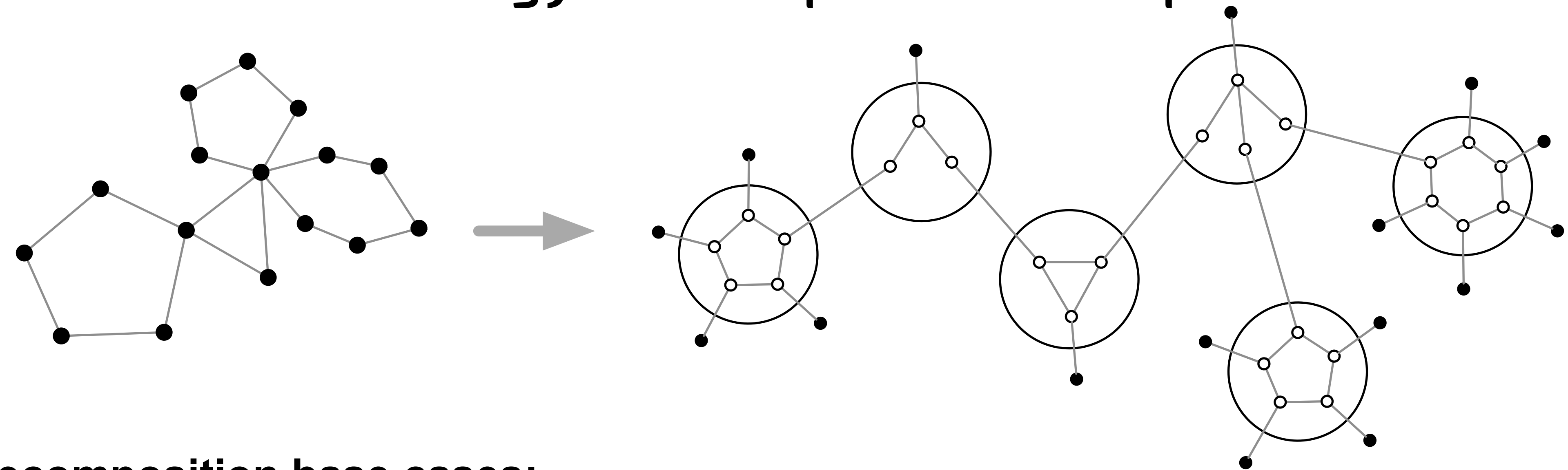
— The edges crossing the bipartition induce a complete bipartite graph.





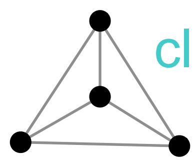
— Can read adjacencies from *alternated paths*.

Methodology: The Split Decomposition

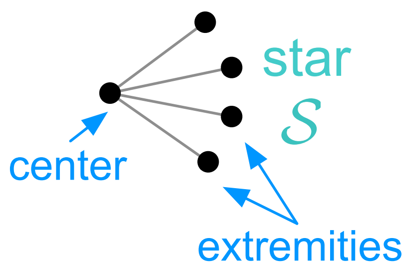


Decomposition base cases:

— *degenerate* nodes**:**



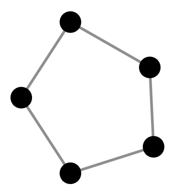
clique
$\mathcal{K}$



star
$\mathcal{S}$

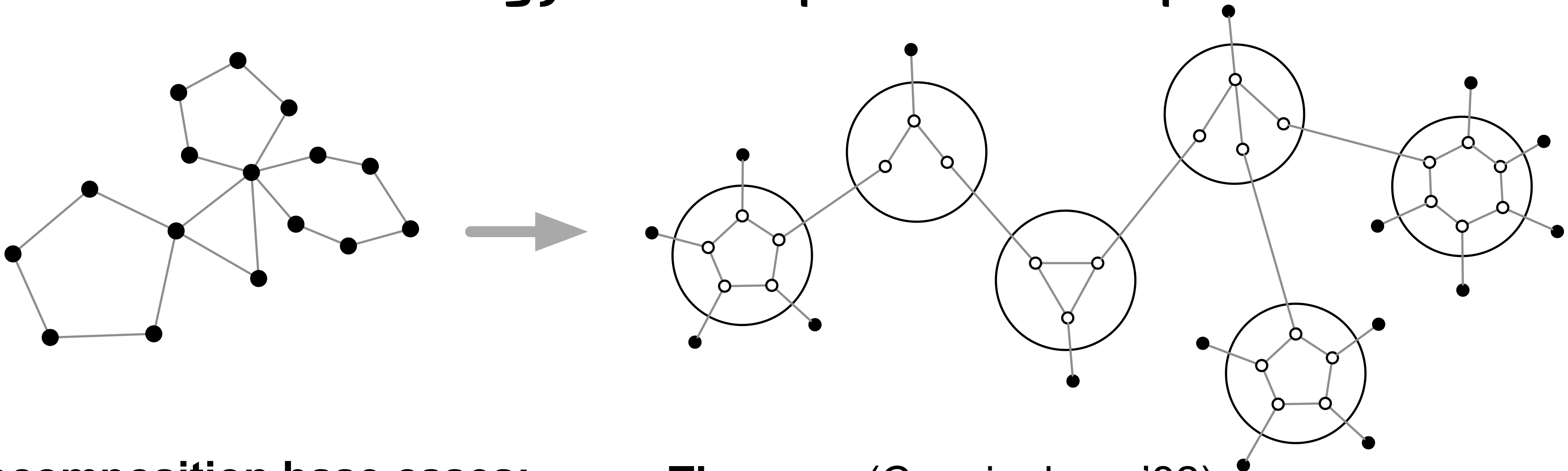
— *prime* nodes:

e.g.



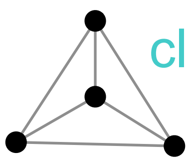
cycle
$\mathcal{P}$

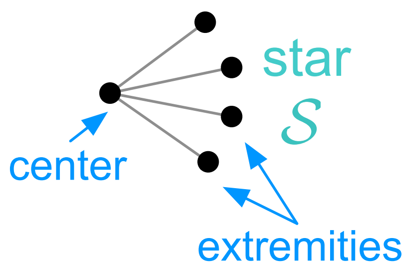
Methodology: The Split Decomposition



Decomposition base cases:

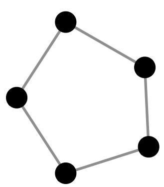
— *degenerate* nodes:





— *prime* nodes:

e.g.

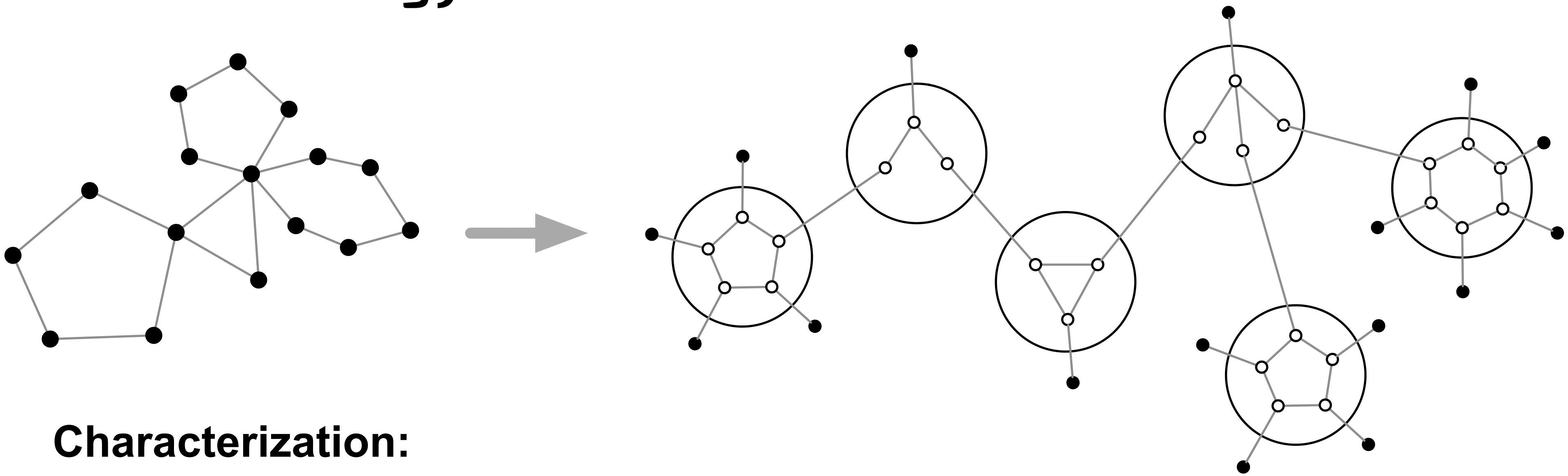


Theorem (Cunningham ’82):
The split decomposition tree into *prime* and *degenerate* nodes is unique as long as certain conditions are met.

Theorem:
Cycles of size at least 5 are prime nodes.

— Gives a *bijection* between cactus graphs and families of graph-labelled trees

Methodology: Characterization and Grammar



Characterization:

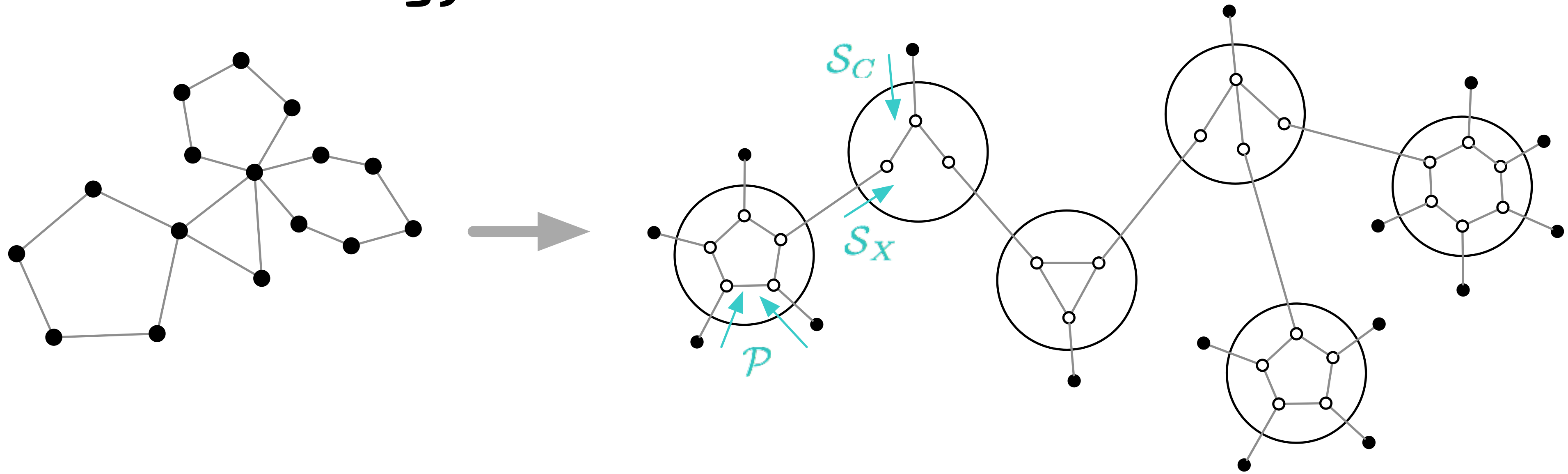
Cactus graphs can are in bijection with graph-labeled trees where

— internal nodes are stars and polygons;
— no polygons are adjacent;
— the centers of star nodes are attached to leaves;
— the extremities of star nodes attached to polygons.

includes leaves for 2-cycles

This characterization can be captured using a symbolic grammar.

Methodology: Characterization and Grammar



Grammar (unlabeled free pure k-cacti):

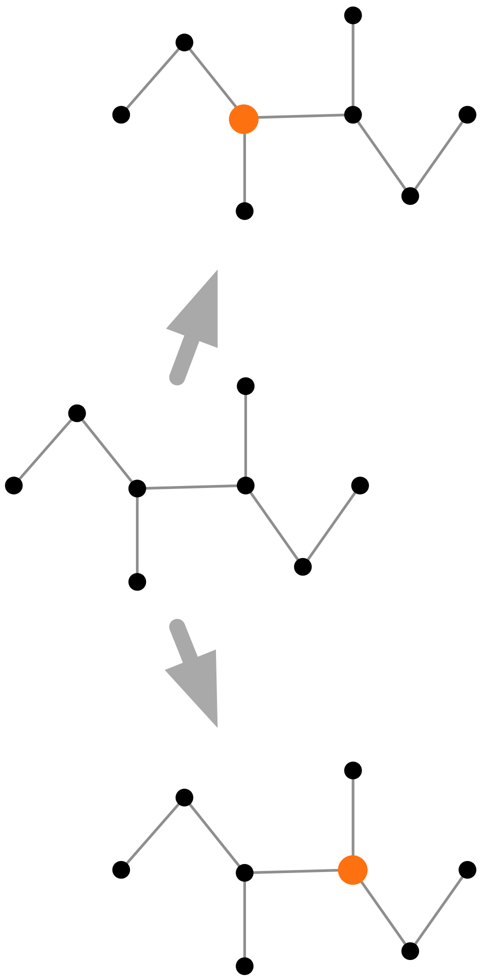
$$\mathcal{CG}_\bullet = \mathcal{Z}_\bullet \times (\mathcal{P} + \mathcal{S}_C)$$
$$\mathcal{P} = \mathrm{USEQ}_{k-1}(\mathcal{Z} + \mathcal{S}_X)$$
$$\mathcal{S}_C = \mathrm{SET}_{\geqslant 2}(\mathcal{P})$$
$$\mathcal{S}_X = \mathcal{Z} \times \mathrm{SET}_{\geqslant 1}(\mathcal{P})$$

$\mathcal{CG}_\bullet$	k-cactus graph rooted at a vertex
$\mathcal{Z}_\bullet$	distinguished leaf
$\mathcal{P}$	polygon entered from a subtree
$\mathcal{S}_X$	star entered from an extremity
$\mathcal{S}_C$	star entered from its center
$\mathrm{SET}_{=n}(\mathcal{A})$	set of n (unordered) elements from $\mathcal{A}$
$\mathrm{USEQ}_{=n}(\mathcal{A})$	undirected sequence of n elements from $\mathcal{A}$

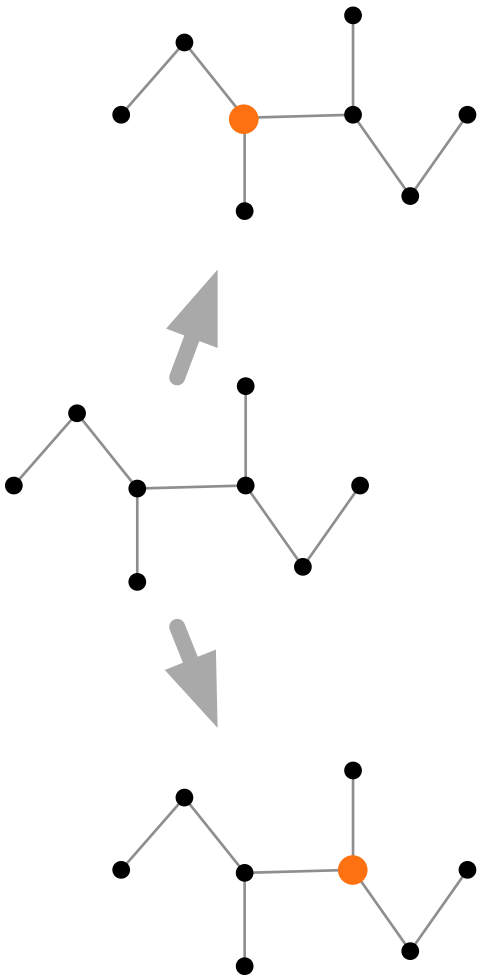
Methodology: Unrooting Subtleties

Where do we start decomposing from?

— unlabeled structures have *symmetries*
— different set of symmetries for different starting points (“roots”)



Methodology: Unrooting Subtleties



Where do we start decomposing from?

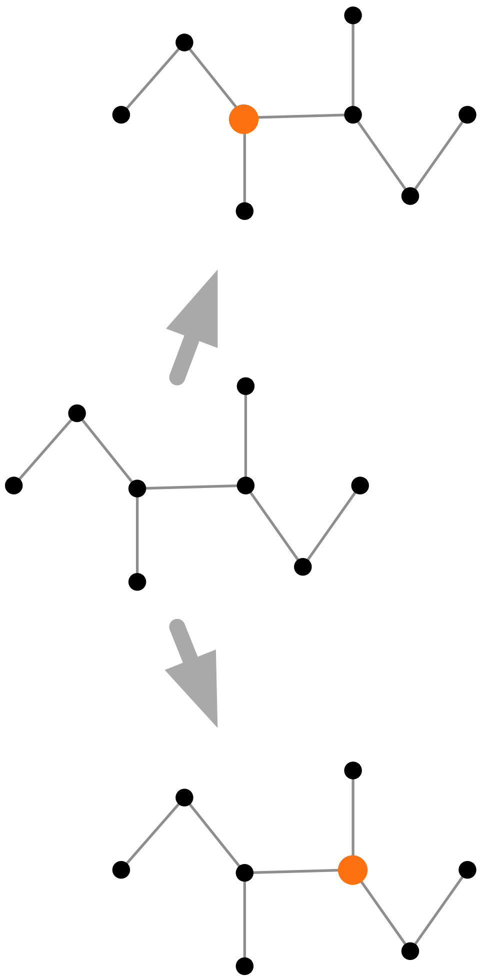
— unlabeled structures have *symmetries*
— different set of symmetries for different starting points (“roots”)

Dissymmetry theorem (Bergeron *et al. 98*):

— allows us to correct for symmetries of trees
— proof by observing that the tree center (midpoint of diameter) is distinguished by definition

$$\mathcal{A} + \mathcal{A}_{\circ\to\circ} \simeq \mathcal{A}_{\circ} + \mathcal{A}_{\circ-\circ}$$

Methodology: Unrooting Subtleties



Where do we start decomposing from?

— unlabeled structures have *symmetries*
— different set of symmetries for different starting points (“roots”)

Dissymmetry theorem (Bergeron *et al. 98*):

— allows us to correct for symmetries of trees
— proof by observing that the tree center (midpoint of diameter) is distinguished by definition

$$\mathcal{A} \quad + \quad \mathcal{A}_{\circ\rightarrow\circ} \quad \simeq \quad \mathcal{A}_{\circ} \quad + \quad \mathcal{A}_{\circ-\circ}$$

Cycle-pointing (based on Pólya theory):

— allows us to correct for symmetries of general graphs
— more difficult but preserves combinatorial nature of grammar (*eg.* can be used to build random samplers)

Verification

Verifying the enumeration:

— proof of the characterization

Theorem 10 (split-decomposition tree characterization of 3-cacti)**.** *A graph G with the reduced split-decomposition tree $(T, \mathcal{F})$ is a triangular cactus graph if and only if*

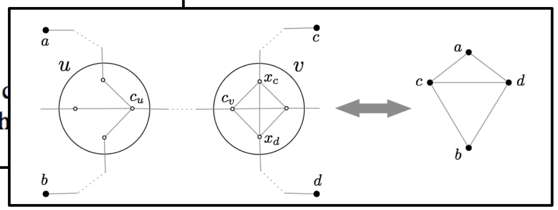
(a) T is a clique-star tree;

(b) the centers of all star-nodes are attached to leaves;

(c) the extremities of star-nodes are only attached to clique-nodes;

(d) every clique-node has degree 3.

Proof. By Lemma 12, we know that 3-cacti c
exactly as the class of block graphs with
induced $K_{\geqslant}4$.



Verification

Verifying the enumeration:

— proof of the characterization

Theorem 10 (split-decomposition tree characterization of 3-cacti)**.** *A graph G with the reduced split-decomposition tree $(T, \mathcal{F})$ is a triangular cactus graph if and only if*

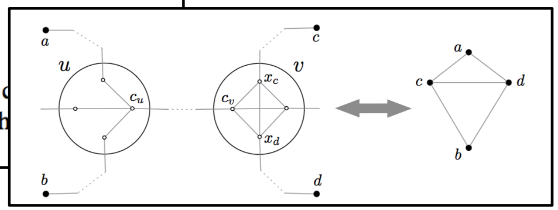
(a) T is a clique-star tree;

(b) the centers of all star-nodes are attached to leaves;

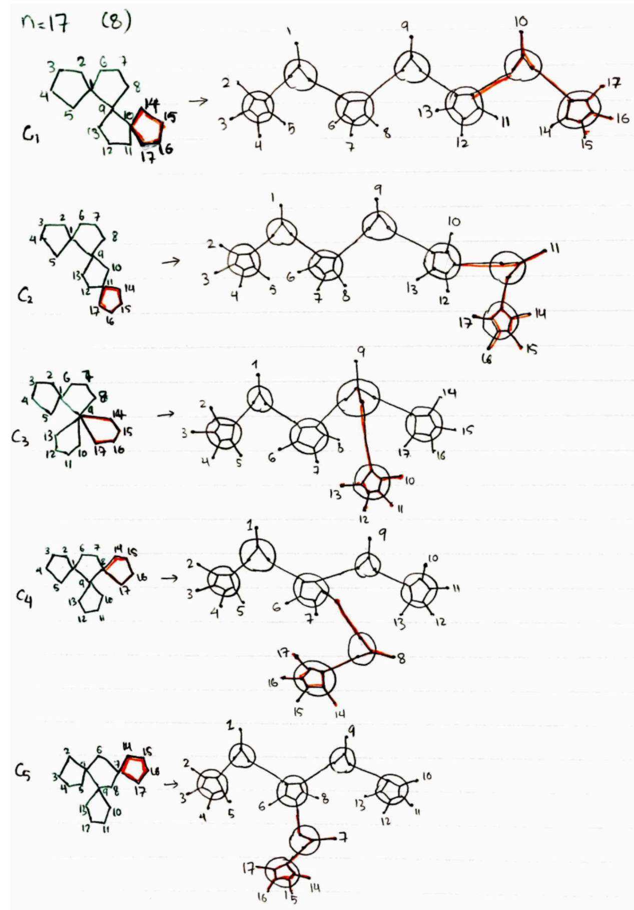
(c) the extremities of star-nodes are only attached to clique-nodes;

(d) every clique-node has degree 3.

Proof. By Lemma 12, we know that 3-cacti c exactly as the class of block graphs with induced $K_{\geqslant}4$.



— manual generation of small instances



Verification

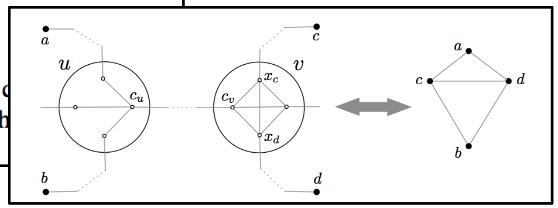
Verifying the enumeration:

— proof of the characterization

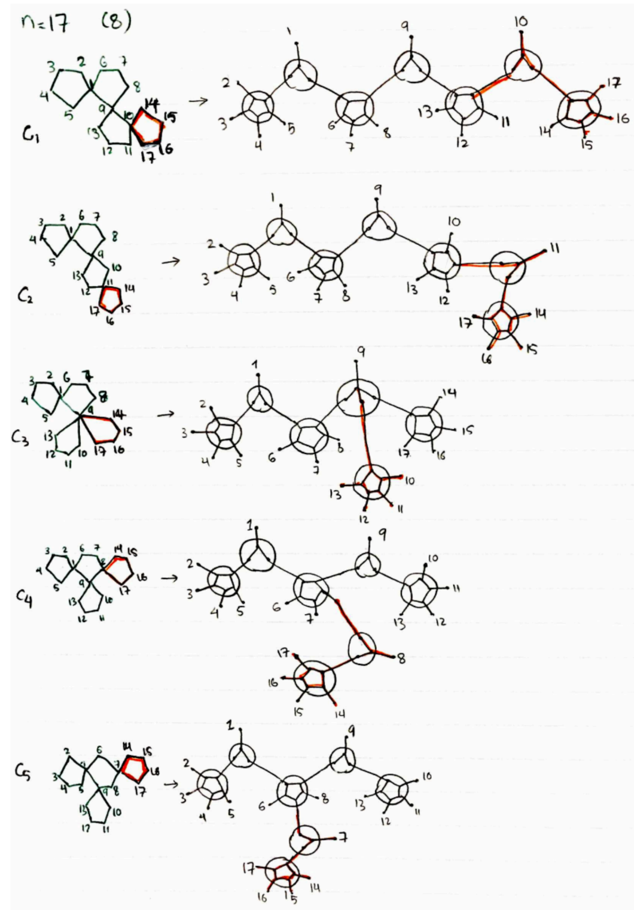
Theorem 10 (split-decomposition tree characterization of 3-cacti)**.** *A graph G with the reduced split-decomposition tree $(T, \mathcal{F})$ is a triangular cactus graph if and only if*

(a) *T is a clique-star tree;*
(b) *the centers of all star-nodes are attached to leaves;*
(c) *the extremities of star-nodes are only attached to clique-nodes;*
(d) *every clique-node has degree 3.*

Proof. By Lemma 12, we know that 3-cacti (…) exactly as the class of block graphs with(…) induced $K_{\geqslant}4$.

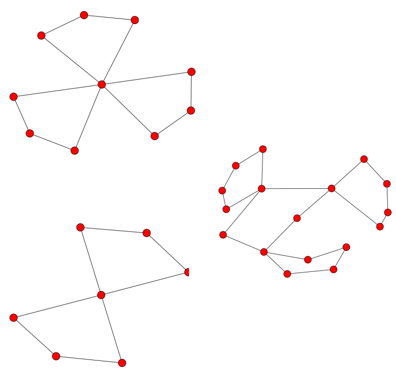


— manual generation of small instances



— brute force generation of small instances

```
class FourCactusGenerator(VertexIncrementalGenerator):

  def __init__(self, size):
    initial = _nx.complete_graph(1)

    self._operations = [ VI_C4 ]
    super(FourCactusGenerator, self).__init__(size = size, initial = initial)

class FiveCactusGenerator(VertexIncrementalGenerator):

  def __init__(self, size):
    initial = _nx.complete_graph(1)

    self._operations = [ VI_C5 ]
    super(FiveCactusGenerator, self).__init__(size = size, initial = initial)

class SixCactusGenerator(VertexIncrementalGenerator):

  def __init__(self, size):
    initial = _nx.complete_graph(1)

    self._operations = [ VI_C6 ]
    super(SixCactusGenerator, self).__init__(size = size, initial = initial)
```



Conclusion

Summary

— Derived an exact enumeration for cactus graphs (previously unknown)

— Explored the split decomposition as a generalizable method for graph enumeration, as first examined with analytic combinatorics by Chauve *et al.* (2014), and extended by Bahrani and Lumbroso (2016)

— For the first time studied a graph class with a split decomposition tree that contains prime nodes

Next Steps

— Asymptotics

— Parameter analysis

— Cycle-pointing and random sampling

— Consider other kinds of prime nodes (*e.g.* bipartite nodes are prime nodes for parity graphs and were studied asymptotically by Shi and Lumbroso (2017), but an exact enumeration is unknown)

Thank you!