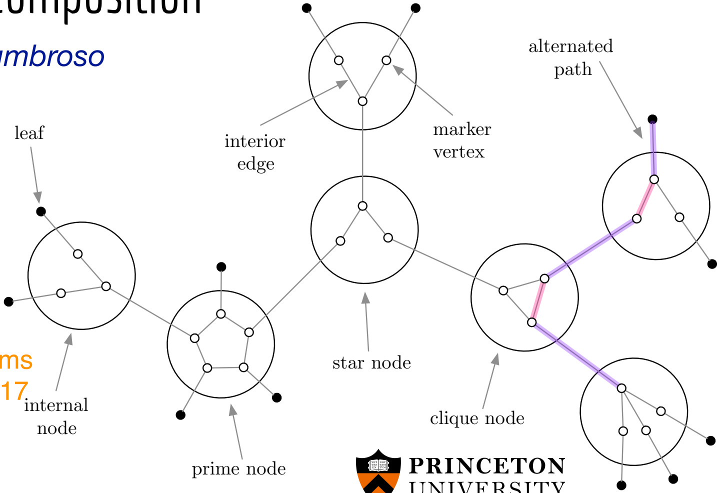
Graph Enumeration with the Split-Decomposition

Jérémie Lumbroso

An Exact Enumeration of Distance-Hereditary Graphs
with Cédric Chauve (Simon Fraser Univ.) and Éric Fusy (Polytechnique)

An Exact Enumeration of Distance-Hereditary Graphs
with Maryam Bahrani (*Princeton Univ.*)



Analysis of Algorithms
Princeton, June 2017

PRINCETON UNIVERSITY

0. Motivation and Outline

Motivation:

- in this talk: precisely enumerate large classes of graphs
- we combine in novel way:
  - classical characterization of graphs by tree-decompositions—because trees are **easier to count**
  - "graph labeled tree" framework (Gioan and Paul, 2012)
  - techniques in analytic combinatorics (symbolic method + asymptotic theorems)
  - technique from species theory (dissymetry theorem on trees)
- obtain exact and asymptotic enumerations + more

Outline:

- present definitions (graph decomposition, split decomposition, symbolic method)
- illustrate our approach for a simpler class of graphs (3-leaf power graphs)
- results for distance-hereditary graphs
- perspectives

context: some direct predecessors of our method

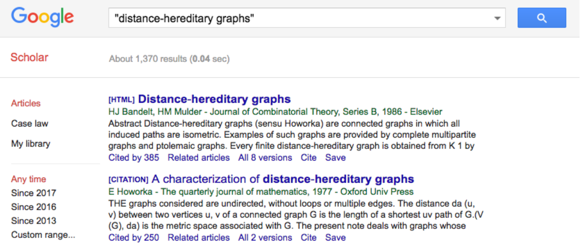
this work is informed by a long line of research on graph decomposition (see Gioan and Paul especially), but two prior works are particular relevant:

- **Thimonier and Ravelomanana 2002:** asymptotic enumeration of cographs (totally decomposable graphs for modular decomposition) using analytic combinatorics techniques
- **Nakano *et al.* 2007:** encoding and upper-bound for enumeration of distance-hereditary graphs (totally decomposable graphs for split decomposition) using algorithmic construction
- **Gioan and Paul, 2009-2012:** introduced the notion of graph-labeled tree and way to characterize split-decomposition output

context: distance-hereditary graphs (1)

goal: develop general methods cover vast subsets of **perfect graphs**[1]

starting point distance-hereditary graphs: [all as of Jan. 16th]



- planar graphs: 44 500 results
- interval graphs: 11 600 results [imperfect: incl. in perf. gr.]
- perfect graphs: 9 990 result
- chordal graphs: 8 860 results
- series-parallel graphs: 4 720 results
- cographs: 2 690 results
- block graphs: 1940 results

[1] chromatic number of every induced subgraph = size of max-clique of subgraph

context: distance-hereditary graphs (2)

- **1977, Howorka:** defines DH graphs (respect isometric distance: all induced paths between two vertices are same length)
- **1982, Cunningham:** introduces split-decomposition (as "join decomposition")
- **1986, Bandelt and Mulder:** vertex-incremental characterization
- **1990, Hammer and Maffray:** DH graphs are totally decomposable by the split-decomposition
- **2003, Spinrad:** upper-bound of enumeration sequence $2^{O(n \log n)}$
- **2009, Nakano *et al.*:** upper-bound of $2^{\lceil 3.59n \rceil}$ (approx. within factor 2)
- **2014-16, Chauve, Fusy, L.:** exact enumeration + full asymptotic (= constant, polynomial and exp. terms)

1. Graph decompositions

Def: a *graph-labeled tree* (GLT) is a pair $(T, \mathcal{F})$, with T a tree and $\mathcal{F}$ a set of graphs such that:

- a node v of degree k of T is labeled by graph $G_v \in \mathcal{F}$ on k vertices;
- there is a bijection ρ_v from the tree-edges incident to v to the vertices of G_v.

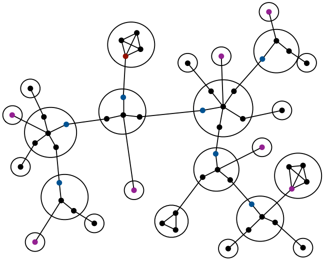
1. Graph decompositions

Def: a *graph-labeled tree* (GLT) is a pair $(T, \mathcal{F})$, with T a tree and $\mathcal{F}$ a set of graphs such that:

- a node v of degree k of T is labeled by graph $G_v \in \mathcal{F}$ on k vertices;
- there is a bijection ρ_v from the tree-edges incident to v to the vertices of G_v.

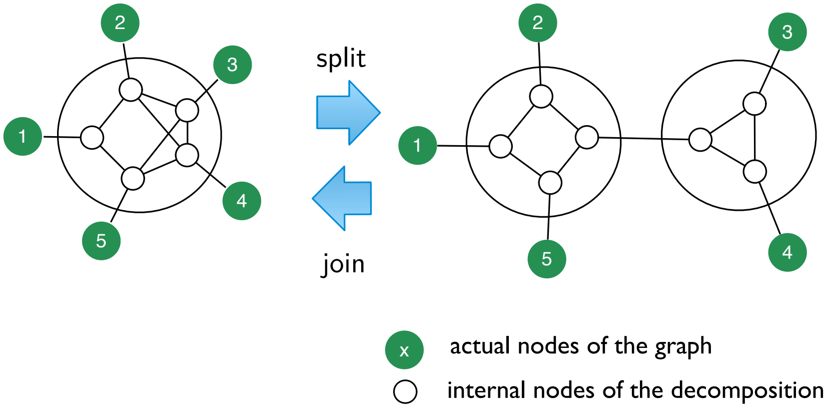
Def: a *rooted graph-labeled tree* is a graph-labeled tree of which one internal node is distinguished.

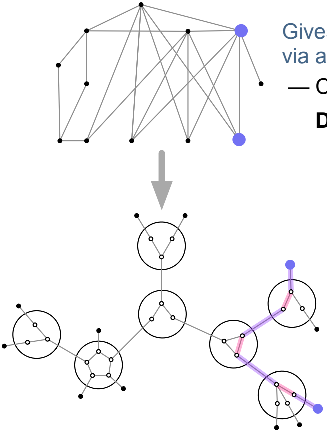
Remark: several types of decompositions of graphs (modular, split...); each decomposition has **totally decomposition graphs** for which the decomposition does not contain internal prime graphs.



Def: a bipartition (A, B) of a the vertices of a graph is a *split* iff

- $|A| \geqslant 2$, $|B| \geqslant 2$;
- for $x \in A$ and $y \in B$, $xy \in E$ iff $x \in N(B)$ and $y \in N(A)$.





Gives a *graph-labeled tree* representation of a graph via a series of *split* operations

— Can read adjacencies from *alternated paths*.

Decomposition base cases:

degenerate nodes: *prime* nodes:

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.

Remark:

- *distance-hereditary graphs*: graphs that are totally decomposable by split decomposition: internal nodes are star-nodes or clique-nodes;
- *3-leaf power graphs*: **subset** of distance-hereditary graphs, with additional constraint that star nodes form connected subtree.

2. Specifiable Combinatorial Classes

a class $\mathcal{A}$ is a specifiable combinatorial class if:

- described by symbolic rules (= grammar)

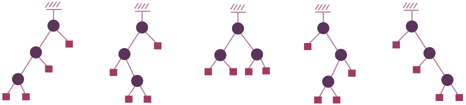
$\mathcal{Z}$, ε	$+$, $\times$, Seq, Set, Cyc, ...
building blocks	ways to combine them

- possible recursive (defined using itself)
- the number a_n of objects of size n is finite

Example: class $\mathcal{B}$ of binary trees specified by

$$\mathcal{B} = \varepsilon + \mathcal{B} \times \mathcal{Z} \times \mathcal{B}$$

all binary trees of size 3 (with 3 internal nodes $\bullet$):



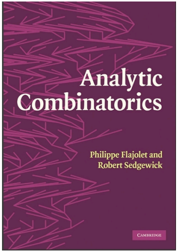
symbolic method [Flajolet & Sedgewick 09]

the **generating function** $A(z)$ of class $\mathcal{A}$ encodes, within a function, the complete enumeration (the number of objects for each size) of the class:

$$A(z) = \sum_{n=0}^{\infty} a_n z^n$$

- in the general case, this generating function (GF) is a formal object; however the GF of decomposable classes is often convergent
- dictionary: correspondence which exactly relates specific. and GF

construction	specification	GF
neutral element	ε	1
atome	$\mathcal{Z}$	z
union	$\mathcal{A} + \mathcal{B}$	$A(z) + B(z)$
Cartesian product	$\mathcal{A} \times \mathcal{B}$	$A(z) \cdot B(z)$
sequence	$\mathrm{Seq}(\mathcal{A})$	$\frac{1}{1-A(z)}$



example: class $\mathcal{B}$ of binary trees

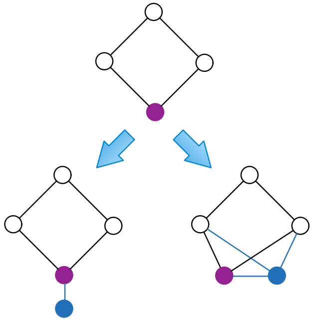
$$\mathcal{B} = \varepsilon + \mathcal{B} \times \mathcal{Z} \times \mathcal{B} \quad \Rightarrow \quad B(z) = 1 + B(z) \cdot z \cdot B(z) = \frac{1 - \sqrt{1-4z}}{2}$$

3. 3-LEAF POWER graphs

(One Possible) Def: a connected graph is a 3-leaf power graphs (3LP) iff it results from a tree by replacing every vertex by a clique of arbitrary size.

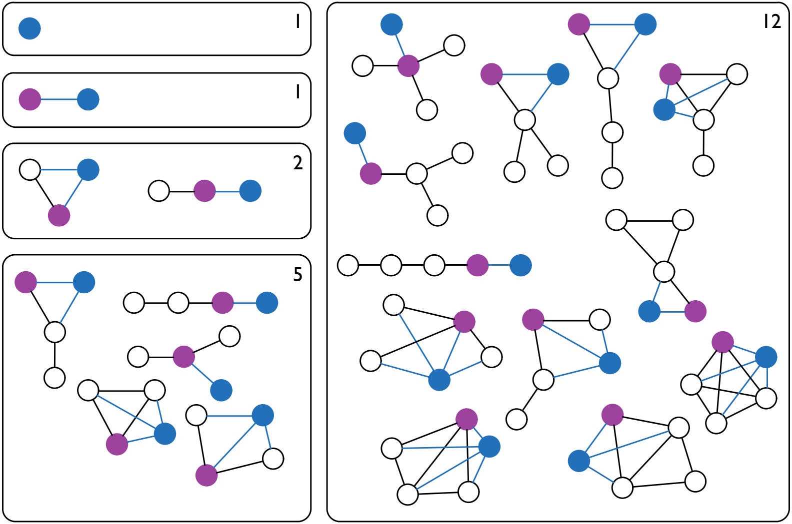
Algorithmic Characterization: 3LP graphs are obtained from a single vertex by

- **first** iterating arbitrary additions of pendant vertex;
- **then** iterating arbitrary additions of true twins.



(This caracterization is especially useful when establishing a reference, brute-force enumeration of these graphs!)

the first few 3-leaf power graphs



if these graphs were to be constructed by incremental construction, the **blue vertex** represents the vertex added from a smaller graph

obtaining **rooted** grammar of 3LP

Split-tree characterization of 3LP graphs (Gioan & Paul 2009):

1. its split tree $\mathrm{ST}(G)$ has only of clique-nodes and star-nodes;
2. the set of star-nodes forms a connected subtree of $\mathrm{ST}(G)$;
3. the center of a star-node is incident either to a leaf or a clique-node.

Thm (Chauve et al.) From this, we describe **rooted** tree decomposition, by walking through the tree

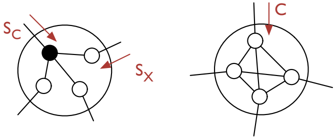
$$3\mathcal{LP}_\bullet = \mathcal{L}_\bullet \times (\mathcal{S}_C + \mathcal{S}_X) + \mathcal{C}_\bullet \qquad \mathcal{S}_C = \mathrm{Set}_{\geq 2}(\mathcal{L} + \mathcal{S}_X)$$

$$\mathcal{S}_X = \mathcal{L} \times \mathrm{Set}_{\geqslant 1}(\mathcal{L} + \mathcal{S}_X) \qquad \mathcal{L} = \mathcal{Z} + \mathrm{Set}_{\geqslant 2}(\mathcal{Z})$$

$$\mathcal{L}_\bullet = \mathcal{Z}_\bullet + \mathcal{Z}_\bullet \times \mathrm{Set}_{\geqslant 1}(\mathcal{Z}) \qquad \mathcal{C}_\bullet = \mathcal{Z}_\bullet \times \mathrm{Set}_{\geq 2}(\mathcal{Z})$$

where

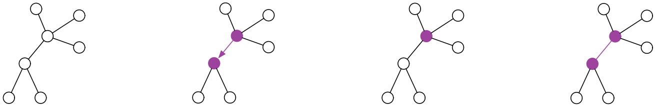
- $\mathcal{S}_C$ are star-nodes entered through their center; $\mathcal{S}_X$, their extremities;
- $\mathcal{A}_\bullet$ is a class where one vertex is distinguished;
- $\mathcal{L}$ are leaves (either cliques or single vertices) and $\mathcal{C}$ (clique).



from rooted to **unrooted**: dissymetry theorem for trees

- the grammars obtained describe a class of **rooted** trees; so the identical graphs are counted several times
- we need a tool to transform these grammars into grammars for the **equivalent unrooted class**;
- one such tool, the **Dissymetry Theorem for Trees** [Bergeron *et al.* 98] states

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



with

  - $\mathcal{A}$, unrooted class (which we are looking for)
  - $\mathcal{A}_{\circ}$, class rooted **node** (which we have)
  - $\mathcal{A}_{\circ-\circ}$ and $\mathcal{A}_{\circ\to\circ}$, class respectively rooted in **undirected edge** and **directed edge** (easy to obtain from $\mathcal{A}_{\circ}$)

- alternate tool: cycle pointing, Bodirsky et al. 2011 (more difficult but preserves combinatorial grammar)

unrooted grammar — just for your information

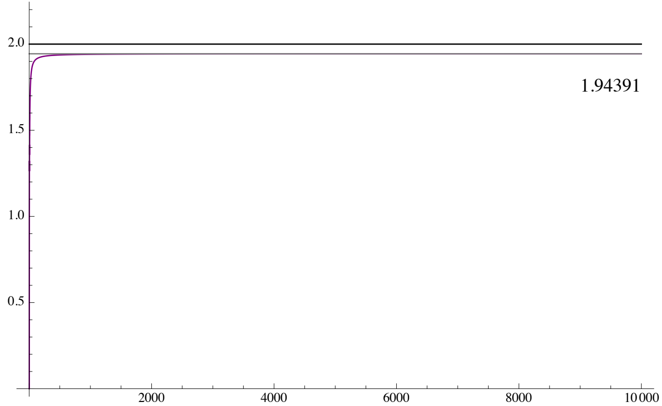
- from dissymetry theorem, we deduce $\mathcal{A} = \mathcal{A}_\circ + \mathcal{A}_{\circ-\circ} - \mathcal{A}_{\circ\to\circ}$ *for the purposes of enumeration*
- thus the *unrooted* 3LP graphs are described by

$$
\begin{aligned}
3\mathcal{LP} &= \mathcal{C} + \mathcal{T}_S + \mathcal{T}_{S-S} - \mathcal{T}_{S\to S} \\
\mathcal{T}_S &= \mathcal{L} \times \mathcal{S}_C \\
\mathcal{T}_{S-S} &= \mathrm{Set}_2\left(\mathcal{S}_X\right) \\
\mathcal{T}_{S\to S} &= \mathcal{S}_X \times \mathcal{S}_X \\
\mathcal{S}_C &= \mathrm{Set}_{\geqslant 2}\left(\mathcal{L} + \mathcal{S}_X\right) \\
\mathcal{S}_X &= \mathcal{L} \times \mathrm{Set}_{\geqslant 1}\left(\mathcal{L} + \mathcal{S}_X\right) \\
\mathcal{L} &= \mathcal{Z} + \mathrm{Set}_{\geqslant 2}\left(\mathcal{Z}\right) \\
\mathcal{C} &= \mathrm{Set}_{\geqslant 3}\left(\mathcal{Z}\right).
\end{aligned}
$$

- remark:
  - original terms: $\mathcal{S}_C$, $\mathcal{S}_X$, $\mathcal{L}$, $\mathcal{C}$
  - terms from the dissymetry theorem: $\mathcal{T}_S$, $\mathcal{T}_{S-S}$, $\mathcal{T}_{S\to S}$
  - main term in the form of $\mathcal{A} = \mathcal{A}_\circ + \mathcal{A}_{\circ-\circ} - \mathcal{A}_{\circ\to\circ}$

experimental enumerations for graphs of size up to 10 000 (1)

- t_n: # of unlabeled and unrooted 3LP graphs of size n
- we know that $t_n = O(\alpha^n)$, want to find α
- here, plot of $\log_2(t_n/t_{n-1})$
- suggests growths of $\alpha = 2^{1.943\ldots}$ for 3-Leaf Power Graphs



experimental enumerations for graphs of size up to 10 000 (2)

Maple code to obtain previous plot, which allows to conjecture the asymptotic enumeration, once a grammar for the trees is found.

```
with(combstruct): with(plots):
TLP_UNROOTED_PARTS := {
  z = Atom,
  G_SUPERSET = Union(C, Union(TS, TSSu)),
  TS         = Prod(L, SC),
  TSSu       = Set(SX, card=2),
  TSSd       = Prod(SX, SX),
  SC         = Set(Union(L, SX), card >= 2),
  SX         = Prod(L, Set(Union(L, SX), card >= 1)),
  L          = Union(z, Set(z, card >= 2)),
  C          = Set(z, card >= 3)
}:
N := 10000:
OGF_TLP_SUPERSET := add(count([G_SUPERSET, TLP_UNROOTED_PARTS, unlabeled],
                               size = n) * x^n, n = 1 .. N):
OGF_TLP_TSSd := add(count([TSSd, TLP_UNROOTED_PARTS, unlabeled], size = n) *
                    x^n, n = 1 .. N):
OGF_TLP := OGF_TLP_SUPERSET - OGF_TLP_TSSd:
TLP_RATIOS := [seq([i, evalf(log(coeff(OGF_TLP, x, i)/coeff(OGF_TLP, x, i-1)))
                              /log(2)], i = 10 .. N)]:
plot(TLP_LOGS);
```

- the asymptotics of a algebraic grammar (described only with $+$ and $\times$, not sets) is well-known under theorem of Drmota-Lalley-Woods
- usually extends with no problem to other operations, under some niceness hypotheses [for ex., Chapuy *et al.* 08]

Method (without correctness proof):

1. let combinatorial system $\mathcal{S}$

$$\mathcal{S}\begin{cases} \mathcal{X}_1 & = & \Phi_1(\mathcal{X}_1, \ldots, \mathcal{X}_m) \\ \vdots & \vdots & \vdots \\ \mathcal{X}_m & = & \Phi_m(\mathcal{X}_1, \ldots, \mathcal{X}_m) \end{cases}$$

2. translate to equations on generating functions

$$\begin{array}{ccc} 0 & = & -X_1(z) + \phi_1(X_1(z), \ldots, X_m(z), z) \\ \vdots & \vdots & \vdots \\ 0 & = & -X_m(z) + \phi_m(X_1(z), \ldots, X_m(z), z) \end{array}$$

   with additional equation for recursion well-foundness

$$0 = \det(\mathrm{Jacobian}(\mathcal{S}))$$

3. solve numerically

Practical tweaks:

- our grammars involve unlabeled set operations, which result in infinite Polya series: these must be truncated
- additionally, singularity (= inverse of exponential growth) of rooted and unrooted classes is same: so work on (simpler) rooted grammar

Result: implemented algorithm in Maple, to obtain asymptotic of graph-decomposition with **arbitrary precision**:

```
TLP_ROOTED := {
  Gp = Union(Prod(Lp, Union(SC, SX)), Cp),
  SC = Set(Union(L, SX), card >= 2),
  SX = Prod(L, Set(Union(L, SX), card >= 1)),
  Cp = Prod(v, Set(v, card >= 2)), v = Atom, # [... snipped ...]
}:
fsolve_combsys(TLP_ROOTED, 100, z);
```

Eq1 = 0.02370404136, Eq2 = 0.5329652240, Eq3 = 0.3510690027,
Eq4 = 0.3510690027, Eq5 = 0.8016703909, Eq6 = 0.6489309973,
Eq7 = 0.2598453536, **z = 0.2598453536**

asymptotic exponential growth $= 1/z$

summary of 3LP and DH enumerations

We have used the example of 3-Leaf Power Graphs, because it is simpler to present, but **all results obtained for Distance-Hereditary graphs**.

Exact and asymptotic results for two major classes, previously unknown.

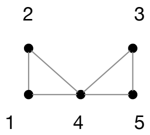
3-Leaf Power Graphs:

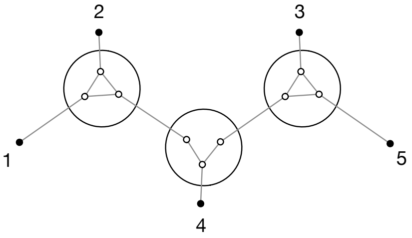
- exact enumeration: 1, 1, 2, 5, 12, 32, 82, 227, 629, 1840, 5456, 16701, 51939, 164688, ... (calculated linearly as function of size n)
- asymptotics: $c \cdot 3.848442876\ldots^n \cdot n^{-5/2}$ with $c \approx 0.70955825396\ldots$ (bound: $2^{1.9442748333}$)

Distance-Hereditary Graphs:

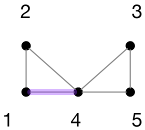
- exact enumeration: 1, 1, 2, 6, 18, 73, 308, 1484, 7492, 40010, 220676, 1253940, ... (calculated linearly as function of size n)
- asymptotics: $c \cdot 7.249751250\ldots^n \cdot n^{-5/2}$ with $c \approx 0.02337516194\ldots$ (bound: $2^{2.857931495}$)

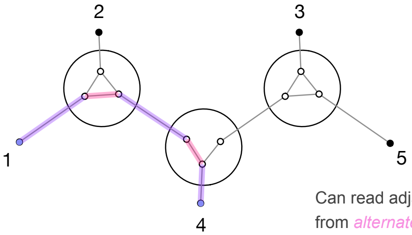
Split-Tree Examples (1)





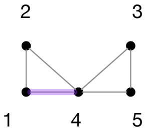
Split-Tree Examples (1)

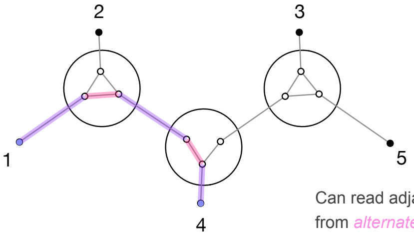




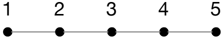
Can read adjacencies from *alternated paths*.

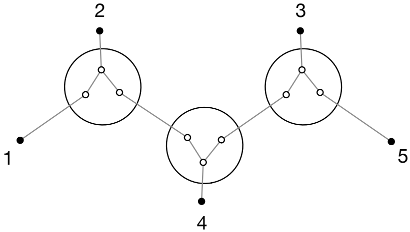
Split-Tree Examples (1)



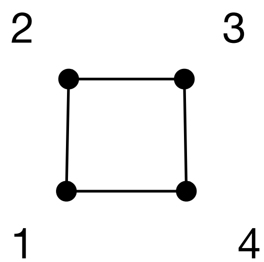


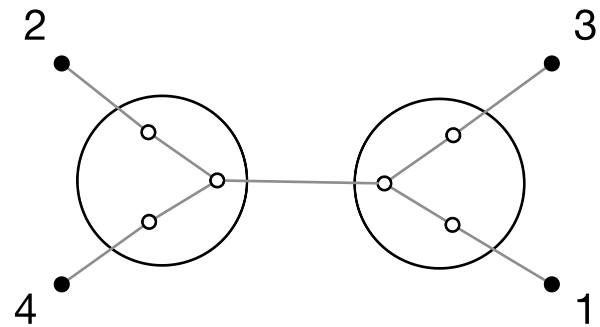
Can read adjacencies from *alternated paths*.



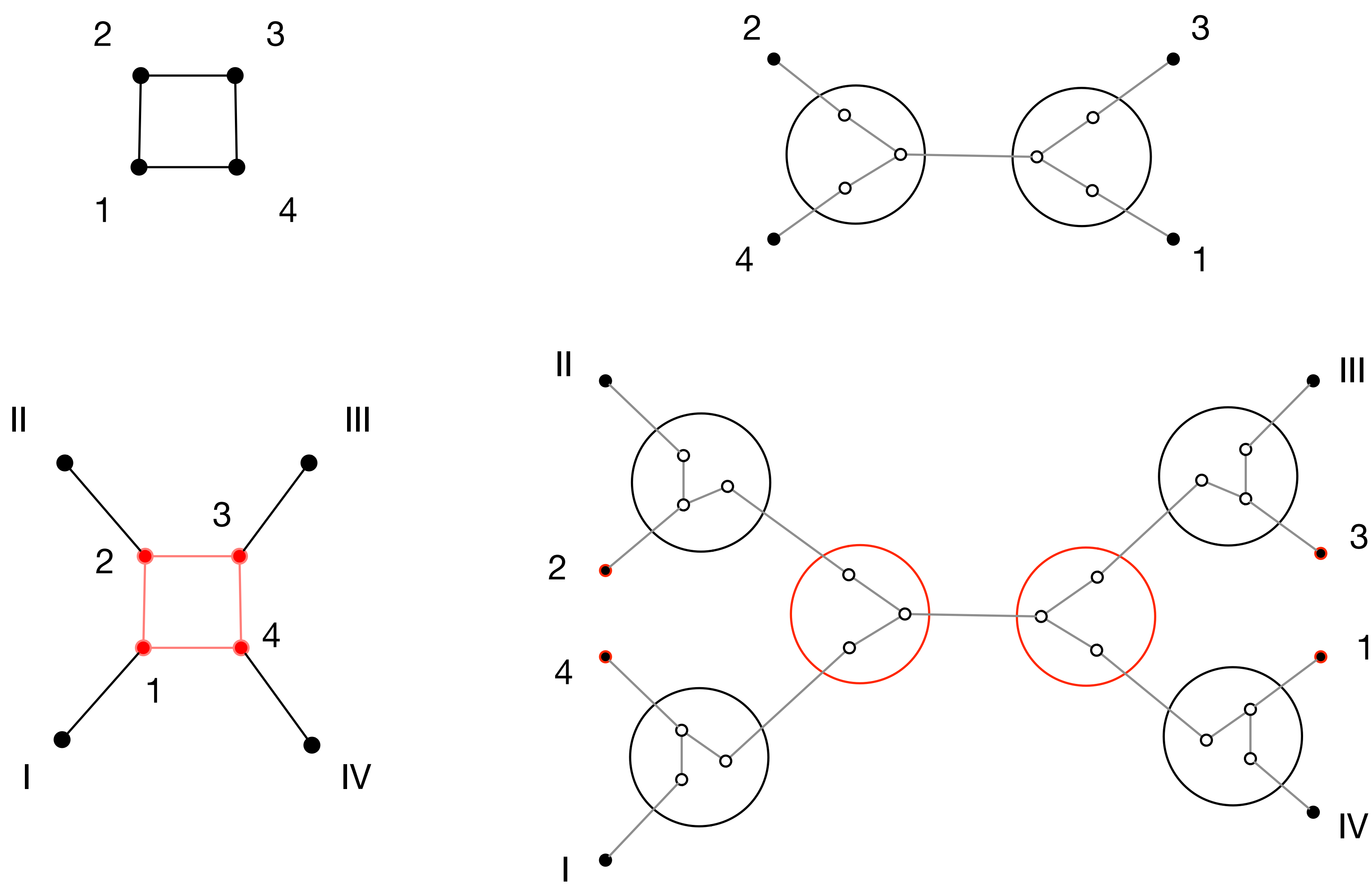


Split-Tree Examples (2)



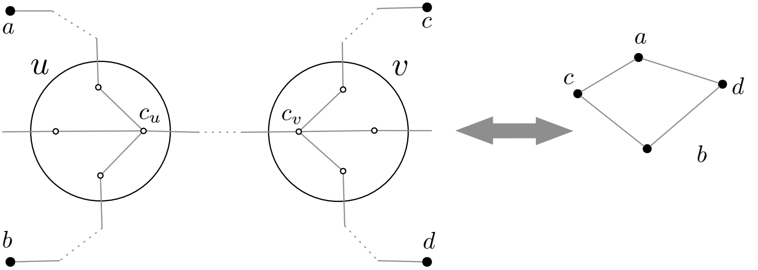


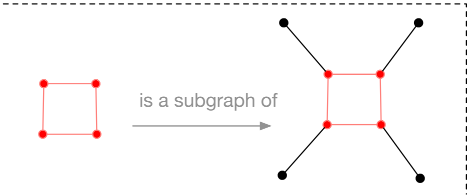
Split-Tree Examples (2)



C4 as a subgraph

THM (Bahrani, L.). In a **clique-star** split-decomposition tree:

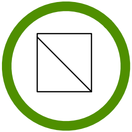




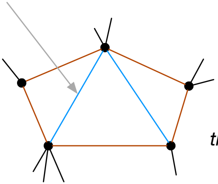
+ analog results for diamond, bridge, etc.

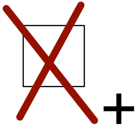
Ptolemaic graphs =

(Howorka 1981, Kay and Chartrand 1965)

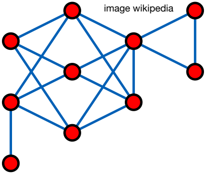


chord of (among others) the red cycle





there are no C4's



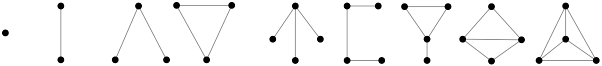
chordal graphs
(every cycle of size 4 has a chord)

distance hereditary
(all distances preserved in induced subgraph)

=> forbidding that two star nodes be connected by their center

=> restricting split-decomposition tree to only use star and clique nodes

result: grammar + enumerations in labeled and unlabeled cases



All ptolemaic graphs with at most four vertices.

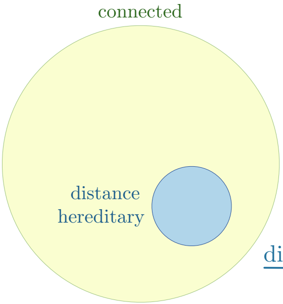
	Rooted	Labeled	EIS	Enumeration
Ptolemaic graphs	✓	✓		1, 2, 12, 140, 2405, 54252, 1512539, 50168456, 1928240622, 84240029730, 4121792058791, 223248397559376, ...
Ptolemaic graphs		✓		1, 1, 4, 35, 481, 9042, 216077, 6271057, 214248958, 8424002973, 374708368981, 18604033129948, 1019915376831963, ...
Ptolemaic graphs	✓			1, 1, 3, 10, 40, 168, 764, 3589, 17460, 86858, 440507, 2267491, 11819232, 62250491, 330794053, 1771283115, 9547905381, ...
Ptolemaic graphs				1, 1, 2, 5, 14, 47, 170, 676, 2834, 12471, 56675, 264906,1264851, 6150187, 30357300, 151798497, 767573729, 3919462385, ...

Family Characterization	Split-Decomposition Tree Characterization
Distance hereditary with no induced C_4	Clique-star tree with no center-center paths (*i.e.* paths connected the centers of two star nodes).

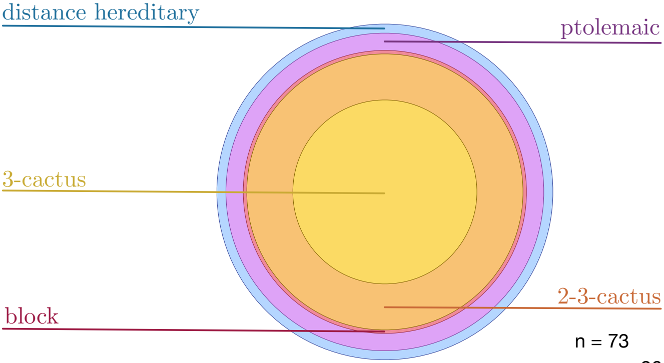
Rooted Grammar	Unrooted Grammar
$\mathcal{PG}_\bullet = \mathcal{Z}_\bullet \times (\mathcal{S}_C + \mathcal{S}_X + \mathcal{K})$	$\mathcal{PG} = \mathcal{T}_K + \mathcal{T}_S + \mathcal{T}_{S-S} - \mathcal{T}_{S\to S} - \mathcal{T}_{S-K}$
$\mathcal{S}_C = \text{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_X)$	$\mathcal{T}_K = \mathcal{S}_C \times \text{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{S}_X) + \text{SET}_{\geqslant 3}(\mathcal{Z} + \mathcal{S}_X)$
$\mathcal{S}_X = (\mathcal{Z} + \overline{\mathcal{K}}) \times \text{SET}_{\geqslant 1}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_X)$	$\mathcal{T}_S = \mathcal{S}_C \times (\mathcal{Z} + \overline{\mathcal{K}})$
$\mathcal{K} = \mathcal{S}_C \times \text{SET}_{\geqslant 1}(\mathcal{Z} + \mathcal{S}_X) + \text{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{S}_X)$	$\mathcal{T}_{S-S} = \text{SET}_2(\mathcal{S}_X)$
$\overline{\mathcal{K}} = \text{SET}_{\geqslant 2}(\mathcal{Z} + \mathcal{S}_X)$	$\mathcal{T}_{S\to S} = \mathcal{S}_X \times \mathcal{S}_X$
	$\mathcal{T}_{S-K} = \mathcal{K} \times \mathcal{S}_X + \overline{\mathcal{K}} \times \mathcal{S}_C$

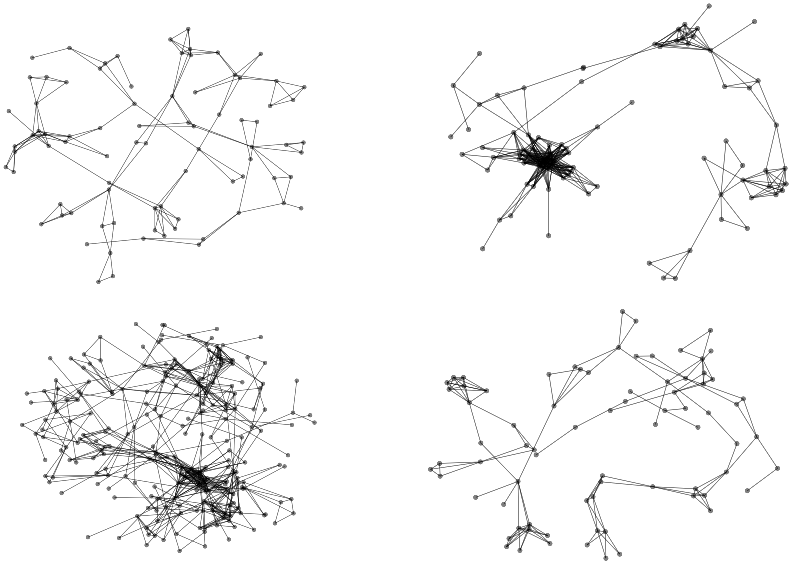
Table 2. Characterization, grammar, and the first few terms of the enumeration of ptolemaic graphs.

Relative “densities”



Logarithmic plot of the number of graphs of each class for a given size.





Images by Alex Iriza, obtained by Boltzmann generator using cycle pointing. Implementation on GitHub.

5. Perspectives and upcoming results

Analyses:

- **Parameter analysis:** analyzing, either theoretically or experimentally (already possible using random generation) various parameters of these graphs; such as distribution of star-nodes, clique-nodes, etc.
- **Other classes:** extending methodology to non-totally decomposable classes of graphs—either for modular decomposition or split decomposition (challenge is characterizing prime graphs in grammars).
  - bounds on parity graphs with bipartite prime [Shi, 2016 + ongoing]
  - forbidden subgraph characterizations [Bahrani and L., 2016]
  - cactus graphs [Bahrani and L., 2017]

Applications:

- **Encoding:** asymptotic result suggests more efficient encoding than the one provided by Nakano *et al.* 2007 (which uses 2^{4n} bits)?
  - automatic bounds given any vertex-incremental characterization [Shi, 2016]
- **Random generation:** efficient random generation already possible using cycle pointing [Fusy *et al.* 2007] [Iriza *et al.* 2015].