JASS 2008

Bounded Treewidth

Stephan Holzer
Technische Universität München

Plan of talk

History

Definitions

Examples

Applications

Plan of talk

History

Definitions

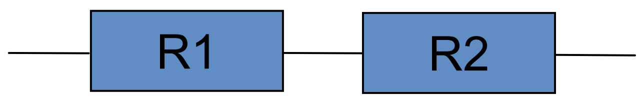
Examples

Applications

History

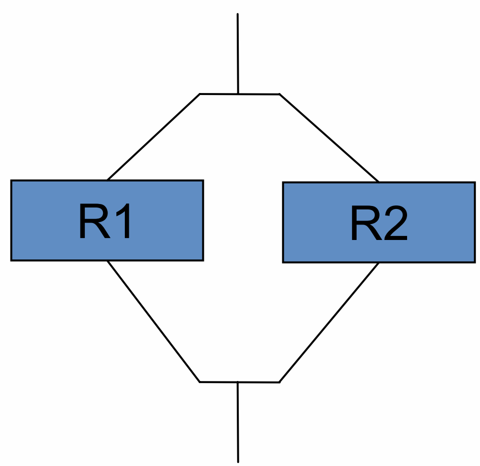
- Ohm's laws:

series



$$R = R_1 + R_2$$

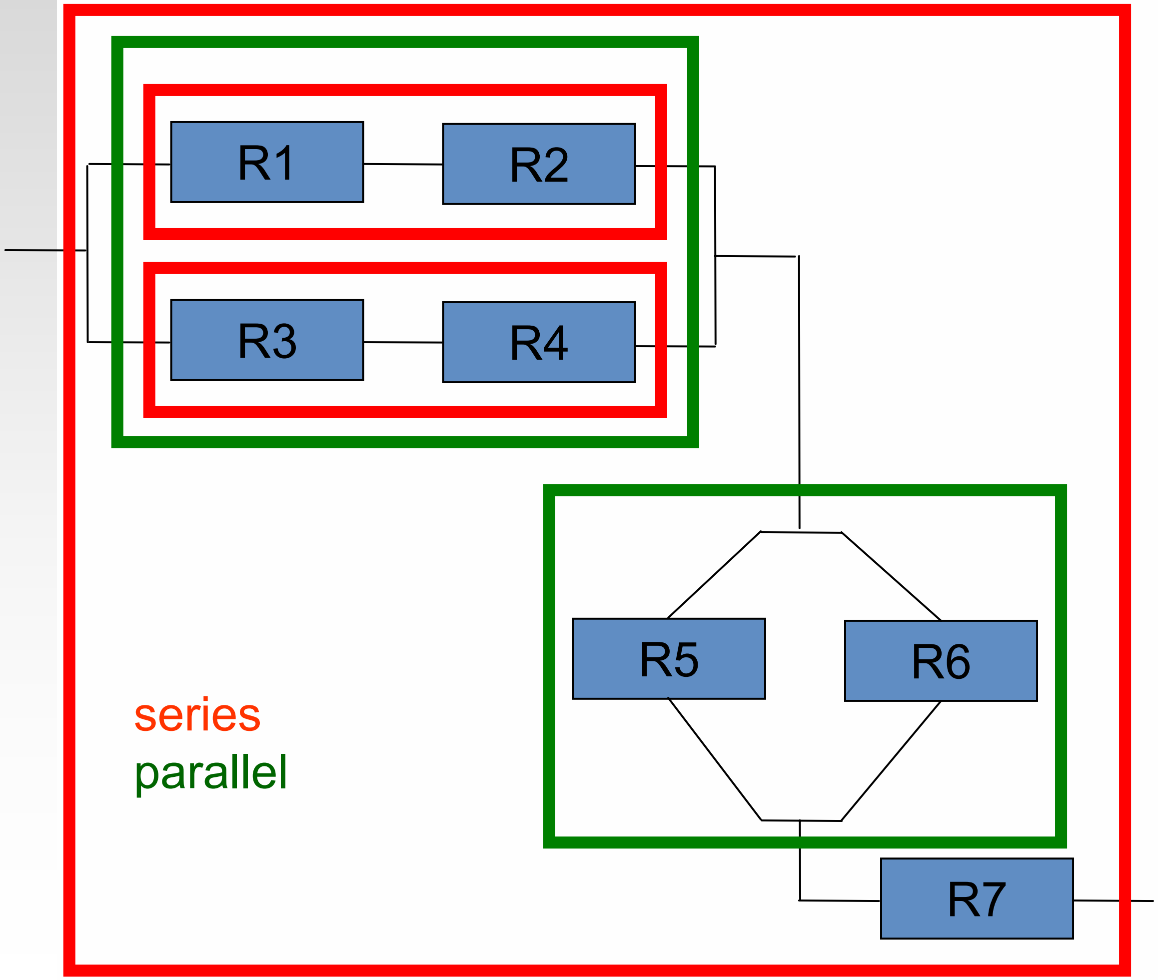
parallel



$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$$

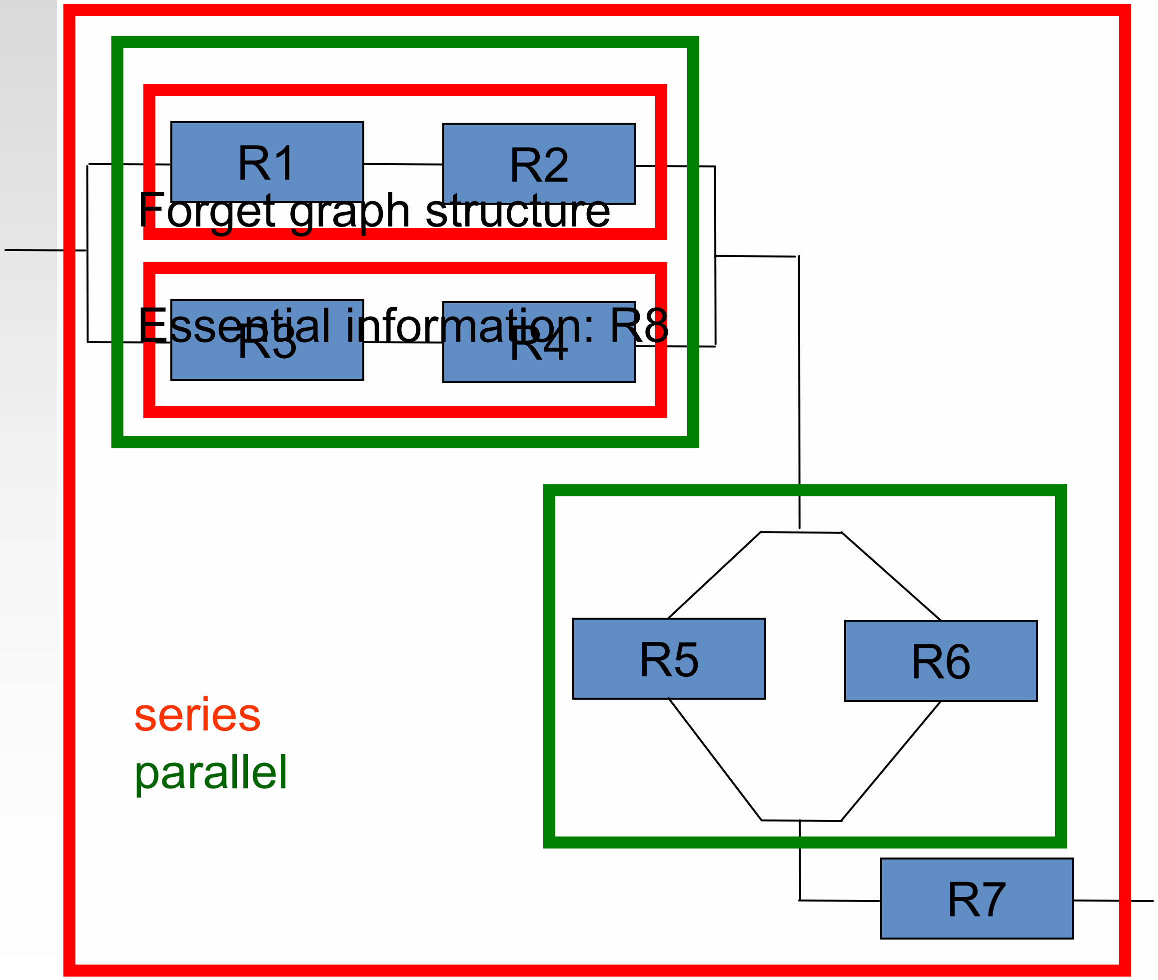
History

Series parallel graphs



History

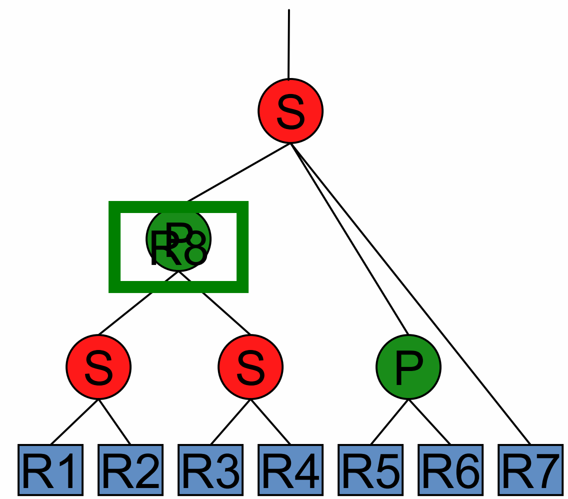
Series parallel graphs



Forget graph structure

Essential information: R8

Tree structure:



Early results (1960s / 70s)

- Tree structures help!
- Some NP-complete problems ~~on general graphs are hard~~ restricted to
  - trees
  - series parallel graphs

  easy!

Even some PSPACE-complete problems become easy

Tree-decomposition, Treewidth

Paul Seymour

Neil Robertson

GENERALIZE!

Plan of talk

History

Definitions

Examples

Applications

Tree-decomposition

Definition 1

Tree-decomposition

Let

- $G = (V, E)$ be a graph

A tree-decomposition of G is a pair $(\{X_i | i \in I\}, T = (I, F))$

- $T = (I, F)$ a tree
- $\{X_i | i \in I\}$ a family of subsets of V for each node $i \in I$ of T

Such that

- $\bigcup_{i \in I} X_i = V$
- if $(v, w) \in E$ is an edge, then there exists an $i \in I$ with $v \in X_i$ and $w \in X_i$
- all nodes j on any i-k-path satisfy $X_i \cap X_k \subseteq X_j$

Treewidth

Definition 2

Treewidth

Of a tree-decomposition $td = (\{X_i | i \in I\}, T = (I, F))$

- $treewidth(td) := \max_{i \in I} |X_i| - 1$

Of a graph G

- $TD :=$ all tree-decompositions of G
- $treewidth(G) := min_{td \in TD}(treewitdh(td))$

Definition 1

Tree-decomposition

Let

- $G = (V, E)$ be a graph

A tree-decomposition of G is a pair $(\{X_i | i \in I\}, T = (I, F))$

- $T = (I, F)$ a tree
- $\{X_i | i \in I\}$ a family of subsets of V for each node $i \in I$ of T

Such that

- $\bigcup_{i \in I} X_i = V$
- if $(v, w) \in E$ is an edge, then there exists an $i \in I$ with $v \in X_i$ and $w \in X_i$
- all nodes j on any i-k-path satisfy $X_i \cap X_k \subseteq X_j$

Plan of talk

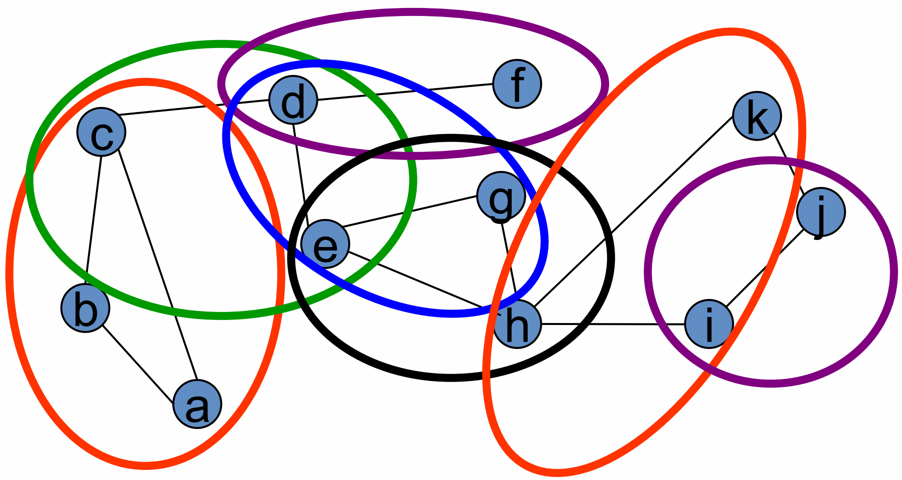
History

Definitions

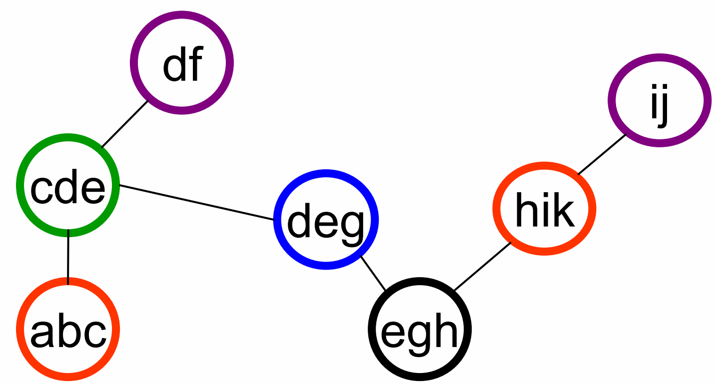
Examples

Applications

Example



tw(G)=2



Definition 1

Tree-decomposition

Let

- $G = (V, E)$ be a graph

A tree-decomposition of G is a pair $(\{X_i | i \in I\}, T = (I, F))$

- $T = (I, F)$ a tree
- $\{X_i | i \in I\}$ a family of subsets of V for each node $i \in I$ of T

Such that

- $\bigcup_{i \in I} X_i = V$
- if $(v, w) \in E$ is an edge, then there exists an $i \in I$ with $v \in X_i$ and $w \in X_i$
- all nodes j on any i-k-path satisfy $X_i \cap X_k \subseteq X_j$

Definition 2

Treewidth

Of a tree-decomposition $td = (\{X_i | i \in I\}, T = (I, F))$

- $treewidth(td) := \max_{i \in I} |X_i| - 1$

Of a graph G

- $TD :=$ all tree-decompositions of G
- $treewidth(G) := min_{td \in TD}(treewitdh(td))$

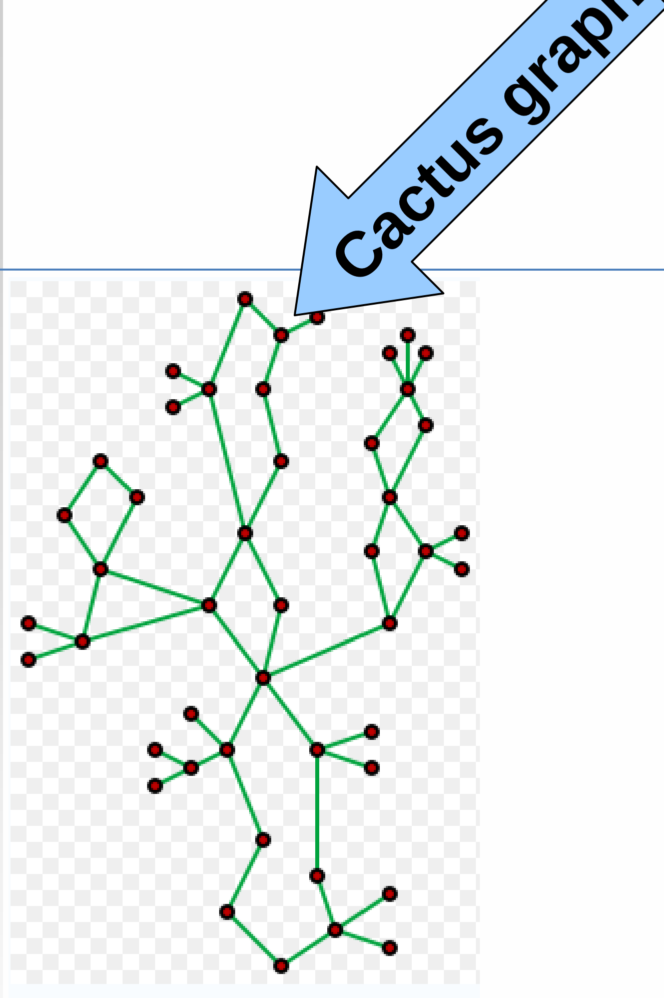
Examples

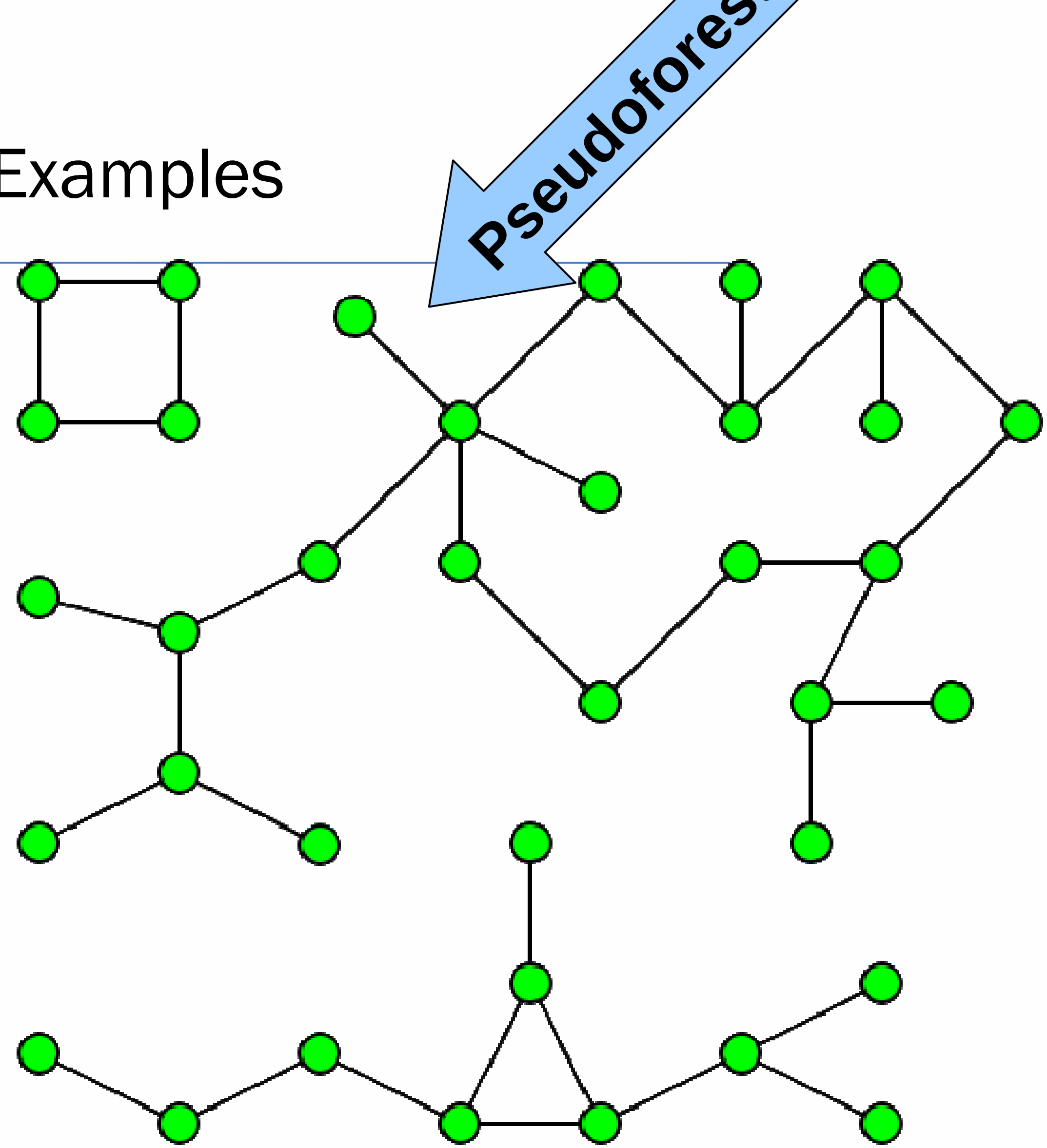
Graph classes with bounded treewidth include

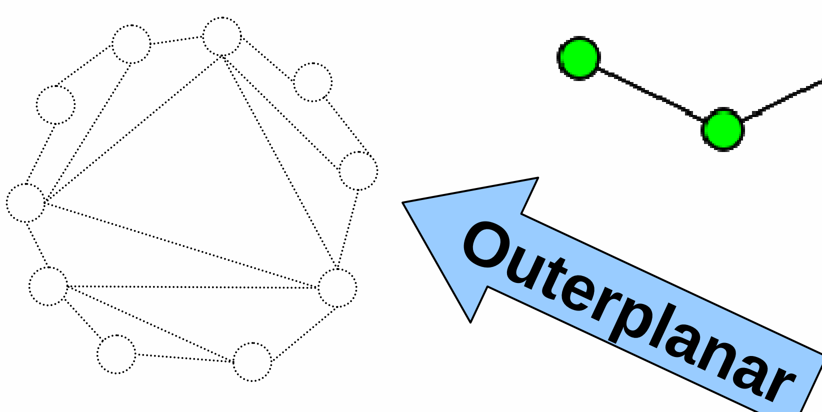
- Trees tw=1
- Series-parallel graphs tw=2
- Outerplanar graphs tw=3
- Halin graphs tw=3
- Pseudoforests tw=2
- Cactus graphs tw=2
- Control flow graphs arising in the compilation of structured programs

All results for bounded treewidth apply to theses classes!

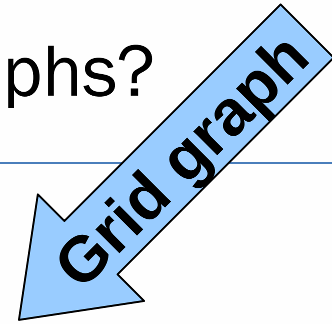
Examples

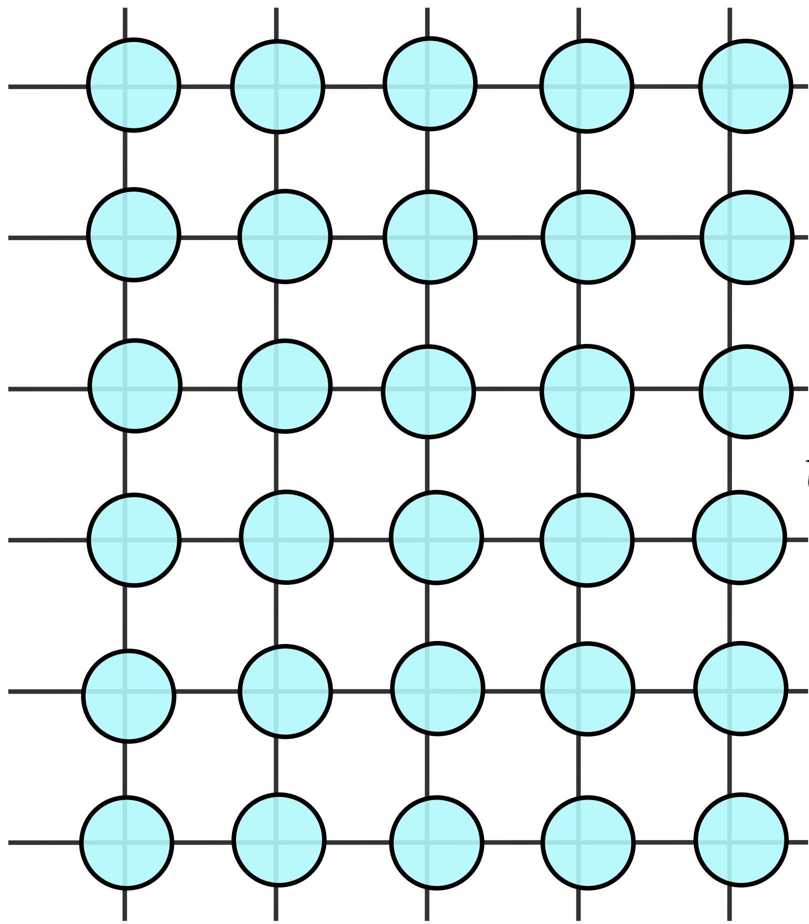






Planar graphs?





$$treewidth(n \times n \quad gridgraph) = n$$

Plan of talk

History

Definitions

Examples

Applications

Applications

- Cholesky factorisation
- Evolutionary Theory
- Expert Systems
- VLSI Layouts (via pathwidth)*
- Natural language processing
- Computing Tutte-polynomial of graphs with bounded treewidth in polynomial time

- Linear time algorithms (for Independent Set)*
- and other NP-complete problems*

* when the undelying class of graphs has constant bounded tree width

Treewidth and Cholesky factorisation

Input:
Symmetric positive-definite matrix M

Output:
decomposition $M = L \cdot L^T$ with L a lower triangular matrix

$$\begin{bmatrix} d & v^T \\ v & B \end{bmatrix} = \begin{bmatrix} \sqrt{d} & 0 \\ v/\sqrt{d} & I \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & B - v \cdot v^T/d \end{bmatrix} \cdot \begin{bmatrix} \sqrt{d} & v^T/\sqrt{d} \\ 0 & I \end{bmatrix}$$

Relation to graph: blackboard

Treewidth and Cholesky factorisation

Matrix sparse:
find elimination order (raws/colums) s.t.

all matrices $v \cdot v^T$ are small

many 0-rows/columns

Treewidth and evolutionary theory

Input:

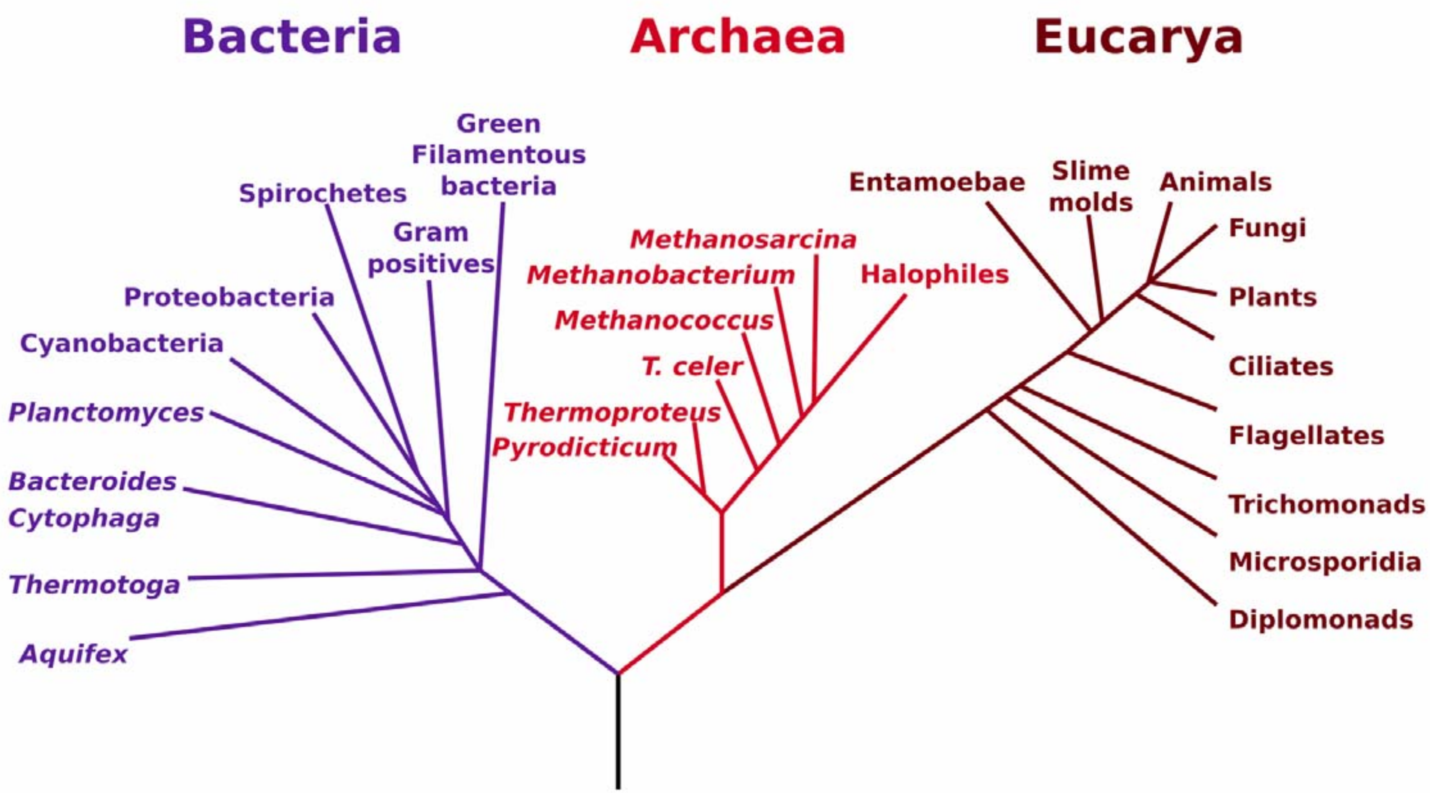
- set of n species
- set of m characteristics
- $n \times m$-array: value(species,characteristic)

Goal:
a 'good' evolution tree for these species
and their possibly extinct ancestors

variant: PERFECT-PHYLOGENY-PROBLEM

variant: PERFECT-PHYLOGENY-PROBLEM

Phylogenetic Tree of Life



V… …Y-PROBLEM

e…
I…

chordal??

Decide:
Can we add edges to G s.t. the resulting graph G' is chordal but has no edges between vertices of the same color?

Chordal graphs

Definition
Let $G = (V, E)$ be an undirected graph.
G is *chordal* if it contains no induced cycles $C_n, n \geq 4$

So every induced cycle is a triangle!

variant: PERFECT-PHYLOGENY-PROBLEM

Equivalently:
Does there exist a tree-decomposition $(\{X_i | i \in I\}, T)$ of G
s.t. for all $i \in I$:
if $v, w \in X_i, v \neq w$, then v and w have different colors.

Necessary condition: $treewidth(G) <$ number of colors.

Necessary condition: $treewidth(G) <$ number of colors.

Treewidth: linear time algorithm for IS

Input:
graph $G = (V, E)$

- tree-decomposition $(\{X_i | i \in I\}, T = (I, F))$ (binary)
- root r of T
- treewidth k

Output:
maximum size of a set $W \subseteq V$ s.t. for all $v, w \in W : (v, w) \notin E$

Treewidth: linear time algorithm for IS

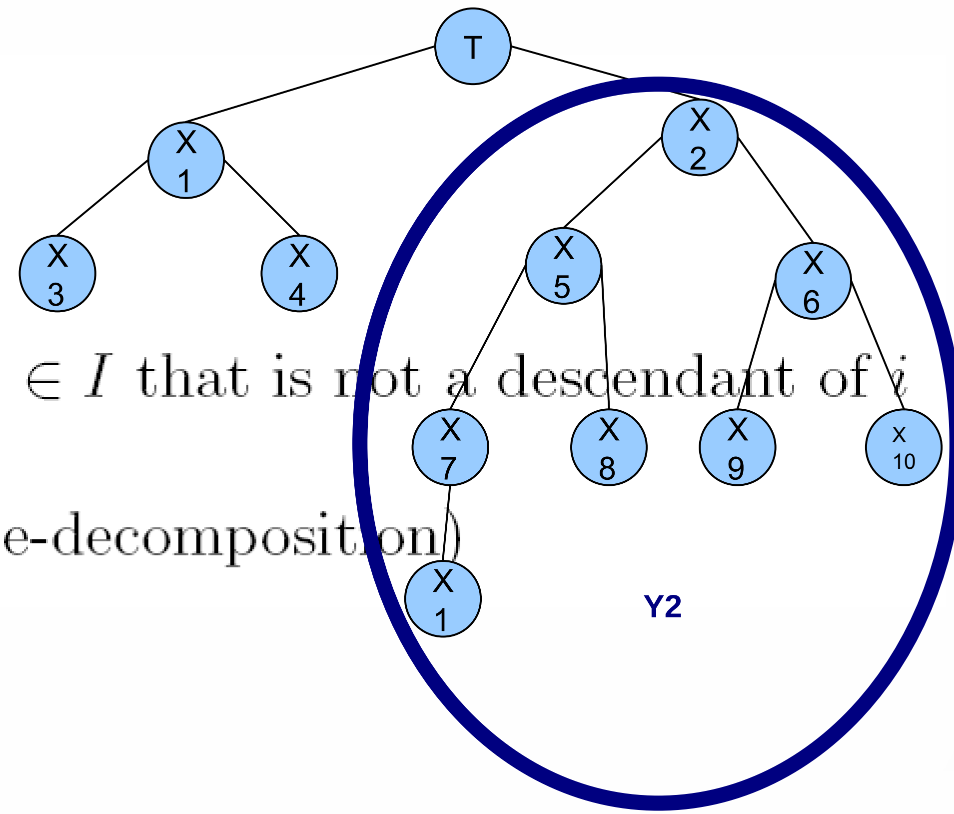
for each $i \in I$

define $Y_i := \{v \in X_j | j = i \text{ or } j \text{ is a descendant of } i\}$

if

- $v \in Y_i$
- $v \in X_j$ for some vertex $j \in I$ that is not a descendant of i

then: $v \in X_i$ (def. tree-decomposition)



Treewidth: linear time algorithm for IS

for each $i \in I$
define $Y_i := \{v \in X_i | j = i$ or j is a descendant of $i\}$

if

- $v \in Y_i$
- $v \in X_j$ adjacent to $w \in X_j$ with j a descendant of i

then: $v \in X_i$ (def. tree-decomposition)
or $w \in X_i$

Treewidth: linear time algorithm for IS

New problem

Input:
an Independent Set W of the subgraph $G[Y_i]$ induced by Y_i

Goal:
extend W to IS of G

only $|W|$ is important

Now:
important is only which vertices in X_i belong to W, not which vertices in $X_i - Y_i$ belong to W.

Treewidth: linear time algorithm for IS

For $i \in I, Z \subseteq X_i$,
define $is_i(Z)$ to be the maximum size
of an independent set W in $G[Y_i]$ with $W \cap X_i = Z$.

Take $is_i(Z) = -\infty$ if no such set exists.

Algorithm:
Compute all tables is_i in a bottom-up manner.

Treewidth: linear time algorithm for IS

leaf nodes i:

$$is_i(Z) = \begin{cases} |Z| & \text{if } \forall v, w \in Z \quad : (v,w) \notin E \\ -\infty & \text{if } \exists v, w \in Z \quad : (v,w) \in E \end{cases}$$

Treewidth: linear time algorithm for IS

internal node i (with two children j, k:

$$is_i(Z) = \begin{cases} \max\{is_j(Z') + is_k(Z'') \\ +|Z \cap (X_i - X_j - X_k)| \\ -|Z \cap X_j \cap X_k| \\ \text{where } Z \cap X_j = Z' \cap X_i \\ \text{and } Z \cap X_k = Z'' \cap X_i\} & : \forall v, w \in Z : (v,w) \notin E \\ \\ -\infty & : \exists v, w \in Z : (v,w) \in E \end{cases}$$

Treewidth: linear time algorithm for IS

Algorithm:
Compute table is in bottom-up order
Solution $= \max_{Z \subseteq X_r} is_r(Z)$

Runtime:
$O(2^{3k}n)$

Treewidth: linear time algorithms for ~~IS~~ many problems

General idea:

• Each table entry gives information about an equivalence class of partial solutions

• Number of equivalence classes is bounded by some constant when treewidth is bounded by a constant

• Tables can be computed using only the tables of the children of the node

Treewidth: linear time algorithm for IS

Other results:

- Large classes of problems can be solved in linear time when a tree-decomposition with constant bounded treewidth is available

- Approximation results

Finding tree-decompositions

Problem:

Input:
Graph $G = (V, E)$, integer k

Dedide:
Is $tw(G) \leq k$

?

Class	Treewidth	Pathwidth
Bounded degree	N [35]	N [101] (3)
Trees/Forests	C	P [133]
Series-parallel graphs	C	P [32]
Outerplanar graphs	C	P [32]
Halin graphs	C [143]	P [32]
k-Outerplanar graphs	C [20]	P [32]
Planar graphs	O	N [101] (3)
Chordal graphs	P (1)	N [68]
Starlike chordal graphs	P (1)	N [68]
k-Starlike chordal graphs	P (1)	P [68]
Co-chordal graphs	P [85]	P [85]
Split graphs	P (1)	P [68, 84]
Bipartite graphs	N	N
Permutation graphs	P [34]	P [34]
Circular permutation graphs	P [34]	O
Cocomparability graphs	N [6, 72]	N [6, 72]
Cographs	P [36]	P [36]
Chordal bipartite graphs	P [86]	N [35]
Interval graphs	P (2)	P (2)
Circular arc graphs	P [135]	O
Circle graphs	P [83]	N [35]

Finding tree-decompositions

- For constant treewidth k=1,2,3,4 graphs can be recognized in linear time

- Arbitrary fixed k: O(n logn)

Thank You