Split Decomposition and Circle Graph Recognition in Quasi-Linear Time

Christophe Paul

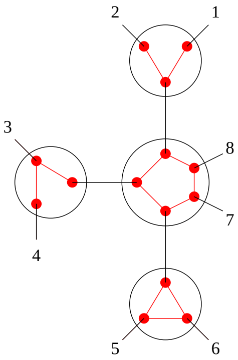
CNRS - LIRMM - Université Montpellier II, France

March 25, 2009

Joint work with **D. Corneil** (U. of Toronto), **E. Gioan** (CNRS LIRMM) and **M. Tedder** (U. of Toronto)

Graph-labeled tree (GLT)

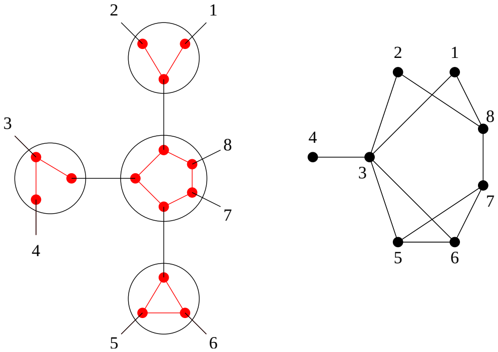
- a pair $(T, \mathcal{F})$ with T a tree and $\mathcal{F}$ a set of graphs
- each node v of degree k of T is labelled by a graph $G_v \in \mathcal{F}$
- a bijection ρ_v from the tree-edges incident to v to $V(G_v)$



Accessiblity graph of a GLT

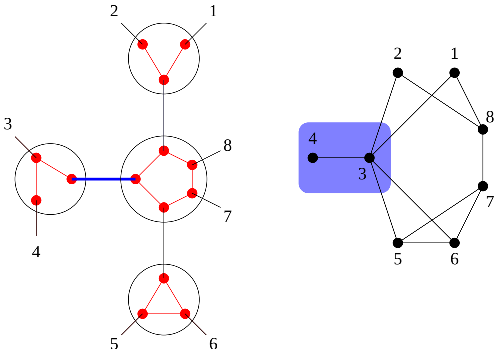
Given a GLT $(T, \mathcal{F})$, define the graph $G(T, \mathcal{F})$ with

- vertex set of $G(T, \mathcal{F})$ = set of leaves of T
- xy is an edge iff $\rho_v(uv)\rho_v(vw) \in E(G_v)$
  for all tree-edges uv, vw on the x, y-path in T

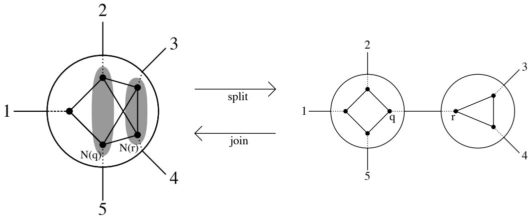


A split in a graph $G = (V, E)$ is a bipartition (A, B) of V st.

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



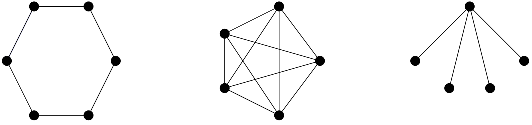
Every **edge** of a GLT defines a **split**



Theorem [Cunnigham'82 reformulated]

For any connected graph G, there exists a unique graph-labelled tree $ST(G) = (T, \mathcal{F})$ with a minumun number of nodes such that

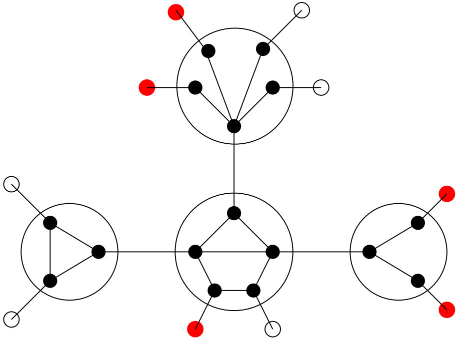
- $G = G(T, \mathcal{F})$,
- any graph of $\mathcal{F}$ is prime or degenerate.



Prime Degenerate

Vertex incremental characterization

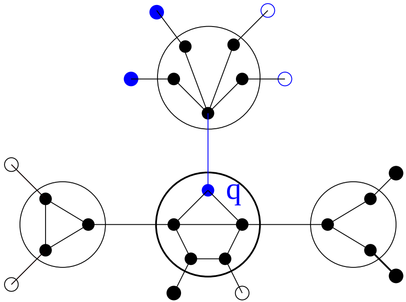
A vertex x is added to a graph $G = (V, E)$ with neighbors $S \subseteq V$.
(*S* is represented in red)



- How to compute the split tree $ST(G + (x, S))$ from $ST(G)$?

States of marker vertices

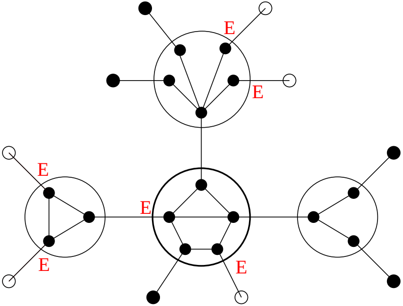
Let q be marker vertex (or a leaf)
Let $L(q)$ be the set of leaves l such that the path between l and q contains the edge of the tree associated with q



States of marker vertices

Let q be marker vertex (or a leaf)
Let $L(q)$ be the set of leaves l such that the path between l and q contains the edge of the tree associated with q

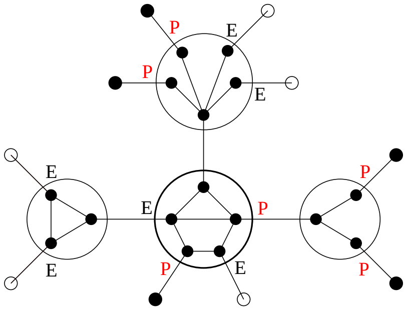
- q is empty if $L(q) \cap S = \emptyset$



States of marker vertices

Let q be marker vertex (or a leaf)
Let $L(q)$ be the set of leaves l such that the path between l and q contains the edge of the tree associated with q

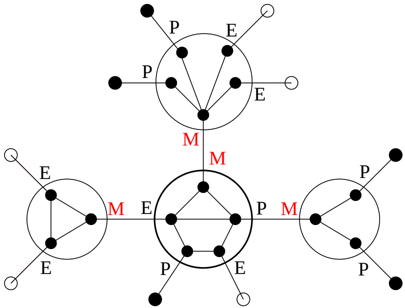
- q is empty if $L(q) \cap S = \emptyset$
- q is perfect if every leaf in $L(q) \cap S$ is accessible from q



States of marker vertices

Let q be marker vertex (or a leaf)
Let $L(q)$ be the set of leaves l such that the path between l and q contains the edge of the tree associated with q

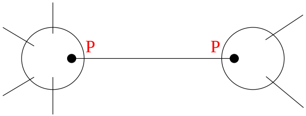
- q is empty if $L(q) \cap S = \emptyset$
- q is perfect if every leaf in $L(q) \cap S$ is accessible from q
- q is mixed otherwise



Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

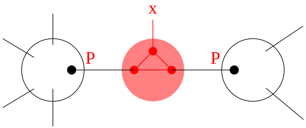
- 1 - T has an edge e whose extremities are perfect



Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

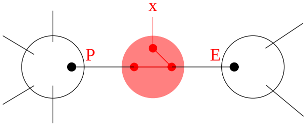
- 1 - T has an edge e whose extremities are perfect



Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

- 1 - T has an edge e whose extremities are perfect
- 2 - T has an edge e, one extremity of which is perfect and the other empty

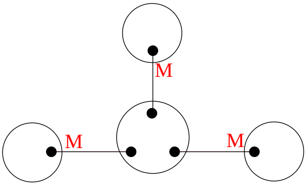


Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

- 1 - T has an edge e whose extremities are perfect
- 2 - T has an edge e, one extremity of which is perfect and the other empty
- 3 - T has a fully-mixed node u

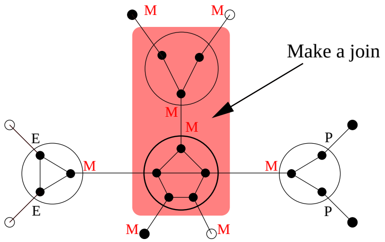
every marker vertex adjacent to the node u is mixed



Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

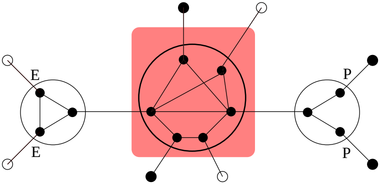
- 1 - T has an edge e whose extremities are perfect
- 2 - T has an edge e, one extremity of which is perfect and the other empty
- 3 - T has a fully-mixed node u (the set of fully-mixed nodes is a subtree of T)



Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

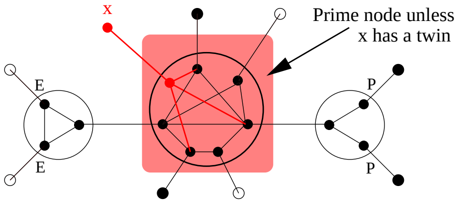
- 1 - T has an edge e whose extremities are perfect
- 2 - T has an edge e, one extremity of which is perfect and the other empty
- 3 - T has a fully-mixed node u (the set of fully-mixed nodes is a subtree of T)



Vertex Incremental Characterization

The split tree of $G + (x, S)$ is obtained from $ST(G)$ by 3 cases:

- 1 - T has an edge e whose extremities are perfect
- 2 - T has an edge e, one extremity of which is perfect and the other empty
- 3 - T has a fully-mixed node u (the set of fully-mixed nodes is a subtree of T)



Theorem: the incremental split decomposition algorithm runs in
$$O((n+m)\alpha(n,m))$$

Ingredients for the complexity analysis

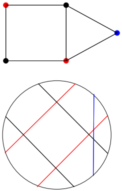
- the vertex insertion ordering has to be a LexBFS ordering
- a carefully amortizing analysing based on discharging techniques

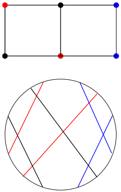
Previous results

- [Cunningham 1982] $O(nm)$
- [Ma, Spinrad 1994] $O(n^2)$
- [Dahlhaus 2000] $O(n+m)$ (tough !)
- [Charbit, De Montgolfier, Raffinot 2009] $O(n+m)$ (today)

Application to circle graphs recognition

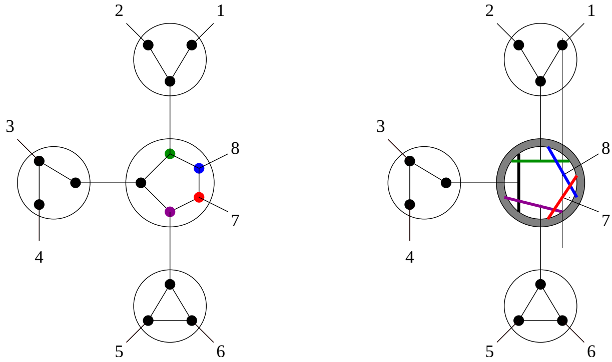
- A **circle** graph is the intersection graph of chords in a circle.





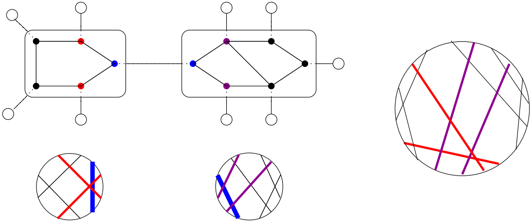
Application to circle graphs recognition

- A **circle** graph is the intersection graph of chords in a circle.
- A graph is **circle** iff all the prime nodes of its split tree are circle graphs



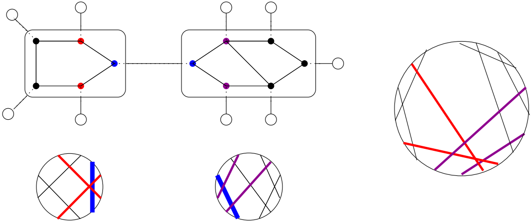
Application to circle graphs recognition

- A **circle** graph is the intersection graph of chords in a circle.
- A graph is **circle** iff all the prime nodes of its split tree are circle graphs
- A circle graph is prime for the split decomposition iff it has a unique (up to mirror) realizer



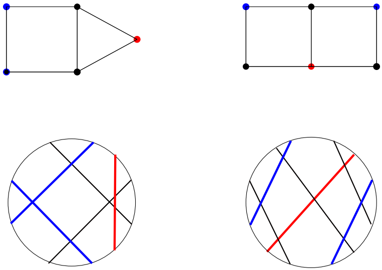
Application to circle graphs recognition

- A **circle** graph is the intersection graph of chords in a circle.
- A graph is **circle** iff all the prime nodes of its split tree are circle graphs
- A circle graph is prime for the split decomposition iff it has a unique (up to mirror) realizer



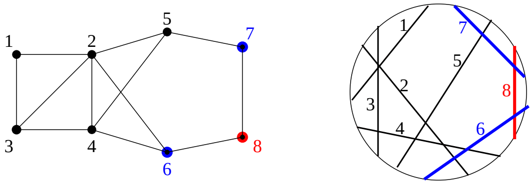
Extreme chord - vertex

A vertex x is **extreme** if its chord $c(x)$ cut the realizer in a way that the chords of all the non-neighors of x are either all on the right or all on the left of $c(x)$.



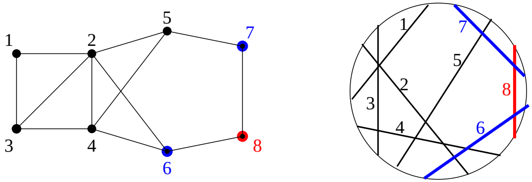
First LexBFS Lemma

Let G be a circle graph and σ be a LexBFS ordering of G ending at x. Then there exists a realizer of G in which the chord $c(x)$ is extreme.



First LexBFS Lemma

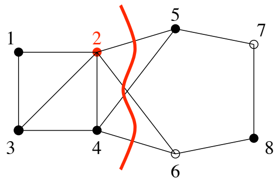
Let G be a circle graph and σ be a LexBFS ordering of G ending at x. Then there exists a realizer of G in which the chord $c(x)$ is extreme.

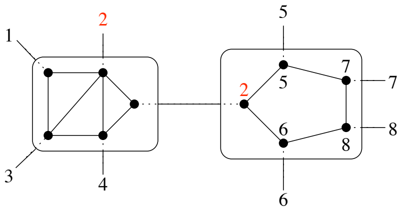


- Consequence: we know in constant time where to insert x in a given realizer of G.
- But many realizers unless prime

A second LexBFS Lemma

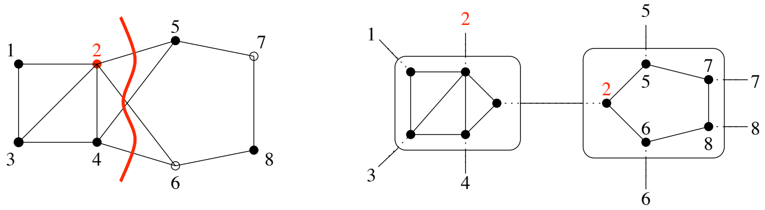
Let σ be a LexBFS ordering of $G = G_S(T, \mathcal{F})$. The *"induced"* ordering σ_u of the marker vertices of G_u is a LexBFS ordering.





A second LexBFS Lemma

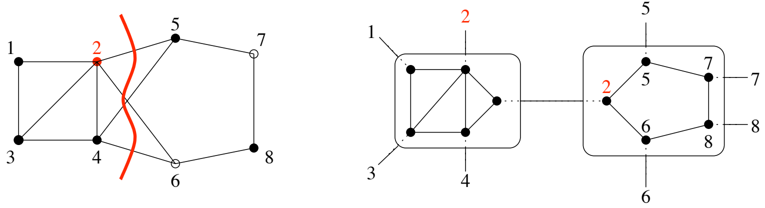
Let σ be a LexBFS ordering of $G = G_S(T, \mathcal{F})$. The *"induced"* ordering σ_u of the marker vertices of G_u is a LexBFS ordering.



- consequence: we can apply the first LexBFS lemma on each node of the split tree.

A second LexBFS Lemma

Let σ be a LexBFS ordering of $G = G_S(T, \mathcal{F})$. The *"induced"* ordering σ_u of the marker vertices of G_u is a LexBFS ordering.



- consequence: we can apply the first LexBFS lemma on each node of the split tree.
- what remains to do ?
  handle the merging of the fully-mixed nodes before insertion.

Circle graph recognition - ingredients

1. Insert vertices with respect to a LexBFS ordering.
2. Maintain the split tree in which each prime node are represented by its realizer.
3. Proceed the merging step on the realizers of the fully-mixed nodes according to the LexBFS Lemmas.
   $\Rightarrow$ need some small tricks for realizer representation

Circle graph recognition - ingredients

1. Insert vertices with respect to a LexBFS ordering.
2. Maintain the split tree in which each prime node are represented by its realizer.
3. Proceed the merging step on the realizers of the fully-mixed nodes according to the LexBFS Lemmas.
   $\Rightarrow$ need some small tricks for realizer representation

Theorem: The circle graph recognition problem can be solved in time $O((n+m)\alpha(n,m))$

$\rightarrow$ Previous complexity: $O(n^2)$ [Spinrad, J. of Alg. (16), 1994]

Thank you...

- Int. Workshop on Graph Theoretical Concepts in Computer Science
  WG 2009
  Montpellier, June 24-26th
  http://www.lirmm.fr/wg2009/

- Spring School on Fixed Parameter and Exact Algorithms
  AGAPE 2009
  May 25-29, Lozari, Corsica
  http://www-sop.inria.fr/mascotte/seminaires/AGAPE/