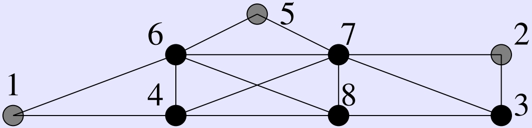
Last course

Michel Habib
habib@irif.fr
http://www.irif.fr/~habib

28 octobre 2016

Schedule

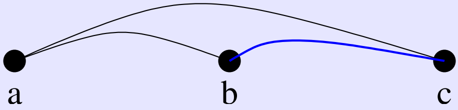
Chordal graph



A vertex is simplicial if its neighbourhood is a clique.

Simplicial elimination scheme

$\sigma = [x_1 \dots x_i \dots x_n]$ is a simplicial elimination scheme if x_i is simplicial in the subgraph $G_i = G[\{x_i \dots x_n\}]$



Lexicographic Breadth First Search (LBFS)

Data: a graph $G = (V, E)$ and a start vertex s

Result: an ordering σ of V

Assign the label $\emptyset$ to all vertices

$label(s) \leftarrow \{n\}$

for $i \leftarrow n$ **à** 1 **do**

    Pick an unumbered vertex v with lexicographically largest label

    $\sigma(i) \leftarrow v$

    foreach *unnumbered vertex* w *adjacent to* v **do**

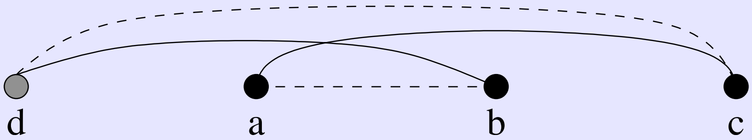
        $label(w) \leftarrow label(w).\{i\}$

    end

end

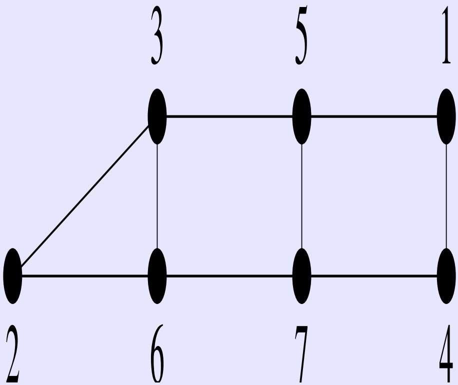
Property (LexB)

For an ordering σ on V, if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then it must exist a vertex d such that $d <_\sigma a$ et $db \in E$ et $dc \notin E$.



Theorem

For a graph $G = (V, E)$, an ordering σ on V is a LBFS of G iff σ satisfies property (LexB).


3
5
1
2
6
7
4

Theorem [Tarjan et Yannakakis, 1984]

G is chordal iff every LexBFS ordering yields a simplicial elimination scheme.

One of the standard examples of antimatroid

Let G be a chordal graph.
$\mathcal{F}= \{X \subseteq V(G) | X$ is the beginning of a simplicial elimination scheme of $G\}$
$AM = (V, \mathcal{F})$ is an antimatroid.

proof

Axiomes 1 and 2 are trivially satisfied by $\mathcal{F}$.
Let us consider the third one. First we use the Y as the starting of a simplicial elimination of G. Consider the ordering τ of X which is the starting of a simplicial elimination scheme of G. Let σ be an elemination scheme starting with Y. Let x_1 the first element of $X \setminus Y$ with respect to σ.
Necessarily just after Y we can use x_1 as a simplicial vertex in the remaining graph, since all necessary vertices as been eliminated before (in $Y \cap X$) and x_1 was simplicial in τ.
Therefore $Y \cup \{x_1\}$ is the beginning of a simplicial elimination scheme of G.

So far for a chordal graph

1. The edges of the tree structure associated to any graph search is a branching greedoid (this does not depend on chordal graphs and is true for any undirected graph),
2. But also the beginnings of any LBFS (resp. LDFS) visiting ordering considered as sets of vertices define a series of Monophonic convex sets.

- Points 2 and 3 are not true for BFS or DFS

Natural question

Which are the conditions that must satisfies an elimination scheme for a particular class of graphes to yield an antimatroid ?
(false or true twins for cographs, pending vertex and twin for distance hereditary graphs . . .)

Main chordal graphs characterization theorem

Using results of Dirac 1961, Fulkerson, Gross 1965, Buneman 1974, Gavril 1974 and Rose, Tarjan and Lueker 1976 :

For a connected graph, the following statements are equivalent and characterize chordal graphs :

(0) G has no induced cycle of length > 3

(i) G admits a simplicial elimination scheme

(ii) Every minimal separator is a clique

(iii) G admits a maximal clique tree.

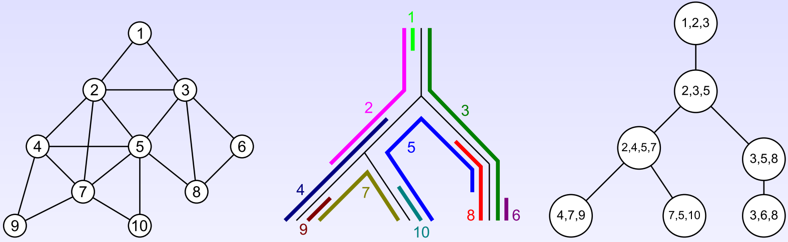
(iv) G is the intersection graph of subtrees in a tree.

(v) Any MNS (LexBFS, LexDFS, MCS) provides a simplicial elimination scheme.

Two subtrees intersect iff they have at least one vertex in common, not necessarily an edge in common.

By no way, these representations can be uniquely defined !

An example



Consequences of maximal clique tree

Theorem

Every minimal separator belongs to every maximal clique tree.

Lemma

Every minimal separator is the intersection of at least 2 maximal cliques of G.

Corollary

There are at most n minimal separators.

Proof of the lemma

Démonstration.

Since G is chordal, every minimal separator S is a clique. Let us consider G_1 a connected component of G-S. Let $x_1, \ldots, x_k$ be the vertices of G_1 having a maximal neighbourhood in S.
If $k = 1$ then x_1 must be universal to S, since S is a minimal separator. Then $x_1 \cup S$ is a clique.
Else, consider a shortest path μ in G_1 from x_1 to x_k. Necessarily x_1 (resp. x_k) has a private neighbour z (resp. t) in S. Else they would have the same maximal neighbourhood in S, and since S is a minimal separator, this neighbourhood must be S. And then $x_1 \cup S$ is a clique.
Else the cycle $[x_1, \mu, x_k, t, z]$ has no chord, a contradiction.
Therefore $x_1 \cup S$ is a clique, and is contained in some maximal clique C in G_1. We finish the proof by considering another connected component of $G - S$.

Proof of the theorem

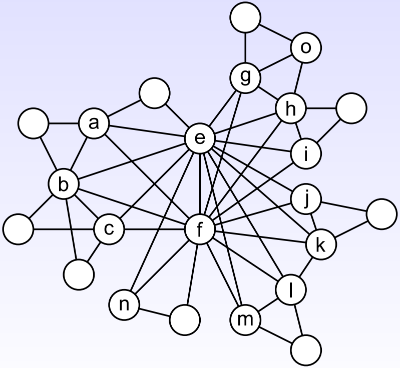
Démonstration.

Therefore $S = C' \cap C''$. These two maximal cliques belong to any maximal clique tree T of G. Let us consider the unique path μ in T joigning C' to C''.
All the internal maximal cliques in μ must contain S. Suppose that all the edges of μ are labelled with minimal separators strictly containing S, then we can construct a path in G from $C' - S$ to $C'' - S$ avoiding S, a contradiction. So at least one edge of μ is labelled with S.

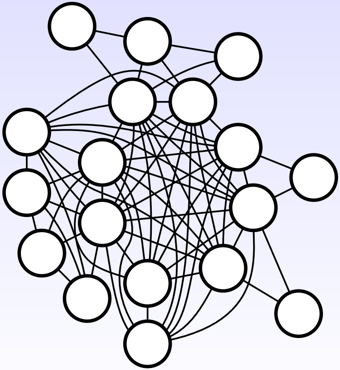
□

Clique graph

the *clique graph* $\mathcal{C}(G)$ of G = intersection graph of maximal cliques of G



G



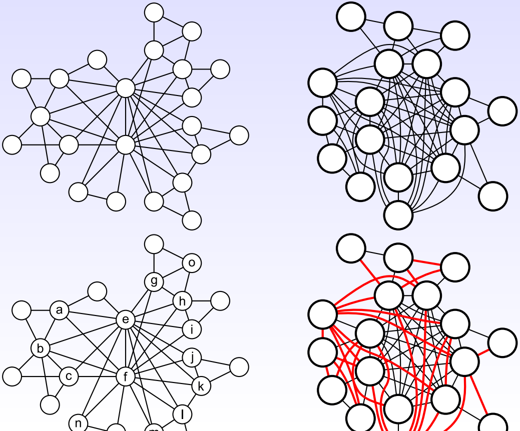
$\mathcal{C}(G)$

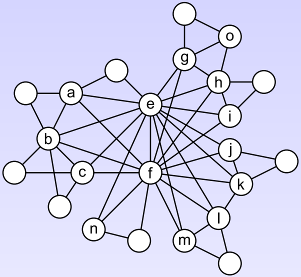
- If G is connected then also $\mathcal{C}(G)$ [1] is connected.
- When G is chordal, every maximal clique tree of G is a spanning tree of $\mathcal{C}(G)$.
- Edges of $\mathcal{C}(G)$ correspond to clique intersection, but they are not all minimal separators.
- So we introduced the reduced clique graph.

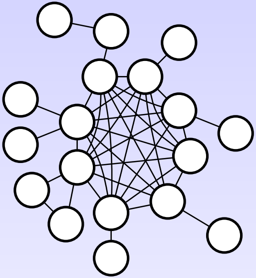
1. $\mathcal{C}(G)$ is also denoted as $K(G)$

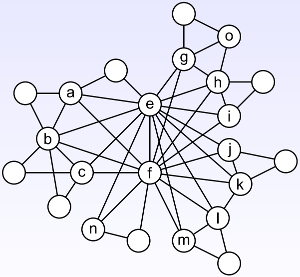
Reduced clique graph

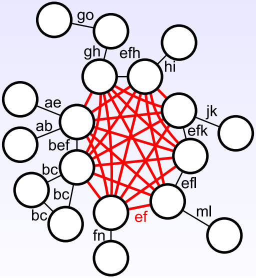
the *reduced clique graph* $\mathcal{C}_r(G)$ of G = graph on maximal cliques of G where CC' is an edge of $\mathcal{C}_r(G)$ $\iff$ $C \cap C'$ is a minimal separator.




o
g
h
a
e
i
b
j
c
f
k
l
n
m




o
g
h
a
e
i
b
j
c
f
k
l
n
m


go
gh
efh
hi
ae
ab
jk
bef
efk
bc
bc
efl
bc
ml
ef
fn

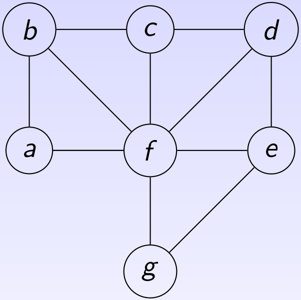


FIGURE: G a chordal graph

FIGURE: Its reduced clique graph.

On this example $\mathcal{C}(G)$ is the complete graph on these 5 maximal cliques, and $\mathcal{C}_r(G)$ is the unique maximal clique tree.

Size of $\mathcal{C}_r(G)$

Considering a star on n vertices, for which $\mathcal{C} = \mathcal{C}_r(G)$ and is a complete graph

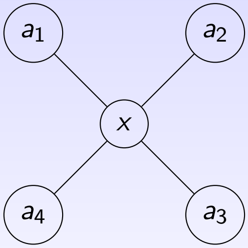


FIGURE: A star

shows $|\mathcal{C}_r(G)|$ is $\in \Omega(n^2)$
As a consequence $|\mathcal{C}_r(G)|$ is not always linear in the size of G

Combinatorial structure of $\mathcal{C}_r(G)$

Lemma 1 : M.H and C. Paul 95

If C_1, C_2, C_3 is a triangle in $\mathcal{C}_r(G)$, with S_{12}, S_{23} and S_{23} be the associated minimal separators then two of these three separators are equal and included in the third.

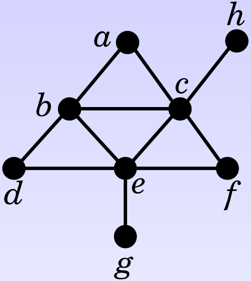
Lemma 2 : M.H. and C. Paul 95

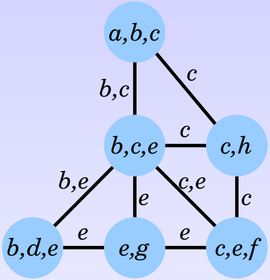
Let C_1, C_2, C_3 be 3 maximal cliques, if $C_1 \cap C_2 = S_{12} \subset S_{23} = C_2 \cap C_3$ then (C_1, C_2, C_3) is a triangle in $\mathcal{C}_r(G)$

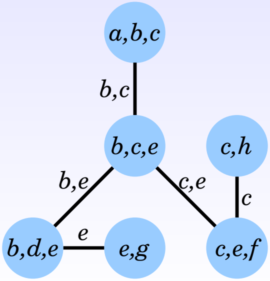
Lemma 3 : Equality case

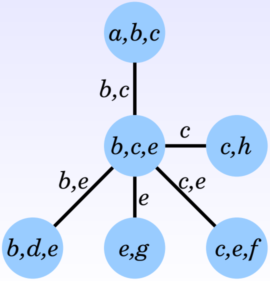
Let C_1, C_2, C_3 be 3 maximal cliques, if $S_{12} = S_{23}$ then :

- either the $C_1 \cap C_3 = S_{13}$ is a minimal separator
- or the edges $C_1 C_2$ and $C_2 C_3$ cannot belong together to a maximal clique tree of G.


a
h
c
b
d
e
f
g


a,b,c
b,c
c
b,c,e
c
c,h
b,e
e
c,e
c
b,d,e
e
e,g
e
c,e,f


a,b,c
b,c
b,c,e
c,h
b,e
c,e
c
b,d,e
e
e,g
c,e,f


a,b,c
b,c
b,c,e
c
c,h
b,e
e
c,e
b,d,e
e,g
c,e,f

Consequences

- If T, T' are two maximal clique trees, there exists a sequence of exchange on triangles to go from T to T'.
- All clique trees have exactly the same labels, including repetitions.

Theorem (Gavril 87, Shibata 1988, Blayr and Payton 93)

The clique trees of G are precisely the maximum weight spanning trees of $\mathcal{C}(G)$ where the weight of an edge CC' is defined as $|C \cap C'|$.

Theorem (Galinier, Habib, Paul 1995)

The clique trees of G are precisely the maximum weight spanning trees of $\mathcal{C}_r(G)$ where the weight of an edge CC' is defined as $|C \cap C'|$.

Moreover, $\mathcal{C}_r(G)$ is the union of all clique trees of G.

Maximal Cardinality Search : MCS

Data: a graph $G = (V, E)$ and a start vertex s
Result: an ordering σ of V
Assign the label 0 to all vertices
$label(s) \leftarrow 1$
for $i \leftarrow n$ **à** 1 **do**
    Pick an unnumbered vertex v with largest label
    $\sigma(i) \leftarrow v$
    foreach *unnumbered vertex* w *adjacent to* v **do**
        $label(w) \leftarrow label(w) + 1$
    end
end

Maximum spanning trees

Maximal Cardinality Search can be seen as Prim algorithm for computing a maximal spanning tree of $\mathcal{C}_r(G)$.

Let us consider a weighted undirected graph G, equipped with a valuation $\omega : E(G) \rightarrow \mathcal{R}^+$.

Algorithm 1: Prim's Algorithm for maximal spanning tree

$F \leftarrow \emptyset$;
$A \leftarrow \{x_0\}$;
while $|F| < n - 1$ **do**
  Compute the edge $e = (a, x)$ (with $a \in A$) with maximum weight of the cocycle $\omega(A)$;[2] $F \leftarrow F \cup \{e\}$;
  $A \leftarrow A \cup \{x\}$
end

2. a cocycle is also called a cut

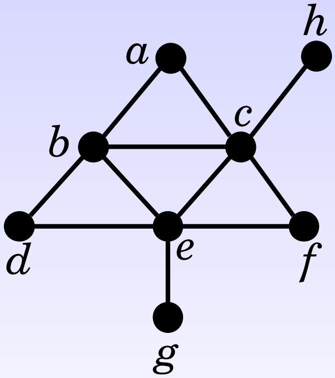
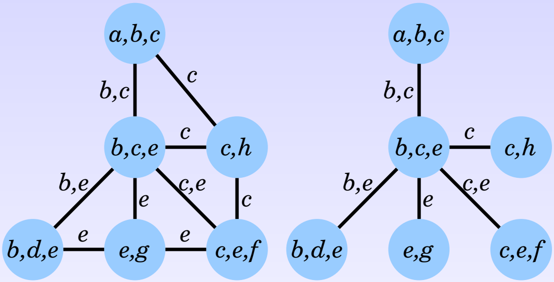


FIGURE: A MCS-ordering : $\sigma = a, b, c | e | d | f | g | h$



$\sigma = a, b, c | e | d | f | g | h$ induces an ordering
$\tau = (abc)(bce)(bde)(cef)(cg)(gh)$ on the maximal cliques, which yields the maximal spanning tree on the right.

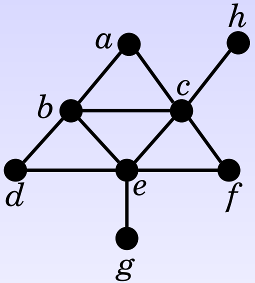
The tricks is :

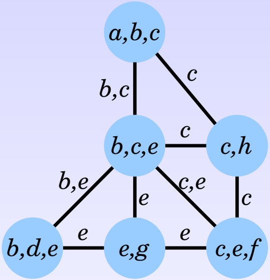
$\sigma = a, b, c | e | d | f | g | h$

| means the sequence of labels is not strictly increasing

LexBFS, MCS or MNS visit maximal cliques "consecutively" (i.e. when the search explores a vertex x of a maximal clique C that does not belong to any of the previously visited maximal cliques then all the unvisited vertices of C will appear consecutively just after x).

Therefore when applying a search (LBFS, MCS or MNS) one can compute a clique tree, by considering the strictly increasing sequences of labels.





$b, a, c|e|d|h|f|g$ is a LBFS ordering.
we can find the maximal cliques abc then bc, e then be, d then c, h then ce, f and e, g.

- How to compute the minimal separators and a maximal clique tree ?
- Keeping track of the labels (for example in LBFS) yields the minimal separator.
- To obtain a tree, just attach the new visited clique to the last one that contains this minimal separator.
- There is some degree of freedom to associate a maximal clique tree to a LBFS

Simplicial elimination schemes

1. Choose a maximal clique tree T
2. While T is not empty do
   Select a vertex $x \in F - S$ in a leaf F of T ;
   $F \leftarrow F - x$;
   If $F = S$ delete F ;

Canonical simplicial elimination scheme

1. Choose a maximal clique tree T
2. While T is not empty do
   Choose a leaf F of T ;
   Select successively all vertices in $F - S$
   delete F ;

Exercises :

1. Does there exist other simplicial elimination scheme ?
2. We know that MNS, LBFS, LDFS, MCS provide simplicial elimination schemes, but which graph search can generate them all ?

Size of a maximal clique tree in a chordal graph

- Let $G = (V, E)$ be a chordal graph.
- G admits at most $|V|$ maximal cliques and therefore the tree is also bounded by $|V|$ (vertices and edges).
- But some vertices can be repeated in the cliques. If we consider a simplicial elimination ordering the size of a given maximal clique is bounded by the neighbourhood of the first vertex of the maximal clique.
- Therefore any maximal clique tree is bounded by $|V| + |E|$.

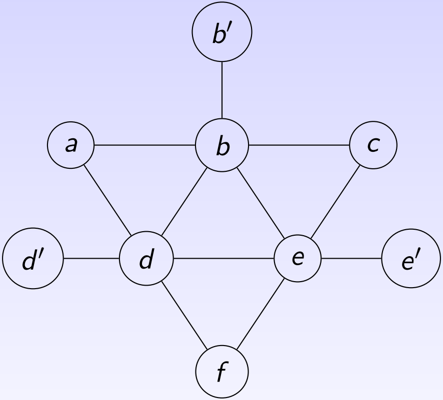
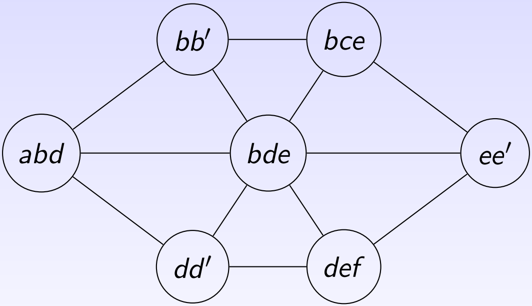


FIGURE: G a chordal graph

FIGURE: $\mathcal{CS}(G)$



Unfortunately $\mathcal{CS}(G)$ is not always chordal !

In fact $\mathcal{C}_r(G)$ is dually chordal (almost chordal) and $\mathcal{C}_r(\mathcal{C}_r(G))$ is chordal.

Cannonical representation

- For an interval graph, its PQ-tree represents all its possible models and can be taken as a cannonical representation of the graph (for example for graph isomorphism)
- But even path graphs are isomorphism complete. Therefore a canonical tree representation is not obvious for chordal graphs.
- $\mathcal{C}_r(G)$ is a Pretty Structure to study chordal graphs.
  To prove structural properties of all maximal clique trees of a given chordal graph.

Results so far on $\mathcal{C}_r(G)$ as a labelled graphs

Reduced clique graphs can be completely characterized as graphs.
They are perfect and have many properties.

	Maximum w. Hamilton Path	Leafage
G arbitrary	NP-complete	NP-complete
Labelled $\mathcal{C}_r(G)$	linear Interval graph recognition	polynomial $O(n^3)$

- Philippe Galinier, Michel Habib, Christophe Paul : Chordal Graphs and Their Clique Graphs. WG 1995 : 358-371
- Michel Habib, Ross M. McConnell, Christophe Paul, Laurent Viennot : Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. Theor. Comput. Sci. 234(1-2) : 59-84 (2000)
- Michel Habib, Juraj Stacho : Polynomial-Time Algorithm for the Leafage of Chordal Graphs. ESA 2009 : 290-300
- Michel Habib, Juraj Stacho : Reduced clique graphs of chordal graphs. Eur. J. Comb. 33(5) : 712-735 (2012)

Exercise 1

1. Show that the last maximal clique visited can be taken as the end of some chain of cliques if G is an interval graph.
2. If we consider the edges of the clique tree labelled with the size of the minimal separators, show that :
   for every maximal clique tree T
   $weight(T) = \Sigma_{1 \leq i \leq k} |C_i| - n$, where $C_1, \ldots C_k$ are the maximal cliques of G.

Characterization Theorem for interval graphs (Folklore)

(i) $G = (V, E)$ is an interval graph, i.e. ; G is the intersection graph of a family of intervals of the real line

(ii) There exists a total ordering τ of the vertices of V s.t. $\forall x, y, z \in G$ with $x \leq_\tau y \leq_\tau z$ and $xz \in E$ then $xy \in E$.

(iii) G admits a maximal clique tree which is a chain.

(iii) implies (i) is obvious, since the chain of maximal cliques gives the interval representation.
(i) implies (iii). From the interval representation it is easy $O(n)$ to compute the sequence of maximal cliques.
(i) implies (ii) : one can associate to every vertex x of G an interval $[left(x), right(x)]$ of the real line.
Let us define τ as the left sides ordering : $x \leq_\tau y$ iff $left(x) \leq left(y)$. It is easy to verify that τ satisfies the condition.
(ii) implies (i) : starting with τ, for every vertex x we associate an interval in τ $[x, last(x)]$, where $last(x)$ is its rightmost neighbor in τ. This provides an interval representation of G.

To recognize an interval graph, we just have to compute a maximal clique tree and check if it is a chain ?
Difficulty : an interval graph has many clique trees and among them some are chains

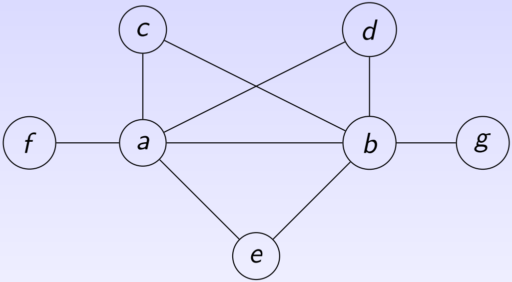


Figure: G a chordal graph

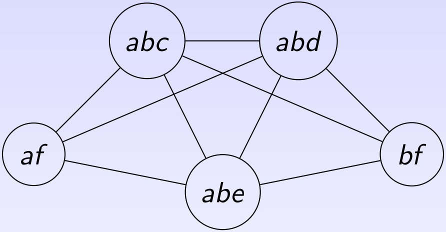


FIGURE: $\mathcal{C}_r(G)$ its reduced clique graph

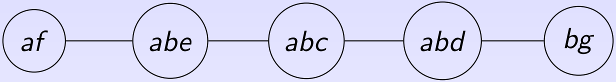


FIGURE: A good maximal clique tree T_1 showing that G is an interval graph

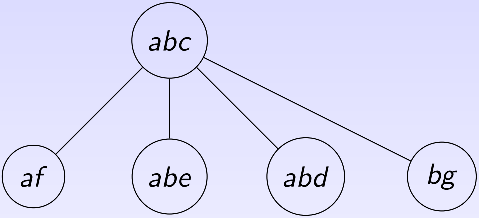


FIGURE: Another maximal clique tree T_2 obtained via LBFS
$\sigma = c, a, b | d | e | f | g$

How can we transform T_2 to obtain T_1 ?

Many linear time algorithms already proposed for interval graph recognition ....
using nice algorithmic tools :
graph searches, modular decomposition, partition refinement, PQ-trees . . .

Linear time recognition algorithms for interval graphs

- Booth and Lueker 1976, using PQ-trees.
- Korte and Mohring 1981 using LBFS and Modified PQ-trees.
- Hsu and Ma 1995, using modular decomposition and a variation on Maximal Cardinality Search.
- Corneil, Olariu and Stewart SODA 1998, using a series of 6 consecutive LBFS, published in 2010.
- M.H, McConnell, Paul and Viennot 2000, using LBFS and partition refinement on maximal cliques.
- P. Li, Y. Wu 2014, using a series of 4 kind of LBFS
- . . .

A partition refinement algorithm working on maximal cliques

1. Compute a tree T using LBFS
   If T is not a maximal clique tree ; then G is not chordal, neither interval.
2. Start from the last maximal clique visited by the search
   Refine the cliques with the minimal separator.
3. Refine until each part is a singleton
4. If a part is not a singleton start recursively from the last clique of this part according to LBFS.
5. Check if the last partition is a chain of maximal cliques.

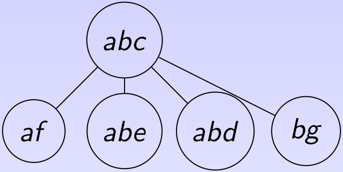


FIGURE: T_2 obtained via LBFS $\sigma = c, a, b|d|e|f|g$

- $\{bg\}|\{abc, abd, abe, af\}$
- refine with b gives :
  $\{bg\}|\{abc, abd, abe\}, |\{af\}$
- refine with a does not change the partition
- abe is the last maximal clique of the central part.
  $\{bg\}|\{abe\}|\{abc, abd\}, |\{af\}$
  refine with a, b does not change the partition
- abd is the last maximal clique of the central part.
  $\{bg\}|\{abe\}|\{abd\}|\{abc\}, |\{af\}$

Variations on these representations

- Path graphs = Paths in a tree
  In between chordal and interval. But still no linear time algorithm for the recognition of this class.
- Directed path graphs = directed paths in a rooted directed tree
  appear in some polynomial CSP class.