

Fully dynamic representations of interval graphs

Mikolaj Jachowicz

May 21, 2026

Table of Contents

1 Definitions

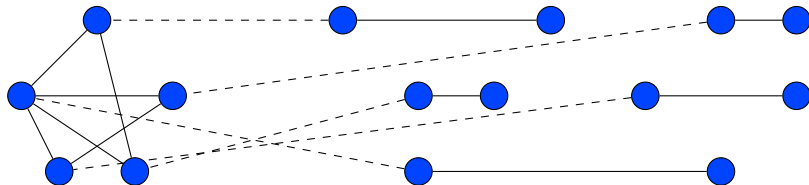
2 Three Representations

3 Dynamic Operations

Interval Graph

Interval Graph

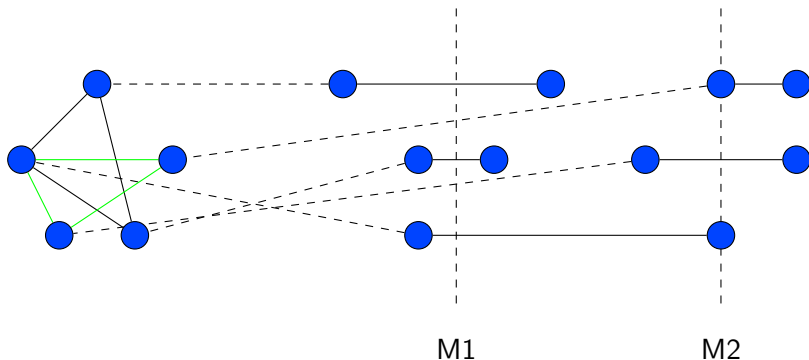
Graph G is an interval graph iff it is formed from a set of intervals on a line, with vertices corresponding to intervals connected iff the respective intervals overlap



Interval Graph Characterization

Interval Graph Characterization

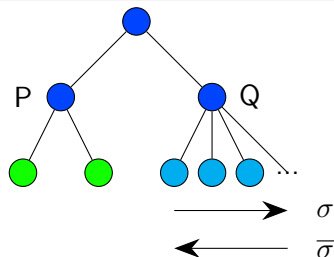
G is an interval graph iff all maximal cliques can be ordered into a sequence $M_1, \dots, M_k$ such that for all vertices v_i there are $i_i^1 \leq i_i^2$ such that $v_i \in M_j \iff j \in [i_i^1, i_i^2]$. This sequence is a Consecutive Ordering



PQ Tree

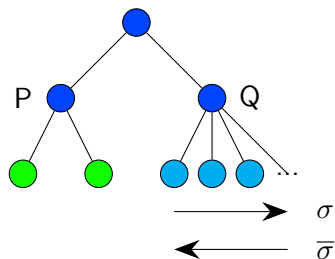
PQ tree

PQ tree is a tree that has nodes labeled Q if they have more than 2 children and P otherwise. Q nodes have 2 linear orderings, σ and $\bar{\sigma}$ that are inverses of one another. Ordering of P nodes is trivial.



Solidification

Solidification of a PQ tree T is T with one of the ordering of every nodes children rejected



Solidification

Solidification of a PQ tree T is T with one of the ordering of every nodes children rejected

For interval graph G there is a (unique) PQ tree with leaves representing maximal cliques such that set of solidifications maps 1-1 to the set of Consecutive Orderings

Modular Decomposition

Module

$M \subseteq V$ is a module iff $\forall u, v \in M, w \notin M : w \in N(v) \iff w \in N(u)$

A strong module is a module that is maximal in terms of inclusion

Modular Decomposition

Modular Decomposition Tree of graph G is a tree where nodes are modules, root is V and children of M are the strong modules of $G[M]$

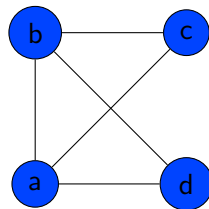
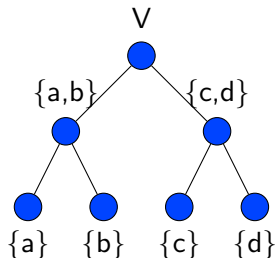


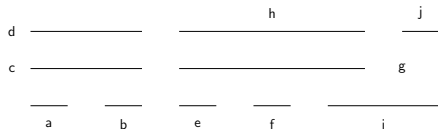
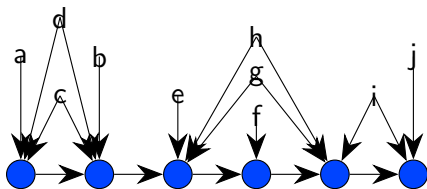
Table of Contents

1 Definitions

2 Three Representations

3 Dynamic Operations

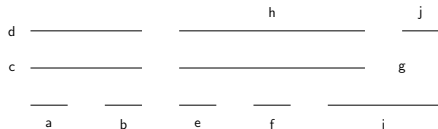
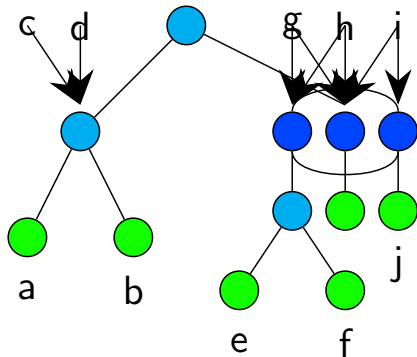
Minimal Interval Model



Minimal Interval Model

Minimal Interval Model corresponds directly to a Consecutive Ordering. Consecutive cliques are nodes on a list, vertices have pointers to nodes corresponding to their first and last clique

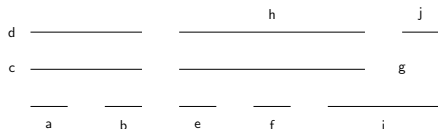
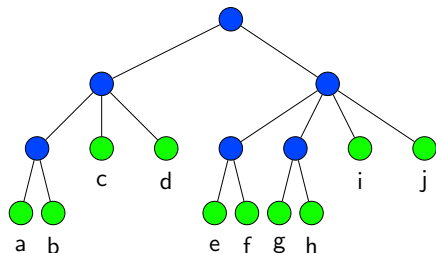
PQ Tree Model



PQ Tree Model

The PQ Tree model of G is the unique PQ Tree of G , leaves correspond to maximal cliques and vertices have pointers to nodes which children all contain that vertex (sometimes 2 pointers for Q nodes)

MD Tree Model



MD Tree Model

MD Tree Model of G contains Modular Decomposition Tree of G
Module Relations Graphs are preserved as MIM

Purpose

Dynamic Operations

Using the 3 representations it is possible to maintain dynamic representation of an interval graph under insertion and deletion operations, in $O(n)$ time per operation

Model Equivalence Lemma

The 3 representations are linearly equivalent i.e. given any representation, all others can be constructed in linear time

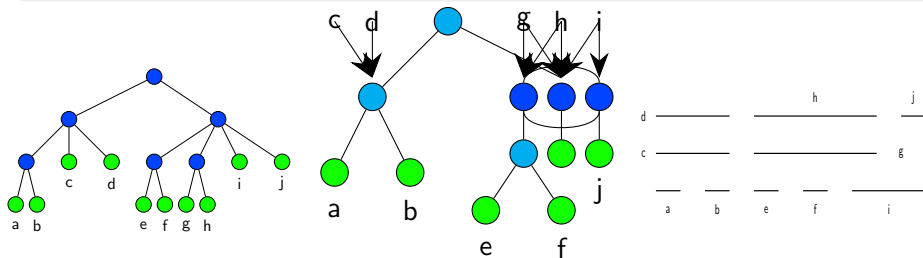


Table of Contents

1 Definitions

2 Three Representations

3 Dynamic Operations

G interval implies $G[V - \{x\}]$ interval

This operation can be completed on MIM

- 1 remove the vertex
- 2 go over the list. If there is a node that isn't pointed to, remove it

$G + x$ may not be an interval graph
this requires proceeding in steps

- 1 marking step - find insertion node w in MD
- 2 testing step - check if $G + x$ is an interval graph using w
- 3 insertion step - insert x into MD or reject x depending on testing results

Insertion - Testing Step

definitions

Set S is linked (relative to x) iff $S \not\subseteq \overline{N(x)}$

S is mixed (relative to x) iff $S \not\subseteq \overline{N(x)} \wedge S \not\subseteq N(x)$

$T_w^c(G)$ is a subtree of PQ-tree of G rooted at w

for node u in T^c , B_u is the set of vertices belonging to at least one of the maximal cliques that correspond to leaves of T_u^c . It is full if all $v \in B_u$ are connected to inserting vertex x

node is mixed iff it's B is mixed

Interval Graph Condition Lemma 1

If $G + x$ is an interval graph and w is the insertion node of x then any node $u \in T_w^c(G) \setminus \{x\}$ has at most one mixed child. If B_u is not full the u has at most one linked child.

Interval Graph Condition Lemma 1

If $G + x$ is an interval graph and w is the insertion node of x then any node $u \in T_w^c(G) \setminus \{x\}$ has at most one mixed child. If B_u is not full the u has at most one linked child.

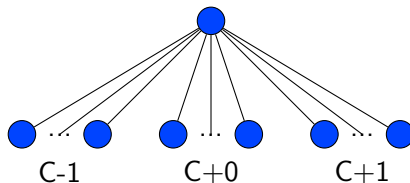
B_u is not full $\implies \exists z \in B_u \setminus N(x)$

assume u has 2 linked children u_1, u_2 their respective sets intersect with $N(x)$ and they must intersect with u . Meaning that must be a 4-cycle which implies $G + x$ is not an interval graph

Insertion - MD Insertion

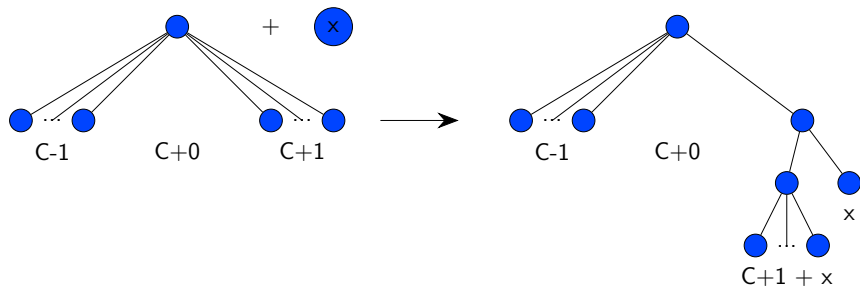
procedure for adding vertex x connected to $N \subseteq V$ to MD tree T

- Mark all modules with 'all-connected', 'none-connected' or 'mixed' based on connection of their vertices to x
- Cases where root of T is not 'mixed' are trivial
- If root of T is mixed then x is either it's own maximal module or is a part of another maximal module



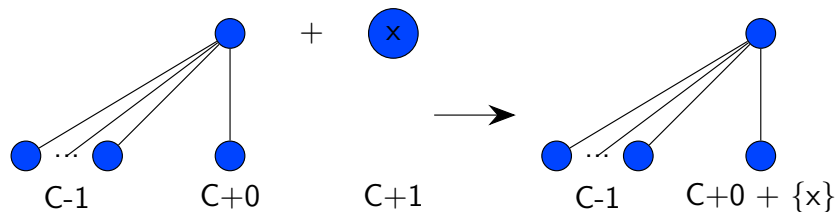
Insertion - MD Insertion

- If root of T is mixed then x is either it's own maximal module or is a part of another maximal module
 - example: G is not connected, $C - 1 / C + 0 / C + 1$ are sets of children of root that are 'all-connected' / 'mixed' / 'none-connected'



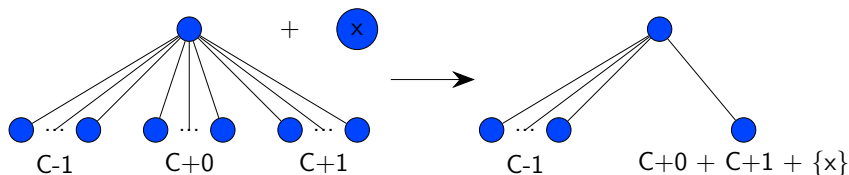
Insertion - MD Insertion

- If root of T is mixed then x is either it's own maximal module or is a part of another maximal module
 - example: G is not connected, $C - 1 / C + 0 / C + 1$ are sets of children of root that are 'all-connected' / 'mixed' / 'none-connected'



Insertion - MD Insertion

- If root of T is mixed then x is either it's own maximal module or is a part of another maximal module
 - example: G is not connected, $C - 1 / C + 0 / C + 1$ are sets of children of root that are 'all-connected' / 'mixed' / 'none-connected'



Insertion - MD Insertion

- Mark all modules with 'all-connected', 'none-connected' or 'mixed' based on connection of their vertices to x
- Cases where root of T is not 'mixed' are trivial
- If root of T is mixed then x is either it's own maximal module or is a part of another maximal module
- modules which are 'mixed' may be broken by addition, have to be reexamined

Addendum: Simpler Algorithm

If $O(n + m)$ time was given per operation this problem is trivial:
On hash-structures based adjacency matrix equivalent insertion/deletion
can be done in $O(n)$ time
There is a $O(n + m)$ time algorithm for recognition interval graphs

Fin