Interval Graph Isomorphism

Contents.

- Intersection graphs.
- Interval graphs. Interval representation.
- Characterization of interval graphs. Matrix formulation.
- Recognition of interval graphs. $PQ$-trees.
- Isomorphism of interval graphs.

Bibliography.

Interval Graph Isomorphism

Definition. The intersection graph of a family of nonempty sets is obtained by representing each set in the family by a vertex and connecting two vertices by an edge if and only if their corresponding sets have nonempty intersection.

- An undirected graph is the intersection graph of an arbitrary family of sets.
- An interval graph is the intersection graph of a family of intervals of a linearly ordered set.
- A circular-arc graph is the intersection graph of a family of arcs of the circle.
- A chordal graph is the intersection graph of a family of subtrees of a tree.
Interval Graph Isomorphism

**Definition.** An undirected graph is called an *interval graph* if its vertices can be put into one-to-one correspondence with a set of intervals of a linearly ordered set, such that two vertices are connected by an edge if and only if their corresponding intervals have nonempty intersection.

**Example.** The interval graph

![Graph Diagram]

has, for instance, the following interval representation:

\[
\begin{array}{ccc}
I_1 & I_3 & I_6 \\
I_2 & I_4 & I_7 \\
I_5 & I_8
\end{array}
\]
Interval Graph Isomorphism

**Theorem.** An undirected graph $G$ is an interval graph if and only if the maximal cliques of $G$ can be linearly ordered such that, for every vertex $v$ of $G,$ the maximal cliques containing vertex $v$ occur consecutively.


**Example.** Consider the following interval representation of the interval graph of the previous example:

```
K_2  K_4  K_4  K_2  K_2
```

![Diagram of interval graph]

---

---
Interval Graph Isomorphism

The previous theorem has an interesting matrix formulation.

**Definition.** A matrix whose entries are zeros and ones, is said to have the *consecutive ones property for columns* if its rows can be permuted in such a way that the ones in each column occur consecutively.

**Example.** The following matrix has the consecutive ones property for columns:

\[
\begin{pmatrix}
1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}
\]

**Example.** The following matrix does not have the consecutive ones property for columns:

\[
\begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
Interval Graph Isomorphism

**Definition.** The **clique matrix** of an undirected graph is an incidence matrix having maximal cliques as rows and vertices as columns.

**Corollary.** An undirected graph $G$ is an interval graph if and only if the clique matrix of $G$ has the consecutive ones property for columns.


**Proof.** Let $G$ be an undirected graph and $M$ the clique matrix of $G$. An ordering of the maximal cliques of $G$ corresponds to a permutation of the rows of $M$. The corollary follows from the Gilmore-Hoffman theorem. □

**Example.** The clique matrix of the interval graph of the previous example can be permuted as follows:

$$
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}
$$
Interval Graph Isomorphism

**Theorem.** Interval graphs can be recognized in $O(n + m)$ time.


Sketch of recognition algorithm. Given an undirected graph $G = (V, E)$,

1. Verify that $G$ is chordal and, if so, enumerate its maximal cliques.
2. Test whether or not the cliques can be ordered so that those which contain vertex $v$ occur consecutively, for each $v \in V$.

- Step 1 takes $O(n + m)$ time and produces at most $n$ maximal cliques. [Details will be given in the lecture on chordal graph isomorphism.]
- Step 2 also takes $O(n + m)$ time, because the clique matrix of an interval graph has $O(n + m)$ nonzero entries, and PQ-trees allow to test a zero-one matrix with $r$ rows, $c$ columns, and $f$ nonzero entries for the consecutive ones property for columns in $O(r + c + f)$ time.
**Interval Graph Isomorphism**

**Definition.** Given a finite set $X$ and a collection $F$ of subsets of $X$, the **consecutive arrangement problem** is to determine whether or not there exists a permutation $\pi$ of $X$ in which the elements of each subset $S \in F$ appear as a consecutive subsequence of $\pi$.

- $X$ is the set of maximal cliques of $G$.
- $F = \{ S(v) \mid v \in V \}$, where $S(v)$ is the set of all maximal cliques of $G$ containing vertex $v$.

The consecutive arrangement problem (over a finite set $X$) and the consecutive ones problem (over a zero-one matrix $M$) are equivalent.

- Each row of $M$ corresponds to an element of $X$.
- Each column of $M$ corresponds to a subset of $X$ consisting of those rows of $M$ containing a one in the specified column.

The $PQ$-tree is a data structure allowing to represent in a small amount of space all the permutations of $X$ which are consistent with the constraints of consecutivity determined by $F$. 
Interval Graph Isomorphism

**Definition.** A **PQ-tree** $T$ is a rooted, ordered tree whose nonterminal nodes fall into two classes, namely $P$-nodes and $Q$-nodes.

- The children of a $P$-node occur in no particular order, while those of a $Q$-node appear in an order which must be locally preserved.
- $P$-nodes are designated by circles and $Q$-nodes by wide rectangles.
- The leaves of $T$ are labeled bijectively by the elements of set $X$.

**Definition.** *The frontier* of a **PQ**-tree is the permutation of $X$ obtained by reading the labels of the leaves from left to right. The frontier of a node is the frontier of the subtree rooted at the node.

**Example.** *The frontier of the following PQ-tree is* $[A, B, C, D, E, F, G, H, I, J]$.

![Diagram of a PQ-tree](image-url)
**Interval Graph Isomorphism**

**Definition.**  A $PQ$-tree is **proper** if each $P$-node has at least two children, and each $Q$-node has at least three children.

All $PQ$-trees to be considered henceforth are assumed to be proper.

**Definition.**  Two $PQ$-trees $T_1$ and $T_2$ are **equivalent**, denoted $T_1 \equiv T_2$, if one can be obtained from the other by applying a sequence of the following transformation rules:

1. Arbitrarily permute the children of a $P$-node.
2. Reverse the children of a $Q$-node.

**Example.**  The following $PQ$-tree is equivalent to the one of the previous example.
Interval Graph Isomorphism

Definition. An ordering of the leaves of a PQ-tree $T$ is consistent with $T$ if it is the frontier of a PQ-tree equivalent to $T$. The set of all orderings consistent with $T$ is called the consistent set of $T$, and is denoted $\text{consistent}(T)$.

Let $X = \{x_1, x_2, \ldots, x_n\}$. The class of consistent permutations of PQ-trees over $X$ forms a lattice.

- The **null tree** $T_0$ has no nodes and $\text{consistent}(T_0) = \emptyset$.

- The **universal tree** $T_n$ has one internal $P$-node (the root) and a leaf for every element of $X$, and $\text{consistent}(T_n)$ includes all permutations of $X$.

```
     
   x_1  x_2  \ldots  x_n
   /     /   \    /   /
  /     /     \  /     /
 x_1   x_2   \ldots x_n
```

**Interval Graph Isomorphism**

Let $F$ be a collection of subsets of a finite set $X$, and let $\Pi(F)$ denote the collection of all permutations $\pi$ of $X$ such that the elements of each subset $S \in F$ occur as a consecutive subsequence of $\pi$.

**Example.** Let $F = \{\{A, B, C\}, \{A, D\}\}$. Then, $\Pi(F) = \{[D, A, B, C], [D, A, C, B], [C, B, A, D], [B, C, A, D]\}$.

**Theorem.** [Booth and Lueker, 1976]

i. For every collection of subsets $F$ of $X$ there is a $PQ$-tree $T$ such that $\Pi(F) = \text{consistent}(T)$.

ii. For every $PQ$-tree $T$ there is a collection of subsets $F$ of $X$ such that $\Pi(F) = \text{consistent}(T)$.

**Example.** Let $F =$


Interval Graph Isomorphism

The following algorithm calculates $\Pi(F)$.

1: procedure consecutive($X, F, \Pi$)  
2:   let $\Pi$ be the set of all permutations of $X$  
3:   for all $S \in F$ do  
4:     remove from $\Pi$ those permutations in which the elements of $S$ do not occur as a subsequence  
5: end procedure  

Despite the initially exponential size of $\Pi$, $PQ$-trees allow to represent $\Pi$ using only $O(|X|)$ space.

1: procedure consecutive($X, F, \Pi$)  
2:   let $T$ be the universal $PQ$-tree over $X$  
3:   for all $S \in F$ do  
4:     reduce $T$ using $S$  
5: end procedure  

The pattern matching procedure reduce attempts to apply from the bottom to the top of the $PQ$-tree a set of 11 templates, consisting of a pattern to be matched against the current $PQ$-tree and a replacement to be substituted for the pattern.
Interval Graph Isomorphism

Theorem. [Booth and Lueker, 1976] The $PQ$-tree representation $T$ of the class of permutations $\Pi(F)$ can be computed in $O(|F| + |X| + \sum_{S \in F} |S|)$ time.

- $X$ is the set of maximal cliques of $G$.
- Each $S \in F$ is the set of all maximal cliques of $G$ containing a given vertex of $G$.
- $F$ is the set of $S$ for all the vertices of $G$.

Corollary. Let $M$ be a zero-one matrix with $r$ rows, $c$ columns, and $f$ nonzero entries. Then, $M$ can be tested for the consecutive ones property for columns in $O(r + c + f)$ time.

Theorem. [Booth and Lueker, 1976] Interval graphs can be recognized in $O(n + m)$ time. Moreover, if $G$ is an interval graph, then there is an algorithm taking $O(n + m)$ time to construct a proper $PQ$-tree $T$ such that $\text{consistent}(T)$ is the set of orderings of the maximal cliques of $G$ in which, for every vertex $v$ of $G$, the maximal cliques containing vertex $v$ occur consecutively.
Interval Graph Isomorphism

Let $T(G)$ denote the proper $PQ$-tree constructed for an interval graph $G$ by the recognition algorithm.

It turns out that isomorphic interval graphs will have equivalent $PQ$-trees.

**Theorem.** If $T_1$ and $T_2$ are $PQ$-trees, with the same number of leaves, such that $\text{consistent}(T_1) = \text{consistent}(T_2)$, then $T_1 \equiv T_2$.


It is possible, though, for interval graphs which are not isomorphic to have equivalent $PQ$-trees.
Interval Graph Isomorphism

Example. The interval graphs with the following interval representation are not isomorphic,

\[ C_1 \quad C_2 \quad C_3 \quad C_4 \quad C_5 \]

\[ C_1 \quad C_2 \quad C_3 \quad C_4 \quad C_5 \]

but the following tree is a proper PQ-tree of either.

\[ C_1 \quad C_4 \quad C_5 \]

\[ C_2 \quad C_3 \]

Therefore, the PQ-tree will have to be extended with more information about the structure of the interval graph.
Interval Graph Isomorphism

**Definition.** For any vertex $v$ in an interval graph $G$, the **characteristic node** of $v$ in a PQ-tree $T$ of $G$ is the deepest node $x$ in $T$ such that the frontier of node $x$ includes the set of maximal cliques containing vertex $v$.

**Example.** Given the following interval representation of the previous example,

```
I_1   I_3   I_6
I_2   I_4
I_5
I_7   I_8
C_1   C_2   C_3   C_4   C_5
```

the characteristic nodes of all vertices (intervals) are the following:

```
I_2 I_5 I_7

I_1   I_4   I_8
I_3   I_6
```
Interval Graph Isomorphism

Definition. A labeled PQ-tree is a PQ-tree whose nodes are labeled by strings of integers which indicate how the sets of all maximal cliques containing each vertex are distributed over the frontier of the tree, as follows:

- If $x$ is a P-node or a leaf, $\text{label}[x]$ is set to the number of vertices of $G$ which have $x$ as their characteristic node.
- If $x$ is a Q-node, number the children of $x$ as $x_1, x_2, \ldots , x_k$ from left to right. For each vertex $v$ of $G$ having $x$ as characteristic node, form a pair $(i, j)$ such that $x_i$ and $x_j$ are the leftmost and rightmost child of $x$, respectively, whose frontier belongs to the set of maximal cliques containing vertex $v$. Sort all these pairs into lexicographically nondecreasing order and concatenate them to form $\text{label}[x]$.

The resultant labeled PQ-tree is denoted $T_L(G)$. 
Interval Graph Isomorphism

Example. The interval graphs with the following interval representation are not isomorphic,

\[
\begin{align*}
C_1 & \quad C_2 & \quad C_3 & \quad C_4 & \quad C_5 \\
& & & & \\
\end{align*}
\]

and their labeled $PQ$-trees are not equivalent:

\[
\begin{align*}
(1,2)(2,3)(3,4) & \quad (1,2)(2,3)(2,4)(3,4) \\
1 & \quad 1 & \quad 0 & \quad 1 & \quad 1 & \quad 1 \\
1 & \quad 1 & \quad 0 & \quad 2 & \quad 2 & \quad 1
\end{align*}
\]
Interval Graph Isomorphism

**Theorem. [Lueker and Booth, 1979]** A labeled $PQ$-tree contains enough information to reconstruct an interval graph up to isomorphism.

**Definition.** Two labeled $PQ$-trees are **identical** if they are isomorphic as rooted ordered trees and corresponding nodes have identical labels.

**Definition.** Two labeled $PQ$-trees are **equivalent** if one can be made identical to the other by a sequence of equivalence transformations, provided labels of $Q$-nodes whose children are reversed are appropriately modified as follows, for a $Q$-node $x$ with $k$ children:

- Replace each pair $(i, j)$ in `label[x]` by the pair $(k + 1 - j, k + 1 - i)$.
- Resort the pairs into lexicographically nondecreasing order.

**Theorem. [Lueker and Booth, 1979]** Two interval graphs $G_1$ and $G_2$ are isomorphic if and only if $T_L(G_1) \equiv T_L(G_2)$.

**Remark.** Equivalence of labeled $PQ$-trees can be tested using a modification of the Aho-Hopcroft-Ullman tree isomorphism algorithm.
Interval Graph Isomorphism

**Definition.** An interval graph is called proper if it has an interval representation such that no interval is properly contained in another interval.

**Remark.** Isomorphism of proper interval graphs can be tested by just computing and comparing canonical labels for the PQ-trees corresponding to adjacency matrices augmented by adding ones along the main diagonal, without need to find any maximal cliques.