CS762: Graph-Theoretic Algorithms
Lecture 9: Towards a Test For Interval Graphs
January 25, 2002

Scribe: John-Paul Pretti

Abstract

In this lecture we complete a proof of equivalency between three characterizations of interval graphs. This lays the groundwork for a recognition algorithm for interval graphs. We study how to find the maximal cliques of a chordal graph, G and show how these cliques can be ordered if G is an interval graph. Finding such an order can be reduced to the consecutive-ones-property for rows of a vertex/clique incidence matrix. PQ-trees are introduced as the fundamental data structure used to test if a matrix satisfies this property.

1 Introduction

Recall that an *interval graph* is defined by a set of intervals in $\mathbb{R}$. Each vertex corresponds to an interval. Two vertices are joined by an edge if and only if the corresponding intervals intersect. We have seen several applications of interval graphs. A number of problems which are hard in general can be solved efficiently for interval graphs. This suggests that an efficient recognition algorithm for interval graphs would be of both considerable practical and theoretical use. This lecture makes progress towards this goal.

In Section 2 we move towards such a test by completing a proof of equivalency between three characterizations of interval graphs. The second characterization of interval graphs is phrased, in part, in terms of G being a co-comparability graph. Recall that a *comparability graph* is a graph with edges that can be oriented such that the resulting directed graph is transitive and acyclic. Graph G is a *co-comparability graph* if $\overline{G}$ is a comparability graph.

The last of these characterizations defines interval graphs to be those for which the maximal cliques can be ordered such that they are consecutive for all vertices. The bulk of Section 2 is devoted to completing a the proof that these definitions are indeed equivalent.

We can determine if a 'good' order exists by first finding the maximal cliques of a graph. This can be done efficiently for chordal graphs. A *chordal graph* is a graph that does not contain an induced k-cycle for $k \geq 4$. This relates to the problem at hand because all interval graphs are chordal graphs and the first step of the algorithm will work for any chordal graph. This involves finding the maximal cliques of the input graph. We discuss how this can be done efficiently in Section 3.1.

In section 3.2 and and 3.3, we analyze the last step of the test for interval graphs and relate this problem to matrices and the consecutive-ones-property for rows.

The lecture ends with an introduction to PQ-trees. A PQ-tree is the data structure which is fundamental to the problem of finding a 'good' order of the maximal cliques of a graph.

2 Characterizations of interval graphs

Each of the three following statements are equivalent [GH64].

(i) G is an interval graph.

(ii) G is a co-comparability graph that does not have an induced 4-cycle.

(iii) The maximal cliques of G have an order $C_1, C_2, \ldots, C_m$ such that for all vertices v, $\{i \in \{1, \ldots, m\} : v \in C_i\}$ is consecutive.

Last lecture we proved that $(iii) \Rightarrow (i)$ and $(i) \Rightarrow (ii)$. Below, we restate and prove that $(ii) \Rightarrow (iii)$

Theorem 1 *If G is a graph such that $\overline{G}$ is a comparability graph and G does not contain an induced 4-cycle, then the maximal cliques of G can be ordered $C_1, C_2, \ldots, C_m$ such that for all vertices v, $\{i \in \{1, \ldots, m\} : v \in C_i\}$ is consecutive.*

Intuitively, the latter condition means that if G is an interval graph then we can arrange the maximal cliques of G such that if a vertex is in two different cliques C_i and C_j, then it is in all the cliques inbetween C_i and C_j according to this arrangement.

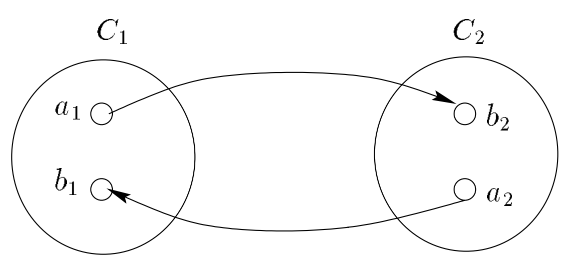
Proof: Let G be a graph such that $\overline{G}$ is a comparability graph and G does not contain an induced 4-cycle. Since $\overline{G}$ is a comparability graph, assume that the edges of $\overline{G}$ are directed in such a way that $\overline{G}$ is both transitive and acyclic.

Define a directed graph H where $V(H)$ is the set of maximal cliques of G and $C \to C' \in E(H)$ if and only $v \to w \in E(\overline{G})$ for some $v \in C$ and $w \in C'$.

We now state and prove four claims which together prove the theorem.

Claim 1 *H does not contain a 2-cycle.*

Proof: By way of contradiction, assume that $C_1 \to C_2$, $C_2 \to C_1 \in E(H)$. Then there exists a_1, $b_1 \in C_1$ and a_2, $b_2 \in C_2$ such that $a_1 \to b_2$, $a_2 \to b_1 \in E(\overline{G})$.



There are three cases.

case (i) $a_1 = b_1$
In this case, $a_2 \to b_2 \in E(\overline{G})$ by transitivity. This contradicts the fact that C_2 is a clique and $(a_2, b_2) \in G$.

case(ii) $a_2 = b_2$
In this case, $a_1 \to b_1 \in E(\overline{G})$ by transitivity. This contradicts the fact that C_1 is a clique and $(a_1, b_1) \in G$.

case(iii) $a_1 \neq b_1$ and $a_2 \neq b_2$

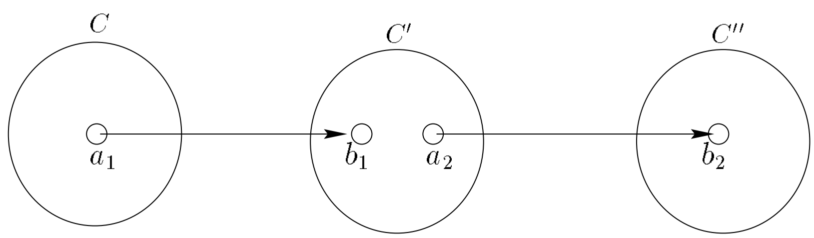
Note that $b_1 \rightarrow b_2 \notin E(\overline{G})$ because the transitivity of $\overline{G}$ would then imply that $a_2 \rightarrow b_2 \in E(\overline{G})$. This is impossible since C_2 is a clique and $(a_2, b_2) \in E(G)$. In a similar way, $(a_1, b_1) \in E(G)$ implies that $b_2 \rightarrow b_1 \notin E(\overline{G})$. Together, this means that $(b_1, b_2) \in E(G)$. Similarly, $(a_1, a_2) \in E(G)$. Therefore (a_1, a_2, b_2, b_1) is an induced 4-cycle in G. This is a contradiction.

All three cases lead to a contradiction which completes the proof of the claim.

□

Claim 2 *H is transitive.*

Proof: Let $C \rightarrow C', C' \rightarrow C'' \in E(H)$. $C \rightarrow C' \in E(H)$ implies that there exists $a_1 \rightarrow b_1 \in E(\overline{G})$ such that $a_1 \in C$ and $b_1 \in C'$. $C' \rightarrow C'' \in E(H)$ implies that there exists $a_2 \rightarrow b_2 \in E(\overline{G})$ such that $a_2 \in C'$ and $b_2 \in C''$. We must show that $C \rightarrow C'' \in E(H)$. We will do this by showing that $a_1 \rightarrow b_2 \in E(\overline{G})$.



There are six cases.

case (i) $a_2 = b_1$

In this case, $a_1 \rightarrow b_2 \in E(\overline{G})$ by transitivity.

case (ii) $a_1 \rightarrow a_2 \in E(\overline{G})$

Also in this case, $a_1 \rightarrow b_2 \in E(\overline{G})$ by transitivity.

case (iii) $b_1 \rightarrow b_2 \in E(\overline{G})$

Again, $a_1 \rightarrow b_2 \in E(\overline{G})$ by transitivity.

case (iv) $a_2 \rightarrow a_1 \in E(\overline{G})$

This cannot happen because it implies $C' \rightarrow C \in E(H)$ which contradicts Claim 1.

case (v) $b_2 \rightarrow b_1 \in E(\overline{G})$

This cannot happen because it implies $C'' \rightarrow C' \in E(H)$ which contradicts Claim 1.

case (vi) $a_1 \rightarrow a_2, a_2 \rightarrow a_1, b_1 \rightarrow b_2, b_2 \rightarrow b_1 \notin E(\overline{G})$ and $a_2 \neq b_1$.

This implies that $(a_1, a_2) \in E(G)$ and $(b_1, b_2) \in E(G)$. We also know that $(a_2, b_1) \in E(G)$ because C' is a clique. By construction, $(a_1, b_1), (a_2, b_2) \notin E(G)$. So $(a_1, b_2) \notin E(G)$ since G does not have an induced 4-cycle. In particular, G does not have the cycle (a_1, a_2, b_1, b_2). Thus either $a_1 \rightarrow b_2 \in E(\overline{G})$ or $b_2 \rightarrow a_1 \in E(\overline{G})$. The latter cannot be true because $\overline{G}$ is a comparability graph and $b_2 \rightarrow b_1 \notin E(\overline{G})$ in this case. Therefore, $a_1 \rightarrow b_2 \in E(\overline{G})$.

Each of the cases either cannot happen or imply that $a_1 \to b_2 \in E(\overline{G})$ which completes the proof of the claim.

$\square$

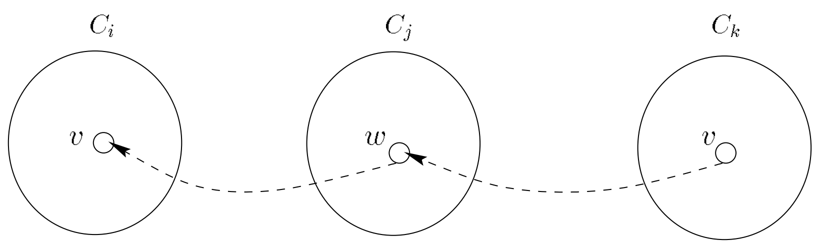
Claim 3 *H has a topological ordering.*

Proof: H cannot have a 2-cycle by Claim 1. If H had a k-cycle for $k > 2$, then by Claim 2, it would also have a 2-cycle. That is, the fact that H is acyclic follows directly from Claim 1 and Claim 2. Therefore H has a topological ordering.

$\square$

Claim 4 *A topological ordering on H induces an ordering on the maximal cliques of G meeting the conclusion of the theorem.*

Proof: Since $V(H)$ is the set of maximal cliques of G, we know that there is an order of the cliques $C_1, C_2, \ldots, C_m$. In order to show that for all vertices v, $\{i \in \{1, \ldots, m\} : v \in C_i\}$ is consecutive, we will assume by way of contradiction that there are cliques C_i, C_j, C_k with $1 \leq i < j < k \leq m$ such that there exists $v \in V(G)$ with $v \in C_i$, C_k but $v \notin C_j$.



Recall that C_j is a maximal clique. This means that $C_j \cup \{v\}$ is not a clique. That is, there exists $w \in C_j$ with $(v, w) \notin E(G)$. Thus, either $v \to w \in E(\overline{G})$ or $w \to v \in E(\overline{G})$. The existence of either of these edges contradicts the fact that $C_1, C_2, \ldots, C_m$ is a topological ordering.

$\square$

This completes the proof of the last of four claims which together prove the theorem. $\square$

3 Testing for interval graphs

The key practical benefit of the above result is that it leads us towards a test for interval graphs. The gist of this algorithm is

```
INPUT: a graph G
OUTPUT: yes, G is an interval graph; or no, G is not an interval graph

1. Find all the maximal cliques of G.
2. Try to order the maximal cliques of G such that the set of cliques
   containing any given vertex of G are consecutive.
```

3.1 Finding maximal cliques

In general, the problem of finding all maximal cliques is not even in NP. There exist graphs with an exponential number of maximal cliques. Luckily, interval graphs do not have more than a linear number of maximal cliques. We will show that this is true more generally. Namely, we will look at the class of chordal graphs of which we know interval graphs are a subclass.

Theorem 2 *There are at most n maximal cliques in a chordal graph on n vertices [FG65].*

Proof: Let G be a chordal graph. Let σ be a perfect elimination order of G. Recall that $\{v\} \cup Pred(v)$ is a clique for all vertices v. There are n such cliques. We will show that every maximal clique is of this form.

Let M be a maximal clique of G. Clearly $M \subseteq \{v\} \cup Pred(v)$ where v is the last vertex in M according to the perfect elimination order. Since M is maximal, it must be that $M = \{v\} \cup Pred(v)$. □

Remark 1 *Is the number of maximal cliques in a comparability graph linear, quadratic or exponential? This was left as an exercise.*

Remark 2 *A stronger statement than that cited above can actually be made. $\{v_i\} \cup Pred(v_i)$ is a maximal clique if and only if for every successor v_j of v_i, at least one predecessor of v_i is not a predecessor of v_j. The proof in both directions is trivial. This implies that we can then use a technique similar to that used to find a perfect elimination order discussed in a past lecture. The runtime of this algorithm is $O(n+m)$. See [Gol80] for the details. This shows that Step 1 of the above algorithm can be done in linear time.*

Remark 3 *One should clearly avoid trying to find the maximal cliques unless we know the graph is a chordal graph. Thus Step (1) of the algorithm should be performed only after executing one of our many tests for chordal graphs.*

3.2 Finding an order on the maximal cliques

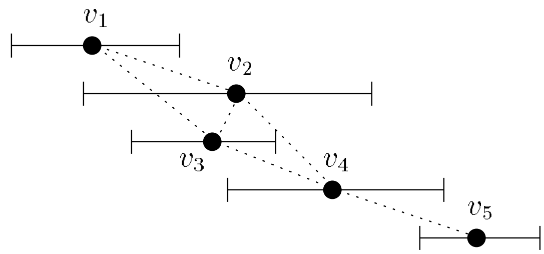
The problem of finding a 'good' order of the maximal cliques can be reduced to a matrix problem known as the consecutive-ones-property for rows.

3.3 Consecutive-ones-property

Let G be a graph. A matrix A is a *vertex/clique incidence matrix* for G if its rows are indexed by the vertices of G, its columns are indexed by the maximal cliques of G and

$$A_{ij} = \begin{cases} 1 & \text{if } v_i \in C_j \\ 0 & \text{otherwise.} \end{cases}$$

Finding a 'good' order of the maximal cliques of G amounts to finding a permutation of the columns of the vertex/clique incidence matrix for G such that all the ones in a given row of the matrix are consecutive. A matrix is said to have the *consecutive-ones-property for rows* if such a permutation can be found.



3.3.1 Example

Consider the graph G below.

The maximal cliques of G are $C_1 = \{v_1, v_2, v_3\}$, $C_2 = \{v_2, v_3, v_4\}$ and $C_3 = \{v_4, v_5\}$. The vertex/clique incidence matrix is shown below.

	C_1	C_2	C_3
v_1	1	0	0
v_2	1	1	0
v_3	1	1	0
v_4	0	1	1
v_5	0	0	1

The ones are consecutive in each row and C_1, C_2, C_3 is indeed a 'good' order of the maximal cliques.

3.4 PQ-trees

The problem of determining whether or not a matrix has the consecutive-ones-property for rows can be rephrased as:

Given: a finite set X and a collection $\mathcal{I}$ of subsets of X.

Want: a permutation $\Pi(X)$ such that $\forall I \in \mathcal{I}$, the elements of I are consecutive in Π.

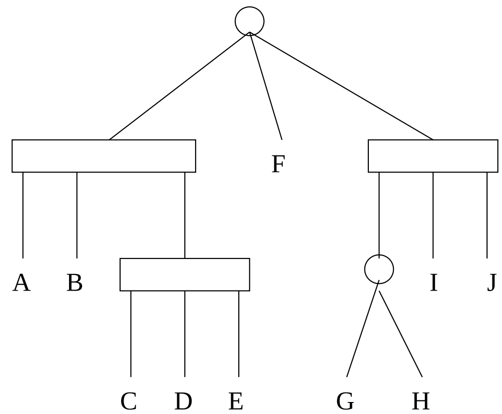
X corresponds to the columns of the matrix. Each row i corresponds to an $I_i \in \mathcal{I}$ where $j \in I_i$ if entry (i, j) is a one. This relates to the problem of finding a 'good' order of the maximal cliques of G by thinking of X as the set of maximal cliques and each $I \in \mathcal{I}$ as the set of cliques containing a given common vertex [FG65].

In 1976, Booth and Loeker devised a *PQ-tree* data structure which helps solve this very problem [BL76].

Reading the leaves of a PQ-tree from left to right corresponds to a permutation $\Pi(X)$. The nodes of a PQ-tree are of two types. A *Q-node*, q asserts that the children of q may be read in the order shown or the reverse of the order shown. Q-nodes are denoted by a rectangle. *P-nodes* assert that any permutation of their children are allowed. A *P-node* is depicted using a circle.

For example, permutations allowed by the above include

- $ABCDEFGHIJ$
- $ABEDCFGHIJ$
- $EDCBAFGHIJ$ and
- $FCDEBAJIHG$.



3.5 The testing algorithm

Step (2) of the algorithm testing if G is an interval graph consists of beginning with a simple PQ-tree with exactly one node, a P-node, and one child for each of the columns of the vertex/clique incidence matrix of G (the maximal cliques of G.) The algorithm then proceeds to consider constraints corresponding to which vertices are in which cliques. In the next lecture, we look into the details of this algorithm and justify that the PQ-tree can be updated efficiently. With care, the algorithm can be made to run in $O(n+m)$ time.

References

[BL75] K.S. Booth and G.S. Lueker. Linear algorithms to recognize interval graphs and test for consecutive ones property. *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 255–265, 1975.

[BL76] K.S. Booth and G.S. Lueker. Testing for the consecutive-ones-property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

[FG65] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[GH64] P.C. Gilmore and A.J. Hoffman. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics*, 16:539–549, 1964.

[Gol80] M.C. Golumbic. *Algorithmic graph theory and perfect graphs.* Academic Press, New York, 1980.