Chordal Graph Isomorphism

Contents.

- Chordal graphs.
- Characterization of chordal graphs.
- Recognition of chordal graphs.
- Chordal graphs have at most n maximal cliques.
- Isomorphism completeness of chordal graphs.

Bibliography.

- M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.
- K. Simon. *Effiziente Algorithmen für perfekte Graphen*. Leitfäden und Monographien der Informatik. B. G. Teubner Stuttgart, 1992.
- R. Shamir. Advanced topics in graph algorithms. Technical report, Tel-Aviv University, 1994.

Chordal Graph Isomorphism

Definition. *An undirected graph is called* ***chordal*** *if every cycle of length greater than three posseses a chord, that is, an edge joining two nonconsecutive vertices of the cycle.*

- An undirected graph is chordal if it does not contain an induced subgraph isomorphic to C_n for $n > 3$.

Remark. *Being chordal is a hereditary property, inherited by all the induced subgraphs of a chordal graph.*

Chordal graphs are also called ***triangulated*** *and* ***perfect elimination*** *graphs.*

Theorem. *Interval graphs are chordal.*

- G. Hajós. Über eine Art von Graphen. *Intern. Math. Nachr.*, 11, 1957.

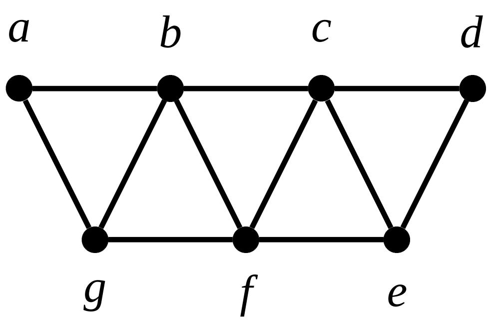
Proof. Let $G = (V,E)$ be an interval graph having a chordless cycle $[v_0, v_1, v_2, \dots, v_{\ell-1}, v_0]$ with $\ell > 3$, and let I_k denote the interval corresponding to vertex v_k. Choose a point $p_i \in I_{i-1} \cap I_i$, for $i = 1, 2, \dots, \ell - 1$. Since I_{i-1} and I_{i+1} do not overlap, the points p_i constitute a strictly increasing or strictly decreasing sequence. Therefore, it is impossible for I_0 and $I_{\ell-1}$ to intersect, contradicting the assumption that $\{v_0, v_{\ell-1}\} \in E$. $\square$

Chordal Graph Isomorphism

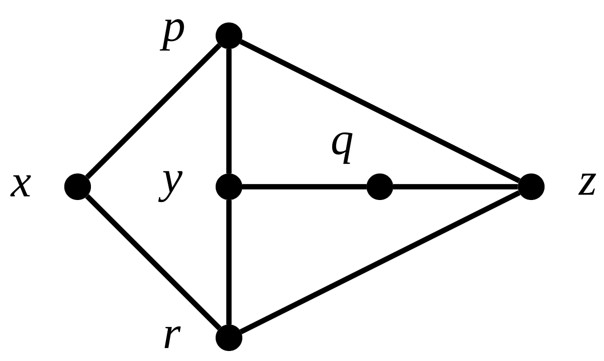
Definition. *A vertex is called **simplicial** if its adjacency set induces a complete subgraph, that is, a clique (not necessarily maximal).*

Definition. *A permutation $\sigma = [v_1, v_2, \ldots, v_n]$ of the vertices of an undirected graph G, or a bijection $\sigma : V \to \{1, \ldots, n\}$, is called a **perfect elimination order** if each v_i is a simplicial vertex of the subgraph of G induced by $\{v_i, \ldots, v_n\}$.*

Example. *The following chordal graph has 96 different perfect elimination orders, one of which is $\sigma = [a, g, b, f, c, e, d]$.*



The following undirected graph is not chordal. It has no simplicial vertex.



Chordal Graph Isomorphism

Theorem. *An undirected graph is chordal if and only if it has a perfect elimination order. Moreover, in such a case, any simplicial vertex can start a perfect elimination order.*

- D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15(3):835–855, 1965.

Lemma. *Every chordal graph has a simplicial vertex and, if it is not a clique, then it has two nonadjacent simplicial vertices.*

- G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25(1–2):71–76, 1961.

Corollary. *Chordal graphs can be recognized by the following iterative procedure: repeatedly locate a simplicial vertex and eliminate it from the graph, until no vertices remain (and the graph is chordal) or at some stage no simplicial vertex exists (and the graph is not chordal).*

Remark. *A naïve implementation of the previous procedure takes* $O(n^4)$ *time. As a matter of fact, testing whether a given vertex is simplicial takes* $O(n^2)$ *time, finding a simplicial vertex requires eventually testing all remaining vertices and thus takes total* $O(n^3)$ *time, and such a step must be repeated* $O(n)$ *times when the graph is chordal.*

Chordal Graph Isomorphism

Chordal graphs can be recognized in linear time, though.

Since a chordal graph G which is not a clique has two nonadjacent simplicial vertices, and every induced subgraph of a chordal graph is chordal, it is possible to choose **any** vertex v of G and decide it will be last in the perfect elimination order, then choose any vertex w adjacent to v and put it in position $n-1$, and so on, choosing vertices **backward** to the perfect elimination order.

Several linear-time algorithms for recognizing chordal graphs are based on this idea.

- D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.

> **Lexicographic search:** Number the vertices from n to 1 in decreasing order. For each unnumbered vertex v, maintain a list of the numbers of the numbered vertices adjacent to v, with the numbers in each list arranged in decreasing order. As the next vertex to number, select the vertex whose list is lexicographically greatest, breaking ties arbitrarily.

Chordal Graph Isomorphism

- R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13(3):566–579, 1984.

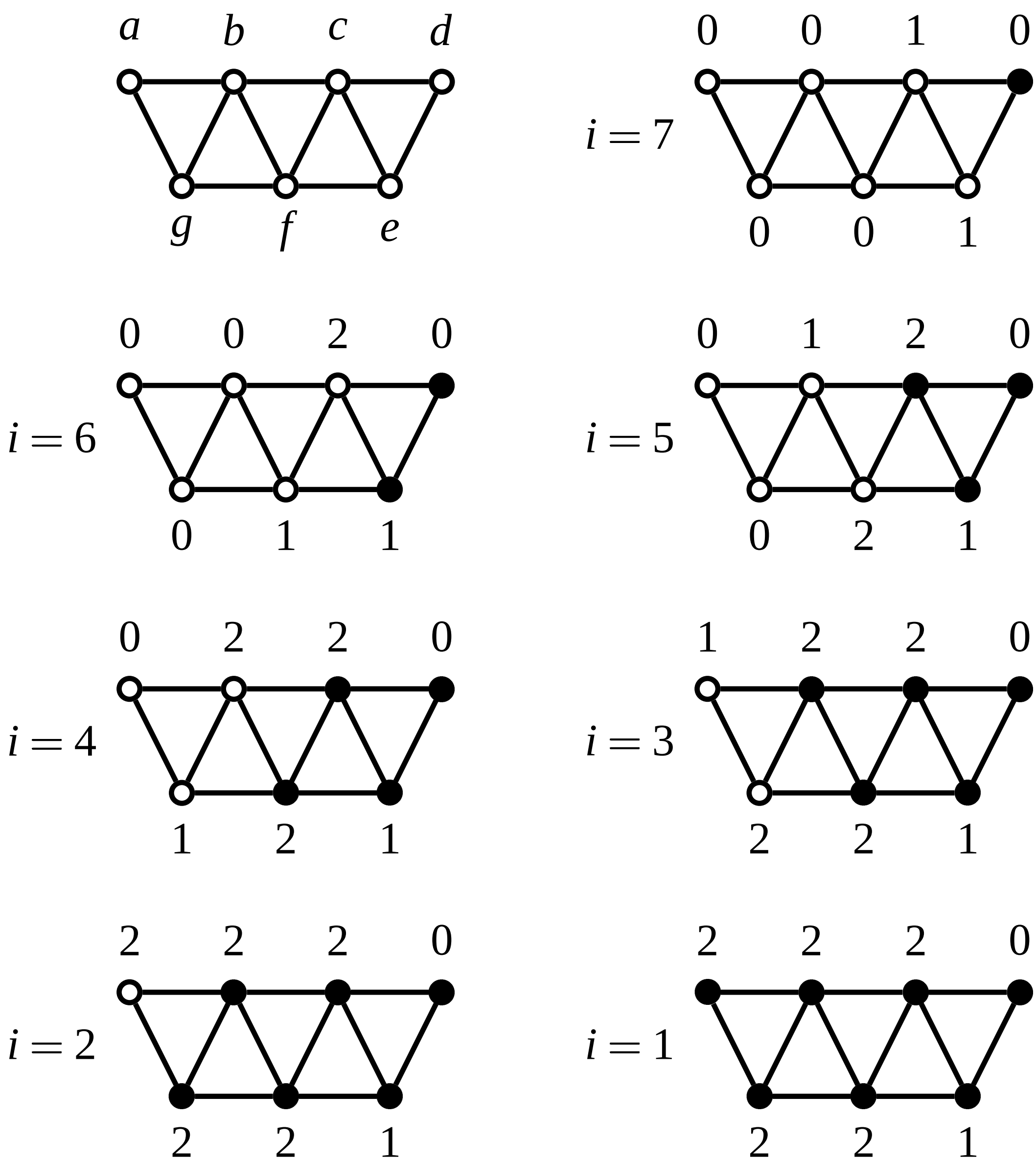
> **Maximum cardinality search:** Number the vertices from n to 1 in decreasing order. As the next vertex to number, select the vertex adjacent to the largest number of previously numbered vertices, breaking ties arbitrarily.

The following algorithm performs a maximum cardinality search on an undirected graph $G = (V, E)$, computing also a perfect elimination order σ when G is chordal.

procedure *maximum cardinality search* (G, σ)
    for all vertices v of G **do**
        set $label[v]$ to zero
    end for
    for all i **from** n **downto** 1 **do**
        choose an unnumbered vertex v with largest label
        set $\sigma(v)$ to i $\{$number vertex $v\}$
        for all unnumbered vertices w adjacent to vertex v **do**
            increment $label[w]$ by one
        end for
    end for
end procedure

Chordal Graph Isomorphism

Example. *The following chordal graph has 96 different perfect elimination orders, one of which,* $\sigma = [a, g, b, f, c, e, d]$*, can be found by maximum cardinality search as follows.*



Chordal Graph Isomorphism

Theorem. *The previous algorithm can be implemented to perform maximum cardinality search in $O(n+m)$ time.*

Proof. Let vertices be labeled by the number of adjacent numbered vertices, and let S_i be the set of unnumbered vertices with label i. Represent each set S_i by a doubly linked list of vertices, to facilitate deletion, and maintain an array of sets of vertices S_i, for $0 \leqslant i \leqslant m-1$. Maintain also an index j to the nonempty set S_j with the largest label, and maintain for each vertex a pointer to the position of the vertex in the corresponding set. Maximum cardinality search proceeds by removing some vertex v from set S_j, numbering vertex v and, for all unnumbered vertices w adjacent to vertex v, moving vertex w from the set containing it, say S_i, to set S_{i+1}. Since this takes $O(1+\deg(v))$ time, maximum cardinality search takes $O(n+m)$ time. $\square$

Chordal Graph Isomorphism

Theorem. [Tarjan and Yannakakis, 1984] *An undirected graph G is chordal if and only if maximum cardinality search(G,σ) produces a perfect elimination order σ.*

Remark. *A naïve procedure for testing whether σ is a perfect elimination order, and thus G is chordal, consists of simulating the vertex elimination process and, for each vertex to be eliminated, testing whether its remaining neighbors form a clique. However, this procedure takes $O(nm)$ time.*

Let σ be the permutation of the vertices of G produced by *maximum cardinality search*(G,σ). The following algorithm tests whether σ is a perfect elimination order.

function *perfect elimination order*(G,σ)
    for all i **from** 1 **to** $n-1$ **do**
        set v to $\sigma^{-1}(i)$
        set $m[v]$ to $\sigma^{-1}(\min\{\sigma(w) \mid w \in adj(v), \sigma(w) > \sigma(v)\}$
        for all vertices w adjacent to vertex v **do**
            if $\sigma(m(v)) < \sigma(w)$ **and** $w \notin adj(m(v))$ **then**
                return *false*
            end if
        end for
    end for
    return *true*
end function

Chordal Graph Isomorphism

Theorem. *The previous algorithm can be implemented to test a perfect elimination order in $O(n+m)$ time.*

Proof. The algorithm returns *false* during the $\sigma(u)$-th iteration if and only if there are vertices v, u, w with $\sigma(v) < \sigma(u) < \sigma(w)$, where $u = m(v)$ is defined during the $\sigma(v)$-th iteration, such that u and w are adjacent to v but u is not adjacent to w. In this case, σ is clearly not a perfect elimination order.

Conversely, suppose σ is not a perfect elimination order and the algorithm returns *true*. Let v be the vertex with the largest $\sigma(v)$ possible such that $W = \{w \in adj(v) \mid \sigma(w) > \sigma(v)\}$ is not complete. Let also $u = m(v)$ be the vertex of W defined during the $\sigma(v)$-th iteration. Since during the $\sigma(u)$-th iteration line 7 (return *false*) is not executed, every vertex $w \in W \setminus \{u\}$ is adjacent to vertex u, and every vertex pair $\{w, z\} \in W \setminus \{u\}$ is adjacent, because of the maximality of $\sigma(v)$, contradicting the assumption that W is not complete.

Regarding time complexity, lines 3–9 of the algorithm take $O(\deg(v))$ time for each vertex v. The algorithm takes thus $O(n+m)$ time. $\square$

Corollary. *Chordal graphs can be recognized in linear time.*

Chordal Graph Isomorphism

Let $X(v) = \{w \in adj(v) \mid \sigma(v) < \sigma(w)\}$.

Lemma. *Every maximal clique of a chordal graph $G = (V, E)$ is of the form $\{v\} \cup X(v)$, for some vertex $v \in V$.*

Proof. The set $\{v\} \cup X(v)$ is complete, because σ is a perfect elimination order. On the other hand, if w is the vertex of a maximal clique C with smallest σ number, then C is of the form $\{w\} \cup X(w)$. $\square$

Lemma. *A chordal graph has at most n maximal cliques.*

Remark. *Suppose $A = \{v\} \cup X(v)$ is not a maximal clique. Then, there is another clique $B = \{w\} \cup X(w)$ such that $A \subseteq B$ and, by the previous to last lemma, it must be $\sigma(w) < \sigma(v)$.*

Recall that $m(w)$ is defined by $\sigma(m(w)) = \min\{\sigma(z) \mid z \in X(w)\} = \min\{\sigma(z) \mid z \in adj(w), \sigma(w) < \sigma(z)\}$.

Among all cliques $B = \{w\} \cup X(w)$ with $A \subseteq B$, the one with largest $\sigma(w)$ implies $m(w) = v$ and, therefore, the clique $A = \{v\} \cup X(v)$ is not maximal if and only if $m(w) = v$ and $|X(v)| \leqslant |X(w)| - 1$ for all vertices $w \in V$.

Chordal Graph Isomorphism

Let σ be the perfect elimination order produced by *maximum cardinality search*(G,σ). The following algorithm enumerates all maximal cliques of G.

procedure *all maximal cliques*(G,σ)
  for all vertices v of G **do**
    set $size[v]$ to zero
  end for
  for all i **from** 1 **to** n **do**
    set v to $\sigma^{-1}(i)$
    if $\deg(v) = 0$ **then**
      output maximal clique $\{v\}$
    end if
    set X to $\{w \in adj(v) \mid \sigma(v) < \sigma(w)\}$
    if $X \neq \emptyset$ **then**
      if $size[v] < |X|$ **then**
        output maximal clique $\{v\} \cup X$
      end if
      set $m[v]$ to $\sigma^{-1}(\min\{\sigma(w) \mid w \in X\}$
      set $size[m[v]]$ to $\max\{size[m[v]], |X| - 1\}$
    end if
  end for
end procedure

Lemma. *The previous algorithm can be implemented to enumerate all maximal cliques in $O(n+m)$ time.*

Chordal Graph Isomorphism

Theorem. [Lueker and Booth, 1979] *Graph isomorphism is polynomially reducible to chordal graph isomorphism.*

Proof. A chordal graph $M(G) = G' = (V', E')$ can be associated to an arbitrary graph $G = (V, E)$ as follows. Let $V' = V \cup E$, and let $E' = \{\{v, w\} \mid v, w \in V\} \cup \{\{v, e\} \mid v \in V, e \in E, v \textit{ is incident with } e\}$. The construction can be implemented to take $O(n+m)$ time.

Consider any cycle of length greater that three in G'. If the cycle contains only V-vertices, then it has a chord, since all V-vertices are adjacent. Otherwise, the cycle contains an E-vertex, and then the two vertices adjacent to this E-vertex must be V-vertices, and they are adjacent. Therefore, G' is chordal.

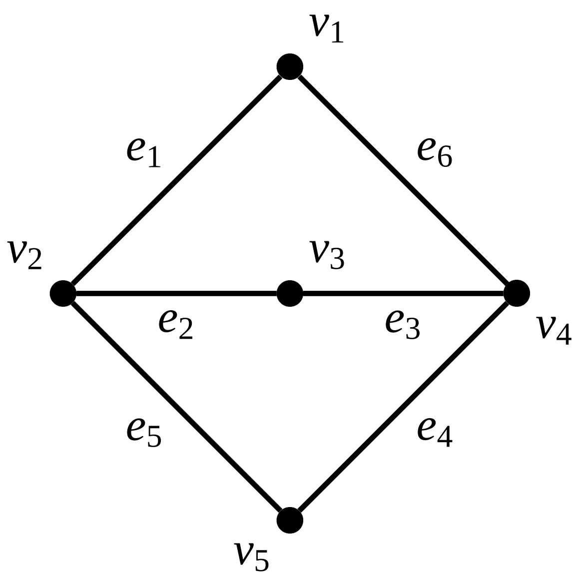
Assume now $n \geqslant 4$. It turns out that G' contains enough structure to allow to reconstruct G, up to isomorphism. As a matter of fact, since all V-vertices are adjacent, all have degree at least equal to $n-1$, which is greater than two, while E-vertices always have degree two, because an E-vertex is adjacent to exactly two V-vertices. Furthermore, two vertices of G are adjacent if the corresponding V-vertices are adjacent to a common E-vertex.

The problem of testing isomorphism of G_1 and G_2 is then polynomially reduced to the problem of testing isomorphism of $M(G_1)$ and $M(G_2)$. $\square$

Chordal Graph Isomorphism

Example. *Construction underlying the reduction of graph isomorphism to chordal graph isomorphism.*

G



$G' = M(G)$