October 15, 2015

Math 3012 – Applied Combinatorics Lecture 16

William T. Trotter
trotter@math.gatech.edu

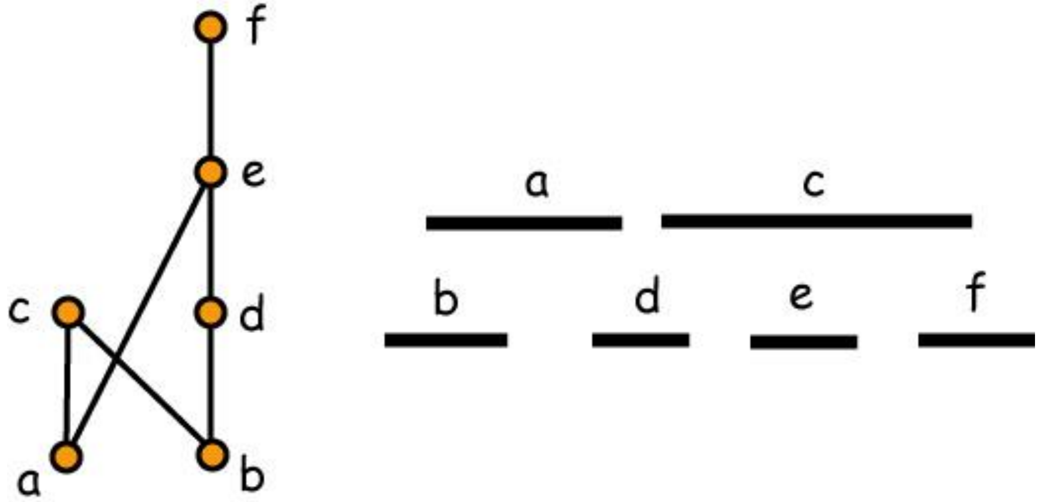
Reminder

Homework 2 Tuesday, October 20, 2015 (Even problems only).

Test 2 Thursday, October 22, 2015. Details on material for which you will be responsible sent by email before this coming weekend. Study hard. Experience shows that the middle portion of this course has considerably more depth and significance than the first part.

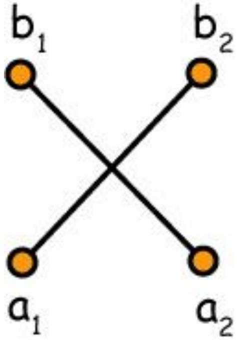
Interval Orders

A poset P is an interval order if there exists a function I assigning to each x in P a closed interval $I(x) = [a_x, b_x]$ of the real line $\mathbf{R}$ so that $x < y$ in P if and only if $b_x < a_y$ in $\mathbf{R}$.



Characterizing Interval Orders

Theorem (Fishburn, 1970) *A* poset is an interval order if and only if it does not contain the standard example S_2.



$$S_2 = 2 + 2$$

Proof of Fishburn's theorem

Proof The condition that P not contain **2 + 2** is obviously necessary. We prove sufficiency by induction on $|P|$. Obviously true when $|P| = 1$. Assume true when $|P| = k$ where $k \geq 1$. The consider the case where $|P| = k + 1$.

Choose maximal element x with $D(x) = \{ y : y < x \text{ in } P\}$ as large as possible. Then $I(x)$ is either empty or it is an antichain. In either case, it is clear that we can alter an interval representation of $P - \{x\}$ to add an interval for x.

Interval Order Algorithm

Step 1 For each x in P, compute the down set $D(x) = \{y : y < x \text{ in } P\}$. If there are two down sets not comparable by inclusion, then we find a **2 + 2**. If they are all comparable, then we know P is an interval order.

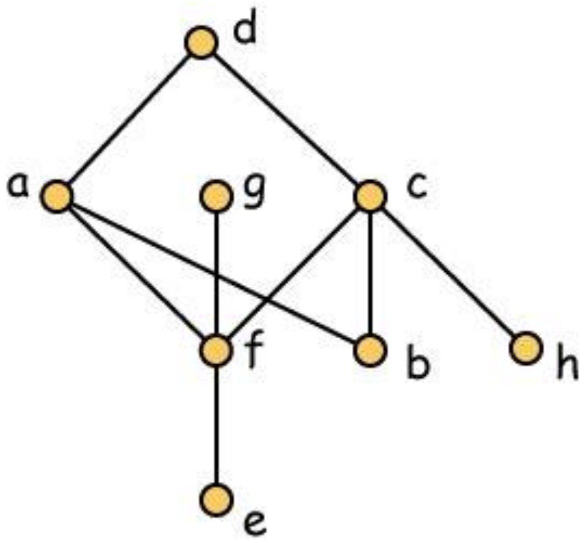
Step 2 Label the down sets from small to large as 1, 2, 3, …, m.

Step 3 Repeat Step 1 for up sets $U(x) = \{y : y > x \text{ in } P\}$. Label them from large to small as 1, 2, …, m.

Step 4 Assign x to interval $[i, j]$ when $D(x)$ gets label i and $U(x)$ gets label j.

Applying the Algorithm

Exercise Test the poset P shown below to see if it's an interval order. If no, find a **2 + 2**. If yes, find an interval representation.



Computational Details

Note The algorithm just described provides a representation of an interval order using the least number of end points. However, it always includes some degenerate intervals. If desired, the representation can be easily modified so that all intervals are non-degenerate. In fact, if needed, it is also easy to make all end points distinct.

Recognizing Interval Graphs

Step 1 Given a graph G, first let H be the complement of G. Then test H to see if it is a comparability graph, i.e., test whether H can be transitively oriented. If no, then G is not an interval graph. If yes, then G might be an interval graph.

Step 2 Let P the poset associated with a transitive orientation of H. Test P to see if it's an interval order. If no, then G is not an interval graph.

If yes, then G is an interval graph and the interval representation just found for P is also an interval representation of G.

The Dilworth Problem for Interval Orders

Observation If a poset P is an interval order, then the algorithm we have just learned finds an interval representation of P. These same intervals are the interval representation of the incomparability graph G of P. If we use the First Fit algorithm to color G (proceeding in the order of left end points), then we solve the Dilworth problem for P, i. e., we find the width of P and a minimum chain partition of P.

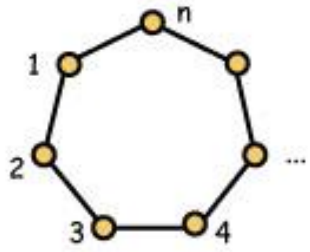
Characterizing Interval Graphs

Observation There is a minimum list $\mathbf{I}$ of forbidden graphs so that a graph G is an interval graph if and only if it does not contain, as an induced subgraph any of the graphs in the list.

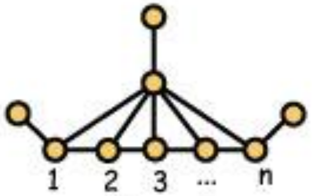
Remark It follows easily that this minimum list of forbidden subgraphs consists of the complements of those graphs in Gallai's list $\boldsymbol{C}$ which do not contain an induced cycle C_4 on 4 vertices.

Forbidden Subgraphs for Interval Graphs

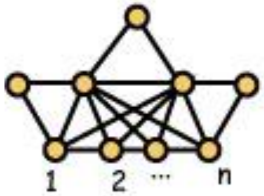
Theorem (Lekkerkerker and Boland, 1962) A graph G is an interval graph if and only if it does not contain any of the graphs shown below as an induced subgraph.



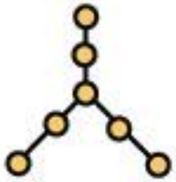
C_n; $n \geq 4$



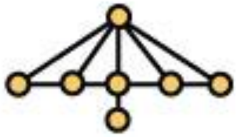
K_n; $n \geq 2$



L_n; $n \geq 1$



B_1



B_2