Advanced Graph Algorithms

Conf. dr. ing. Guillaume Ducoffe

guillaume.ducoffe@fmi.unibuc.ro

Hereditary graph classes

Definition

A class of graphs $\mathcal{G}$ is called hereditary if every induced subgraph of a graph in $\mathcal{G}$ also belongs to the class.

Example:

- Forests
- Bipartite graphs

Property: every hereditary graph class can be characterized as the $\mathcal{H}$-free graphs (a.k.a., all graphs excluding every graph in $\mathcal{H}$ as an induced subgraph), for some possibly infinite family $\mathcal{H}$.

Hereditary graph classes

Definition

A class of graphs $\mathcal{G}$ is called hereditary if every induced subgraph of a graph in $\mathcal{G}$ also belongs to the class.

Example:

- Forests – *Cycle-free* –
- Bipartite graphs – *Odd-cycle-free* –

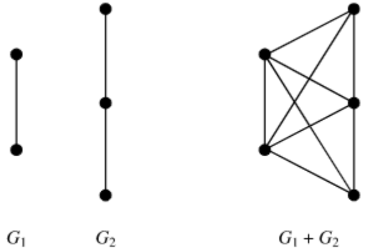
Property: every hereditary graph class can be characterized as the $\mathcal{H}$-free graphs (a.k.a., all graphs excluding every graph in $\mathcal{H}$ as an induced subgraph), for some possibly infinite family $\mathcal{H}$.

P_k-free graphs

- P_1-free graphs: the empty graph...
- P_2-free graphs: edgeless graphs...
- P_3-free graphs: cluster graphs (every connected component must be a clique)
- P_4-free graphs: **cographs** (complement-reducible graphs)

Easy properties of cographs

- If G is connected, then $diam(G) \leq 2$.
- The complement $\overline{G}$ of a cograph G is also a cograph.
- The disjoint union of two cographs G_1, G_2 is also a cograph.
- The **join** of two cographs is also a cograph.



Connectivity

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

Proof by contradiction. Suppose both $G, \overline{G}$ are connected.

Let $v \in V(G)$ be an arbitrary vertex. We partition the vertex set $V(G)$ in $\{v\},\ A = N(v),\ B = V(G) \setminus N[v]$.

Connectivity

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

Proof by contradiction. Suppose both $G, \overline{G}$ are connected.

Let $v \in V(G)$ be an arbitrary vertex. We partition the vertex set $V(G)$ in $\{v\},\ A = N(v),\ B = V(G) \setminus N[v]$.

(1) Both A, B are nonempty.

Connectivity

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

Proof by contradiction. Suppose both $G, \overline{G}$ are connected.

Let $v \in V(G)$ be an arbitrary vertex. We partition the vertex set $V(G)$ in $\{v\},\ A = N(v),\ B = V(G) \setminus N[v]$.

(1) Both A, B are nonempty.

(2) If $u, w \in B$ are adjacent, then $N(u) \cap A = N(w) \cap A$.

Connectivity

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

Proof by contradiction. Suppose both $G, \overline{G}$ are connected.

Let $v \in V(G)$ be an arbitrary vertex. We partition the vertex set $V(G)$ in $\{v\},\ A = N(v),\ B = V(G) \setminus N[v]$.

(1) Both A, B are nonempty.

(2) If $u, w \in B$ are adjacent, then $N(u) \cap A = N(w) \cap A$.

(3) If $u, w \in B$, then $N(u) \cap A, N(w) \cap A$ are comparable for inclusion.

Connectivity

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

Proof by contradiction. Suppose both $G, \overline{G}$ are connected.

Let $v \in V(G)$ be an arbitrary vertex. We partition the vertex set $V(G)$ in $\{v\}$, $A = N(v)$, $B = V(G) \setminus N[v]$.

(1) Both A, B are nonempty.

(2) If $u, w \in B$ are adjacent, then $N(u) \cap A = N(w) \cap A$.

(3) If $u, w \in B$, then $N(u) \cap A, N(w) \cap A$ are comparable for inclusion.

(4) If $x, y \in A$, then $N(x) \cap B, N(y) \cap B$ are comparable for inclusion. If x, y are nonadjacent, then $N(x) \cap B = N(y) \cap B$.

Connectivity cont'd

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

(5) There exists a $x \in A$ such that $B \subseteq N(x)$.

There exists a $u \in B$ such that $A \subseteq N(u)$.

Connectivity cont'd

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

(5) There exists a $x \in A$ such that $B \subseteq N(x)$.

There exists a $u \in B$ such that $A \subseteq N(u)$.

$\Longrightarrow$ Every vertex is adjacent to one of u, x.

Connectivity cont'd

Theorem

If G is a cograph, then either G or $\overline{G}$ is disconnected.

(5) There exists a $x \in A$ such that $B \subseteq N(x)$.

There exists a $u \in B$ such that $A \subseteq N(u)$.

$\Longrightarrow$ Every vertex is adjacent to one of u, x.

$\Longrightarrow$ $diam(\overline{G}) \geq 3$. A contradiction.

Decomposition theorem

Every cograph must be either:

- Reduced to one vertex
- The disjoint union of two cographs.
- The join of two cographs.

Remark: these three cases are mutually exclusive.

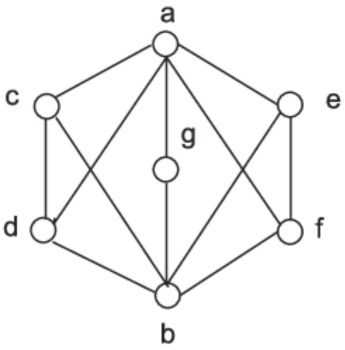
The cotree

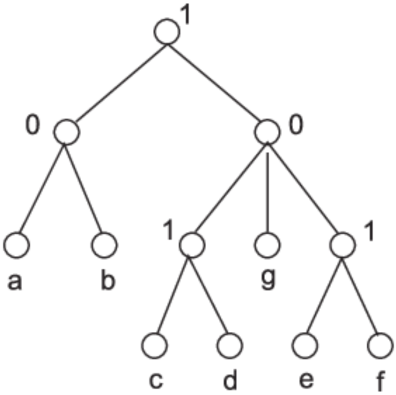
Every cograph G can be uniquely represented by a rooted tree, whose nodes represent subgraphs of G.

- The root represents G itself.
- The leaves represent the vertices of G.
- If a node represents a connected subgraph H, then it is labelled 1 and each child represents a different co-connected component of H.
- If a node represents a disconnected subgrah H, then it is labelled 0 and each child represents a different connected component of H.

Remark: The cotree fully determines the cograph.

Example





Consequence: many NP-hard problems can be solved in linear time on cographs (see the seminars. . .).

Pruning sequence

Two vertices u, v are **twins** if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

- They are false twins if u, v are nonadjacent;
- true twins otherwise.

Theorem

Every cograph contains a pair of twin vertices.

Proof: Take a deeper internal node x in the cotree. All its children are leaves. The corresponding vertices of graph G are true twins if x is labelled 1, and they are false twins if x is labelled 0.

$\rightarrow$ New characterization of cographs: there exists a vertex ordering $v_1, v_2, \ldots, v_n$ such that each v_j, $j < n$, has a twin in the subgraph induced by $v_j, v_{j+1}, \ldots, v_n$.

Recognition of cographs

1) If G is disconnected then:

- Compute the connected components $C_1, C_2, \dots, C_p$ of G
- Check whether $G[C_i]$ is a cograph for every $1 \leq i \leq p$.

2) Else, if $\overline{G}$ is disconnected, then:

- Compute the co-connected components $C'_1, C'_2, \dots, C'_q$ of G
- Check whether $G[C'_j]$ is a cograph for every $1 \leq j \leq q$.

3) Else, reject.

Complexity: $\mathcal{O}(nm)$. The algorithm also computes the cotree.

Incremental algorithm

1) If G is a cograph, then so must be $G - v$, for any v. We apply our algorithm recursively on $G - v$. Doing so, we compute a cotree (T', r').

2) If v is universal, then there are two cases.

- r' is labelled 1: we add one leaf labelled with v as a child of r'.
- r' is labelled 0: we add one new root r with label 1 and respective children r' and a new leaf with label v.

Proceed similarly if v is isolated.

3) For every $u \in N(v)$, mark all the nodes of T' between u and the root. This can be done in $\mathcal{O}(n)$ time.

4) Wlog the root has label 0. At least one child must be unmarked. If one child is marked, then we proceed recursively. Otherwise, each connected component must be either disjoint from $N(v)$ or fully contained in $N(v)$. Then, we correct the cotree by adding three more nodes.

Complexity: $\mathcal{O}(m + n^2) = \mathcal{O}(n^2)$.

Recursion depth

We proceed recursively if, for instance:

- the root r_0 is labelled 0, v is not isolated, and there is one marked child r_1;
- r_1 is labelled 1, v is not universal, and all its non-neighbours are descendants of one child r_2 ;
- r_2 is labelled 0, v is not isolated, and there is one marked child r_3;
- ...

Observation 1: all nodes r_i, except maybe the last one, are marked.

Observation 2: if r_i is labelled 1, then some neighbour u is on another branch than r_{i+1}.

$\Longrightarrow$ recursion depth in $\mathcal{O}(d(v))$.

Better analysis

Observation: The bottleneck is the marking process, that runs in $\mathcal{O}(n)$ time for every new vertex v to be inserted.

Let $H_1, H_2, \ldots, H_q$ be the marked children at the root. Note that $q = \mathcal{O}(d(v))$. Furthermore, for every i such that $1 \leq i \leq q$, the number of nodes in the subtree of T' rooted at H_i is in $\mathcal{O}(|V(H_i)|)$.

If G is a cograph and v is neither isolated nor universal, then either $q = 1$, $q = d_{T'}(r') - 1$, or we must have $V(H_i) \subseteq N(v)$ for every i such that $1 \leq i \leq q$.

Therefore, the marking process actually runs in $\mathcal{O}(d(v))$ time!

Theorem

We can recognize cographs, and construct a corresponding cotree, in $\mathcal{O}(n+m)$ time.

Questions