Lexicographic Cotree Factorization

Contents

1 Theoretical Concept

1.1 Trees

THE PROBLEM THAT C CANT BE A NORMAL SET SHOULD BE CLEARED
Trees are graphs (V, E) which have to be connected, acyclic and undirected. Rooted trees are just trees with one vertice specified as root. We may consider a inductive definition, detached from pure graph theory:
Let $\mathbb{T}_M$ be the set of all trees with labels in M:

$T = (r, \mathcal{C}) \in \mathbb{T}_M$ if:

- T is called leaf $\iff \mathcal{C} = \varnothing$, $r \in M$
- $\mathcal{C} \subseteq \mathbb{T}_M$, $|\mathcal{C}| \in \mathbb{N}$, $r \in M$

1.1.1 Definitions on Trees

Let $T = (r, \mathcal{C}) \in \mathbb{T}_M$ be a Tree.

- Nodes of T, named $\mathcal{N}(T) \subseteq \mathbb{T}_M$ are inductively defined by:
  - Every child $C = (r_C, \mathcal{C}_C) \in \mathcal{C}$ is called node of T
  - All nodes of the children $\mathcal{N}(C)$ are nodes of T as well

  $$\implies \mathcal{N}(T) = \{T\} \cup \left(\bigcup_{C \in \mathcal{C}} \mathcal{N}(C) \cup \{C\} \right)$$

  Nodes will usually denote the corresponding graph node, but with this notation the child nodes are acquired simultaneously.
  The experienced reader will verify, that this matches the later defined complete subtrees. Which should be synonymous from here on.

- Children of Node N of T: For $N = (r_N, \mathcal{C}_N)$
  - Children of N as defined the set $\mathcal{C}_N$

- Parent of Node N of T:
  - Parent of N, denoted by $\mathcal{P}(N) = P \in \mathcal{N}(T)$ with $N \in \mathcal{C}_P$.
    While $\mathcal{P}(T) = \textit{undef}$

- Leafes of T:
  - $\mathcal{L}(T) = \mathcal{N}(T) \cap \{(r, \varnothing) \mid r \in M\}$ are called leafs as above.

1.1.2 Level

The classical definition for a nodes level is by the distance of the node to the root of the tree. With the above definition we get: Let $T \in \mathbb{T}_M$ be a tree, N one of its nodes.

- the level of N will be:

$$l_T(N) = n \in \mathbb{N} : \qquad \text{with } \mathcal{P}^n(N) = \underbrace{\mathcal{P}(\mathcal{P}(\cdots \mathcal{P}}_{n \text{ times}}(N))) = T$$

1.1.3 Depth

The depth of some node N of T is like turning the definition of level upside down. With that the depth will be referencing to the shortest path to a leaf of the corresponding complete subtree. We will define the depth of node $N = (r_N, \mathcal{C}_N)$ by:

$$d(N) = max\left(\{d(C) + 1 \mid C \in \mathcal{C}_N\} \cup \{0\}\right)$$

From this definition it follows that: $d(N) = 0 \Leftrightarrow N$ is leaf.

The depth said to be unambiguous if

$$d(N) = d(C) + 1 \qquad \forall C \in \mathcal{C}_N$$

and the depth of every child of N is unambiguous.

1.1.4 Subtree

$S = (r_S, \mathcal{C}_S)$ is subtree of a given Tree $T = (r_T, \mathcal{C}_T)$, denoted by $S \subseteq T$, if there exists a node $N = (r_N, \mathcal{C}_N) \in \mathcal{N}(T)$ with

$$r_N = r_S \text{ and } \forall K \in \mathcal{C}_S \, \exists C \in \mathcal{C}_N : \quad K \subseteq C$$

$S = (r_S, \mathcal{C}_S)$ will be called complete subtree of $T = (r_T, \mathcal{C}_T)$, denoted by $S \subseteqq T$, if there exists a node $N = (r_N, \mathcal{C}_N) \in \mathcal{N}(T)$ with

$$r_N = r_S \text{ and } \forall K \in \mathcal{C}_S \, \exists C \in \mathcal{C}_N : \quad K \subseteqq C$$

1.2 Cotree

With the definition above cotrees will be denoted $\mathbb{T}_{\mathcal{B}}$
With $\mathcal{B} = \{0, 1\}$ where "B" stands for boolean. Every cotree represents a cograph, where every leaf represents a vertice of the cograph. Two of them are connected if their lowest common parent's label, that's the one with lowest depth, is 1.

1.2.1 Minimal Cotree

Let $T = (r, \mathcal{C})$ be a cotree. T said to be minimal if for every path in T the sequence of label is always an alternating one.

$$\text{cotree } T = (r, \mathcal{C}) \text{ is minimal } \iff \forall (r_C, \mathcal{C}_C) = C \in \mathcal{C} : \quad C \text{ is minimal and } r_C \neq r$$

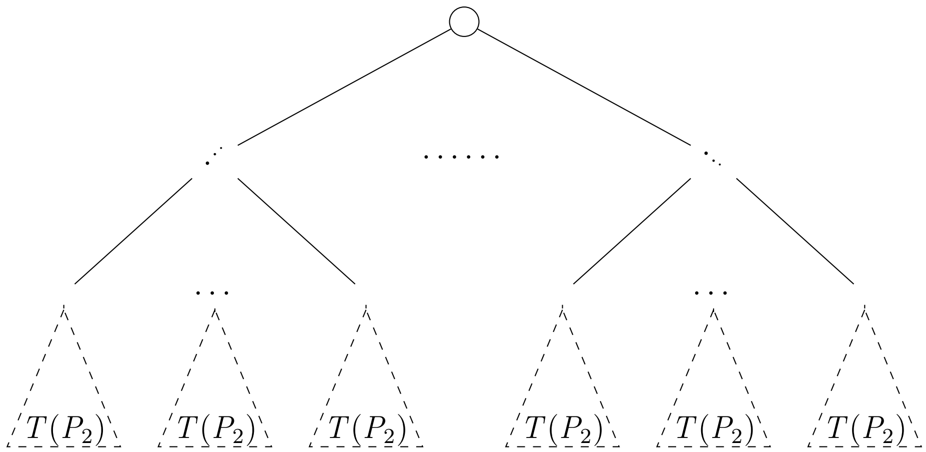
I will denote the minimal cotree of T with T_{min}

1.2.2 Cotree isomorphy

Two cotrees T, T' are called cotree isomorphic if their minimal cotrees $T_{min} = (r_T, \mathcal{C}_T)$ and $T'_{min} = (r_{T'}, \mathcal{C}_{T'})$ are rooted tree isomorphic.

1.2.3 Lexicographic Product

Denote the Cotree of a Cograph C as $T(C)$
Let C_1 be a Cograph. C_1 is Product of two other Cographs P_1,P_2 if it is cotree isomorphic to $T(P_1) \lhd T(P_2)$.
In other Words there exists a cotree, representing the identical cograph of the form:



Where if one would remove every subtree $T(P_2)$ one would get $T(P_1)$, as well as you would remove every former leaf from the tree.
$T(P_1) \lhd T(P_2)$ just means attach $T(P_2)$ where a leaf in $T(P_1)$ is.

Particularly a cotree F is Factor of some cotree T denoted by $T \lhd F$ if

$$\forall L \in \mathcal{L}(T_{min})\, \exists S \subseteq T_{min} : \quad L \in \mathcal{L}(S) \text{ and } S \cong F_{min} \qquad (1)$$

let those subtrees be $S_1, ..., S_n \quad n \in \mathbb{N}$

$$\forall i, j \in \{1, ..., n\} : \quad i \neq j \implies \mathcal{N}(S_i) \backslash \{S_i\} \cap \mathcal{N}(S_j) \backslash \{S_j\} = \varnothing$$

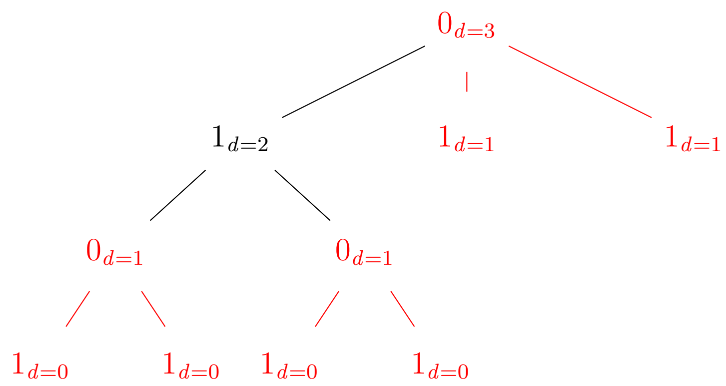
e.g. their nodes, except the root, are pairwise disjoint.
Because F_{min} and T_{min} are minimal we may refer in (1) to the classical rooted tree isomorphy.

From the definition above it schould be clear that $n \,\Big|\, |\mathcal{L}(T_{min})|$ from which follows that $|\mathcal{L}(F_{min})| \,\Big|\, |\mathcal{L}(T_{min})|$.

Another interesting fact arising, for a given cotree T, factor cotree F and minimal subtrees $S_1, ..., S_n \quad n \in \mathbb{N}$:

$$\forall N \in \mathcal{N}(T_{min})\, \exists i \in \{1, ..., n\} : \quad d(N) \leq d(F_{min}) \implies N \in \mathcal{N}(S_i)$$

With this counterexample it will be clear, why the opposite direction won't hold:



Our factor will be the minimal cotree for the cograph $(V, E) = (\{1, 2\}, \varnothing)$. The subtrees S_1, S_2, S_3 (read from left to right) are marked red. As you see the root of the whole tree is root of S_3 too, but with depth 3 it is indeed not smaller than $d(S_3)$.
The background of this happening is the ambiguous depth of this node, because without the left subtree, depth of this node would be 1 as well.
With this in our mind we should think about creating an unambiguous cotree from a normal one.

1.3 Balancing Cotrees

1.4 Labeling

A Labeling for Trees gives every node of the tree a equivalence class with an order between them. Two of them be synonymous if they have the same equivalence classes and there exists an order isomorphism between their equivalence classes.

1.5 Connection of depth and level

Depth and Level are two non synonymous methods for labeling nodes in trees.
But if the tree has unambiguous depth then the two labelings are indeed synonymous with inverse equivalence class ordering.

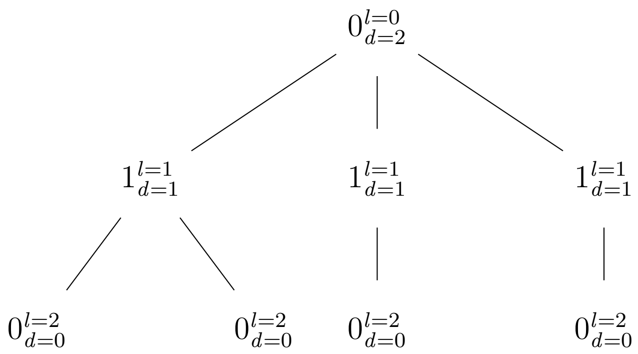
Proof by Induction: Let $T = (r, \mathcal{C})$ be a Tree with unambiguous depth and $d(T) = n$

- Base Case:
  For $n = 0$ $\quad T$ is a leaf $\implies$ both labelings are synonymous
  $$l_T^{(max)} - l_T(T) = 0 - 0 = 0 = d(T)$$
  with $l_T^{(max)}$ the number of levels of the tree T

- Induction Hypothesis:
  Let the labelings be synonymous for trees T with unambiguous depth of n. N node of T and with order isomorphism:
  $$l_T^{(max)} - l_T(N) = d(N)$$
  with $l_T^{(max)}$ the number of levels of the tree T

- Induction Step:
  Let $T = (r, \mathcal{C})$ be a tree of unambiguous depth
  $\implies$ all $C \in \mathcal{C}$ are of unambigous depth.
  For every node N of $C \in \mathcal{C}$:
  $$l_T(N) = l_C(N) + 1$$
  because every child of T is on level 1 and itself is unambiguous.
  With that let N be node of child $C \in \mathcal{C}$:
  $$\begin{aligned} l_T^{(max)} - l_T(N) = l_T^{(max)} - (l_C(N) + 1) &= l_C^{(max)} + 1 - (l_C(N) + 1) && (1) \\ &= l_C^{(max)} - l_C(N) \overset{IH}{=} d(N) && (2) \end{aligned}$$
  And for T: $d(T) = l_T^{(max)} - l_T(T) = l_T^{(max)}$
  with $l_T^{(max)}$ the number of levels of the tree T
  $\implies$ the two labelings are synonymous, because the order isomorphy is continued (see (1)). $\square$

Example:



1.6 Balancing

Because we may alter Cotrees in certain ways to leave them Cotree-Isomorphic to itself, e.g. it represents the same Cograph, we could use that to simplify our problem.
One way to do so is balancing the tree, that its depth will be unambiguos.
The Process starts inductively from the leafs, in other words from every node of the tree with depth equal to 0.

Induction for $d(T) = n$

- Base Case:
  $$n = 0 \implies \text{depth is unambiguous because there are no children}$$

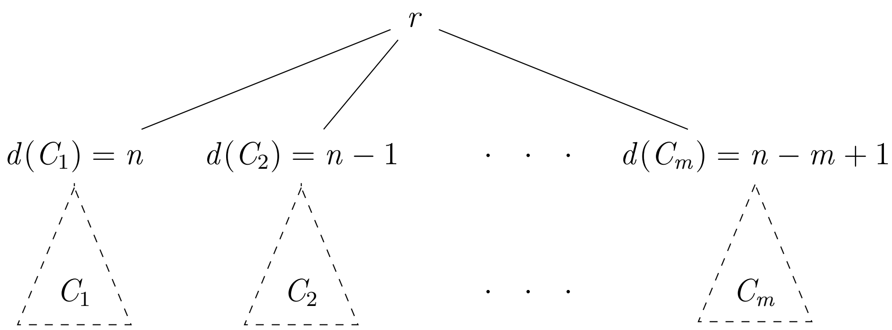
- Induction Hypothesis:
  For every child of T: $C_1, ..., C_m$ the depth is unambiguous.

- Induction Step:
  Let $T = (r, \mathcal{C})$ be a tree with $d(T) = n + 1$
  $\Longrightarrow \ \forall C \in \mathcal{C} : \quad d(C) \leq n$ as well as C's depth is unambiguous
  let $d_{min} = min\,(\{d(C) \mid C \in \mathcal{C}\})$
  Now disconnect every child and attach a chain of nodes each with the same label as r. The length of the chain said to be $n - d_{min}$.
  Let $N_1, ..., N_\delta$ be its nodes, counted from bottom to top. And let N_0 be r
  Now for every disconnected child C attach it to the chain on node:

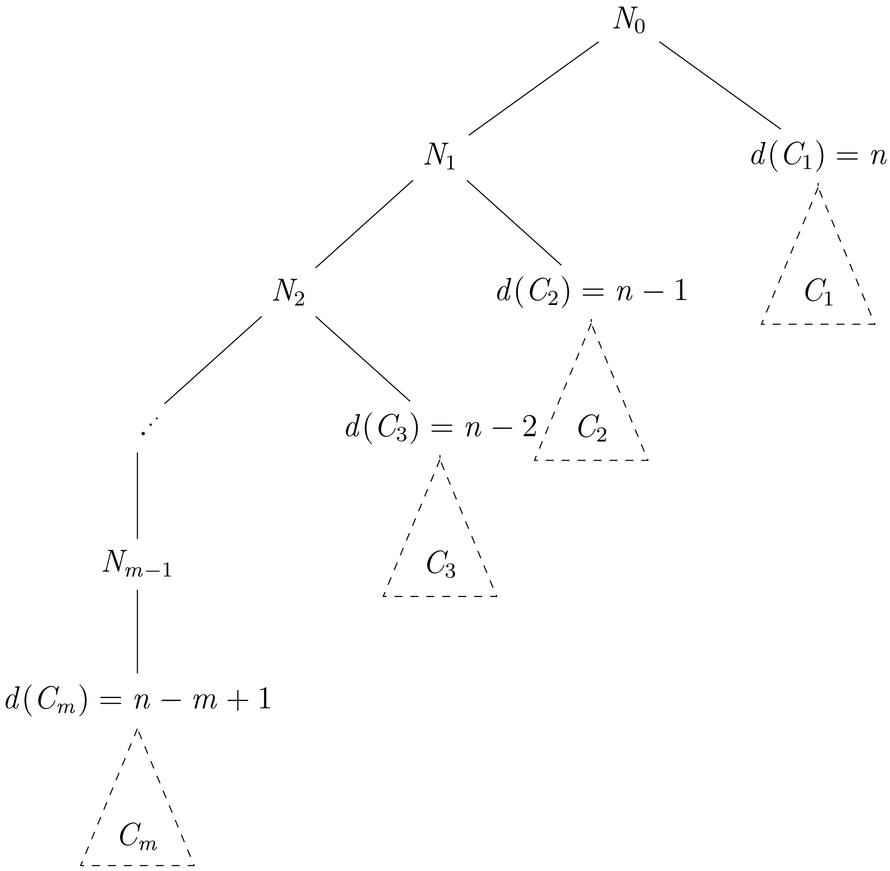
  $$N_{n-d(C)}$$

  Now $N_0, N_1, ..., N_{n-d_{min}}$ have unambiguous depth.
  Node $N_{n-d_{min}}$ has unambiguous depth because all its childs are of depth d_{min}.
  If $N_{n-d_{min}-k}$ has unambiguous depth so has $N_{n-d_{min}-k-1}$, because the depth of every attached child is $d_{min} + k$ as well as for the child node $N_{n-d_{min}-k}$. Every child of $N_{n-d_{min}-k-1}$ has unambiguous depth.

□



$$\Downarrow$$

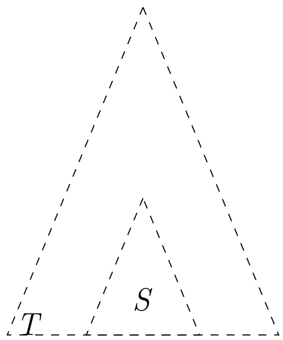


With this method we have got a way for ensuring that level-labeling and depth labeling are indeed synonymous, providing us an easy way to modify the AHU-Algorithm after our needs.

1.7 Properties for the lexicographical Product of Cographs

1.7.1 Relevance of Depth for finding isomorphic complete subtrees

If we want to find if some given minimal cotree $S = (r_S, \mathcal{C}_S)$ is isomorphic to some complete subtree $S' = (r_{S'}, \mathcal{C}_{S'})$ of some other minimal cotree $T = (r_T, \mathcal{C}_T)$, depth will play a certain roll.



As the mindful reader already detected:
If this complete subtree S' exists, its root will be found in all nodes of T with depth $d(S)$

Proof:
Because T is minimal it follows S' is minimal. Because it is isomorphic to S and S itself is minimal: the depth of the roots will be the same. $\square$

1.7.2 Relevance of Depth for finding factors of some cotree

If we want to find a factor of some cograph C with minimal cotree $T(C)_{min}$, we will be searching for a cotree isomorphic complete subtree, which root is located at depth d_f. In particular every node with depth d_f is the root of such a complete subtree iff the cograf of such subtree is a factor of C.

Proof:
The proof follows inductively over d_f:

- Induction base case ($d_f = 0$):
  Because the trivial cograph with one vertice is factor of every cograph the factor is found at every leaf of cotree T
- Induction hypothesis:
- Induction step:

1.8 Find Isomorph Trees with AHU

2 Practical Improvements

3 Algorithm