1 Definitions

Let $G = (V, E)$ be a rooted DAG of n nodes. We denote its roots by R and its set of (directed) paths $\mathcal{P}$. For a given path $p \in \mathcal{P}$, we denote its source $s(p)$, its destination $d(p)$ and its length $l(p)$.

Given a node v, its depht $\delta(v)$ is its distance to the root r.

Given a node v, its succesors are the nodes of the set: $S_v = \{u \in V | \exists e \in E, e = \{v, u\}\}$.

Given a node v, its predecessors are the nodes of the set: $P_v = \{u \in V | \exists e \in E, e = \{u, v\}\}$.

Given a node v, its ancestors are the nodes of the set: $A_v = \{u \in V | \exists p \in \mathcal{P}, s(p) = v \wedge d(p) = u\}$.

Given a node v, its descendents are the nodes of the set: $D_v = \{u \in V | \exists p \in \mathcal{P}, s(p) = u \wedge d(p) = v\}$.

1.1 Critical ancestor of a node

Given a node v, its critical ancestors are the nodes of the set:

$$CA_v = \{u \in A_v | \forall p \in \mathcal{P}, s(p) \in R \wedge d(p) = v \Rightarrow u \in p\}$$

Given a node v, its lowest critical ancestor (denoted LCA_v) is the node u of CA_v satisfying $D_u \cap CA_v = \emptyset$. The lowest critical ancestor is always unique.

we dont need this definition.

1.2 Common critical ancestors of a pair of nodes

Given two nodes u and v, their common critical ancestors are the nodes of the set:

$$CCA_{u,v} = \{w \in A_u \cap A_v | \forall p \in \mathcal{P}, s(p) \in R \wedge (d(p) = u \vee d(p) = v) \Rightarrow w \in p\}$$

This definition can also be summarised as:

$$CCA_{u,v} = \{w \in V | w \in CA_u \wedge w \in CA_v\}$$

1.3 Lowest common critical ancestor of a pair of nodes

Given two nodes u and v, their lowest common critical ancestor (denoted $LCCA_{u,v}$) is the node w of $CSA_{u,v}$ satisfying $D_w \cap CA_{u,v} = \emptyset$.

1.4 Lowest common single ancestor of two nodes

Given two nodes u and v, their common ancestors are the nodes of the set:

$$CA_{u,v} = \{w \in V | (\exists p \in \mathcal{P}, s(p) = w \wedge d(p) = v) \wedge (\exists p \in \mathcal{P}, s(p) = w \wedge d(p) = u)\}$$

.

equivalent to $CA_{u,v} = A_v \cap A_u$ No

Given two nodes u and v, their common single ancestors are the nodes of the set: $CSA_{u,v} = \{w \in CA_{u,v} | (\forall p \in \mathcal{P}, s(p) = r \wedge d(p) = u \Rightarrow w \in p) \wedge (\forall p \in \mathcal{P}, s(p) = r \wedge d(p) = v \Rightarrow w \in p)\}$.

Given two nodes u and v, their lowest common single ancestors (denoted $LCSA_{u,v}$) is the node w of $CSA_{u,v}$ satisfying $D_w \cap CSA_{u,v} = \emptyset$.

A few meetings back we decided to give up the term single ancestor. This is because computing the lowest ancestor inherently gives us a single node and calling ancestors common and single was confusing. So we choose the name "lowest common critical ancestor".

2 Main Property

What is the property between $LCSA_{u,v}$, CA_u, and CA_v used by our algorithm?

Our algorithm defines for every node the set CA_u. We can then use this set to compute the $CCA_{u,v}$ by taking the set intersection as defined earlier.

If we can find a relation with the depth, the proof will become easy! Is the following property correct?

Given two nodes u and v, $LCSA_{u,v}$ is the node of $CA_u \cap CA_v$ with the highest depth.

This is infact correct and except the name. $LCSA_{u,v}$ is $LCCA_{u,v}$ We have to prove this property!

3 Labeling Algorithm

3.1 Algorithm

Given a rooted DAG G, we compute (with a breadth-first traversal) for each node v a set S_v according to the following rule:

$$S_v = \begin{cases} \{v\} \text{ if } v = r \\ \{v\} \cup \{\cap_{u \in P_v} S_u\} \text{ otherwise} \end{cases}$$

Association of each node with its depht (for the LCSA computation)?

3.2 Proof

Lemma 1 *For each node $v \in V$, the set S_v computed by the Labeling Algorithm is CA_v.*

Proof. TO DO □

4 LCSA Algorithm

4.1 Algorithm

What does exactly this algorithm?

I believe that it is something like that (but I am not really sure):

Given a rooted DAG G labeled by the Labeling Algorithm, and two nodes $u, v \in V$, the LCSA Algorithm returns the node of $S_u \cap S_v$ with the highest depth.

4.2 Proof

Lemma 2 *For each pair of node $u, v \in V$, the node computed by the LCSA Algorithm is $LCSA_{u,v}$.*

Proof. TO DO - easy if the main property is true! □