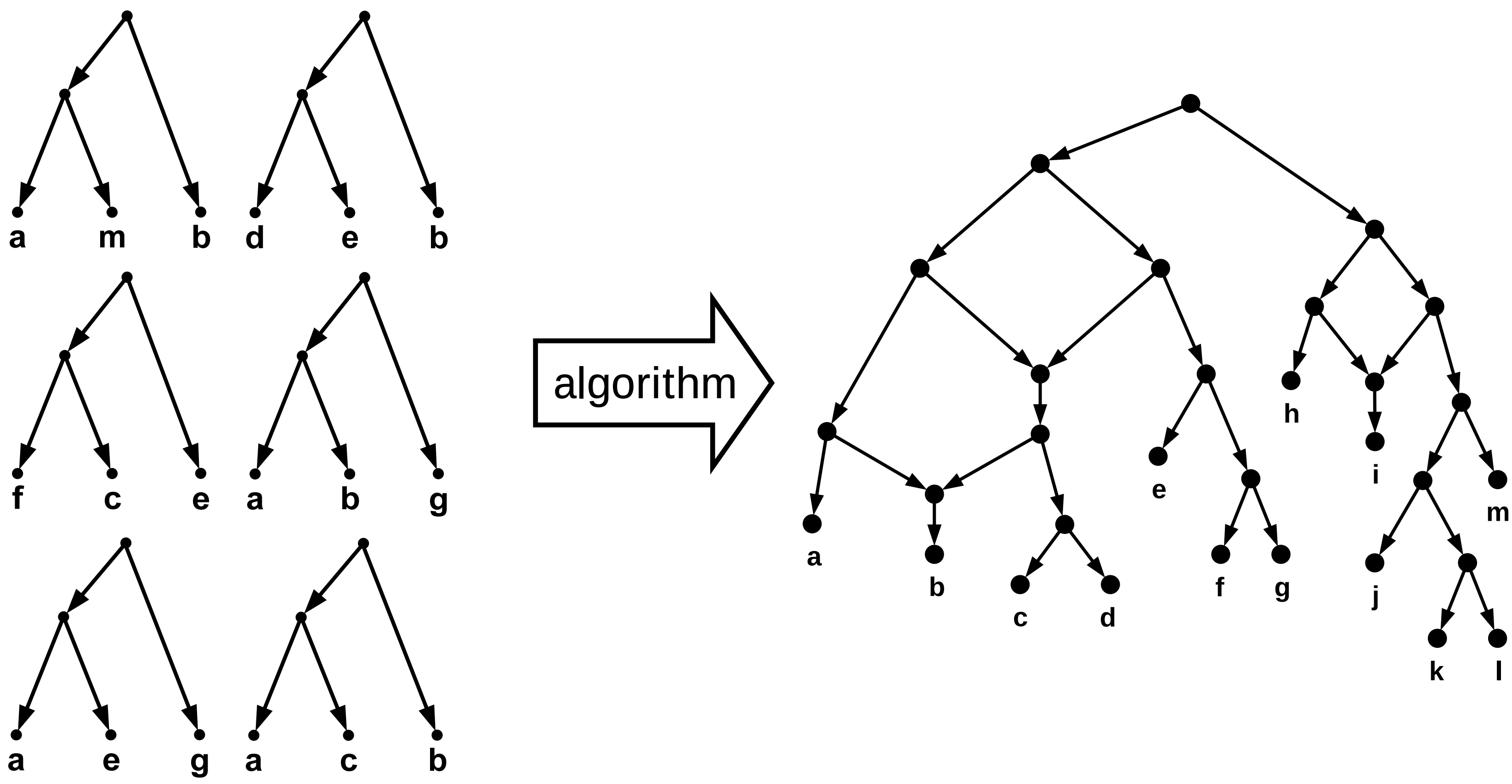
Level-*k* Phylogenetic Networks

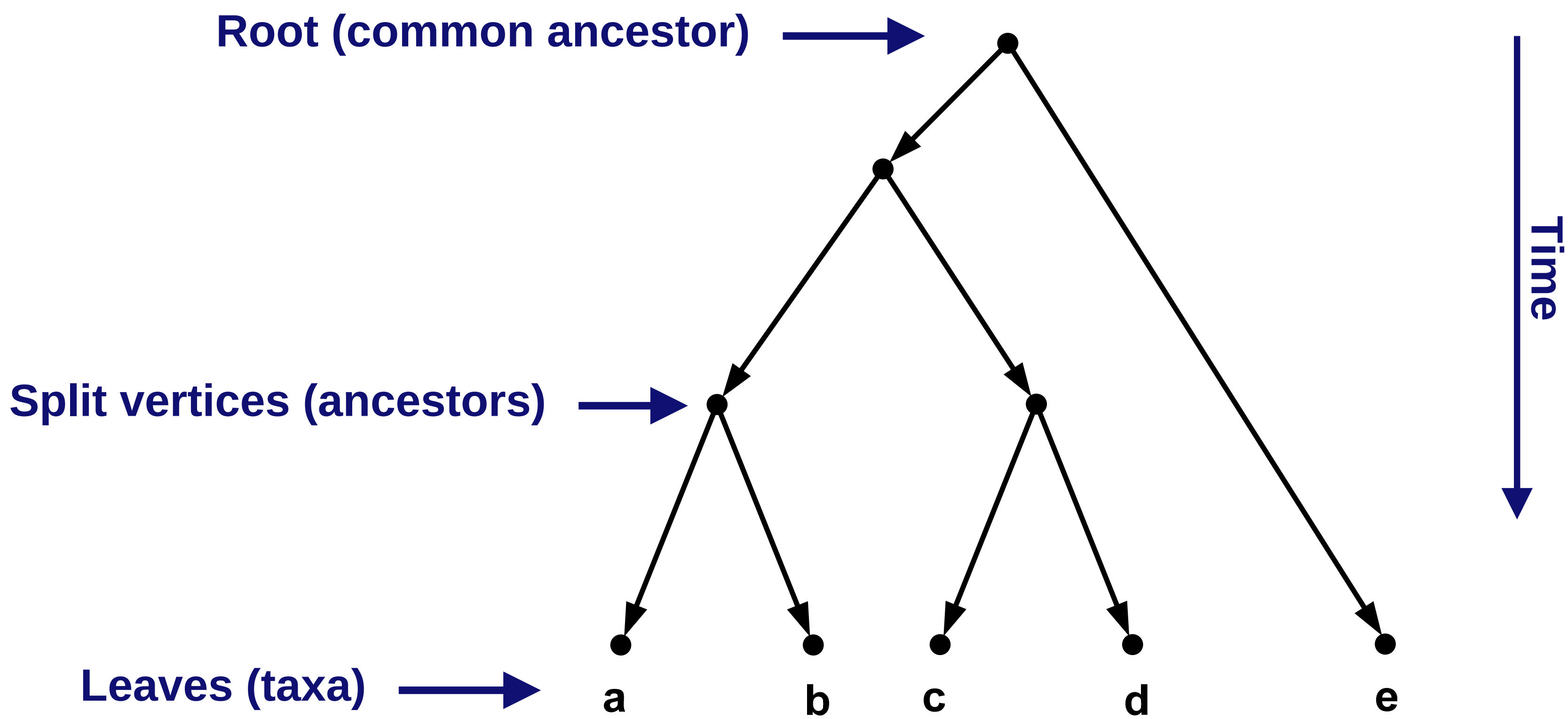
Leo van Iersel, Steven Kelk and Matthias Mnich

Part of this research has been funded by the Dutch BSIK/BRICKS project AFM2.

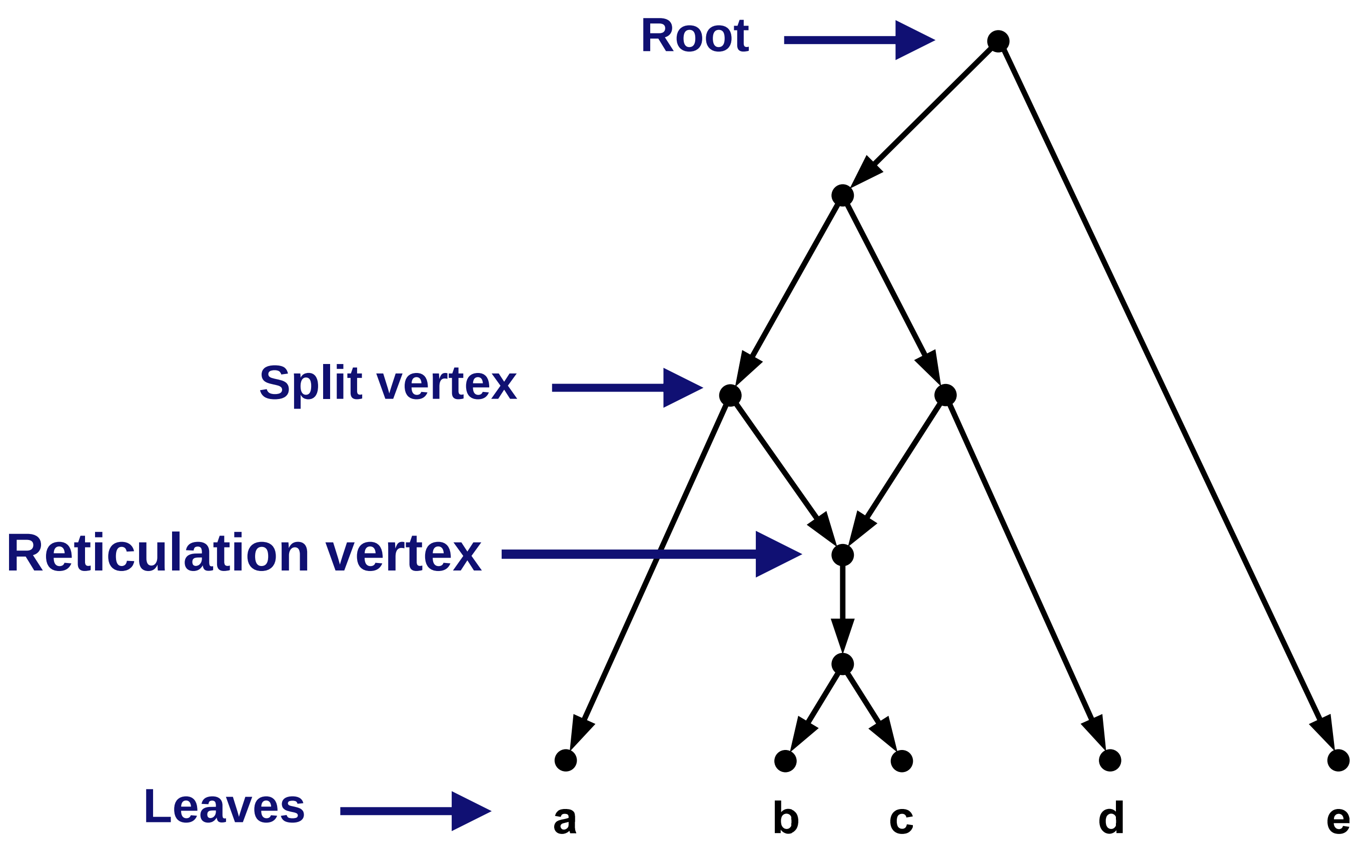
- Combine a set of small trees (triplets) into a single network that is as simple as possible



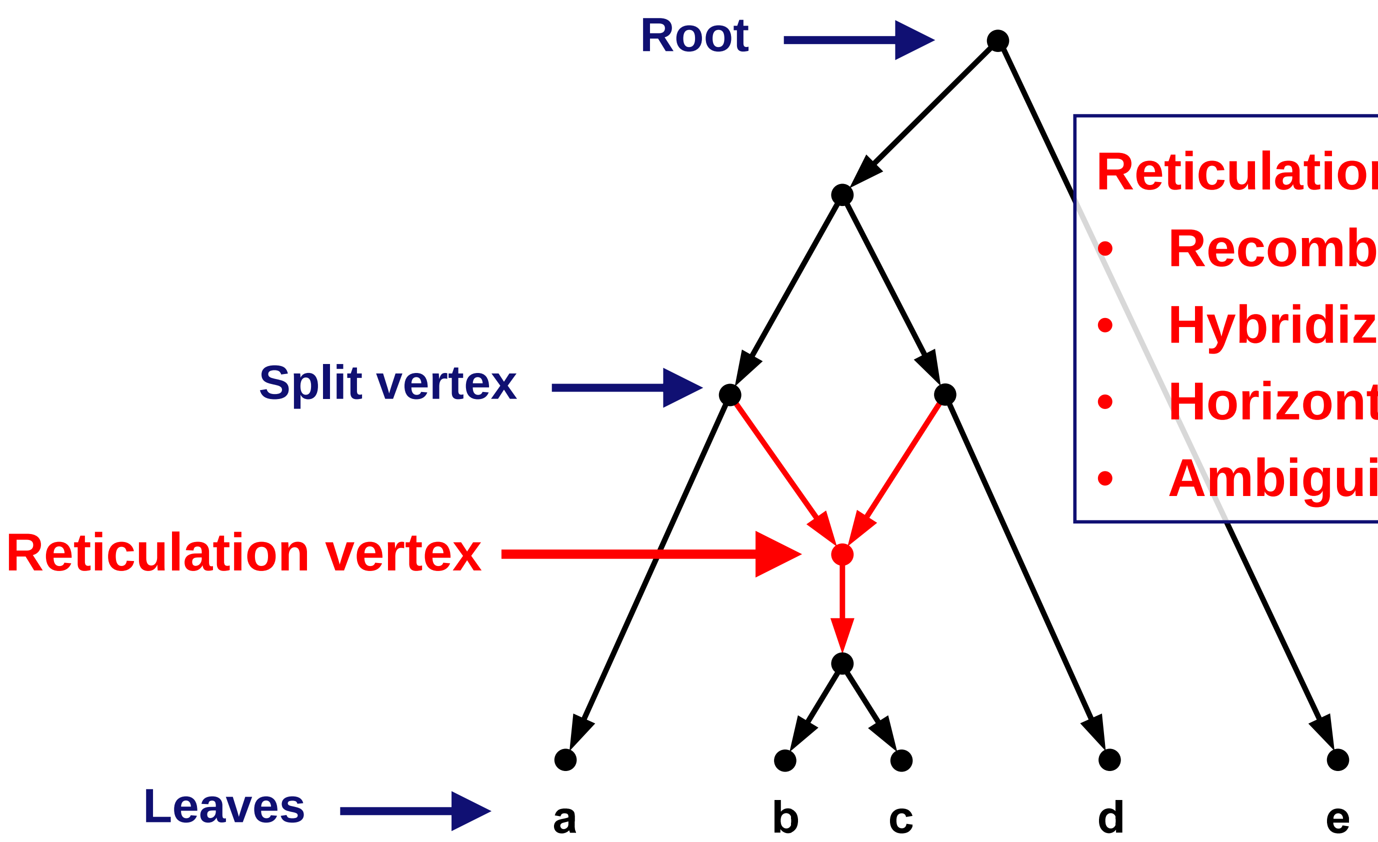
Phylogenetic Trees



Phylogenetic Networks



Phylogenetic Networks



Reticulation can model:

- **Recombination**
- **Hybridization**
- **Horizontal gene transfer**
- **Ambiguity**

How Simple is a Network?

Number of reticulations: total number of reticulation vertices (indegree two vertices)

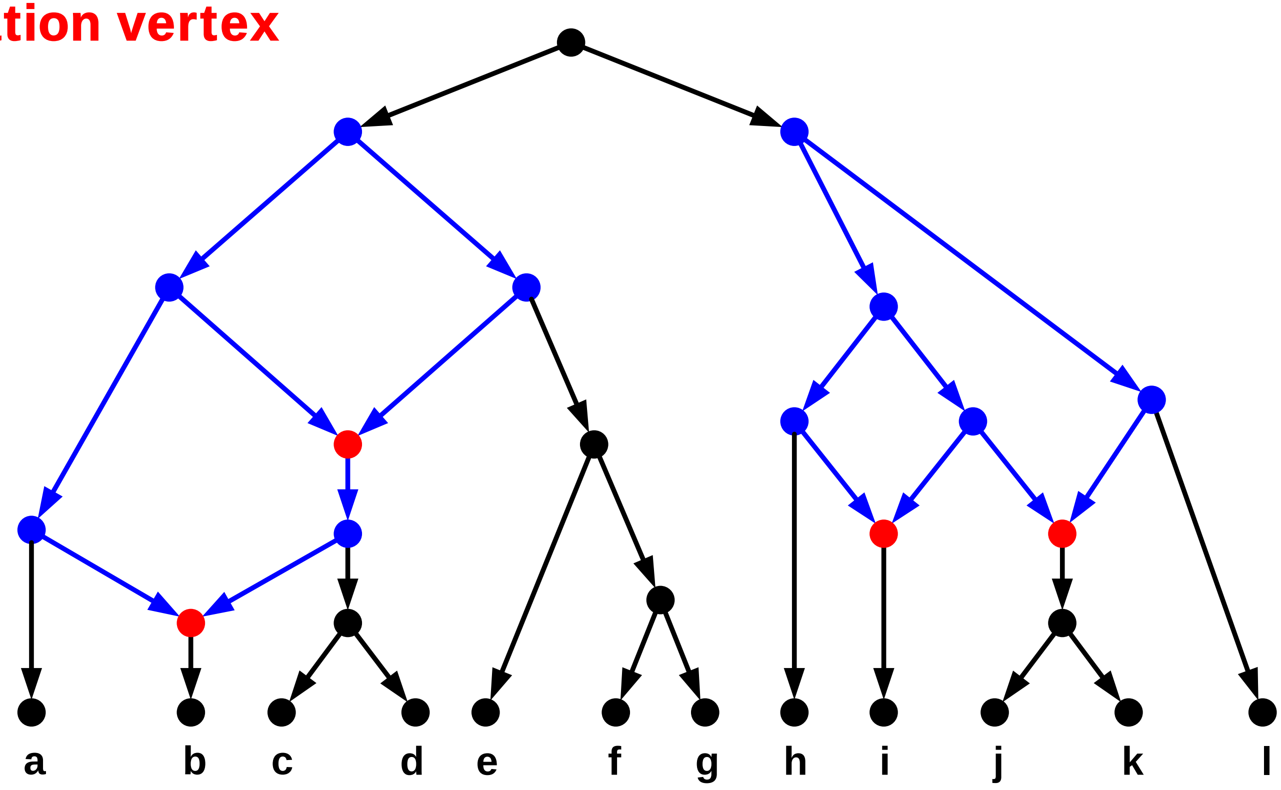
Level: maximum number of reticulation vertices in a biconnected component

- Level-0 networks are trees
- Level-1 networks are galled trees

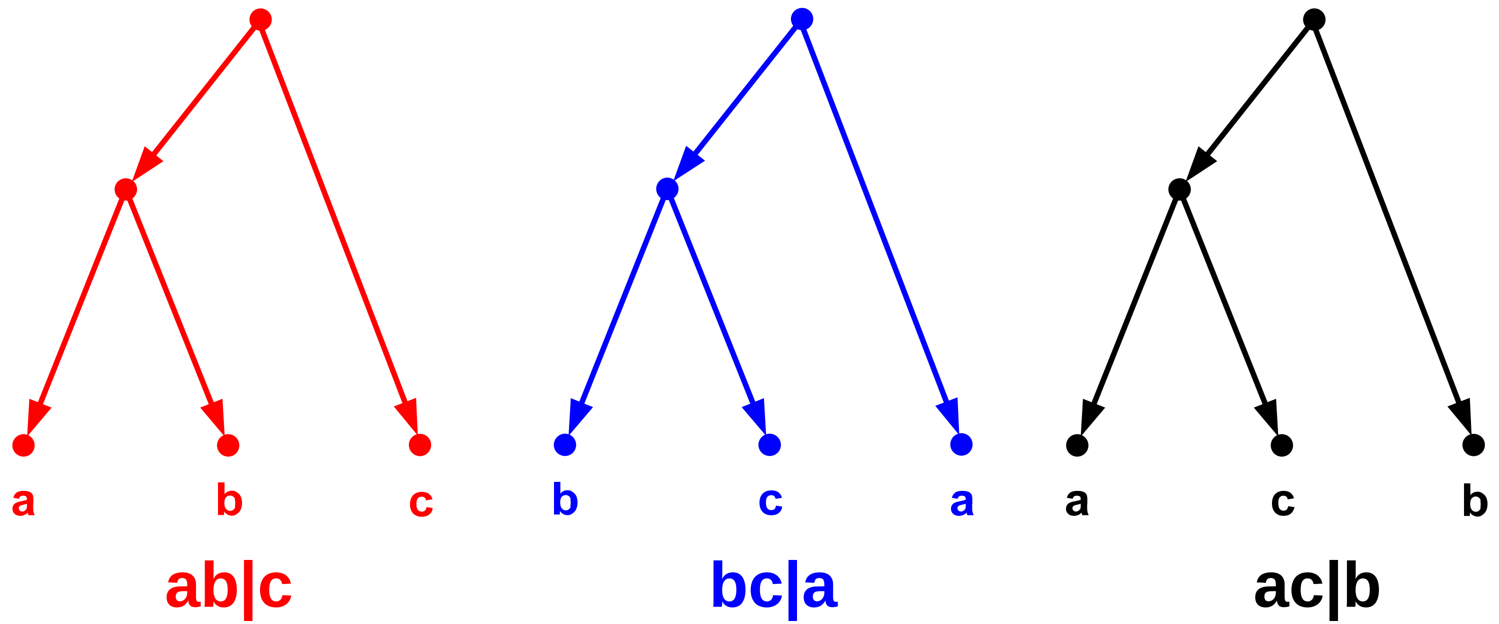
Example: level-2 network with 4 reticulations

blue = biconnected component

red = reticulation vertex



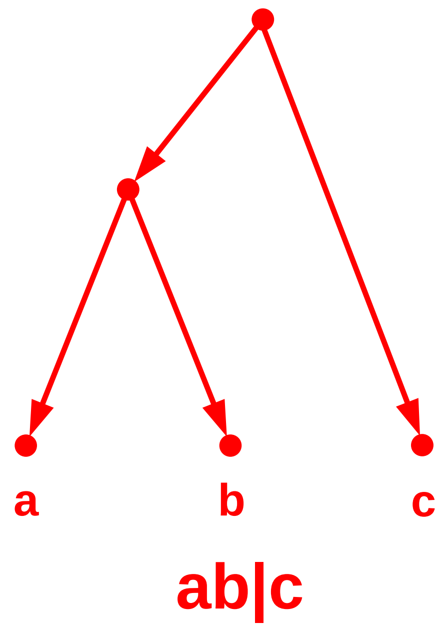
Triplets

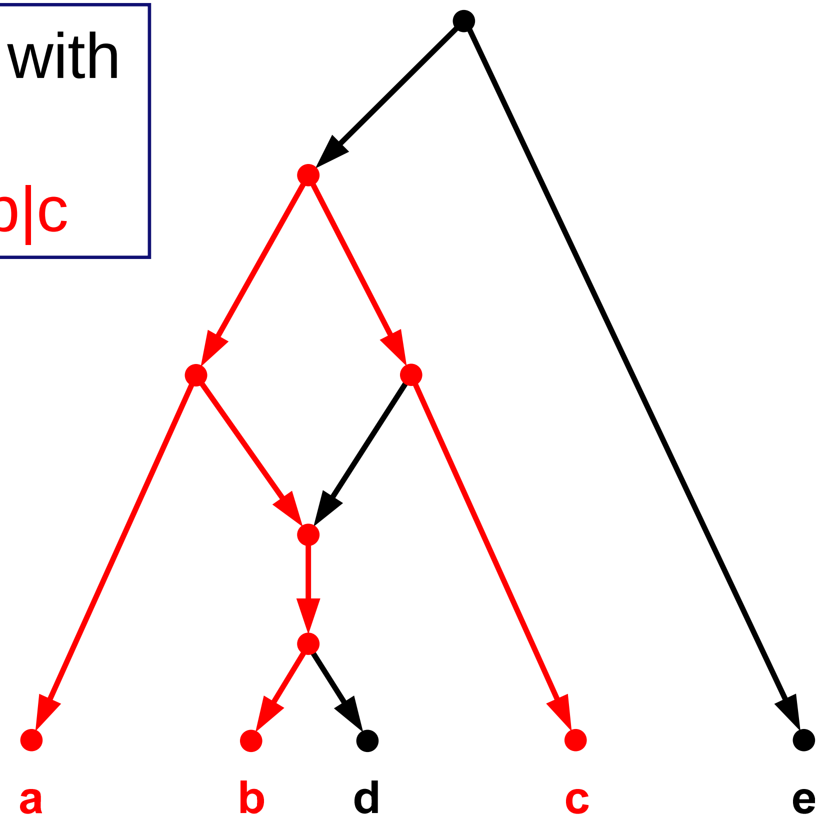


- A triplet set is ***dense*** if for each combination of three leaves it contains at least one of the three possible triplets

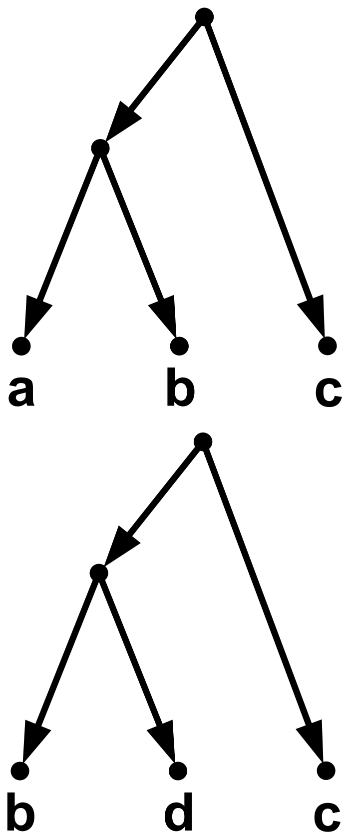
Triplet Consistency

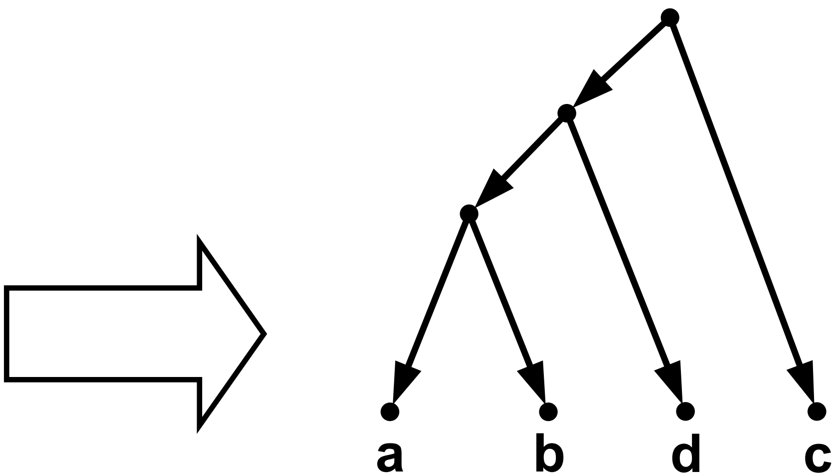
A triplet ab|c is ***consistent*** with a network if this network contains a subdivision of ab|c





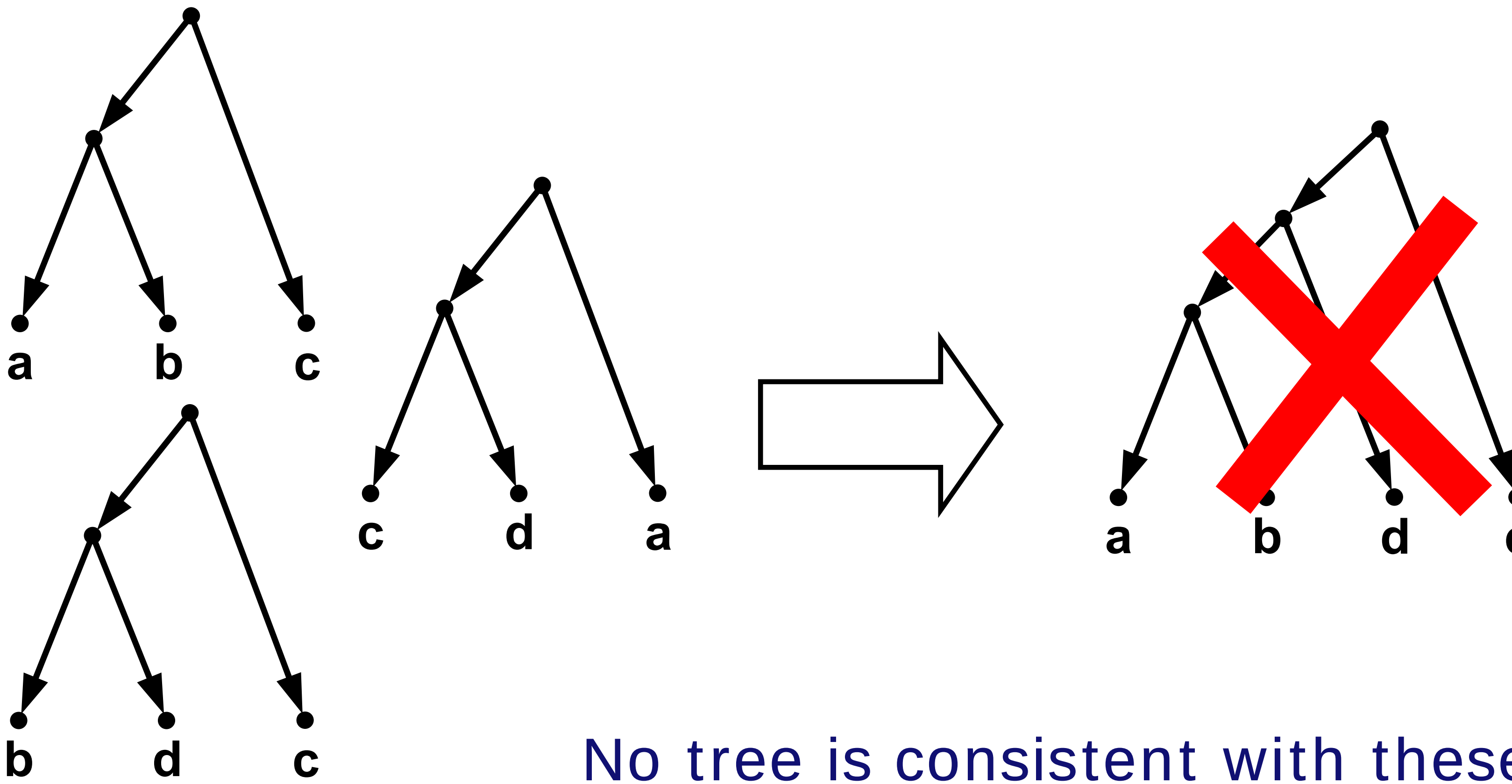
Example.....





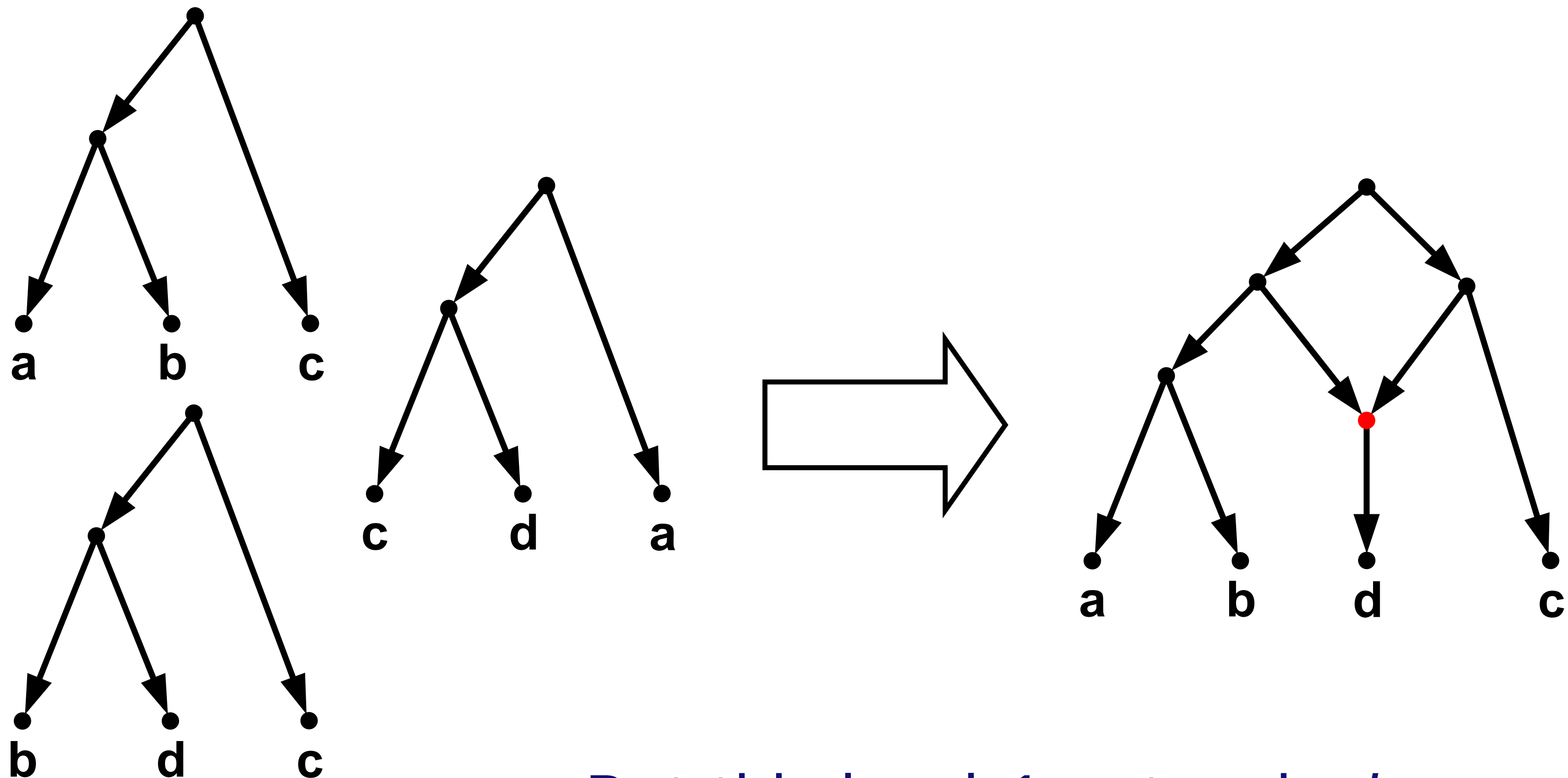
These two triplets can be combined into a tree

Example.....



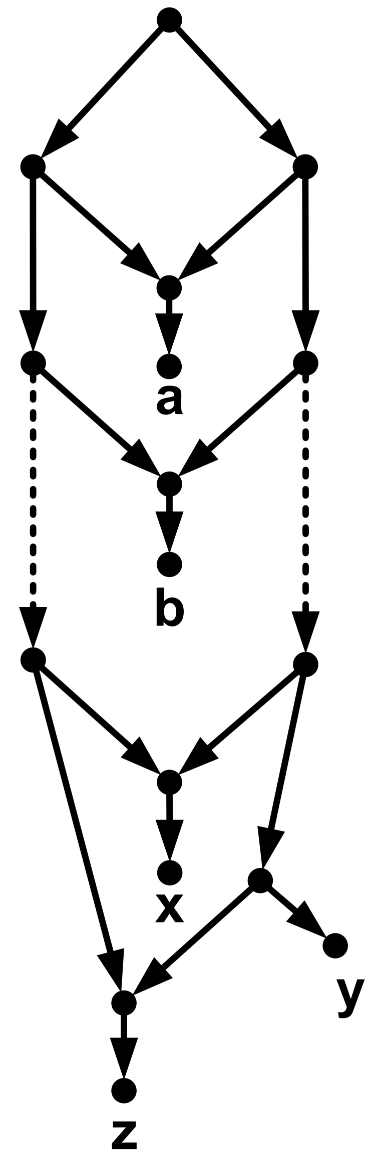
No tree is consistent with these three triplets

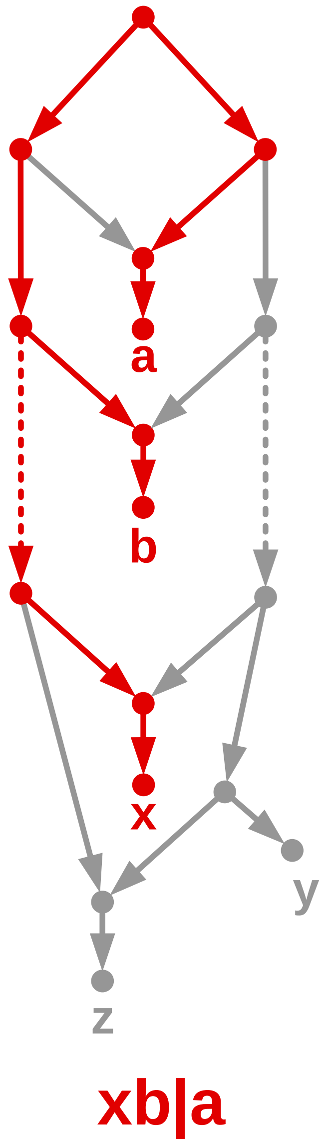
Example.....

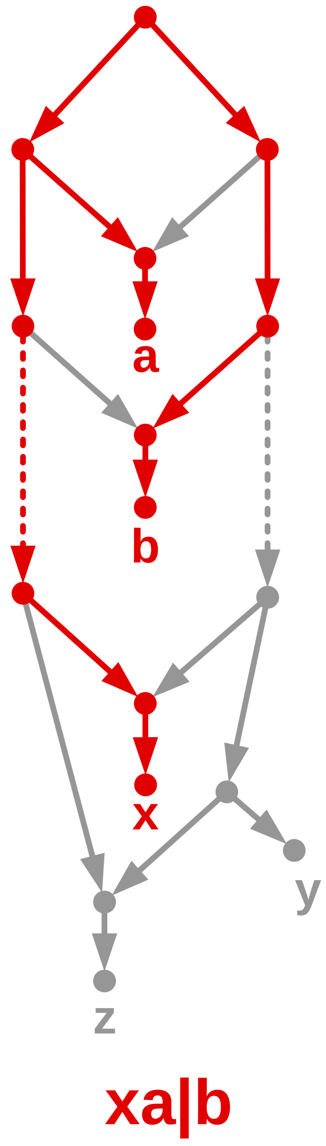


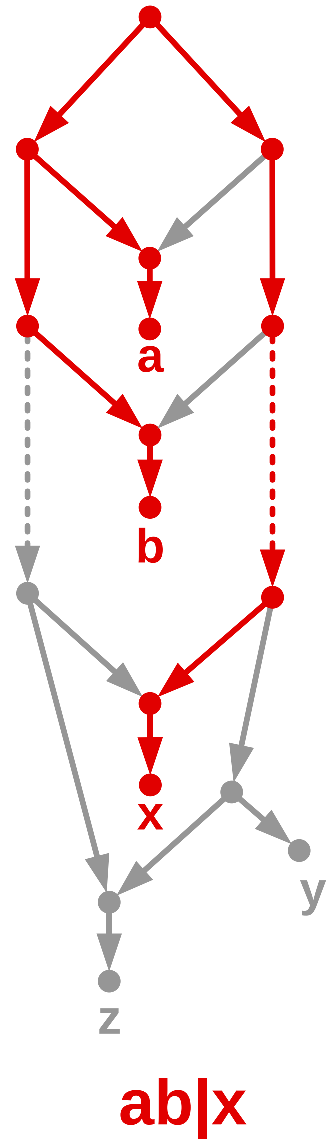
But this level-1 networks *is*.

A level-(n-1) network consistent with **any** triplet set on n leaves




a
b
x
y
z
xb|a


a
b
x
y
z
xa|b


a
b
x
y
z
ab|x

Constructing a level-k network that is…

consistent with a maximum number of input triplets	NP-hard for all k

Constructing a level-k network that is…

consistent with a maximum number of input triplets	NP-hard for all k
consistent with **all** input triplets	NP-hard for all k > 0

Constructing a level-k network that is…

consistent with a maximum number of input triplets	NP-hard for all k
consistent with **all** input triplets	NP-hard for all k > 0
consistent with a maximum number of input triplets from a **dense** triplet set	NP-hard for all k

Approach for proving NP-hardness for all *k*

- For the consistency problem we generalise the reduction for level-1.

Approach for proving NP-hardness for all *k*

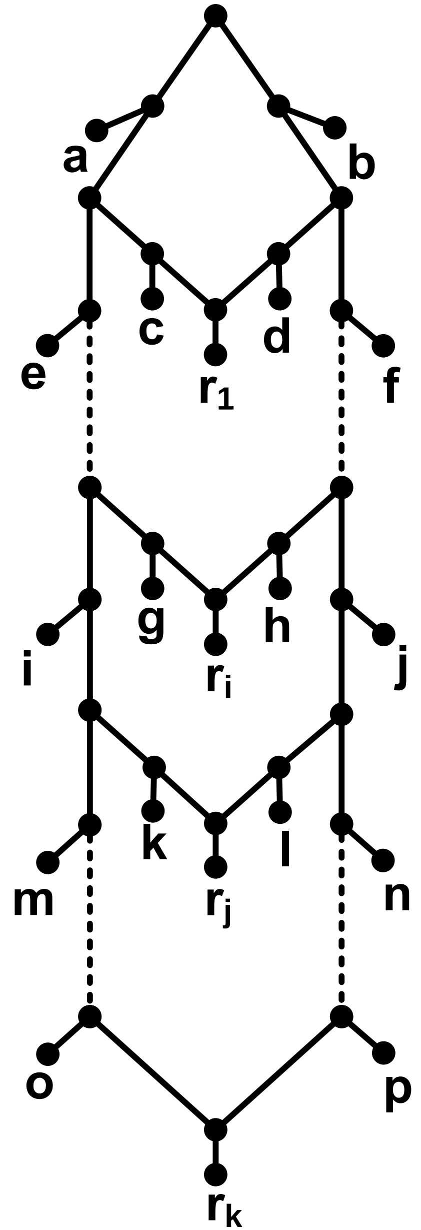
- For the consistency problem we generalise the reduction for level-1.
- For the maximisation problem we generalise the reduction for level-0, and add triplets to make the instance dense.

Approach for proving NP-hardness for all *k*

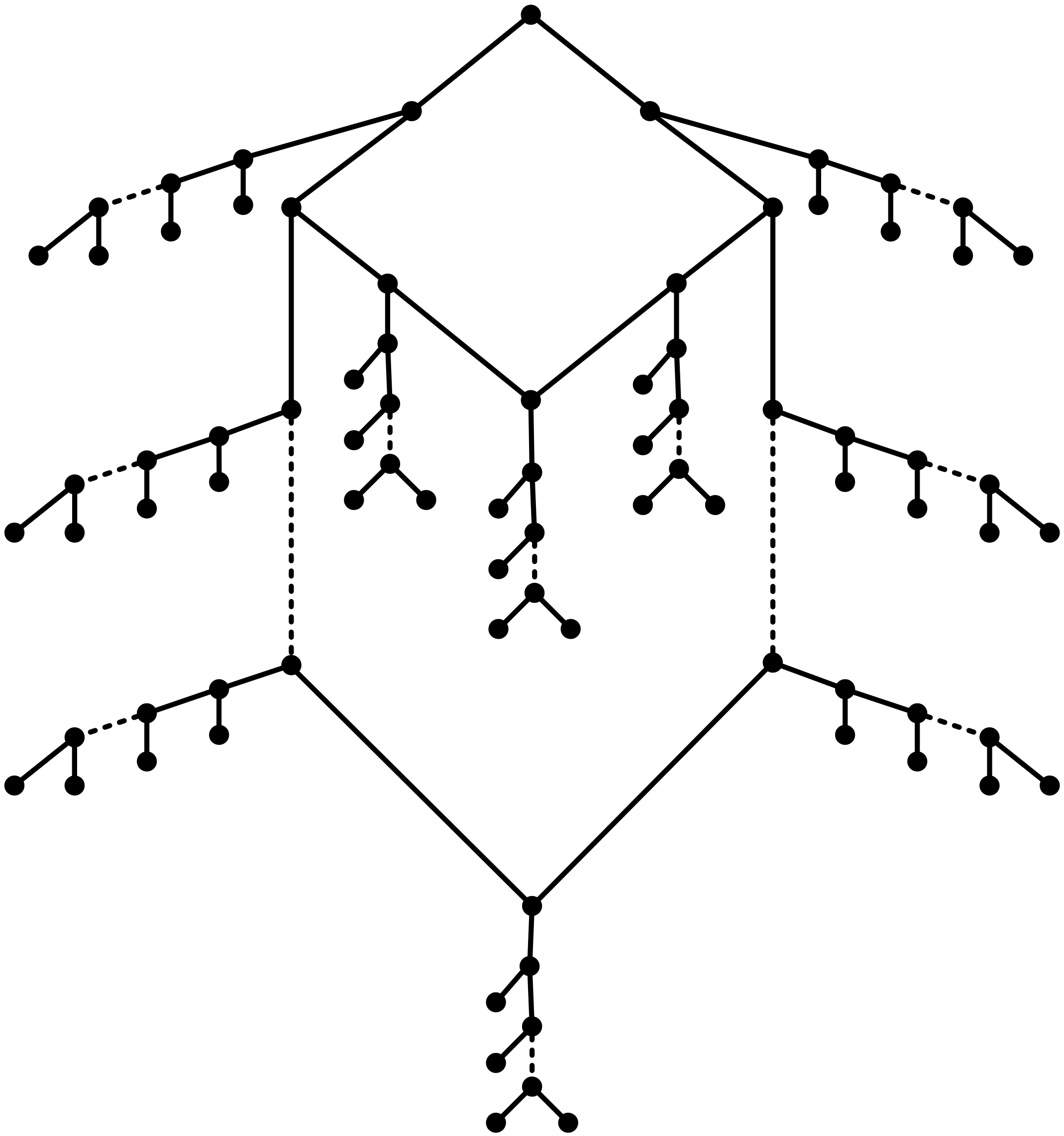
- For the consistency problem we generalise the reduction for level-1.
- For the maximisation problem we generalise the reduction for level-0, and add triplets to make the instance dense.
- To be able to generalise the different reductions we show that the following network is “unique”…

Let T be the set of triplets consistent with this network.

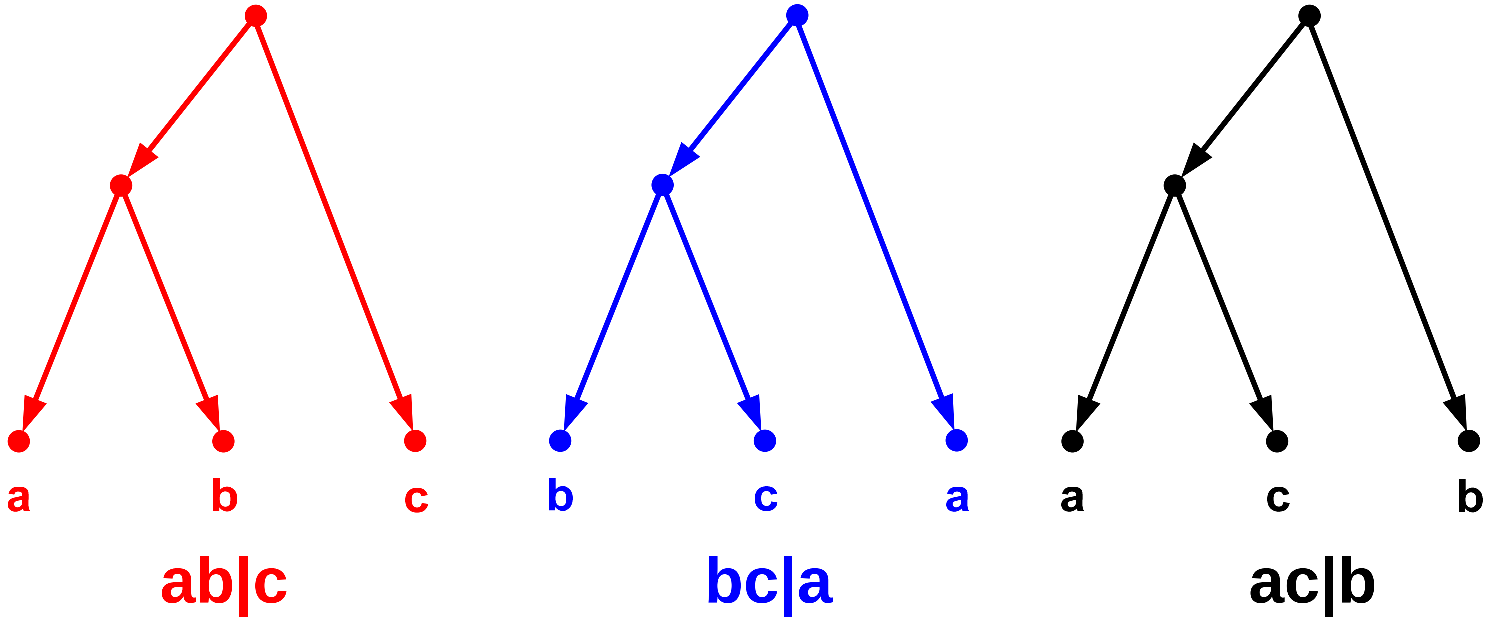
Then this is the only network consistent with T.



This network is even “unique” for the maximisation problem



Density



- A triplet set is ***dense*** if for each combination of three leaves it contains at least one of the three possible triplets

Constructing a level-k network that is…

consistent with all input triplets from a **dense** triplet set	**polynomial-time** for k = 1 and k = 2; **open** for k > 2

Constructing a level-k network that is…

consistent with all input triplets from a **dense** triplet set	**polynomial-time** for k = 1 and k = 2; **open** for k > 2
… and containing a **minimum number of reticulations**	**polynomial-time** for k = 1 and k = 2; **open** for k > 2

Constructing a level-k network that is...

consistent with all input triplets from a **dense** triplet set	**polynomial-time** for k = 1 and k = 2; **open** for k > 2
… and containing a **minimum number of reticulations**	**polynomial-time** for k = 1 and k = 2; **open** for k > 2
consistent with **precisely** the input triplets	**polynomial-time** for all *k*

Given: general set of triplets
Construct: level-k network consistent with a maximum number of input triplets

- Bang Ye Wu gave an $O(3^n)$ algorithm for level-0
- We give an $O(4^n)$ algorithm for level-1

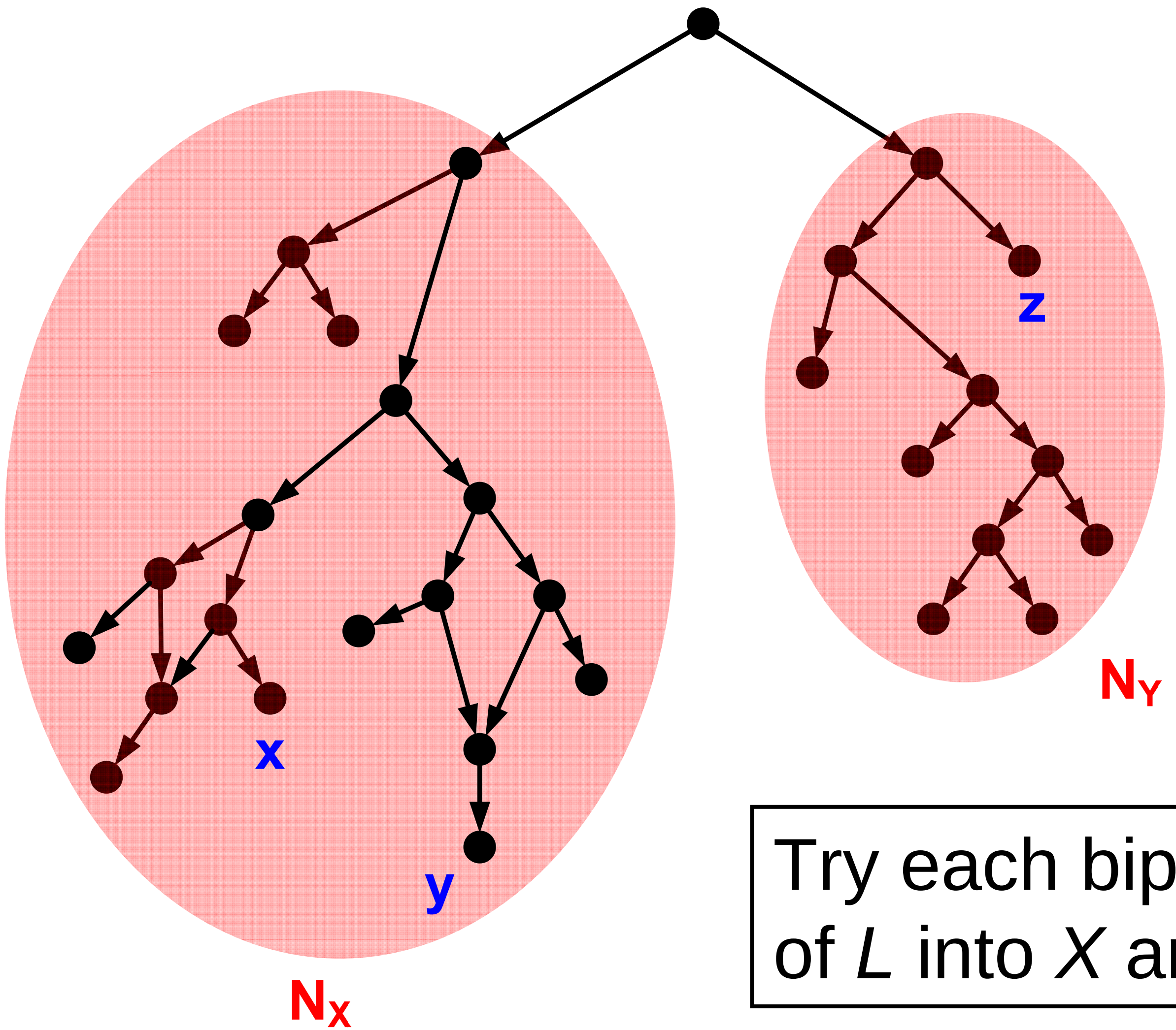
Given: general set of triplets
Construct: level-k network consistent with a maximum number of input triplets

- Bang Ye Wu gave an $O(3^n)$ algorithm for level-0
- We give an $O(4^n)$ algorithm for level-1
- Are faster algorithms possible?
- Is anything better than a 3-approximation possible for level-0?

The Exact Algorithm

- Loop through all subsets of the leaves from small to large
- Compute an optimal network for these leaves based on previously computed optimal networks for smaller leaf-sets

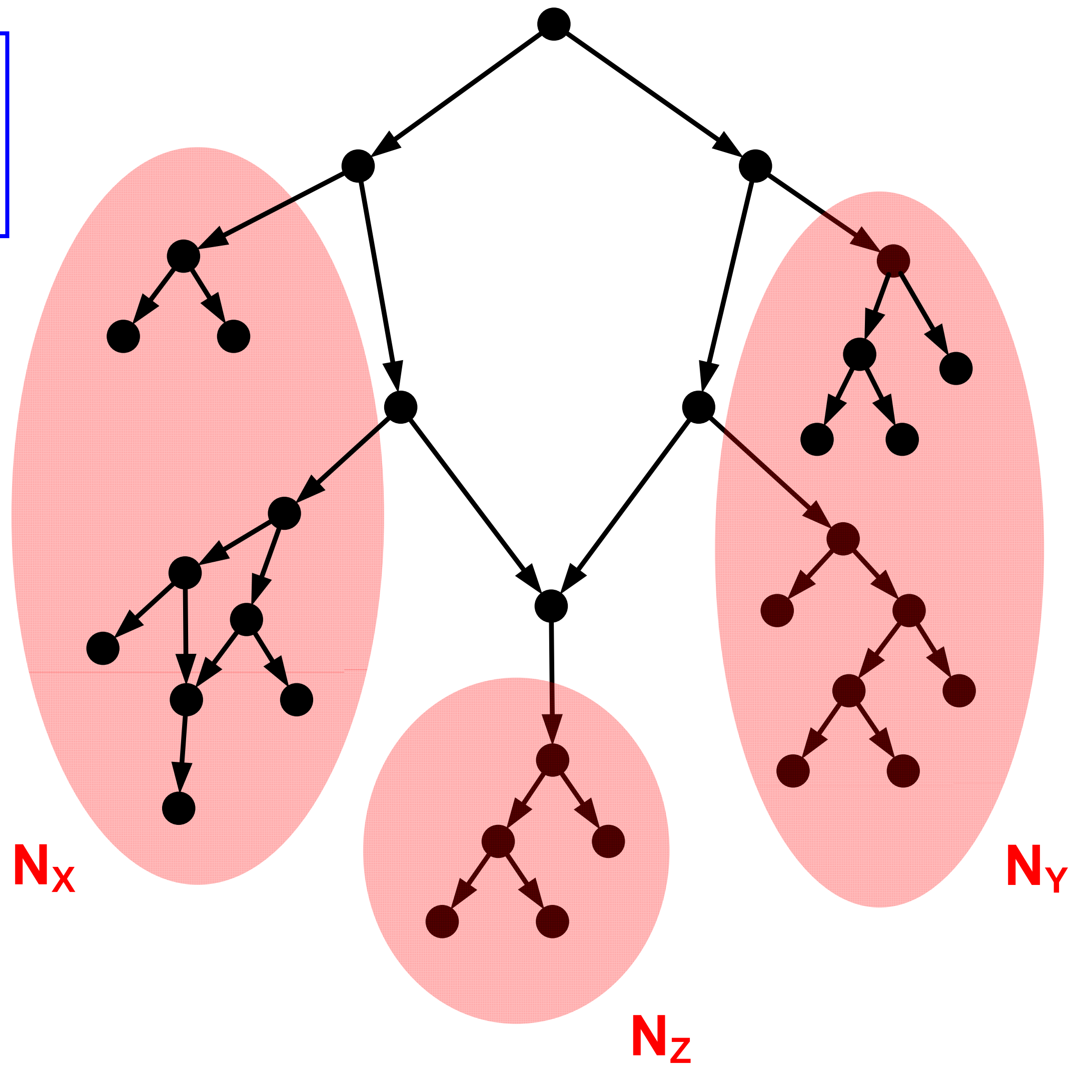
Case 1: root is not in a cycle



Try each bipartition of *L* into *X* and *Y*

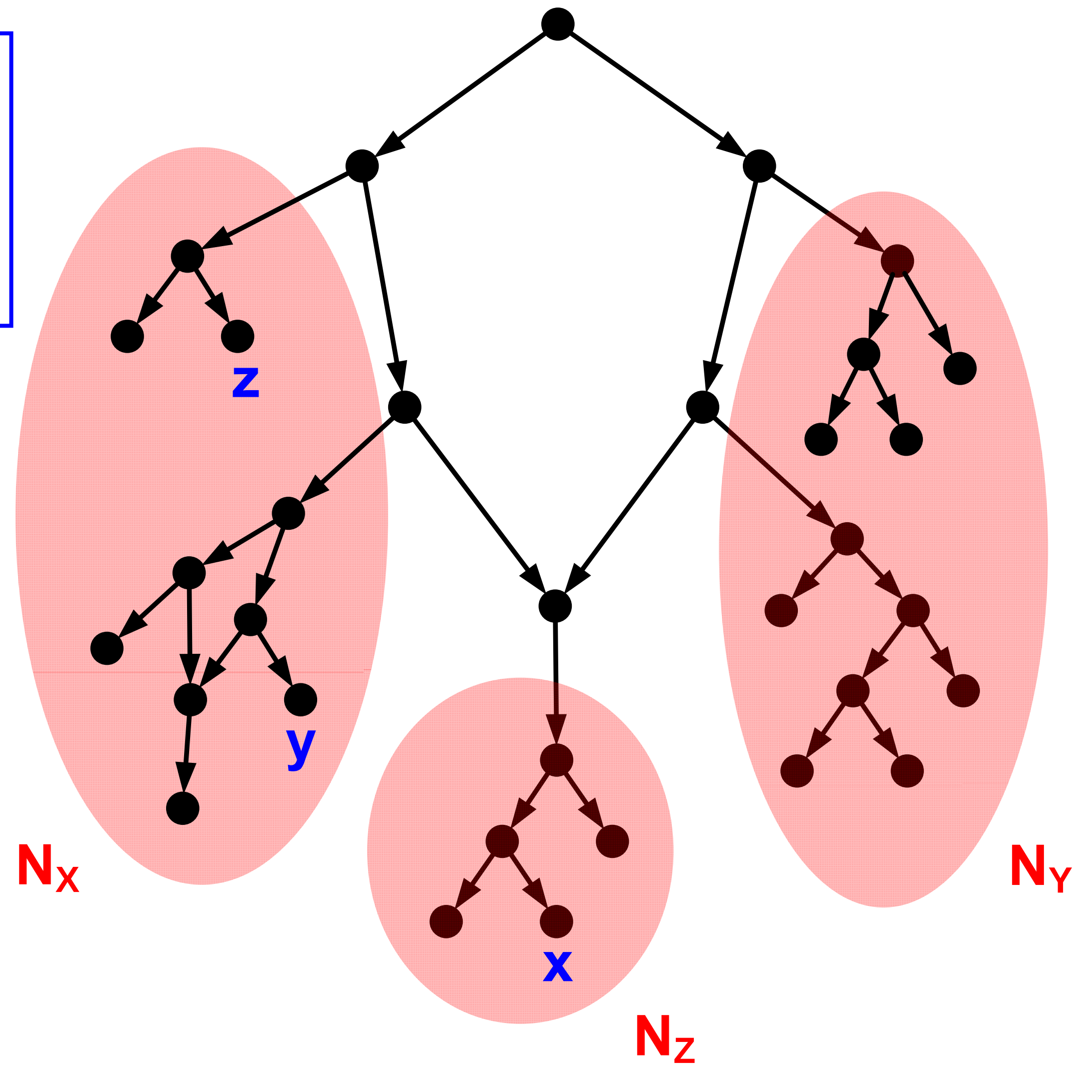
Case 2: root is in a cycle

Try each tripartition of *L* into *X*, *Y* and *Z*



Problem

What happens with triplets that cross the tripartition?

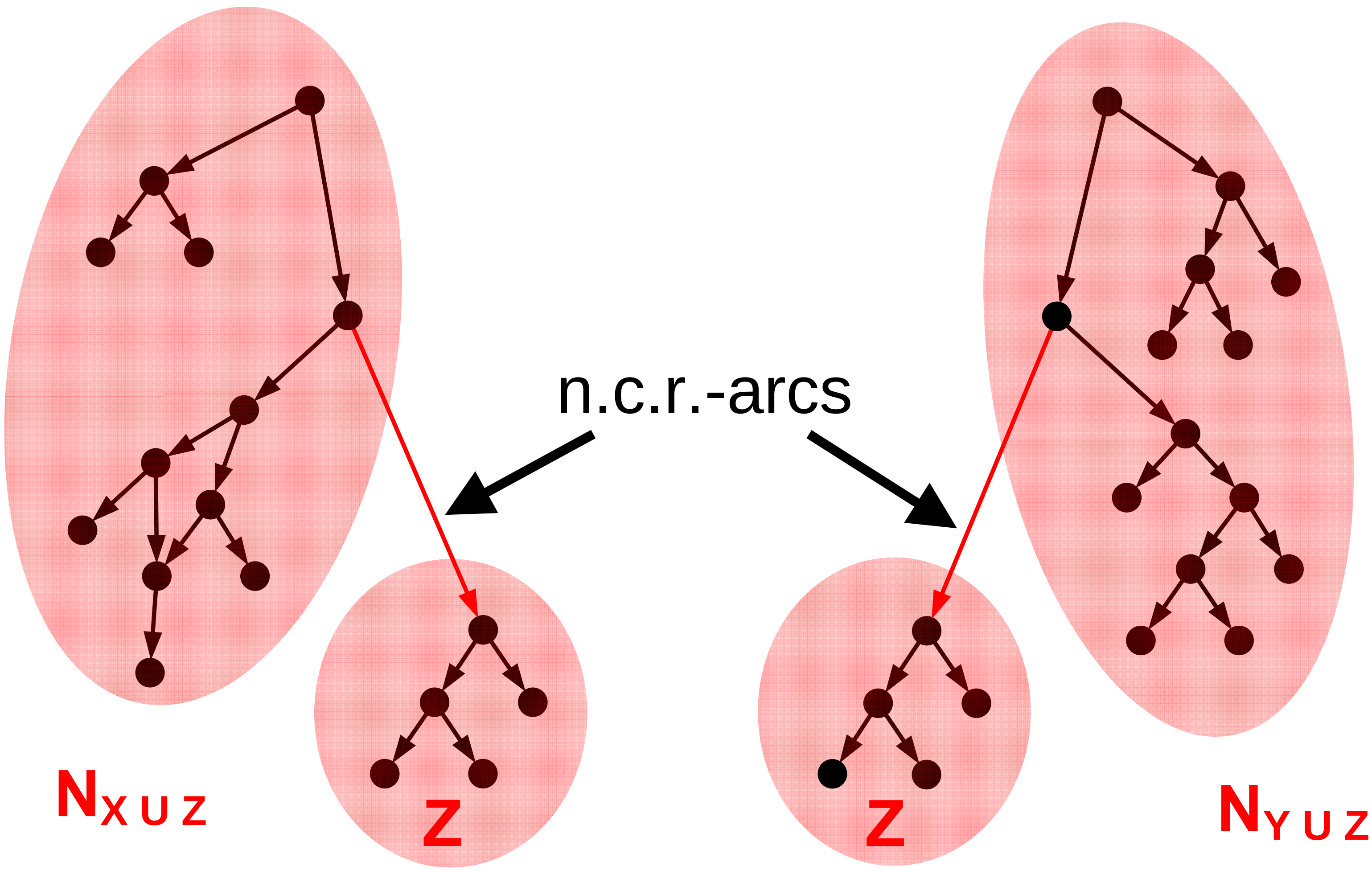


Solution

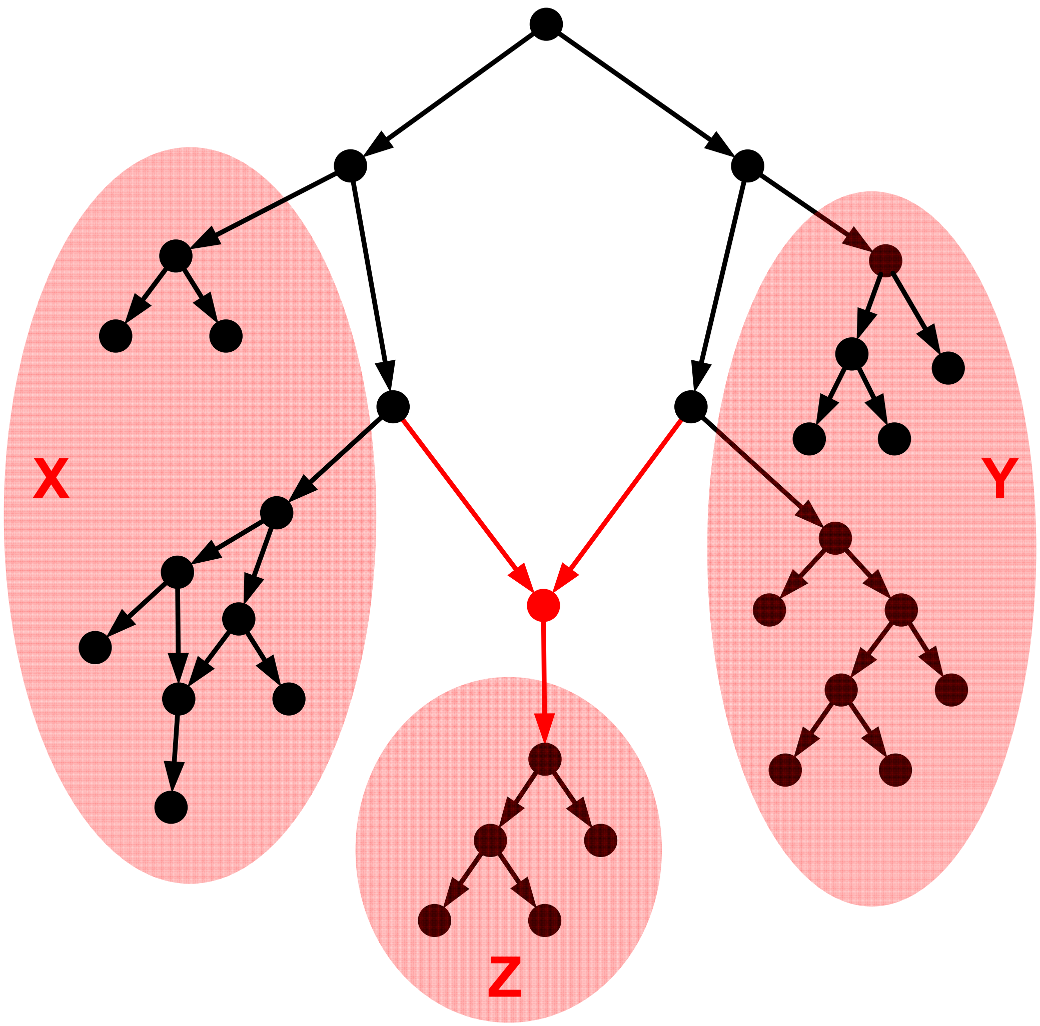
- Combine networks for X U Z and Y U Z.
- Where Z has to be below an “n.c.r.-arc”.

- Definition: an arc is an *n.c.r.-arc* if it is not reachable from any vertex in a cycle.

Solution



Solution



- Leo van Iersel and Steven Kelk, Constructing the Simplest Possible Phylogenetic Network from Triplets, submitted.

- Leo van Iersel, Steven Kelk and Matthias Mnich, Uniqueness, Intractability and Exact Algorithms: Reflections on Level-*k* Phylogenetic Networks, submitted.

- Leo van Iersel, Judith Keijsper, Steven Kelk, Leen Stougie, Ferry Hagen and Teun Boekhout, Constructing Level-2 Phylogenetic Networks from Triplets, in proceedings of RECOMB 2008.

- All papers (and some implementations): **http://www.win.tue.nl/~liersel**