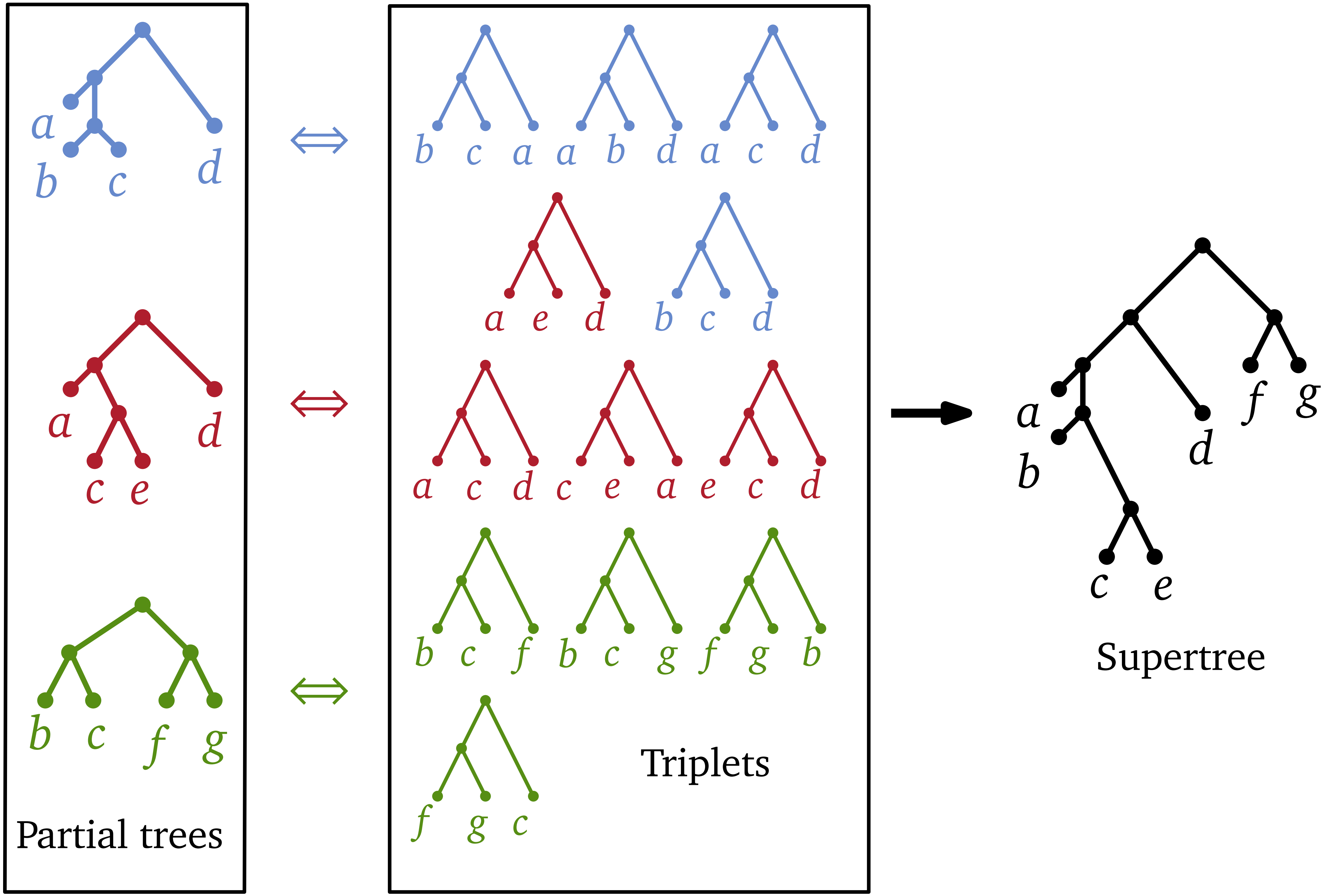
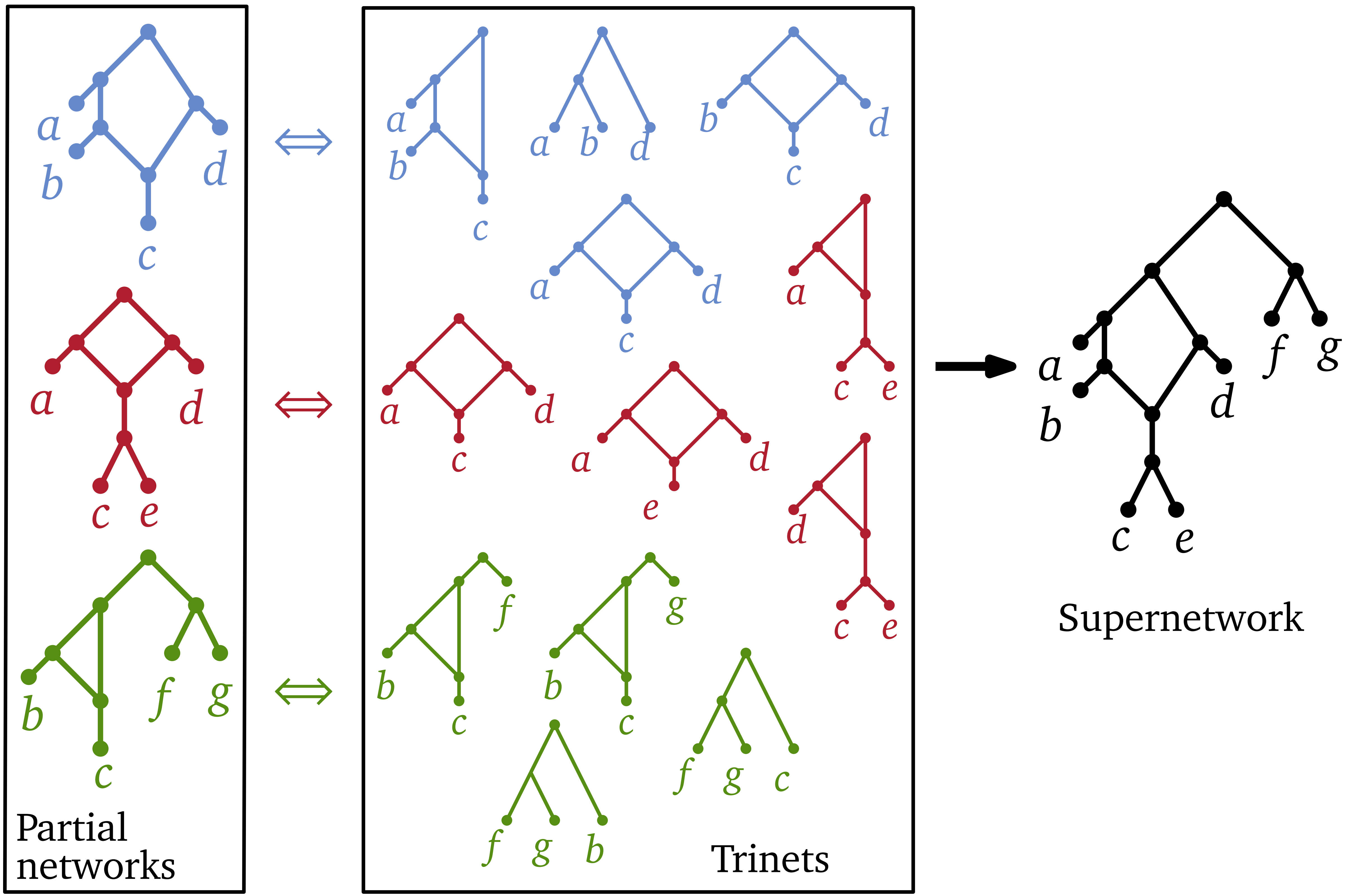
Reconstructing phylogenetic level-1 networks from nondense binet and trinet sets

Katharina Huber, **Leo van Iersel**, Vincent Moulton,
Celine Scornavacca and Taoyang Wu

Supertrees



Supernetworks

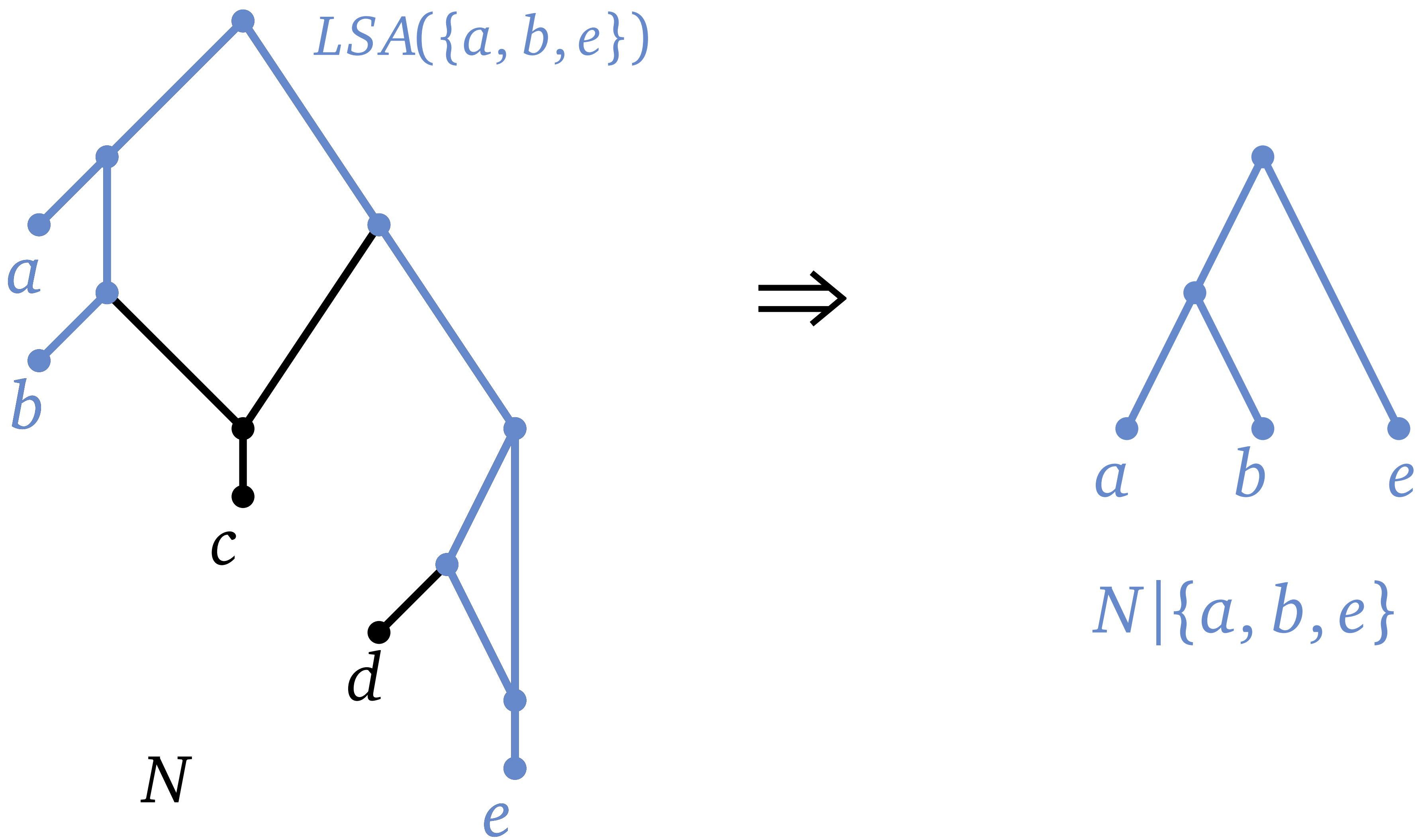


Trinets and Subnets

The ***lowest stable ancestor (LSA)*** of a set X' of taxa is the last vertex that is on all paths from the root to a leaf in X'.

The ***subnet*** $N|X'$ is obtained by

1. taking all directed paths in N from $LSA(X')$ to a leaf in X';
2. suppressing indegree-1 outdegree-1 vertices and parallel arcs.

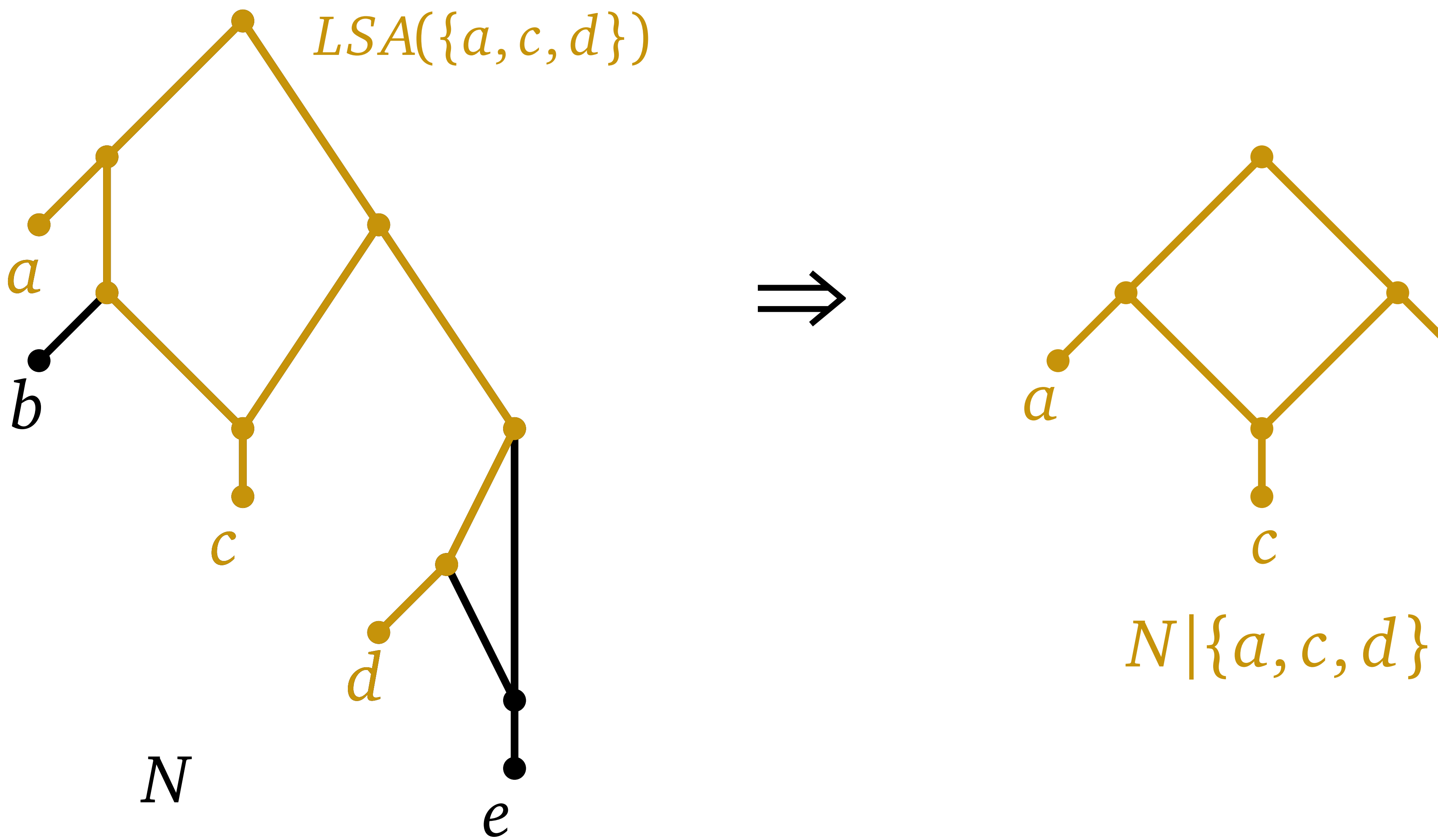


Trinets and Subnets

The ***lowest stable ancestor (LSA)*** of a set X' of taxa is the last vertex that is on all paths from the root to a leaf in X'.

The ***subnet*** $N|X'$ is obtained by

1. taking all directed paths in N from $LSA(X')$ to a leaf in X';
2. suppressing indegree-1 outdegree-1 vertices and parallel arcs.

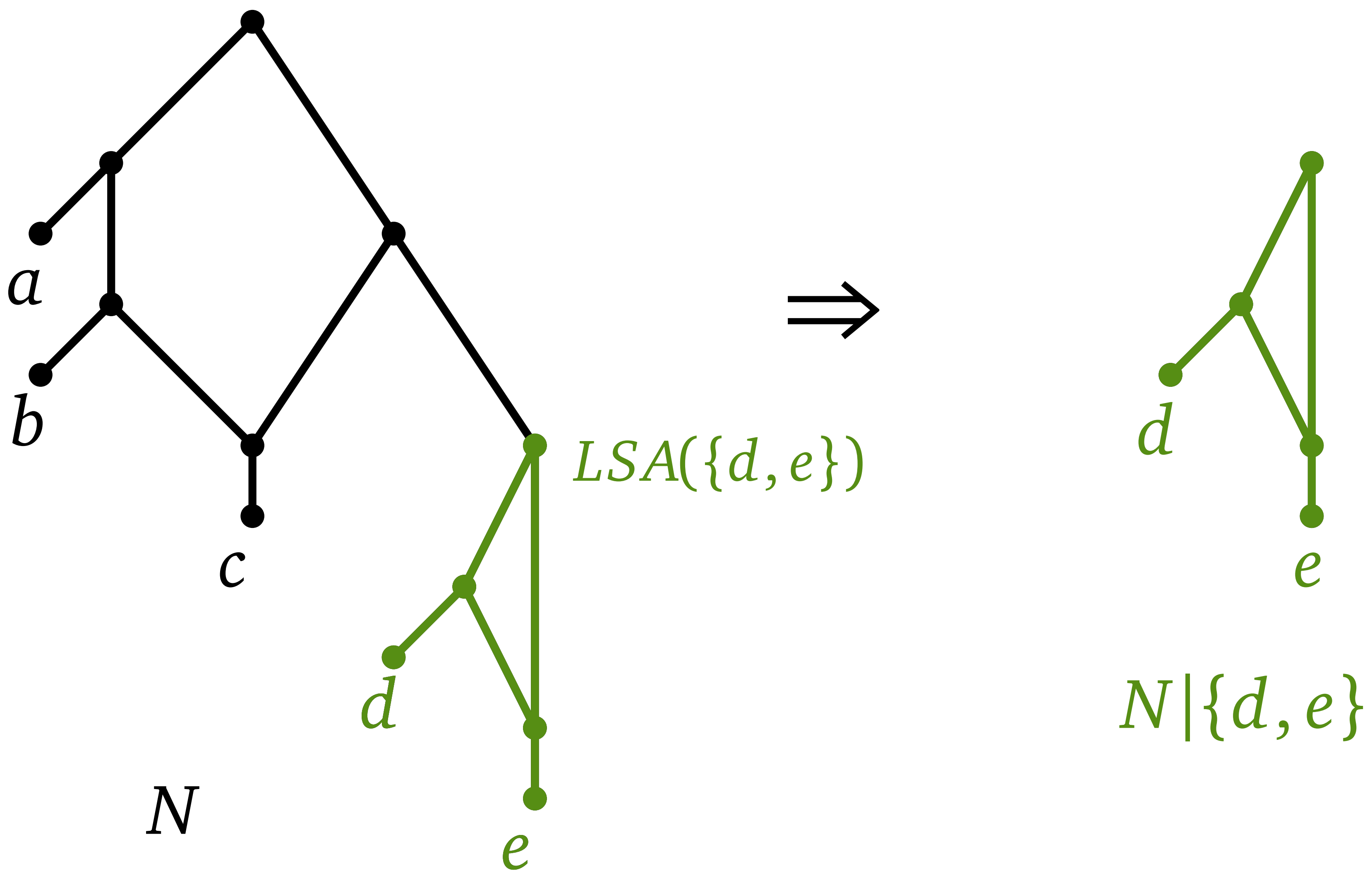


Trinets and Subnets

The ***lowest stable ancestor (LSA)*** of a set X' of taxa is the last vertex that is on all paths from the root to a leaf in X'.

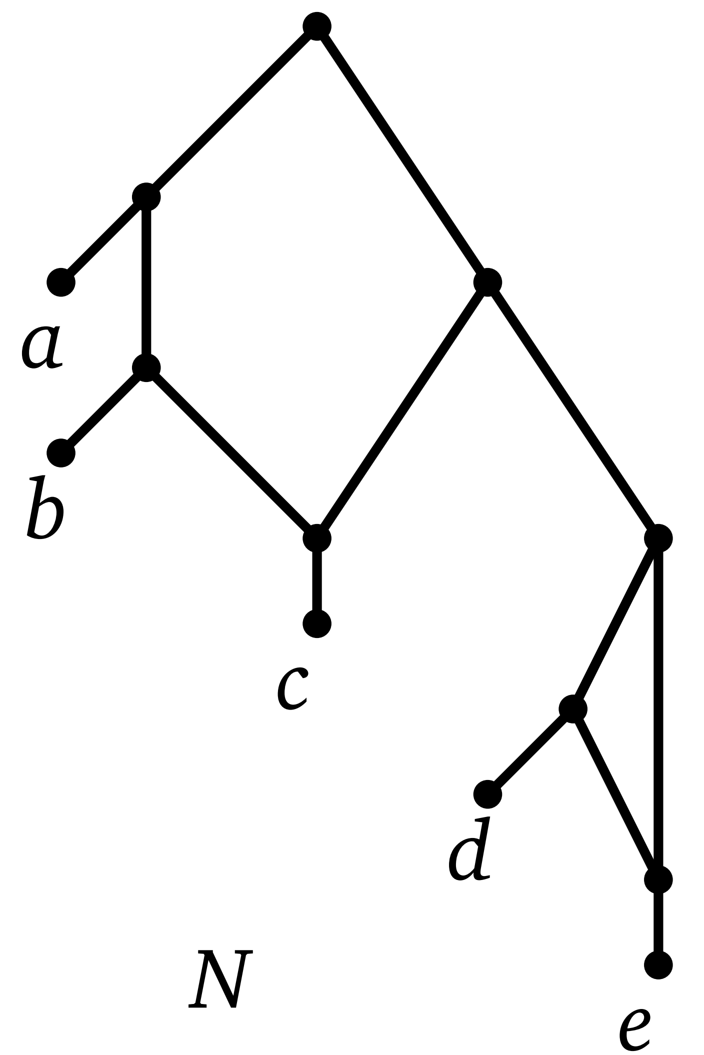
The ***subnet*** $N|X'$ is obtained by

1. taking all directed paths in N from $LSA(X')$ to a leaf in X';
2. suppressing indegree-1 outdegree-1 vertices and parallel arcs.

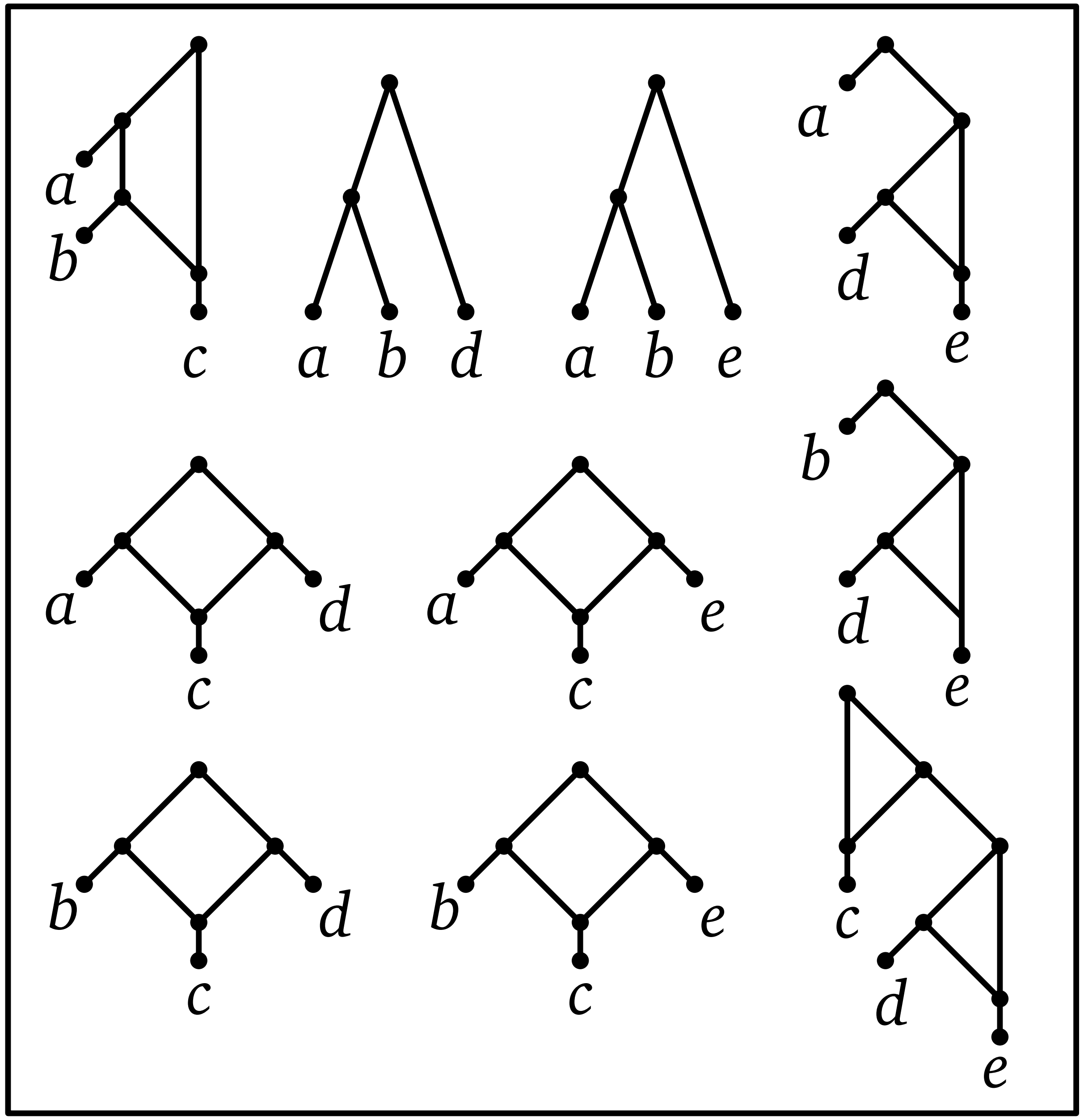


Trinets and Subnets

A ***trinet*** is a subnet with 3 leaves.
A ***binet*** is a subnet with 2 leaves.



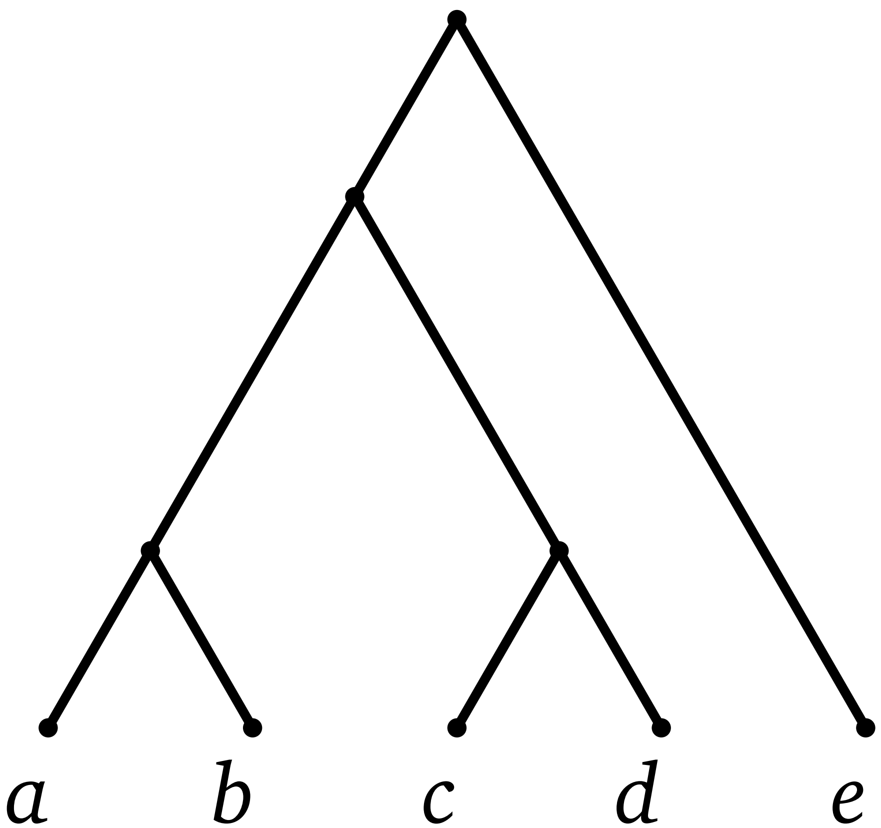
$\Rightarrow$



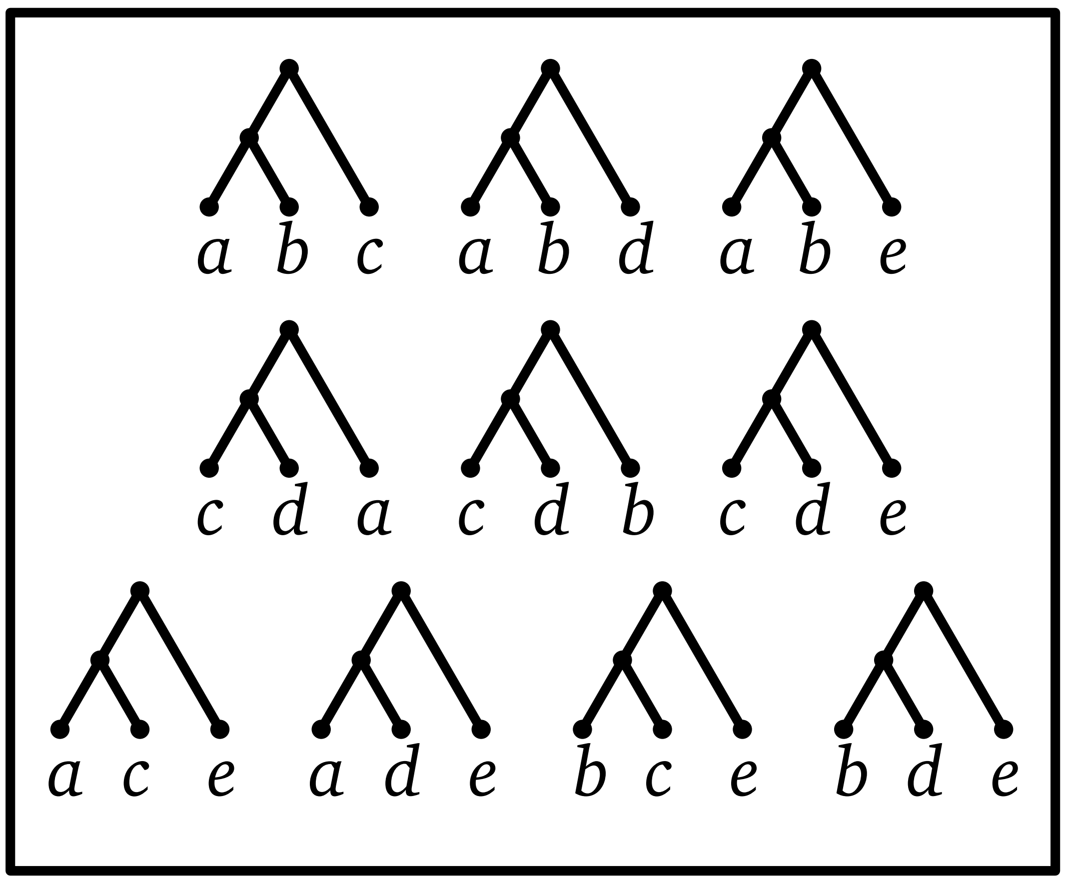
all trinets of N

Encoding Trees

Trees are encoded by their ***triplets***.



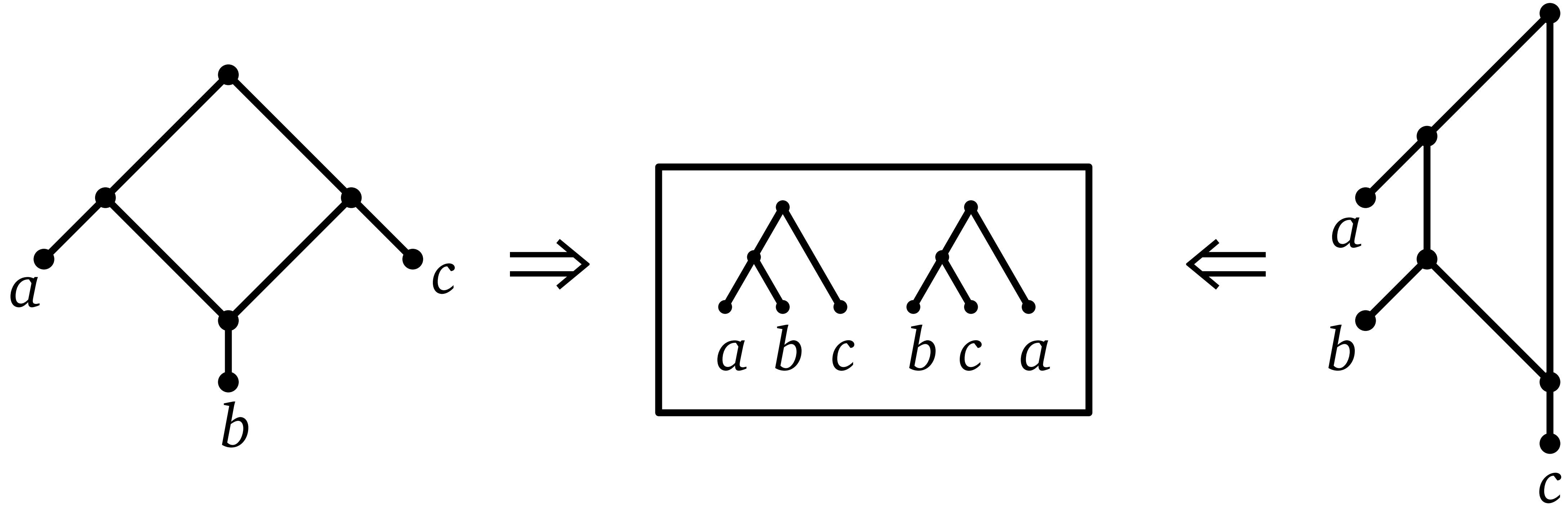
$\Longleftrightarrow$



Encoding Networks

Trees are encoded by their ***triplets***.

Networks are ***not*** encoded by their ***triplets***.

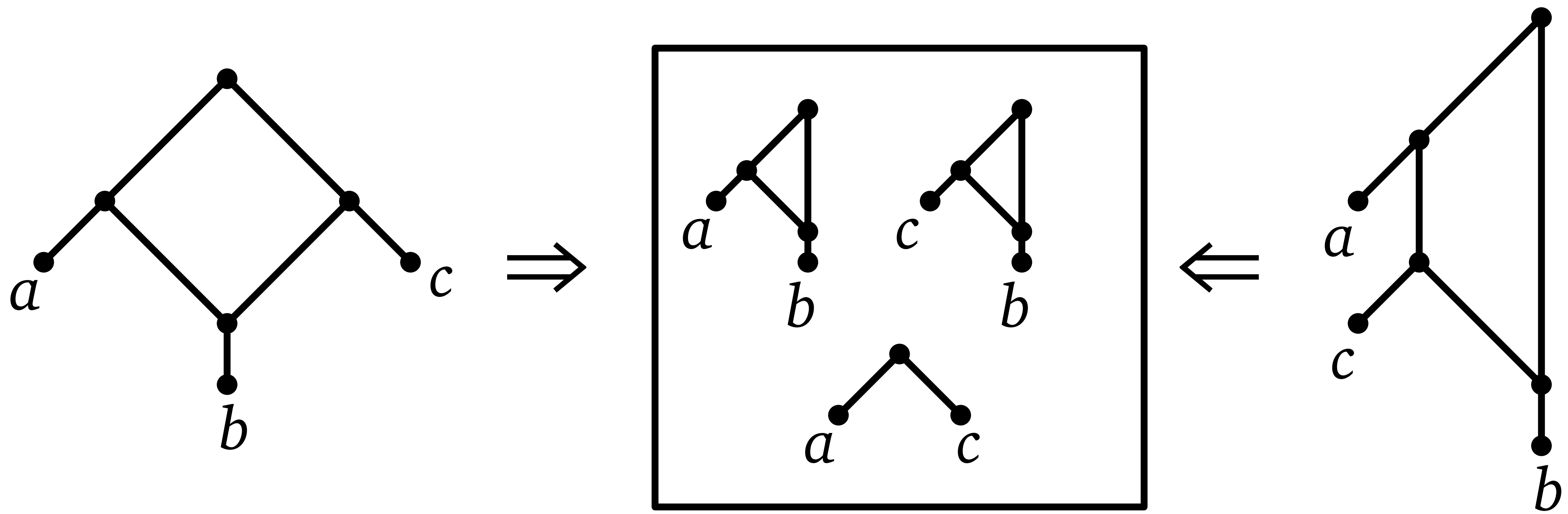


Encoding Networks

Trees are encoded by their ***triplets***.

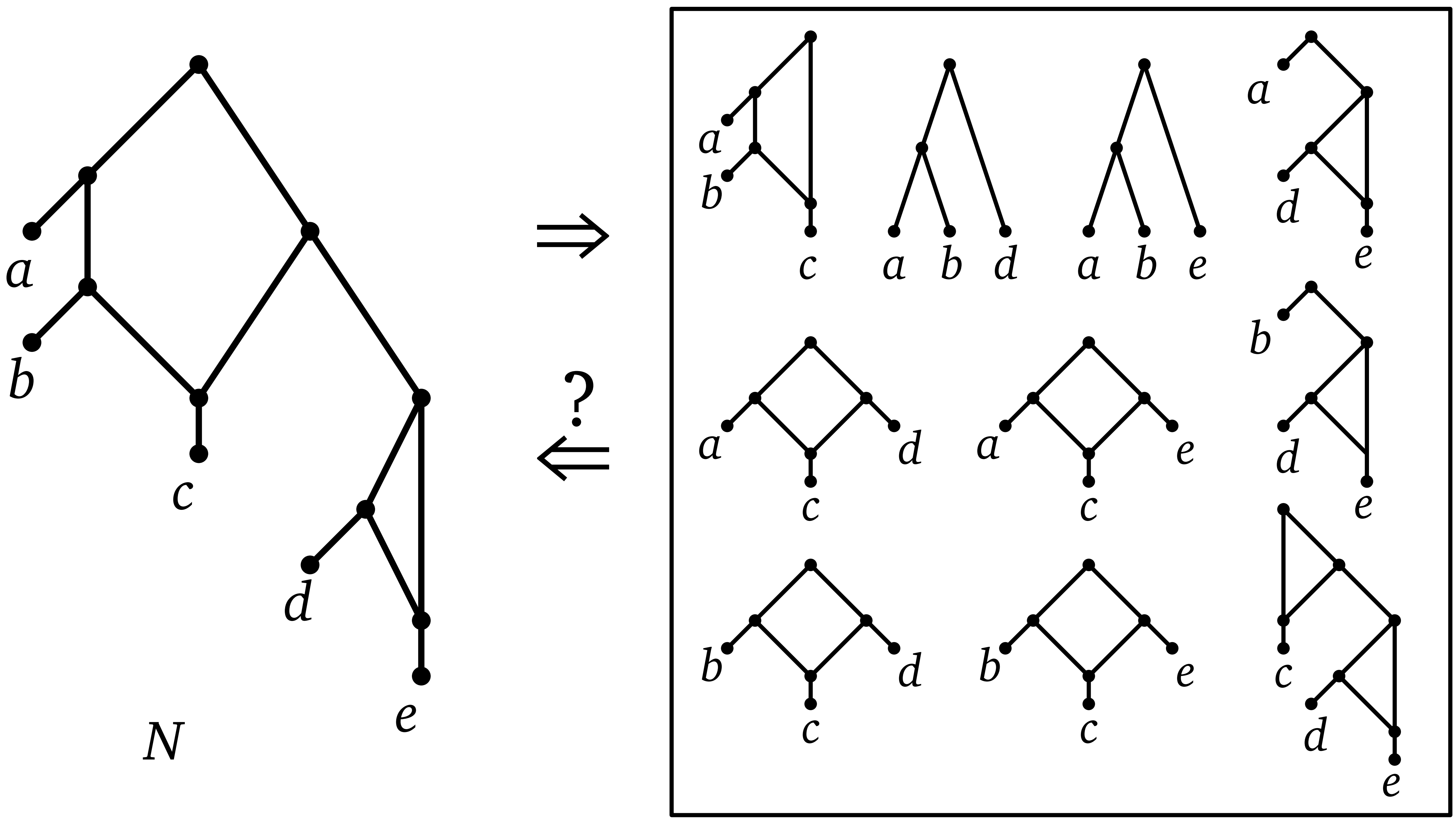
Networks are ***not*** encoded by their ***triplets***.

Networks are also ***not*** encoded by their ***binets***.



Encoding Networks

Are ***networks*** encoded by their ***trinets***?

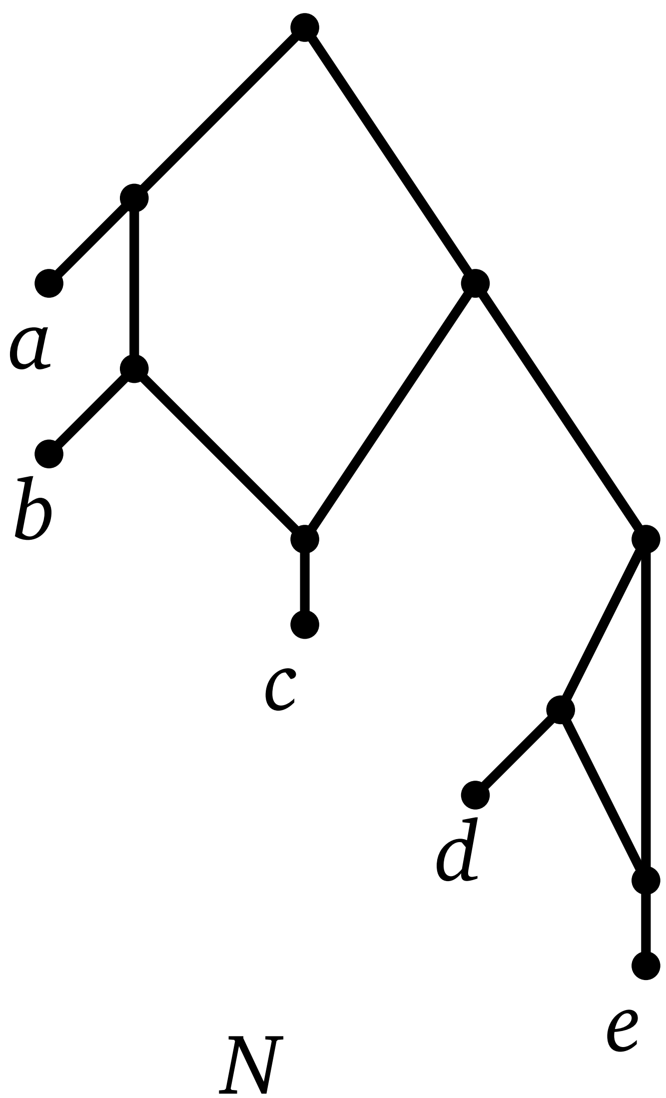


all trinets of N

Encoding Networks

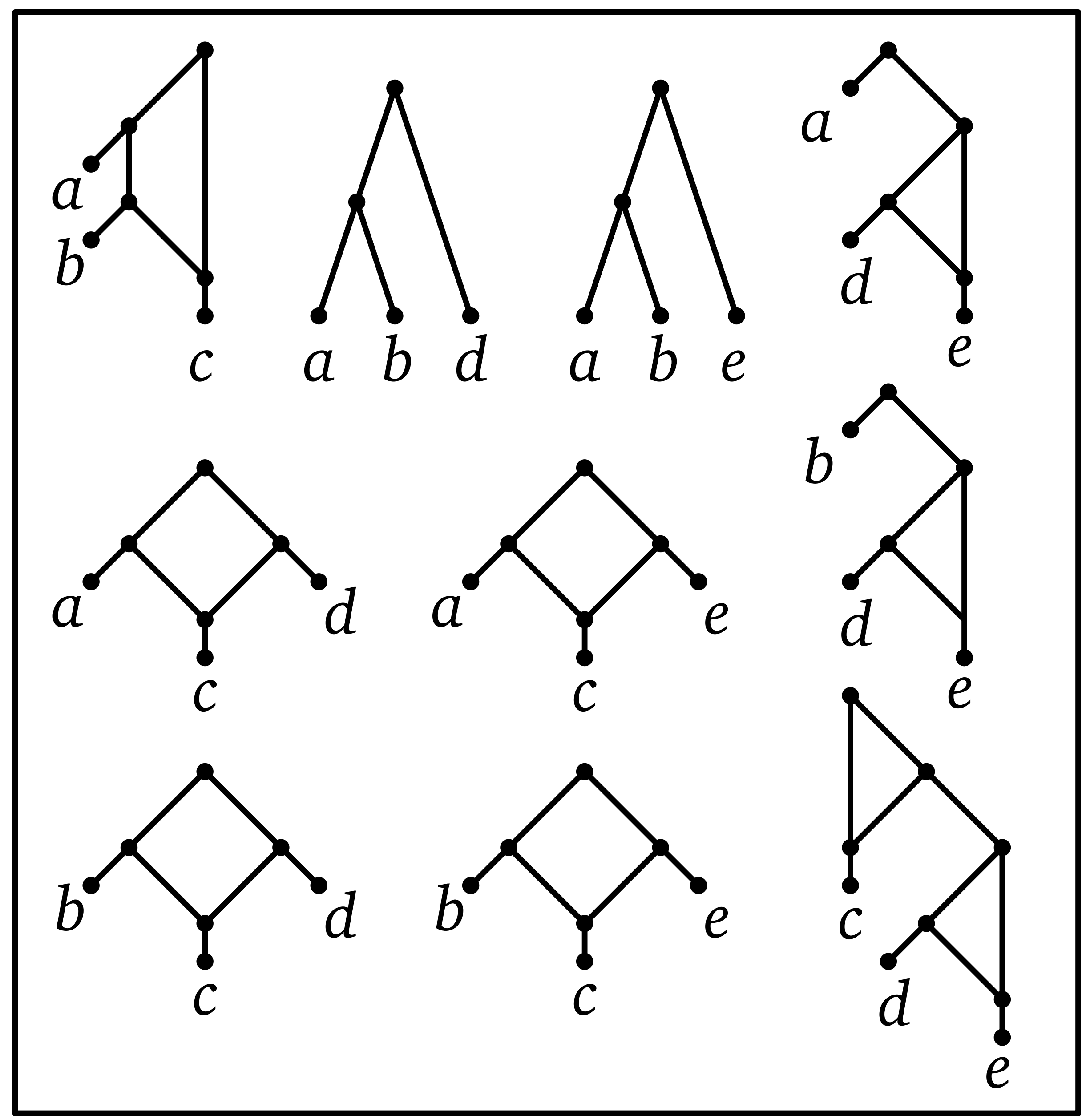
Are ***networks*** encoded by their ***trinets***?

No! (see talk by Taoyang)



$\Rightarrow$

? $\Leftarrow$

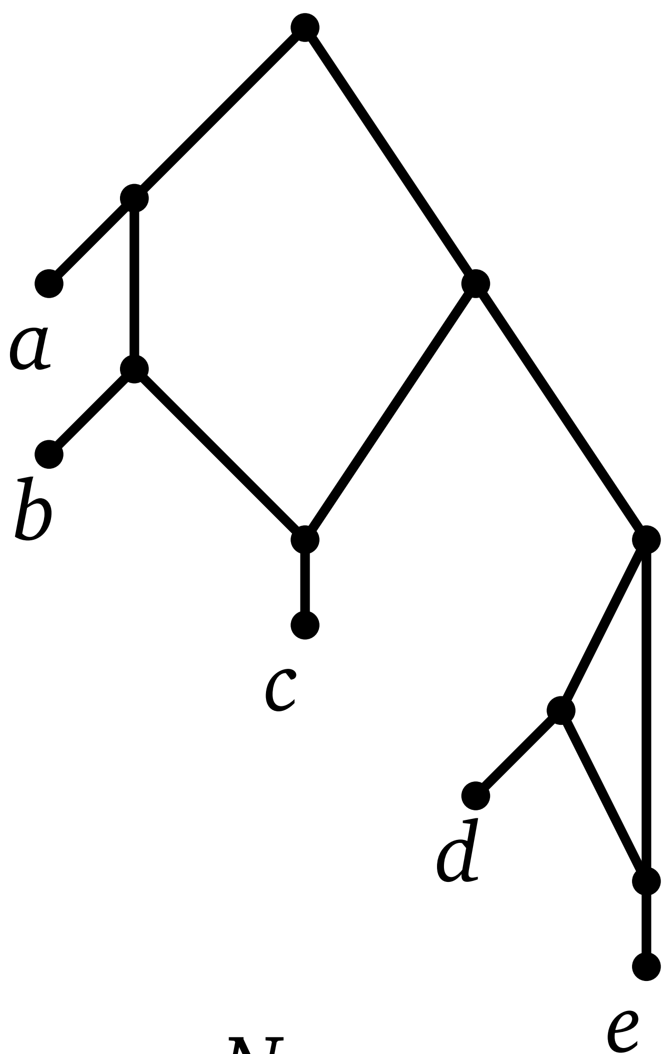


all trinets of N

Encoding Networks

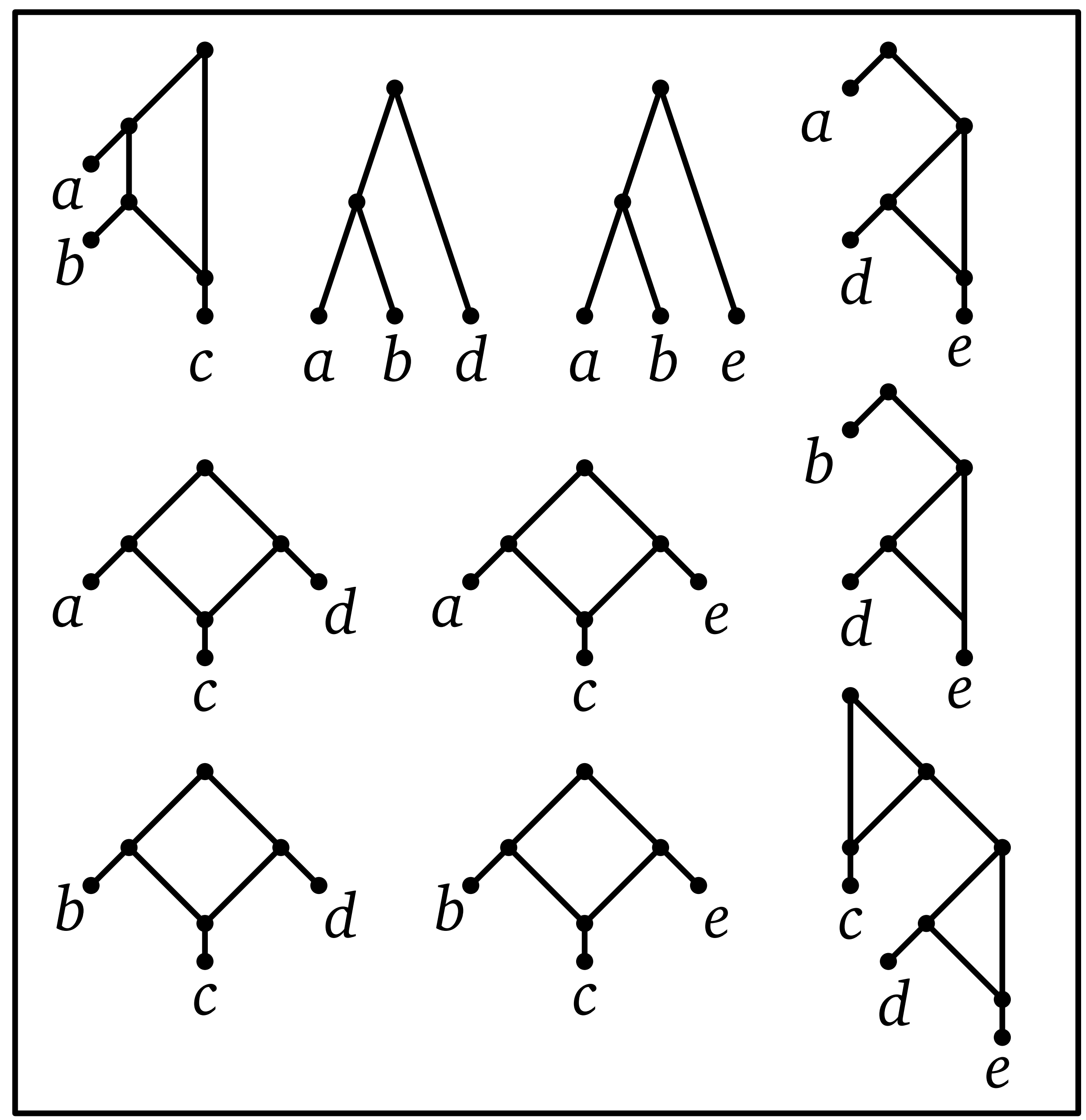
Are ***networks*** encoded by their ***trinets***?

No! (see talk by Taoyang)
but in certain cases yes!



N

$\Longleftrightarrow$

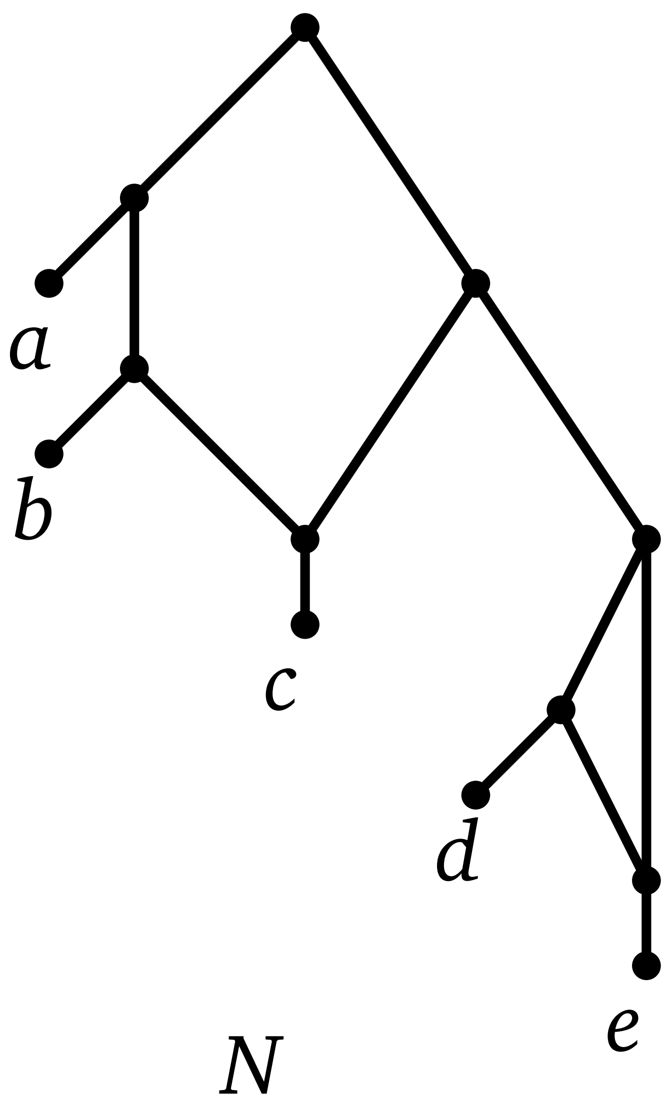


all trinets of N

Encoding Networks

Are ***networks*** encoded by their ***trinets***?

No! (see talk by Taoyang)
but in certain cases yes!



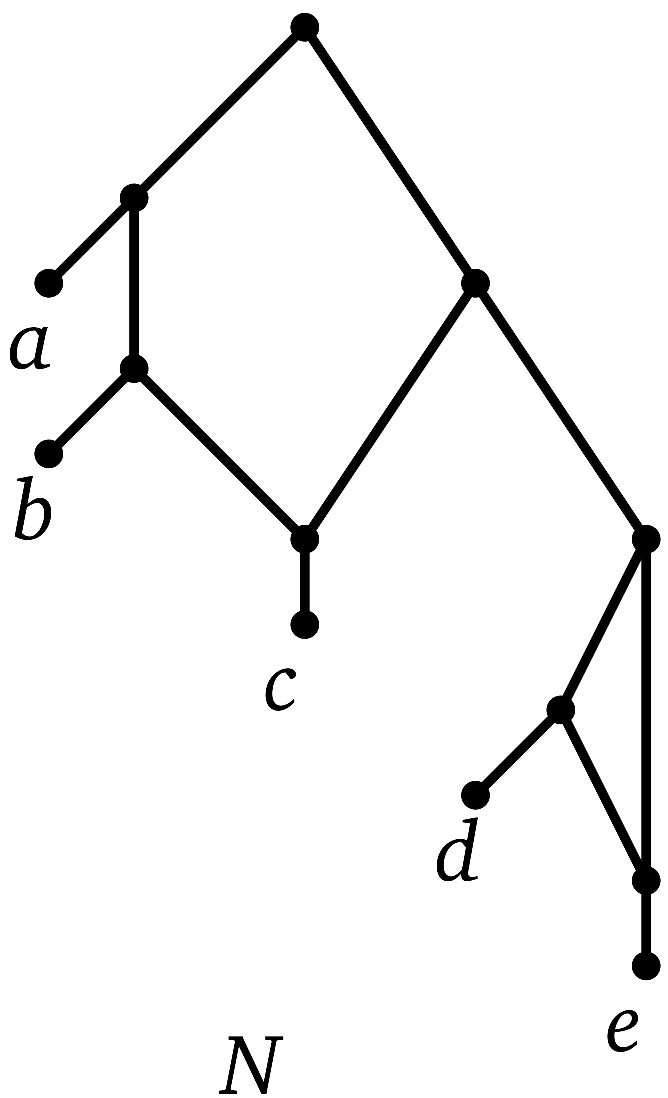
Definition. (assuming binary reticulations)

- *level*-k: at most k reticulations per biconnected component;
- *tree-child*: each non-leaf vertex has a child that is not a reticulation.

Encoding Networks

Are ***networks*** encoded by their ***trinets***?

No! (see talk by Taoyang)
but in certain cases yes!



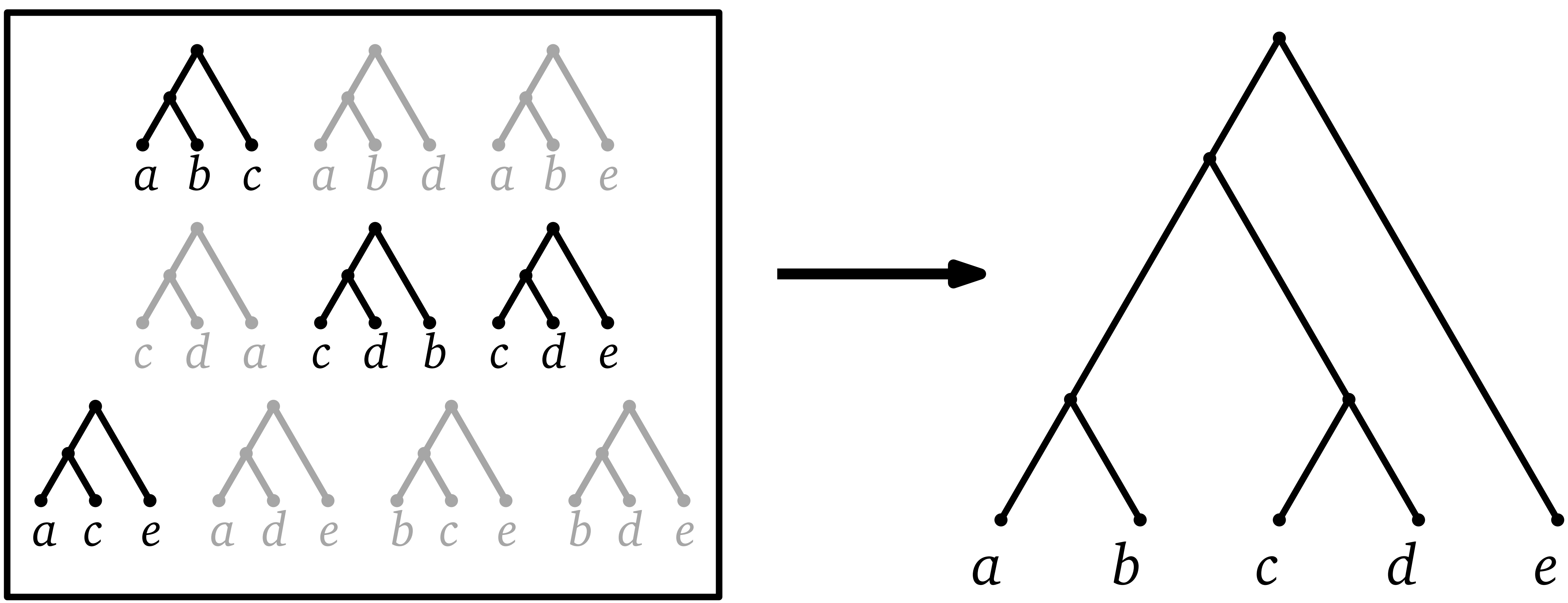
Definition. (assuming binary reticulations)

- *level*-k: at most k reticulations per biconnected component;
- ***tree-child***: each non-leaf vertex has a child that is not a reticulation.

Theorem. (Huber, vI & Moulton)
Binary level-1, level-2 and tree-child networks are all encoded by their trinets.

Reconstruction Algorithms

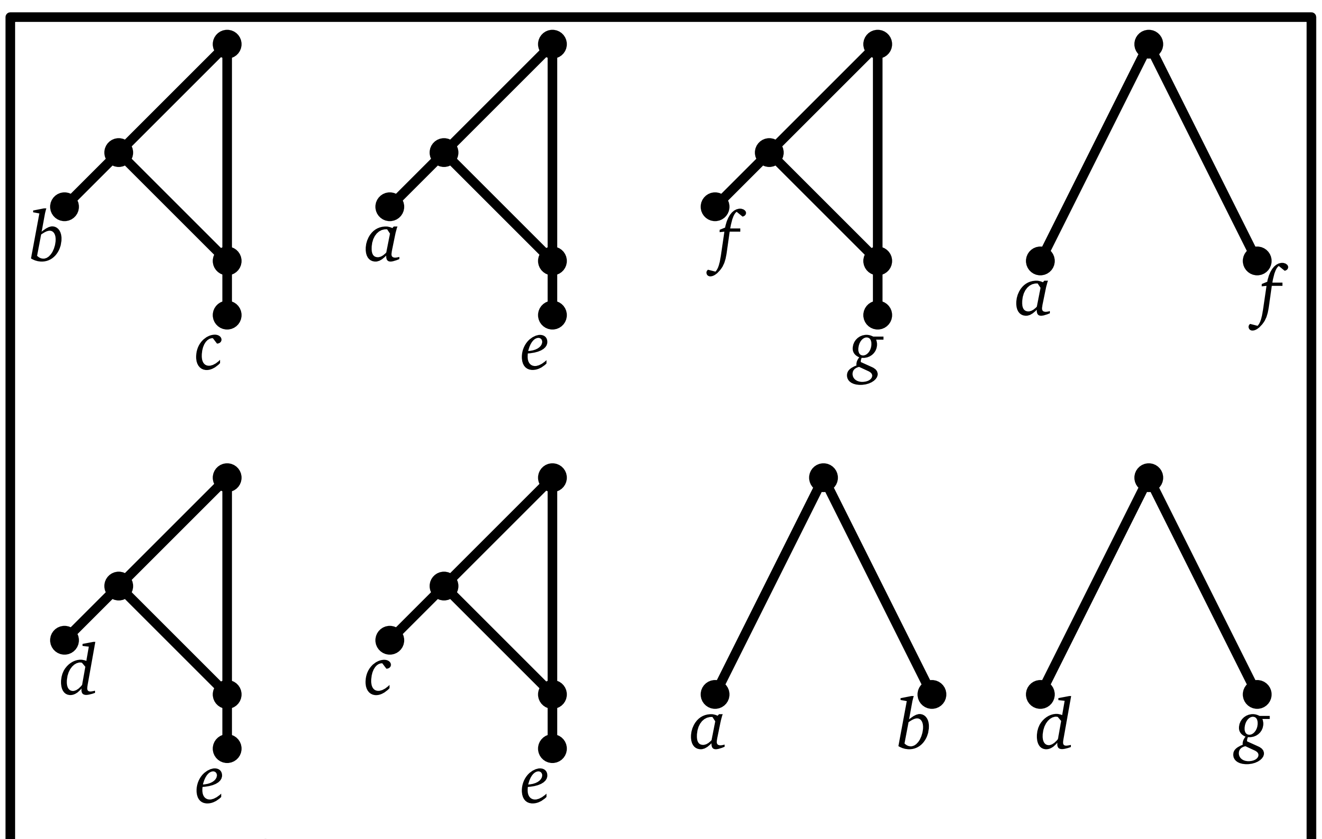
Given any set of triplets, we can construct a tree displaying them, if one exists, in polynomial time. (Aho, Sagiv, Szymanski, Ullman, 1981)



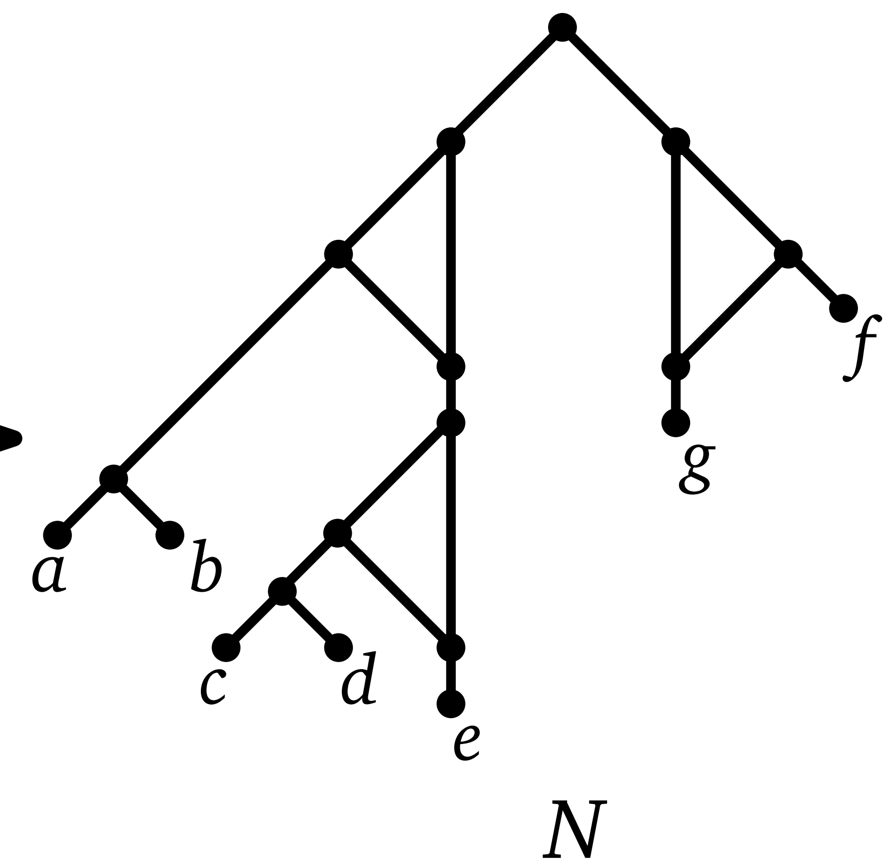
Reconstruction Algorithms

Given any set of triplets, we can construct a tree displaying them, if one exists, in polynomial time. (Aho, Sagiv, Szymanski, Ullman, 1981)

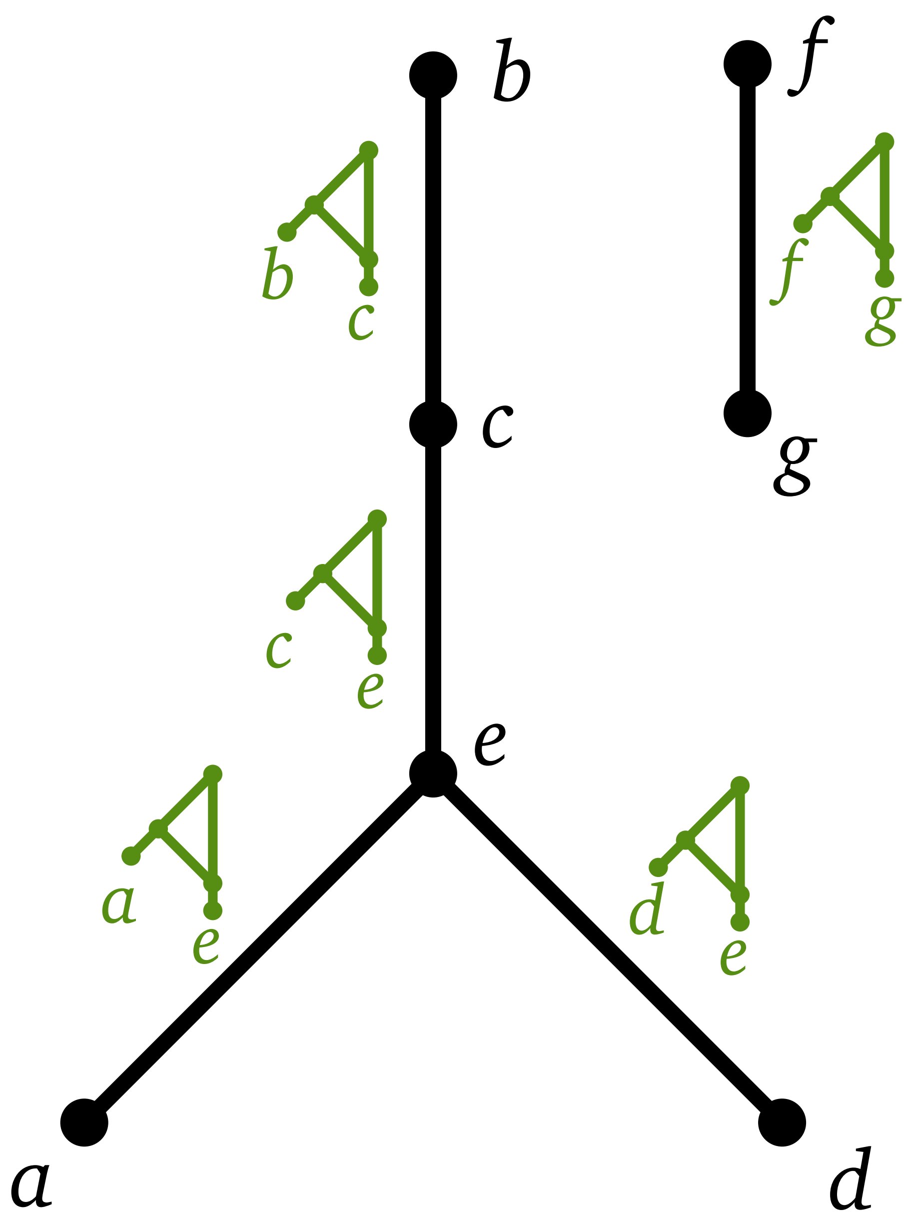
Given any set of ***binets*** we can construct a ***level-1 network*** displaying them, if one exists, in polynomial time.
(Huber, vI, Moulton, Scornavacca, Wu, 2015)

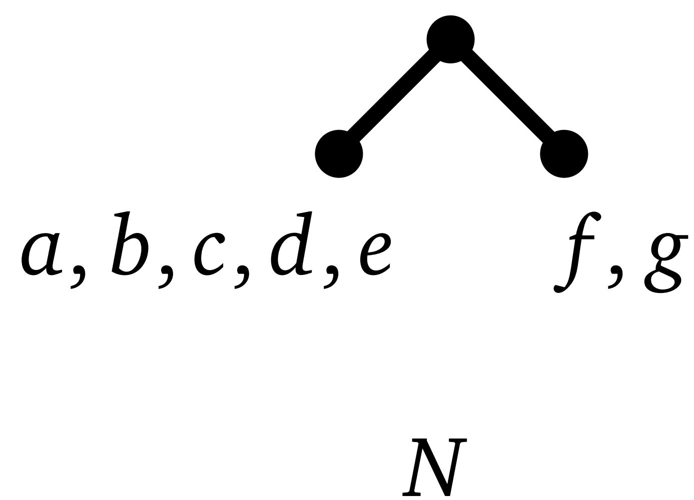


8 of the 21 binets displayed by N



Step 1: can the network have a root that is not in a cycle?

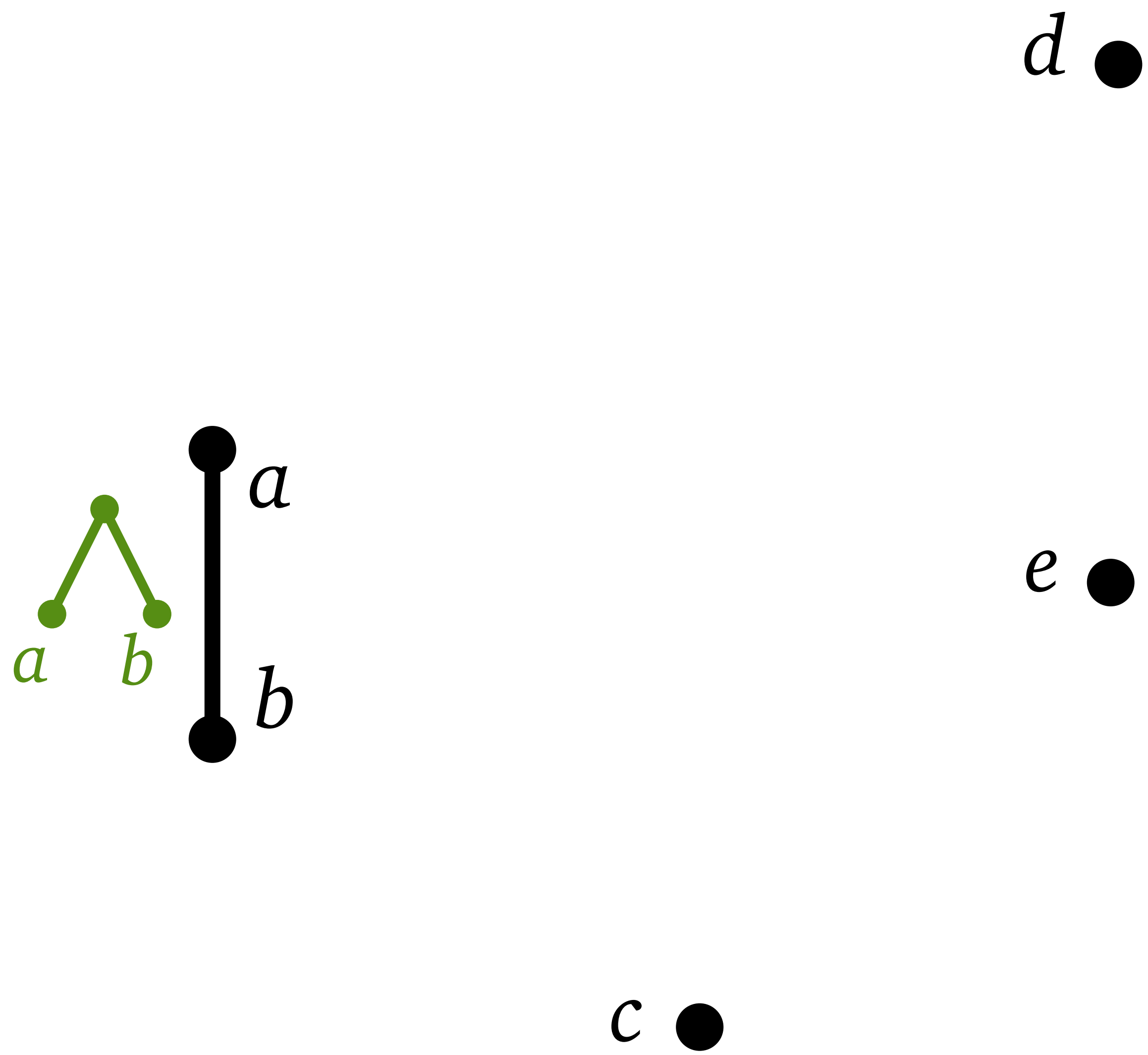




Graph $\mathscr{R}$ connects taxa that have to be on the same side of the root.

Reconstruction Algorithm for Binets

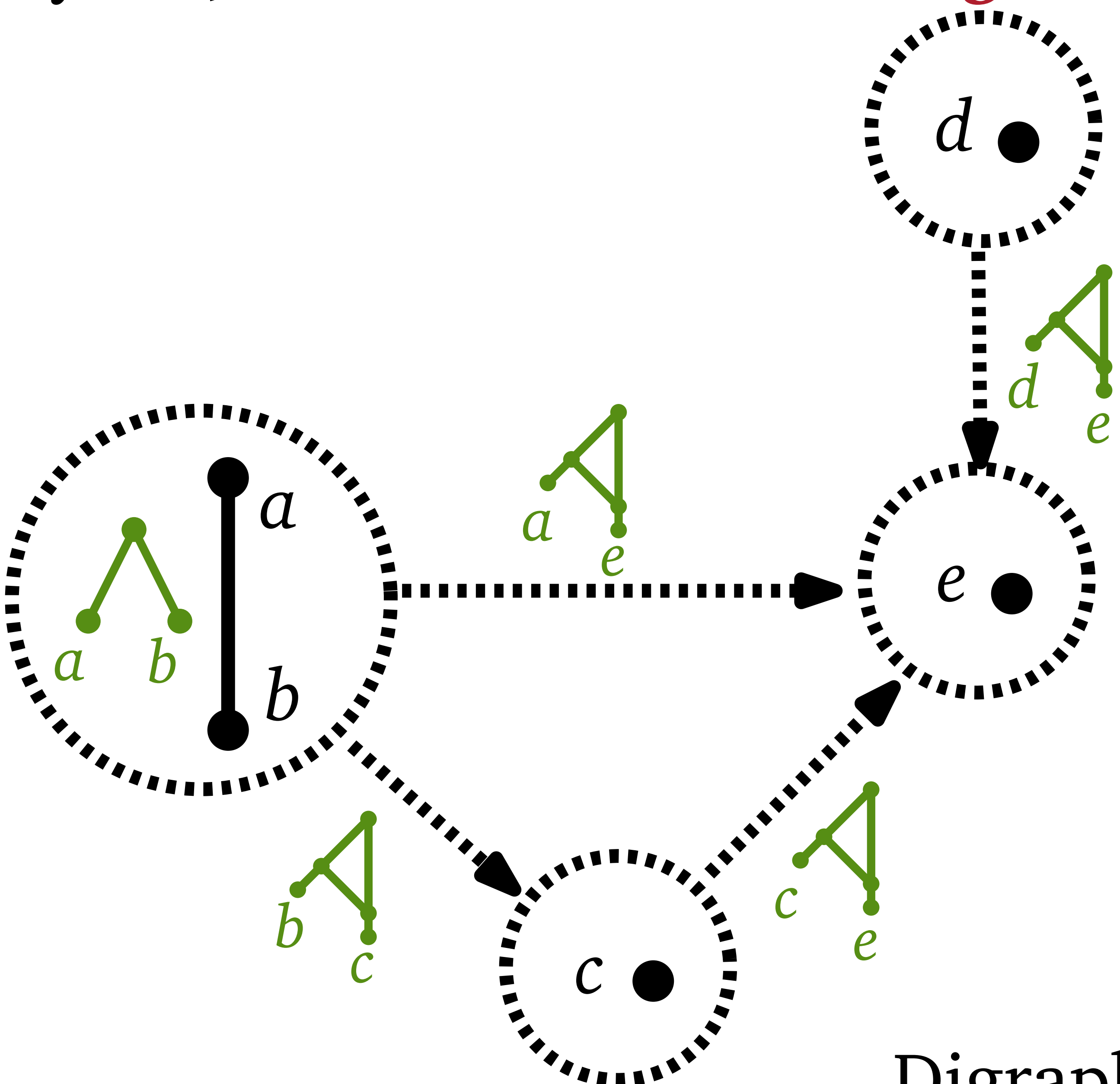
Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?



Graph $\mathscr{K}$ connects taxa that have to be ***at the same height***.

Reconstruction Algorithm for Binets

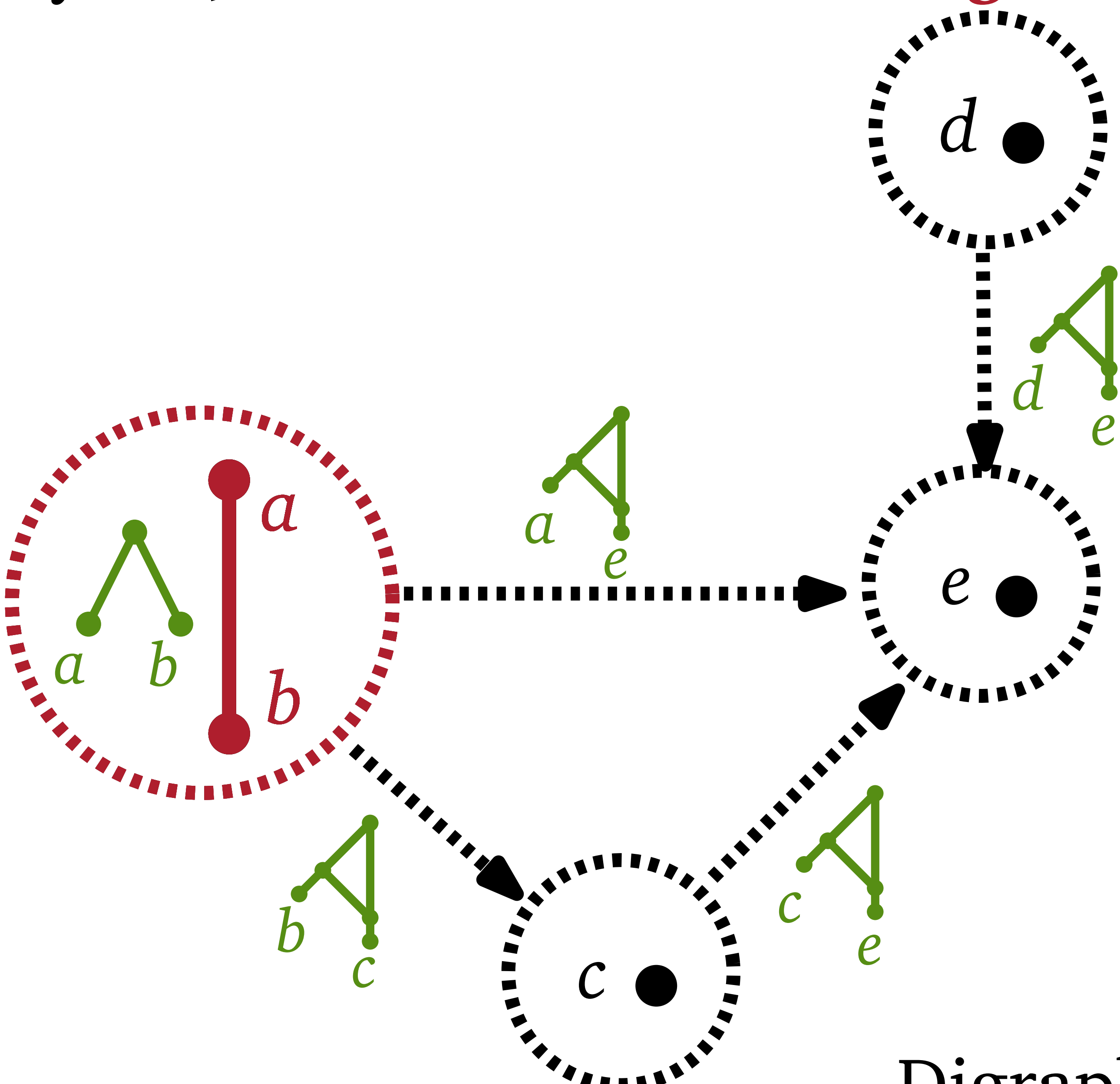
Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?



Digraph Ω has arcs indicating which taxa have to be ***above*** other taxa.

Reconstruction Algorithm for Binets

Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?

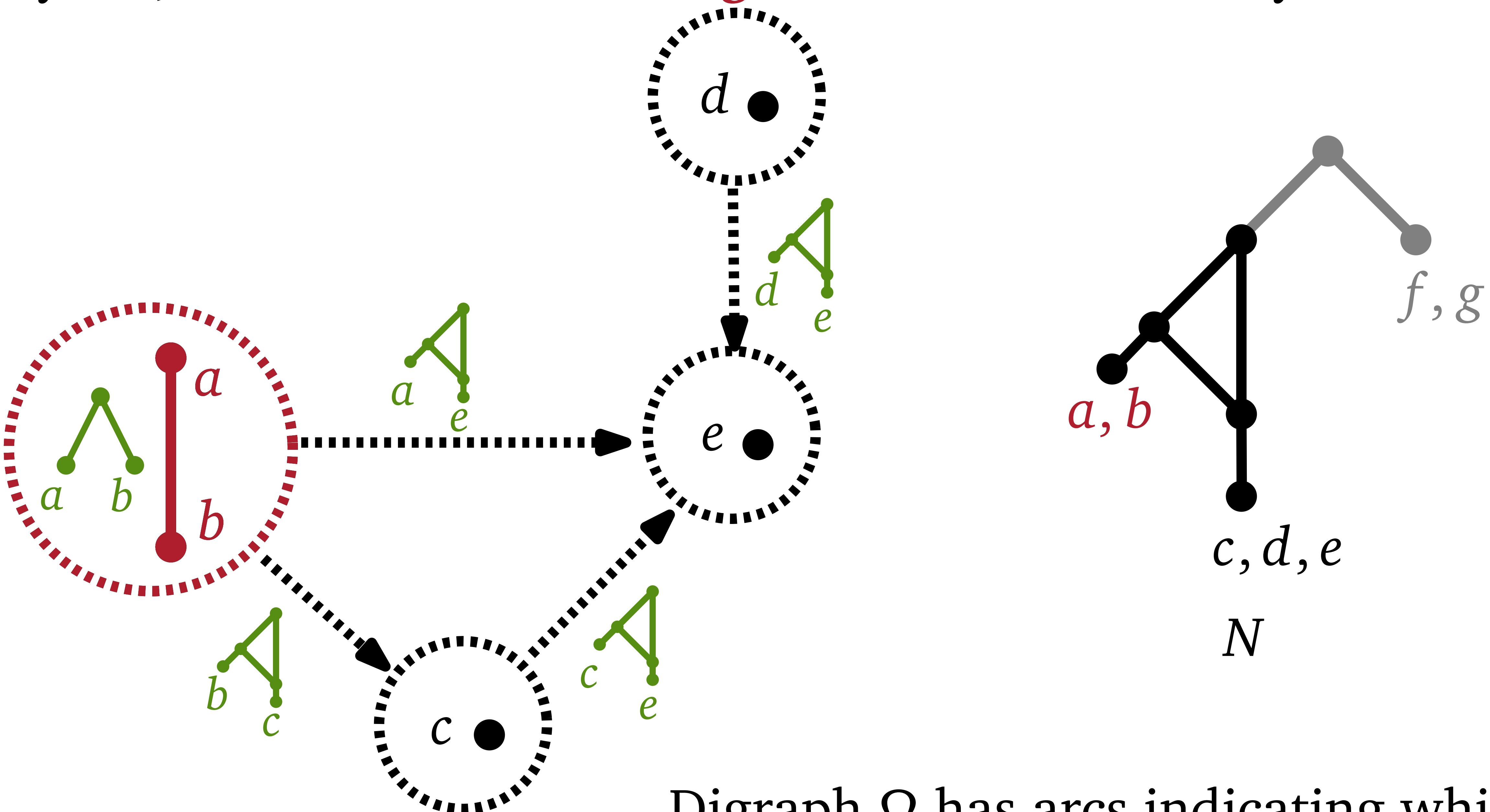


Find a proper subset of the vertices with no incoming arcs. Make those taxa "high".

Digraph Ω has arcs indicating which taxa have to be ***above*** other taxa.

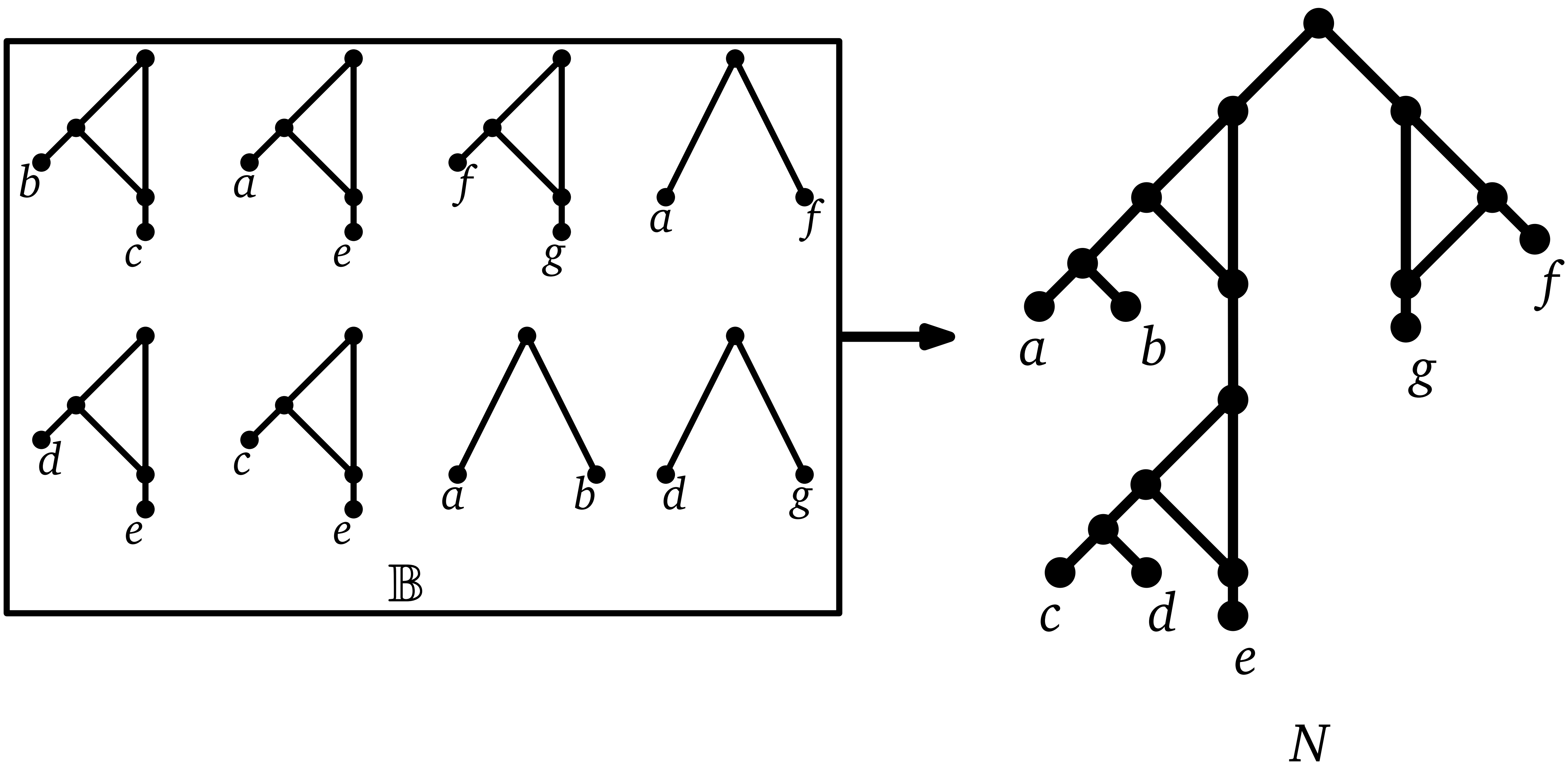
Reconstruction Algorithm for Binets

Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?



Digraph Ω has arcs indicating which taxa have to be ***above*** other taxa.

Step 3: recursively find the sidenetworks



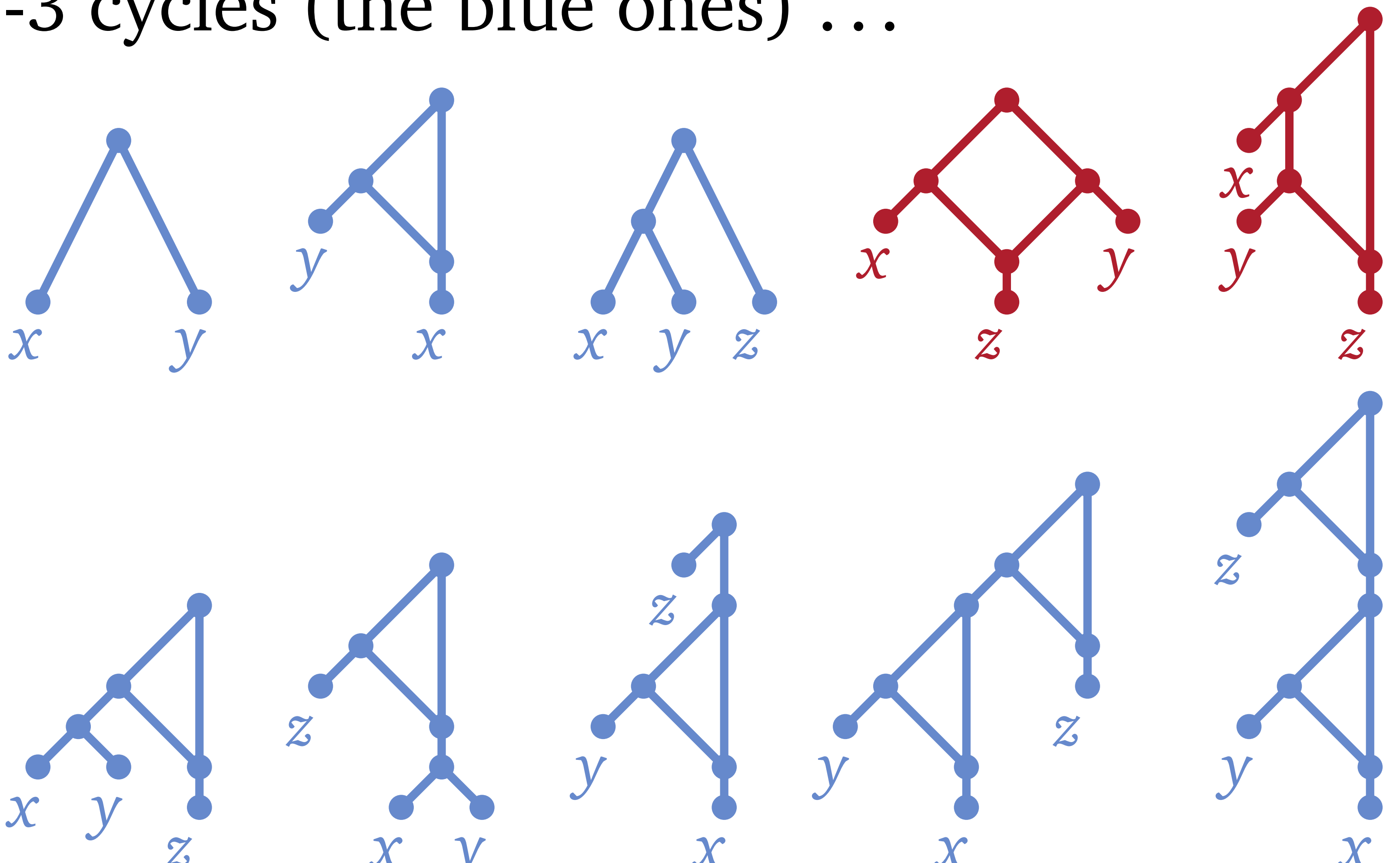
But what about trinets?

Theorem. (Huber, vI, Moulton, Scornavacca & Wu)
Given a set of ***trinets***, it is ***NP-hard*** to decide if there exists a binary level-1 network displaying them…

But what about trinets?

Theorem. (Huber, vI, Moulton, Scornavacca & Wu)
Given a set of ***trinets***, it is ***NP-hard*** to decide if there exists a binary level-1 network displaying them…

…but this problem is polynomial-time solvable for subnets with only size-3 cycles (the blue ones) …



But what about trinets?

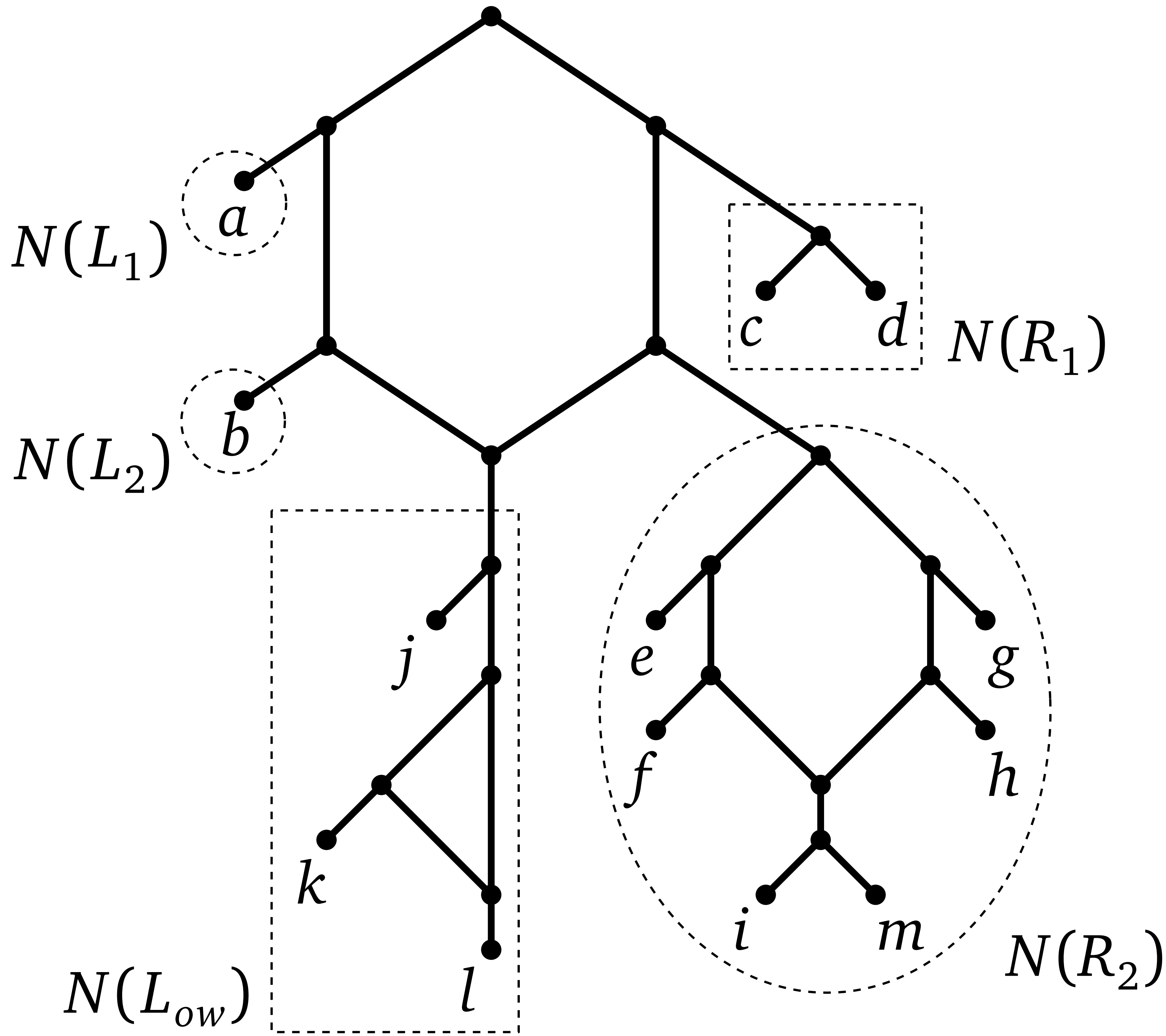
Theorem. (Huber, vI, Moulton, Scornavacca & Wu)
Given a set of ***trinets***, it is ***NP-hard*** to decide if there exists a binary level-1 network displaying them…

…but this problem is polynomial-time solvable for subnets with only size-3 cycles (the blue ones) …

…and there is an $O(3^n \text{poly}(n))$ time algorithm for the general case, with n taxa.

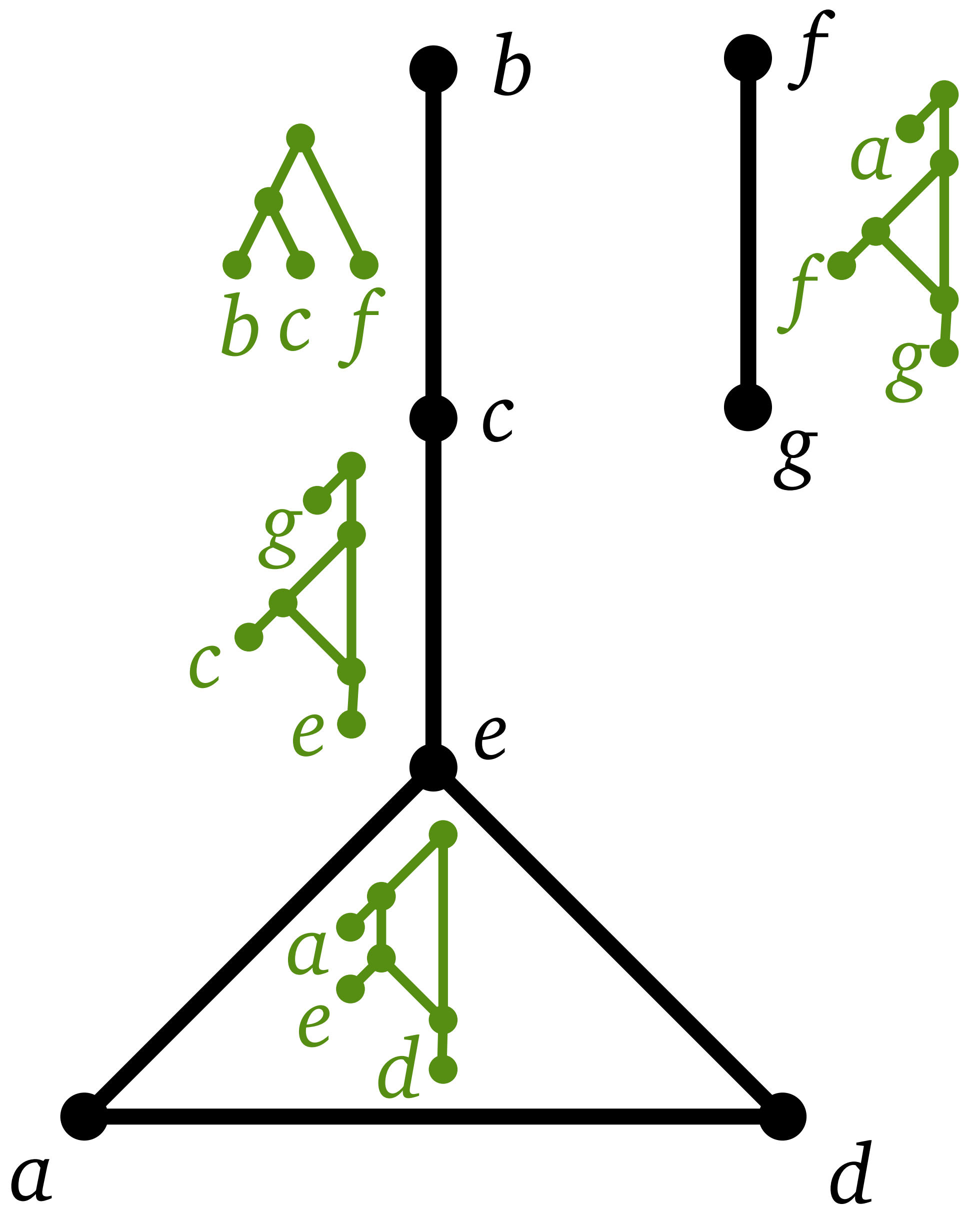
Reconstruction algorithm for trinets

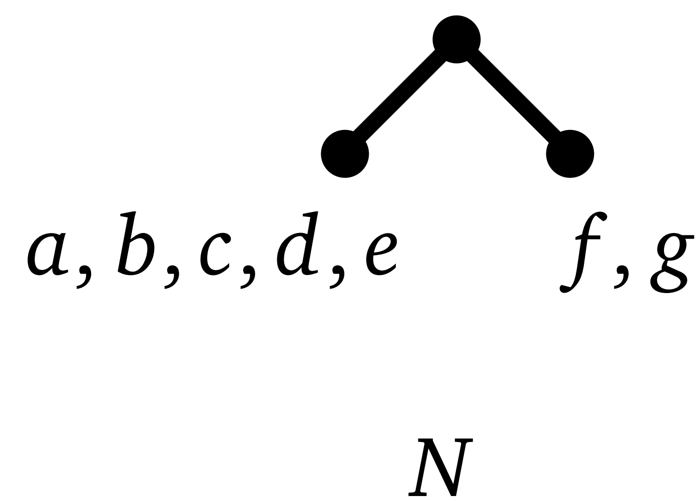
1. Is the root in a cycle?
2. Which taxa are high/low?
3. Which taxa are on the left/right?
4. Partition each side into sidenetworks.
5. Recursively find all sidenetworks.



Reconstruction algorithm for trinets

Step 1: can the network have a root that is not in a cycle?

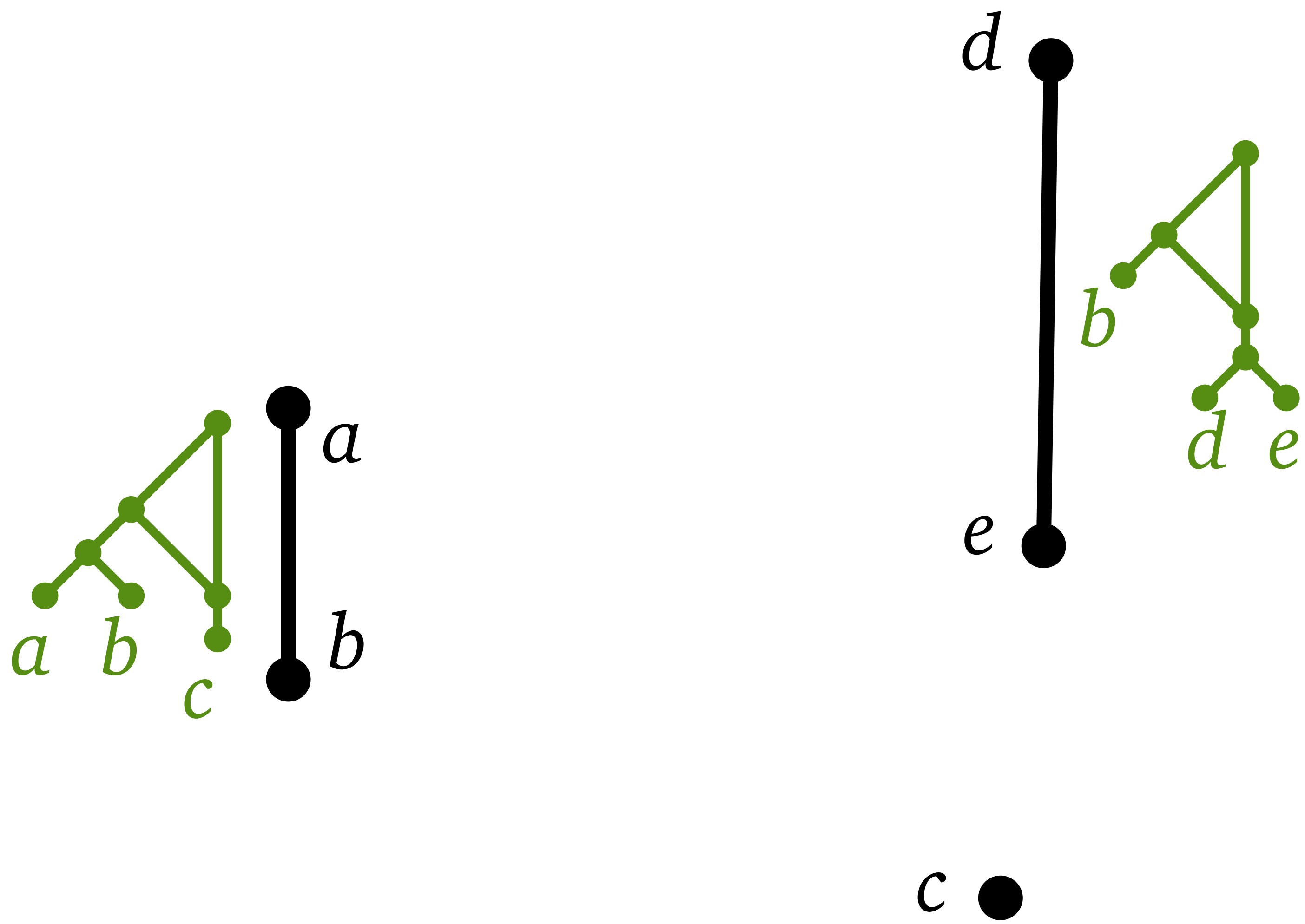




Graph $\mathscr{R}$ connects taxa that have to be on the ***same side of the root***.

Reconstruction Algorithm for Trinets

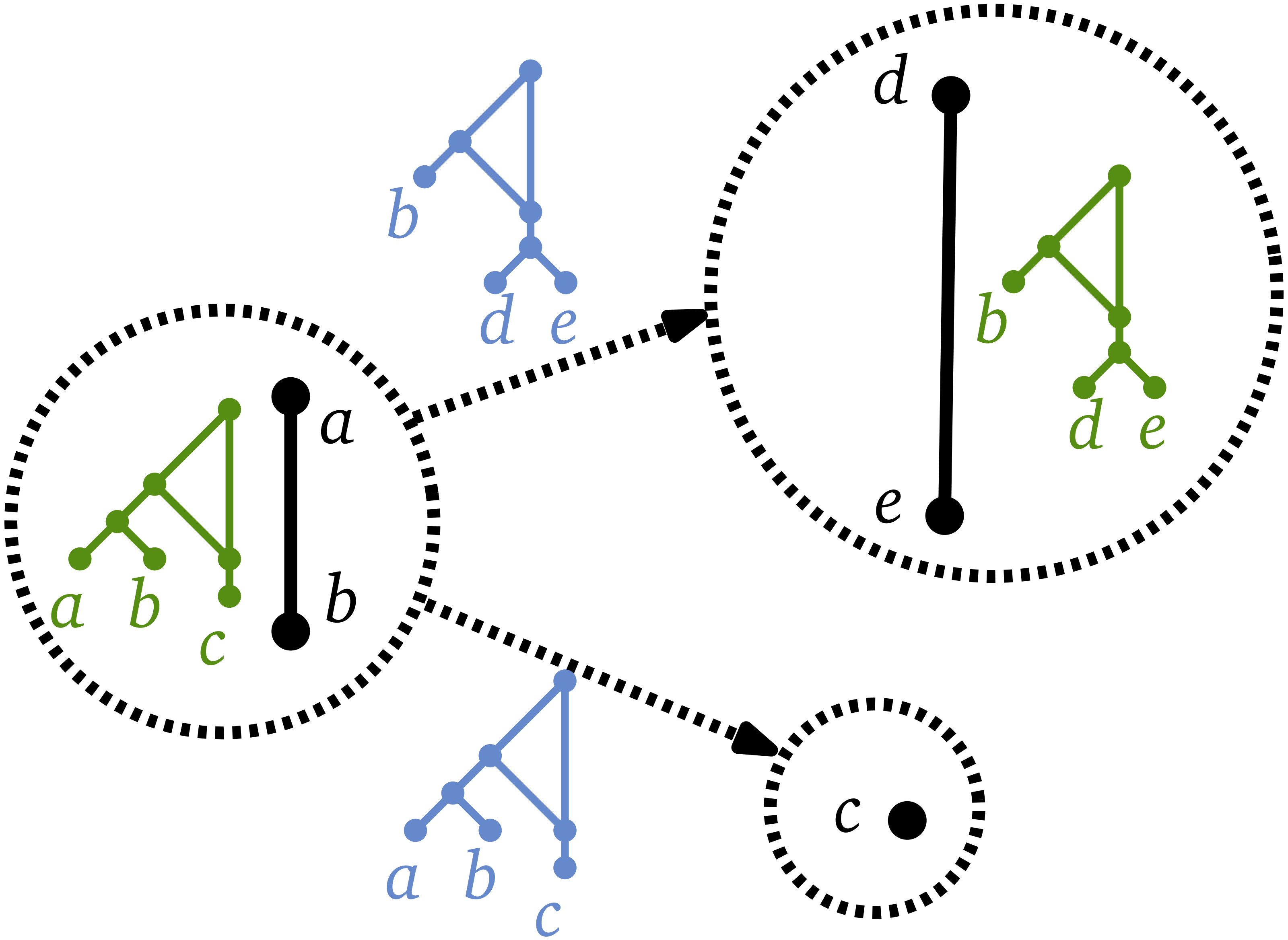
Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?



Graph $\mathscr{K}$ connects taxa that have to be ***at the same height***.

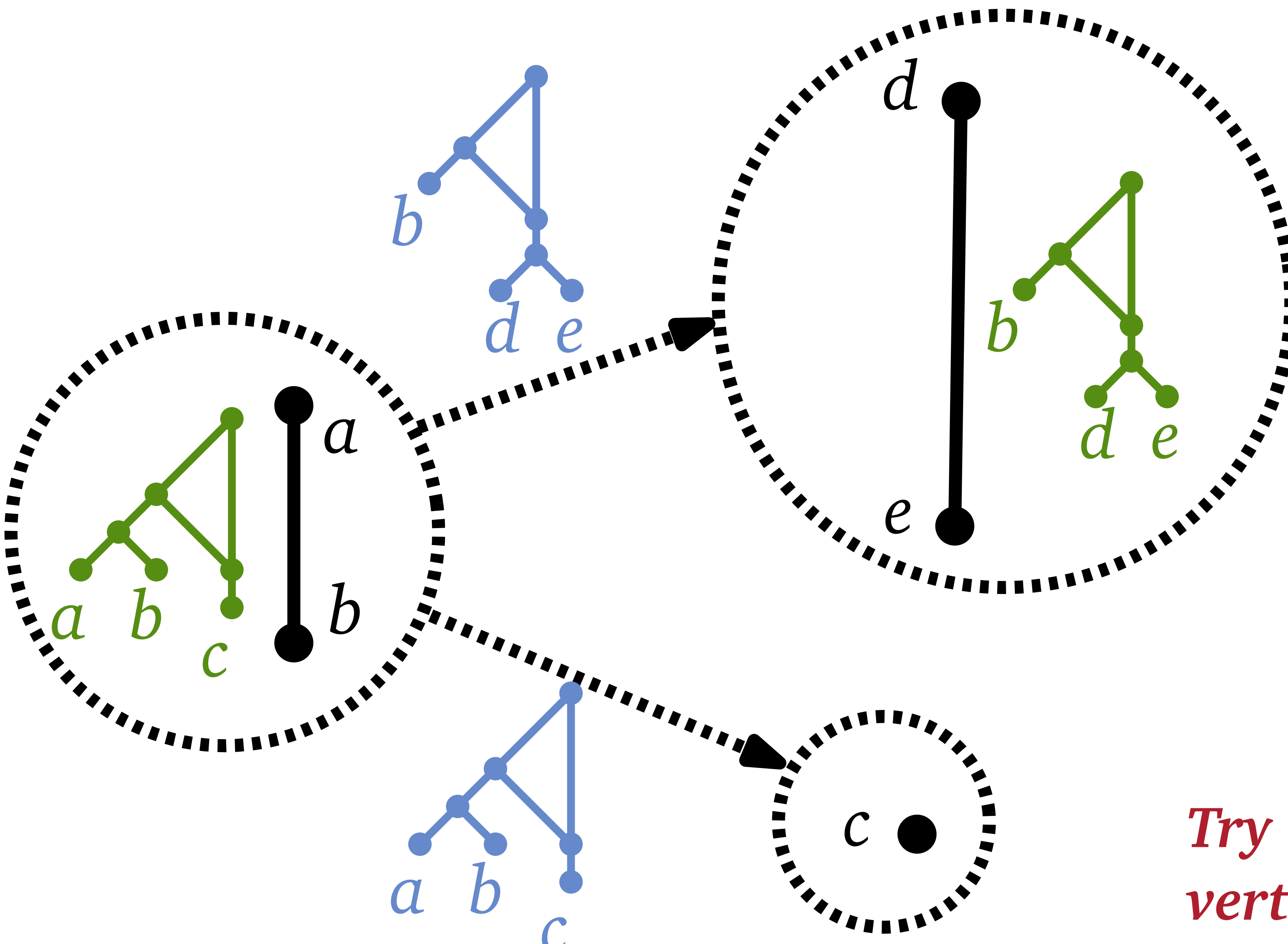
Reconstruction Algorithm for Trinets

Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?



Digraph Ω has arcs indicating which taxa have to be ***above*** other taxa.

Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?

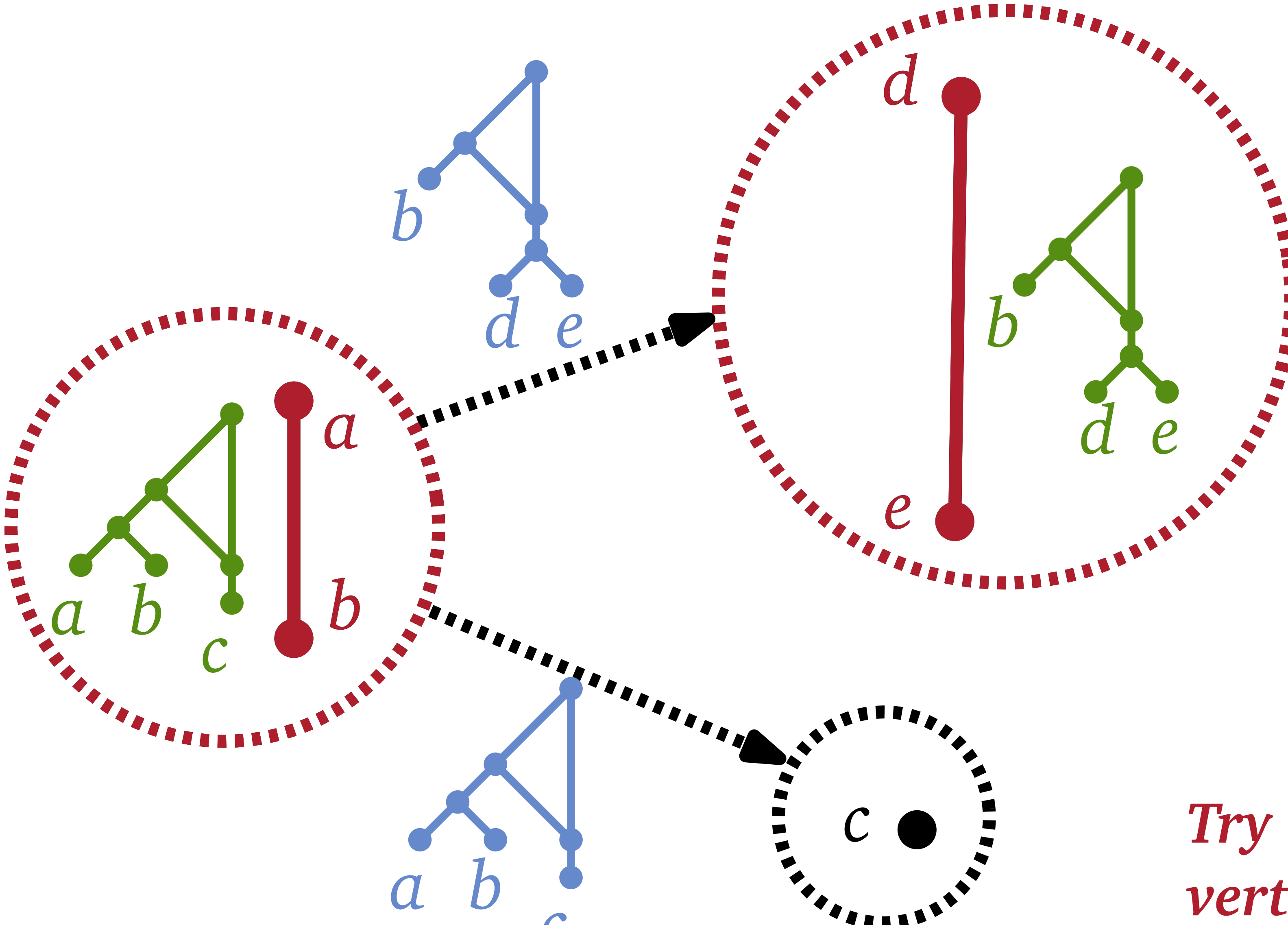


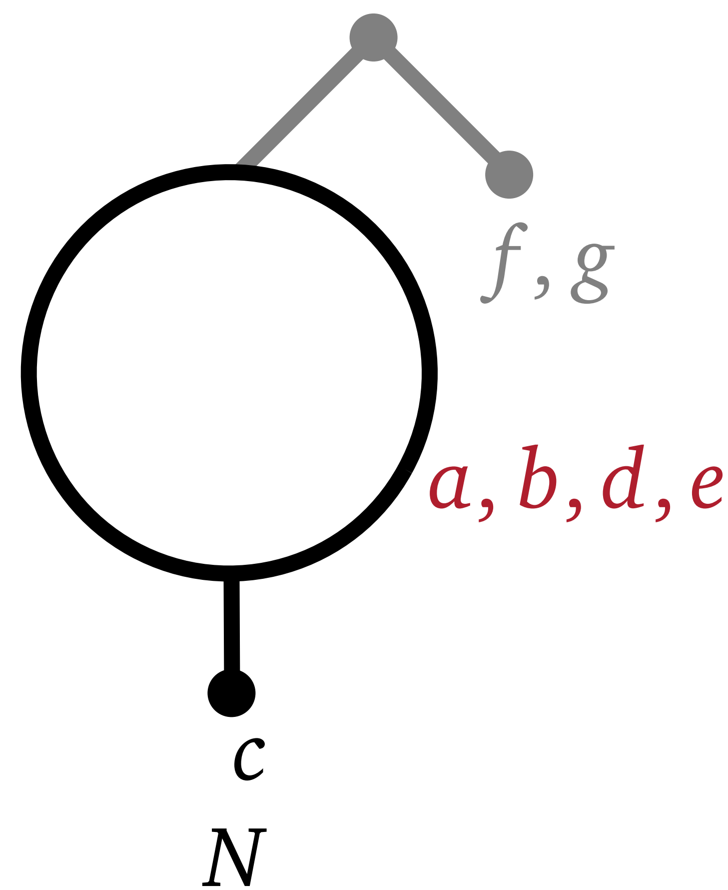
Try all proper subsets of the vertices with no incoming arcs. Make those taxa "high".

Digraph Ω has arcs indicating which taxa have to be ***above*** other taxa.

Reconstruction Algorithm for Trinets

Step 2: if the root is in a cycle, which taxa are ***low*** (below the cycle), and which ones are ***high*** (to the side of the cycle)?



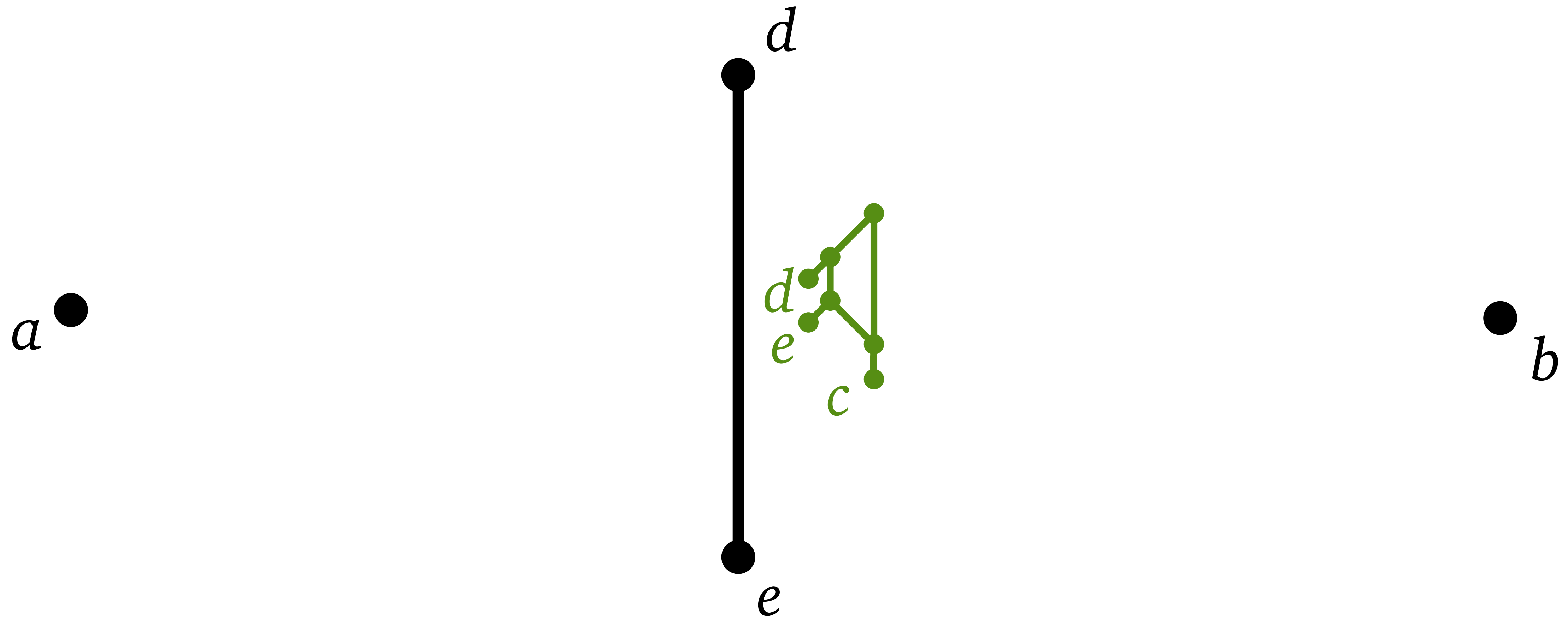


Try all proper subsets of the vertices with no incoming arcs. Make those taxa "high".

Digraph Ω has arcs indicating which taxa have to be ***above*** other taxa.

Reconstruction Algorithm for Trinets

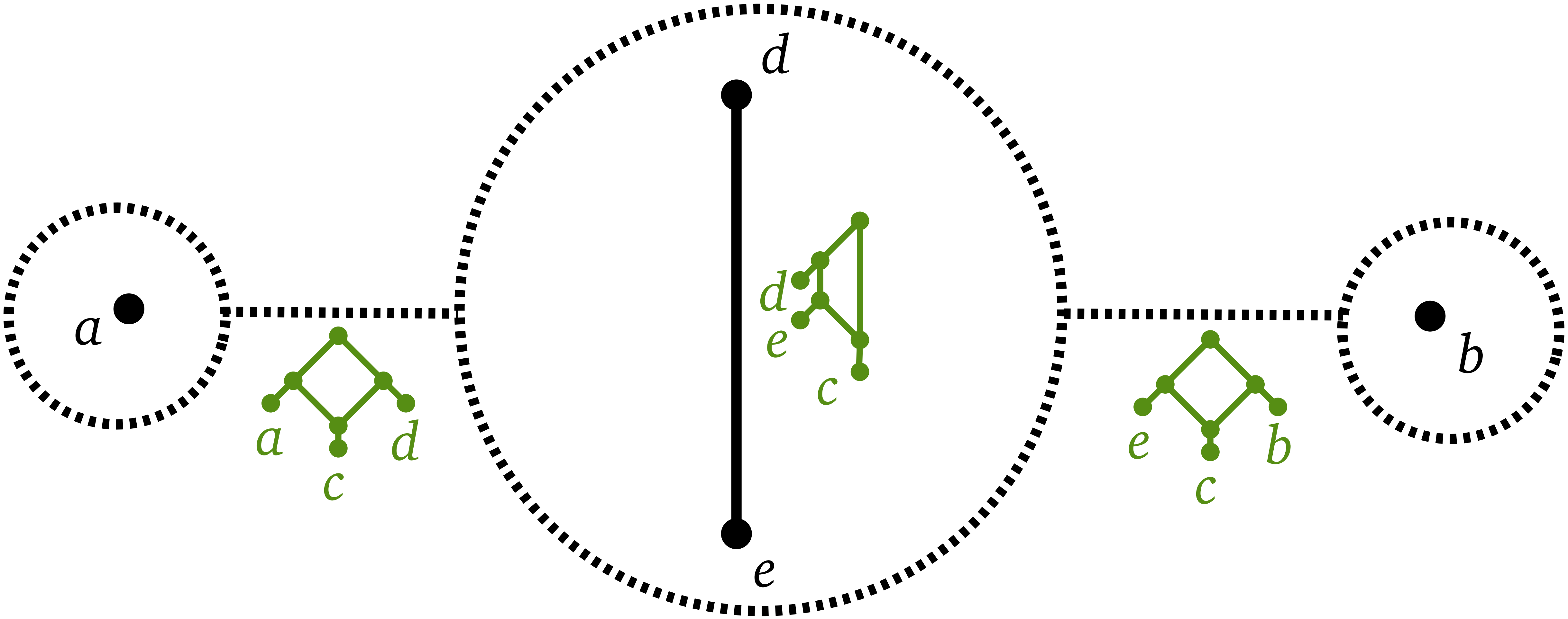
Step 3: which taxa are on the left, which taxa on the right?



Graph $\mathscr{M}$ connects taxa that have to be ***on the same side***.

Reconstruction Algorithm for Trinets

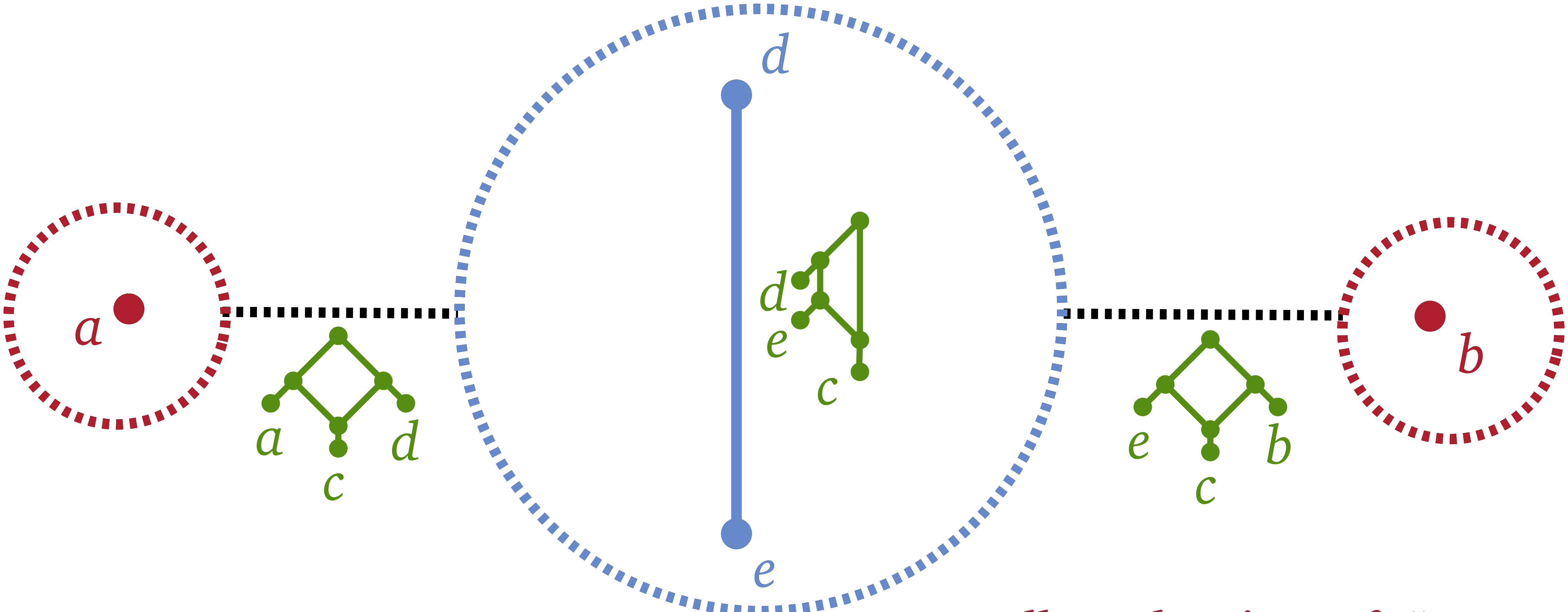
Step 3: which taxa are on the left, which taxa on the right?



Graph $\mathscr{W}$ connects components that have to be ***on different sides***.

Reconstruction Algorithm for Trinets

Step 3: which taxa are on the left, which taxa on the right?

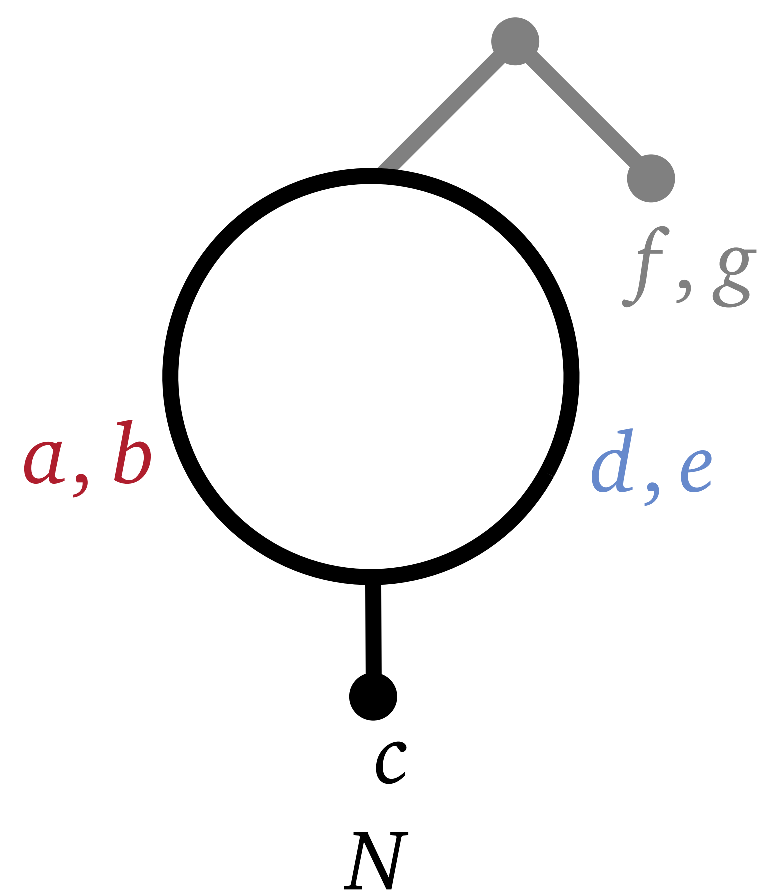


Try all 2-colourings of $\mathscr{W}$. Put the red vertices on the left, and the blue vertices on the right.

Graph $\mathscr{W}$ connects components that have to be ***on different sides***.

Reconstruction Algorithm for Trinets

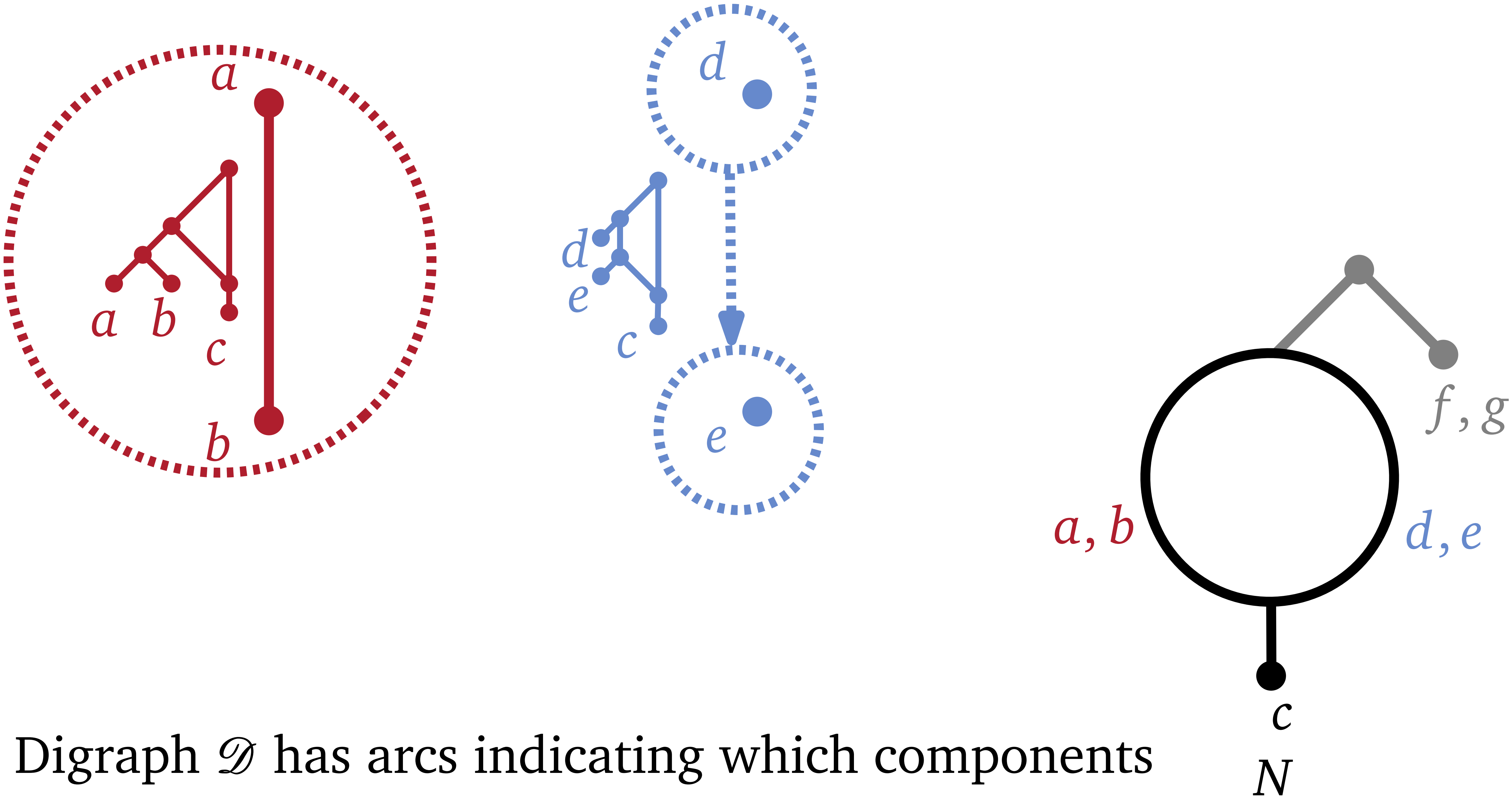
Step 4: partition the taxa on each side into sidenetworks.



Graph $\mathcal{O}$ connects taxa that have to be ***in the same sidenetwork***.

Reconstruction Algorithm for Trinets

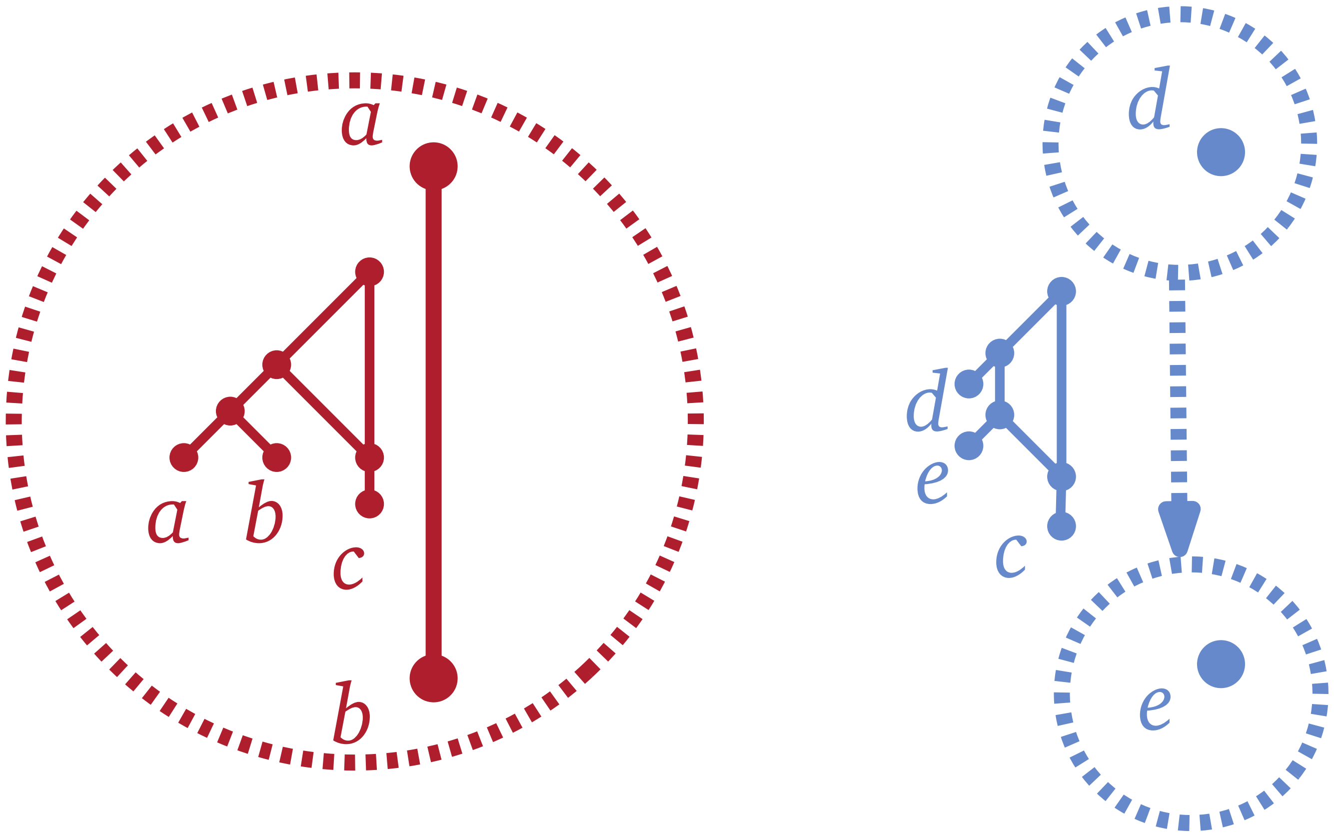
Step 4: partition the taxa on each side into sidenetworks.



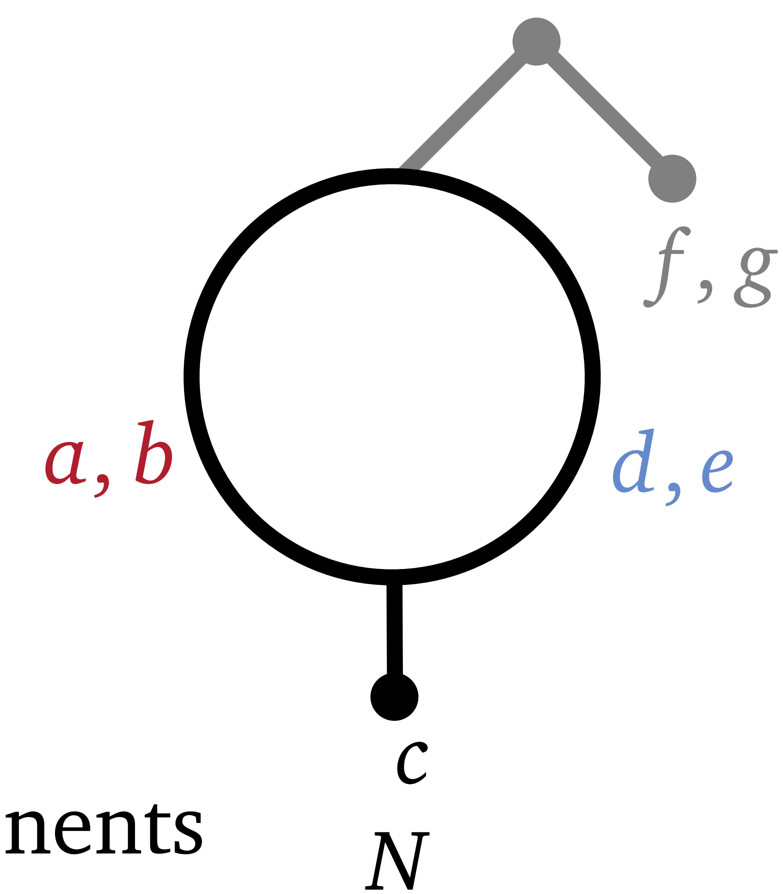
Digraph $\mathscr{D}$ has arcs indicating which components have to be ***above*** other components.

Reconstruction Algorithm for Trinets

Step 4: partition the taxa on each side into sidenetworks.



Find an indegree-0 vertex, put these taxa in the top subnetwork. Repeat.

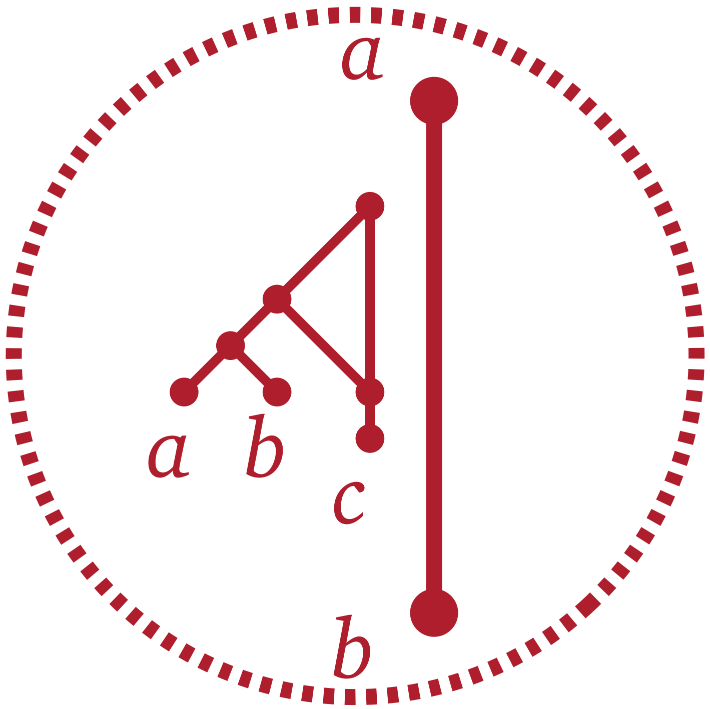


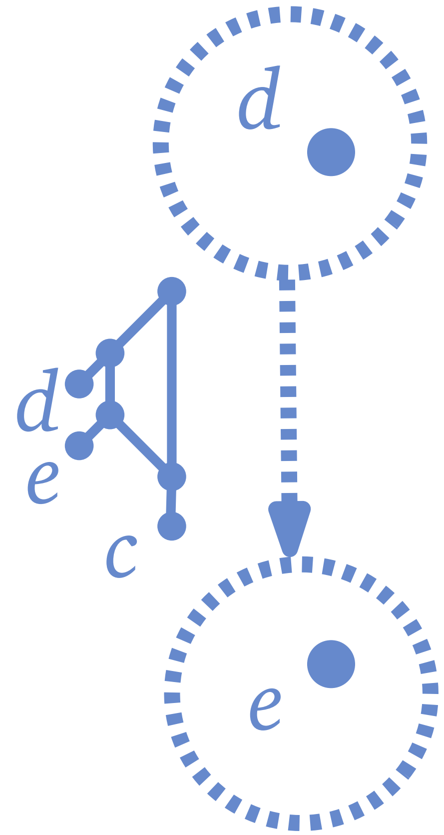
Digraph $\mathscr{D}$ has arcs indicating which components have to be ***above*** other components.

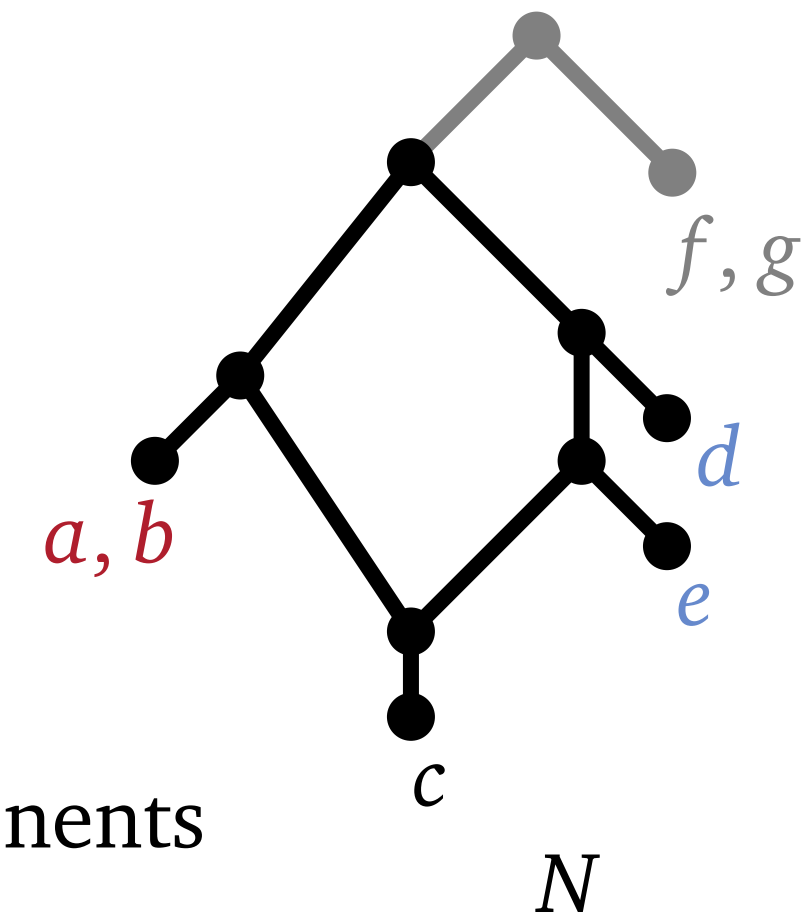
Reconstruction Algorithm for Trinets

Step 4: partition the taxa on each side into sidenetworks.

Find an indegree-0 vertex, put these taxa in the top subnetwork. Repeat.

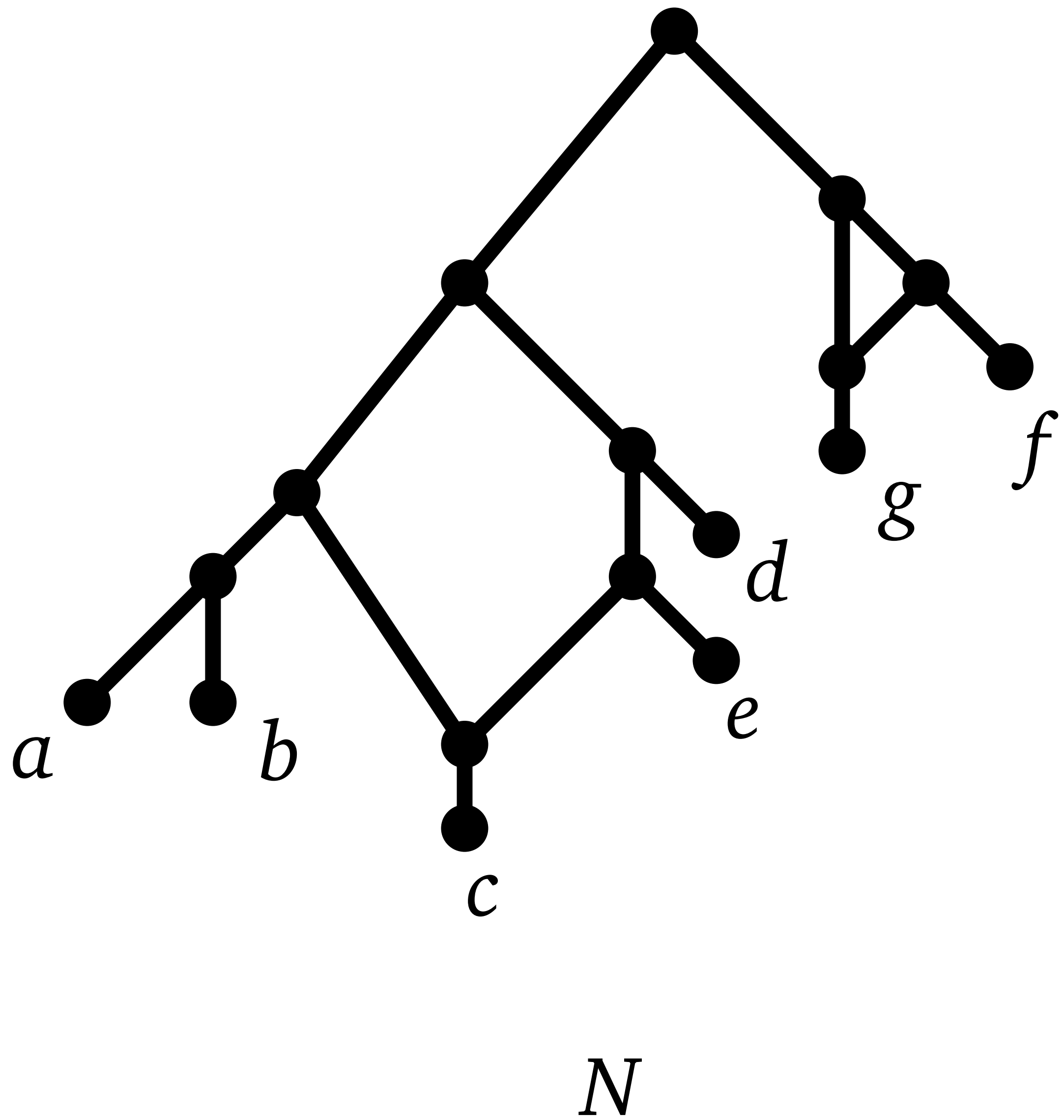






Digraph $\mathscr{D}$ has arcs indicating which components have to be ***above*** other components.

Step 5: recursively find sidenetworks.



Conclusion

- Constructing a level-1 supernetwork from trinets is NP-hard.
- Our exponential-time algorithm heavily exploits the structure of the problem.
- Some special cases can be solved in polynomial-time.
  - binets;
  - subnets with only size-3 cycles.