Summarizing Multiple Gene Trees Using Cluster Networks

Regula Rupp, Daniel H. Huson

MIEP, June 2008

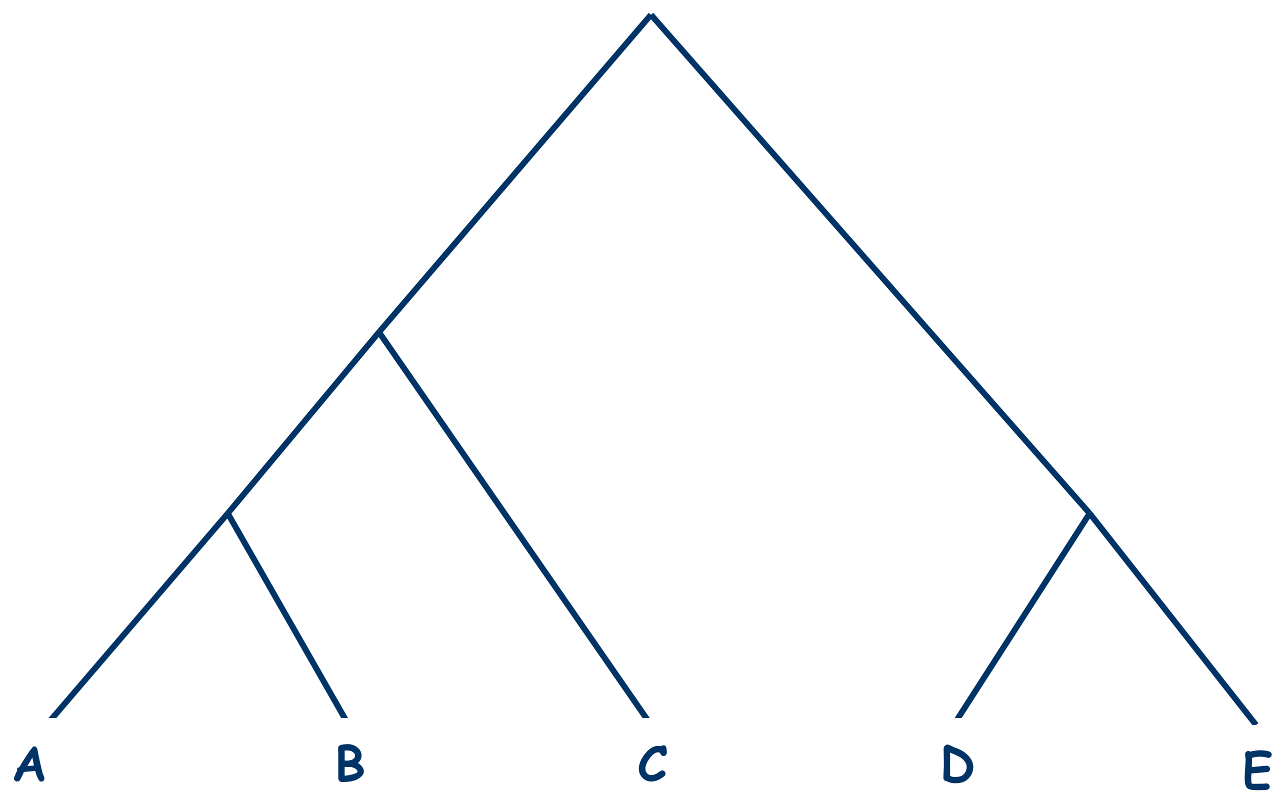
Eberhard Karls Universität Tübingen

Overview

- Trees, Clusters and Cluster Networks
- Hardwired vs. Softwired Networks
- Lowest Single Ancestor (LSA)
- LSA Consensus vs. Adams Consensus

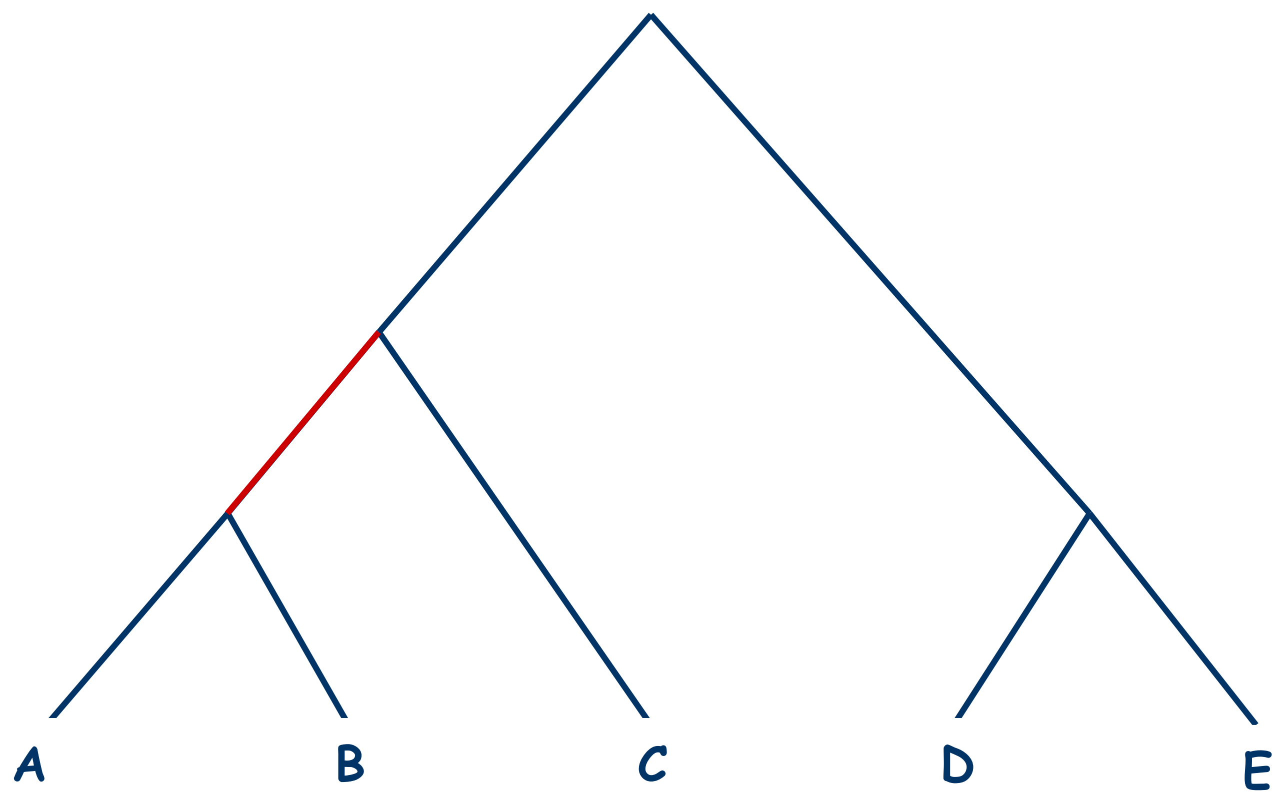
Clusters on Rooted Trees

- Every edge of a rooted tree defines a cluster of the taxon set *X*:



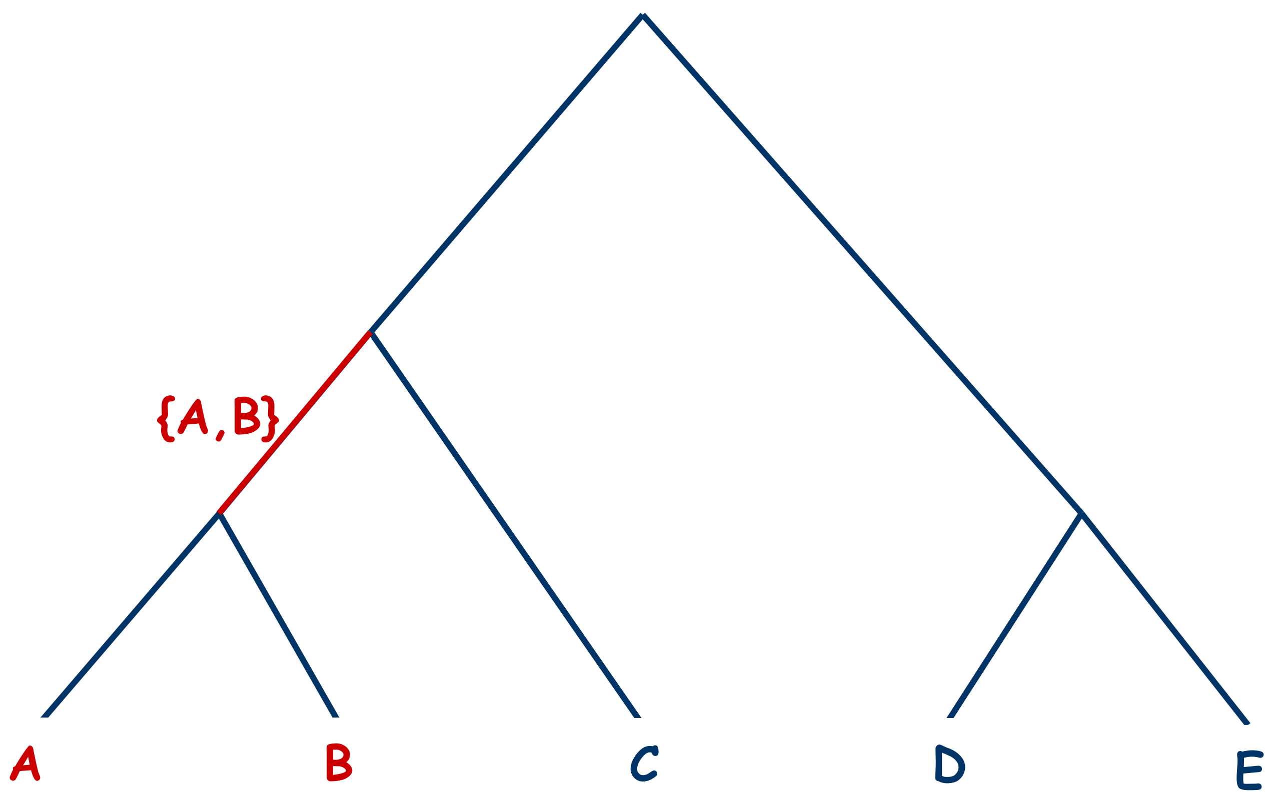
Clusters on Rooted Trees

- Every edge of a rooted tree defines a cluster of the taxon set *X*:

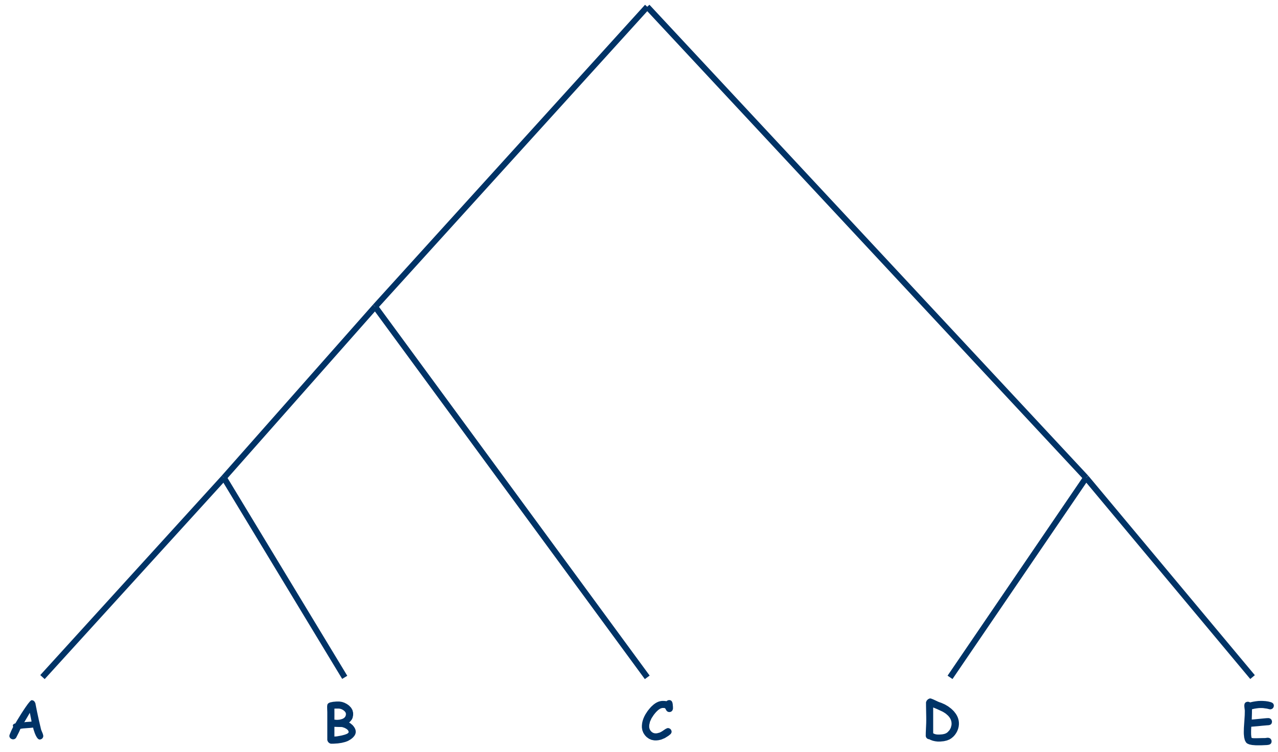


Clusters on Rooted Trees

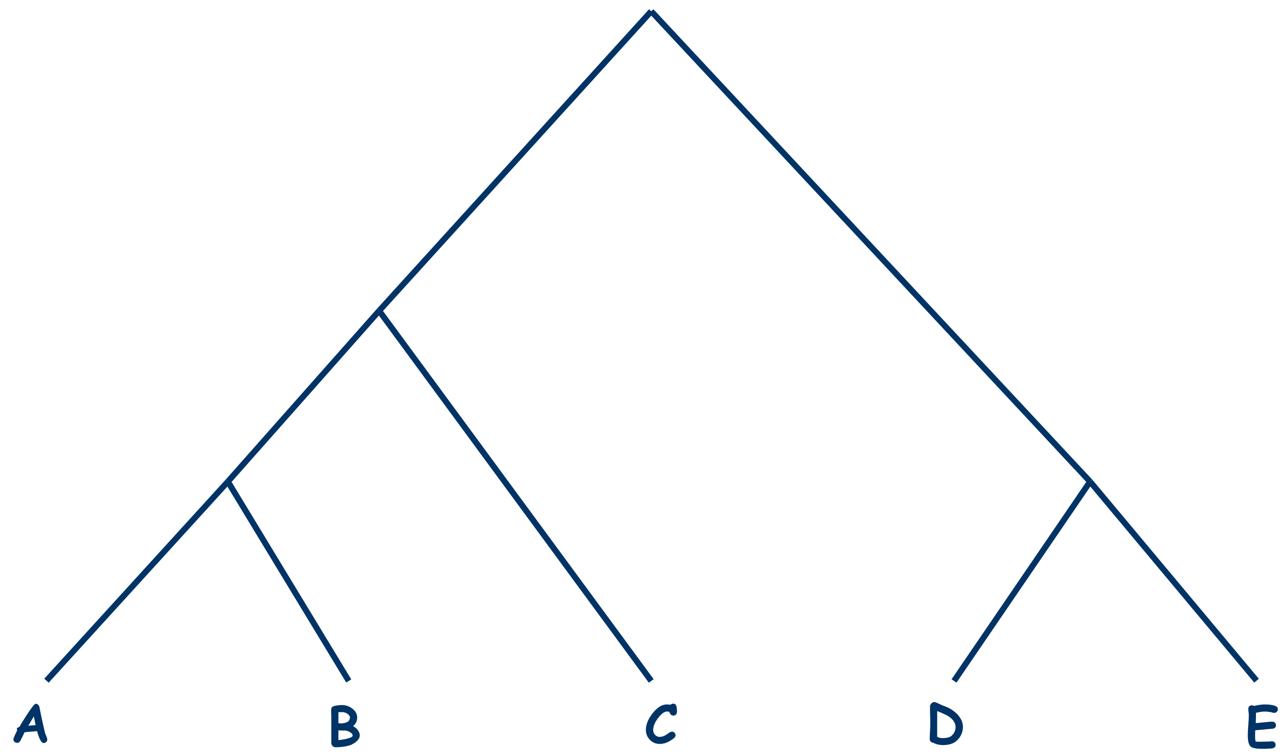
- Every edge of a rooted tree defines a cluster of the taxon set *X*:



Compatible Clusters and Rooted Trees

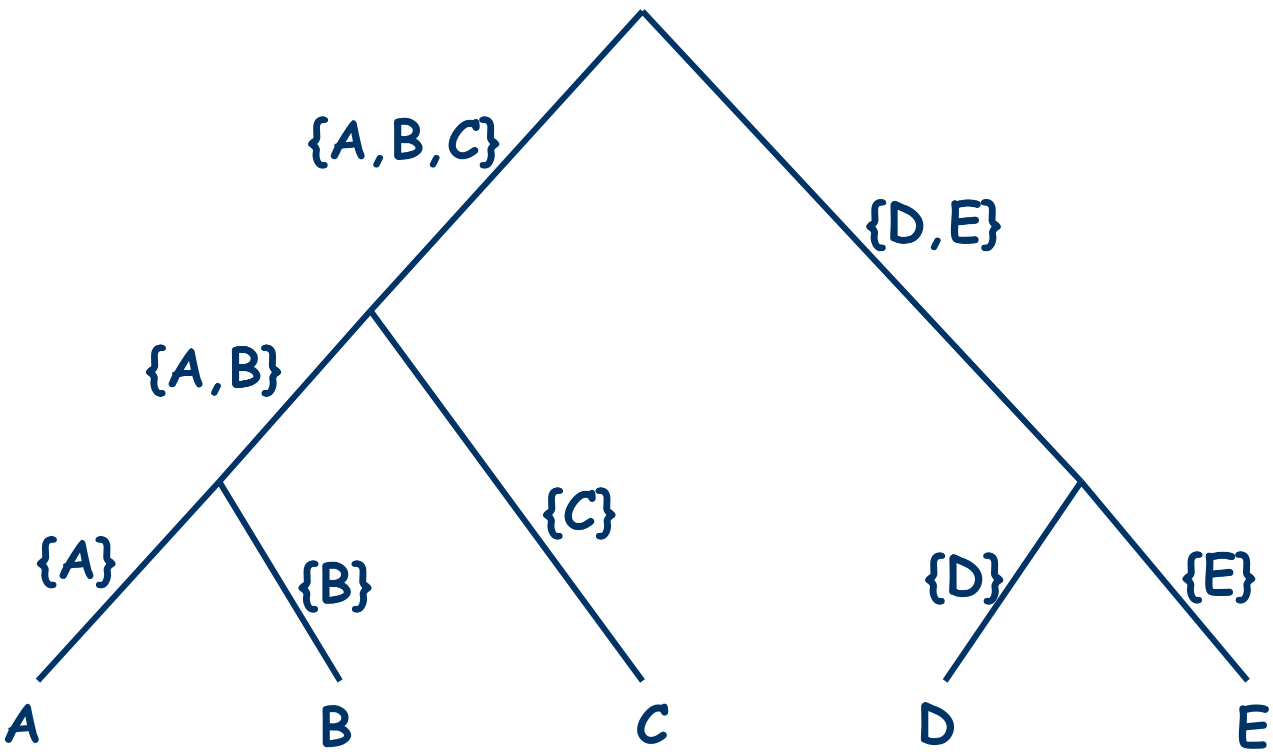


Compatible Clusters and Rooted Trees



Rooted Tree -> Compatible Clusters

Compatible Clusters and Rooted Trees



Rooted Tree -> Compatible Clusters

Compatible Clusters and Rooted Trees

{A,B,C}

{D,E}

{A,B}

{C}

{A}

{B}

{D}

{E}

Rooted Tree -> Compatible Clusters

Compatible Clusters and Rooted Trees

{A,B,C}

{D,E}

{A,B}

{C}

{A}

{B}

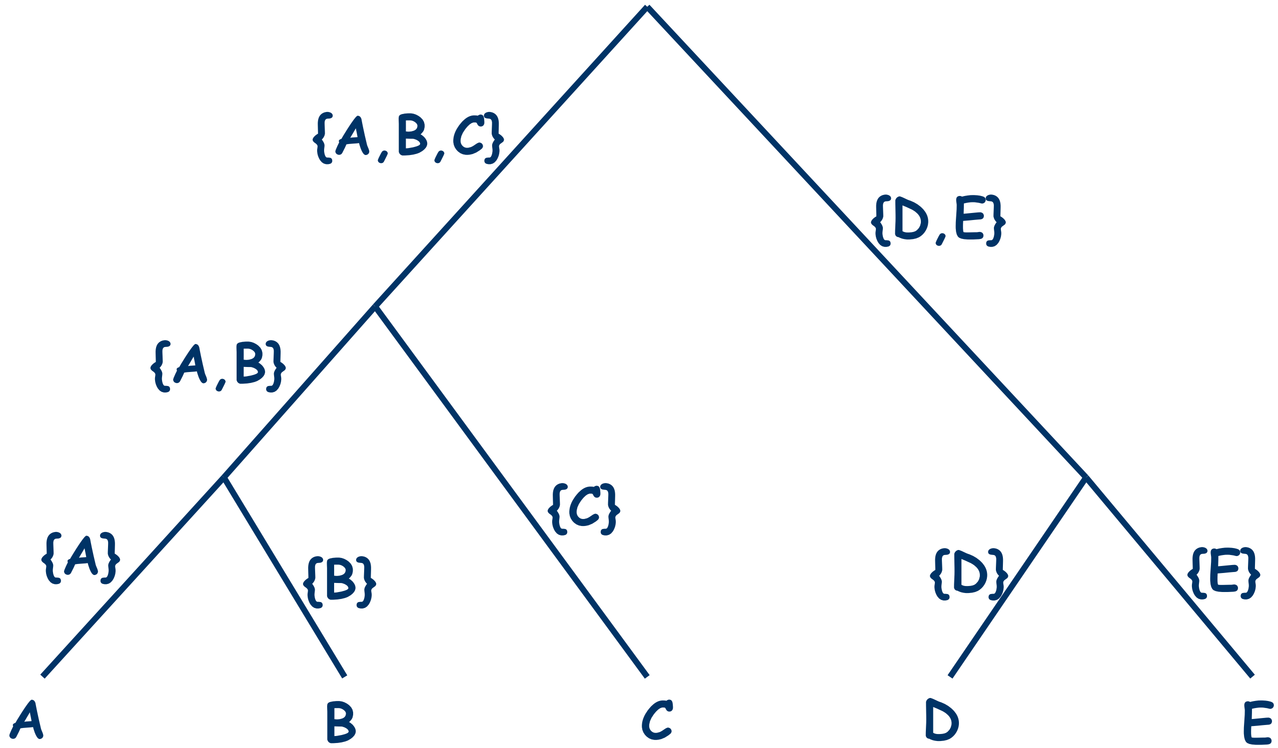
{D}

{E}

Rooted Tree -> Compatible Clusters

Compatible Clusters -> Rooted Trees

Compatible Clusters and Rooted Trees



Rooted Tree -> Compatible Clusters

Compatible Clusters -> Rooted Trees

Incompatible Clusters

Incompatible Clusters

Incompatible Clusters: {A,B} {B,C} …

Incompatible Clusters

Incompatible Clusters: {A,B} {B,C} …

Do not fit on one tree

Incompatible Clusters

Incompatible Clusters: {A,B} {B,C} …

Do not fit on one tree

Question:

How to represent *incompatible* clusters?

Idea: Hasse Diagram (Cover Graph)

Idea: Use a "Hasse diagram" or "cover digraph":

Clusters:

{A} {B} {C} {D} {E} {F}

{A,B} {B,C} {D,E}

{B,C,D,E} {D,E,F}

{A,B,C,D,E}

Idea: Hasse Diagram (Cover Graph)

Idea: Use a “Hasse diagram” or “cover digraph”:

Clusters:

{A} {B} {C} {D} {E} {F}

{A,B} {B,C} {D,E}

{B,C,D,E} {D,E,F}

{A,B,C,D,E}

A B C D E F

Idea: Hasse Diagram (Cover Graph)

Idea: Use a "Hasse diagram" or "cover digraph":

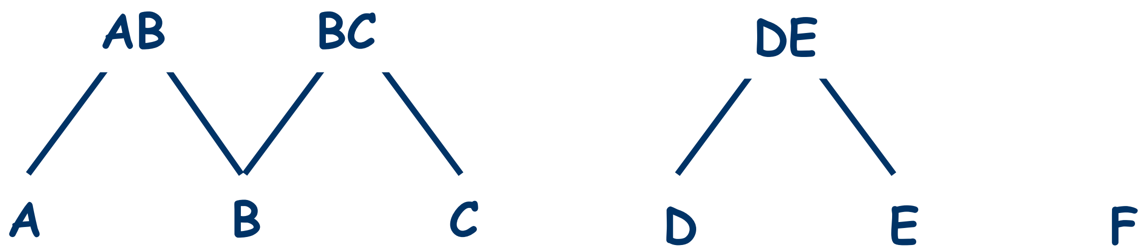
Clusters:

{A} {B} {C} {D} {E} {F}

{A,B} {B,C} {D,E}

{B,C,D,E} {D,E,F}

{A,B,C,D,E}



Idea: Hasse Diagram (Cover Graph)

Idea: Use a “Hasse diagram” or “cover digraph”:

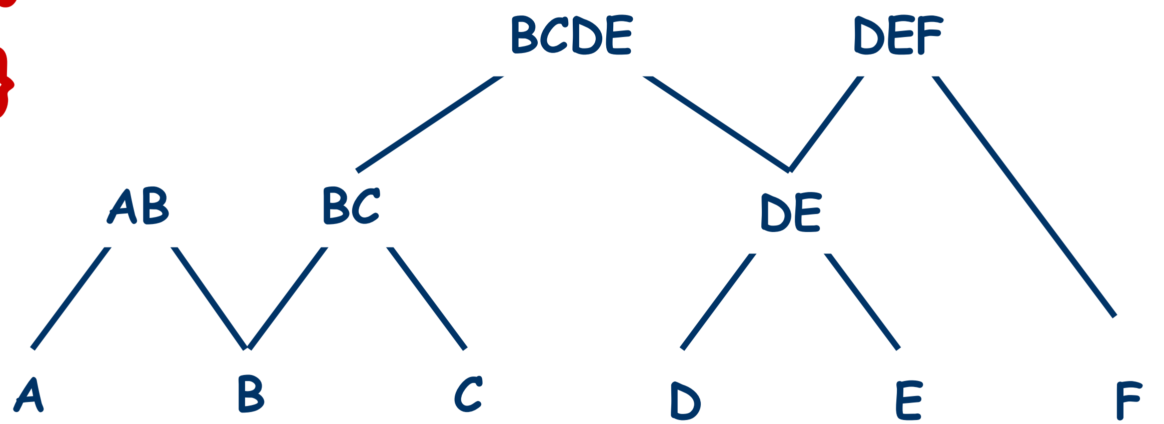
Clusters:

{A} {B} {C} {D} {E} {F}

{A,B} {B,C} {D,E}

{B,C,D,E} {D,E,F}

{A,B,C,D,E}



Idea: Hasse Diagram (Cover Graph)

Idea: Use a “Hasse diagram” or “cover digraph”:

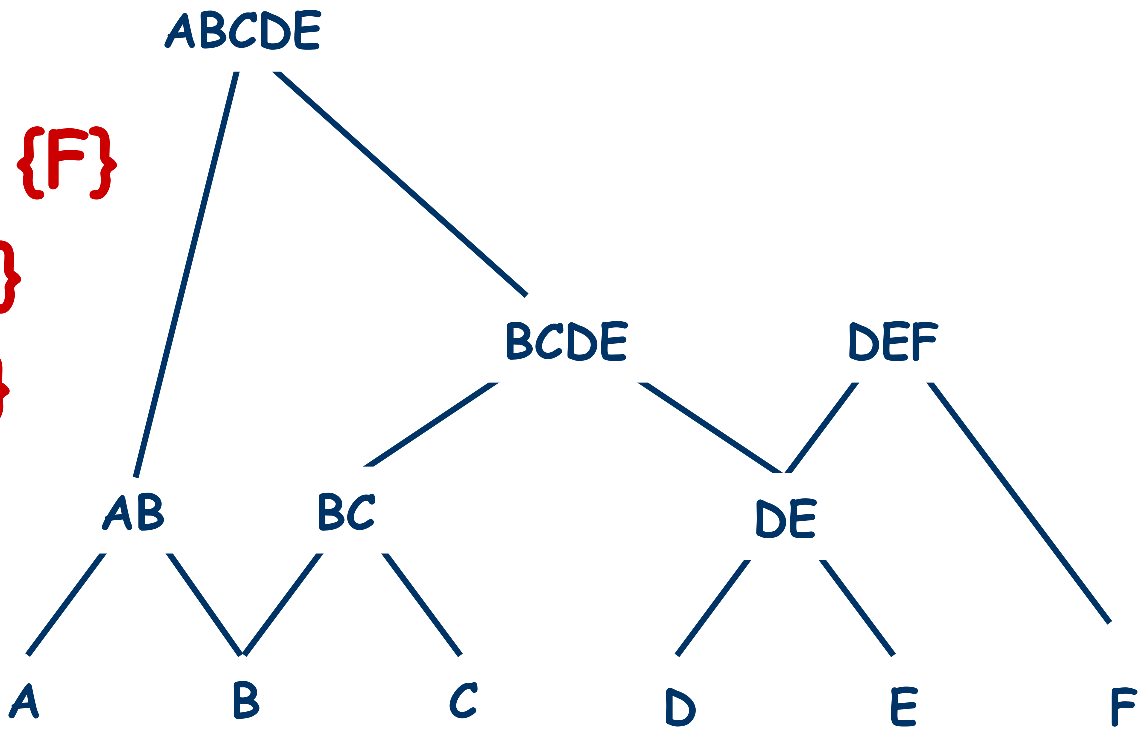
Clusters:

{A} {B} {C} {D} {E} {F}

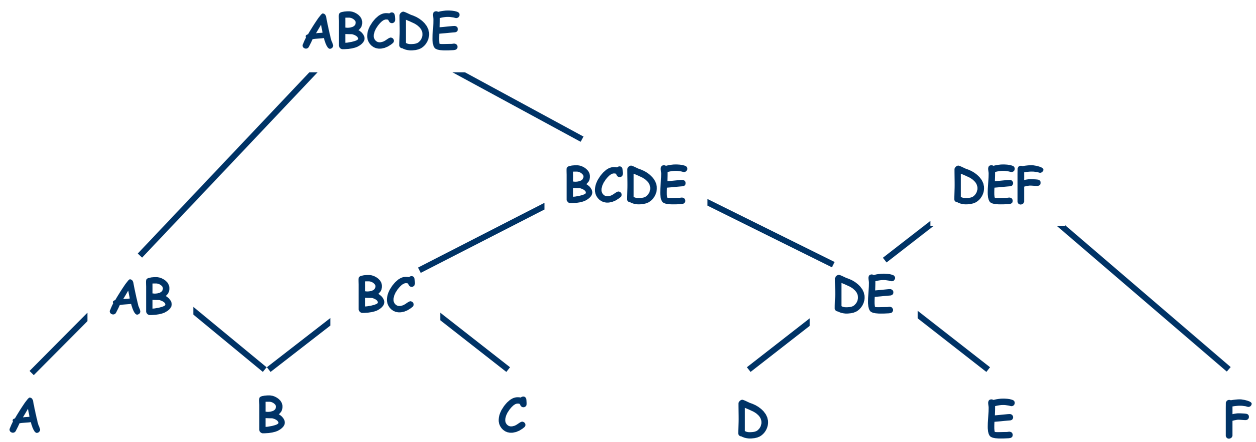
{A,B} {B,C} {D,E}

{B,C,D,E} {D,E,F}

{A,B,C,D,E}

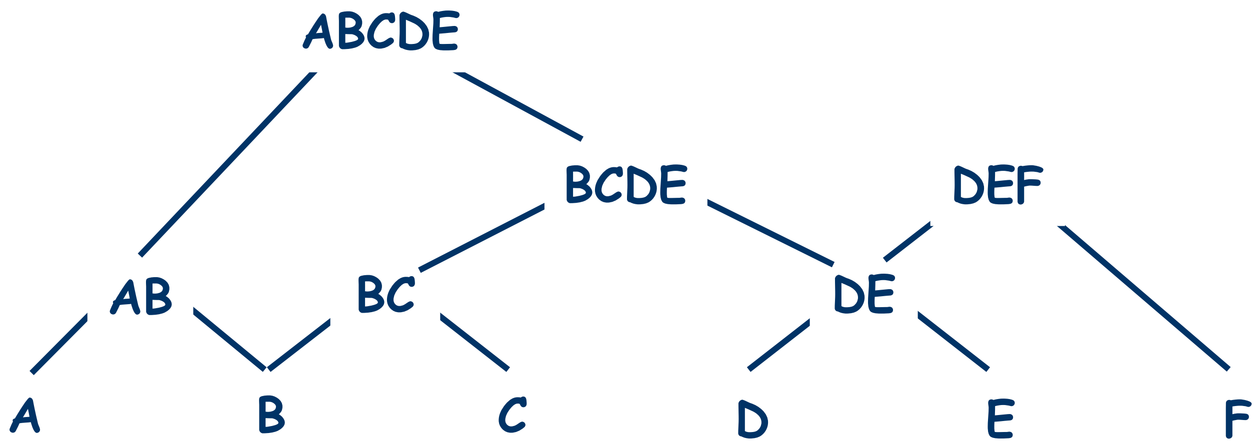


Problems:



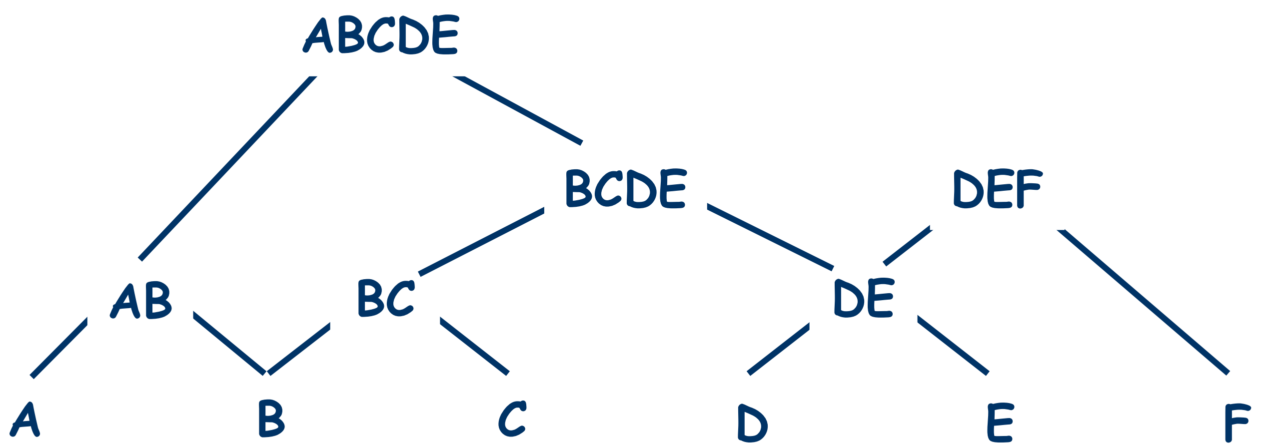
Problems:

- No single "root"



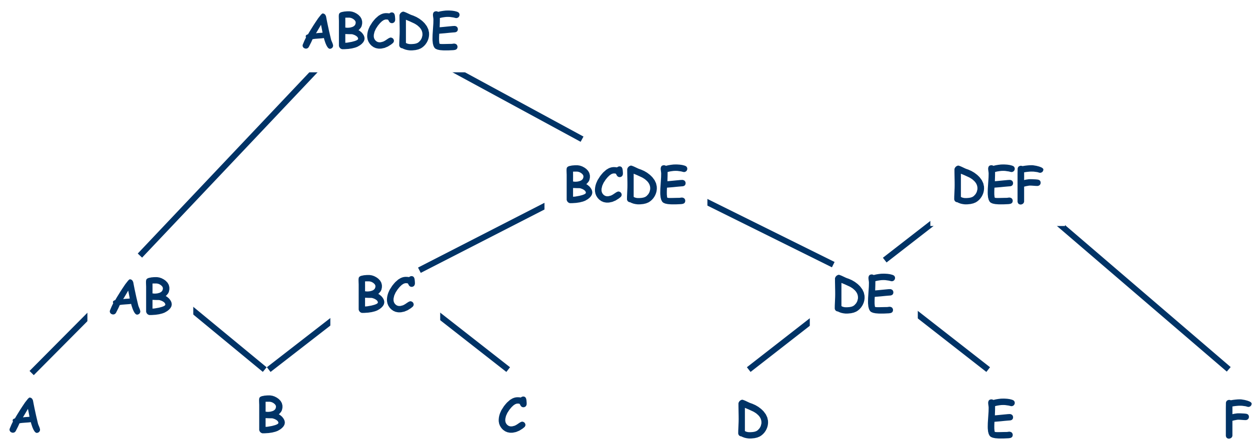
Problems:

- No single “root”
- Leaves with more than one in-edge



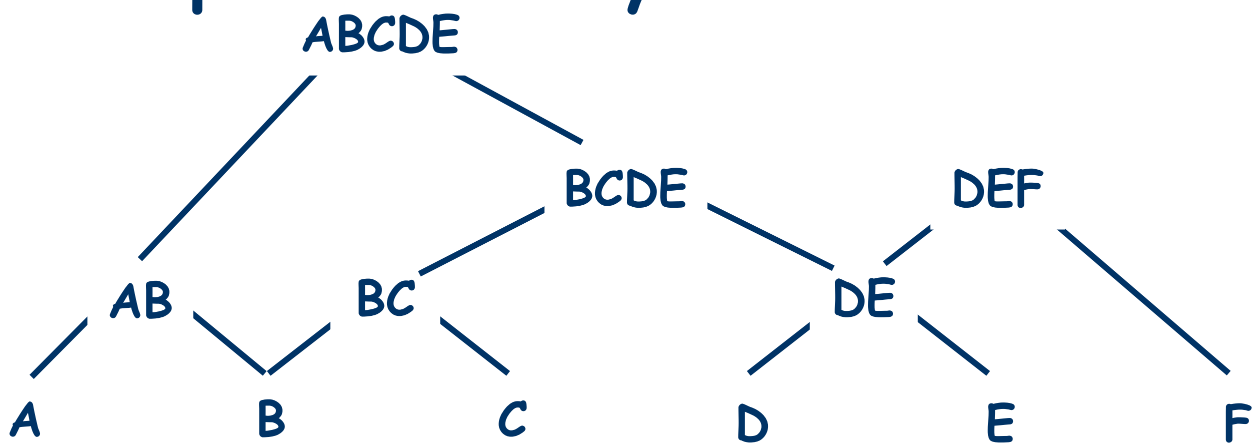
Problems:

- No single “root”
- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges



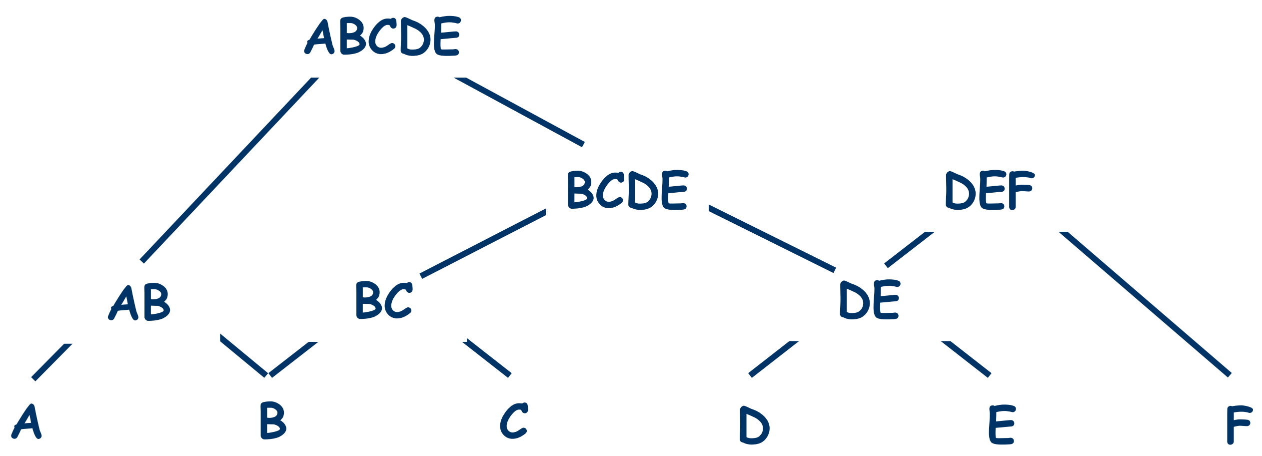
Problems:

- No single "root"
- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges
- Clusters represented by nodes instead of edges



Solutions:

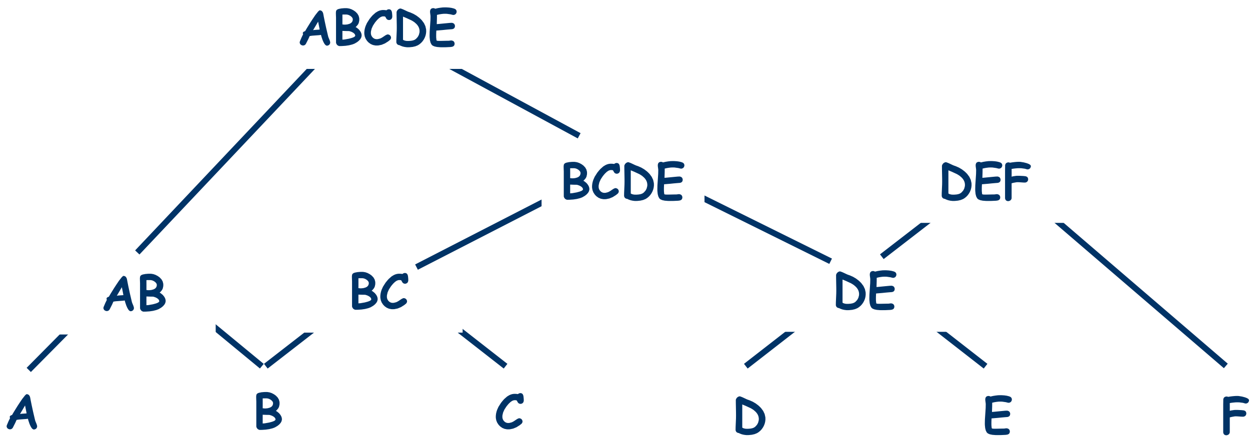
- No single "root"



Solutions:

- No single "root"

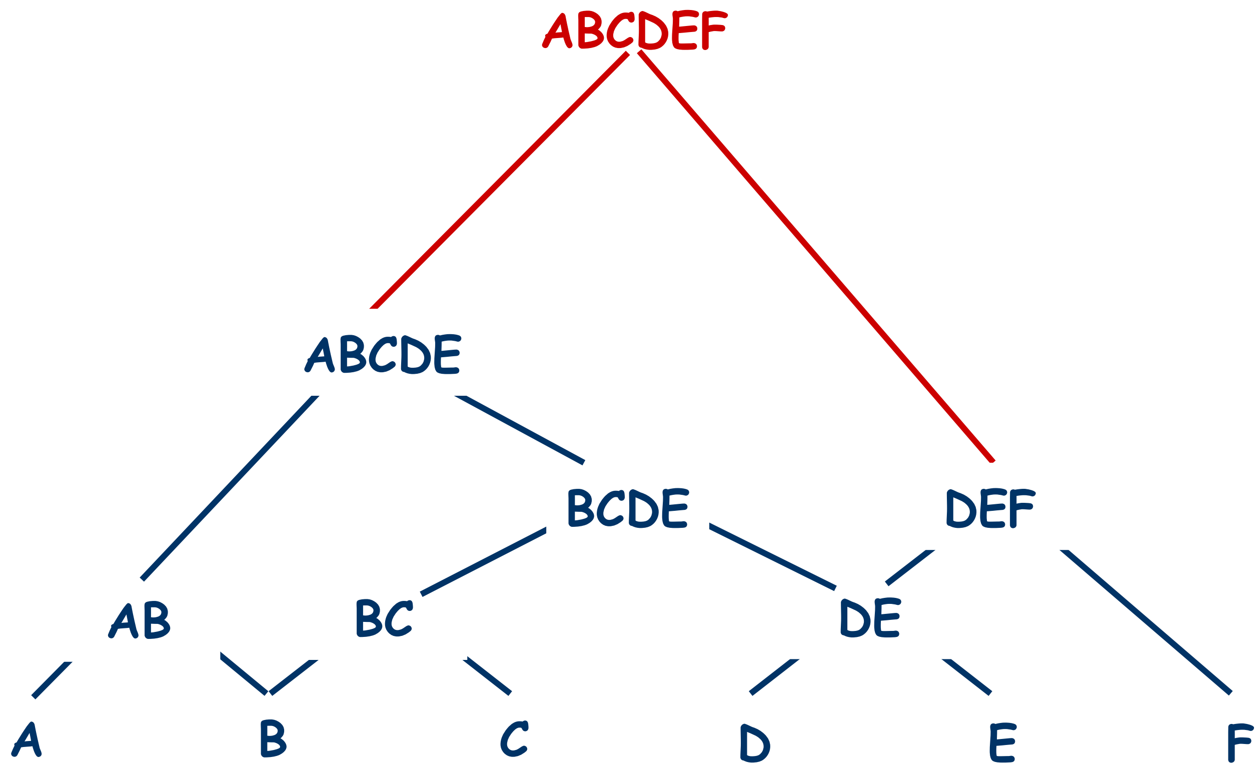
-> Add full set to the clusters



Solutions:

- No single "root"

-> Add full set to the clusters



Solutions:

- No single "root"

-> Add full set to the clusters

- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges

Solutions:

- No single "root"

-> Add full set to the clusters

- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges

-> If in-degree >1, insert new edge:

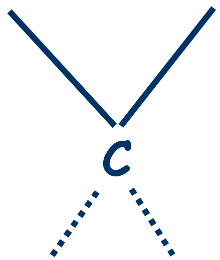
Solutions:

- No single “root”

-> Add full set to the clusters

- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges

-> If in-degree >1, insert new edge:



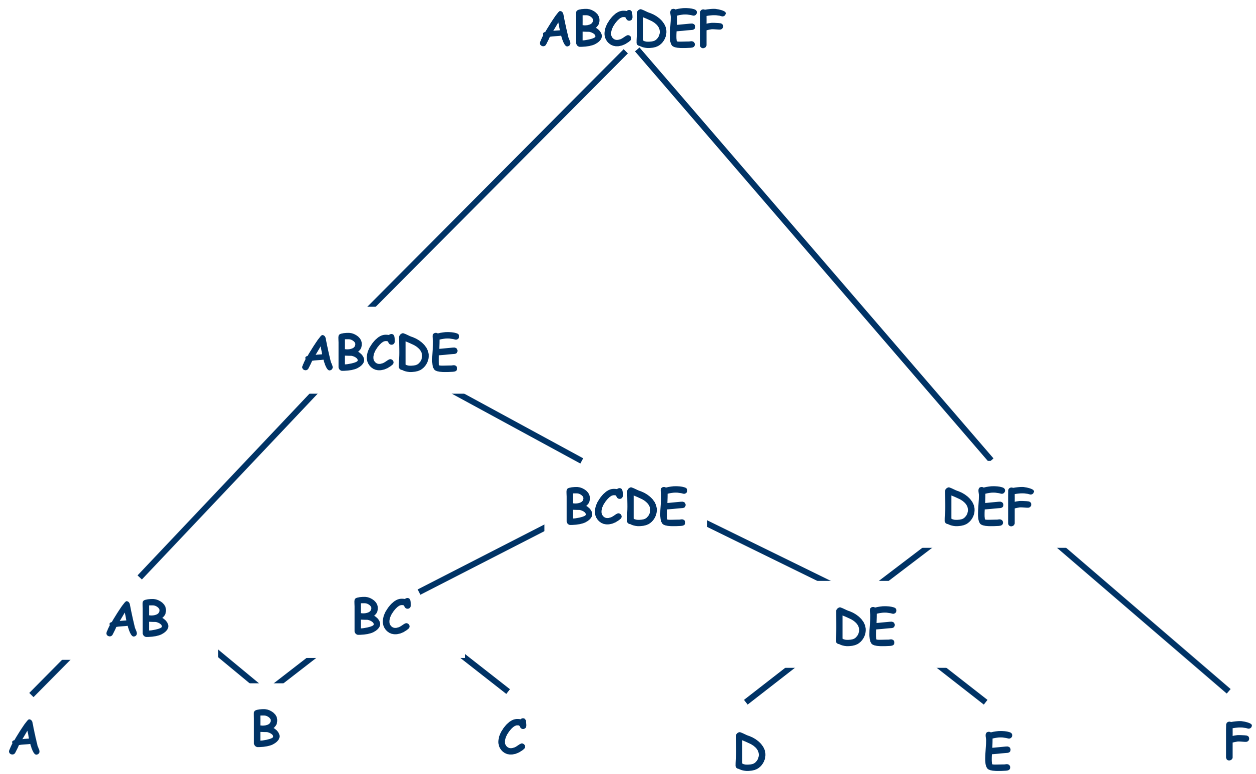
Solutions:

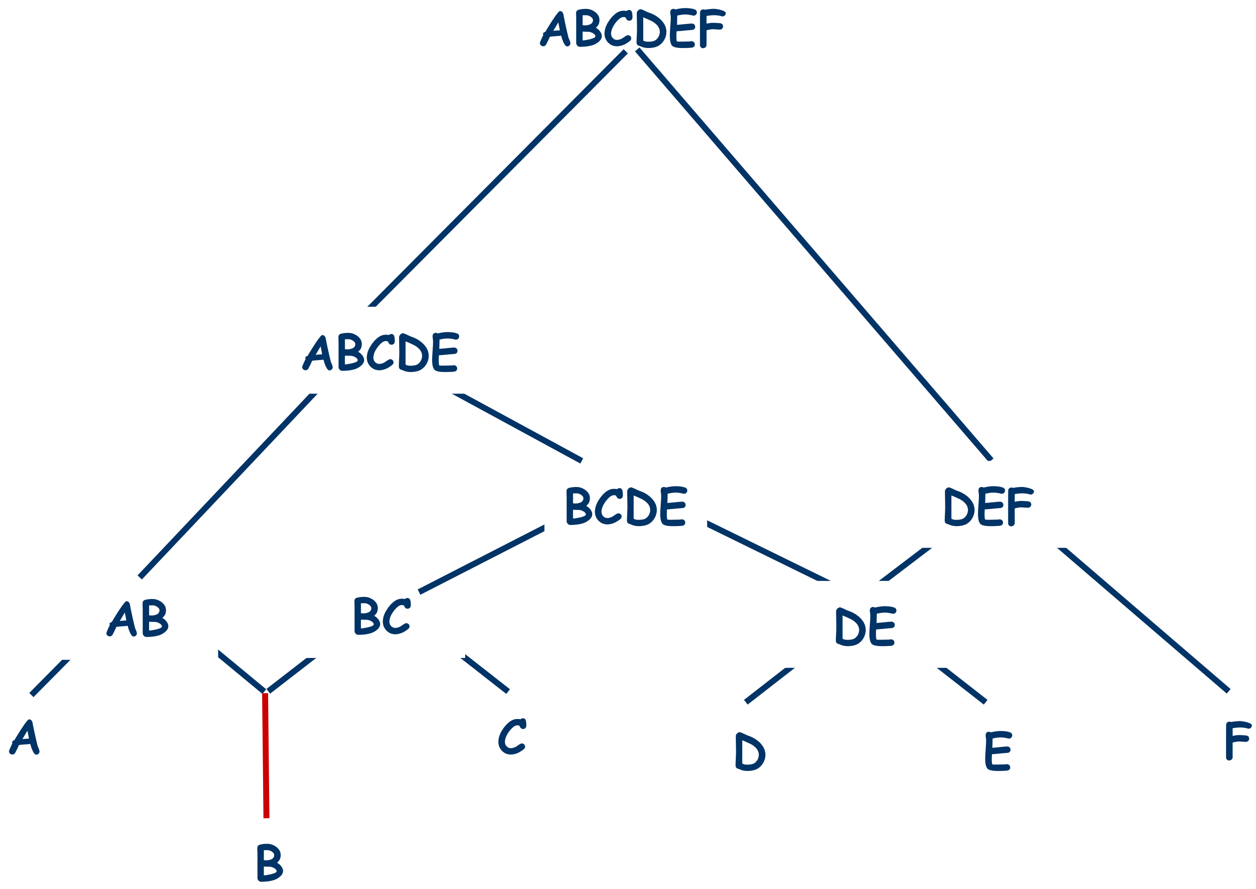
- No single “root”

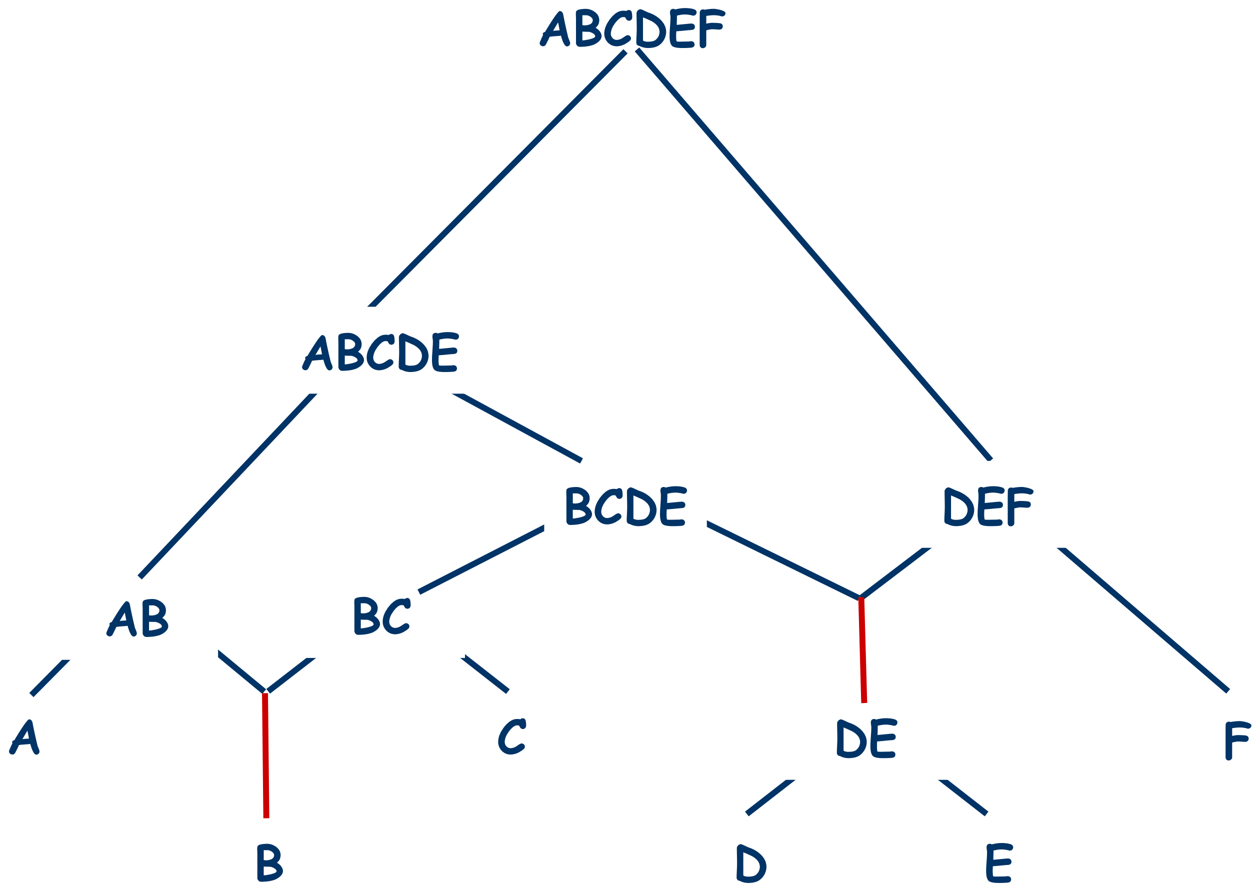
-> Add full set to the clusters

- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges

-> If in-degree >1, insert new edge:


ABCDEF
ABCDE
BCDE
DEF
AB
BC
DE
A
B
C
D
E
F


ABCDEF
ABCDE
BCDE
DEF
AB
BC
DE
A
C
D
E
F
B


ABCDEF
ABCDE
BCDE
DEF
AB
BC
A
C
DE
F
B
D
E

Solutions:

- No single “root”

-> Add full set to the clusters

- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges

-> If in-degree >1, insert new edge

- Clusters represented by nodes instead of edges

Solutions:

- No single "root"

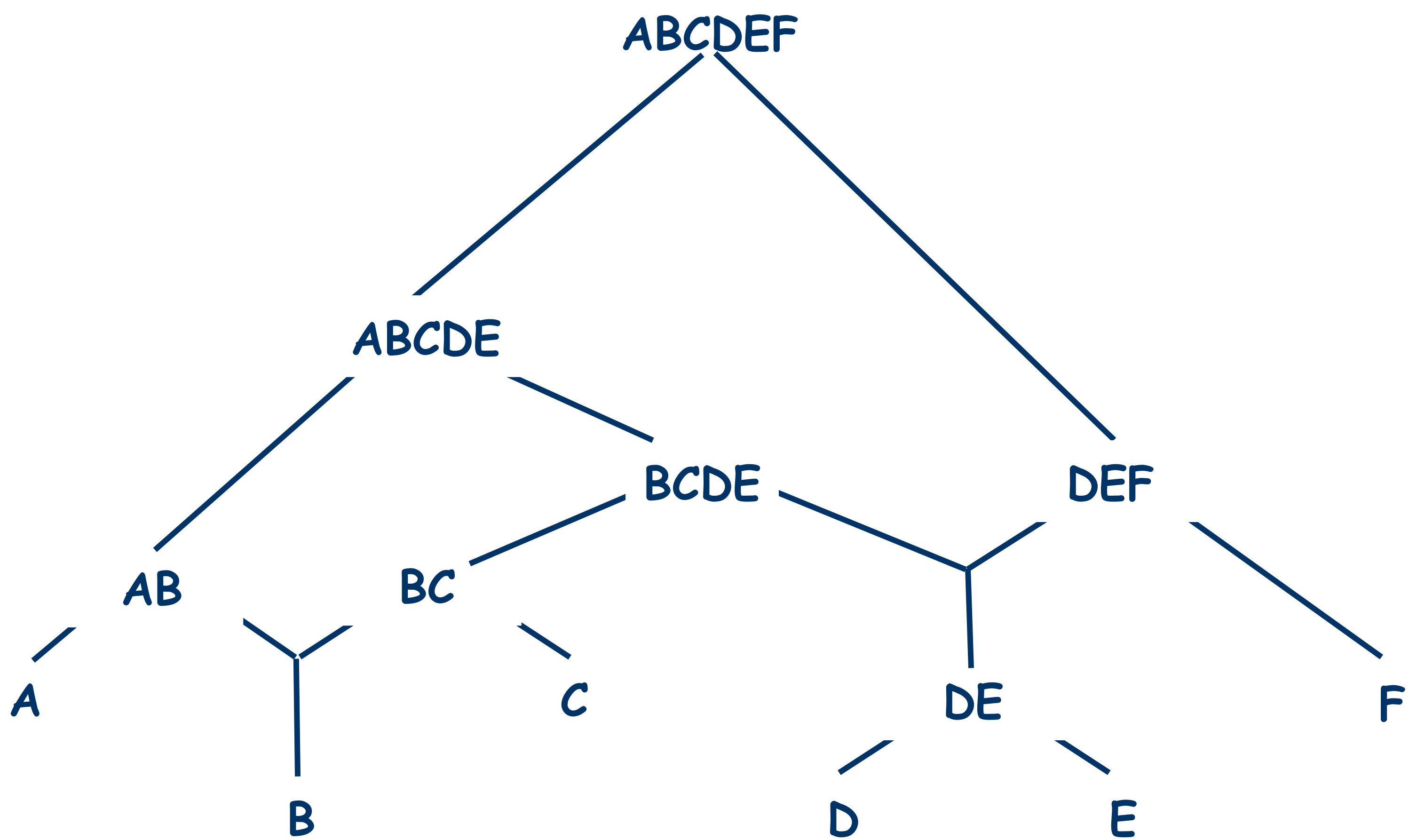
-> Add full set to the clusters

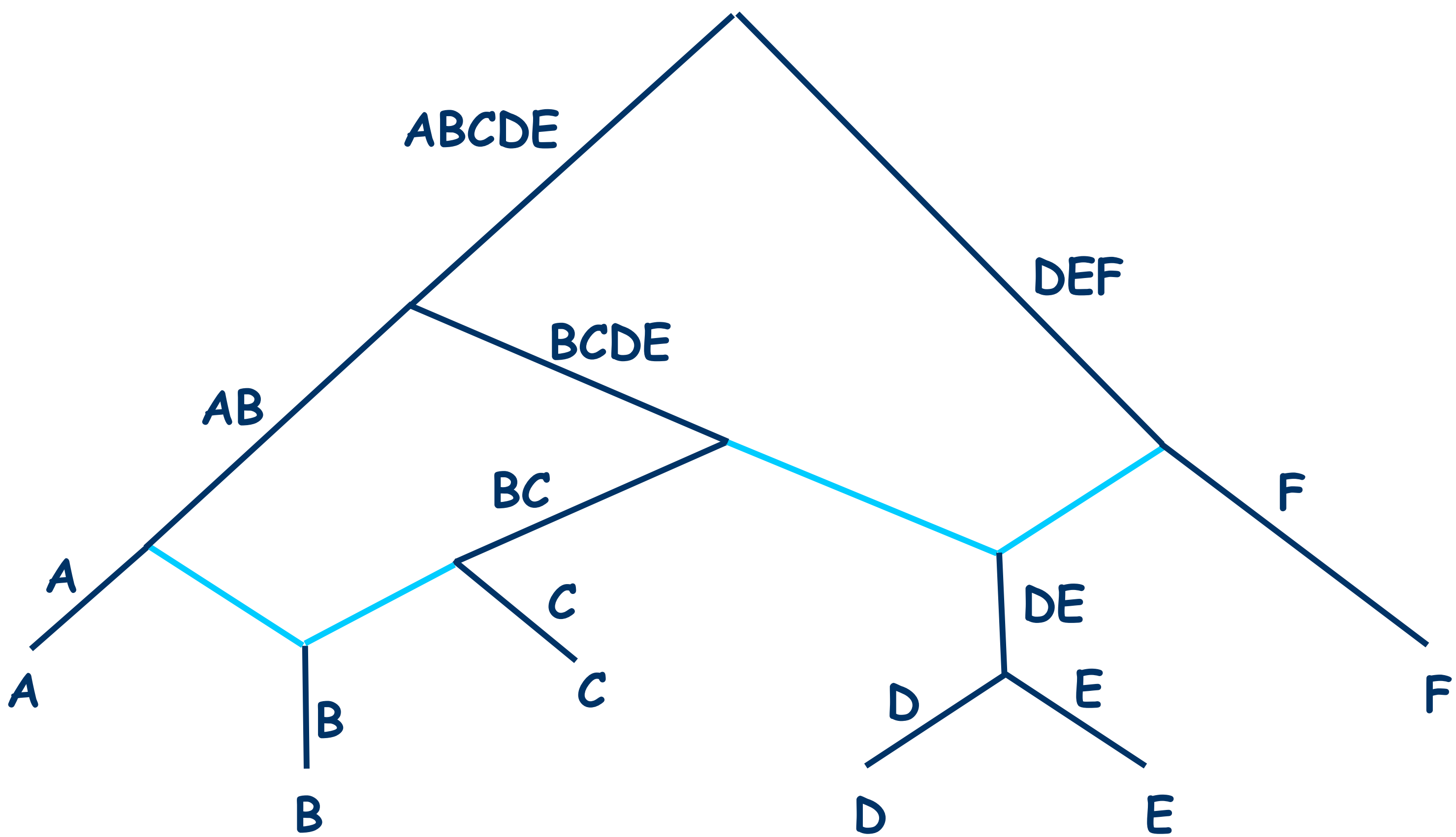
- Leaves with more than one in-edge
- Internal nodes with multiple in-edges and multiple out-edges

-> If in-degree >1, insert new edge

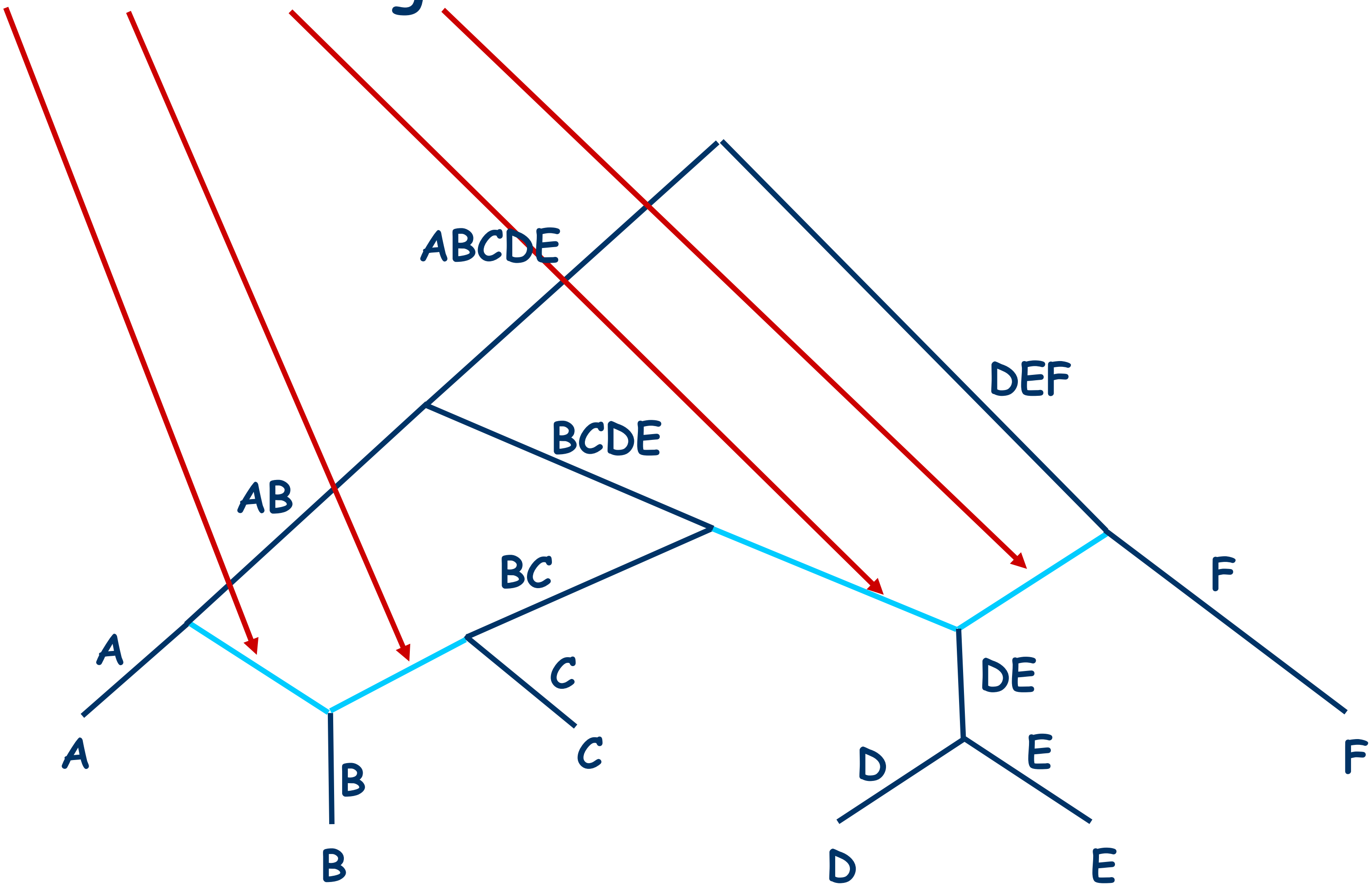
- Clusters represented by nodes instead of edges

-> Represent every cluster by its in-edge
(which is unique now!)


ABCDEF
ABCDE
BCDE
DEF
AB
BC
A
C
DE
F
B
D
E


ABCDE
DEF
BCDE
AB
BC
F
A
C
DE
A
C
D
E
F
B
B
D
E

Reticulation edges



Result: cluster network

Result: cluster network

A cluster network consists of a rooted directed acyclic graph together with a leaf-labeling and 3 additional properties:

Result: cluster network

A cluster network consists of a rooted directed acyclic graph together with a leaf-labeling and 3 additional properties:

- Uniqueness

Result: cluster network

A cluster network consists of a rooted directed acyclic graph together with a leaf-labeling and 3 additional properties:

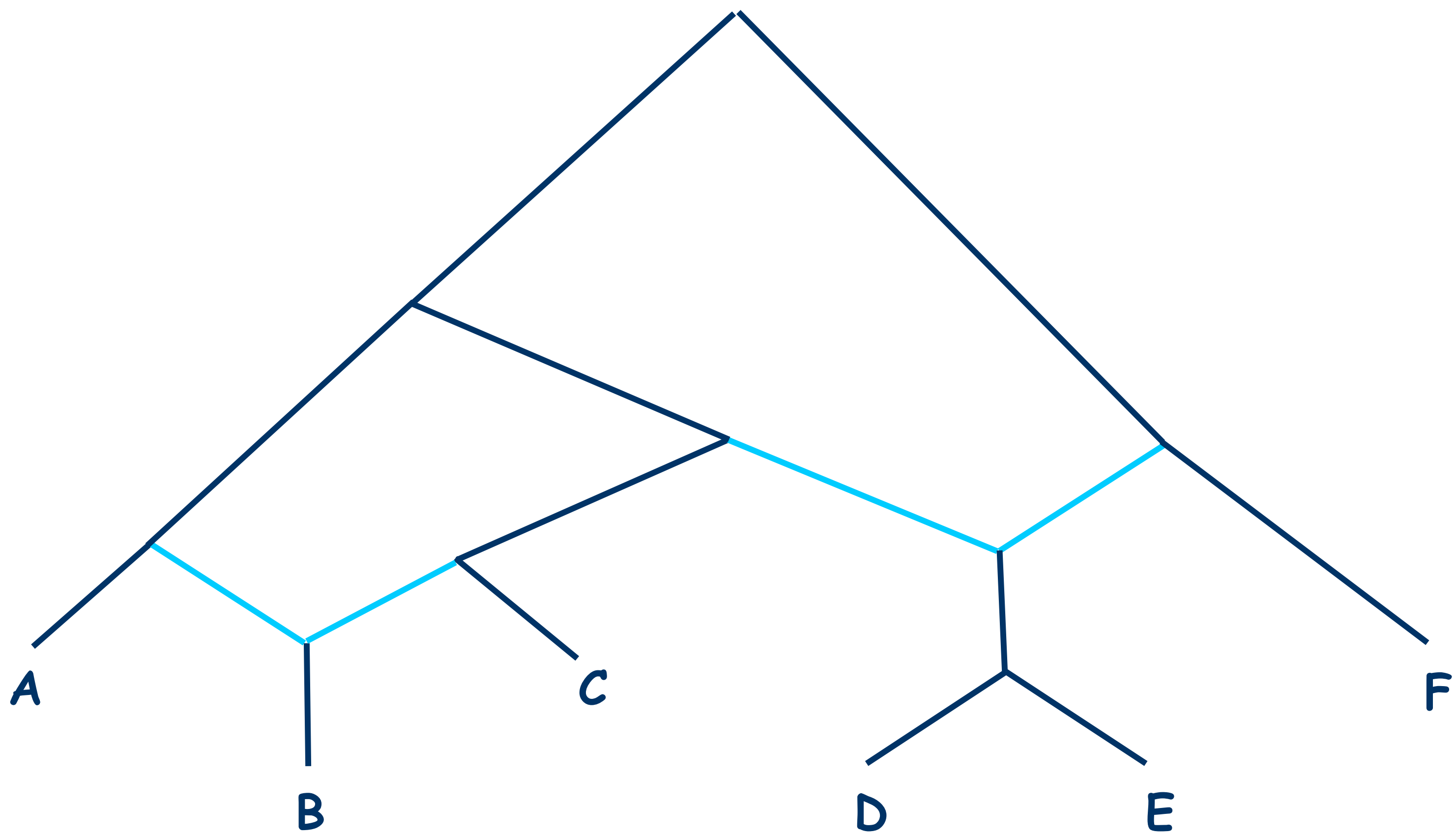
- Uniqueness
- Nestedness

Result: cluster network

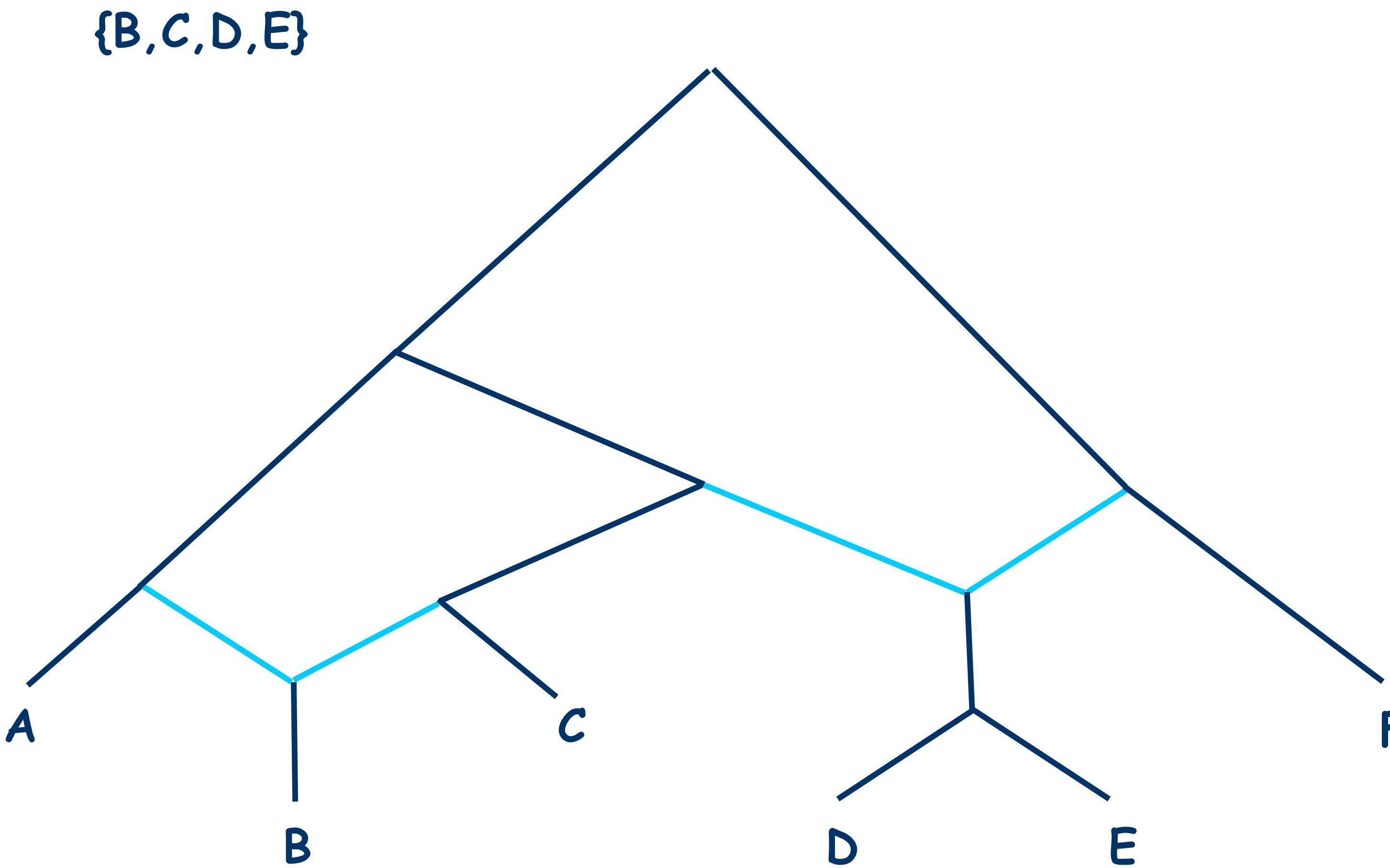
A cluster network consists of a rooted directed acyclic graph together with a leaf-labeling and 3 additional properties:

- Uniqueness
- Nestedness
- Reducedness

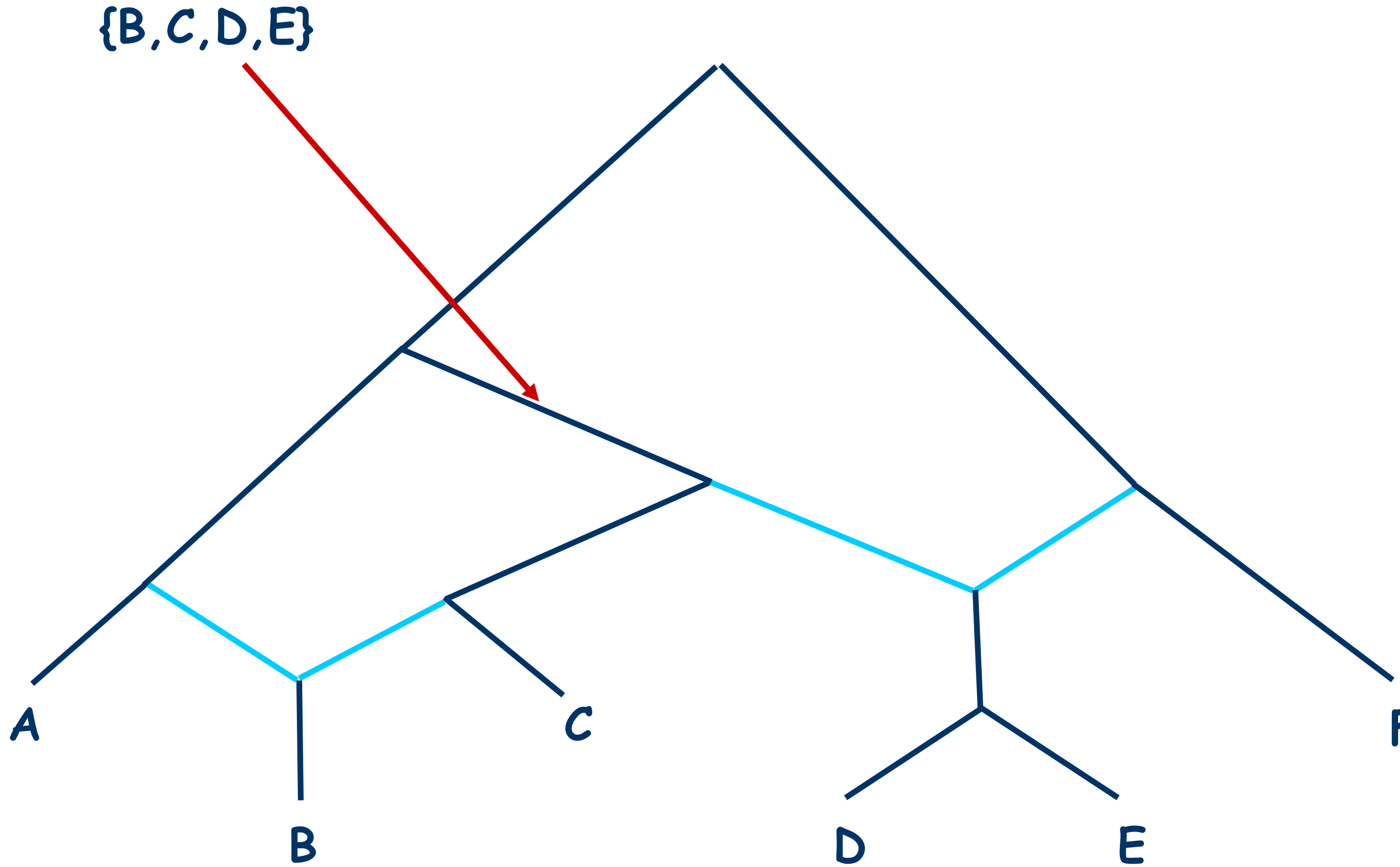
Uniqueness



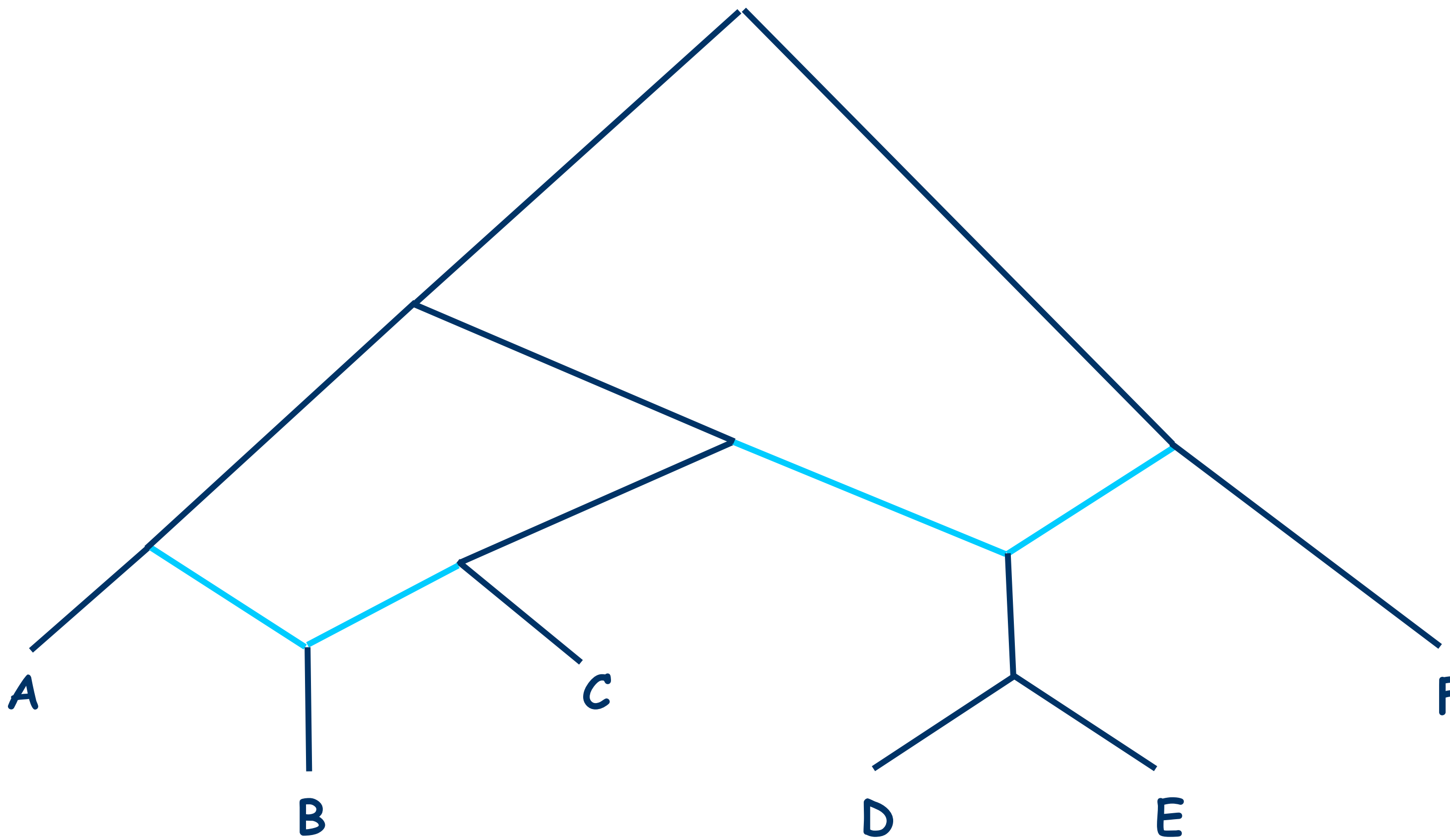
Uniqueness



Uniqueness

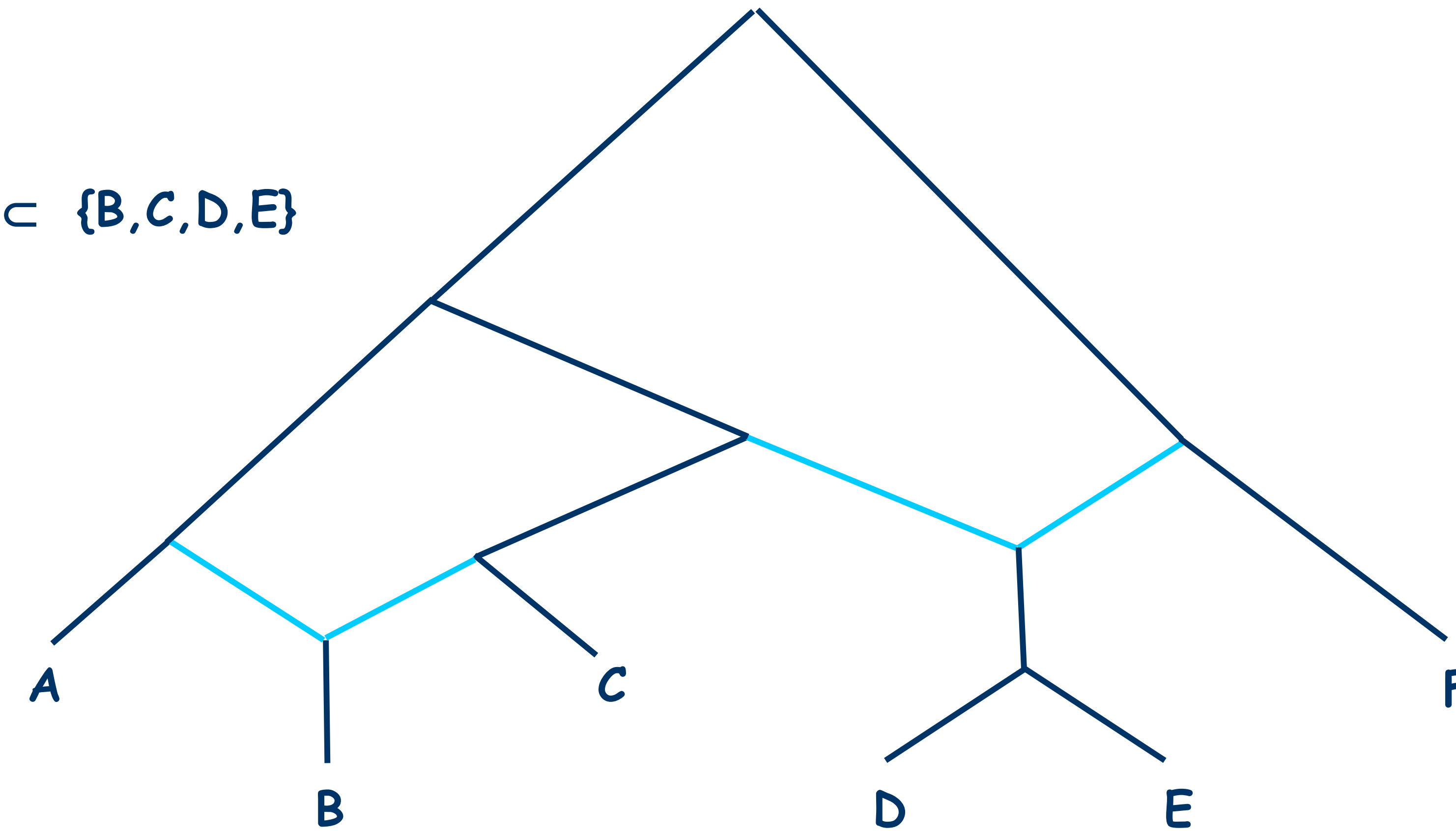


Nestedness

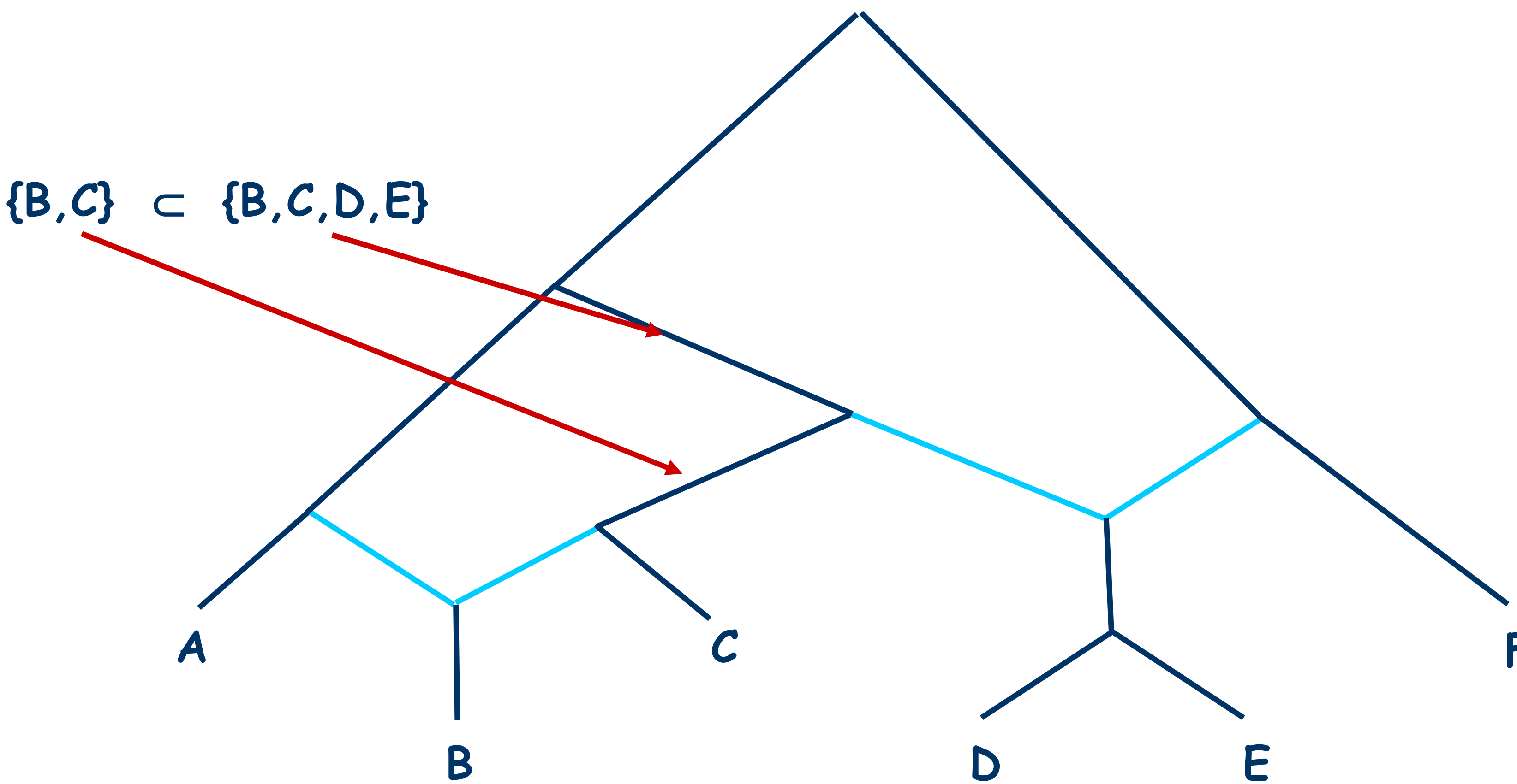


Nestedness

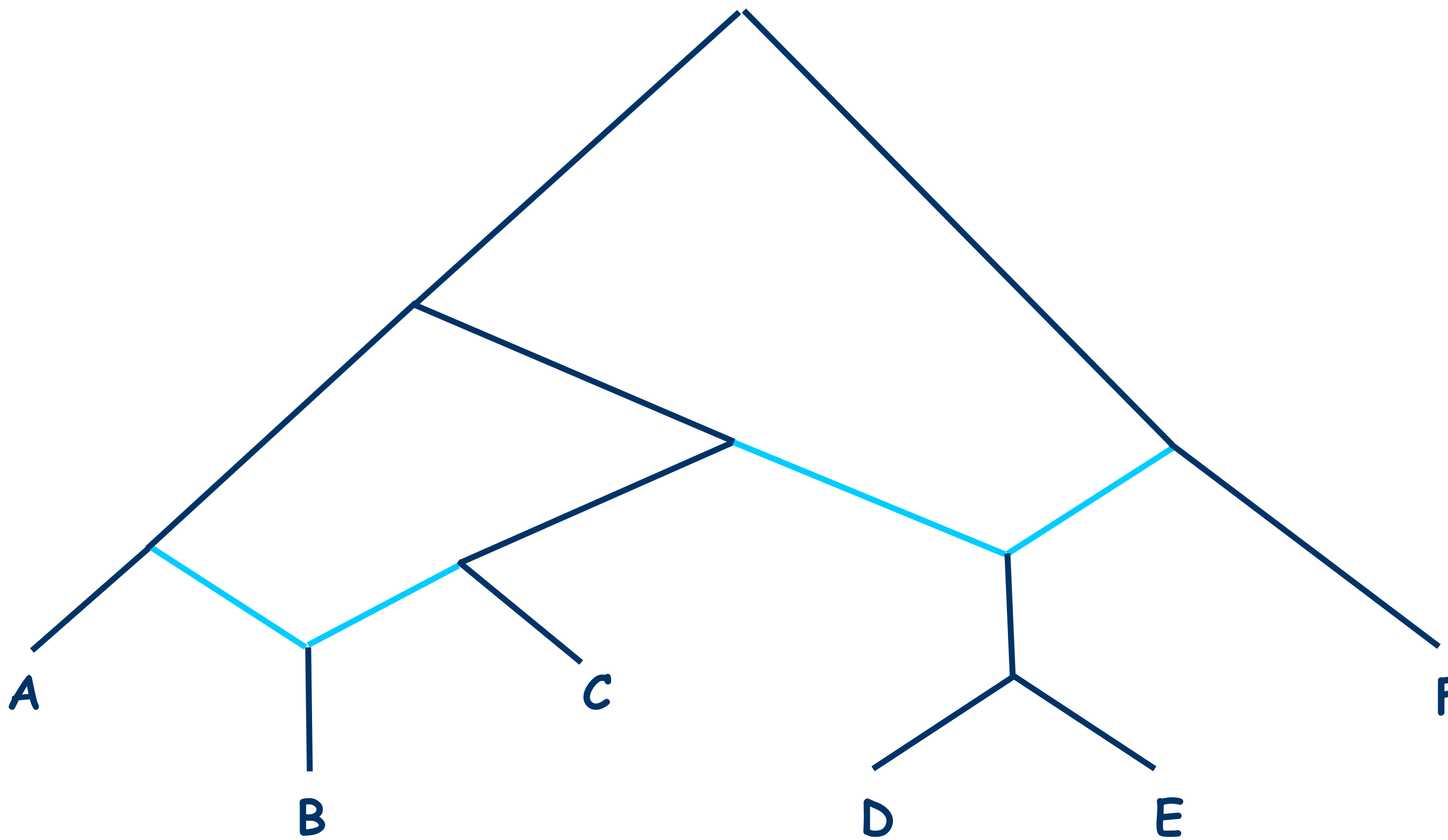
$\{B,C\} \subset \{B,C,D,E\}$



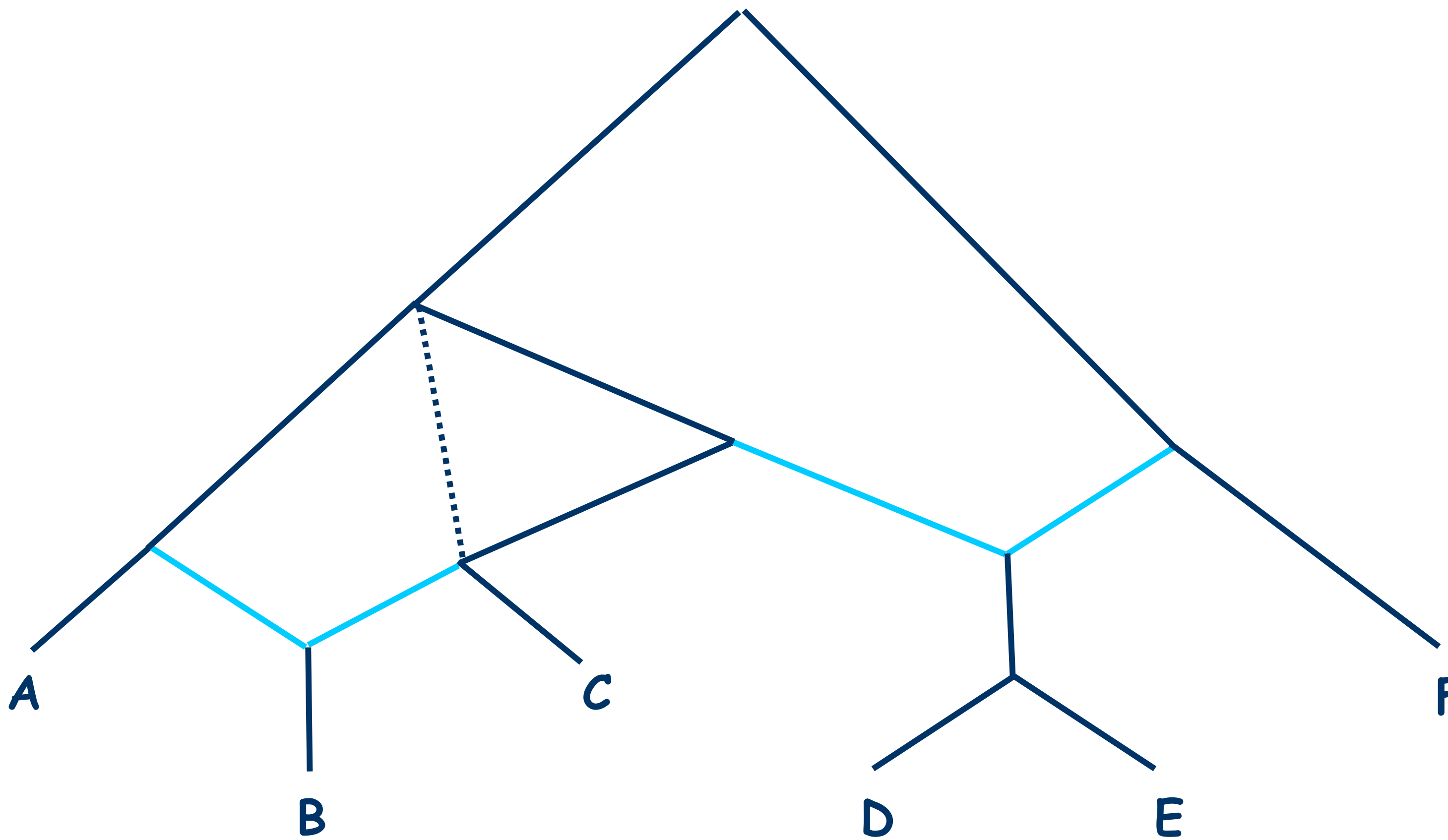
Nestedness



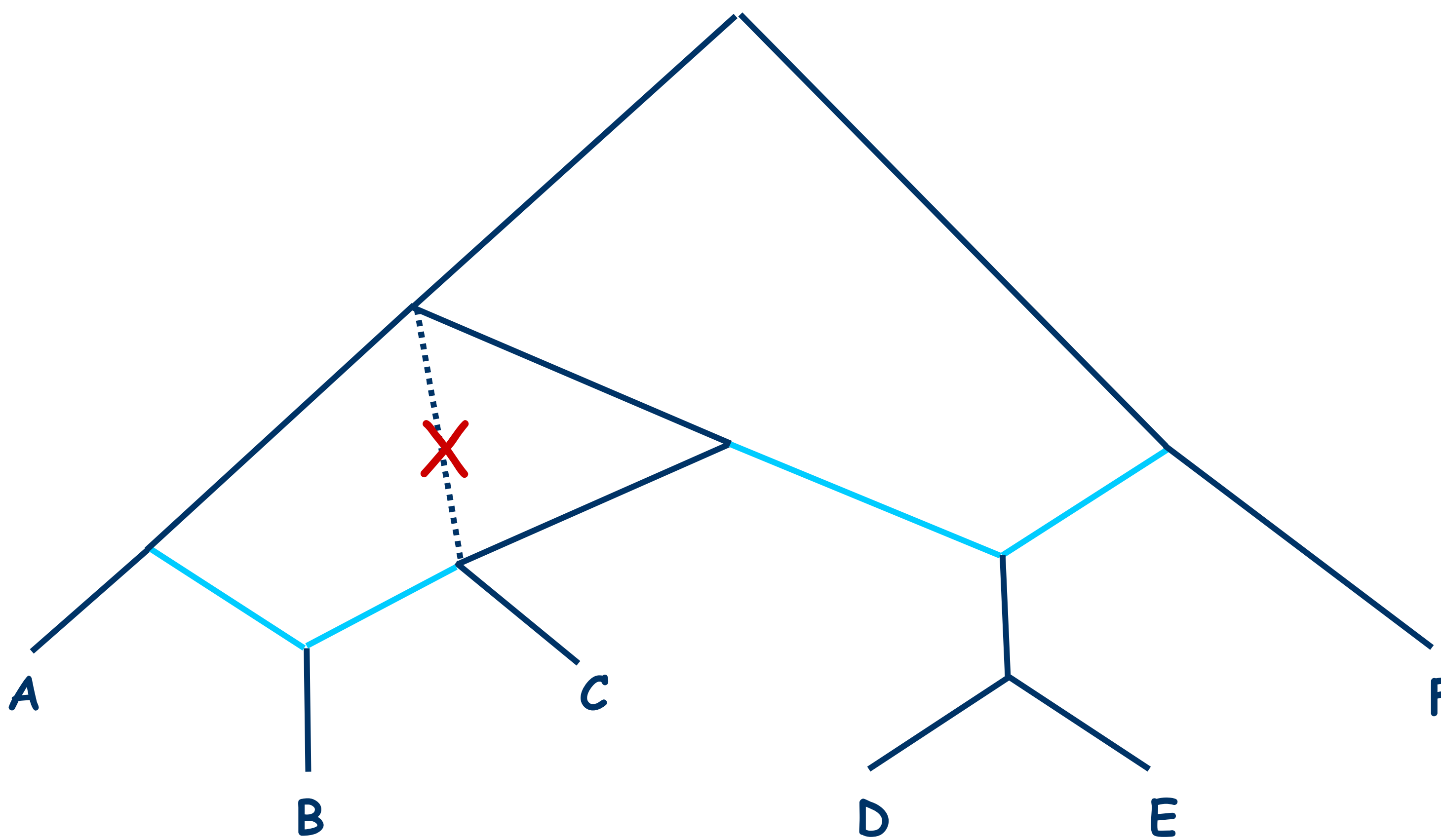
Reducedness



Reducedness



Reducedness



Displaying clusters in cluster networks

Displaying clusters in cluster networks

Every non-reticulation edge e in a cluster network defines a cluster, namely the set of labels of all nodes below e.

Displaying clusters in cluster networks

Every non-reticulation edge e in a cluster network defines a cluster, namely the set of labels of all nodes below e.

We call this the „hardwired interpretation".

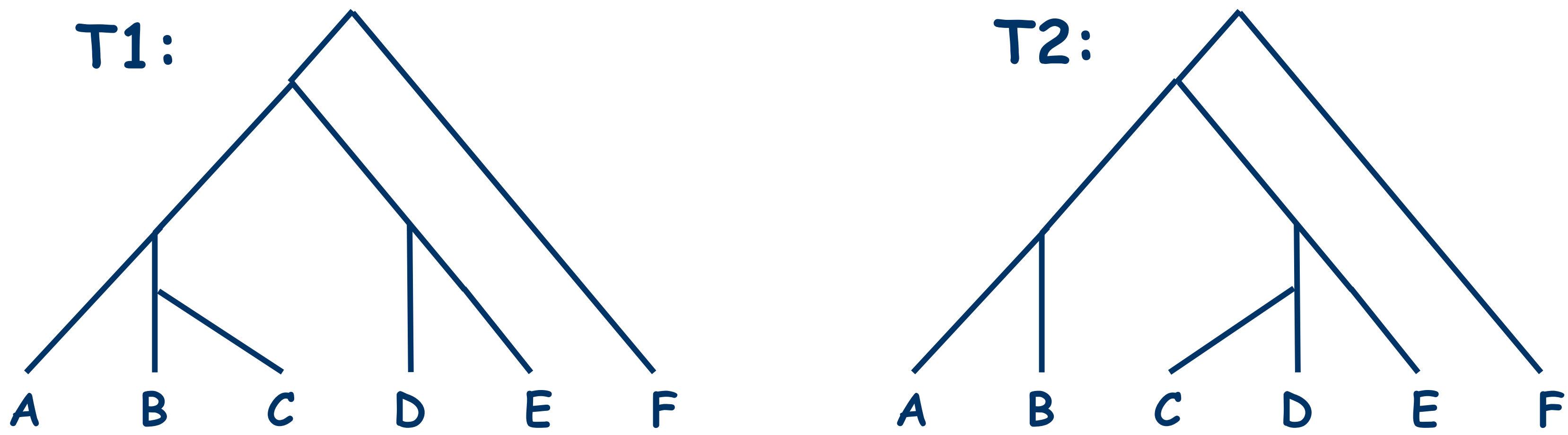
Displaying clusters in cluster networks

Every non-reticulation edge e in a cluster network defines a cluster, namely the set of labels of all nodes below e.

We call this the „hardwired interpretation".

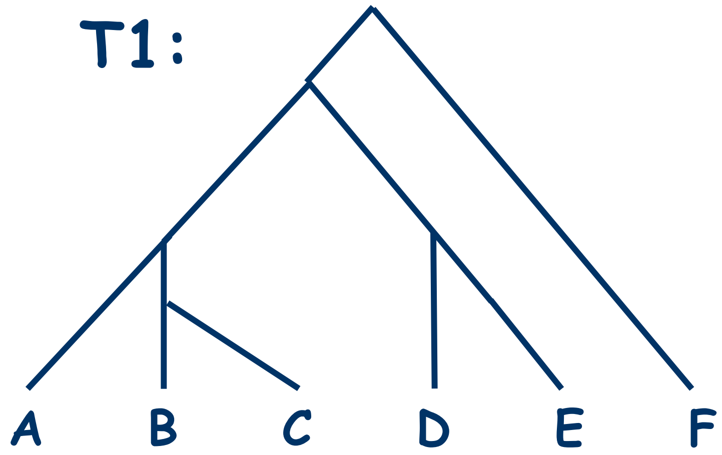
In contrast we define the „softwired interpretation" where we may switch reticulation edges on or off.

Hardwired / Softwired

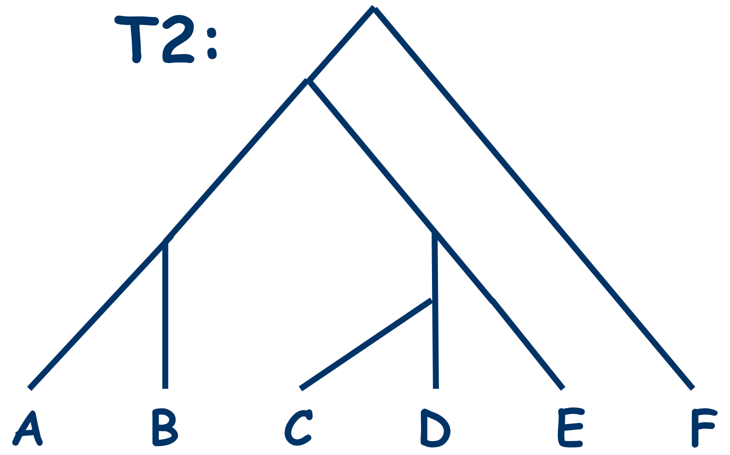


Hardwired / Softwired

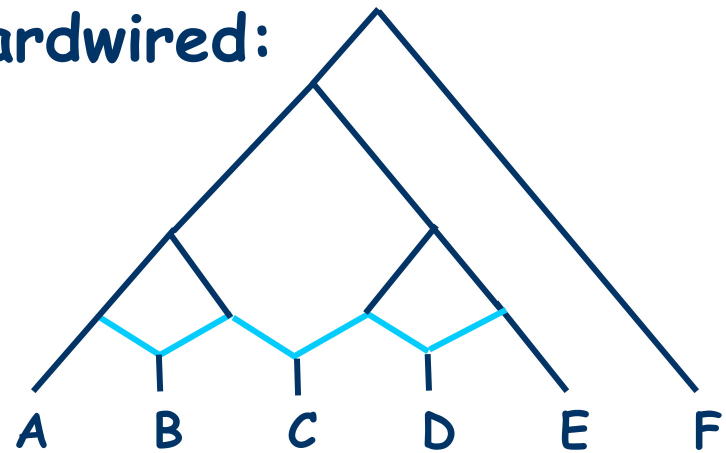
T1:



T2:

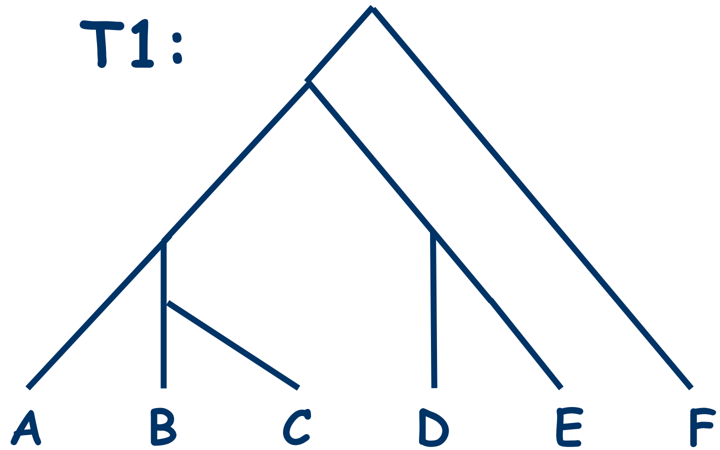


Hardwired:

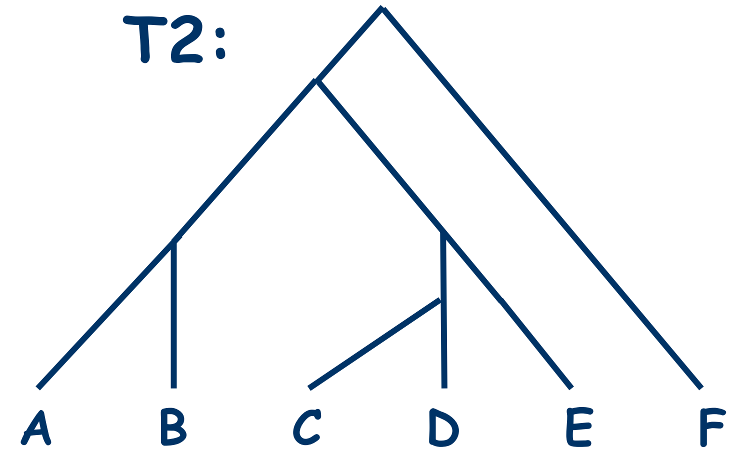


Hardwired / Softwired

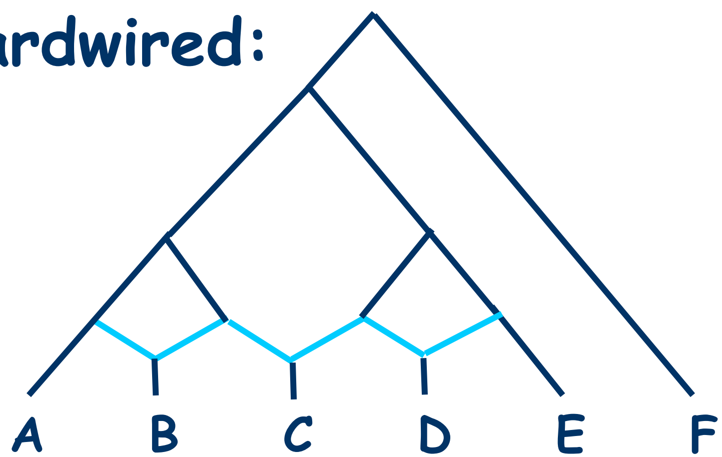
T1:

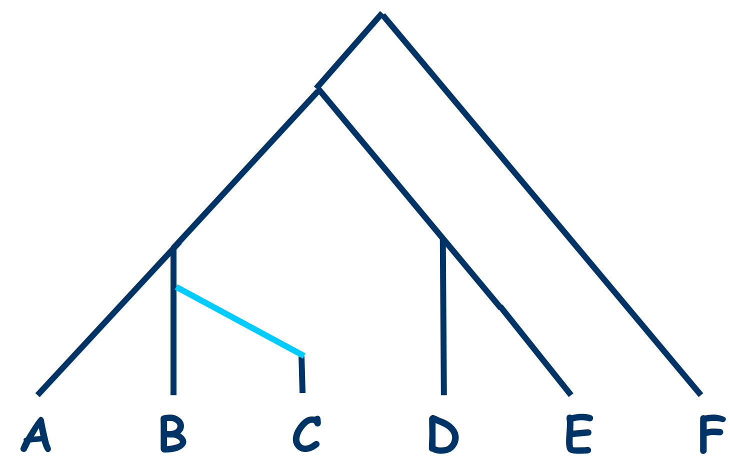


T2:



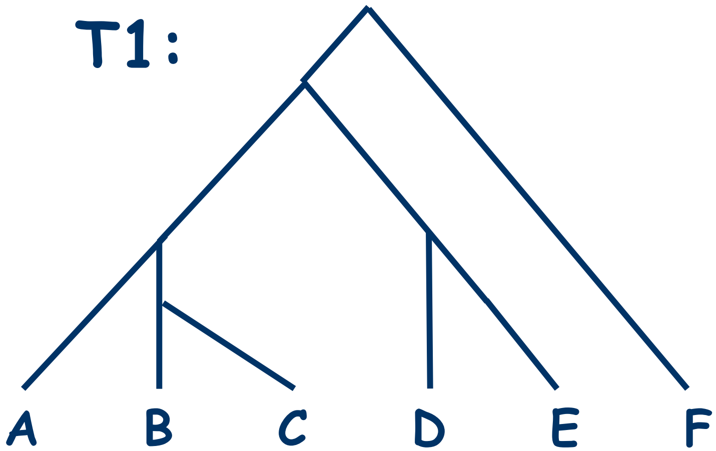
Hardwired:



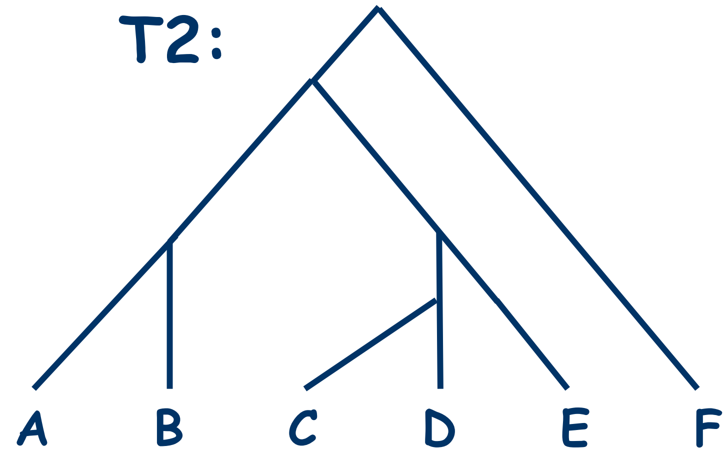


Hardwired / Softwired

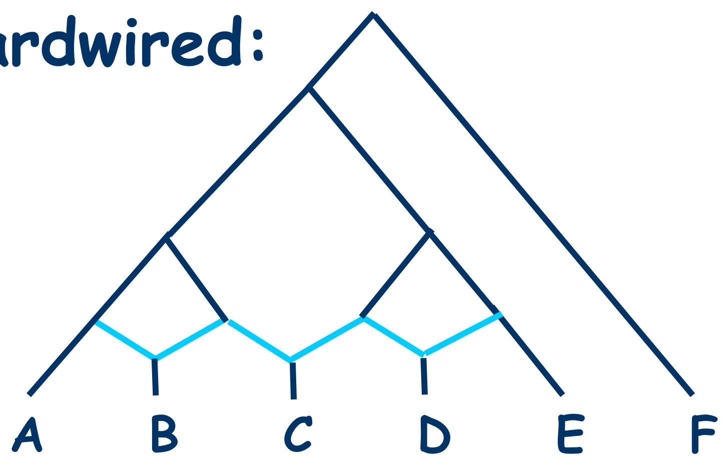
T1:

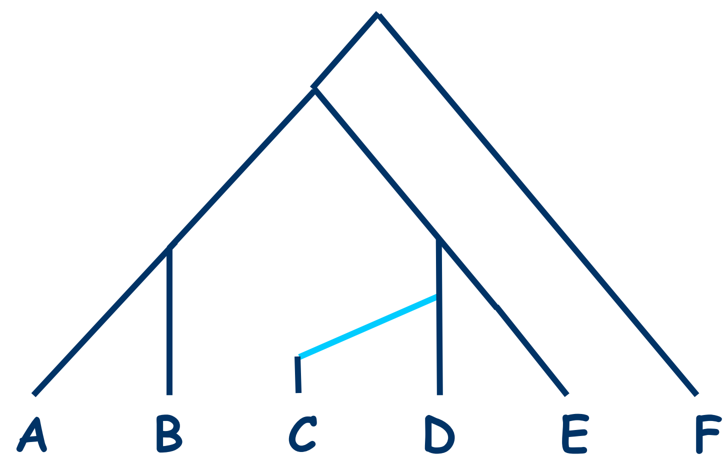


T2:

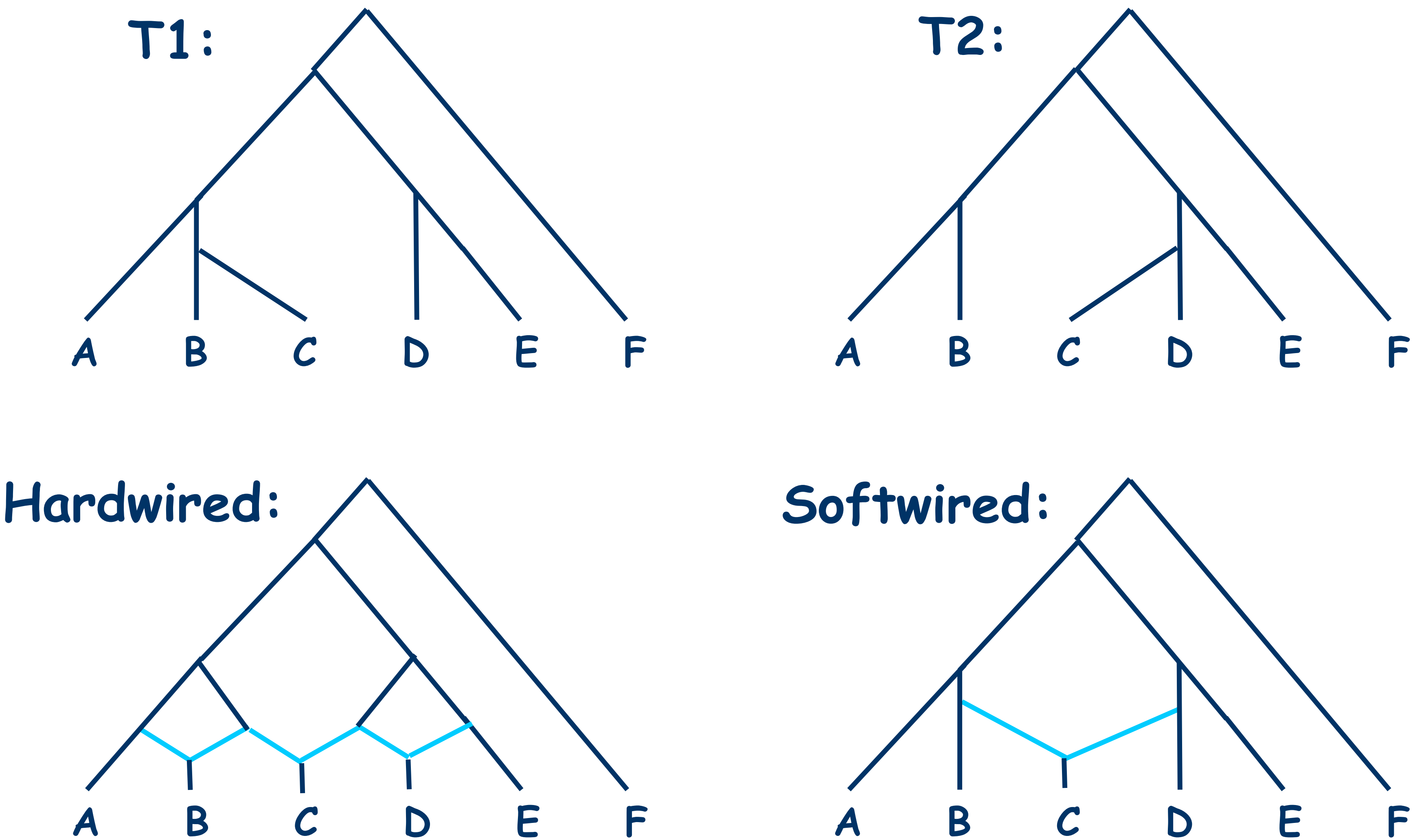


Hardwired:





Hardwired / Softwired



Hardwired / Softwired

- Cluster network, "Hardwired": blue edges always on

- Reticulate network, "Softwired": For any reticulation, any blue edge can be on or off

Hardwired / Softwired

- Cluster network, "Hardwired": blue edges always on
  - More reticulations, "looks complicated"

- Reticulate network, "Softwired": For any reticulation, any blue edge can be on or off

Hardwired / Softwired

- Cluster network, "Hardwired": blue edges always on
  - More reticulations, "looks complicated"

- Reticulate network, "Softwired": For any reticulation, any blue edge can be on or off
  - Number of reticulations can be minimized

Hardwired / Softwired

- Cluster network, "Hardwired": blue edges always on
  - More reticulations, "looks complicated"

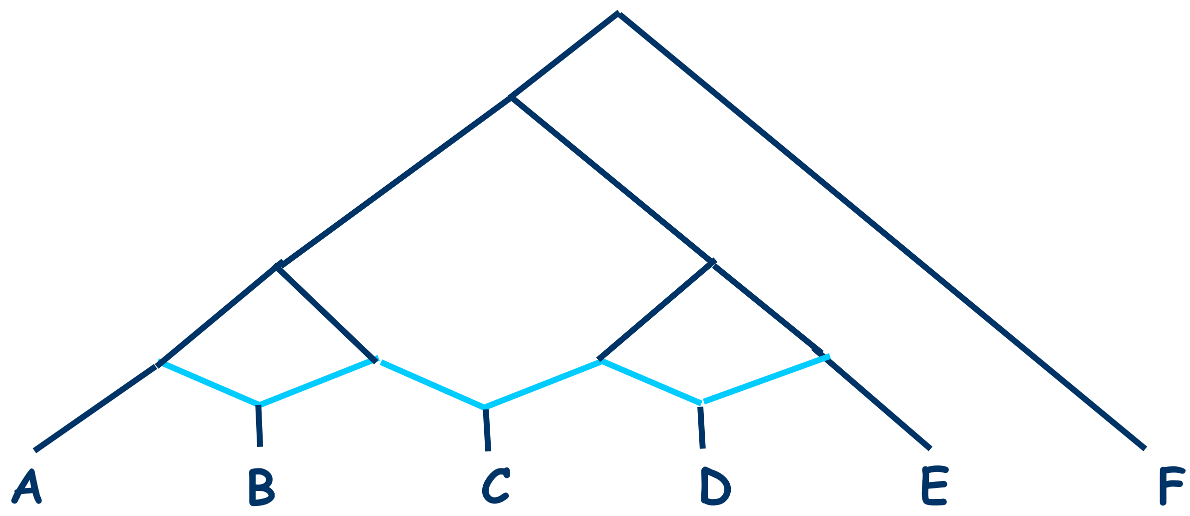
- Reticulate network, "Softwired": For any reticulation, any blue edge can be on or off
  - Number of reticulations can be minimized
  - Computationally hard

Hardwired / Softwired

- Cluster network, "Hardwired": blue edges always on
  - More reticulations, "looks complicated"
  - Canonical network, computationally easy

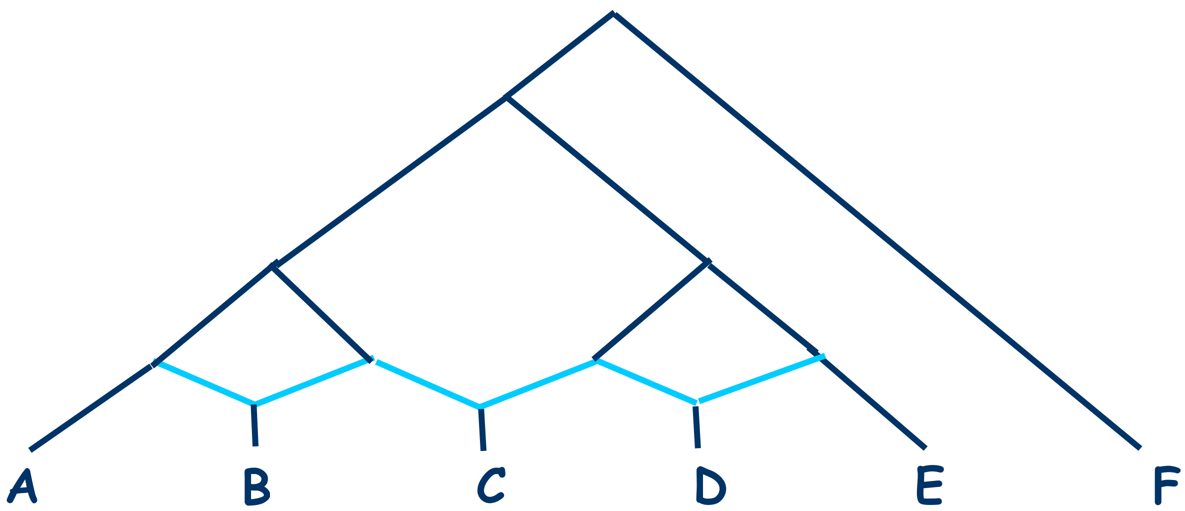
- Reticulate network, "Softwired": For any reticulation, any blue edge can be on or off
  - Number of reticulations can be minimized
  - Computationally hard

Lowest Single Ancestor: LSA



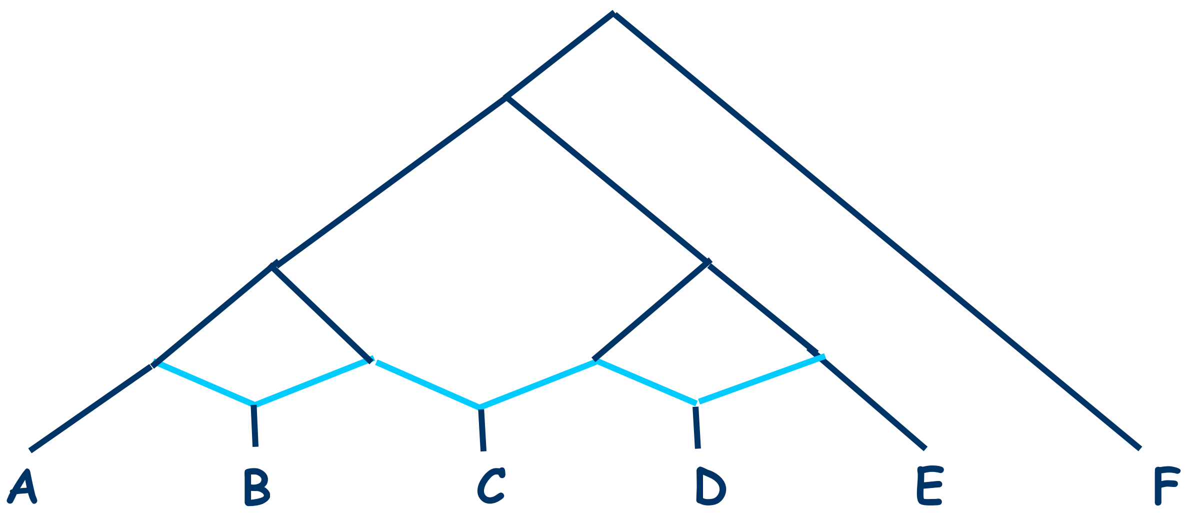
Lowest Single Ancestor: LSA

- In trees: LCA



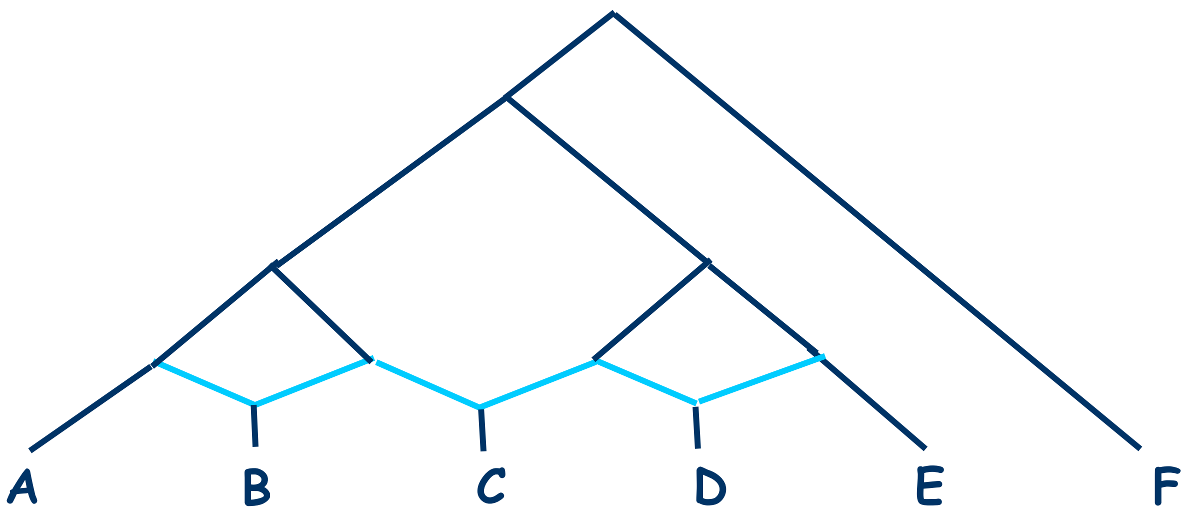
Lowest Single Ancestor: LSA

- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:



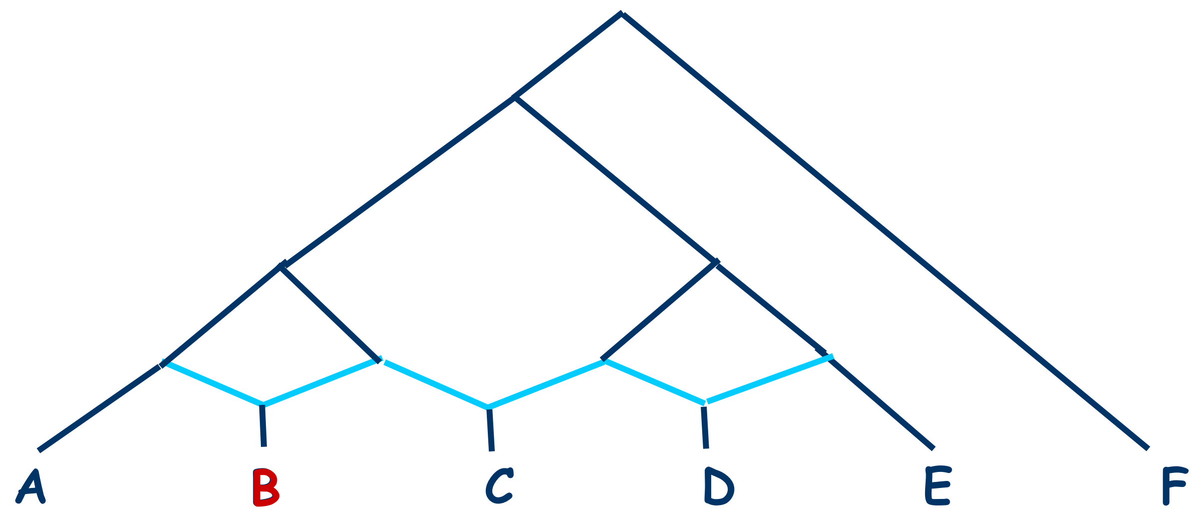
Lowest Single Ancestor: LSA

- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:
  LSA(S) = Lowest node that is on every path from the root to one of the nodes in S.



Lowest Single Ancestor: LSA

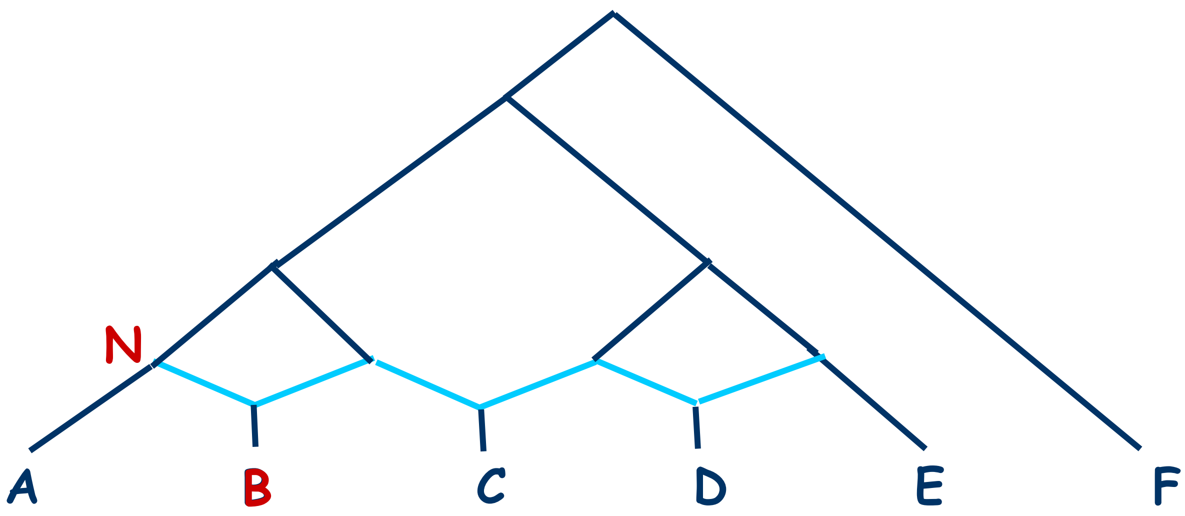
- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:
  LSA(S) = Lowest node that is on every path from the root to one of the nodes in S.



Lowest Single Ancestor: LSA

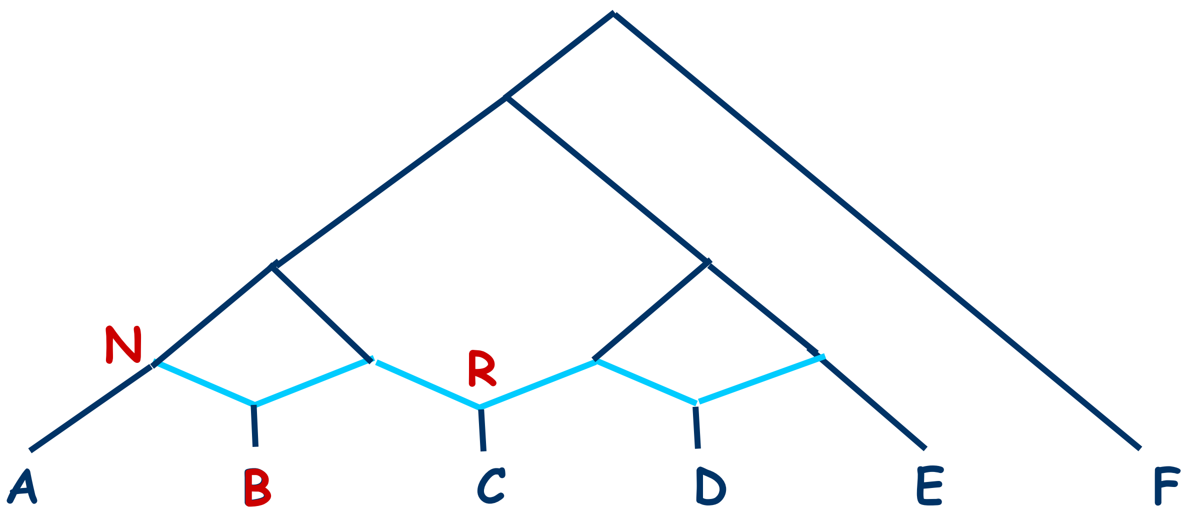
- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:

  LSA(S) = Lowest node that is on every path from the root to one of the nodes in S.



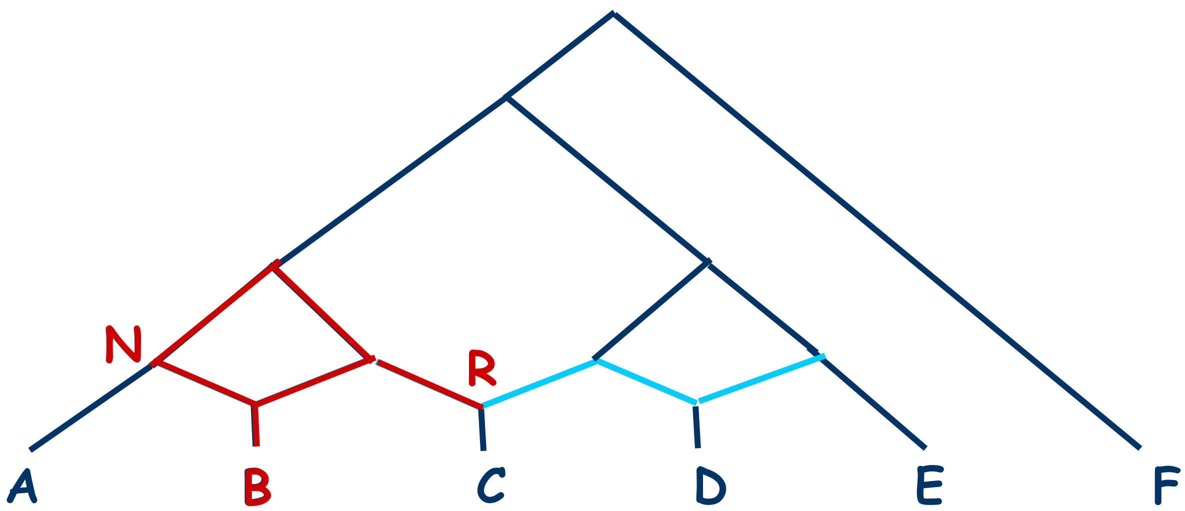
Lowest Single Ancestor: LSA

- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor: LSA(S) = Lowest node that is on every path from the root to one of the nodes in S.



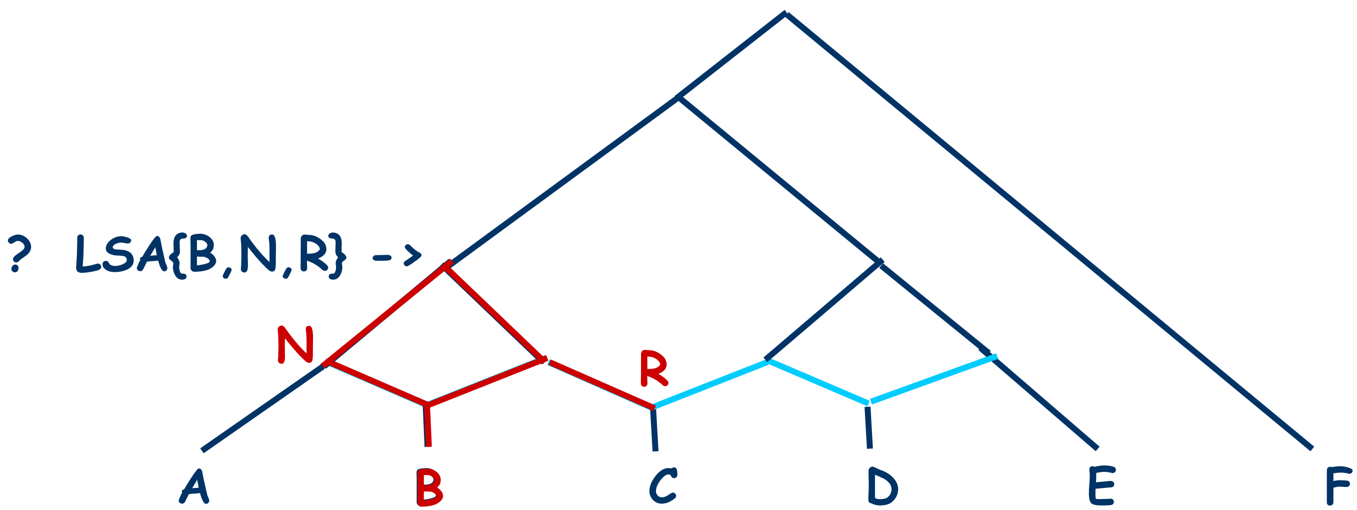
Lowest Single Ancestor: LSA

- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:
  LSA(S) = Lowest node that is on every path
  from the root to one of the nodes in S.



Lowest Single Ancestor: LSA

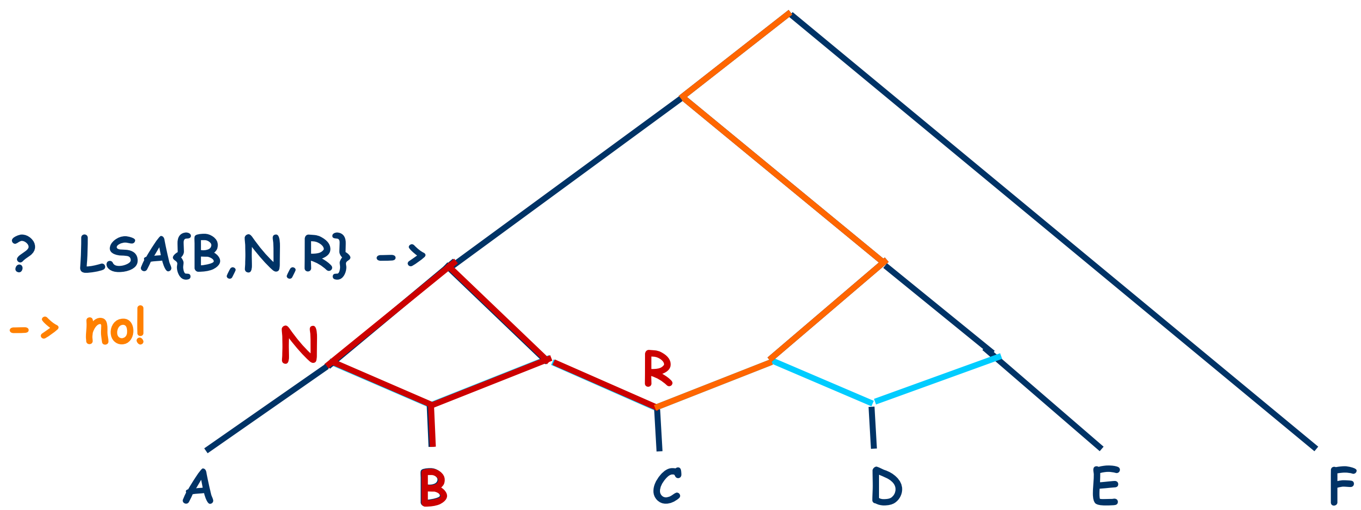
- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:
  LSA(S) = Lowest node that is on every path
  from the root to one of the nodes in S.



Lowest Single Ancestor: LSA

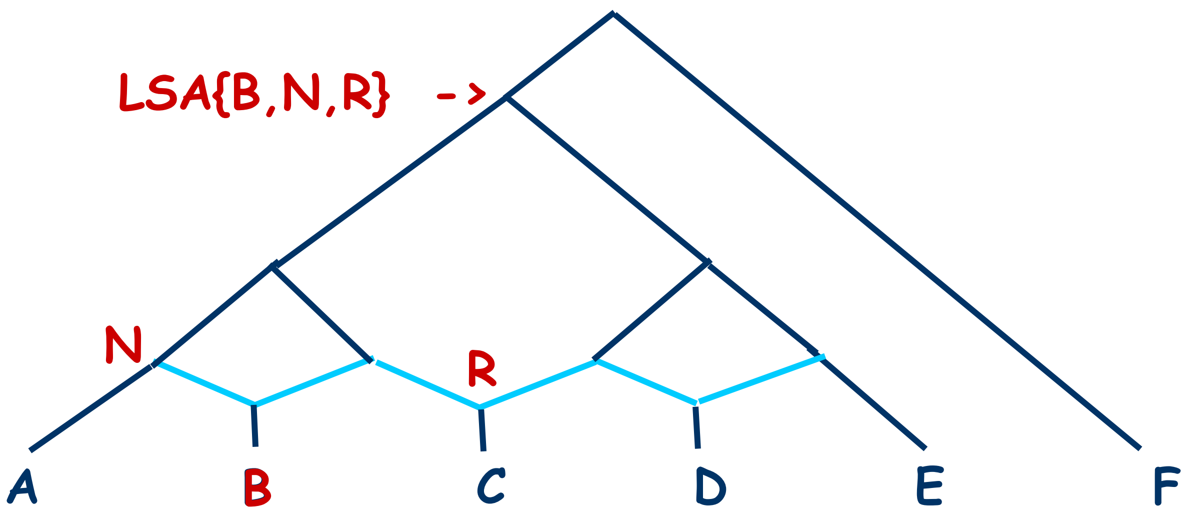
- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:

  LSA(S) = Lowest node that is on every path from the root to one of the nodes in S.



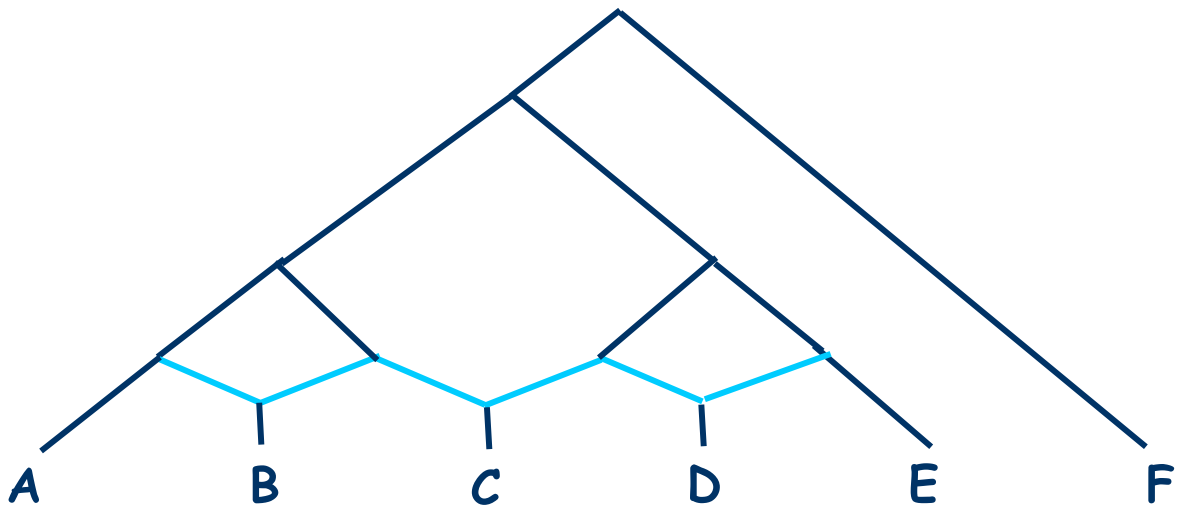
Lowest Single Ancestor: LSA

- In trees: LCA
- In cluster networks: LSA = Lowest Single Ancestor:
  LSA(S) = Lowest node that is on every path
  from the root to one of the nodes in S.



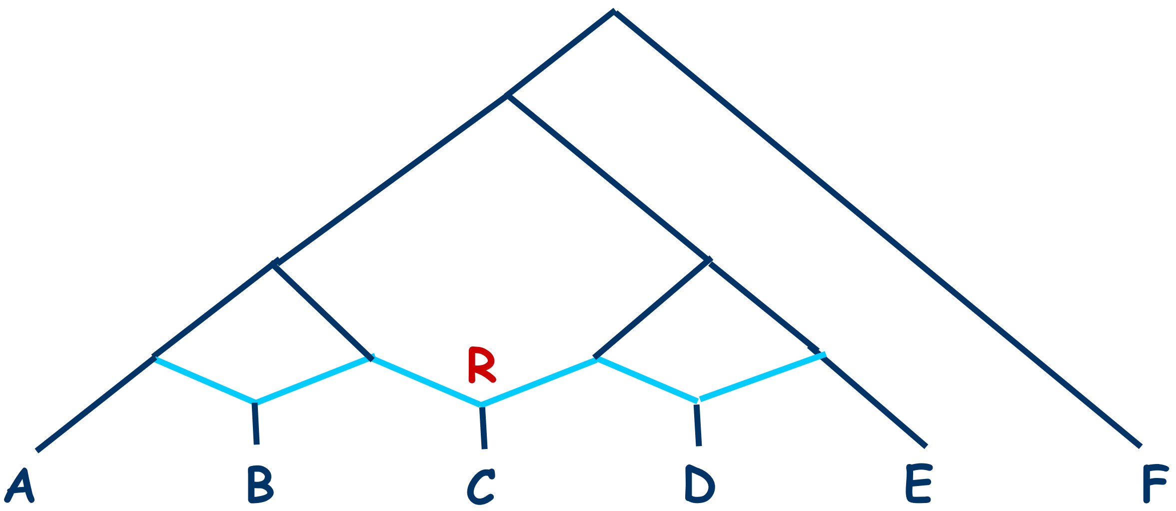
LSA consensus tree

LSA of a reticulation



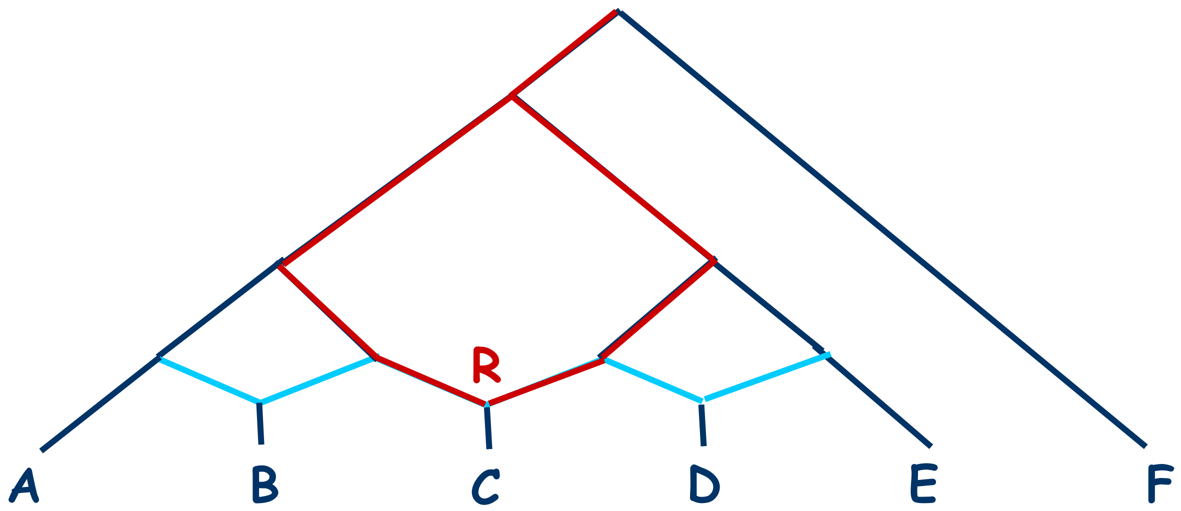
LSA consensus tree

LSA of a reticulation



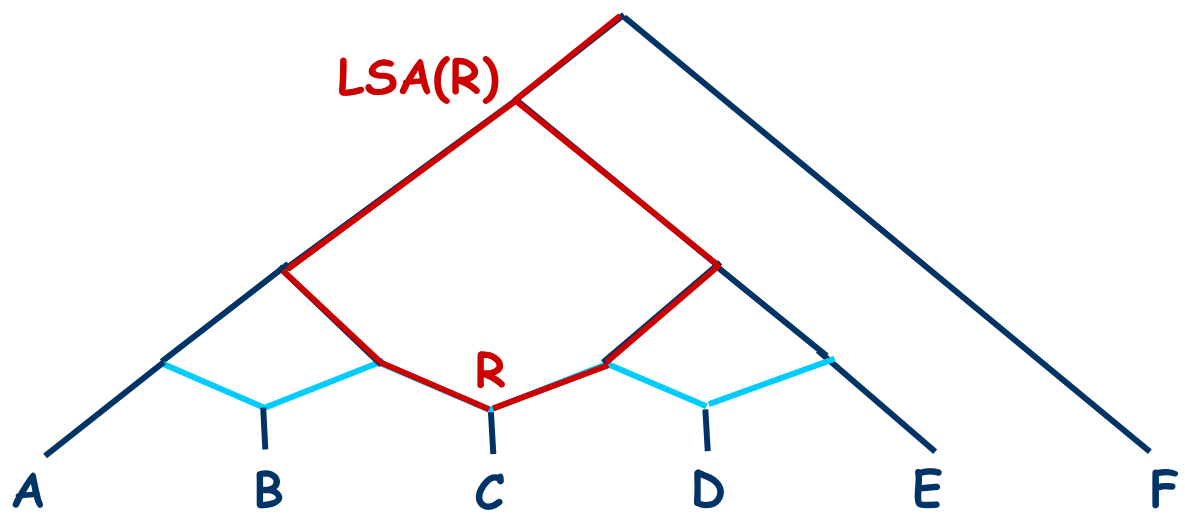
LSA consensus tree

LSA of a reticulation



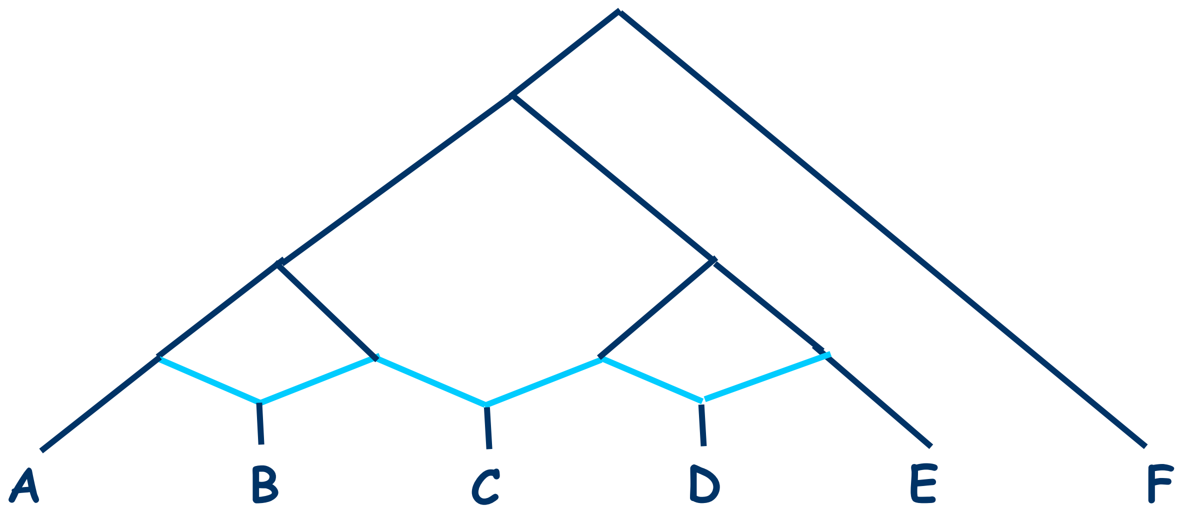
LSA consensus tree

LSA of a reticulation



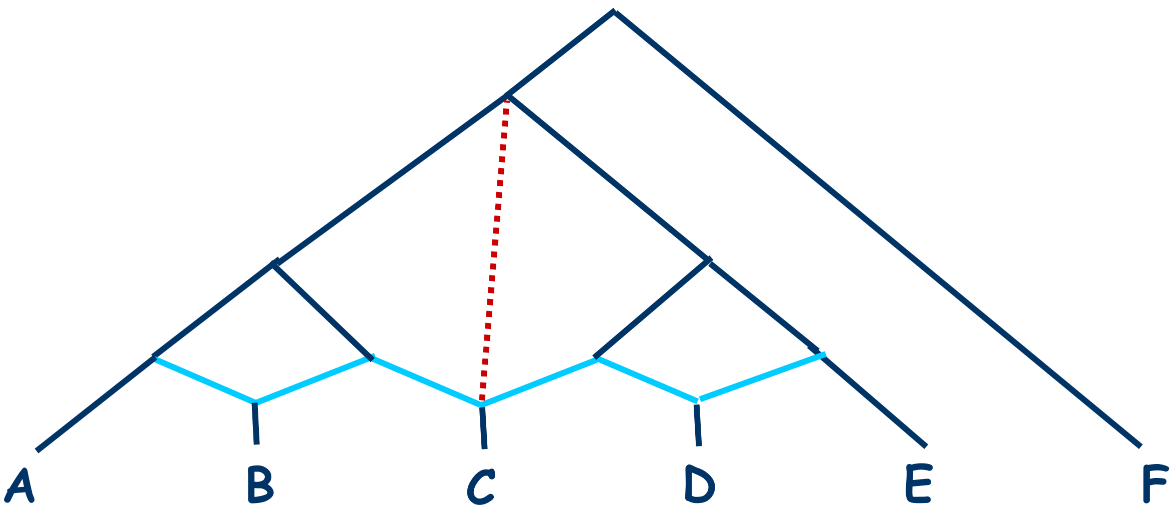
LSA consensus tree

LSA of a reticulation



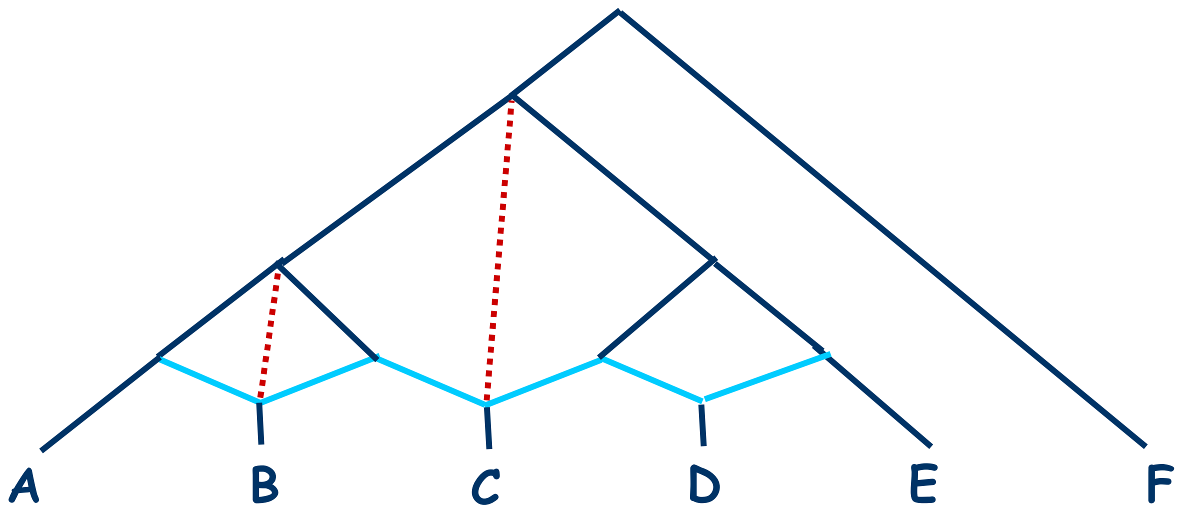
LSA consensus tree

LSA of a reticulation



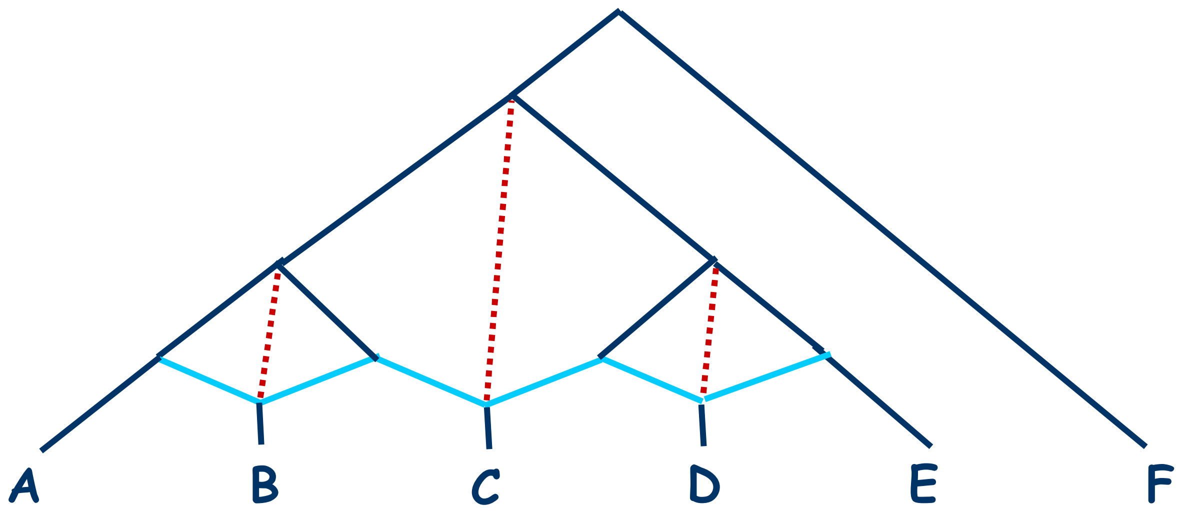
LSA consensus tree

LSA of a reticulation



LSA consensus tree

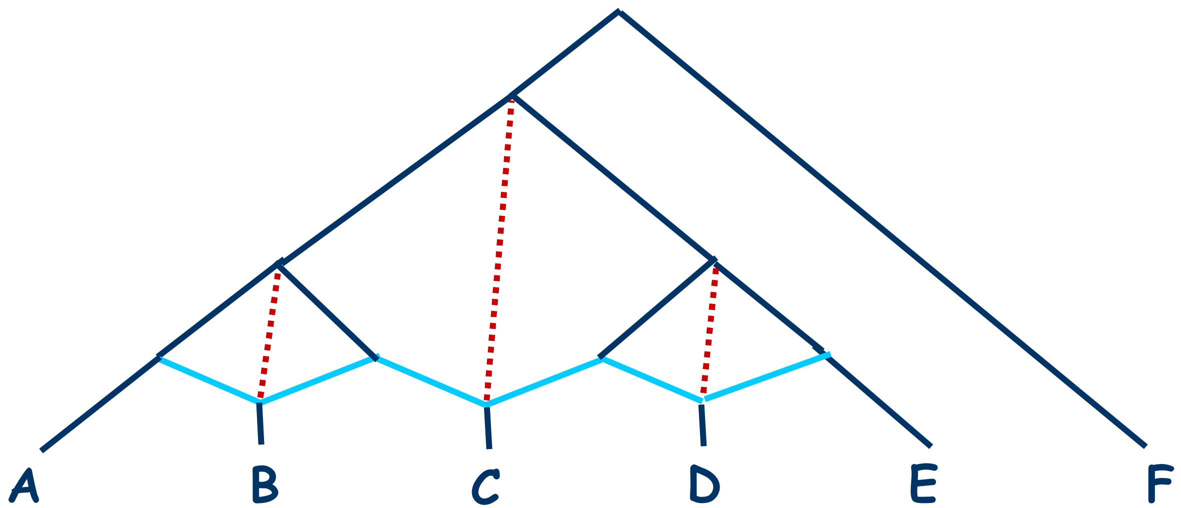
LSA of a reticulation



LSA consensus tree

LSA of a reticulation

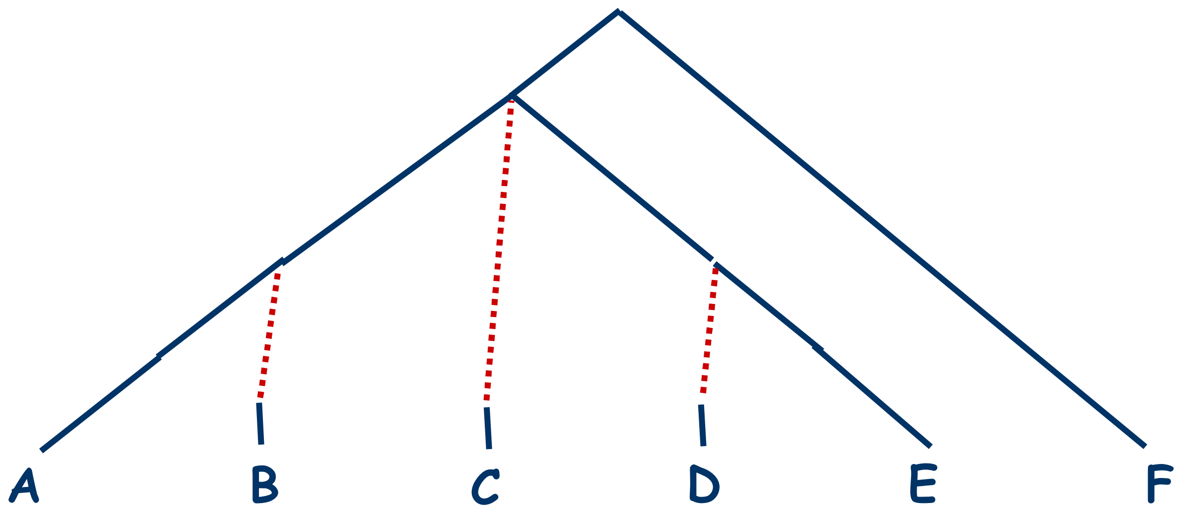
-> new consensus method!



LSA consensus tree

LSA of a reticulation

-> new consensus method!



LSA Consensus

LSA Consensus

- New consensus method

LSA Consensus

- New consensus method
- LSA is lowest node for which all input trees agree that it is an ancestor

LSA Consensus

- New consensus method
- LSA is lowest node for which all input trees agree that it is an ancestor
- Easy to compute

LSA Consensus

- New consensus method
- LSA is lowest node for which all input trees agree that it is an ancestor
- Easy to compute
- Different from all other consensus methods (?)

LSA Consensus

- New consensus method
- LSA is lowest node for which all input trees agree that it is an ancestor
- Easy to compute
- Different from all other consensus methods (?)
- Philippe Gambette: LSA consensus = Adams consensus?

Adams Consensus

Adams Consensus

- Sets of trees T1,T2,...,Tn

Adams Consensus

- Sets of trees T1,T2,...,Tn
- Maximal clusters in Adams Consensus: non-empty intersections of maximal clusters in T1,T2,...,Tn

Adams Consensus

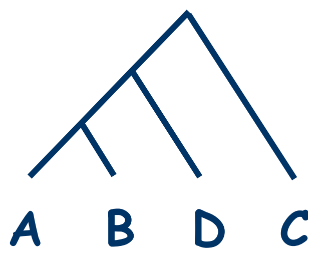
- Sets of trees T1,T2,...,Tn
- Maximal clusters in Adams Consensus: non-empty intersections of maximal clusters in T1,T2,...,Tn
- Restrict trees to maximal clusters of Adams consensus and repeat procedure recursively.

Adams vs. LSA

Question: LSA consensus = Adams consensus?

Adams vs. LSA

Question: LSA consensus = Adams consensus?

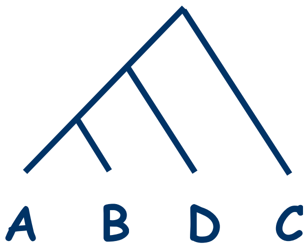


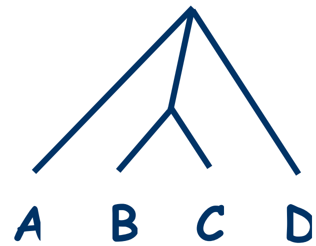
Adams vs. LSA

Question: LSA consensus = Adams consensus?

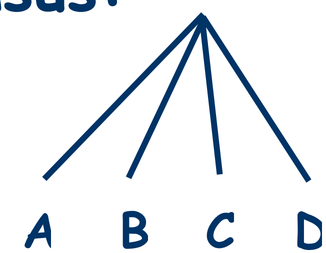
Adams vs. LSA

Question: LSA consensus = Adams consensus?





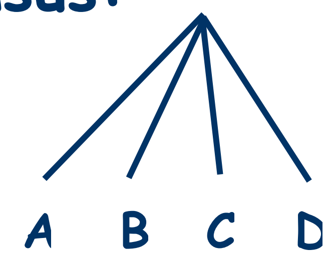
Adams consensus:



Adams vs. LSA

Question: LSA consensus = Adams consensus?

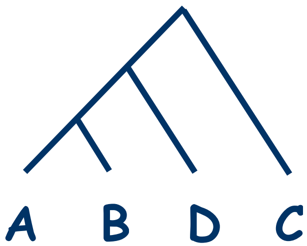
Adams consensus:

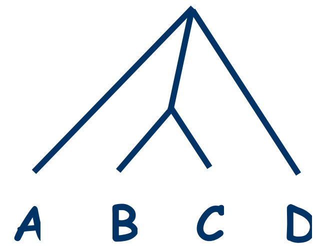


Cluster network:

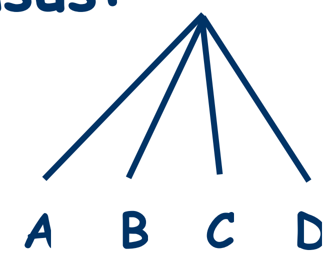
Adams vs. LSA

Question: LSA consensus = Adams consensus?

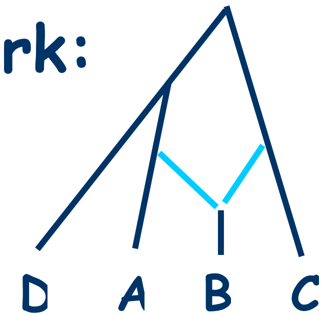




Adams consensus:

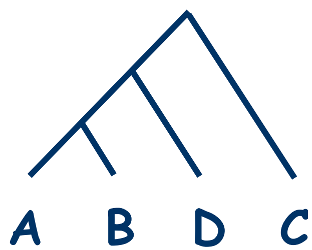


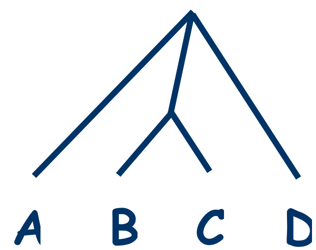
Cluster network:



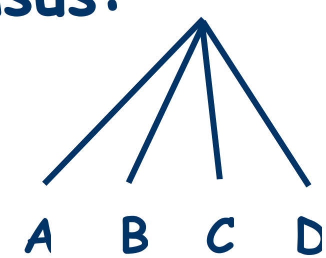
Adams vs. LSA

Question: LSA consensus = Adams consensus?

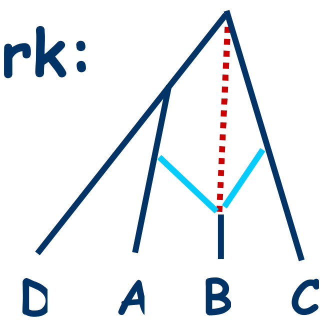




Adams consensus:

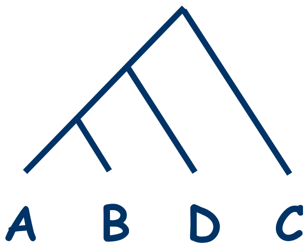


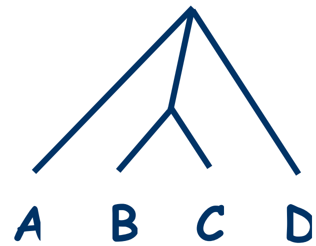
Cluster network:



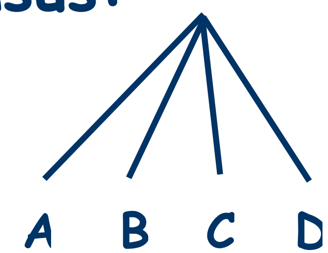
Adams vs. LSA

Question: LSA consensus = Adams consensus?

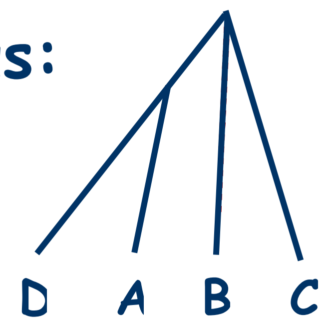




Adams consensus:

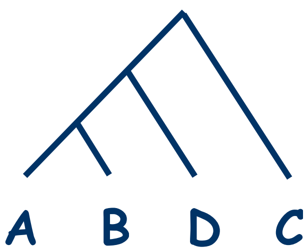


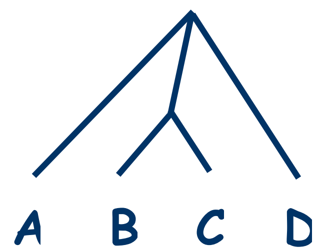
LSA consensus:



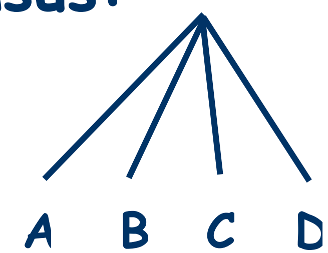
Adams vs. LSA

Question: LSA consensus = Adams consensus?

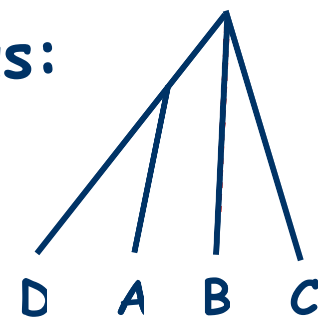




Adams consensus:



LSA consensus:



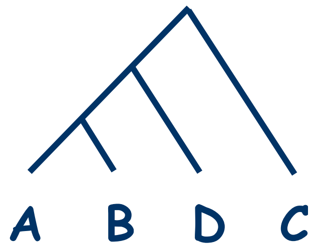
-> LSA consensus ≠ Adams consensus

Adams vs. LSA

Question: LSA cons. refinement of Adams cons.?

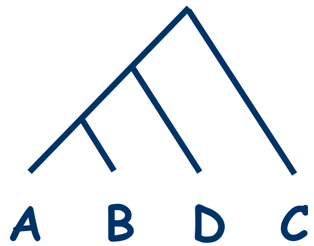
Adams vs. LSA

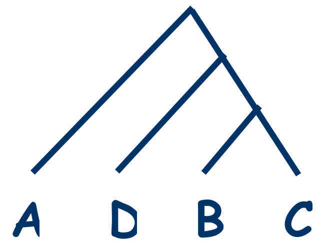
Question: LSA cons. refinement of Adams cons.?



Adams vs. LSA

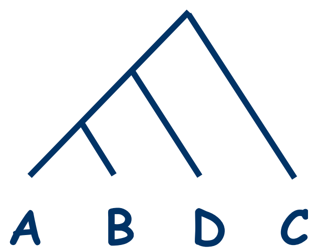
Question: LSA cons. refinement of Adams cons.?



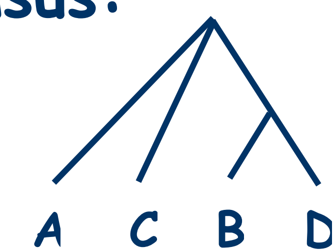


Adams vs. LSA

Question: LSA cons. refinement of Adams cons.?

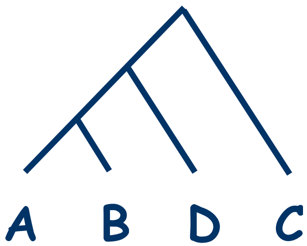


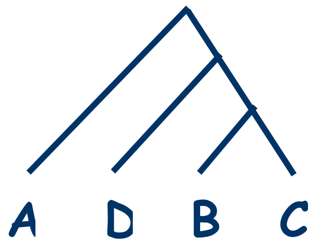
Adams consensus:



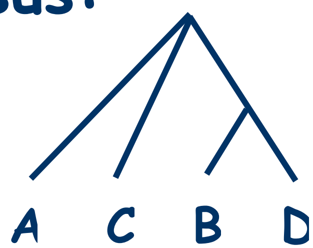
Adams vs. LSA

Question: LSA cons. refinement of Adams cons.?

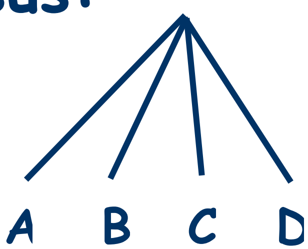




Adams consensus:



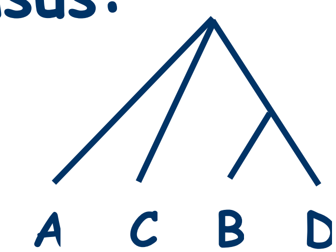
LSA consensus:



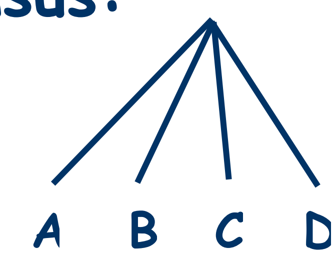
Adams vs. LSA

Question: LSA cons. refinement of Adams cons.?

Adams consensus:



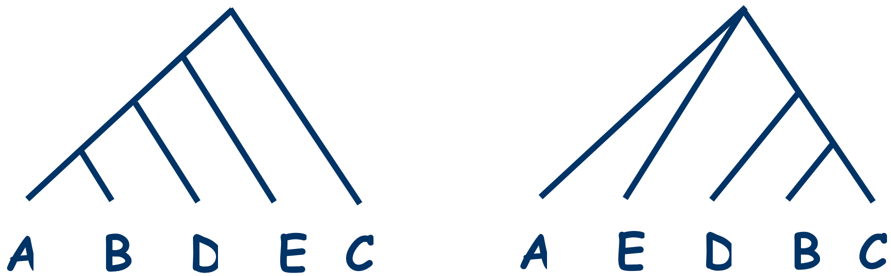
LSA consensus:



-> LSA not a refinement

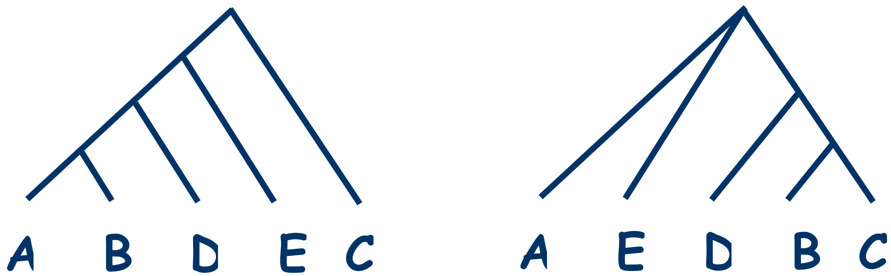
Adams vs. LSA

Combination of two examples:

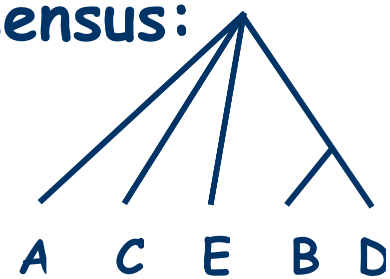


Adams vs. LSA

Combination of two examples:

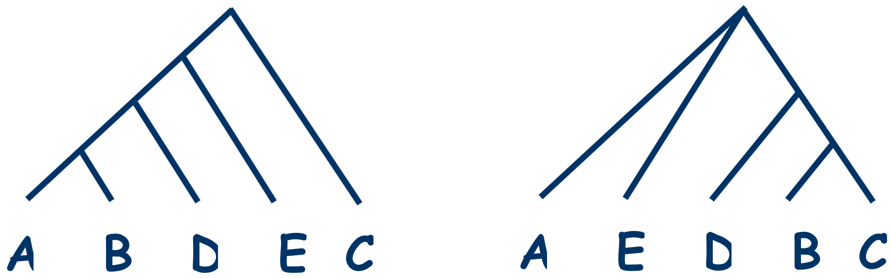


Adams consensus:

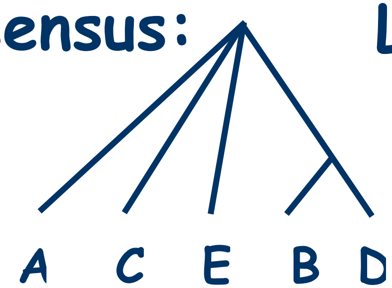


Adams vs. LSA

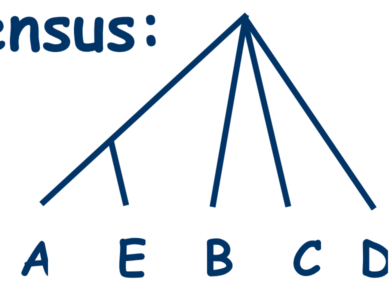
Combination of two examples:



Adams consensus:

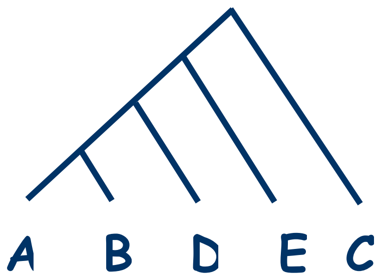


LSA consensus:

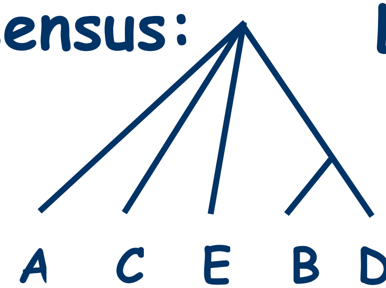


Adams vs. LSA

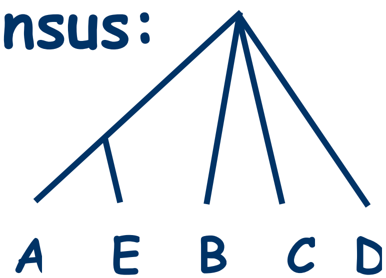
Combination of two examples:



Adams consensus:



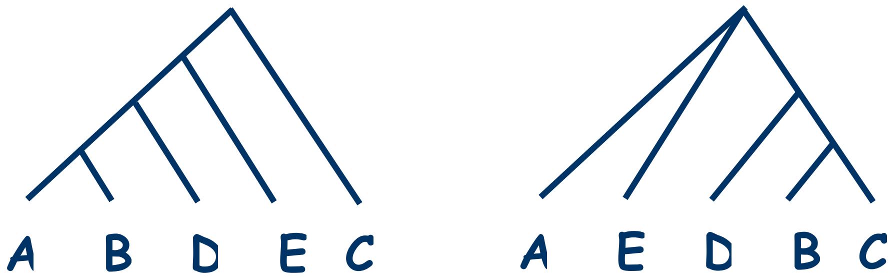
LSA consensus:



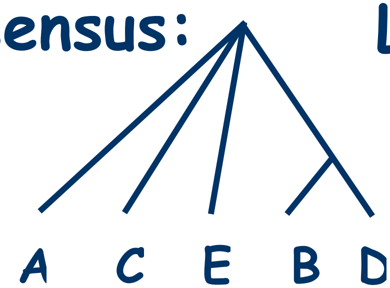
-> consensus trees are compatible

Adams vs. LSA

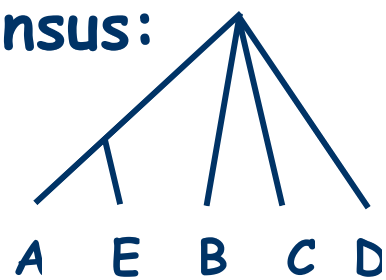
Combination of two examples:



Adams consensus:



LSA consensus:



-> consensus trees are compatible

Is this always true? -> Open Question

Coming soon: Dendroscope 2.0

- Trees and networks
- Cluster networks
- LSA consensus

Dendroscope

by Daniel H. Huson

with contributions from Tobias Dezulian, Markus Franz, Christian Rausch, Daniel C. Richter and Regula Rupp

www-ab.informatik.uni-tuebingen.de/software/dendroscope