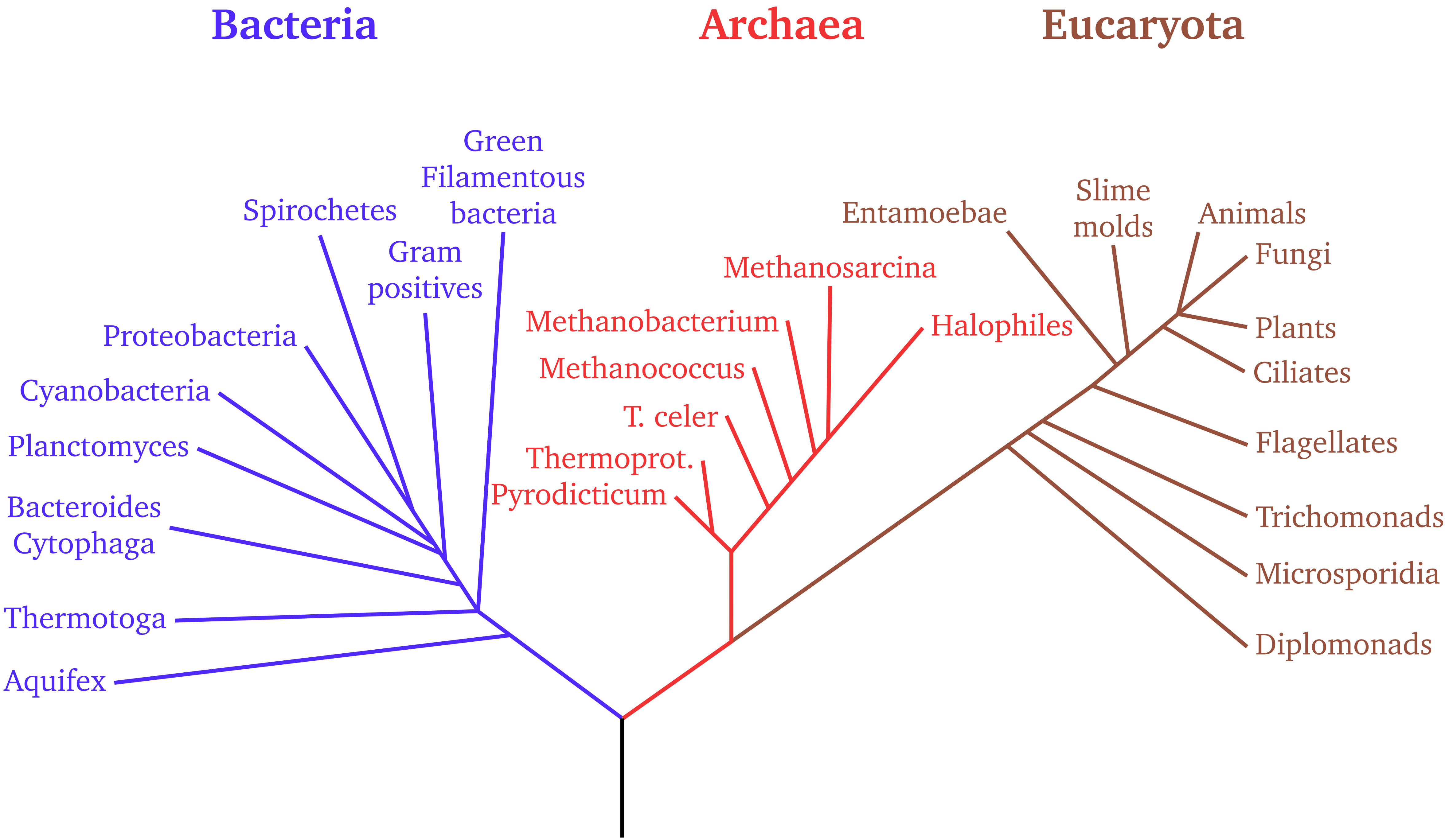
Comparing Phylogenies

Kernelization, Depth-Bounded Search and Beyond

Norbert Zeh

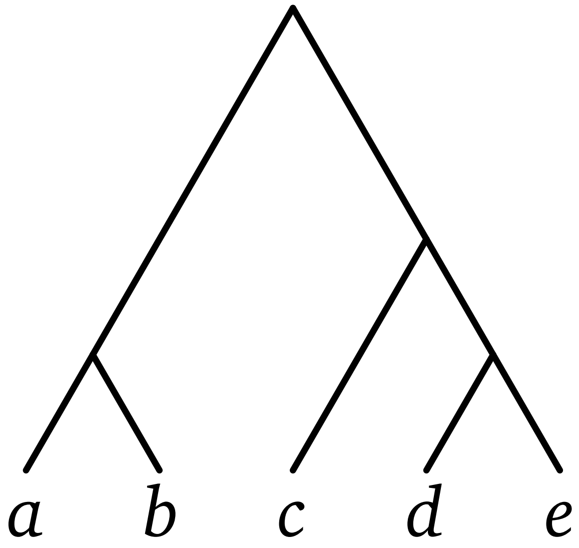
Dalhousie University

Phylogenetic Trees



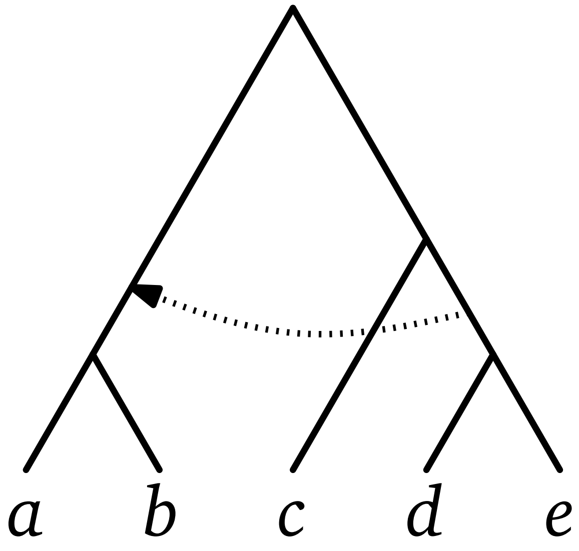
Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)



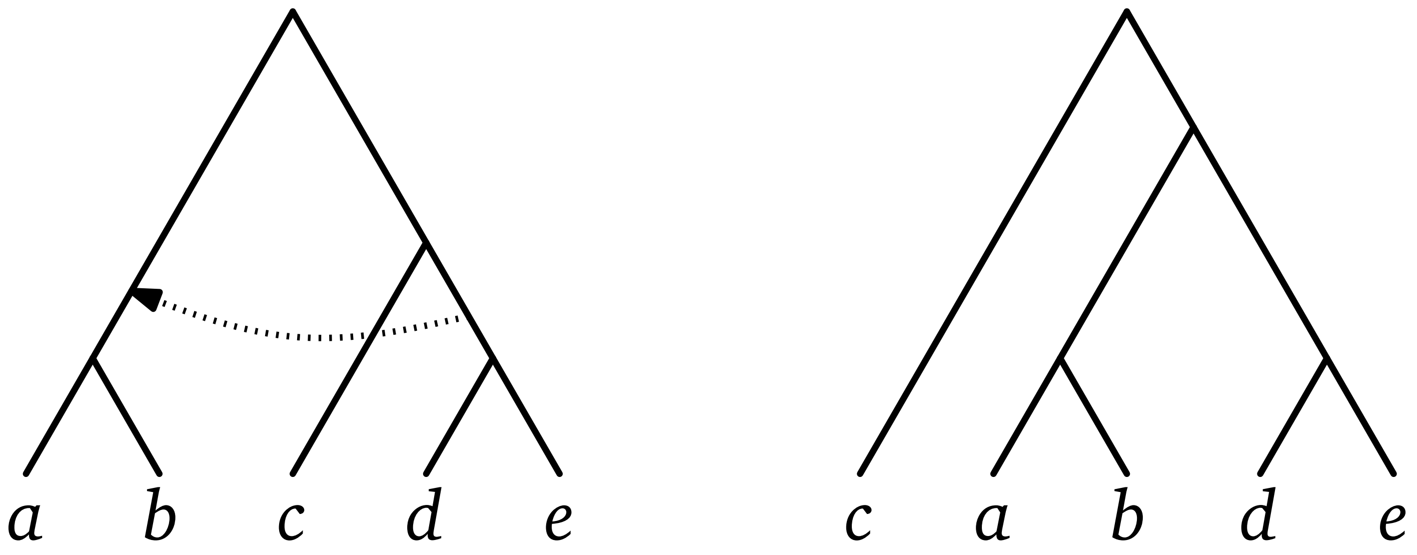
Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)



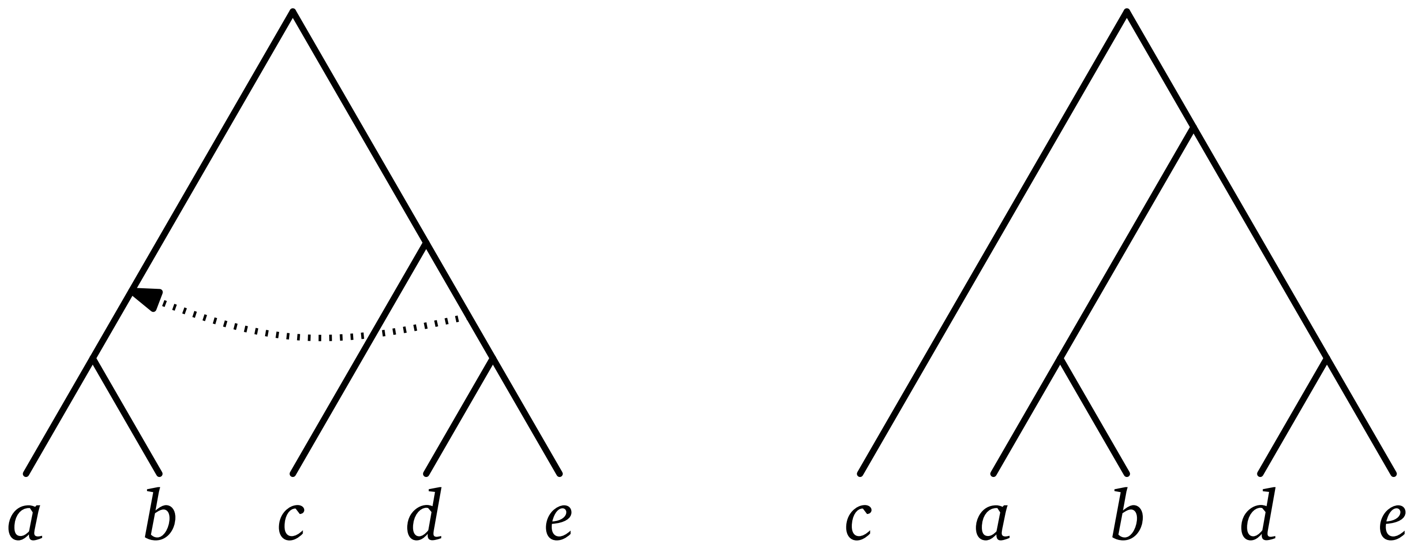
Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)

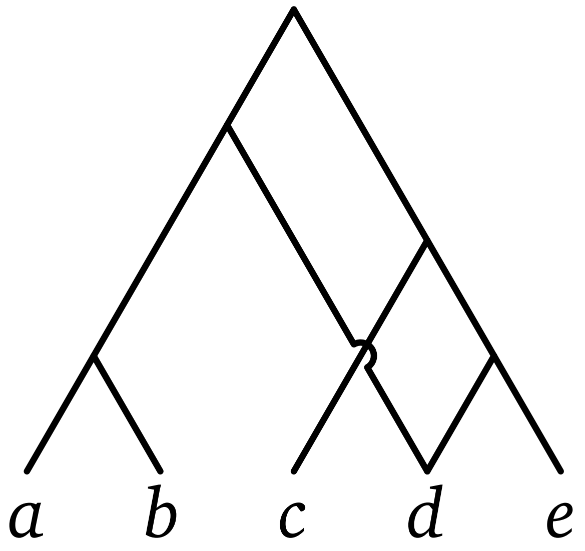


Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)

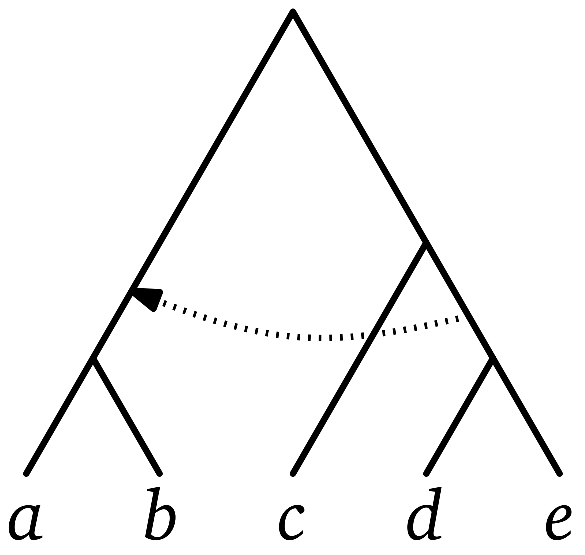


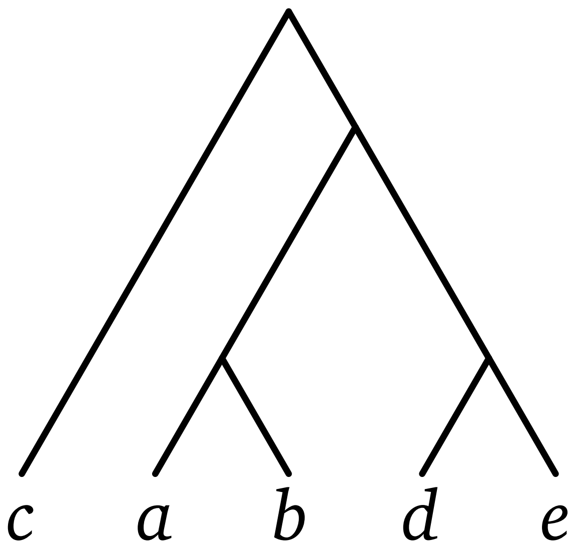
Hybridization



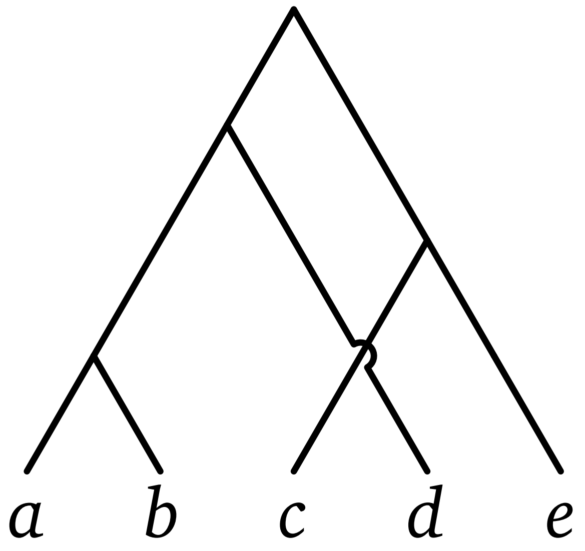
Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)



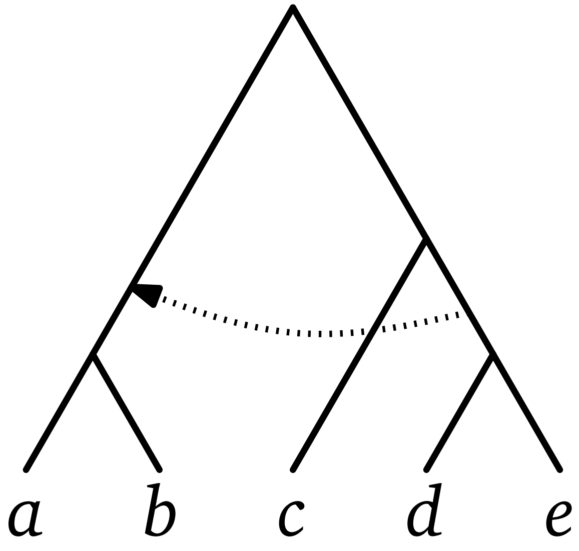


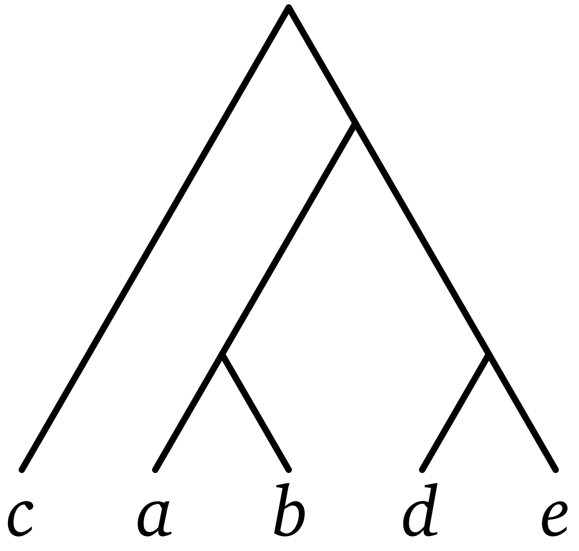
Hybridization



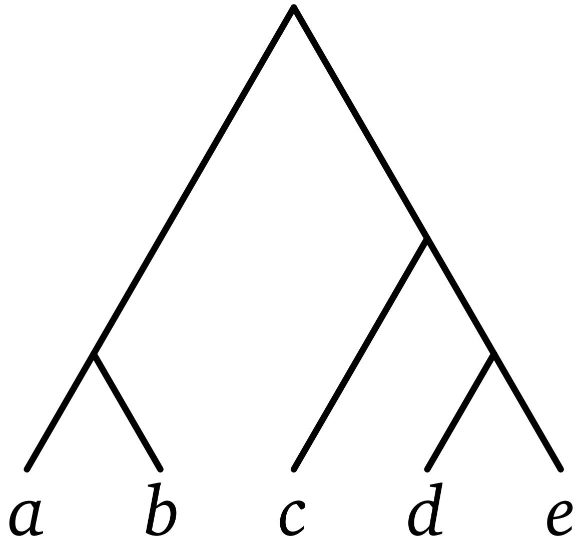
Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)



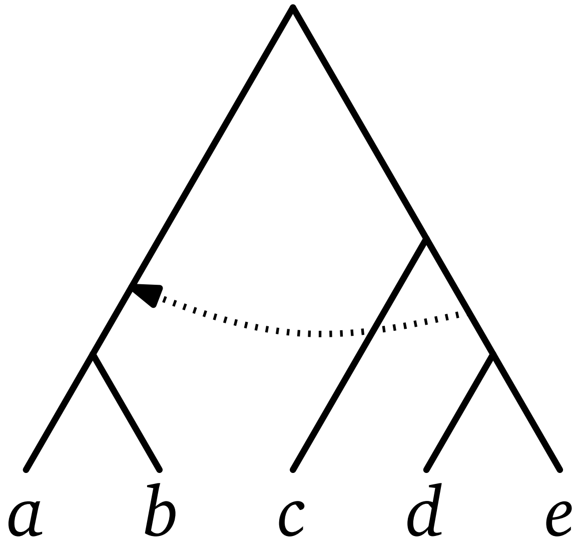


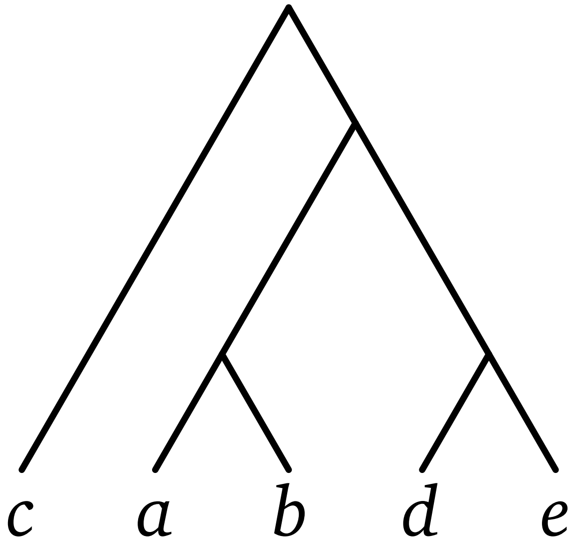
Hybridization



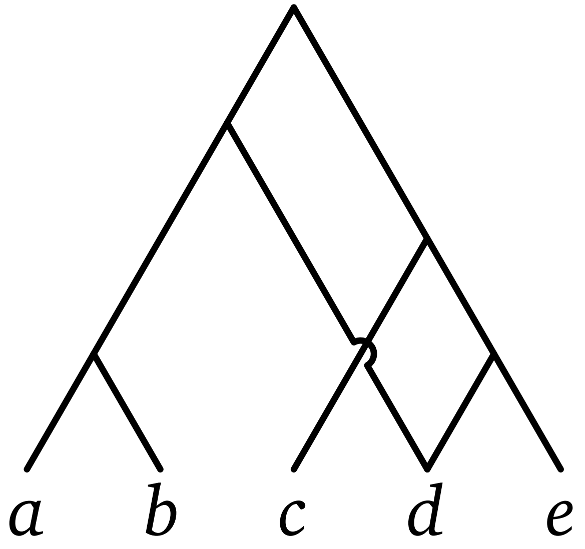
Reticulation Events

Lateral gene transfer (subtree prune-and-regraft)





Hybridization



Zeina the Zonkey

Owklawn Farm Zoo, Nova Scotia

Reconciling Phylogenies

1. Tree distances

- ***SPR distance:*** number of SPR operations to transform one tree into the other

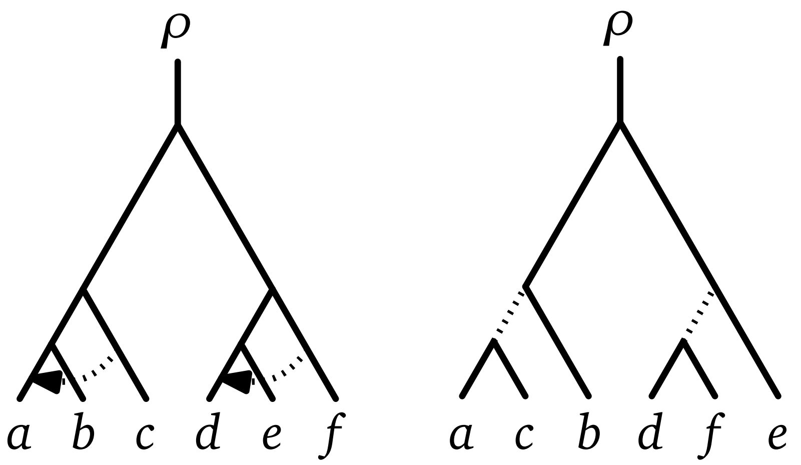
  NP-hard [Bordewich/Semple 2005]

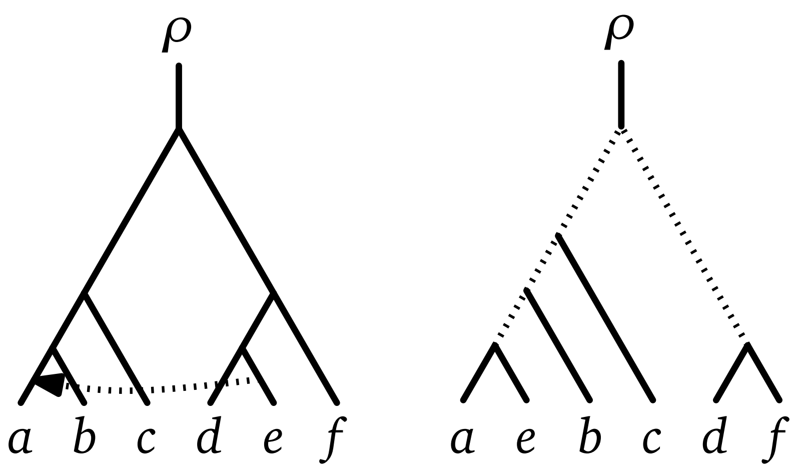
- ***Hybridization number:*** minimum number of nodes with two parents in any network that displays both trees

  NP-hard [Bordewich/Semple 2007]

- ***Robinson-Foulds distance:*** number of bipartitions that disagree

  Linear-time, but . . .

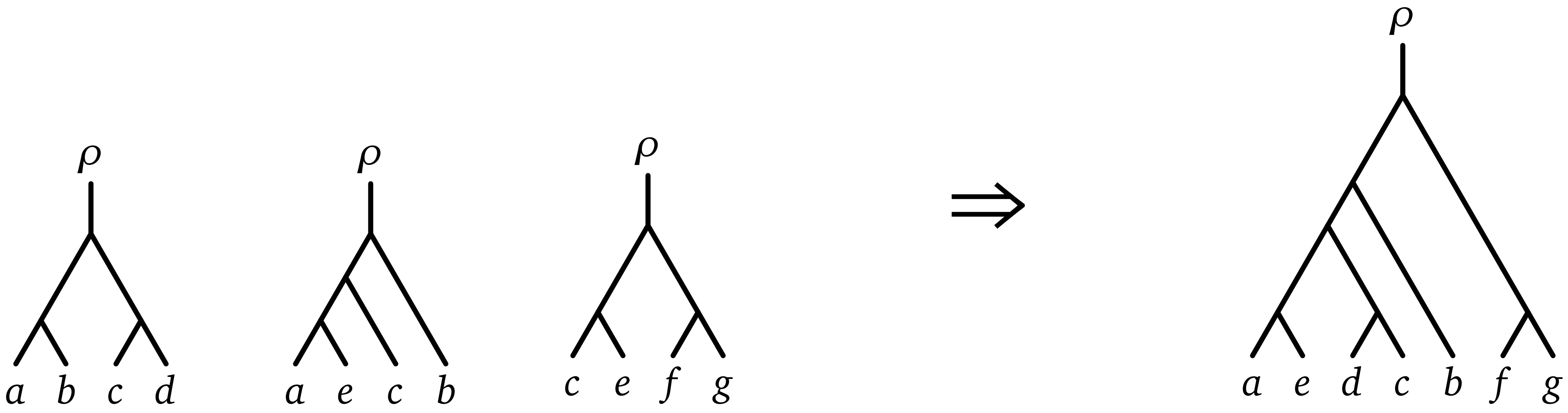




Reconciling Phylogenies

2. Supertrees

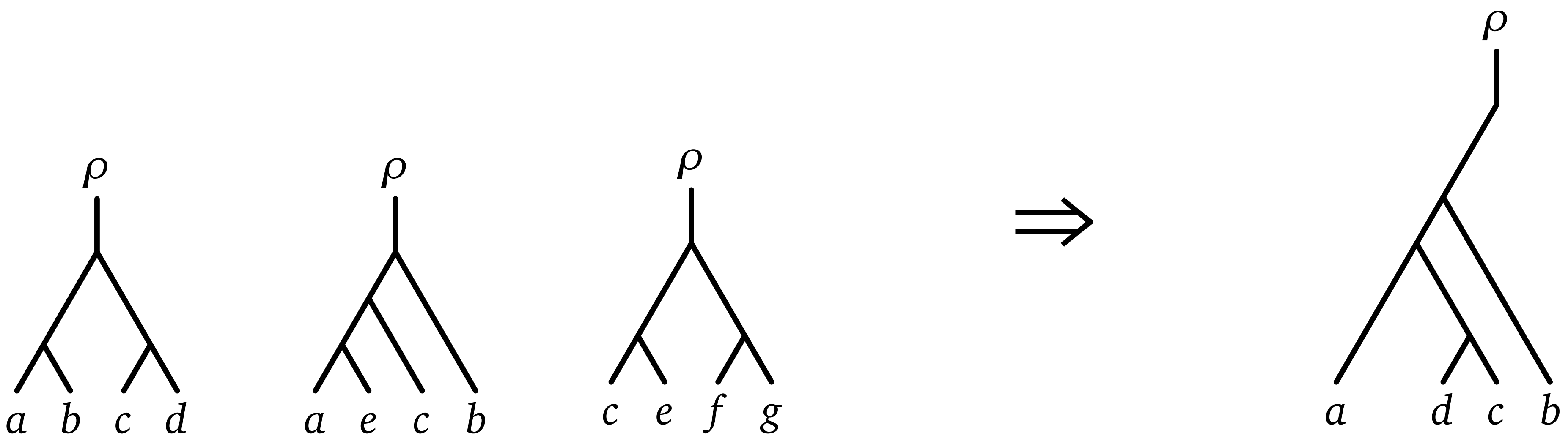
- MRP supertrees [Ragan 1992]
- RF supertrees [Bansal et al. 2010]
- SPR supertrees [Whidden/Zeh/Beiko 2012]
- . . .



Reconciling Phylogenies

2. Supertrees

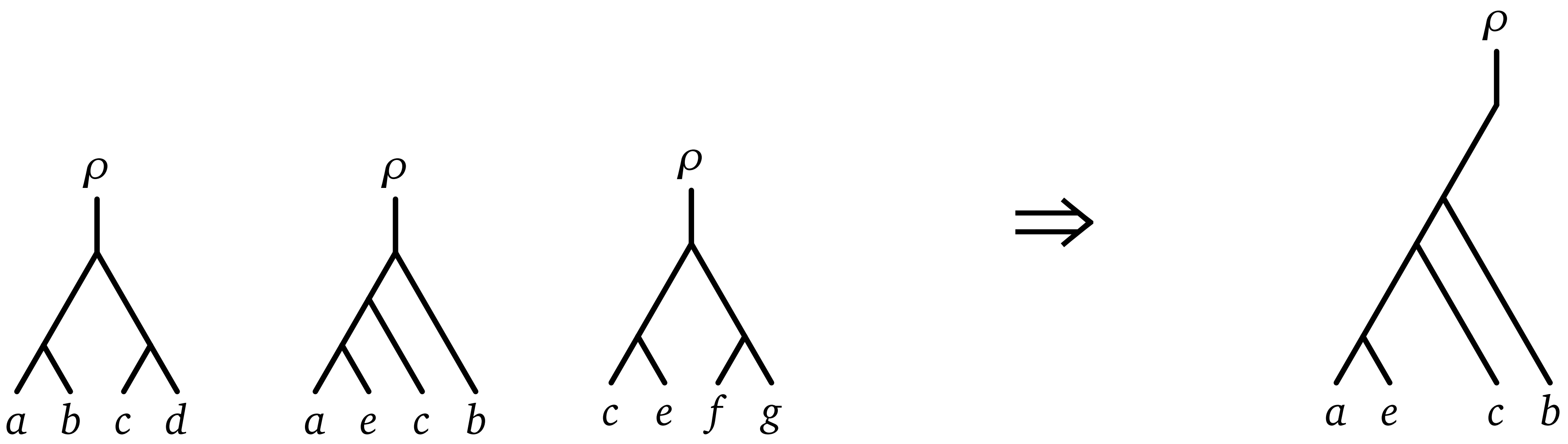
- MRP supertrees [Ragan 1992]
- RF supertrees [Bansal et al. 2010]
- SPR supertrees [Whidden/Zeh/Beiko 2012]
- . . .



Reconciling Phylogenies

2. Supertrees

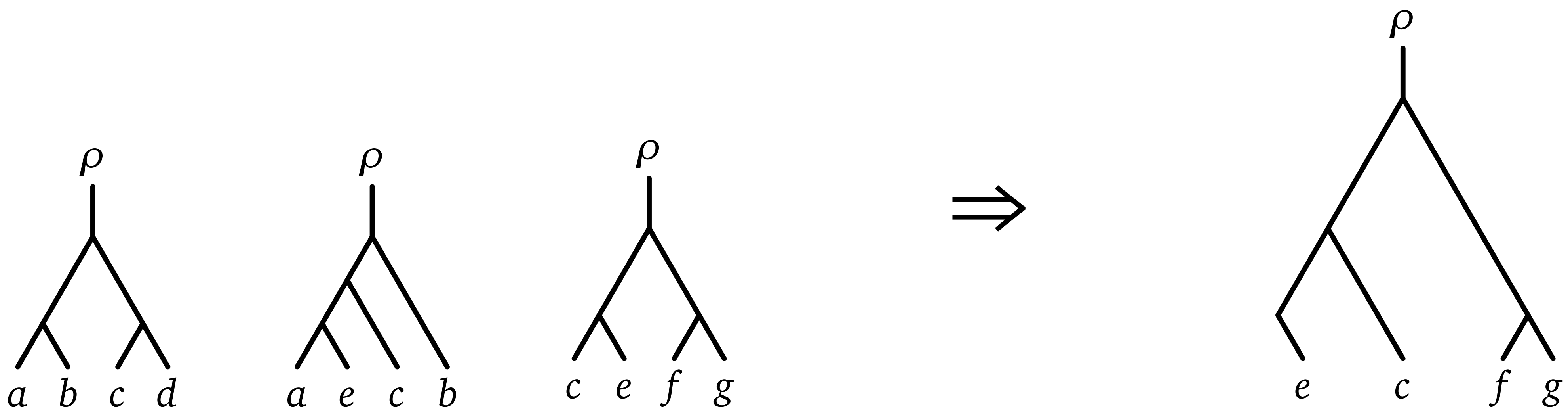
- MRP supertrees [Ragan 1992]
- RF supertrees [Bansal et al. 2010]
- SPR supertrees [Whidden/Zeh/Beiko 2012]
- . . .



Reconciling Phylogenies

2. Supertrees

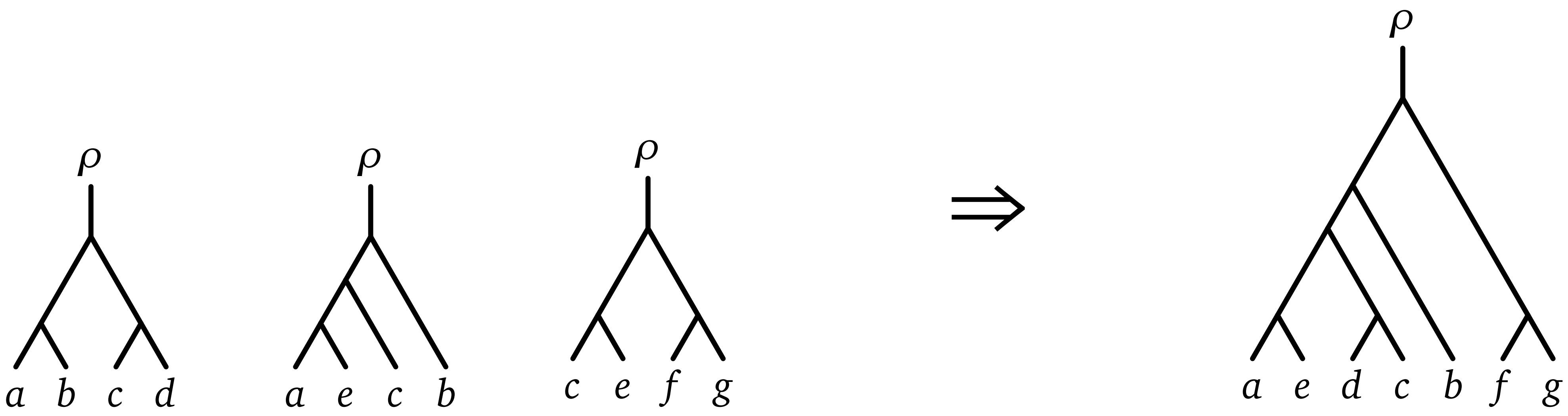
- MRP supertrees [Ragan 1992]
- RF supertrees [Bansal et al. 2010]
- SPR supertrees [Whidden/Zeh/Beiko 2012]
- . . .



Reconciling Phylogenies

2. Supertrees

- MRP supertrees [Ragan 1992]
- RF supertrees [Bansal et al. 2010]
- SPR supertrees [Whidden/Zeh/Beiko 2012]
- . . .



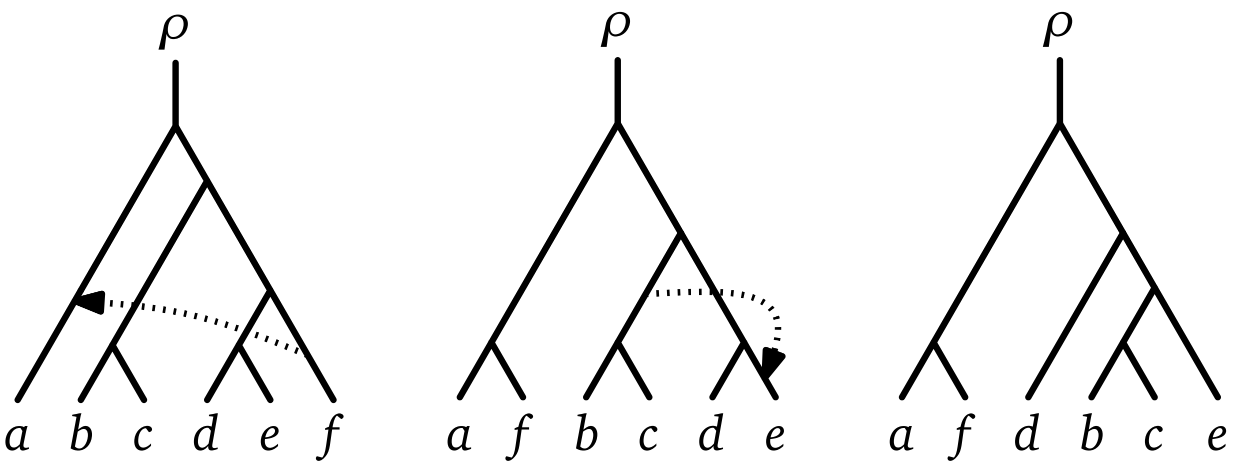
3. Phylogenetic networks

- DLT networks [Hallet/Lagergren 2011, Doyon et al. 2011]
- Recombination networks [Gusfield et al. 2003]
- Level-*k* hybridization networks [van Iersel/Kelk 2011]
- MAAF of multiple trees [Chen/Wang 2012]

Computing SPR Distance

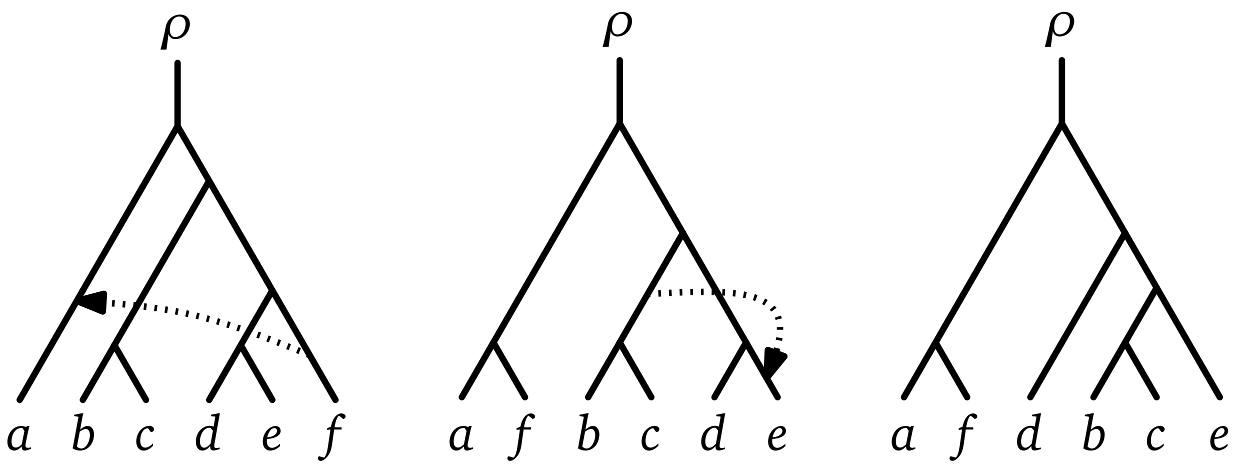
Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?

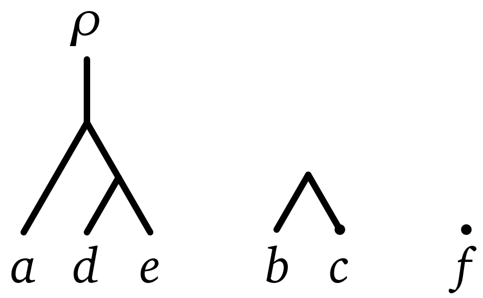


Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?



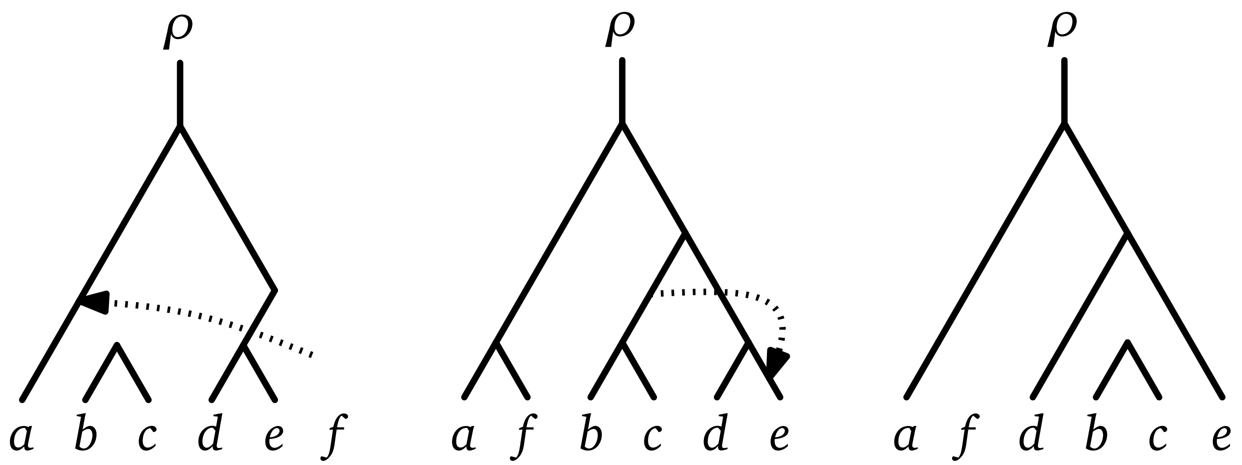
What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?



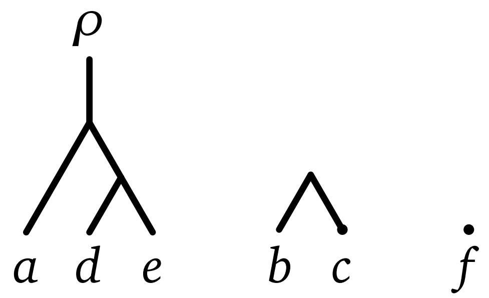
[Bordewich/Semple 2005]

Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?



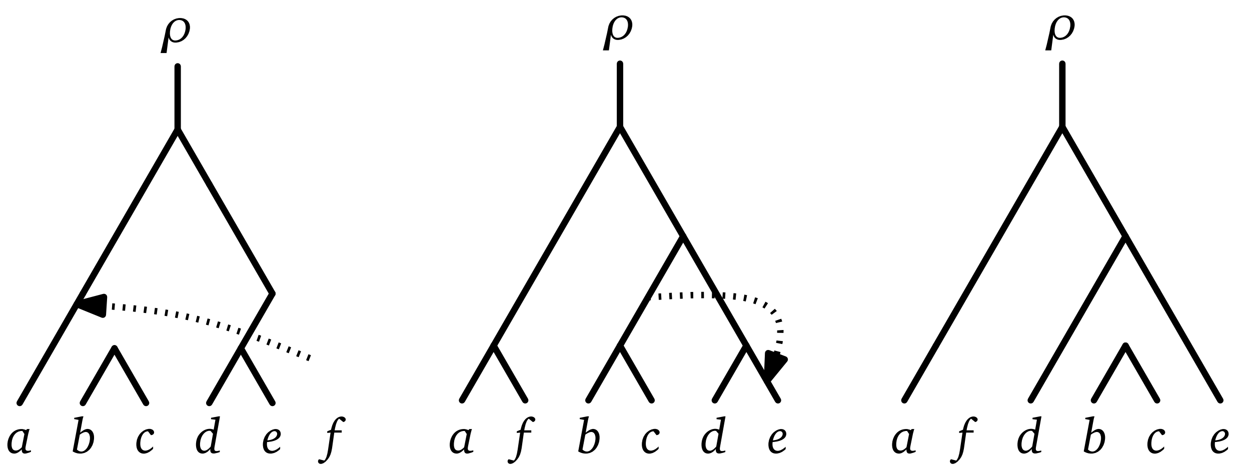
What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?



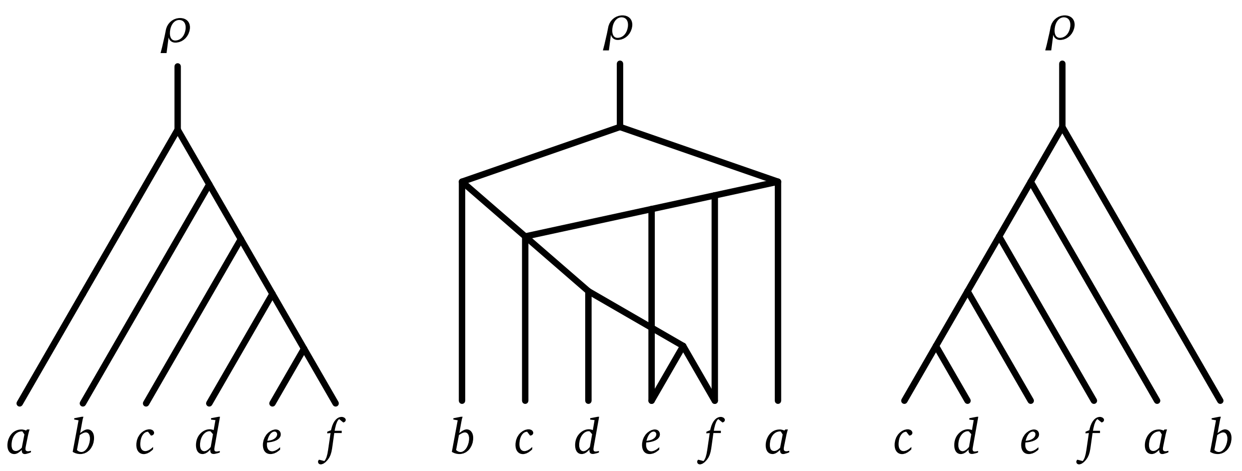
[Bordewich/Semple 2005]

Agreement Forests

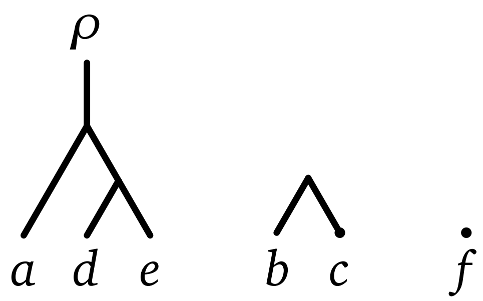
How many SPR operations does it take to turn T_1 into T_2?



What is the smallest hybridization network that displays both T_1 and T_2?



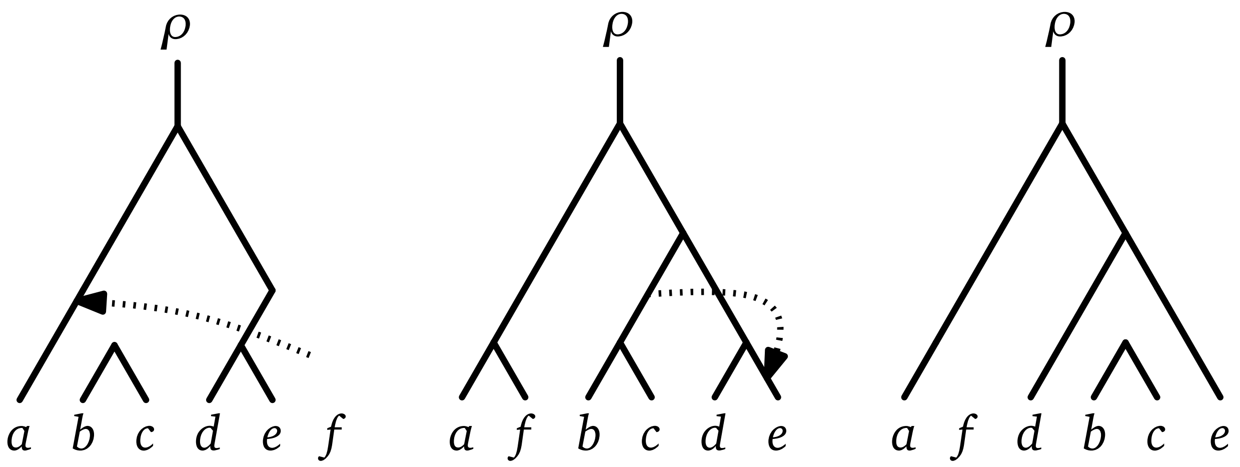
What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?



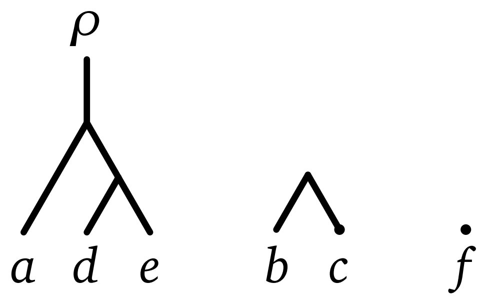
[Bordewich/Semple 2005]

Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?

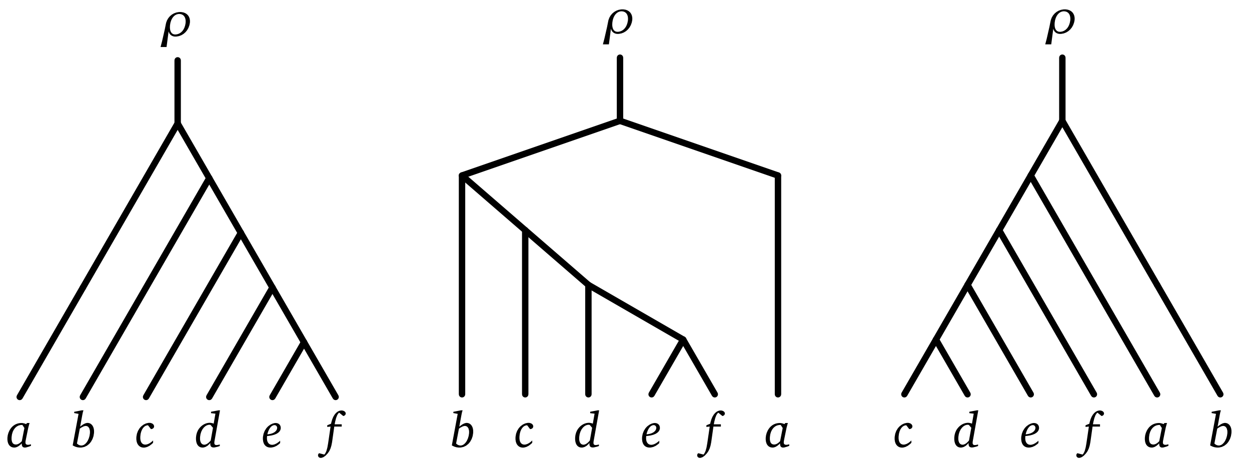


What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?



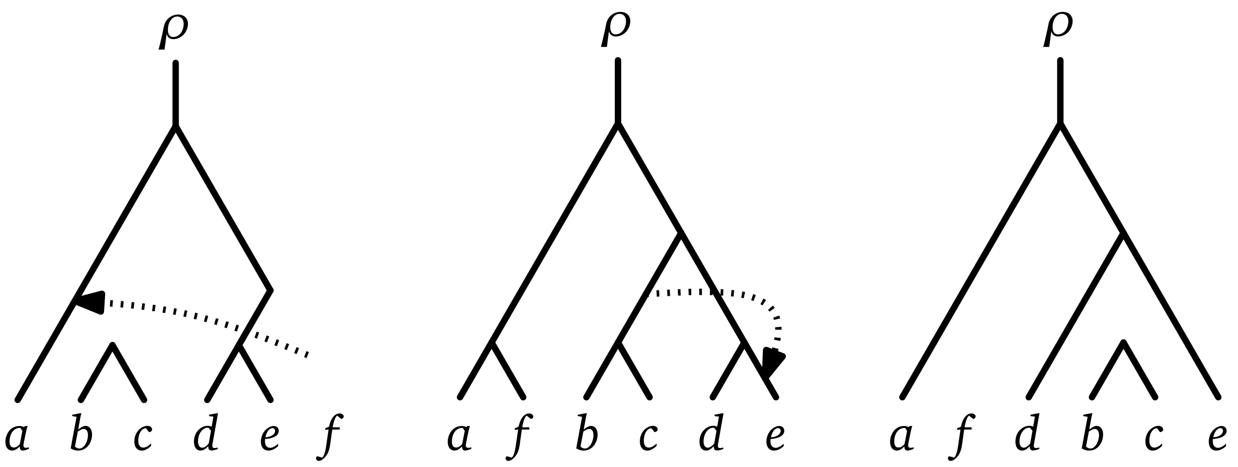
[Bordewich/Semple 2005]

What is the smallest hybridization network that displays both T_1 and T_2?

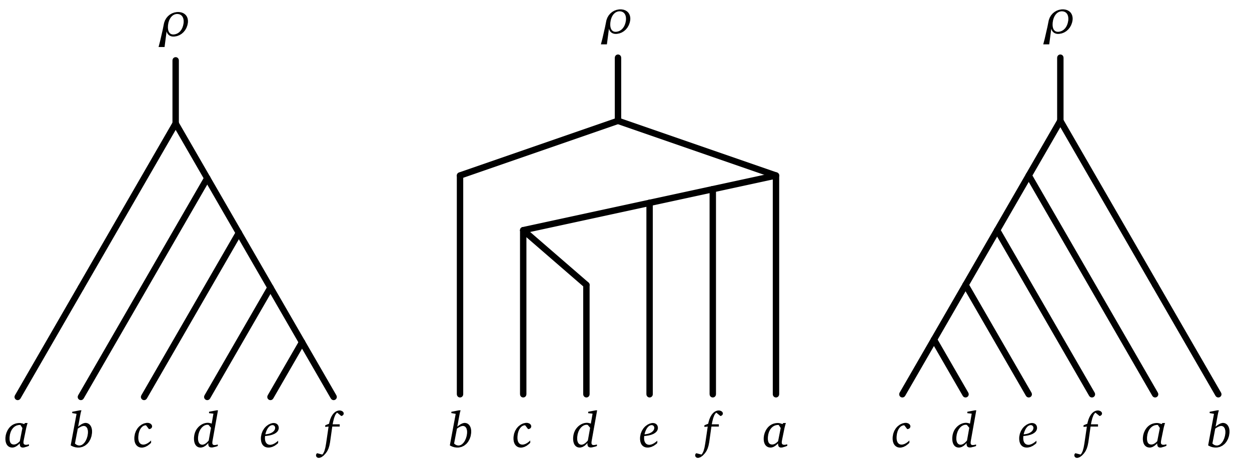


Agreement Forests

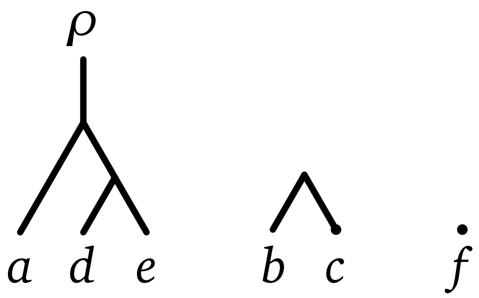
How many SPR operations does it take to turn T_1 into T_2?



What is the smallest hybridization network that displays both T_1 and T_2?



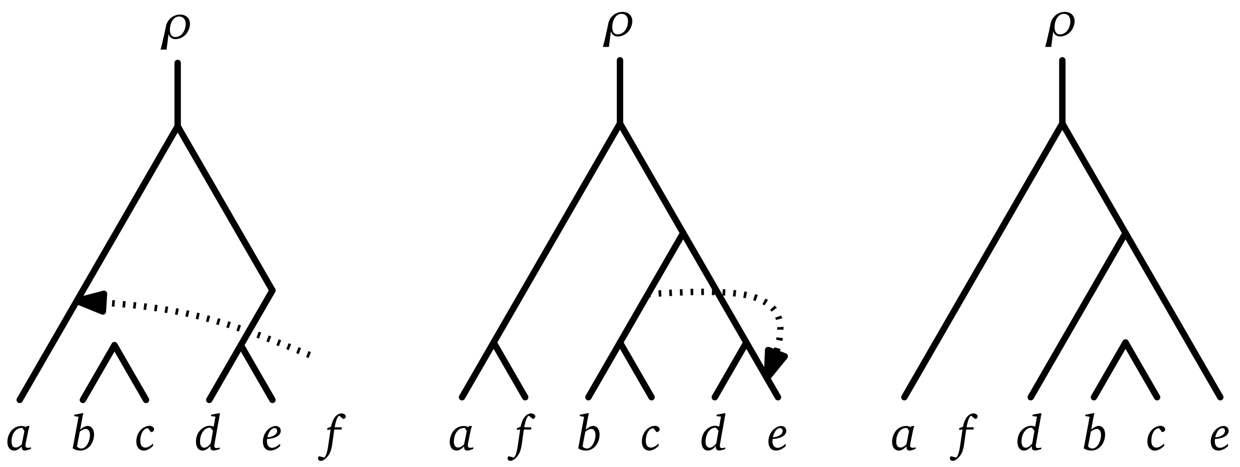
What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?



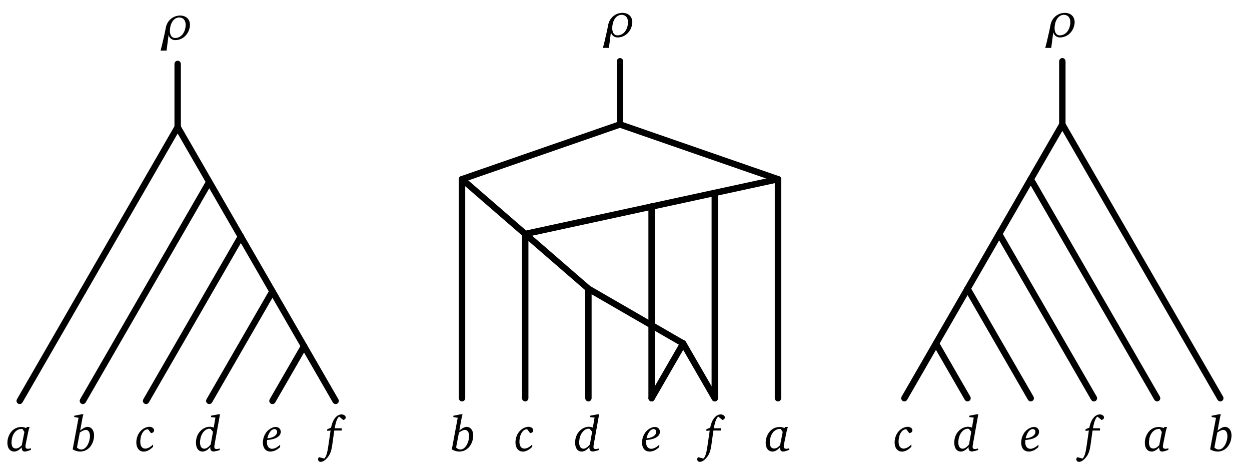
[Bordewich/Semple 2005]

Agreement Forests

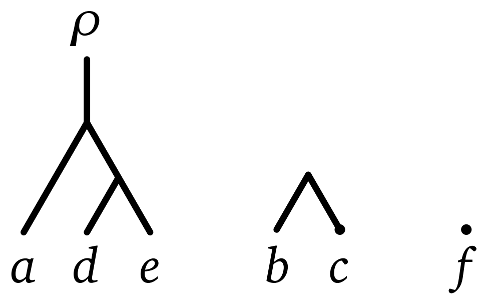
How many SPR operations does it take to turn T_1 into T_2?



What is the smallest hybridization network that displays both T_1 and T_2?



What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?

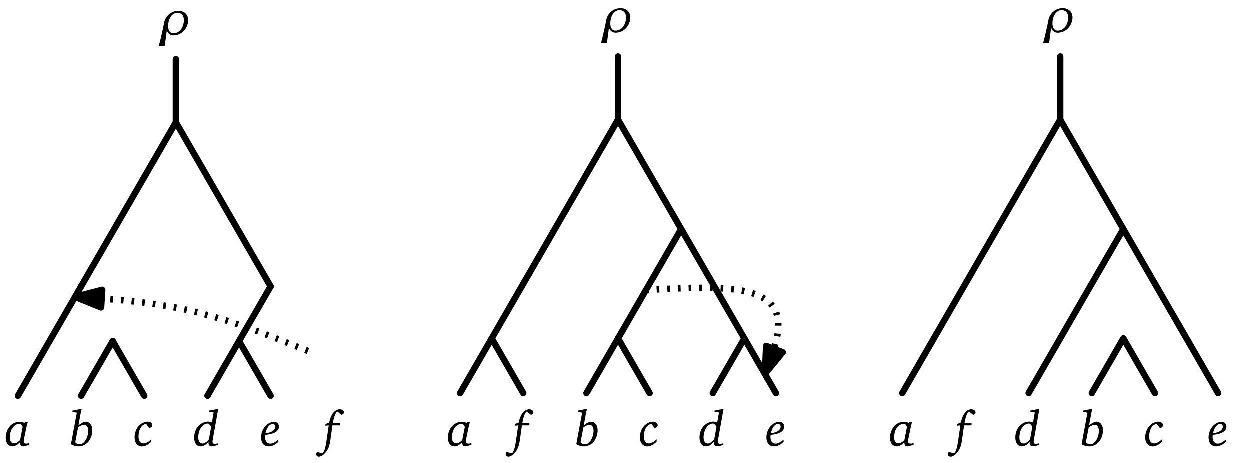


[Bordewich/Semple 2005]

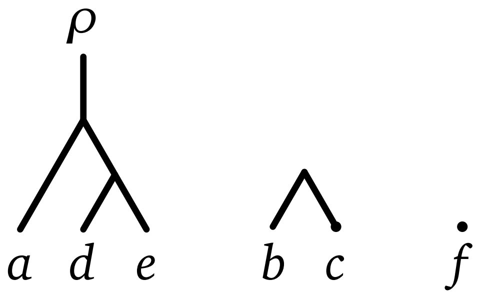
What is the largest ***acyclic*** agreement forest of T_1 and T_2?

Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?

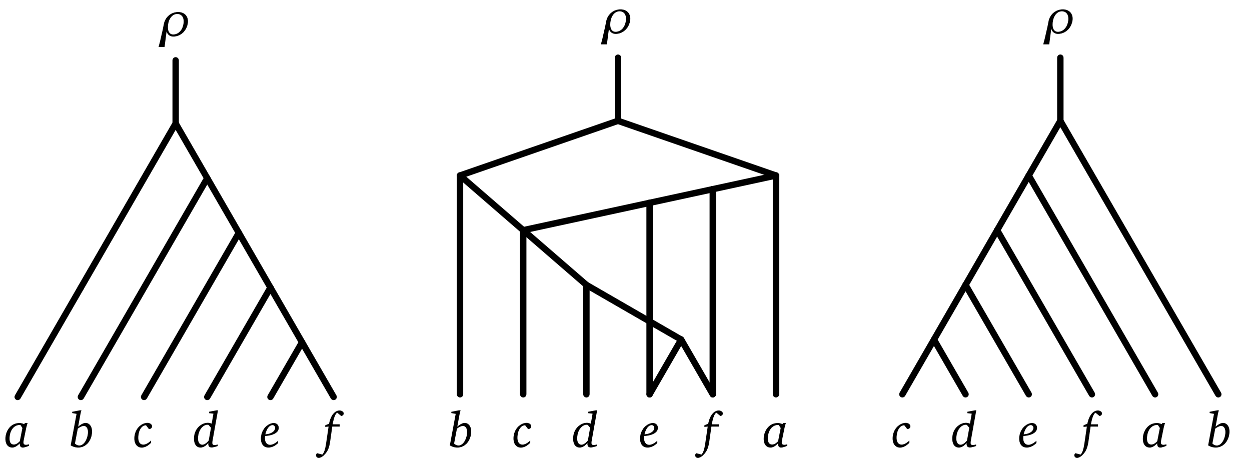


What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?

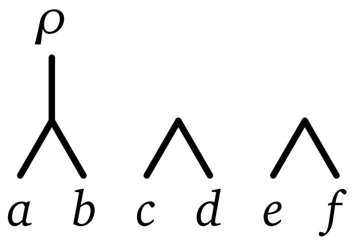


[Bordewich/Semple 2005]

What is the smallest hybridization network that displays both T_1 and T_2?



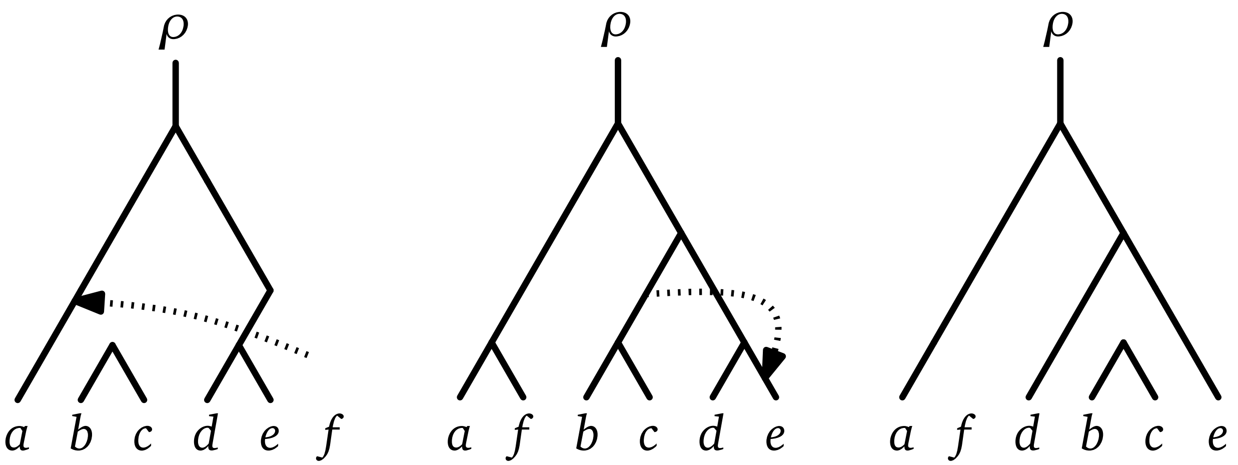
What is the largest ***acyclic*** agreement forest of T_1 and T_2?



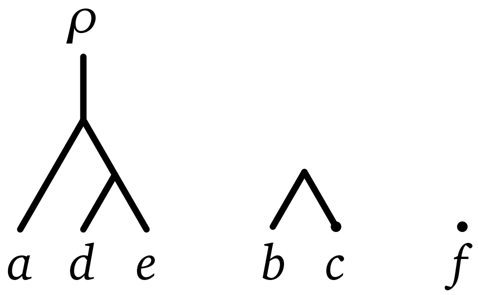
agreement forest

Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?

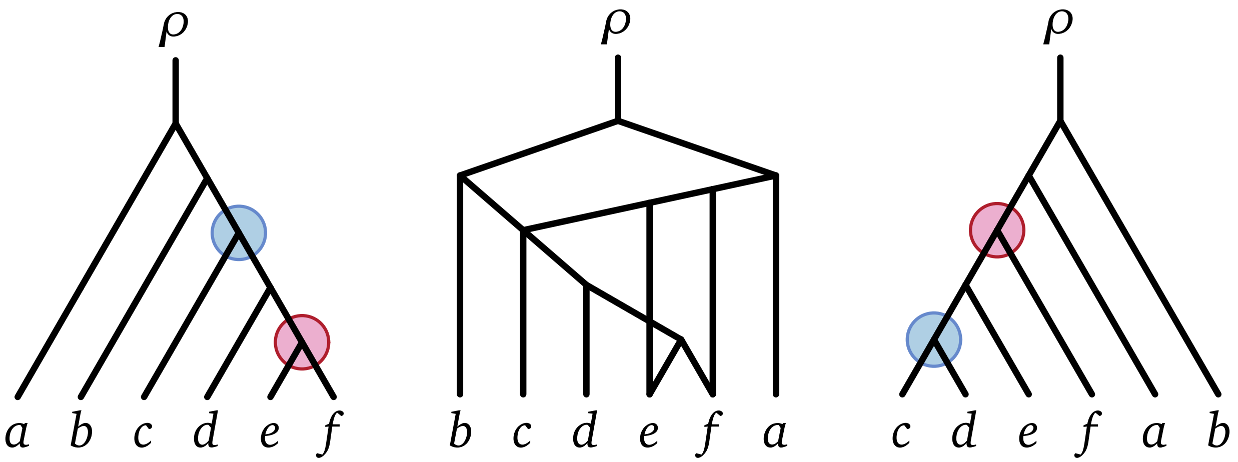


What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?

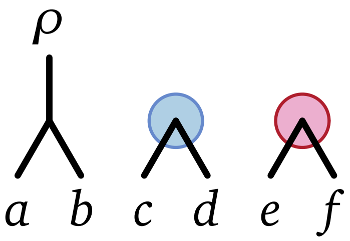


[Bordewich/Semple 2005]

What is the smallest hybridization network that displays both T_1 and T_2?



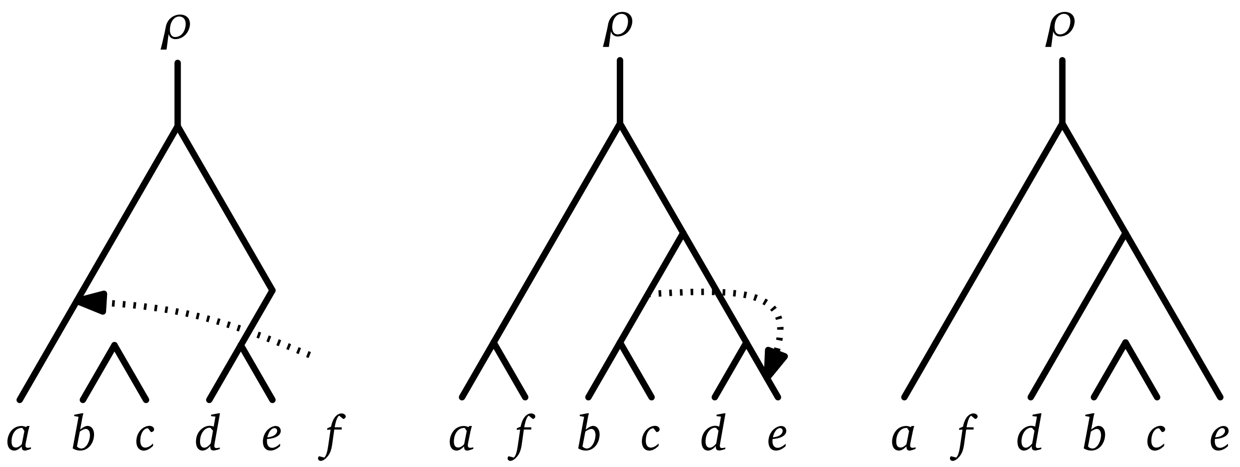
What is the largest ***acyclic*** agreement forest of T_1 and T_2?



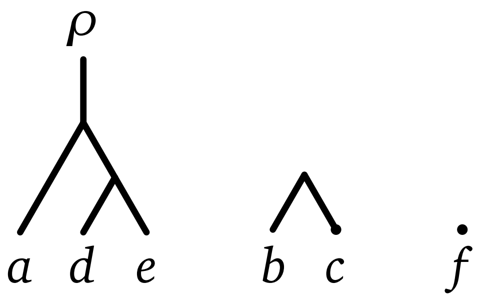
agreement forest

Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?

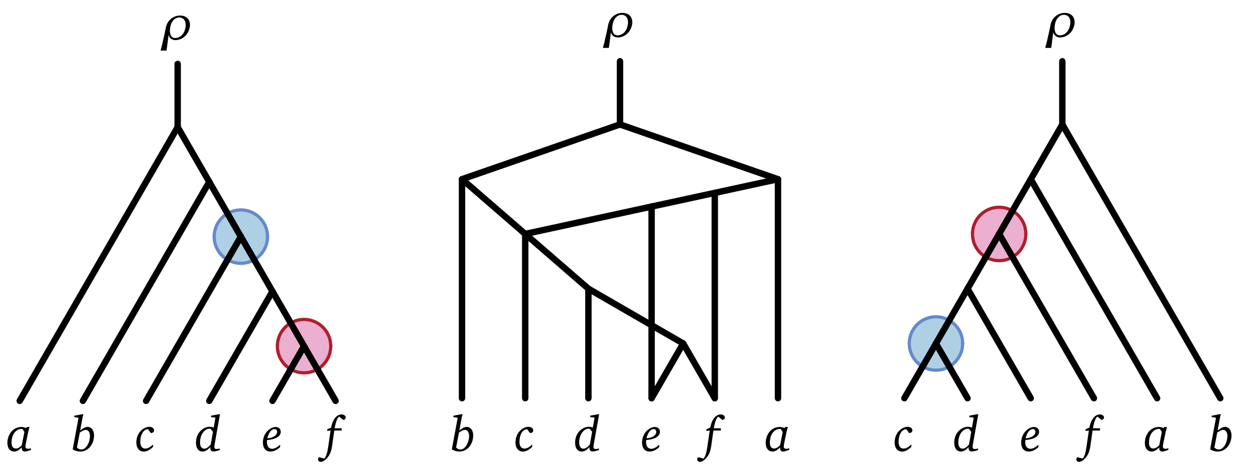


What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?

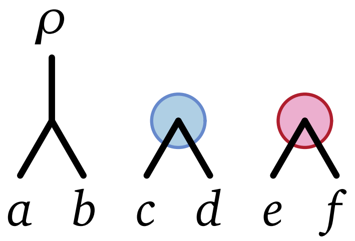


[Bordewich/Semple 2005]

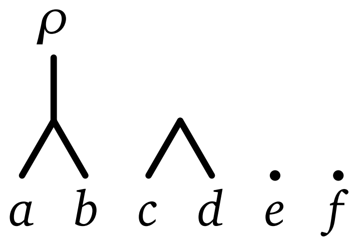
What is the smallest hybridization network that displays both T_1 and T_2?



What is the largest ***acyclic*** agreement forest of T_1 and T_2?



agreement forest

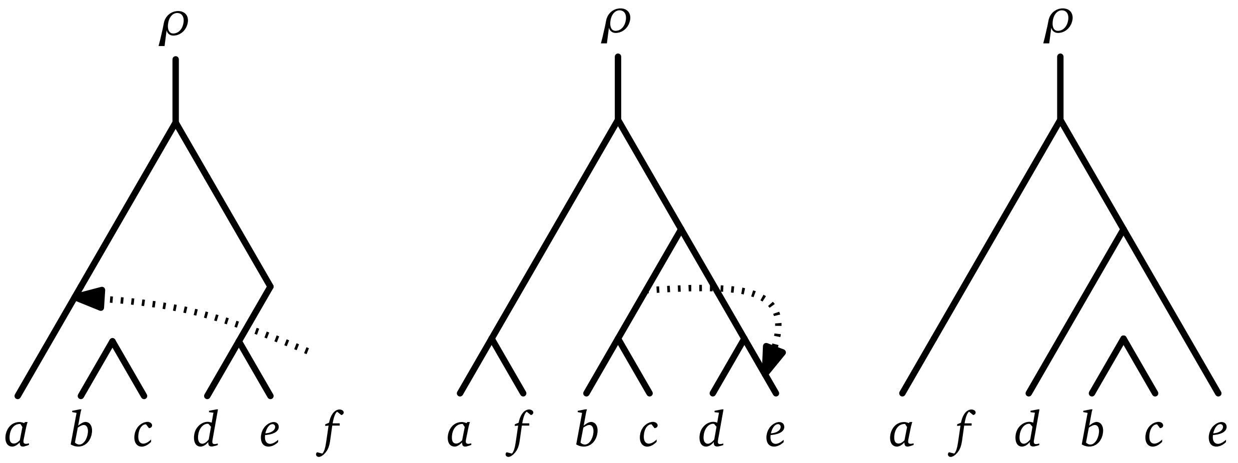


acyclic agreement forest

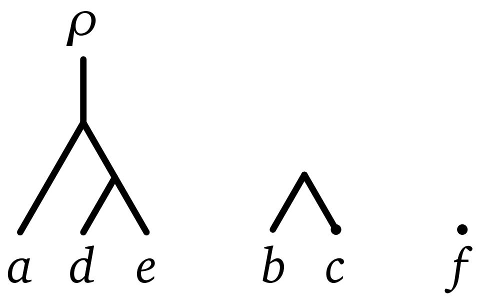
[Bordewich/Semple 2007]

Agreement Forests

How many SPR operations does it take to turn T_1 into T_2?

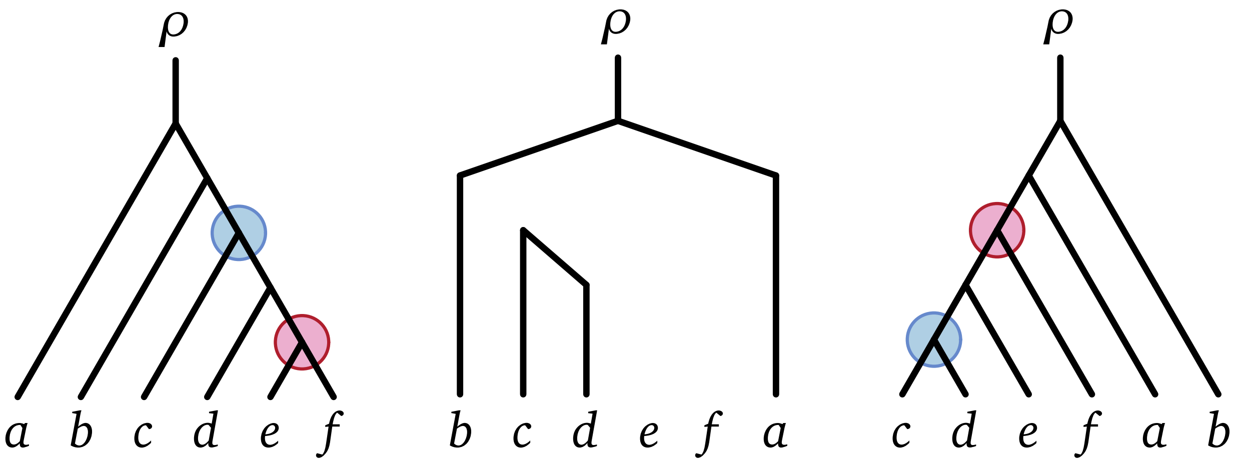


What is the largest forest we can obtain from T_1 and T_2 using edge deletions and forced contractions?

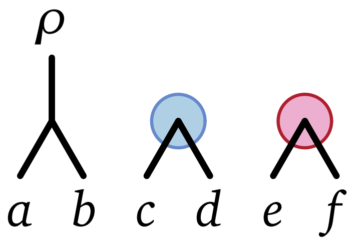


[Bordewich/Semple 2005]

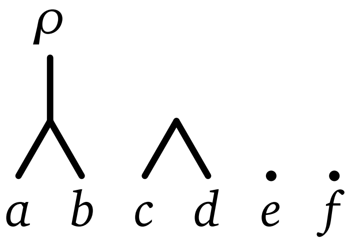
What is the smallest hybridization network that displays both T_1 and T_2?



What is the largest ***acyclic*** agreement forest of T_1 and T_2?



agreement forest

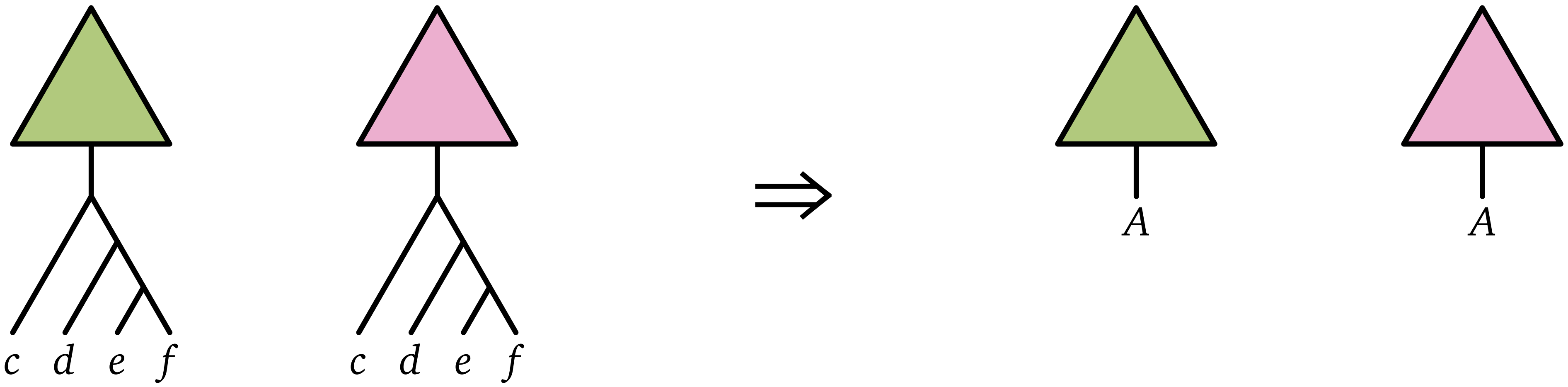


acyclic
agreement forest

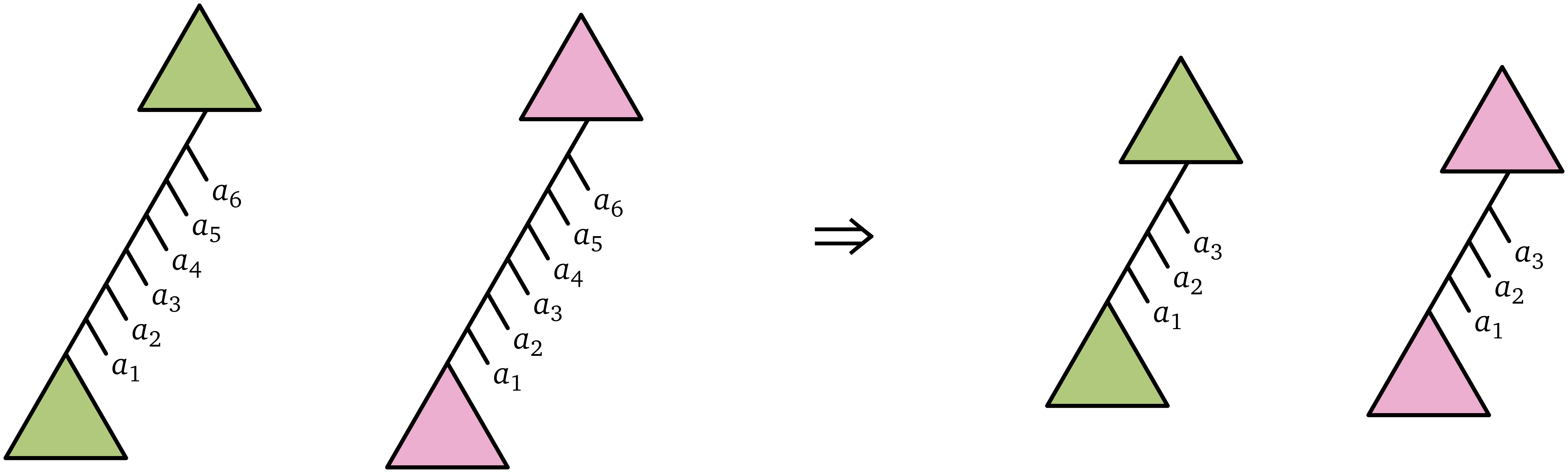
[Bordewich/Semple 2007]

Kernelization for Maximum Agreement Forest (SPR Distance)

Rule 1: Prune agreeing subtrees



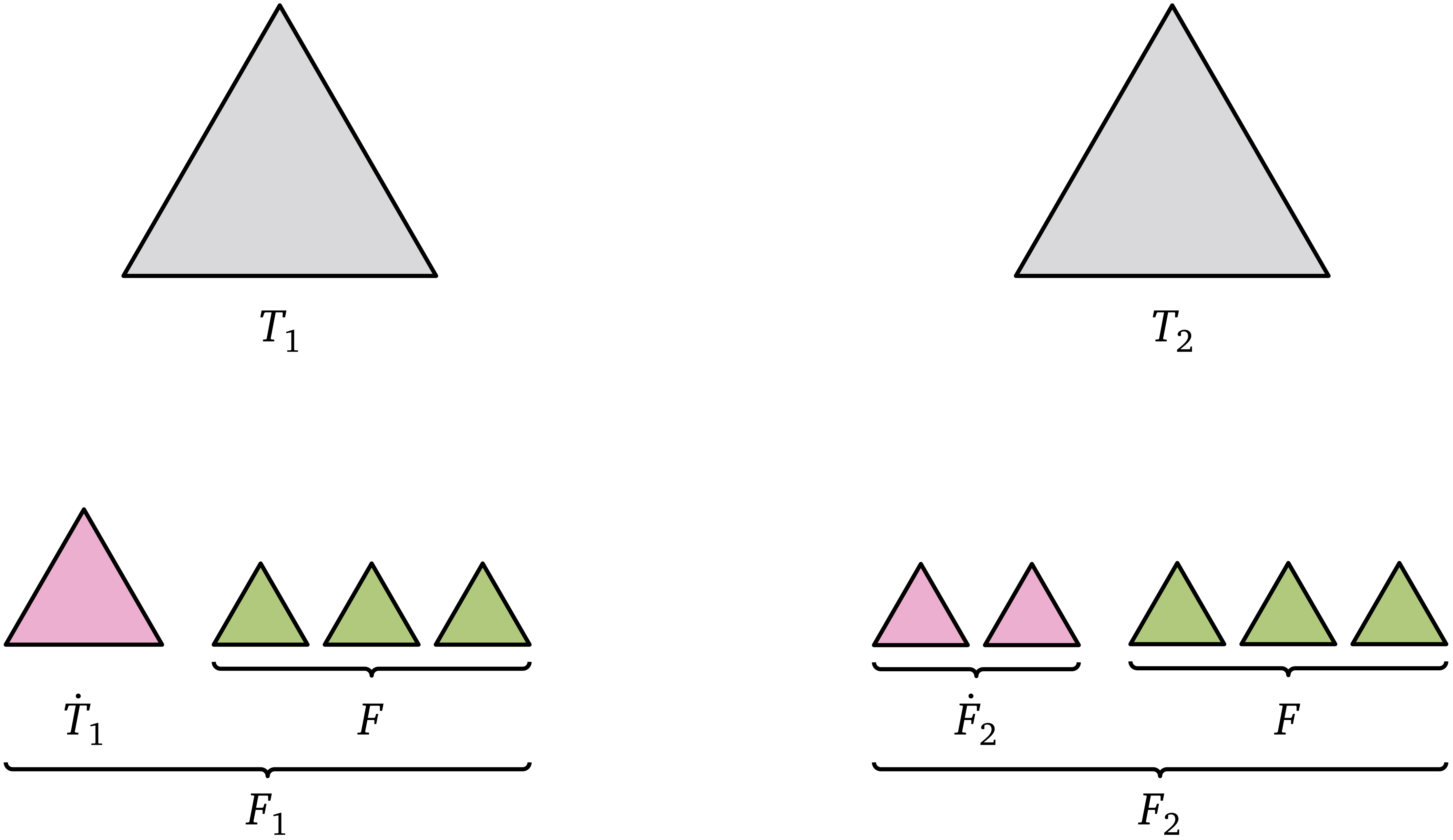
Rule 2: Compress agreeing chains



Running time: $O((56k)^k + \text{poly}(n))$

[Bordewich/Semple 2005]

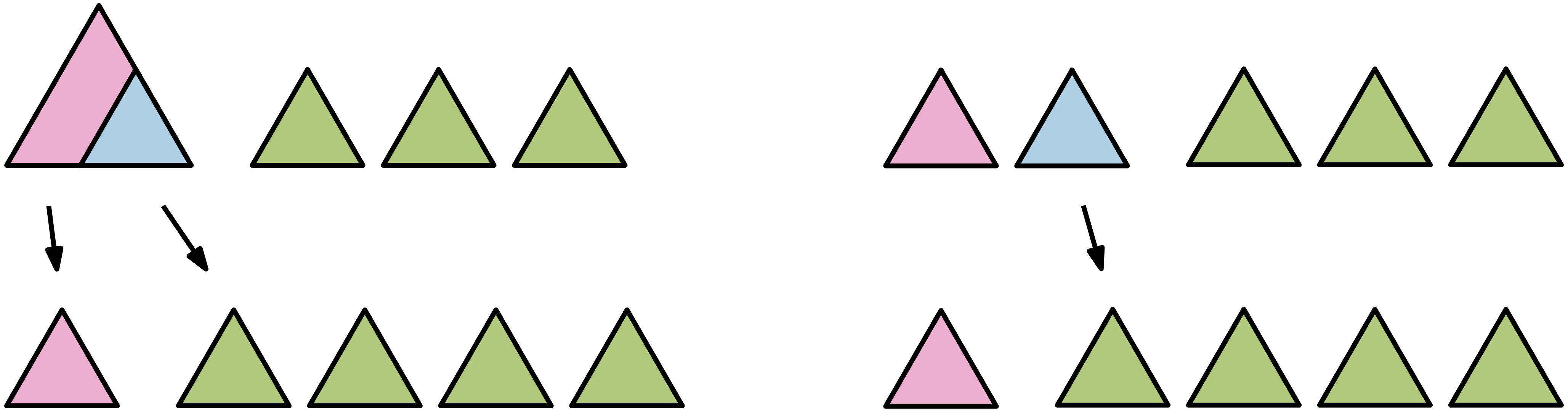
Depth-Bounded Search for MAF [Whidden/Zeh 2009]



An MAF of T_1 and T_2 can be obtained by cutting edges in F_2.

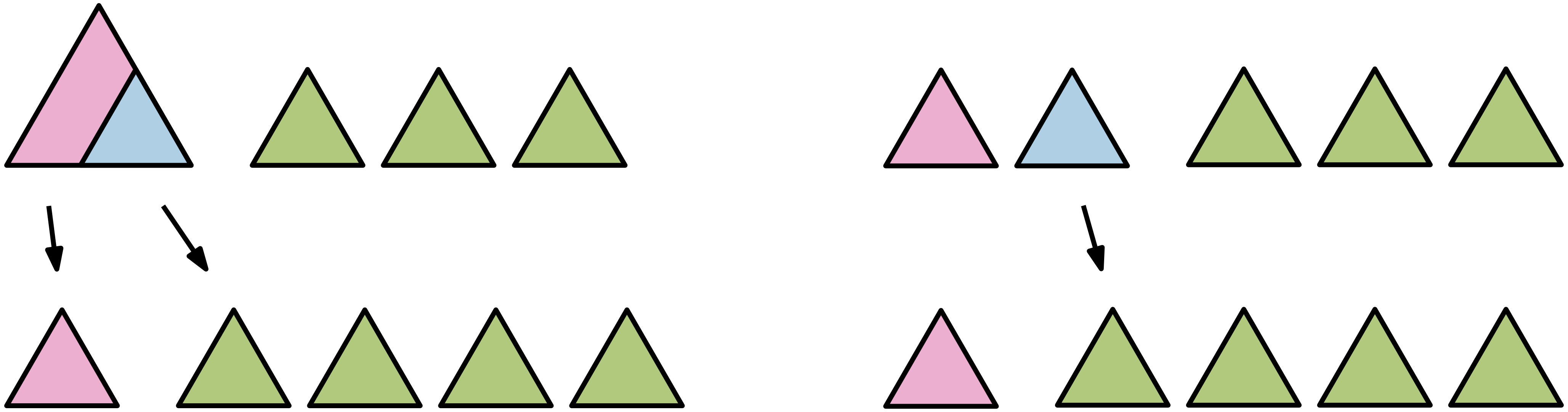
Depth-Bounded Search for MAF [Whidden/Zeh 2009]

Case 1: A whole tree in $\dot{F}_2$ agrees with a subtree of $\dot{T}_1$

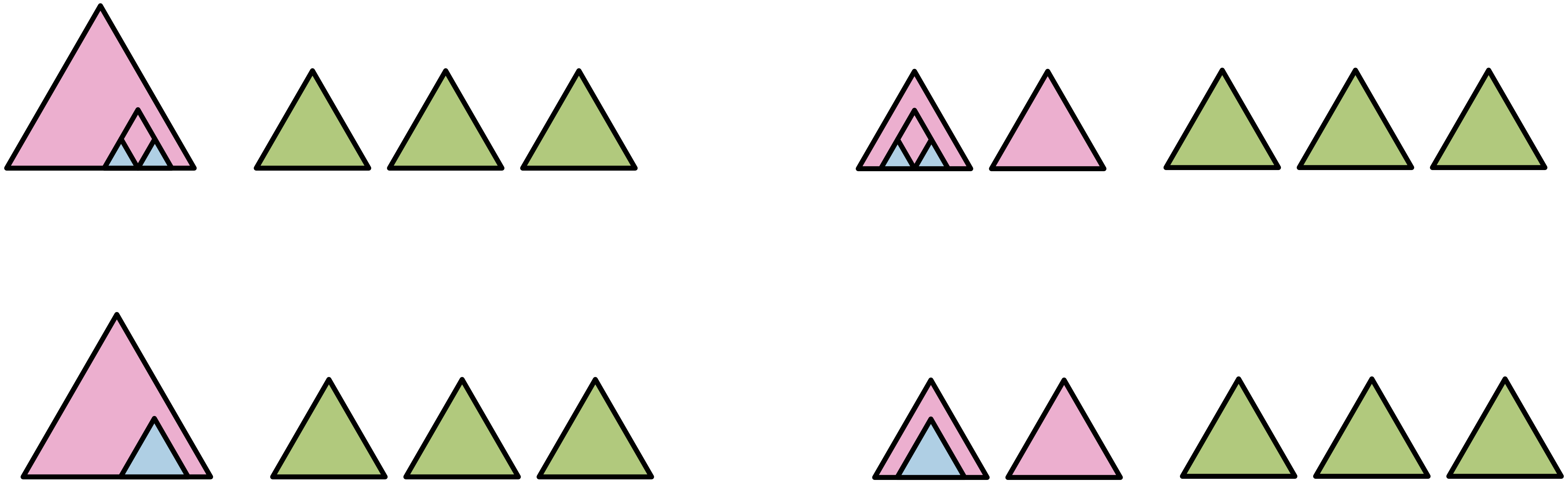


Depth-Bounded Search for MAF [Whidden/Zeh 2009]

Case 1: A whole tree in $\dot{F}_2$ agrees with a subtree of $\dot{T}_1$

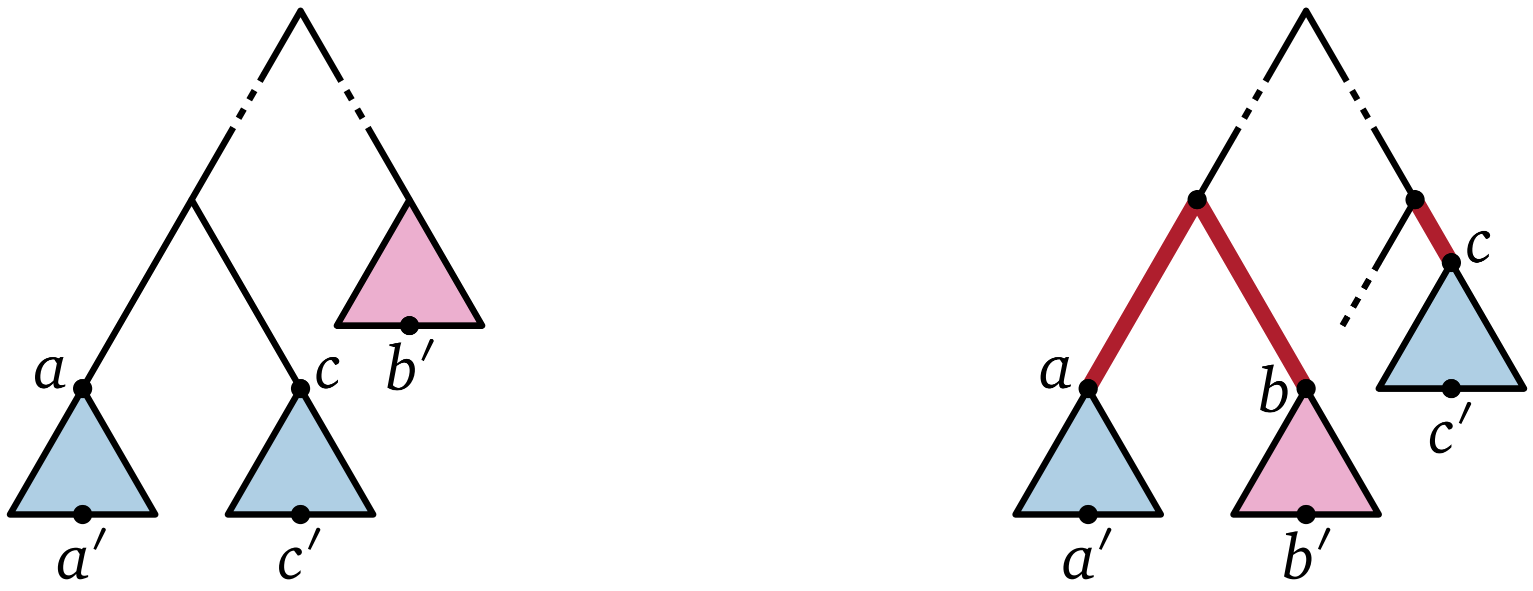


Case 2: Two agreeing subtrees are adjacent in $\dot{T}_1$ and $\dot{F}_2$



Depth-Bounded Search for MAF [Whidden/Zeh 2009]

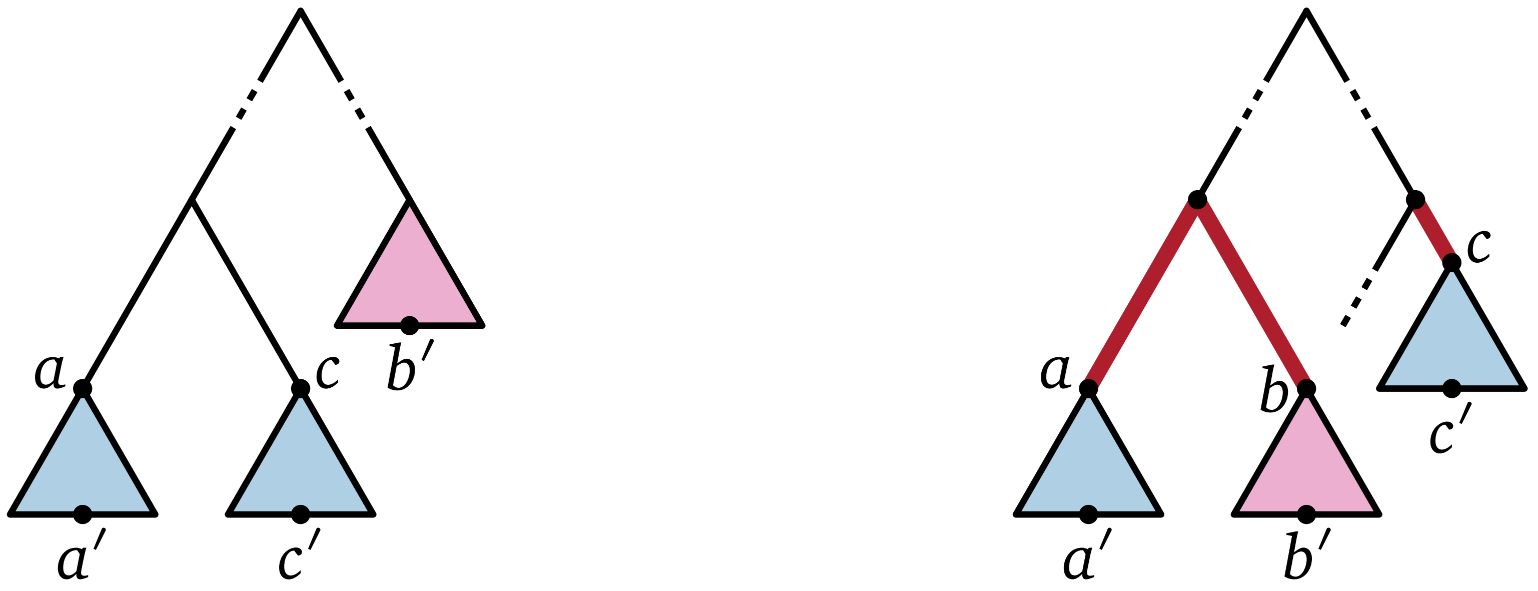
Case 3: Subtrees A and B are adjacent in $\dot{T}_1$ but not in $\dot{F}_2$



One branch per edge

Depth-Bounded Search for MAF [Whidden/Zeh 2009]

Case 3: Subtrees A and B are adjacent in $\dot{T}_1$ but not in $\dot{F}_2$



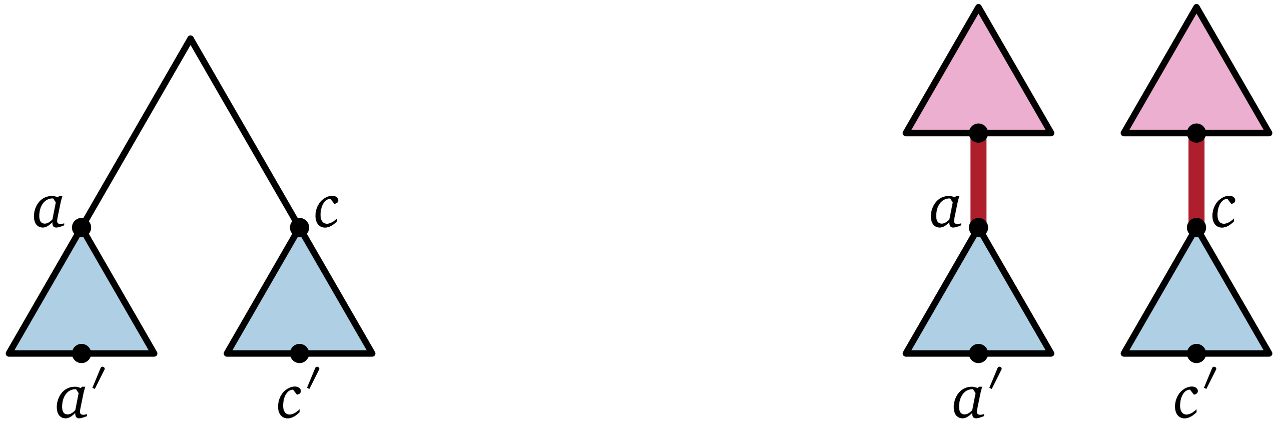
One branch per edge

- Number of recursive calls $= 3^k$
- Each costs $O(n)$ time

Running time: $O(3^k n)$

Improved Branching Rules [Whidden/Beiko/Zeh 2010]

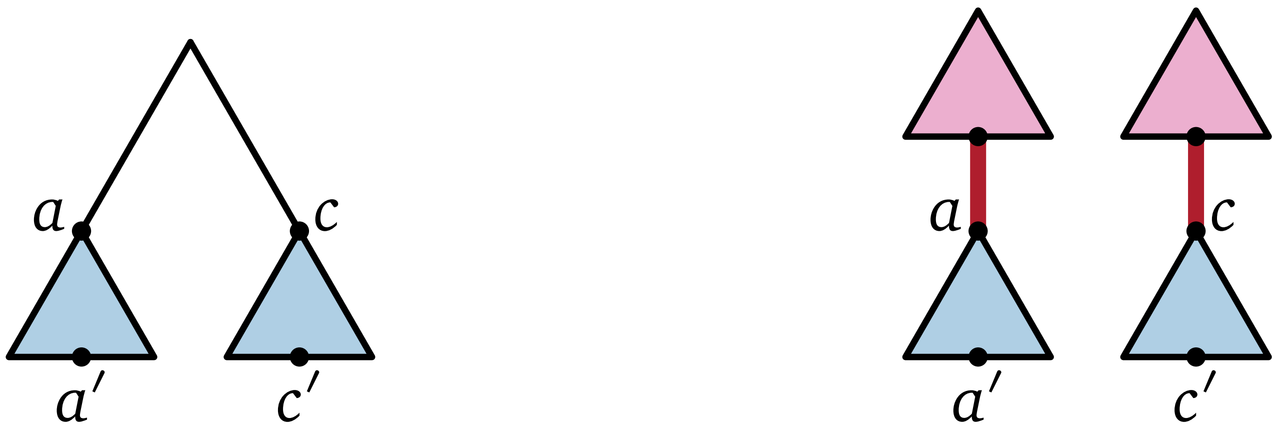
Case 3.1: a and b belong to different subtrees of $\dot{F}_2$



2 recursive calls with parameter $k - 1$

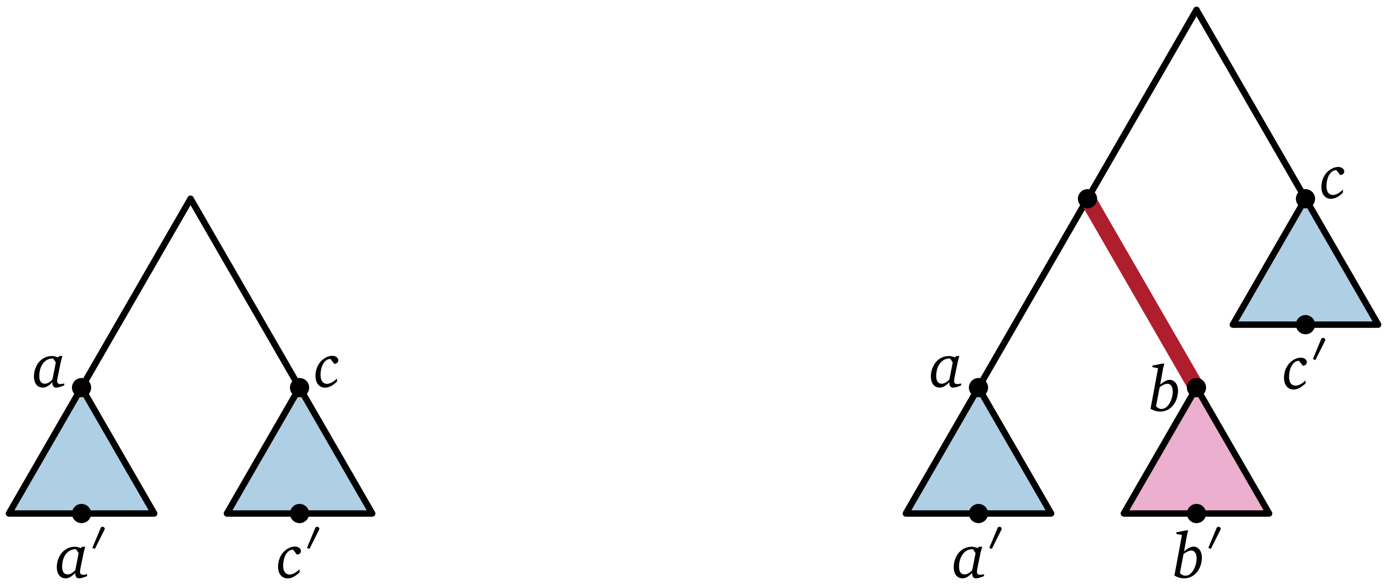
Improved Branching Rules [Whidden/Beiko/Zeh 2010]

Case 3.1: a and b belong to different subtrees of $\dot{F}_2$



2 recursive calls with parameter $k-1$

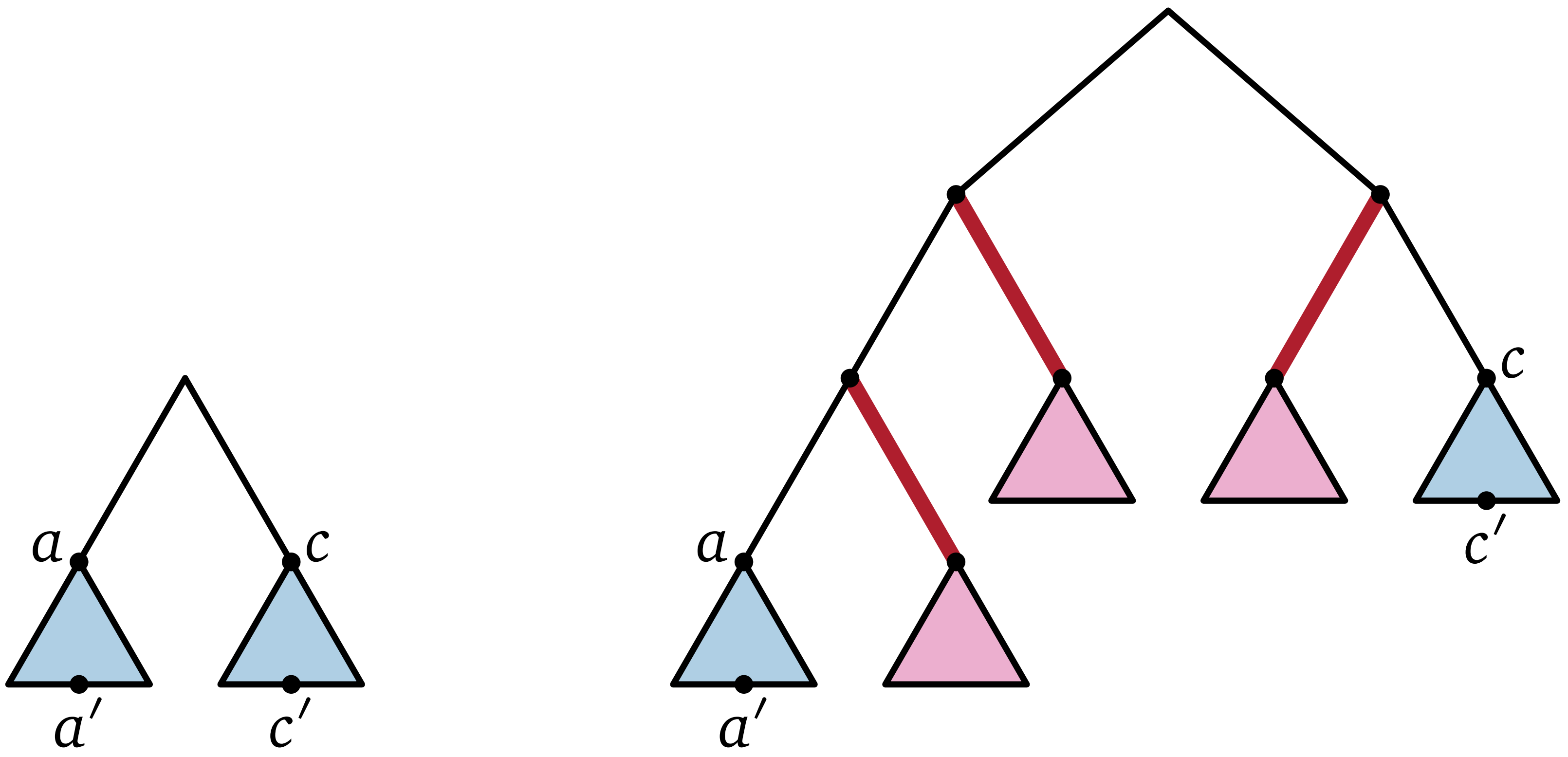
Case 3.2: One pendant subtree on path from a to b in $\dot{F}_2$



1 recursive call with parameter $k-1$

Improved Branching Rules [Whidden/Beiko/Zeh 2010]

Case 3.3: $m \geq 2$ pendant subtrees on path from a to b in $\dot{F}_2$

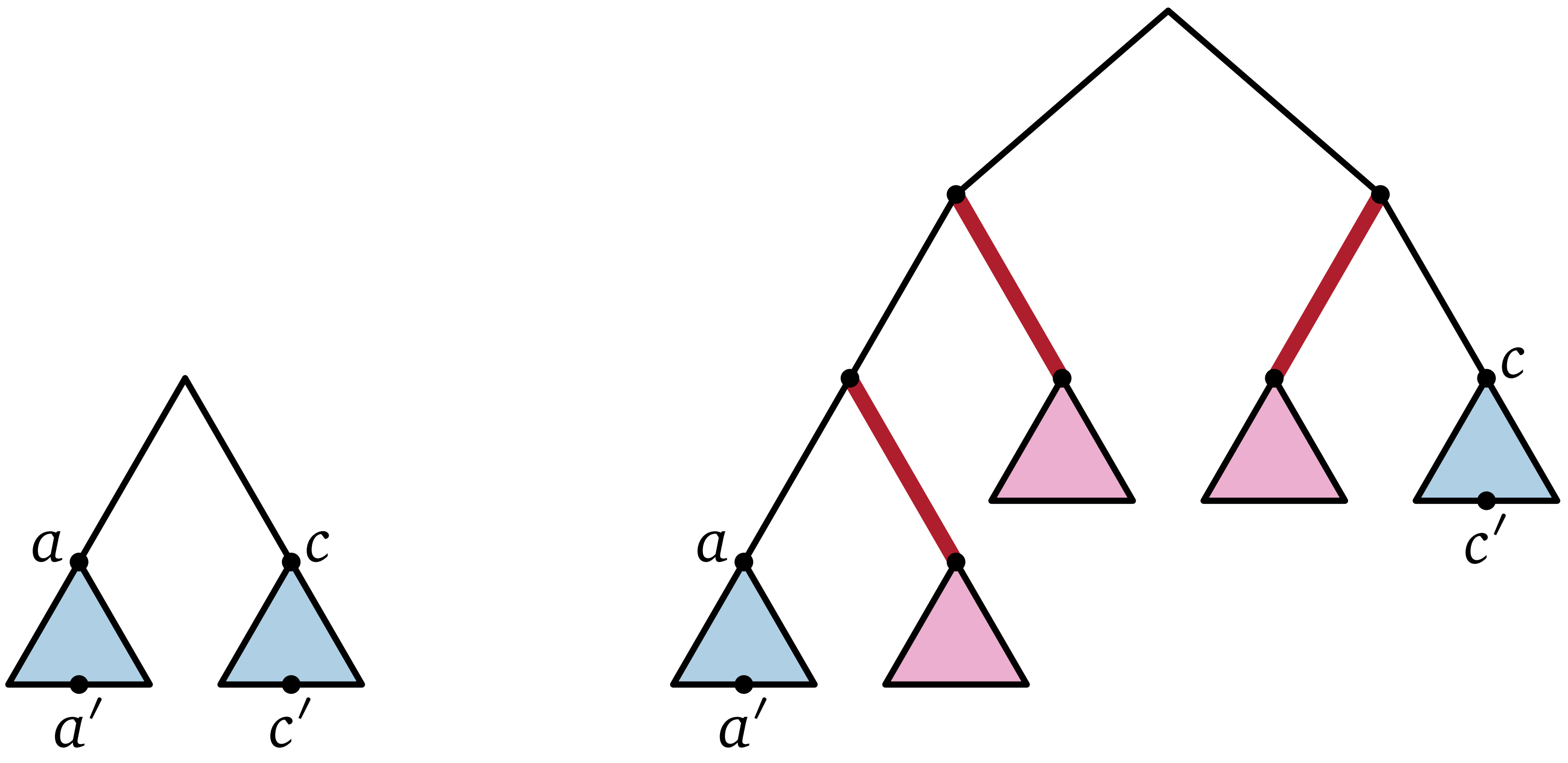


3 recursive calls with parameters

- $k - 1$
- $k - 1$
- $k' \leq k - 2$

Improved Branching Rules [Whidden/Beiko/Zeh 2010]

Case 3.3: $m \geq 2$ pendant subtrees on path from a to b in $\dot{F}_2$



3 recursive calls with parameters

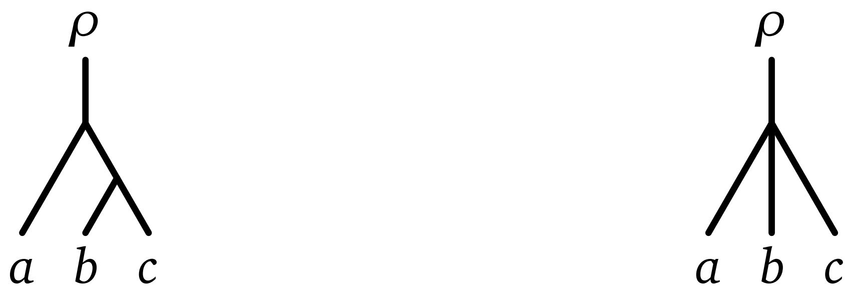
- $k-1$
- $k-1$
- $k' \leq k-2$

Number of recursive invocations

$$I(k) \leq 2I(k-1) + I(k-2) \leq (1+\sqrt{2})^k \approx 2.41^k$$

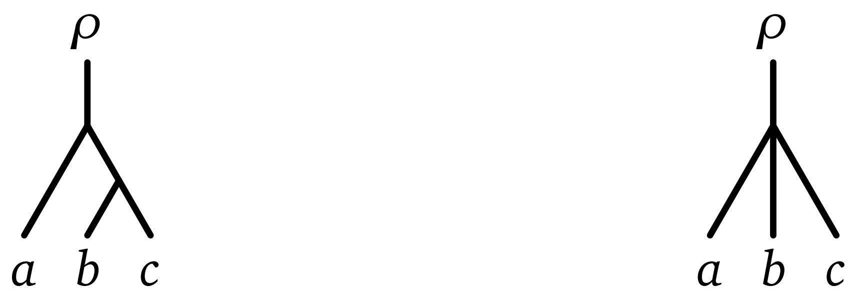
Multifurcating Trees [Whidden/Beiko/Zeh 2012]

Problem 1: What is a meaningful definition of an agreement forest?



Multifurcating Trees [Whidden/Beiko/Zeh 2012]

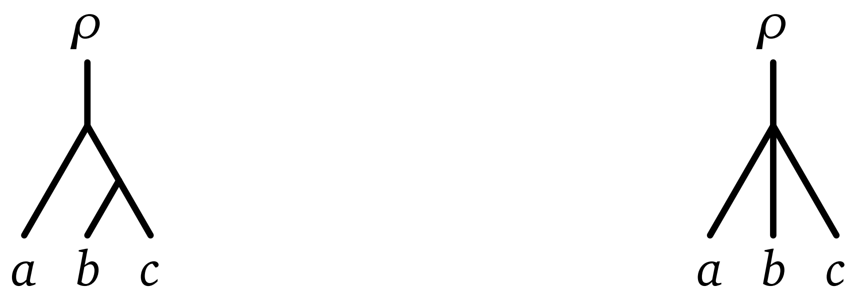
Problem 1: What is a meaningful definition of an agreement forest?



An MAF of two multifurcating phylogenies T_1 and T_2 is the largest forest that is an AF of two binary resolutions of T_1 and T_2.

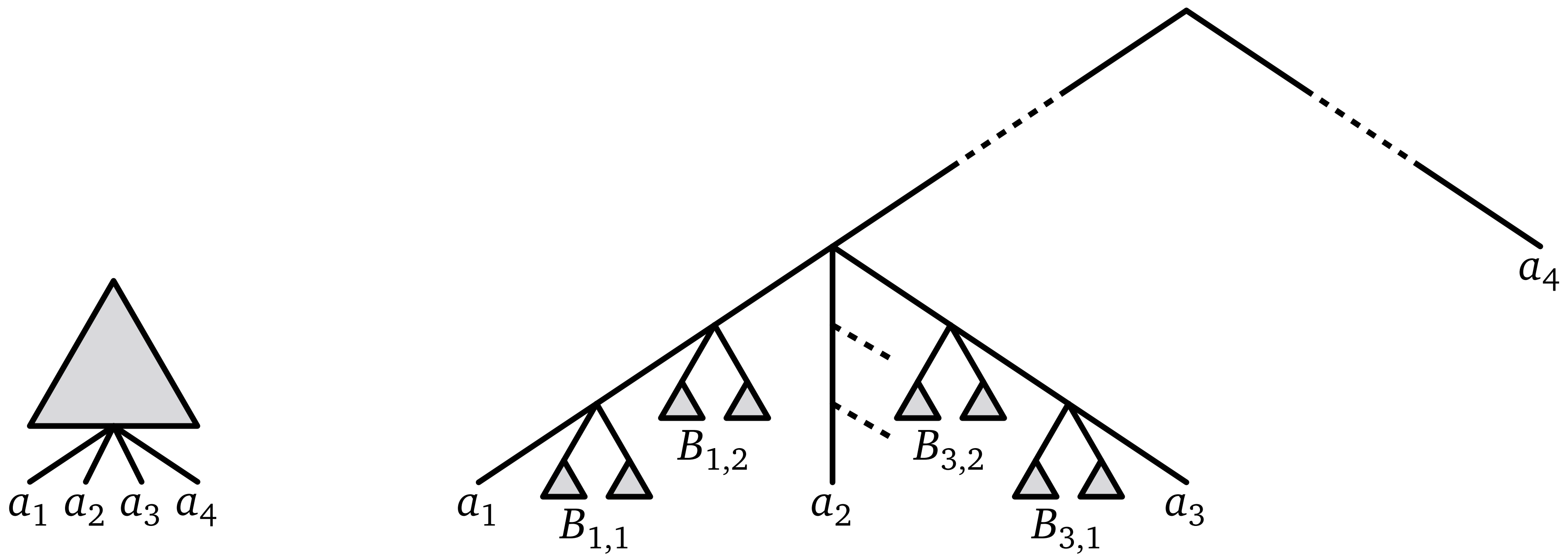
Multifurcating Trees [Whidden/Beiko/Zeh 2012]

Problem 1: What is a meaningful definition of an agreement forest?



An MAF of two multifurcating phylogenies T_1 and T_2 is the largest forest that is an AF of two binary resolutions of T_1 and T_2.

Problem 2: Sibling pairs become sibling groups.



Multifurcating Trees [Whidden/Beiko/Zeh 2012]

It’s FPT, alright . . .

- 5 cases depending on the structure of F_2
- The worst: $I(k) = 1 + 2I(k-1) + 3I(k-2)$

Multifurcating Trees [Whidden/Beiko/Zeh 2012]

It's FPT, alright . . .

- 5 cases depending on the structure of F_2
- The worst: $I(k) = 1 + 2I(k-1) + 3I(k-2)$

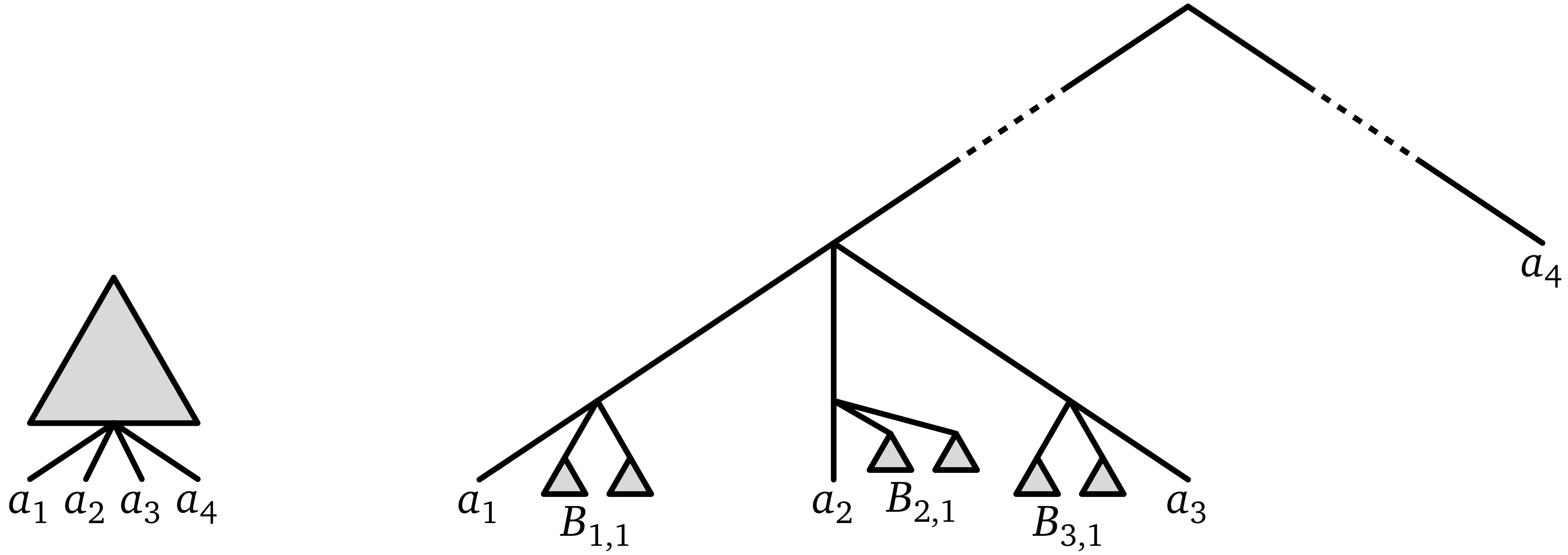
. . . but it ain't fast.

Multifurcating Trees [Whidden/Beiko/Zeh 2012]

It's FPT, alright …

- 5 cases depending on the structure of F_2
- The worst: $I(k) = 1 + 2I(k-1) + 3I(k-2)$

… but it ain't fast.

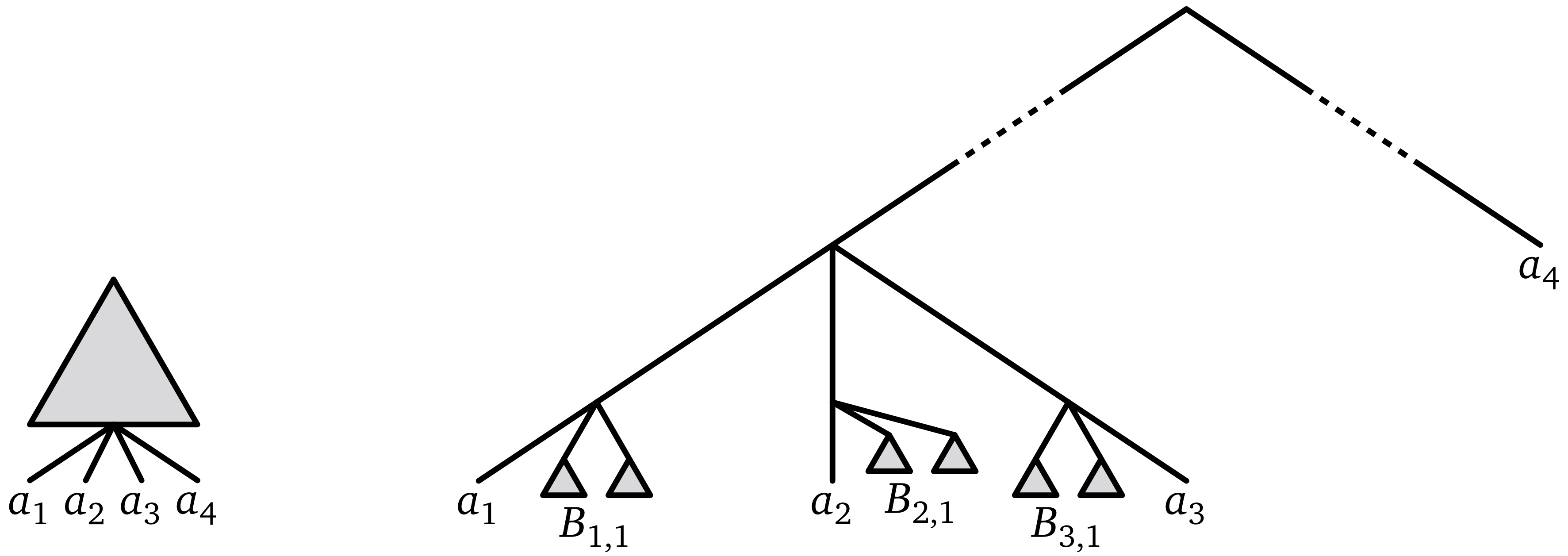


Multifurcating Trees [Whidden/Beiko/Zeh 2012]

It’s FPT, alright . . .

- 5 cases depending on the structure of F_2
- The worst: $I(k) = 1 + 2I(k-1) + 3I(k-2)$

. . . but it ain’t fast.



- Until the protected edges are eliminated, every recursive call becomes a 2-way branch.
- Each such sequence of 2-way branches ends in a “1-way branch”.

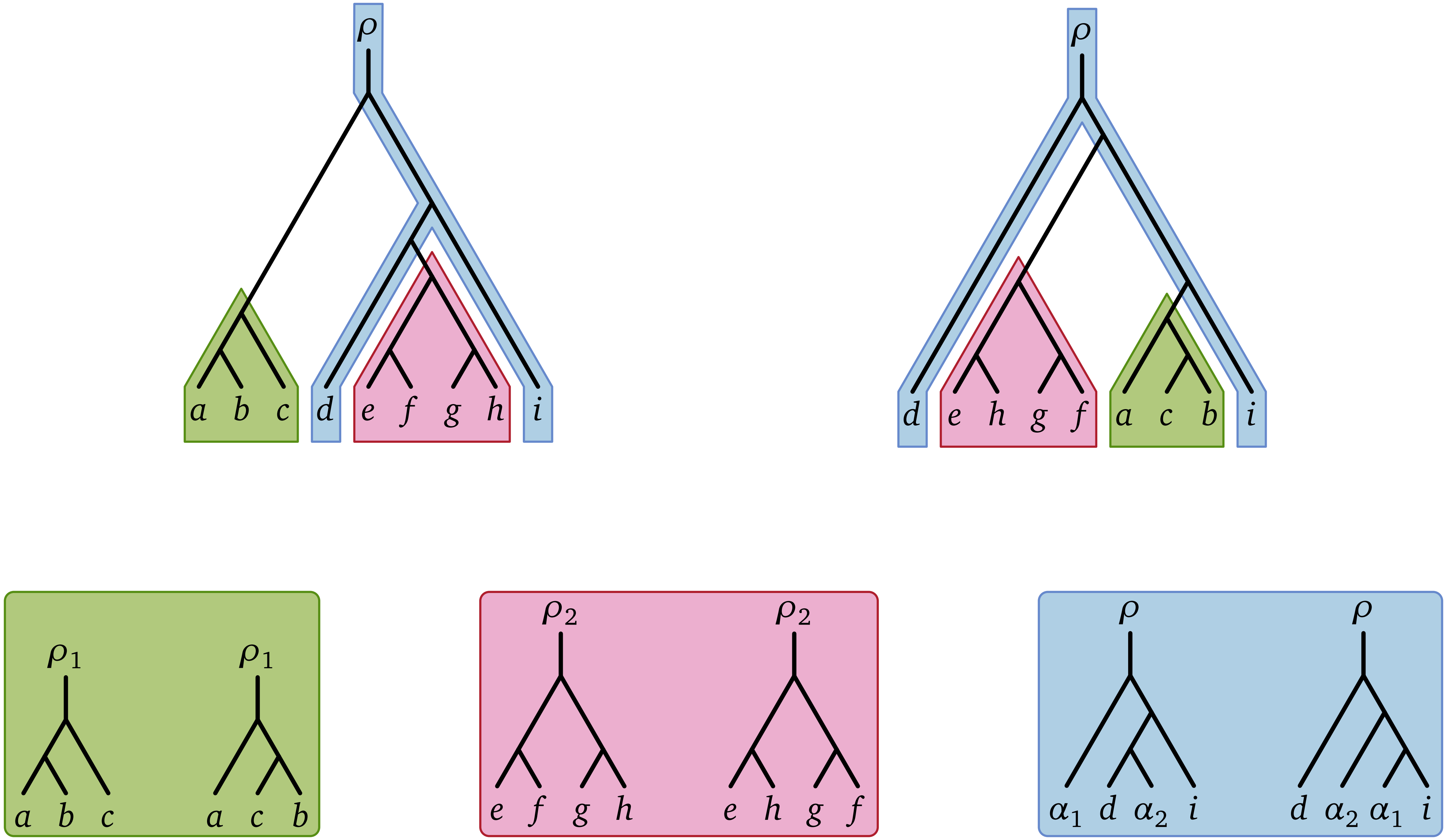
Running time: $O(2.42^k n)$

Binary Trees Even Faster [Whidden/Beiko/Zeh 2012]

- Edge protection idea from the multifurcating algorithm
- A couple of new cases
- A hairy analysis

Running time: $O(2^k n)$

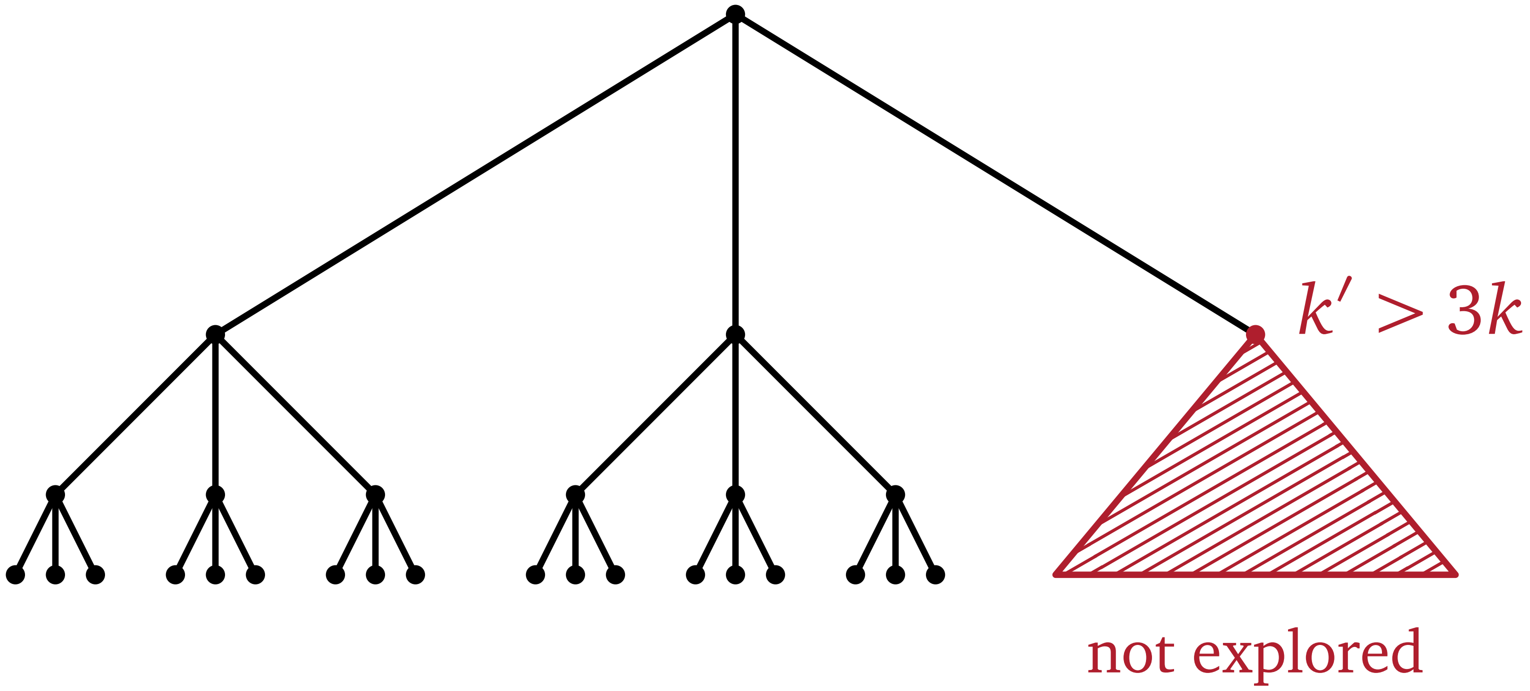
Clustering [Linz/Semple 2009]



An MAF of the two input trees can be computed by computing MAFs of the clusters . . . with a twist.

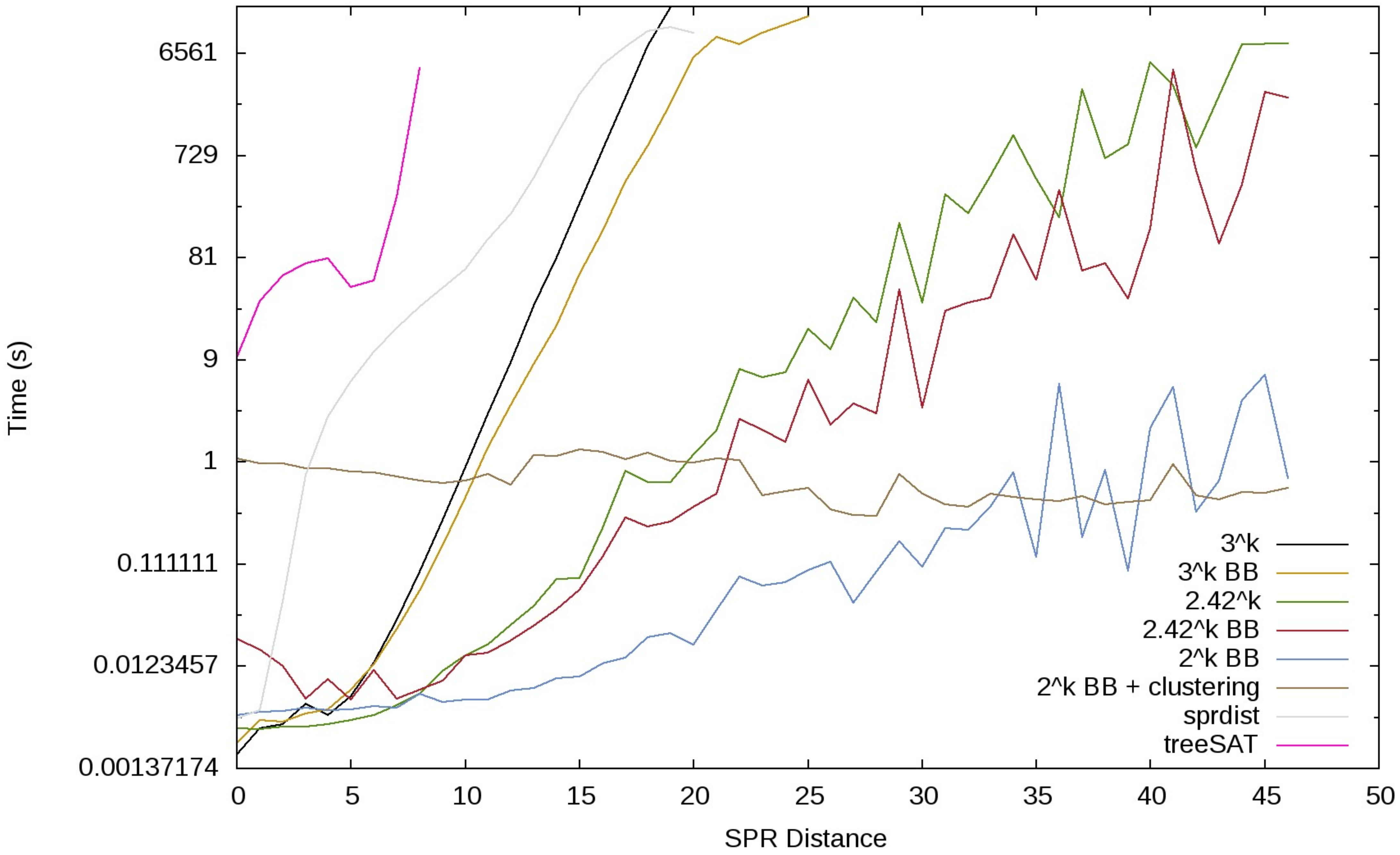
Branch and Bound

- For each invocation, compute 3-approximation k' of number of edges left to be cut.
- If $k' > 3k$, abort.



Added cost per invocation: $O(n)$ [Whidden/Zeh 2009]

Experimental Results



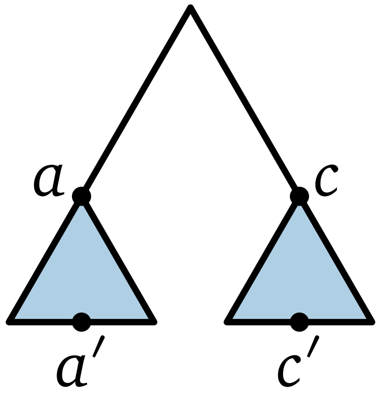
Hybridization [Whidden/Beiko/Zeh 2012]

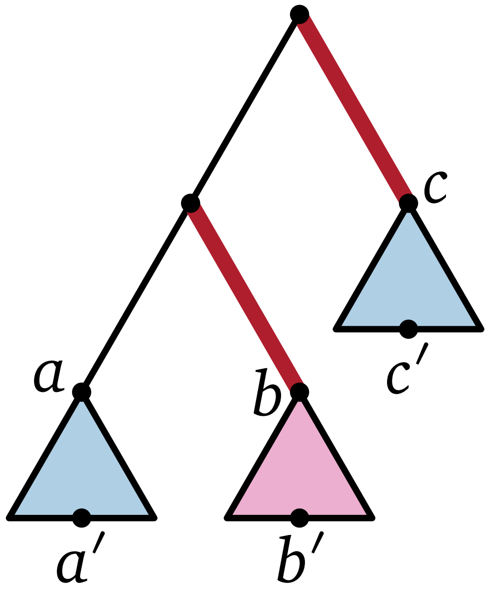
Observation: While F_2 is not an AF of T_1 and T_2, at least one of the branches in each case of the MAF algorithm makes progress towards an MAAF.

Hybridization [Whidden/Beiko/Zeh 2012]

Observation: While F_2 is not an AF of T_1 and T_2, at least one of the branches in each case of the MAF algorithm makes progress towards an MAAF.

Case 3.2′: One pendant subtree on path from a to b in $\dot{F}_2$



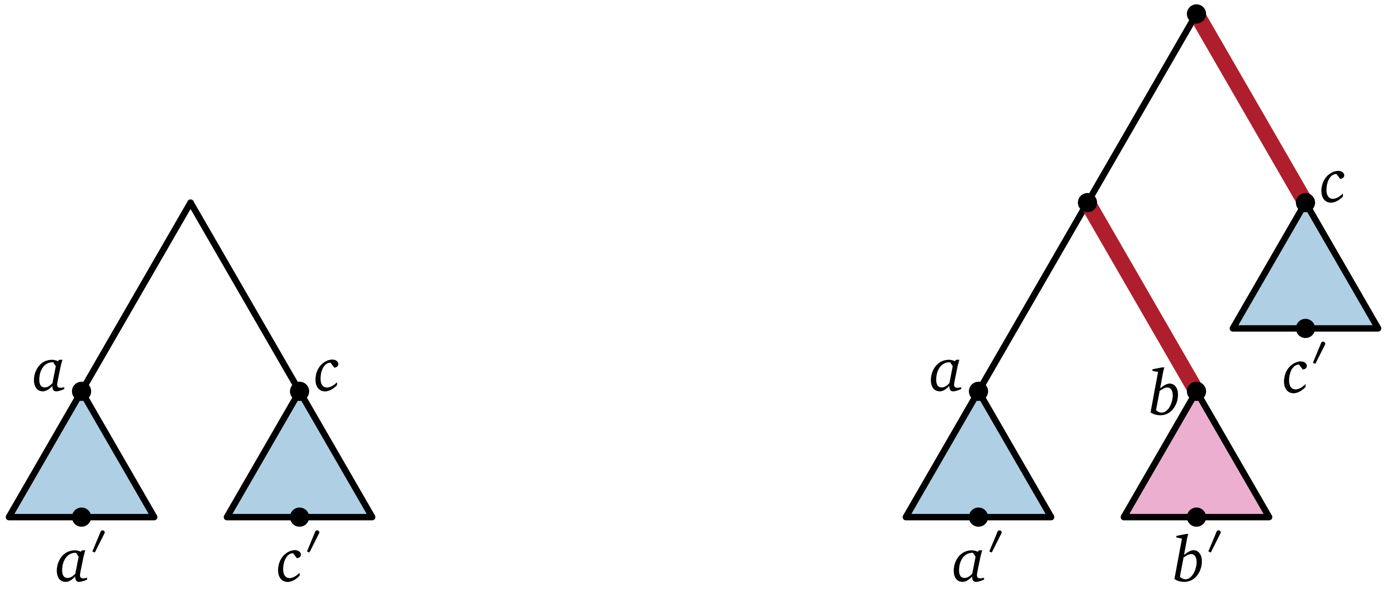


2 recursive calls with parameter $k - 1$

Hybridization [Whidden/Beiko/Zeh 2012]

Observation: While F_2 is not an AF of T_1 and T_2, at least one of the branches in each case of the MAF algorithm makes progress towards an MAAF.

Case 3.2′: One pendant subtree on path from a to b in $\dot{F}_2$

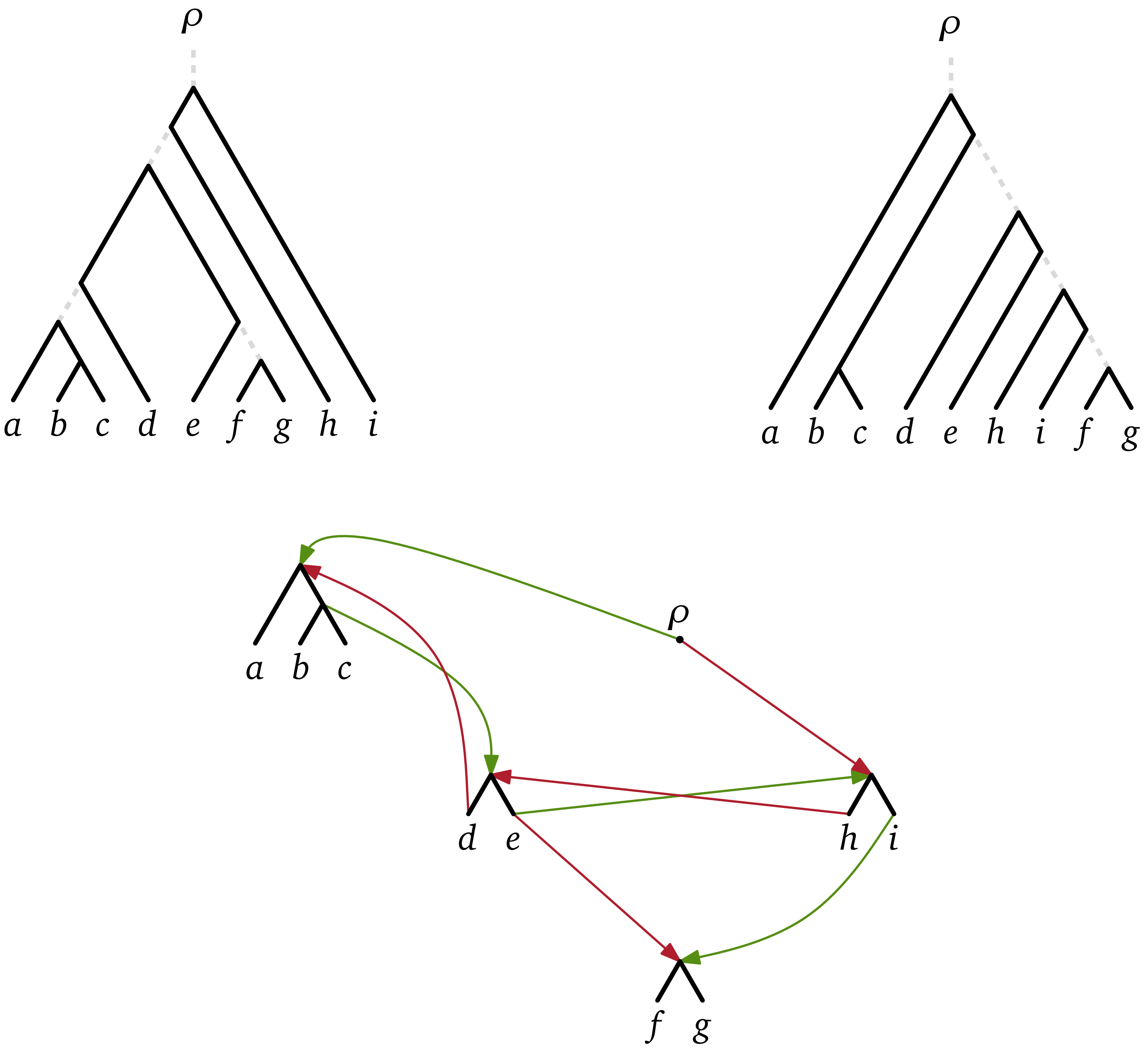


2 recursive calls with parameter $k - 1$

Once an AF is obtained, cut edges to eliminate cycles.

Hybridization [Whidden/Beiko/Zeh 2012]

Cycle graph



Hybridization [Whidden/Beiko/Zeh 2012]

Breaking cycles

- $2k$ edges between components
- For each, may need to eliminate the path to the root of the parent component

$\Rightarrow O(2^{2k} \cdot 2.42^k n) = O(9.68^k n)$ time

Hybridization [Whidden/Beiko/Zeh 2012]

Breaking cycles

- $2k$ edges between components
- For each, may need to eliminate the path to the root of the parent component

$\Rightarrow O(2^{2k} \cdot 2.42^k n) = O(9.68^k n)$ time

Reducing the number of candidate edges

- Can get away with considering only k of the $2k$ edges

$\Rightarrow O(2^k \cdot 2.42^k n) = O(4.84^k n)$ time

Hybridization [Whidden/Beiko/Zeh 2012]

A better analysis

- If the AF has $k' \approx k$ edges, the refinement step considers $\binom{k}{k-k'} \ll 2^k$ choices
- If the AF has $k' \approx 0$ edges, the refinement step considers at most $2^{k'} \ll 2^k$ choices
- If the AF has $k' \approx k/2$ edges, the refinement step considers $\binom{k}{k-k'} \approx 2^k$ choices, but this situation can arise only $2.42^{k'} \ll 2.42^k$ times

$\Rightarrow O(3.18^k n)$ time

Application: SPR Supertrees

SPR Supertrees [Whidden/Zeh/Beiko 2012]

Open problem: Computational complexity of computing an optimal SPR supertree.

SPR Supertrees [Whidden/Zeh/Beiko 2012]

Open problem: Computational complexity of computing an optimal SPR supertree.

Heuristic

- Build up initial supertree
- Iterative improvement using SPR operations

SPR Supertrees [Whidden/Zeh/Beiko 2012]

Open problem: Computational complexity of computing an optimal SPR supertree.

Heuristic

- Build up initial supertree
- Iterative improvement using SPR operations

Initial tree construction

- Start with 4-leaf tree consistent with one of the gene trees
- Attach one leaf at a time
- For each leaf, choose the location that minimizes SPR distance

SPR Supertrees [Whidden/Zeh/Beiko 2012]

Open problem: Computational complexity of computing an optimal SPR supertree.

Heuristic

- Build up initial supertree
- Iterative improvement using SPR operations

Initial tree construction

- Start with 4-leaf tree consistent with one of the gene trees
- Attach one leaf at a time
- For each leaf, choose the location that minimizes SPR distance

Iterative improvement

- Try all $O(n^2)$ SPR operations on current supertree and choose the one that minimizes the SPR distance from gene trees

SPR Supertrees [Whidden/Zeh/Beiko 2012]

Limit number of SPR moves to consider

- Consider only SPR operations across $r = O(1)$ edges $\Rightarrow O(n)$ moves

$\Rightarrow O(tn)$ exact SPR computations

- Rank moves based on approximate SPR distance of resulting tree to gene trees
- Try moves in this order and choose the first one that gives an improvement

$\Rightarrow O(tn^2)$ approximate SPR computations $+ O(t)$ exact SPR computations

SPR Supertrees [Whidden/Zeh/Beiko 2012]

Limit number of SPR moves to consider

- Consider only SPR operations across $r = O(1)$ edges $\Rightarrow O(n)$ moves

$\Rightarrow O(tn)$ exact SPR computations

- Rank moves based on approximate SPR distance of resulting tree to gene trees
- Try moves in this order and choose the first one that gives an improvement

$\Rightarrow O(tn^2)$ approximate SPR computations $+ O(t)$ exact SPR computations

MAF-driven improvements

- In each iteration, every gene tree initiates one SPR move on supertree that reduces its distance by one
- Choose this move using the MAF of gene tree and supertree

$\Rightarrow t$ exact SPR computations

Conclusions

Ongoing and Future Work

Faster supertree search

- FPT approximation to handle really large trees

Ongoing and Future Work

Faster supertree search

- FPT approximation to handle really large trees

Faster M(A)AF algorithms

- Substantially break the 2^k barrier to handle trees with 1,000s of leaves

Ongoing and Future Work

Faster supertree search

- FPT approximation to handle really large trees

Faster M(A)AF algorithms

- Substantially break the 2^k barrier to handle trees with 1,000s of leaves

Compute *all* M(A)AFs [Abrecht et al. 2012]

- Provide more biological insight