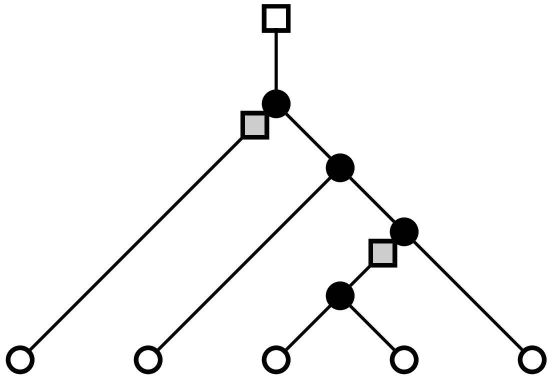
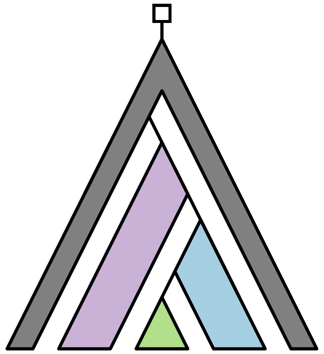
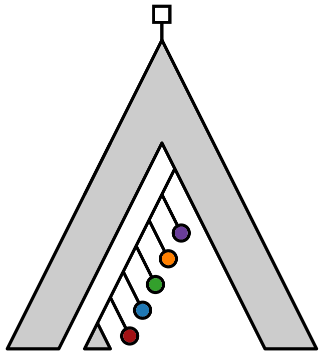
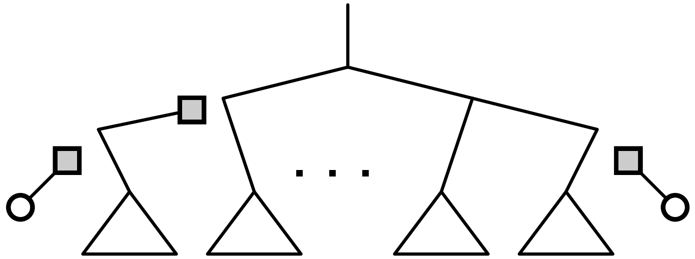
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Advanced Algorithms

Rearrangement distance of phylogenetic trees

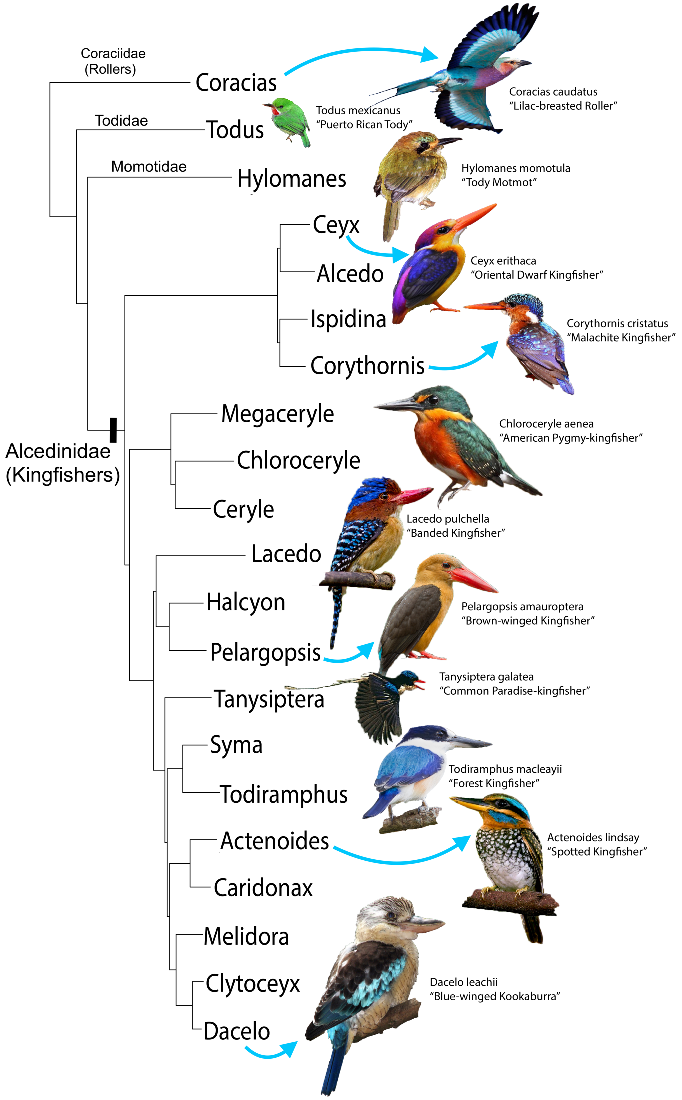
Kernelisation, fpt and approximation algorithm

Jonathan Klawitter · WS20

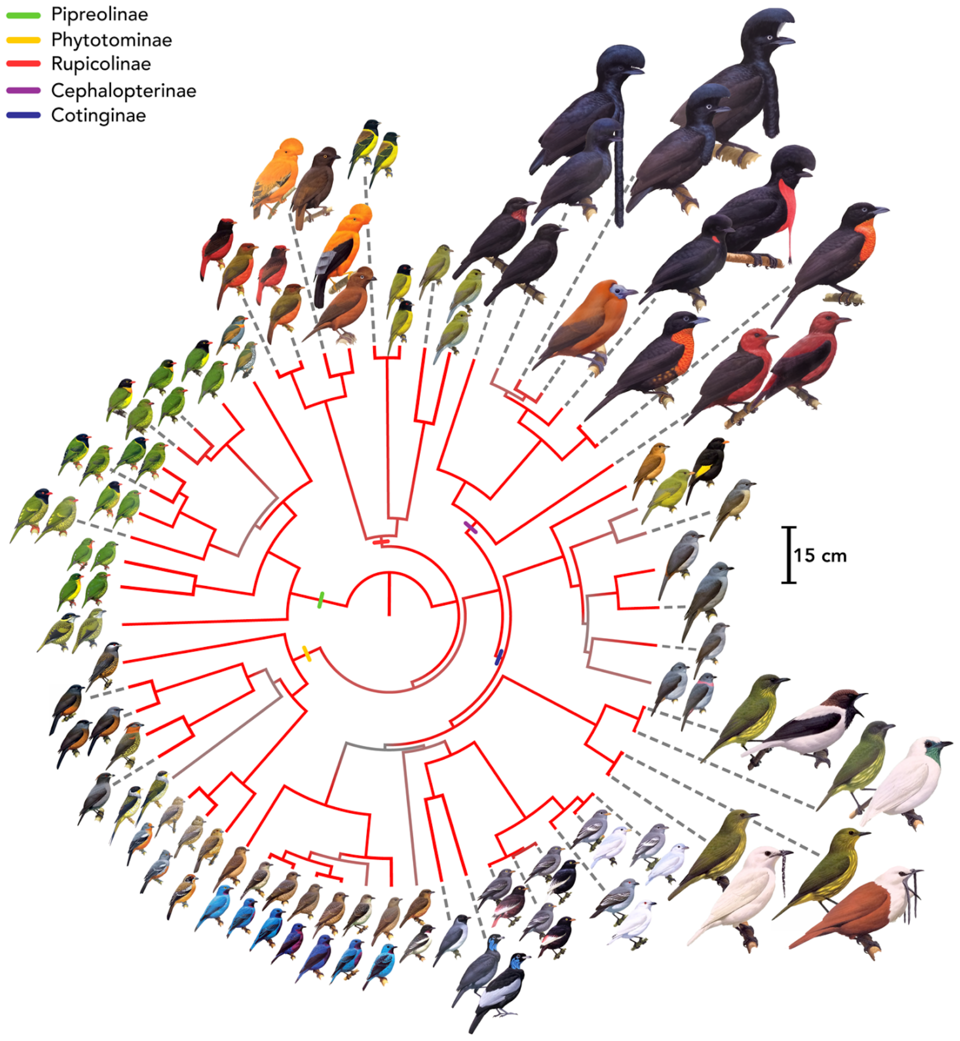





Phylogenetic trees

… represent the evolutionary history of a set of taxa.


Coraciidae (Rollers)
Coracias
Todidae
Todus
Momotidae
Hylomanes
Ceyx
Alcedo
Ispidina
Corythornis
Megaceryle
Chloroceryle
Ceryle
Lacedo
Halcyon
Pelargopsis
Tanysiptera
Syma
Todiramphus
Actenoides
Caridonax
Melidora
Clytoceyx
Dacelo
Alcedinidae (Kingfishers)
Coracias caudatus “Lilac-breasted Roller”
Todus mexicanus “Puerto Rican Tody”
Hylomanes momotula “Tody Motmot”
Ceyx erithaca “Oriental Dwarf Kingfisher”
Corythornis cristatus “Malachite Kingfisher”
Chloroceryle aenea “American Pygmy-kingfisher”
Lacedo pulchella “Banded Kingfisher”
Pelargopsis amauroptera “Brown-winged Kingfisher”
Tanysiptera galatea “Common Paradise-kingfisher”
Todiramphus macleayii “Forest Kingfisher”
Actenoides lindsay “Spotted Kingfisher”
Dacelo leachii “Blue-winged Kookaburra”

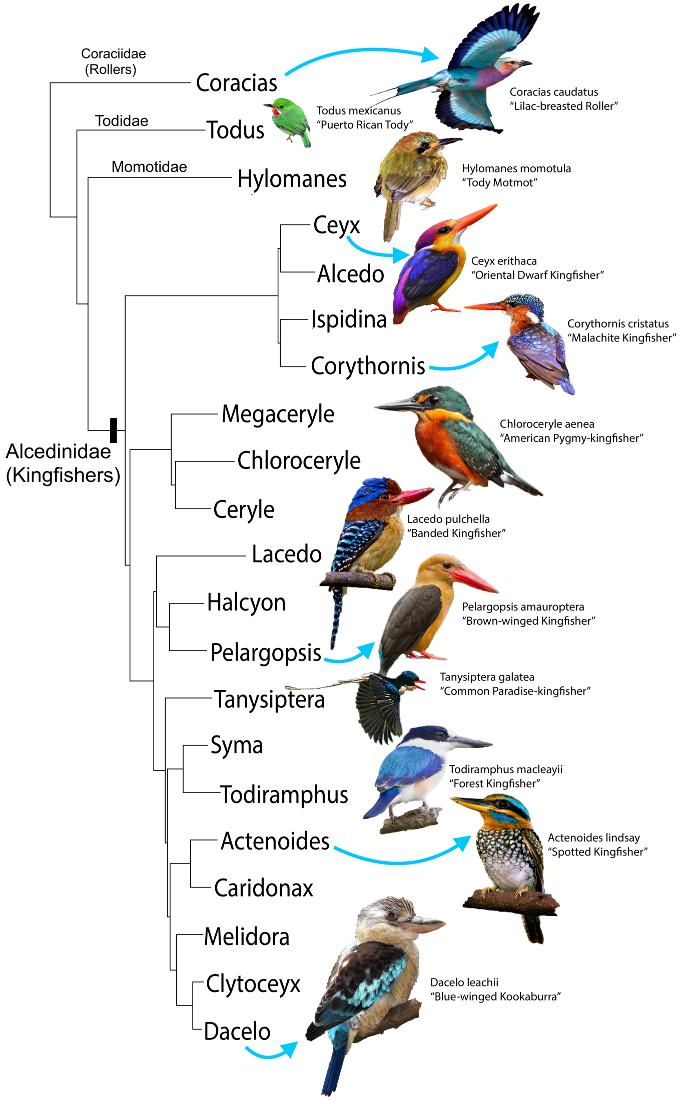
by Jenna McCullough 2016



by Berv & Prum 2014

Phylogenetic trees

... represent the evolutionary history of a set of taxa.

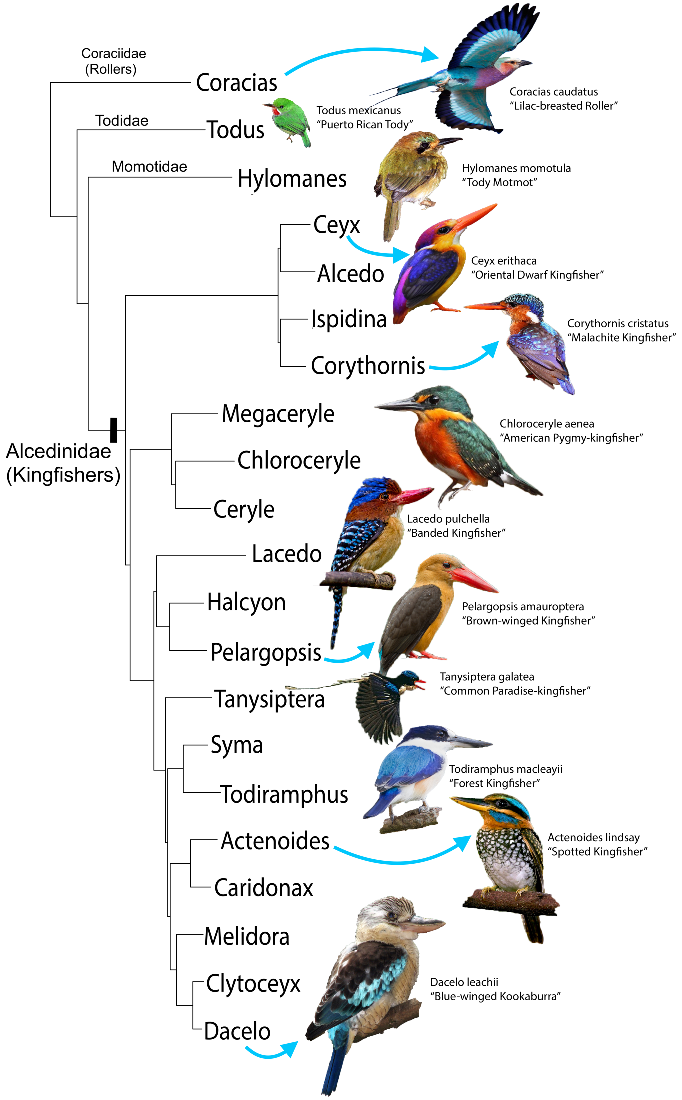


by Jenna McCullough 2016

- Leaves are labelled with taxa.
- Each taxon represents a species, population, individual organism, gene, chromosome, . . .

Phylogenetic trees

... represent the evolutionary history of a set of taxa.

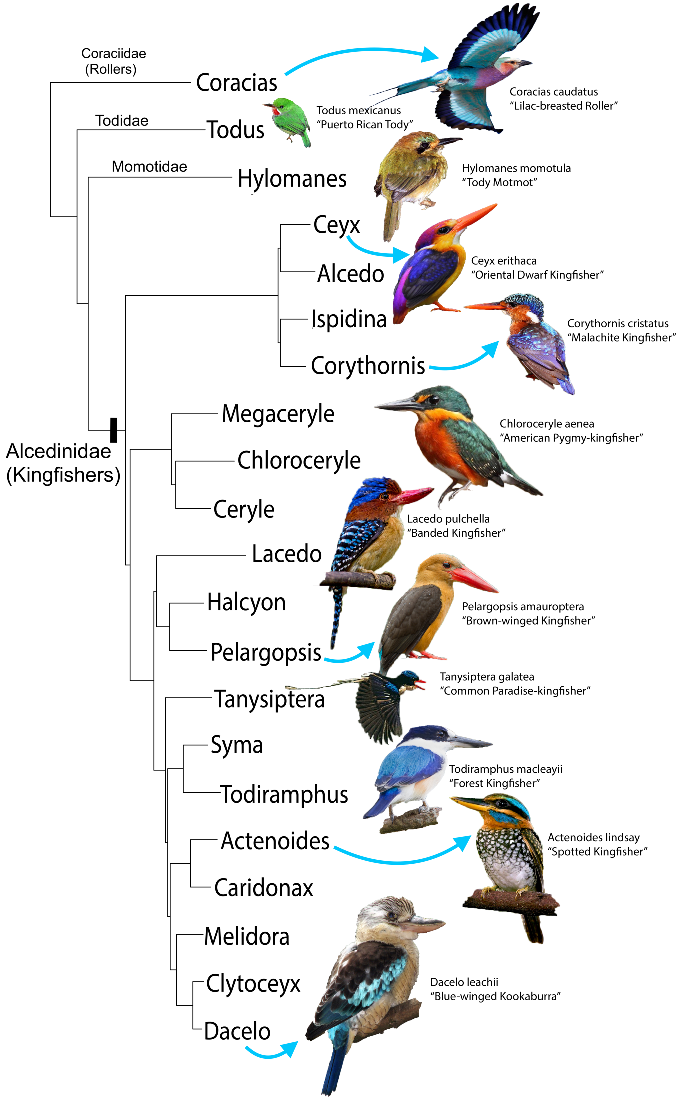


by Jenna McCullough 2016

- Leaves are labelled with taxa.
- Each taxon represents a species, population, individual organism, gene, chromosome, ...
- Edge lenghts represents amount of time passed or genetic distance.

Phylogenetic trees

... represent the evolutionary history of a set of taxa.

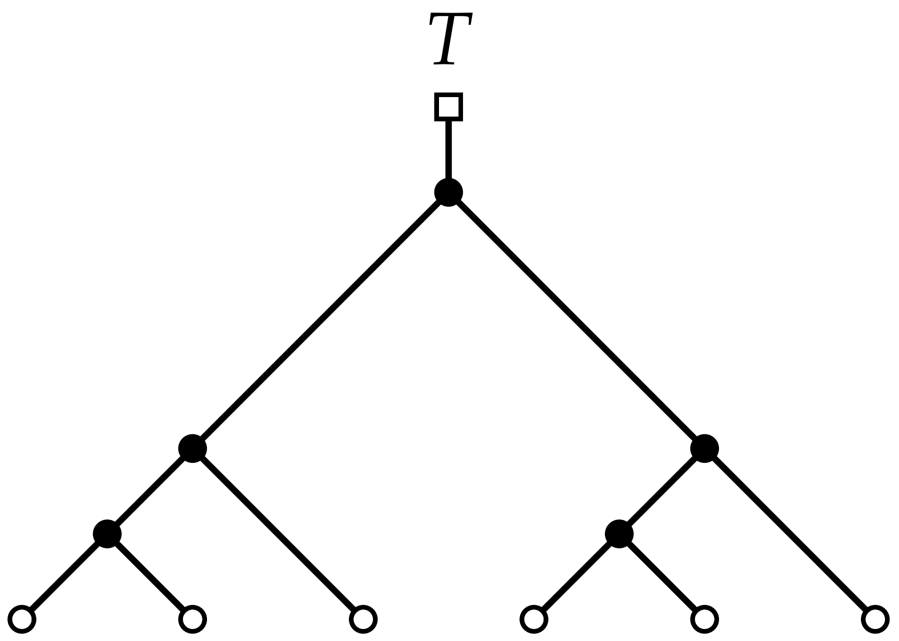


by Jenna McCullough 2016

- Leaves are labelled with taxa.
- Each taxon represents a species, population, individual organism, gene, chromosome, . . .
- Edge lenghts represents amount of time passed or genetic distance.
- Inference methods compute a phylogenetic tree based on some model and data.

Phylogenetic trees

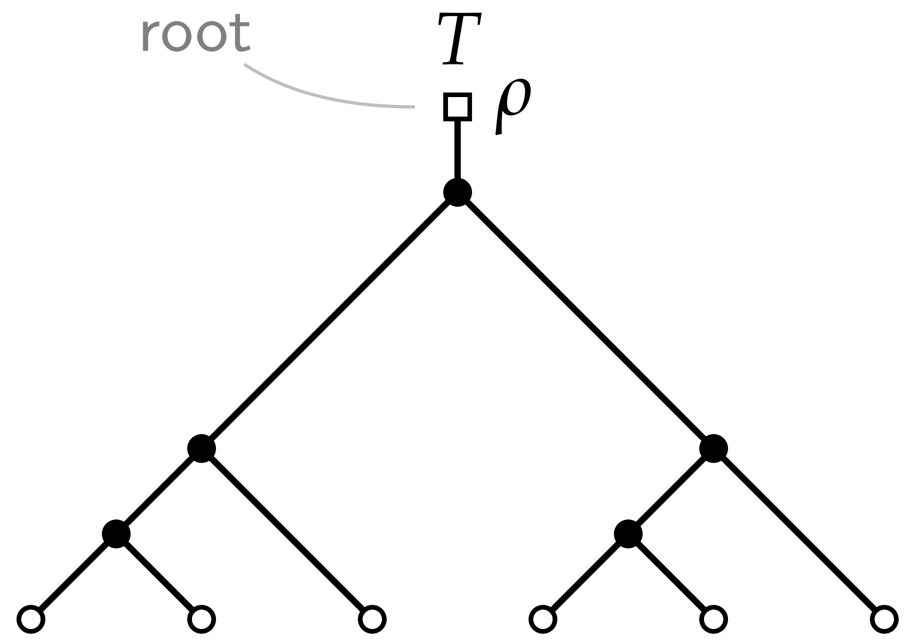
Let $X = \{1, 2, 3, \ldots n\}$.
A **(rooted, binary) phylogenetic tree** T
is a rooted tree with the following
properties:



Phylogenetic trees

Let $X = \{1, 2, 3, \ldots n\}$.
A **(rooted, binary) phylogenetic tree** T is a rooted tree with the following properties:

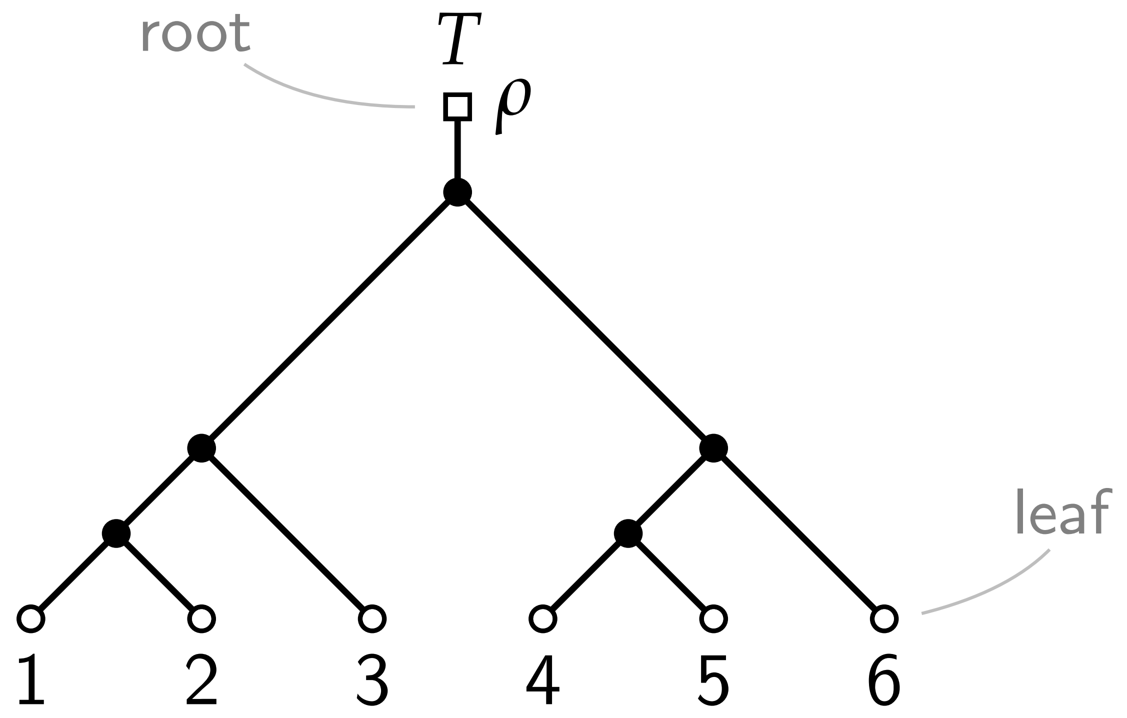
- The unique **root** is labeled ρ and has outdegree 1.



Phylogenetic trees

Let $X = \{1, 2, 3, \ldots n\}$.
A **(rooted, binary) phylogenetic tree** T
is a rooted tree with the following properties:

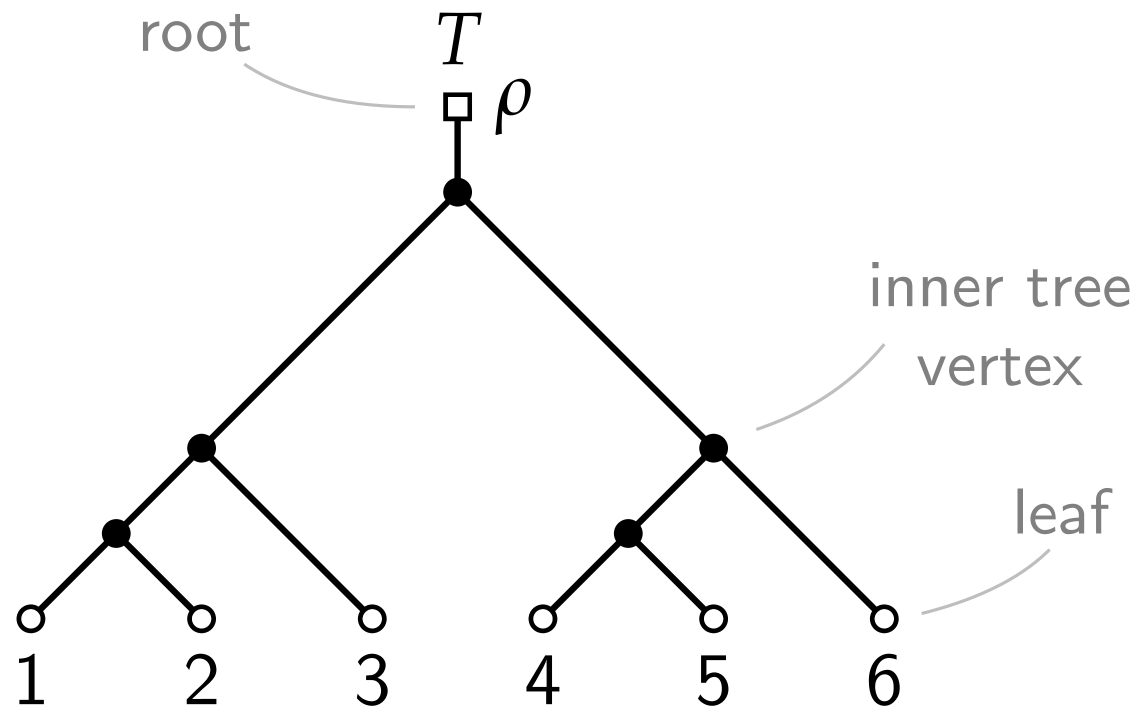
- The unique **root** is labeled ρ and has outdegree 1.
- The leaves are bijectively labeled by X.



Phylogenetic trees

Let $X = \{1, 2, 3, \dots n\}$.
A **(rooted, binary) phylogenetic tree** T is a rooted tree with the following properties:

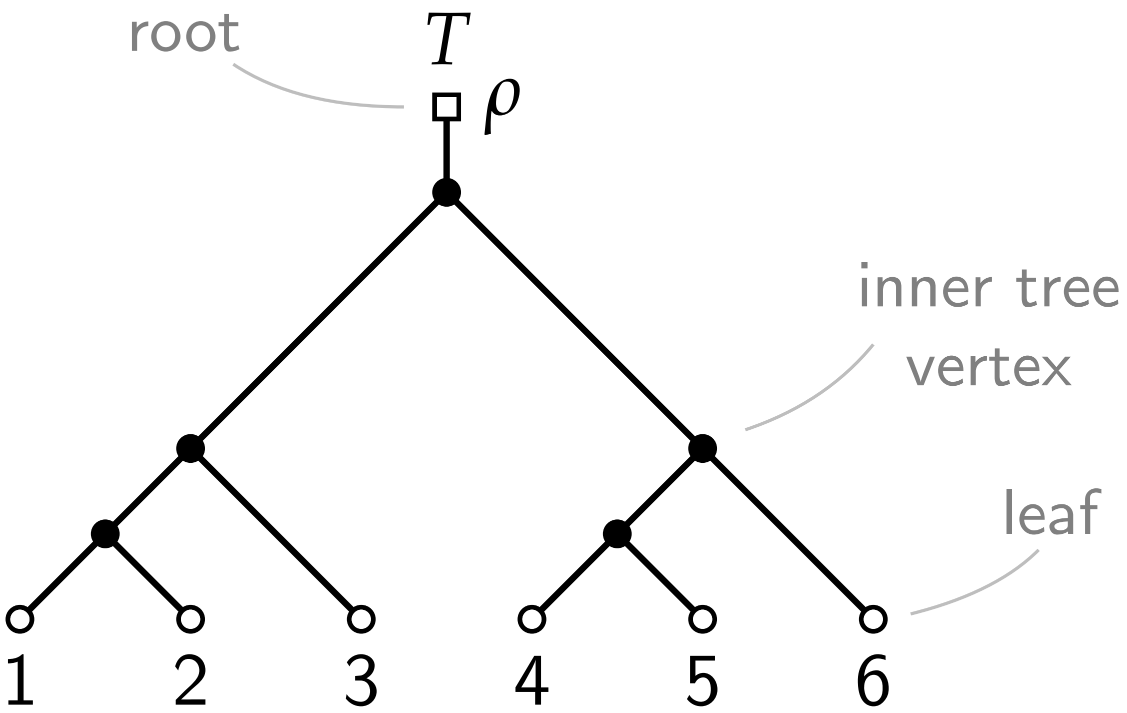
- The unique **root** is labeled ρ and has outdegree 1.
- The leaves are bijectively labeled by X.
- All other vertices have indegree 1 and outdegree 2.



Phylogenetic trees

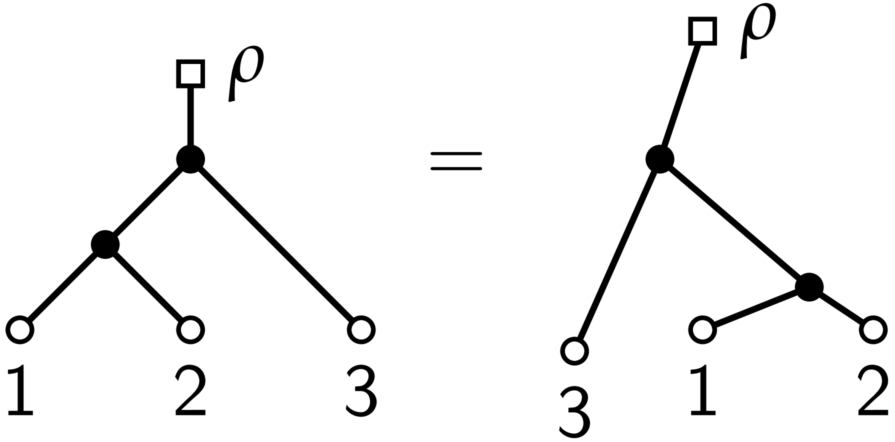
Let $X = \{1, 2, 3, \ldots n\}$.
A **(rooted, binary) phylogenetic tree** T is a rooted tree with the following properties:

- The unique **root** is labeled ρ and has outdegree 1.
- The leaves are bijectively labeled by X.
- All other vertices have indegree 1 and outdegree 2.



Remarks.
Here, in our definition

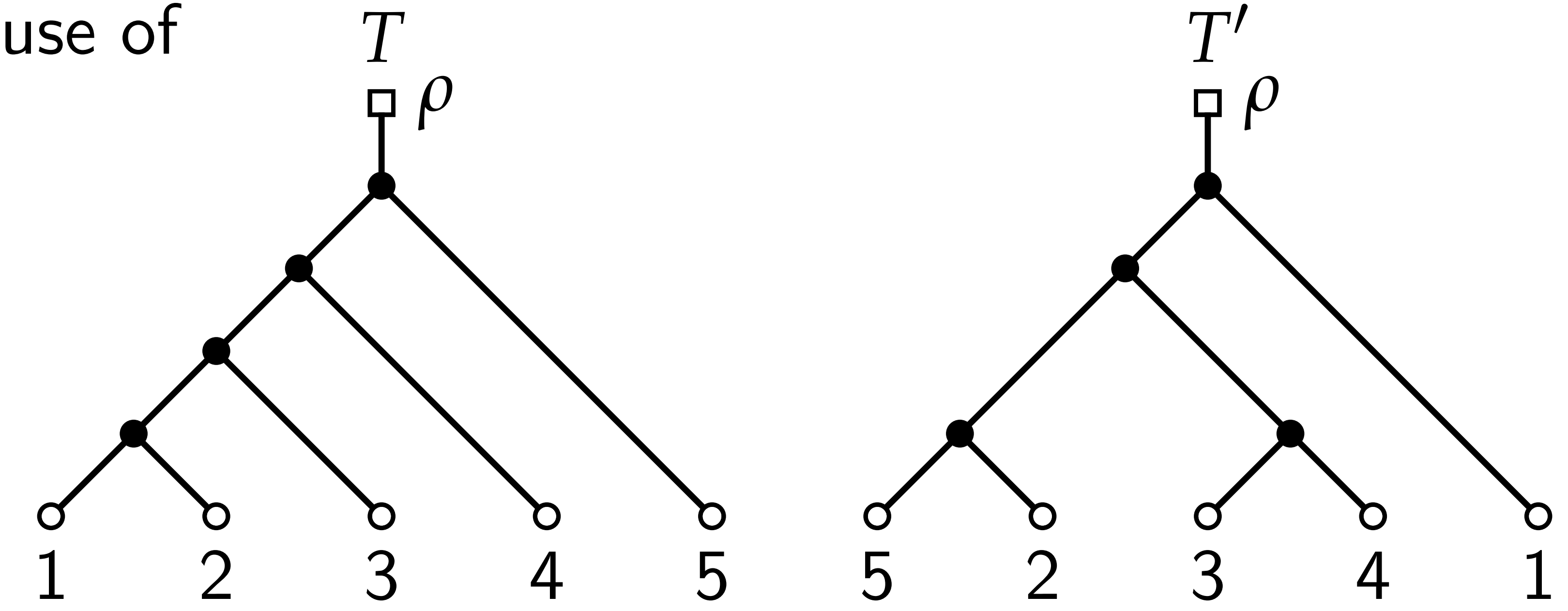
- vertices have **no heights** and
- the order of leaves does not matter.



Problem

For the same taxa, we may infer **different** phylogenetic trees because of the use of

- different inference methods,
- different models, or
- different data.

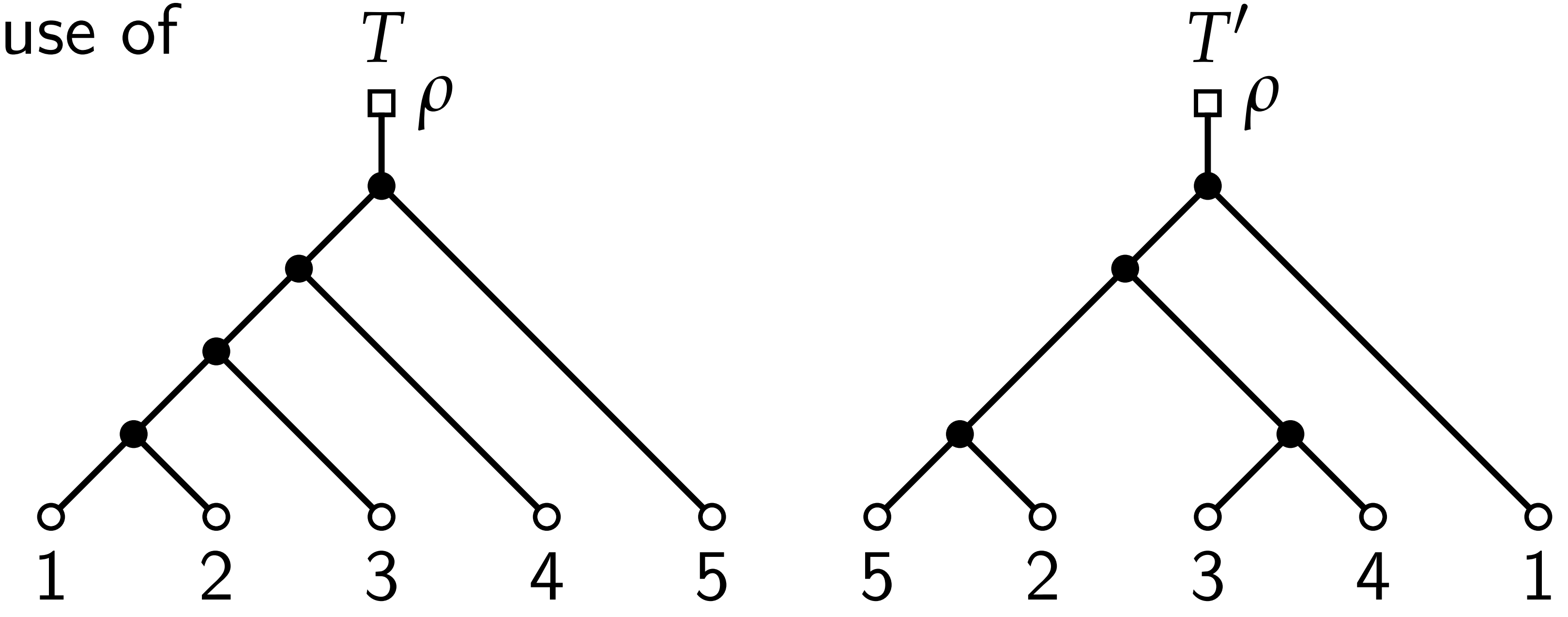


Problem

For the same taxa, we may infer **different** phylogenetic trees because of the use of

- different inference methods,
- different models, or
- different data.

We want to be able to **compare** different phylogenetic trees. How?

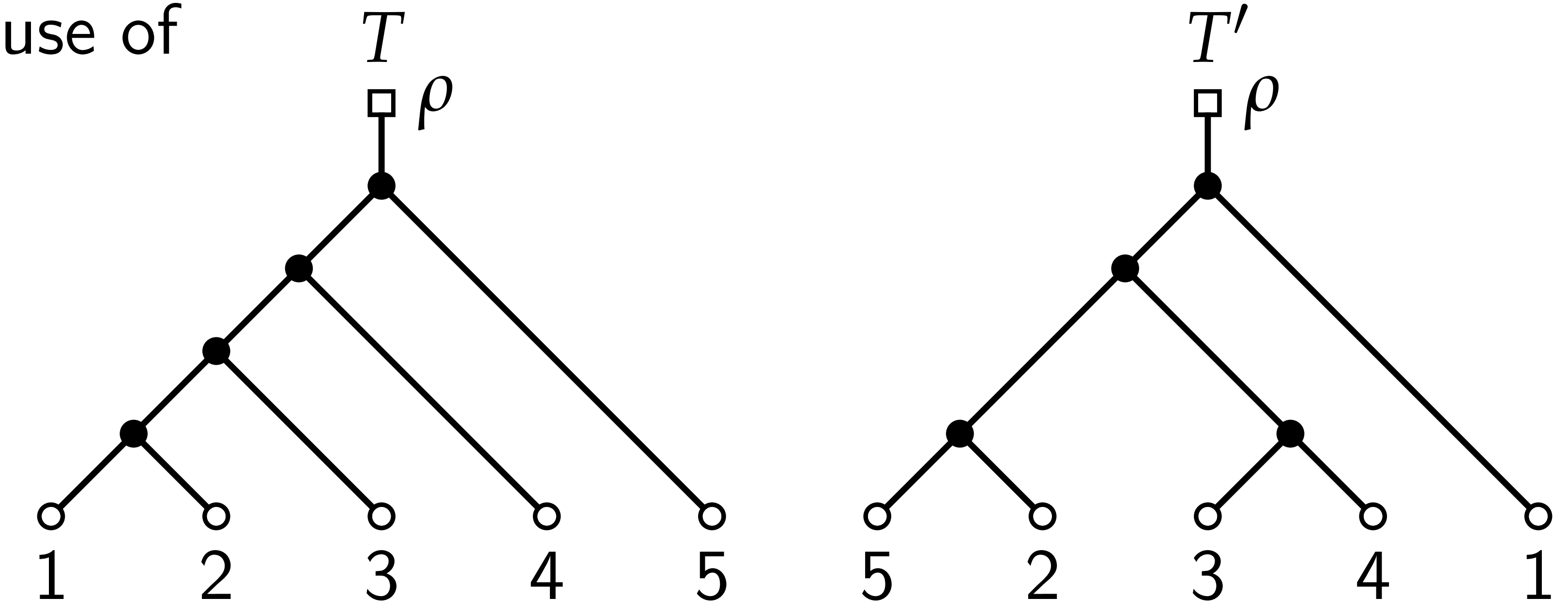


Problem

For the same taxa, we may infer **different** phylogenetic trees because of the use of

- different inference methods,
- different models, or
- different data.

We want to be able to **compare** different phylogenetic trees. How?



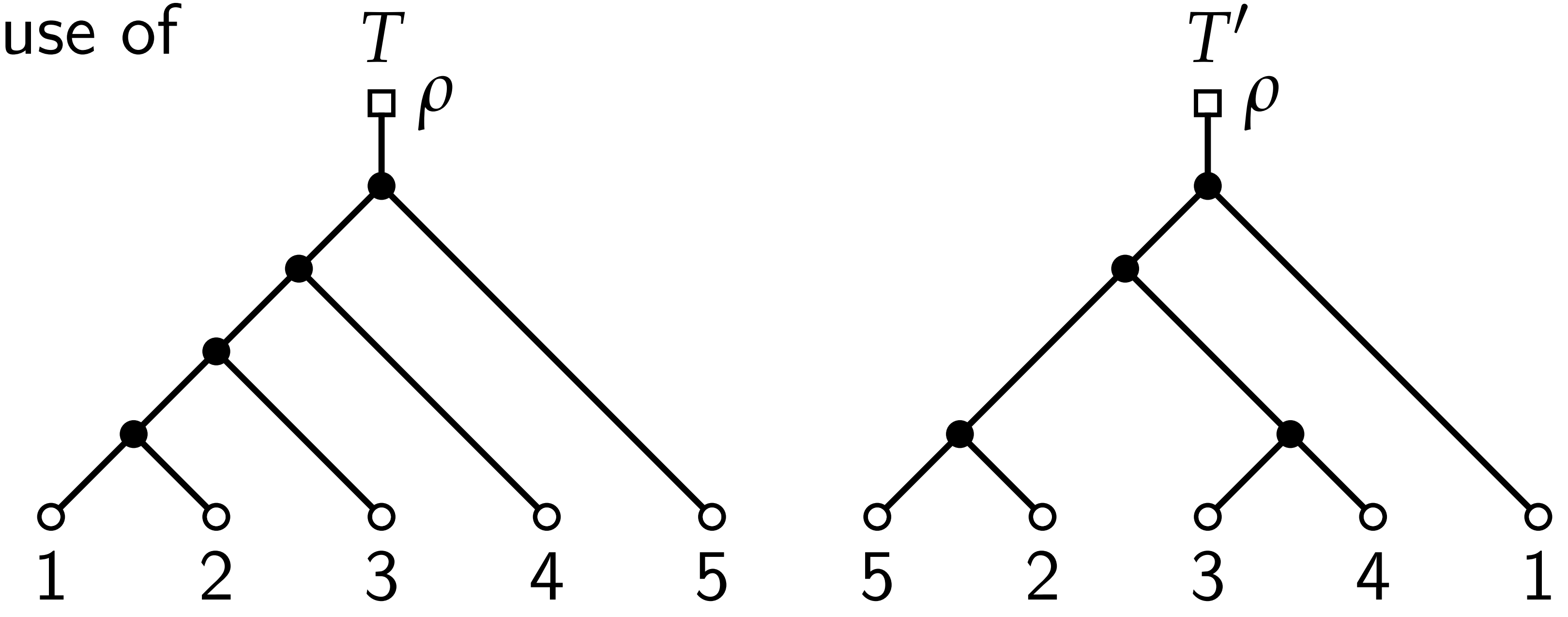
Goal.
Define a **metric** on phylogenetic trees on X and devise algorithms to compute it.

Problem

For the same taxa, we may infer **different** phylogenetic trees because of the use of

- different inference methods,
- different models, or
- different data.

We want to be able to **compare** different phylogenetic trees. How?

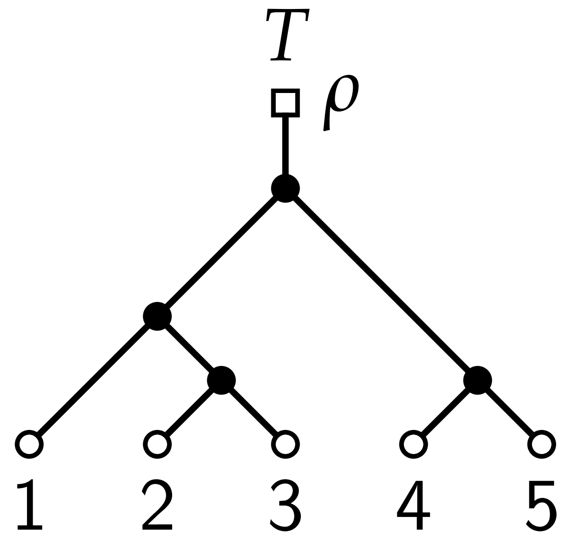


Goal.
Define a **metric** on phylogenetic trees on X and devise algorithms to compute it.

Idea.
Count the number of **rearrangement operations** that are necessary to transform T into T'.

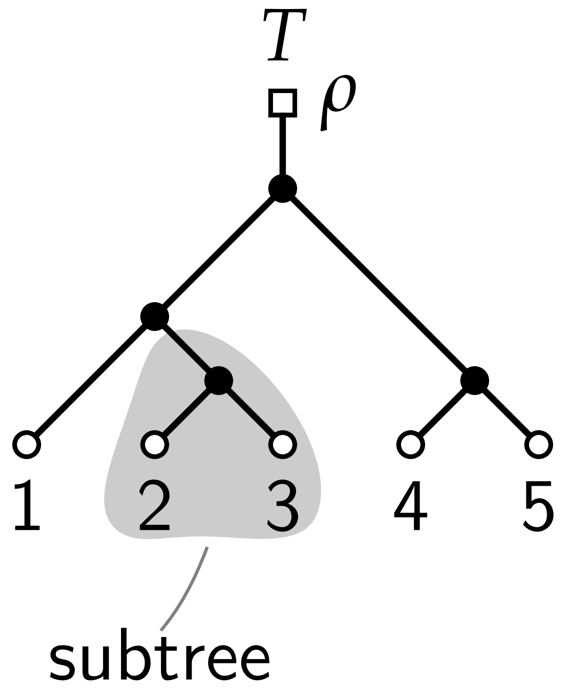
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



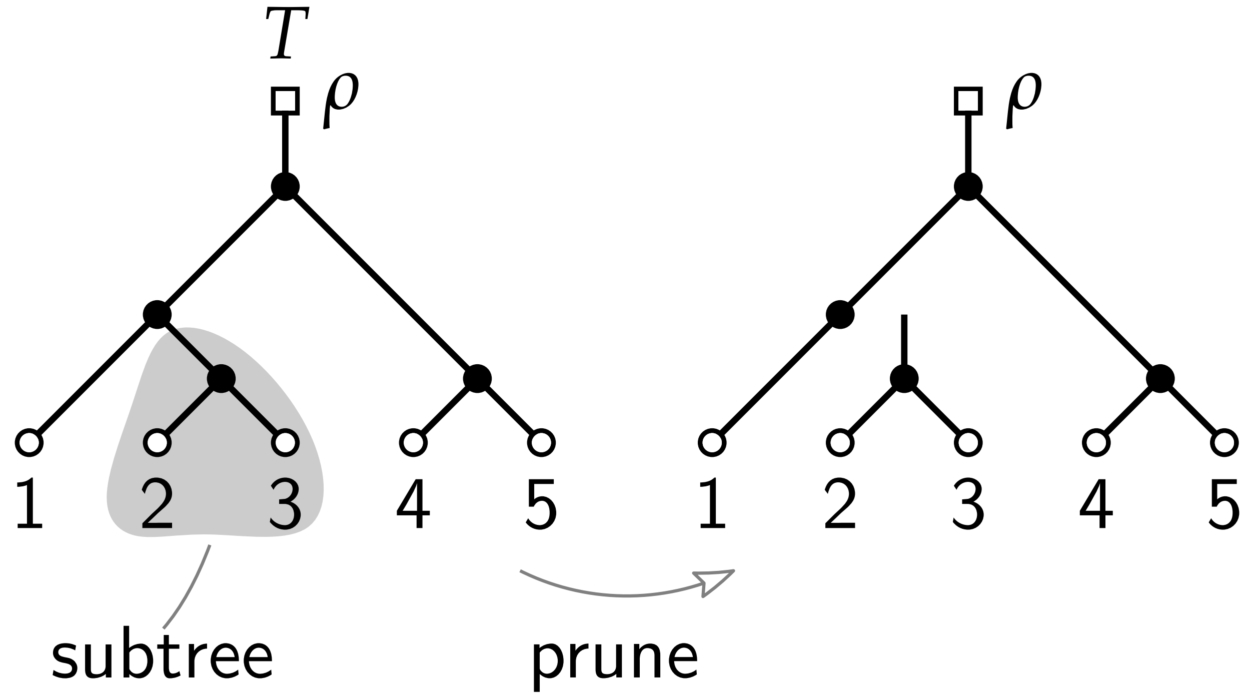
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



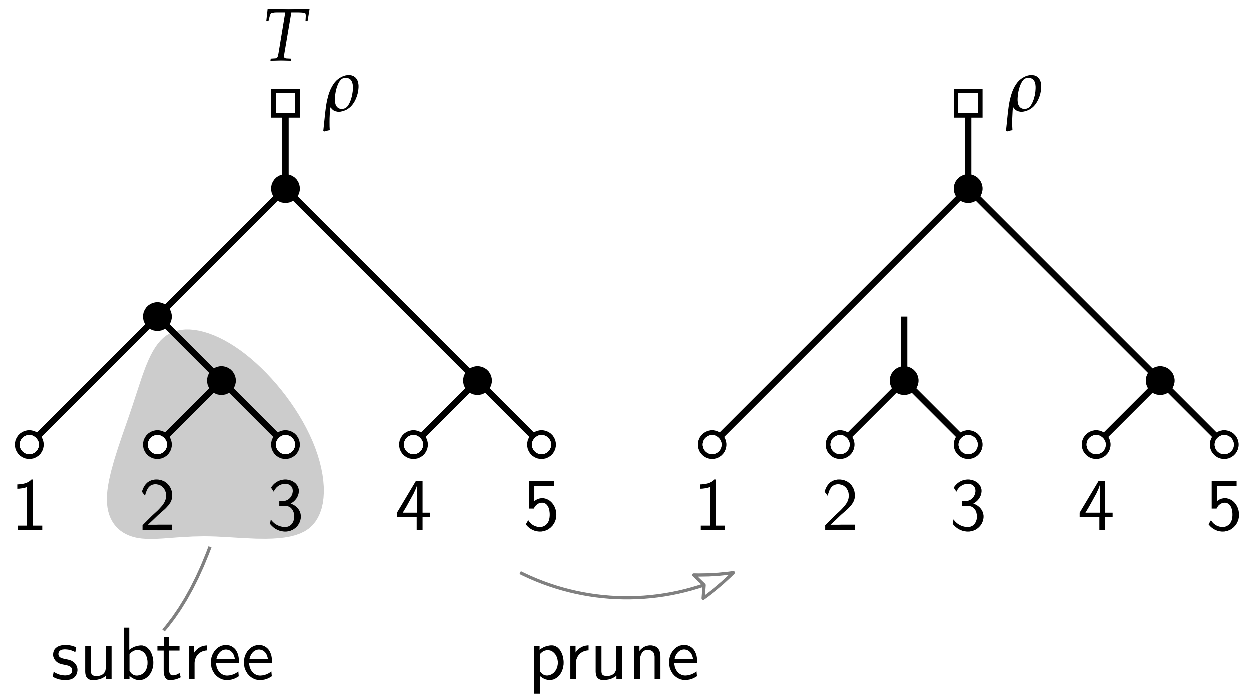
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



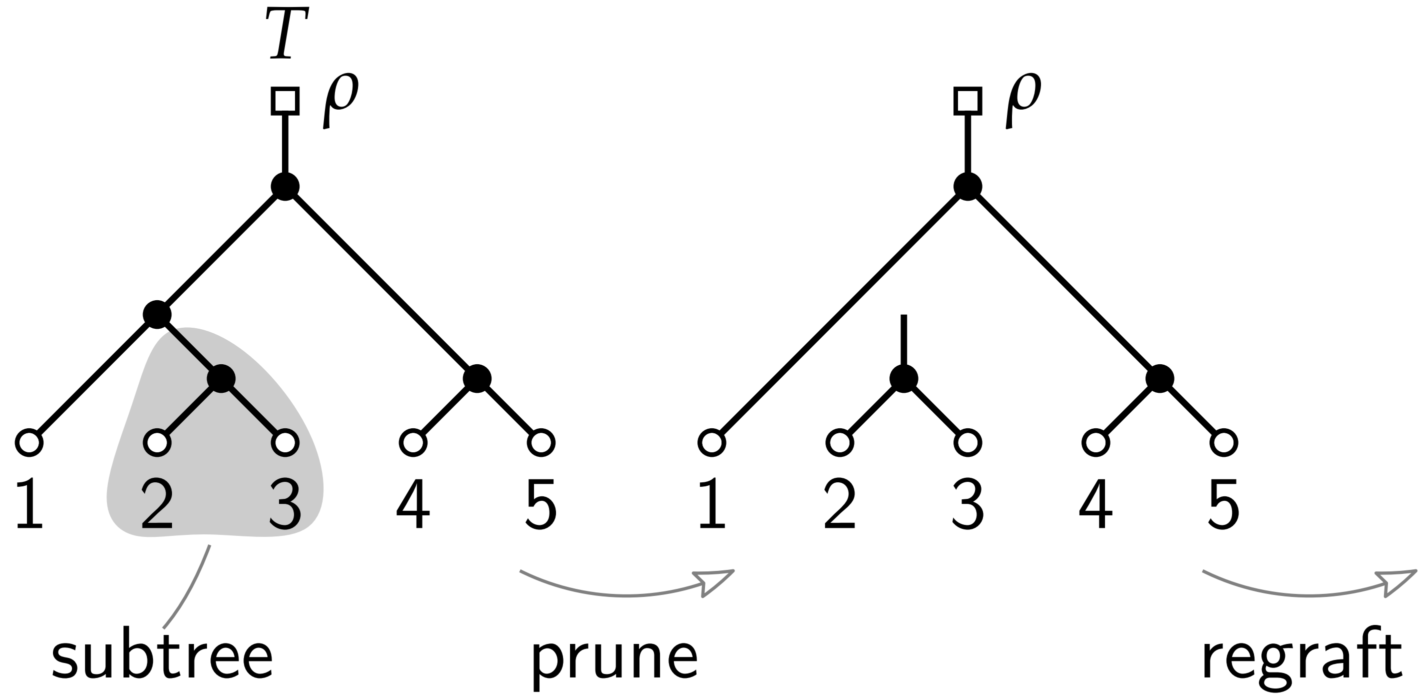
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



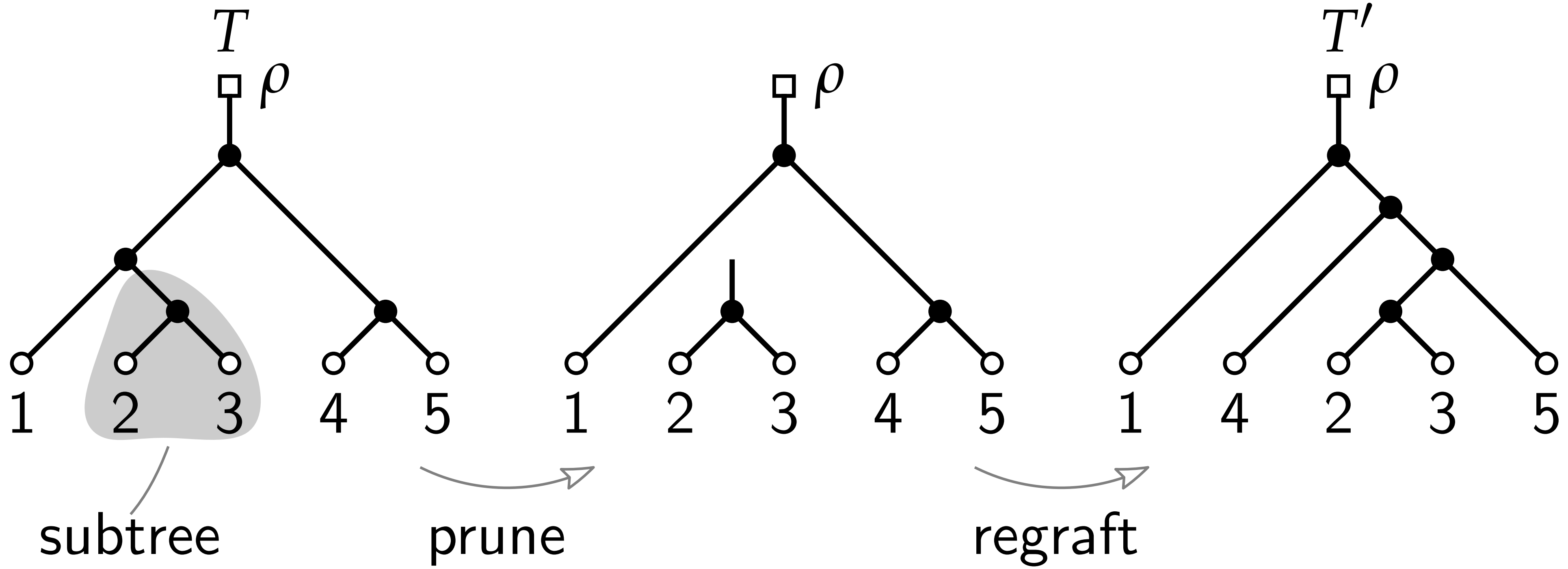
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



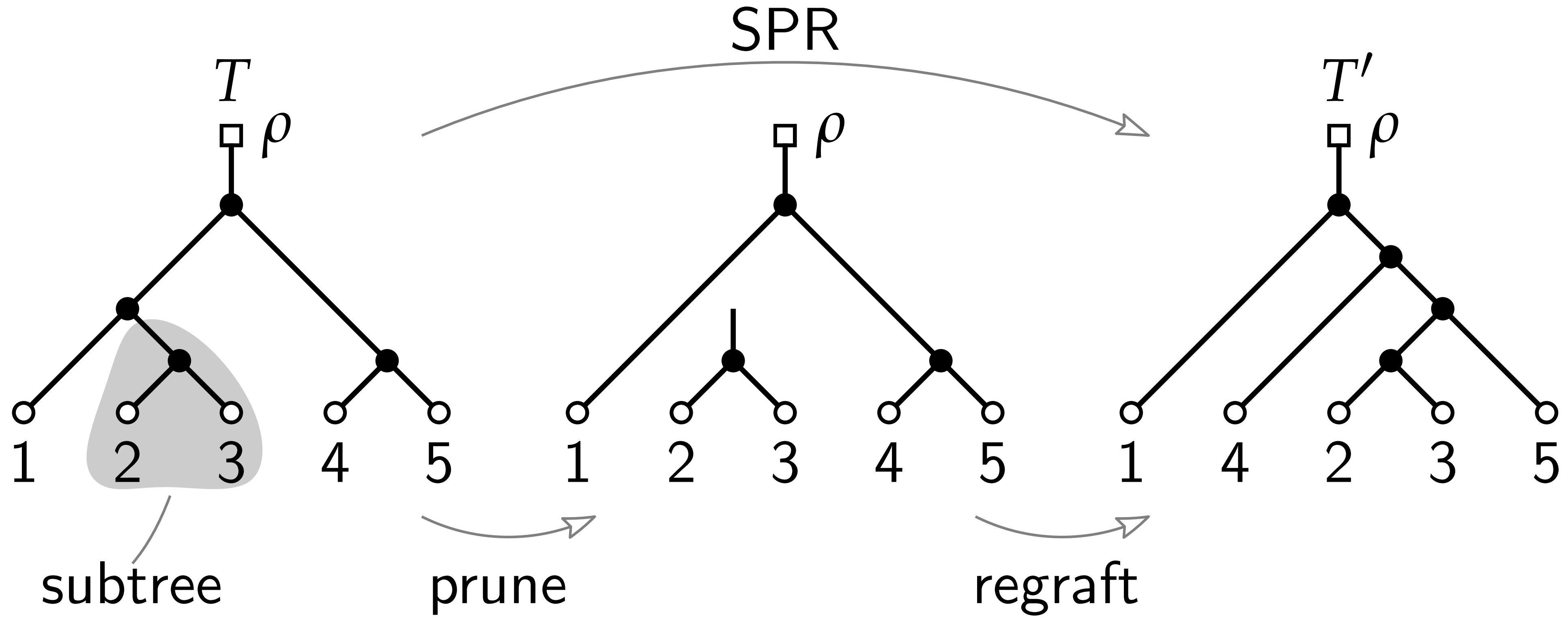
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



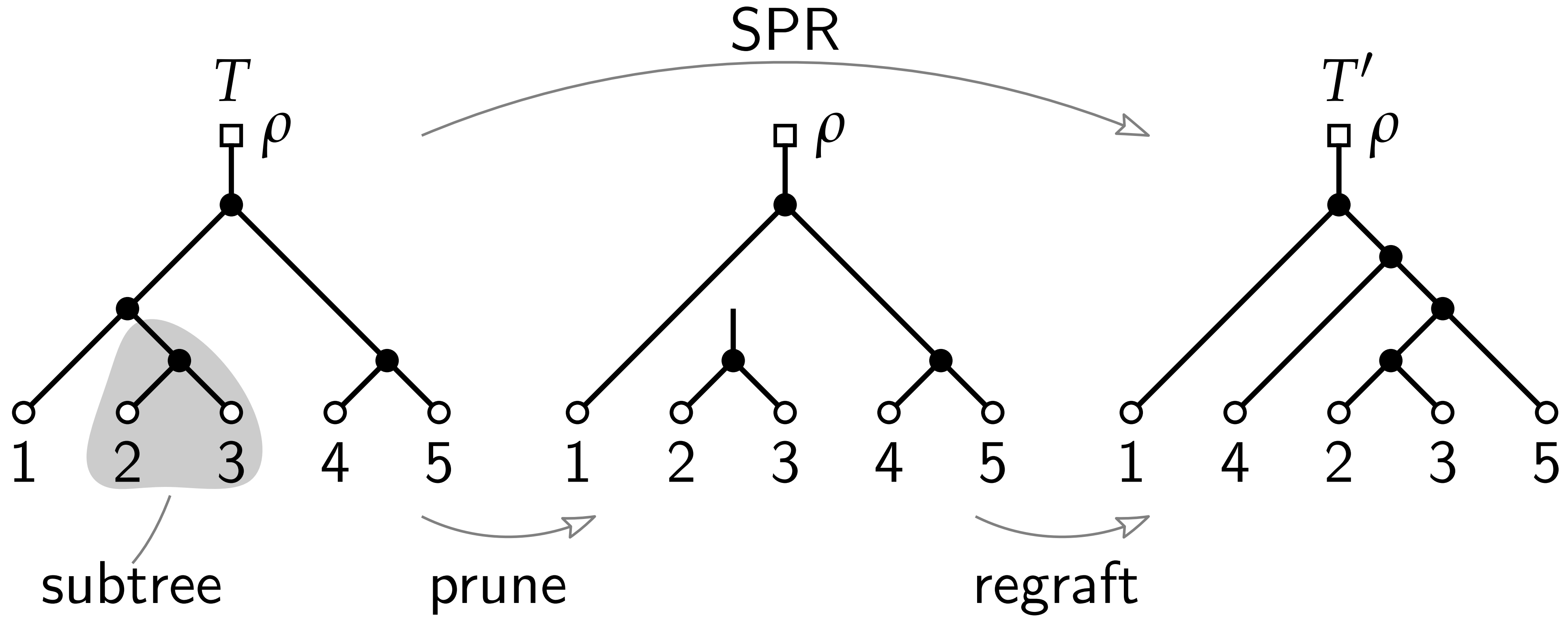
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.



- Note that an SPR operation is reversible.

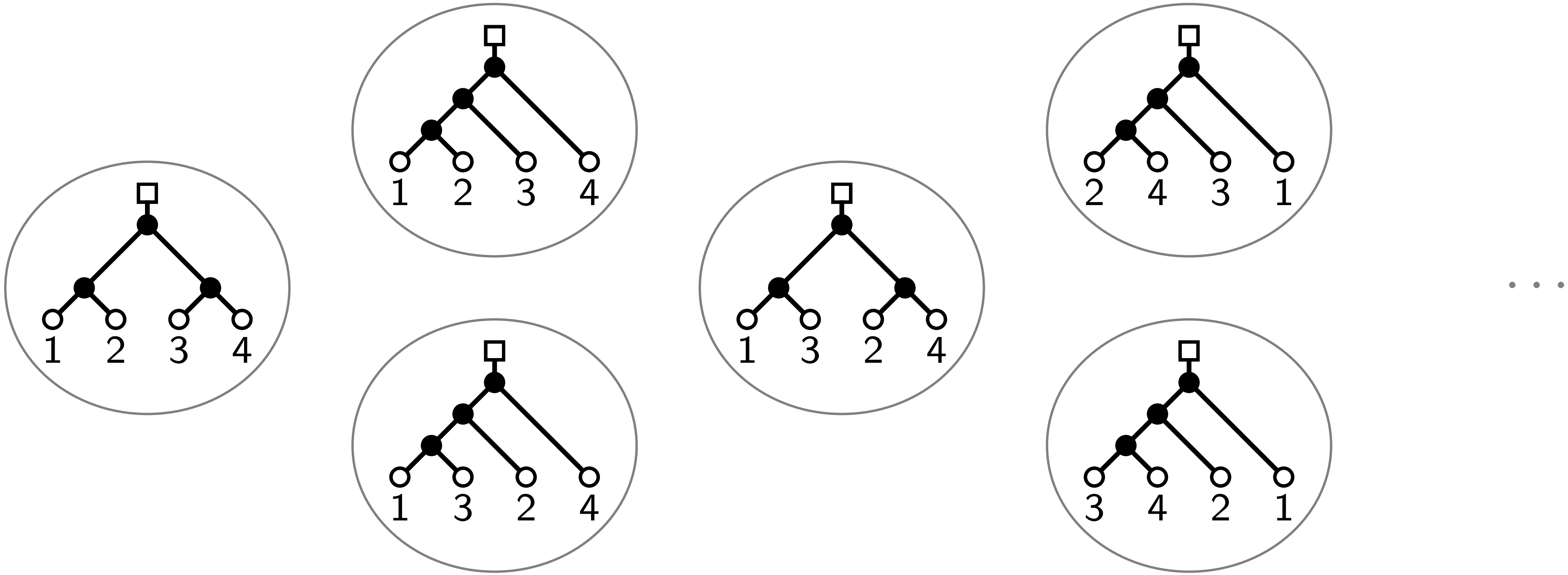
SPR-graph

SPR induces the **SPR-graph** $G = (V, E)$:

SPR-graph

SPR induces the **SPR-graph** $G = (V, E)$:

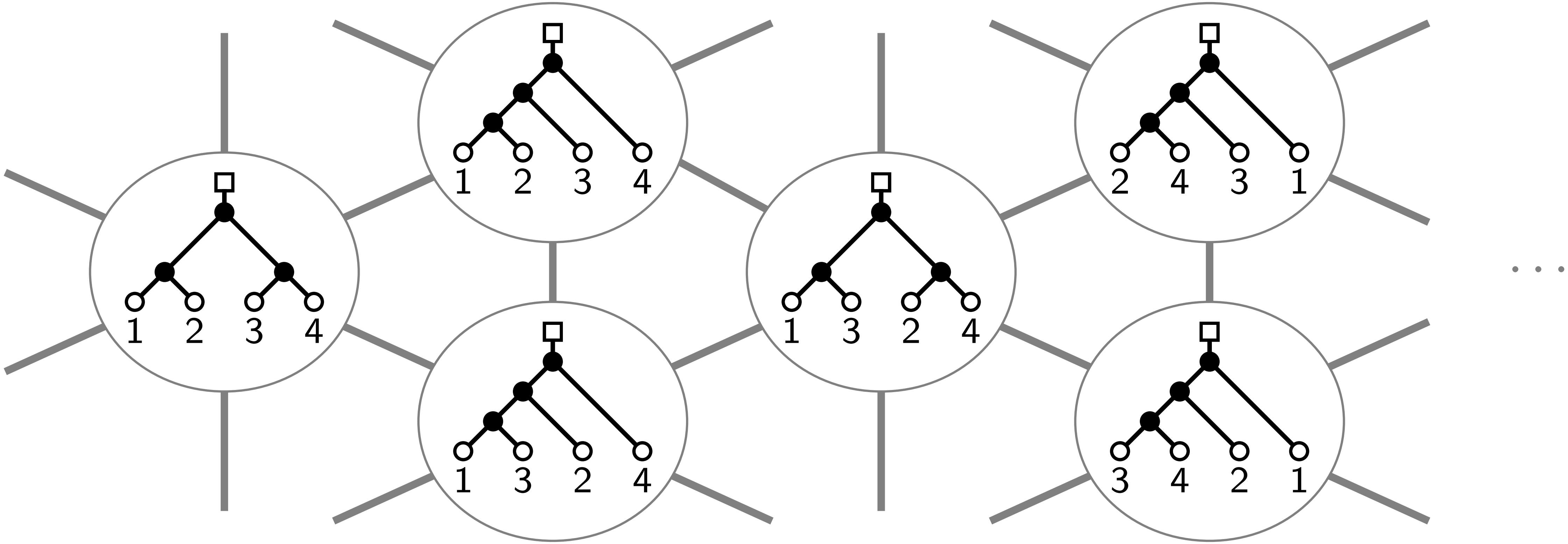
- $V = \{T \mid T$ is a phylogenetic tree on X$\}$



SPR-graph

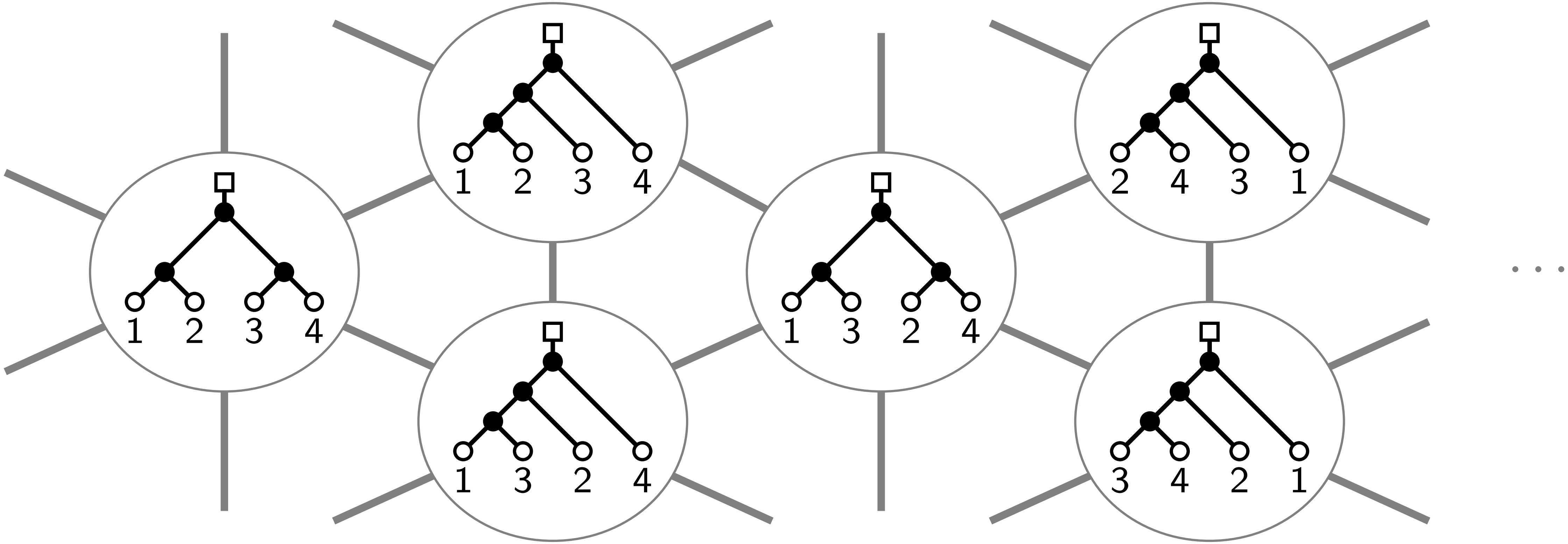
SPR induces the **SPR-graph** $G = (V, E)$:

- $V = \{T \mid T$ is a phylogenetic tree on X$\}$
- $\{T, T'\} \in E$ if T can be transformed into T' with a single SPR operation



SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.



SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

Proof as exercise or in discussion.

SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

Lemma 2.
The SPR-distance is a metric.

Proof as exercise or in discussion.

SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

Proof as exercise or in discussion.

Lemma 2.
The SPR-distance is a metric.

Proof. G is connected and undirected.

SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

Proof as exercise or in discussion.

Lemma 2.
The SPR-distance is a metric.

Proof. G is connected and undirected.

Goal.
Compute the SPR-distance $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

SPR-distance

The **SPR-distance** $\mathsf{d}_{\mathsf{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

Proof as exercise or in discussion.

Lemma 2.
The SPR-distance is a metric.

Proof. G is connected and undirected.

Goal.
Compute the SPR-distance $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

... but G is huge!

$$|V(G)| = (2n-3)!! = (2n-3) \cdot (2n-5) \cdot \ldots \cdot 5 \cdot 3$$

SPR-distance

The **SPR-distance** $\mathrm{d_{SPR}}(T, T')$ of T and T' is defined as the distance of T and T' in the SPR-graph G.

Lemma 1.
The SPR-graph G is connected.

Proof as exercise or in discussion.

Lemma 2.
The SPR-distance is a metric.

Proof. G is connected and undirected.

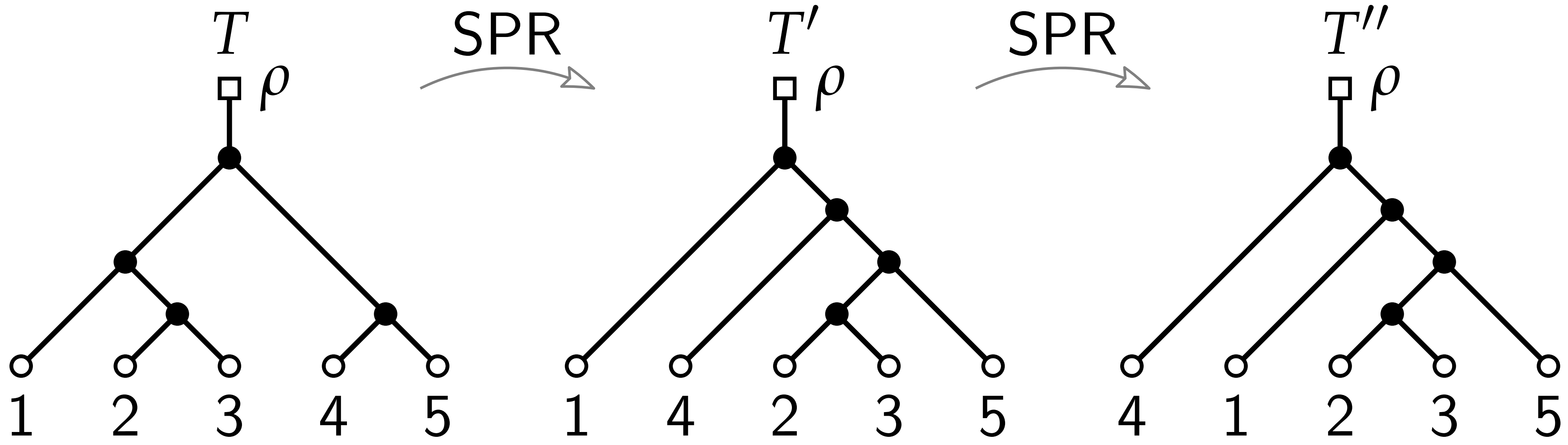
Goal.
Compute the SPR-distance $\mathrm{d_{SPR}}(T, T')$.

. . . but G is huge!

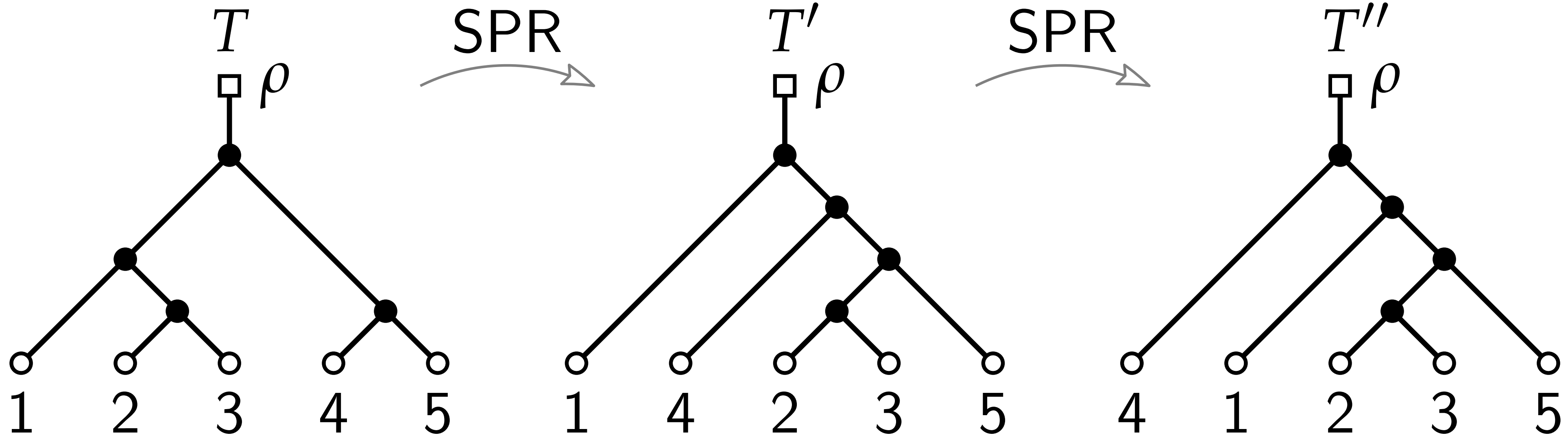
$$|V(G)| = (2n-3)!! = (2n-3)\cdot(2n-5)\cdot\ldots\cdot 5\cdot 3$$

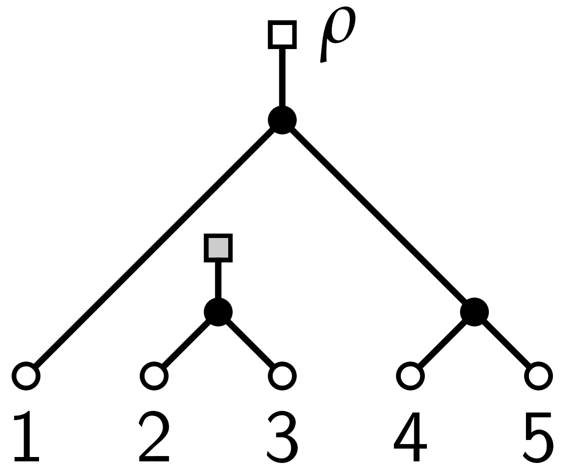
- Can we rephrase the problem?

Maximum agreement forests

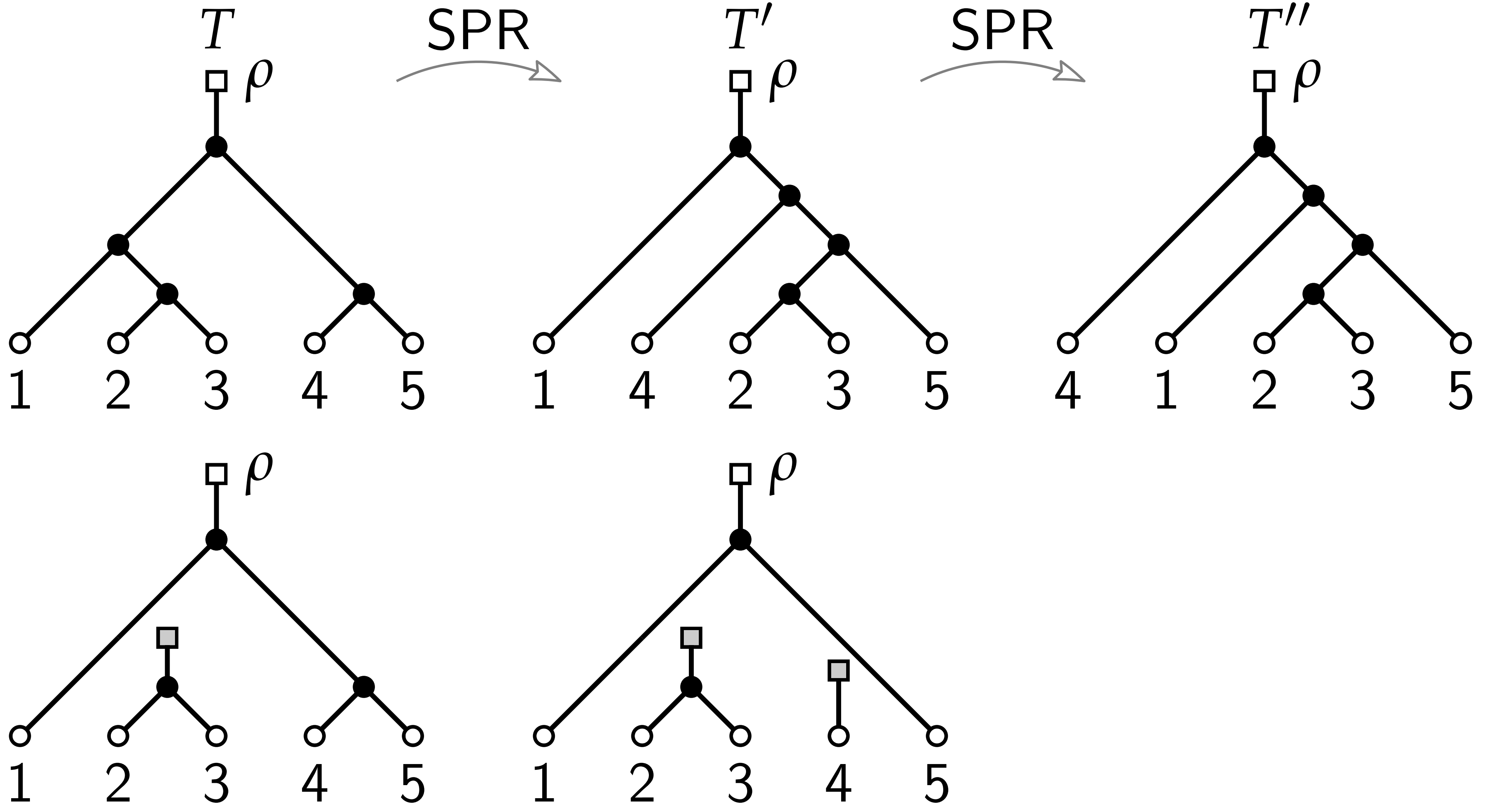


Maximum agreement forests

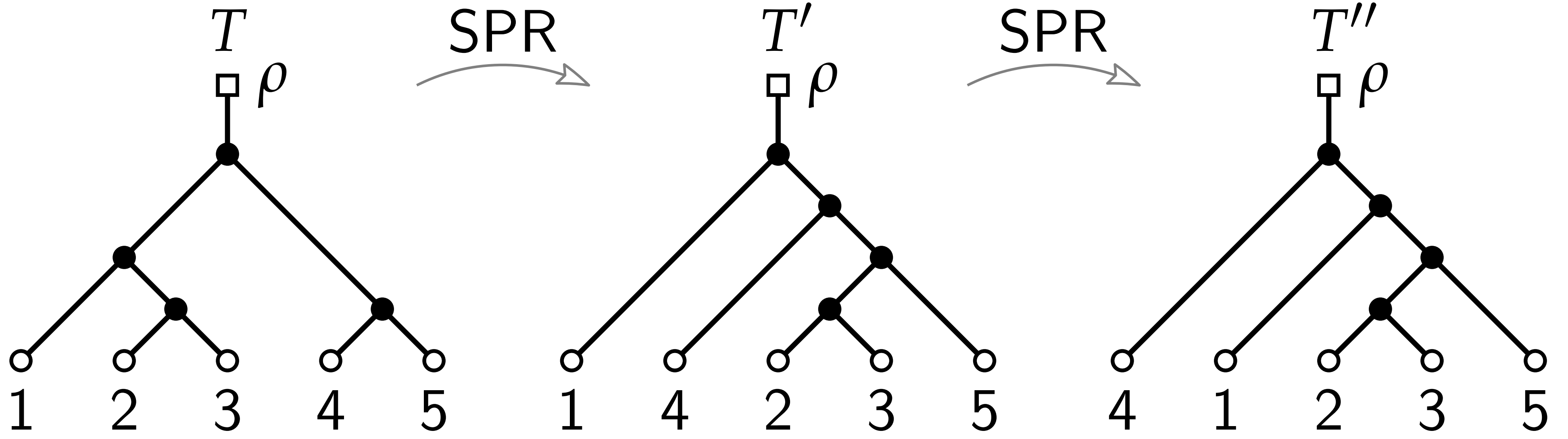


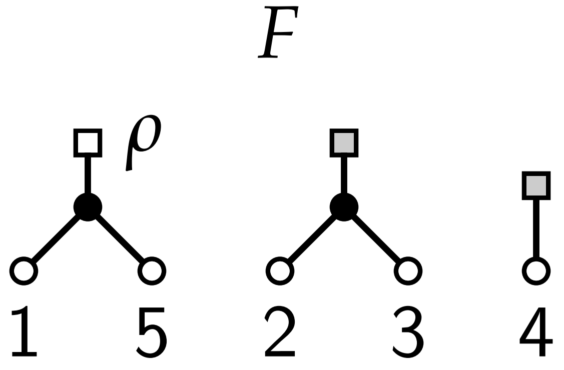


Maximum agreement forests

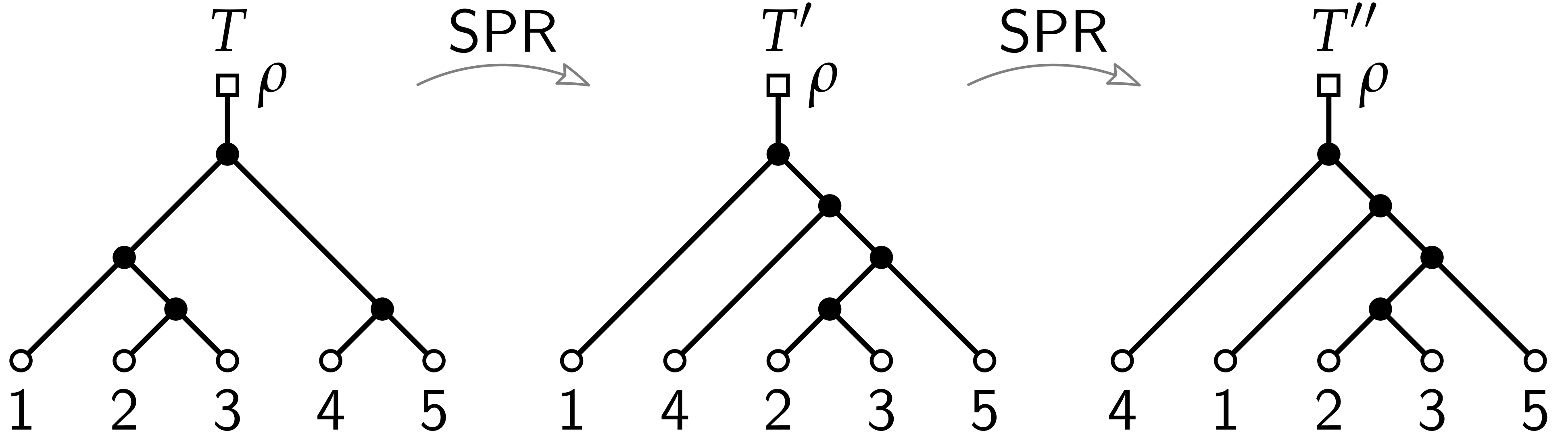


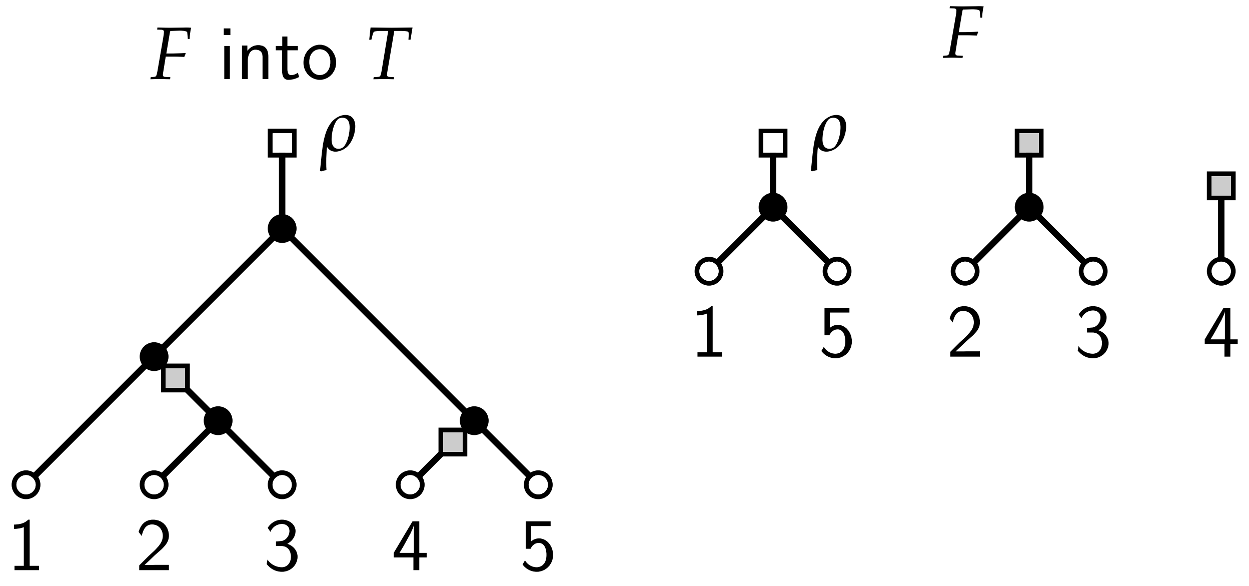
Maximum agreement forests



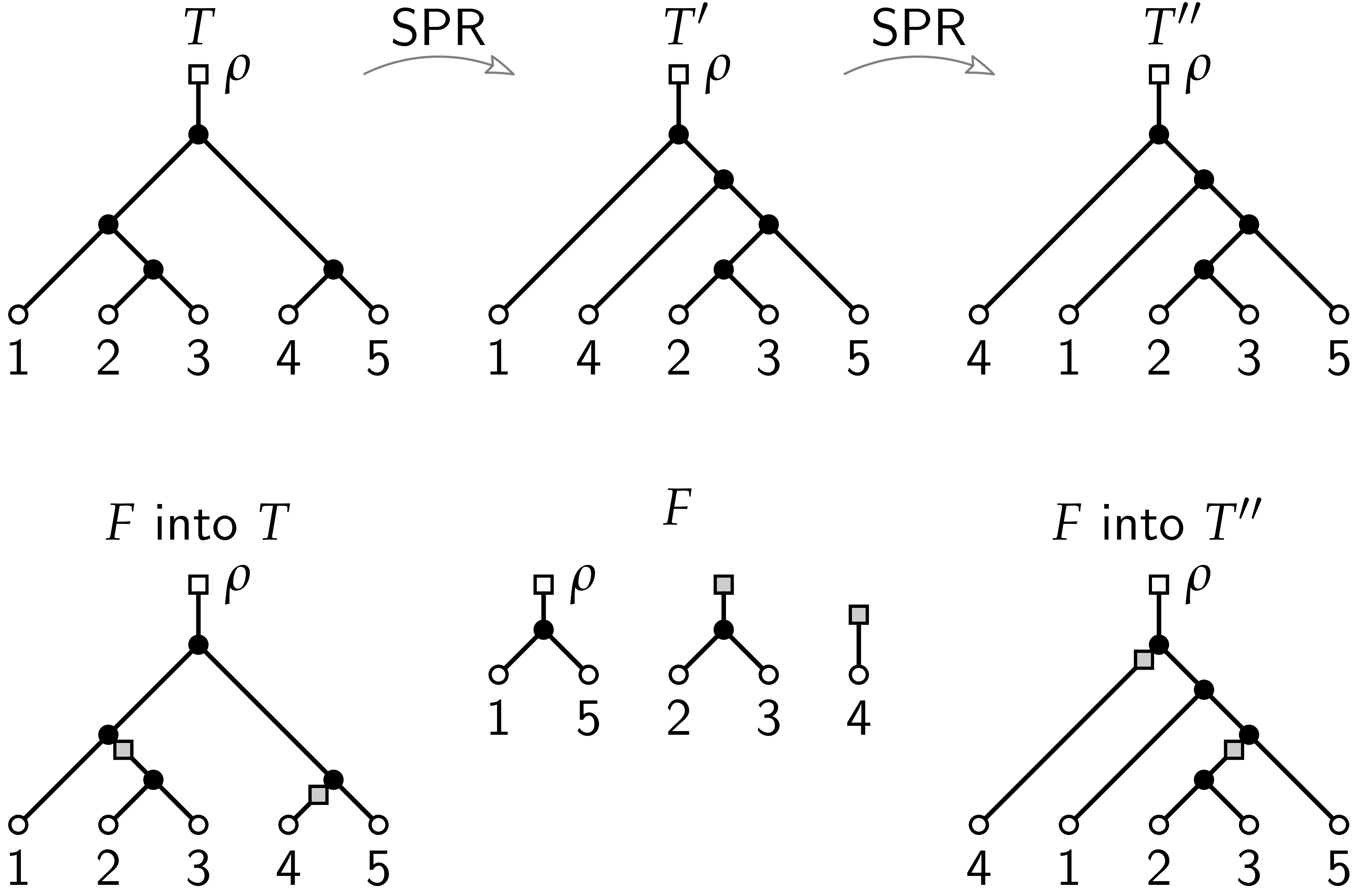


Maximum agreement forests

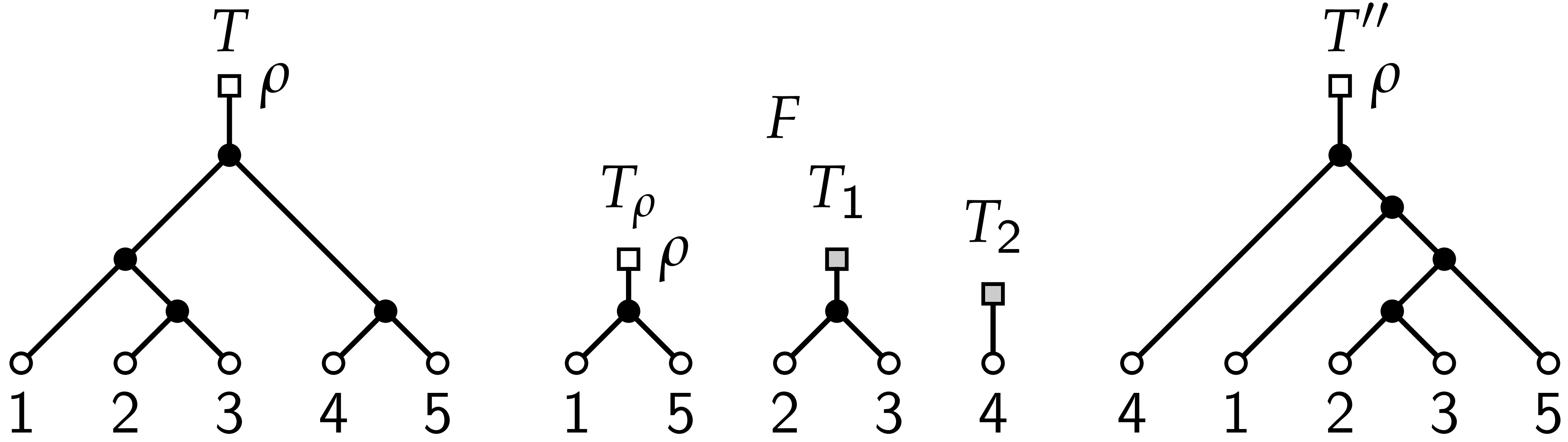




Maximum agreement forests



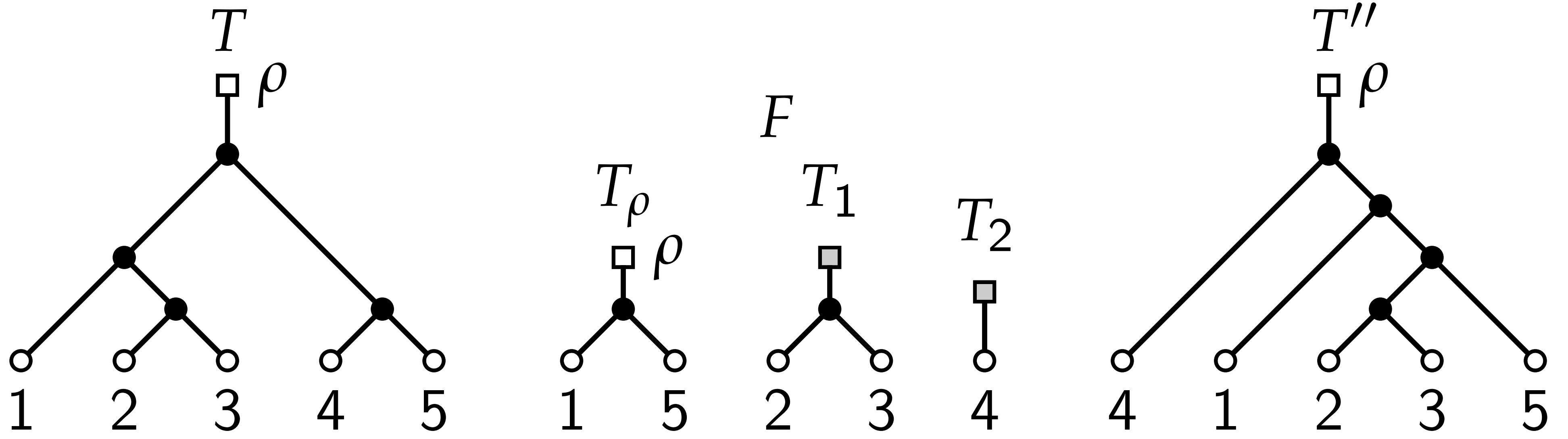
Maximum agreement forests



An **agreement forest** F of T and T'' is a forest $\{T_\rho, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the T_i partition $X \cup \{\rho\}$,

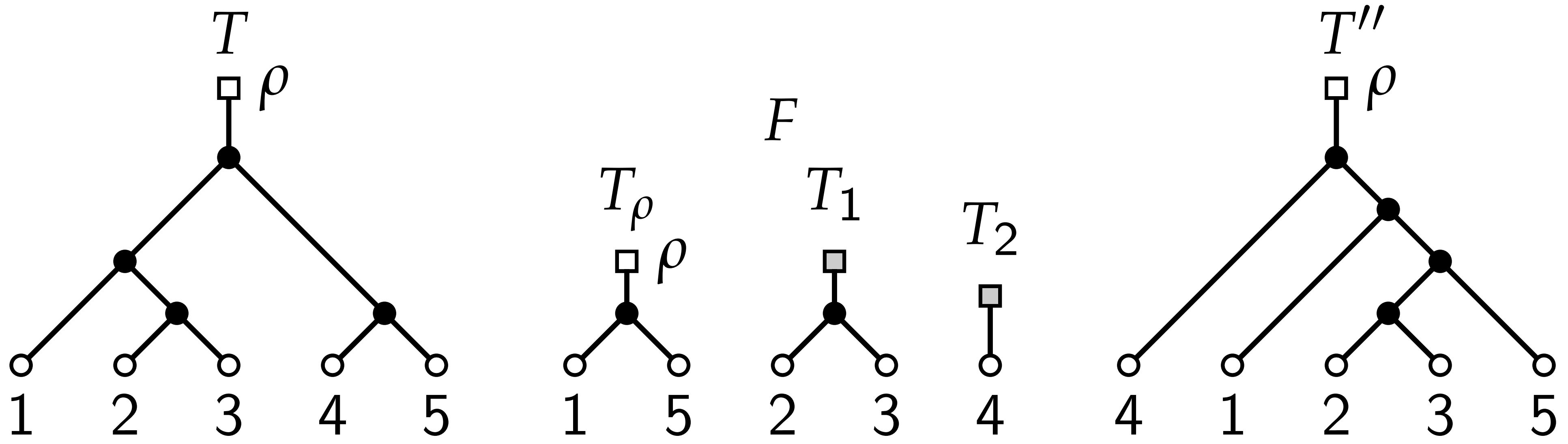
Maximum agreement forests



An **agreement forest** F of T and T'' is a forest $\{T_\rho, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the T_i partition $X \cup \{\rho\}$,
- ρ is in the label set of T_ρ, and

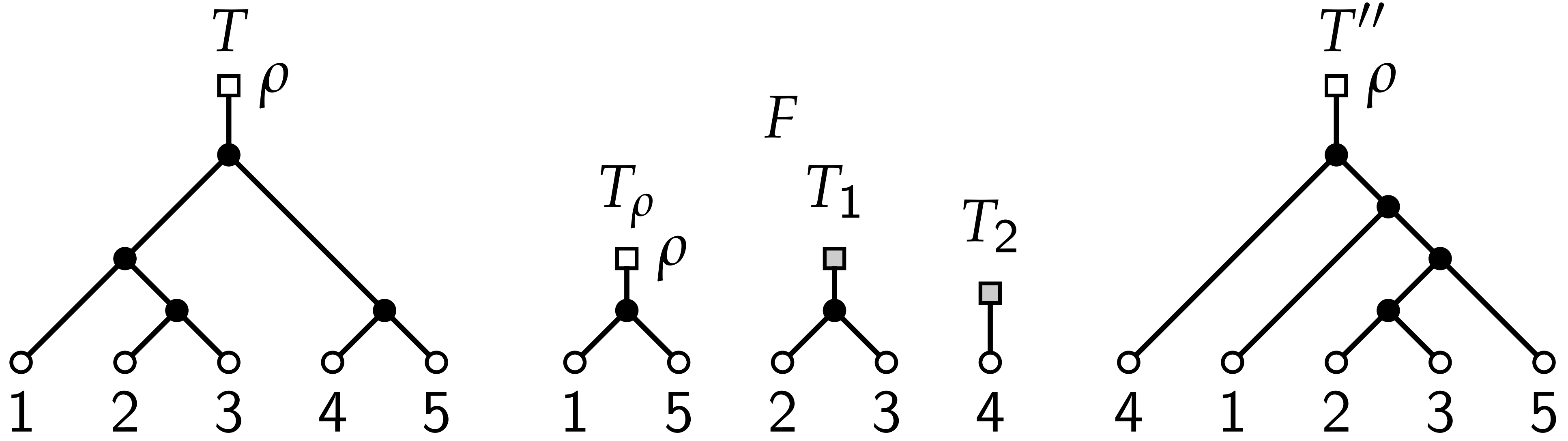
Maximum agreement forests



An **agreement forest** F of T and T'' is a forest $\{T_\rho, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the T_i partition $X \cup \{\rho\}$,
- ρ is in the label set of T_ρ, and
- there exist edge-disjoint embeddings of subdivisions of the T_i's into T and T'' that cover all edges.

Maximum agreement forests



An **agreement forest** F of T and T'' is a forest $\{T_\rho, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the T_i partition $X \cup \{\rho\}$,
- ρ is in the label set of T_ρ, and
- there exist edge-disjoint embeddings of subdivisions of the T_i's into T and T'' that cover all edges.

If k is minimal, F is a **maximum agreement forest (MAF)**.

Characterisation

Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Characterisation

Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathsf{SPR}}(T, T')$

Characterisation

Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathrm{SPR}}(T, T')$.

Characterisation

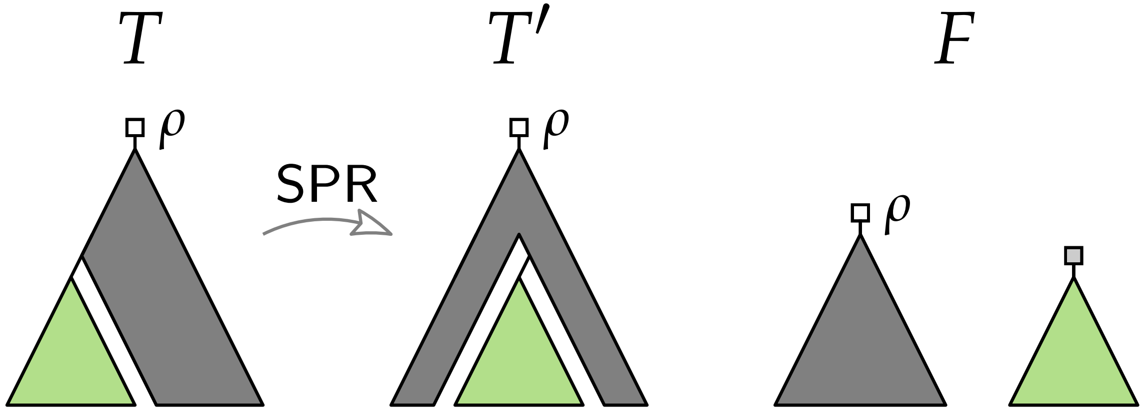
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathrm{SPR}}(T, T')$.

- Case $d = 1$ is easy. ✓



Characterisation

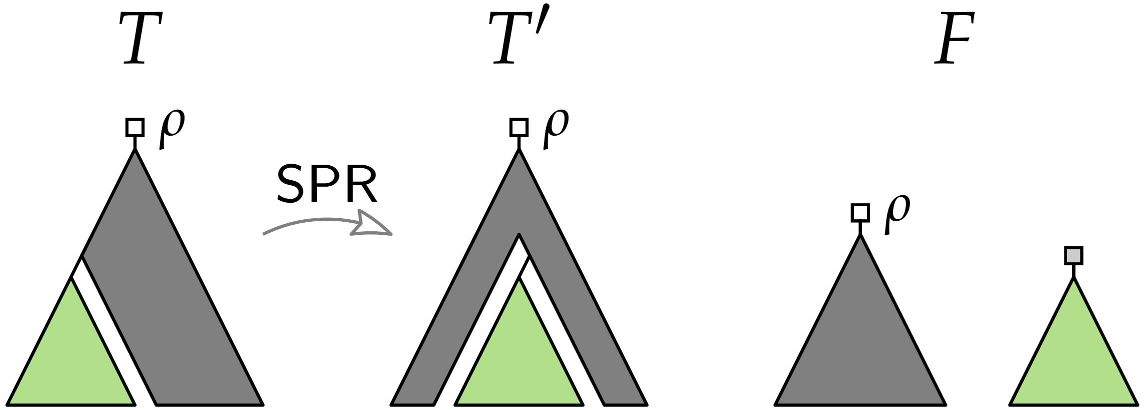
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathrm{SPR}}(T, T')$.

- Case $d = 1$ is easy. ✓
- Assume $\mathrm{m}(T, T') \leq \mathrm{d}_{\mathrm{SPR}}(T, T')$ holds for all $d \leq \ell$.



Characterisation

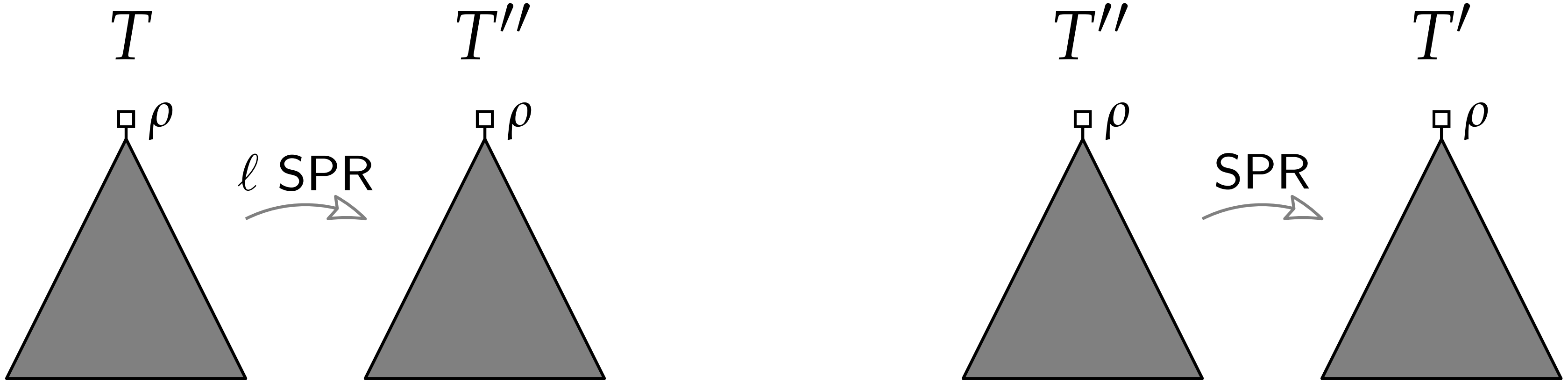
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathrm{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists T'' with $\mathrm{d}_{\mathrm{SPR}}(T, T'') = \ell$ and $\mathrm{d}_{\mathrm{SPR}}(T'', T') = 1$.



Characterisation

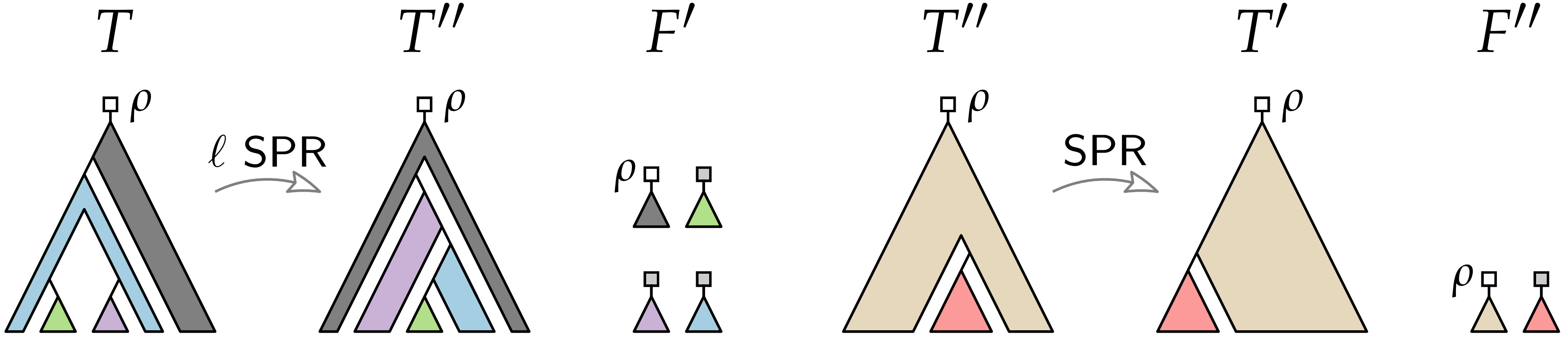
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d_{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d_{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists T'' with $\mathrm{d_{SPR}}(T, T'') = \ell$ and $\mathrm{d_{SPR}}(T'', T') = 1$.
- There exists MAF F' for T and T'' and MAF F'' for T'' and T'.



Characterisation

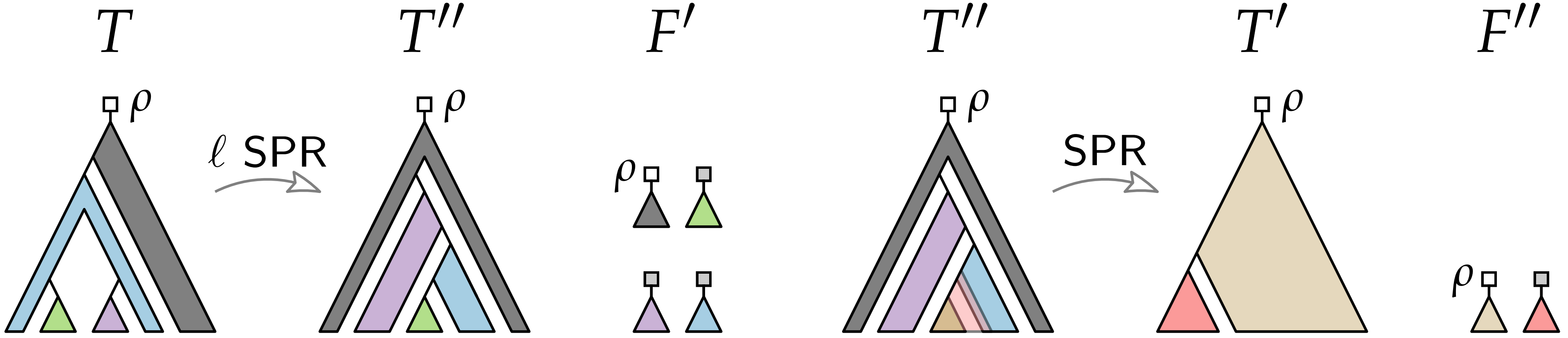
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d_{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d_{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists T'' with $\mathrm{d_{SPR}}(T, T'') = \ell$ and $\mathrm{d_{SPR}}(T'', T') = 1$.
- There exists MAF F' for T and T'' and MAF F'' for T'' and T'.



Characterisation

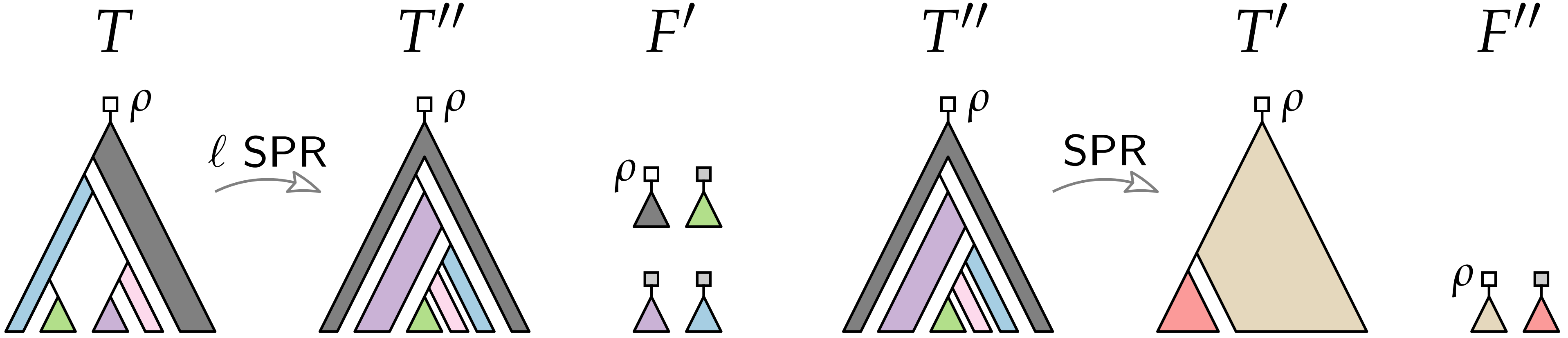
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathrm{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists T'' with $\mathrm{d}_{\mathrm{SPR}}(T, T'') = \ell$ and $\mathrm{d}_{\mathrm{SPR}}(T'', T') = 1$.
- There exists MAF F' for T and T'' and MAF F'' for T'' and T'.



Characterisation

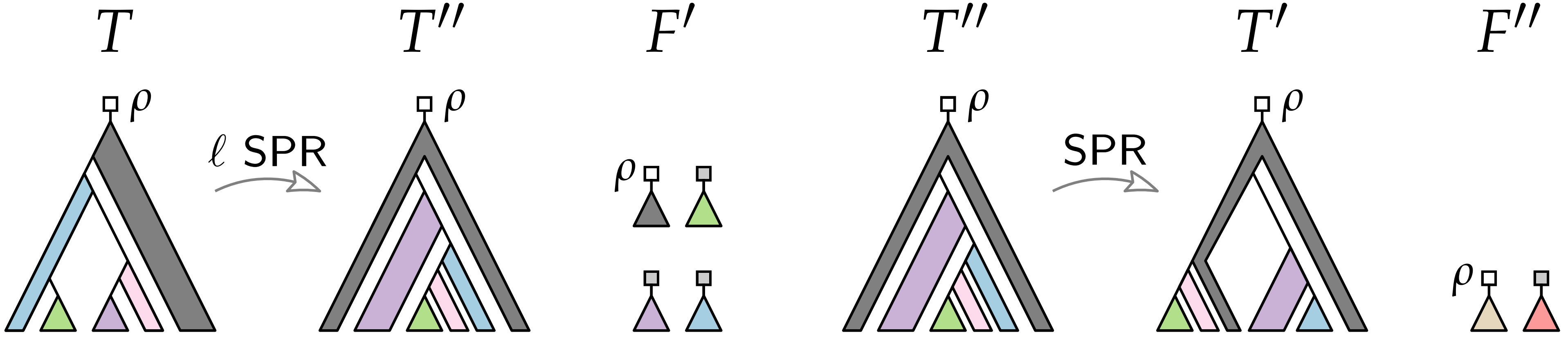
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathsf{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathsf{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists T'' with $\mathrm{d}_{\mathsf{SPR}}(T, T'') = \ell$ and $\mathrm{d}_{\mathsf{SPR}}(T'', T') = 1$.
- There exists MAF F' for T and T'' and MAF F'' for T'' and T'.



Characterisation

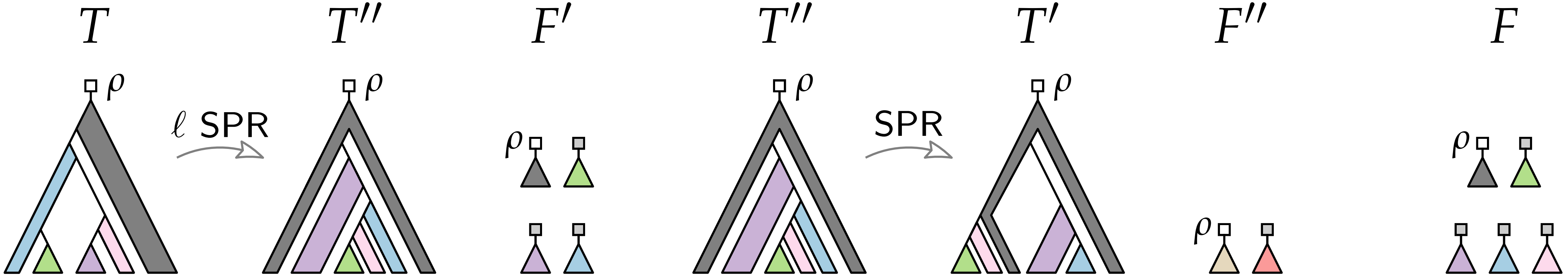
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathsf{SPR}}(T, T')$

Proof of "$\leq$" by induction on $d = \mathrm{d}_{\mathsf{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists T'' with $\mathrm{d}_{\mathsf{SPR}}(T, T'') = \ell$ and $\mathrm{d}_{\mathsf{SPR}}(T'', T') = 1$.
- There exists MAF F' for T and T'' and MAF F'' for T'' and T'.



Characterisation

Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathsf{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

Characterisation

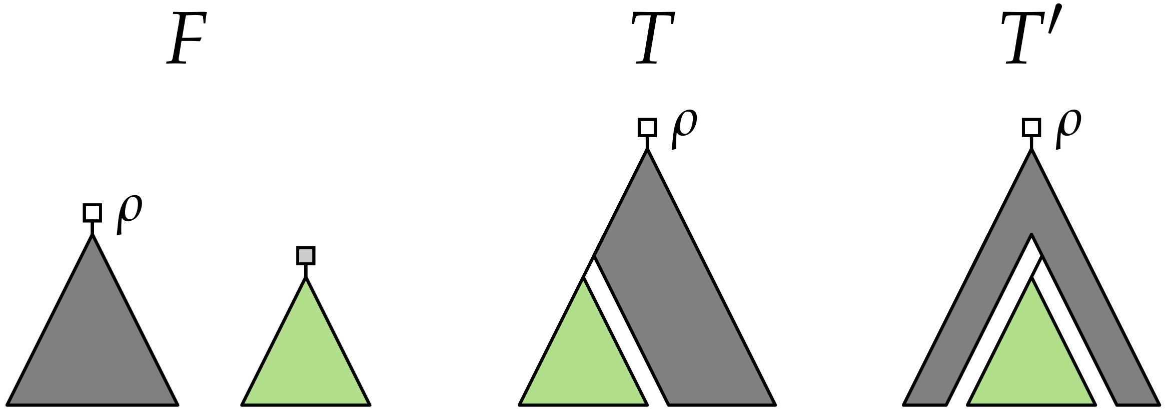
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Case $d = 1$ is easy. ✓



Characterisation

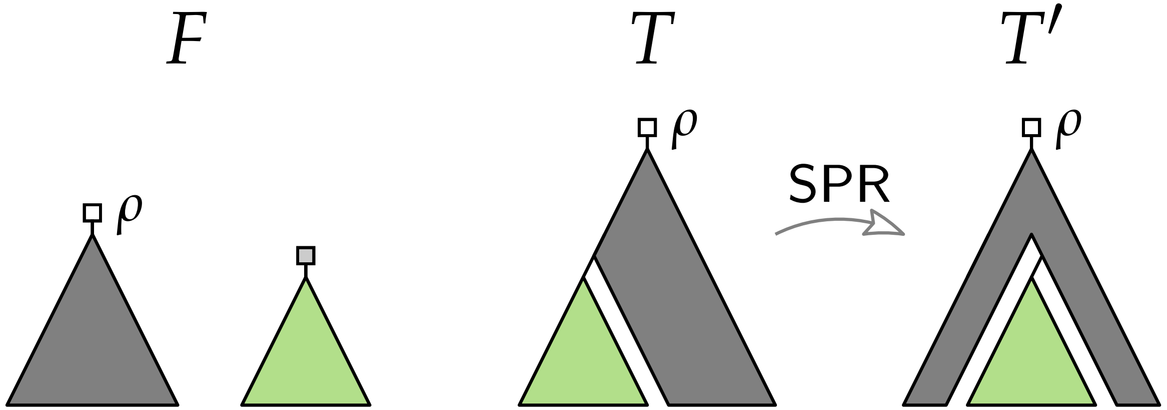
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Case $d = 1$ is easy. ✓



Characterisation

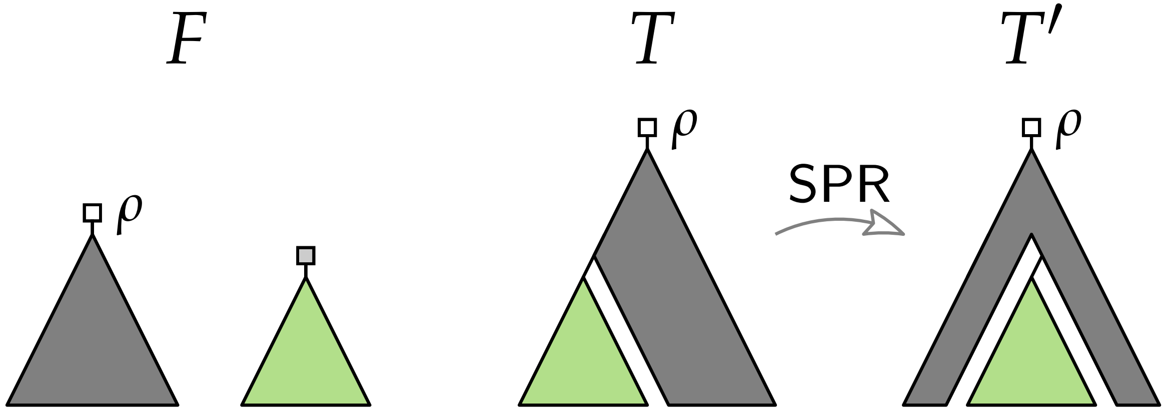
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d_{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Case $d = 1$ is easy. ✓
- Assume $\mathrm{m}(T, T') \geq \mathrm{d_{SPR}}(T, T')$ holds for all $d \leq \ell$.



Characterisation

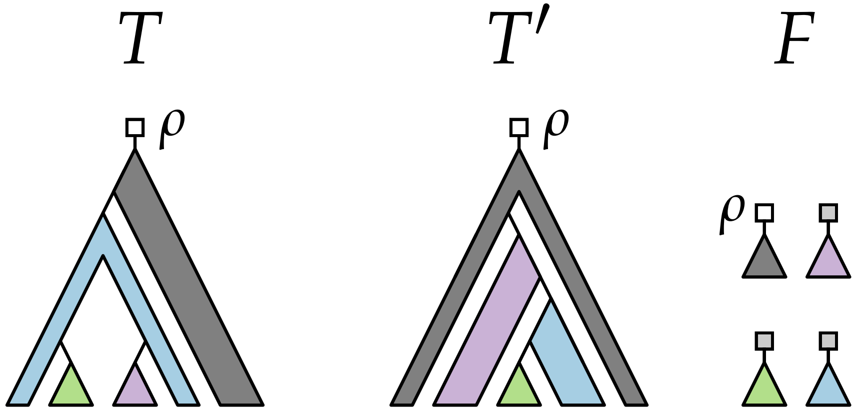
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Let F be a MAF of T and T' of size $\ell + 2$. $\Rightarrow m = \ell + 1$



Characterisation

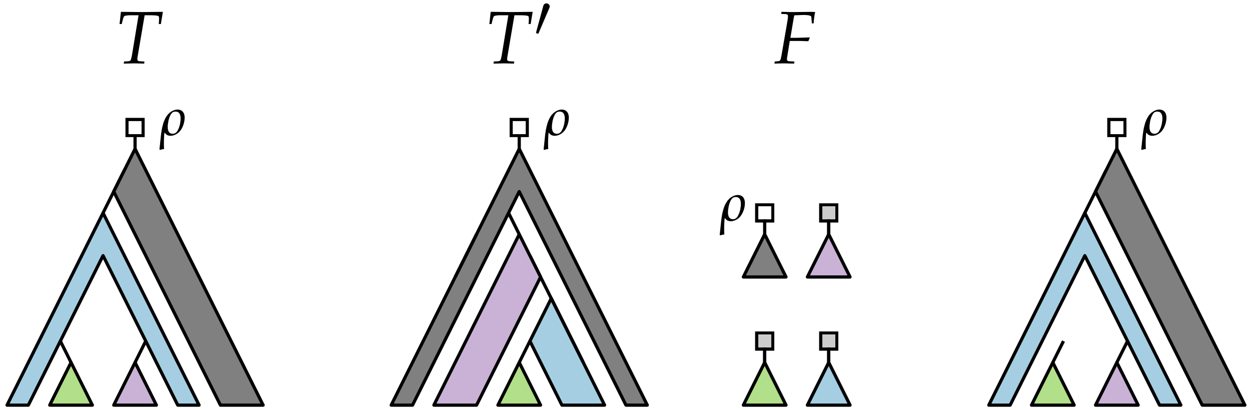
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Let F be a MAF of T and T' of size $\ell + 2$. $\Rightarrow m = \ell + 1$
- There exists a T_i that can be pruned in T.



Characterisation

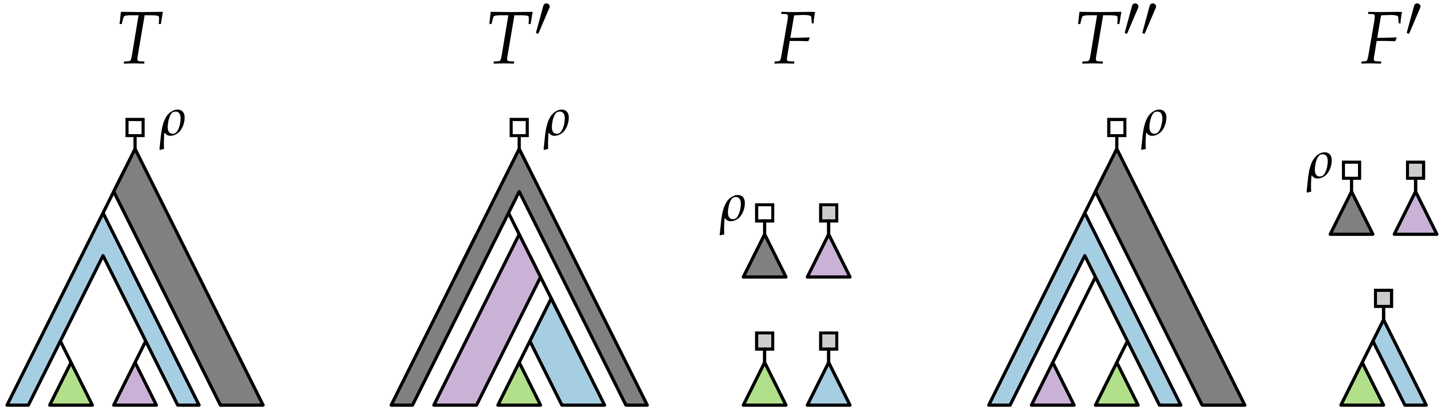
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Let F be a MAF of T and T' of size $\ell + 2$.
- There exists a T_i that can be pruned in T.
- Regraft T_i according to the embedding of F into $T' \Rightarrow T''$ & F'



Characterisation

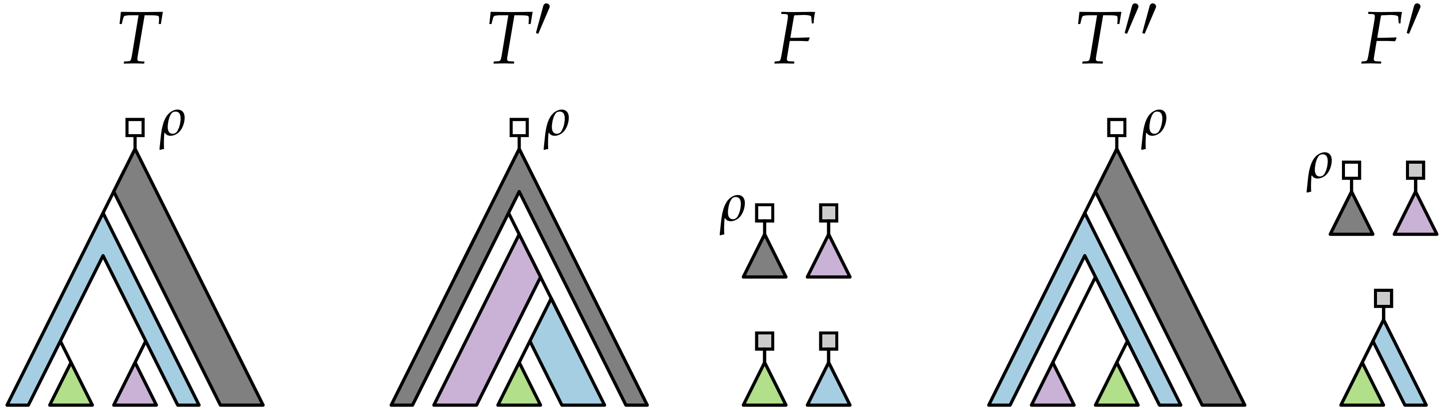
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d_{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Let F be a MAF of T and T' of size $\ell + 2$.
- There exists a T_i that can be pruned in T.



- Regraft T_i according to the embedding of F into T' $\Rightarrow$ T'' & F'
- F' is an AF for T' and T''
- $\Rightarrow \mathrm{d_{SPR}}(T'', T') \leq \ell$

Characterisation

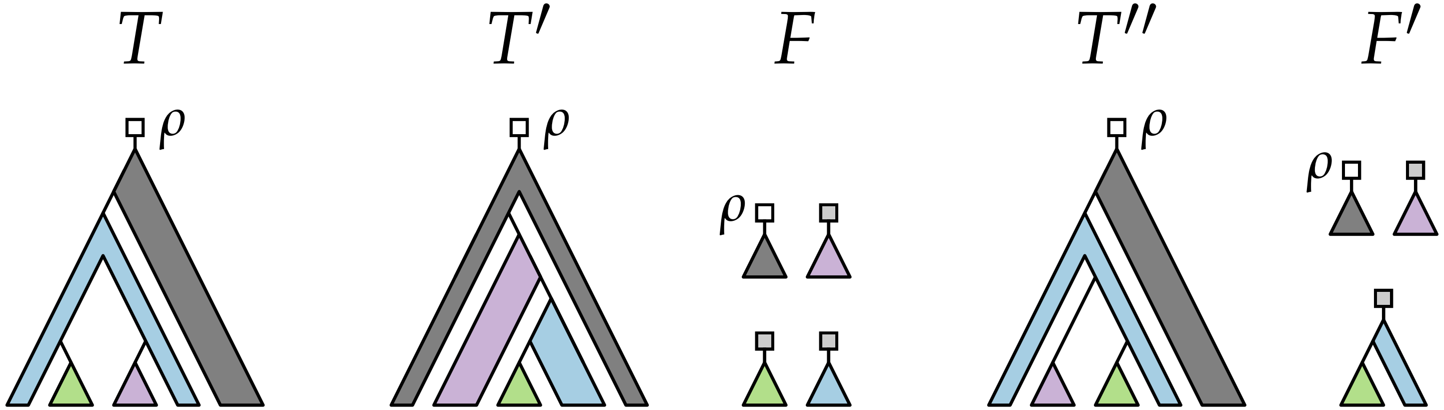
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d_{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Let F be a MAF of T and T' of size $\ell + 2$.
- There exists a T_i that can be pruned in T.



- Regraft T_i according to the embedding of F into $T' \Rightarrow T''$ & F'
- F' is an AF for T' and T''
- $\Rightarrow \mathrm{d_{SPR}}(T'', T') \leq \ell$
- $\mathrm{d_{SPR}}(T, T'') = 1$

Characterisation

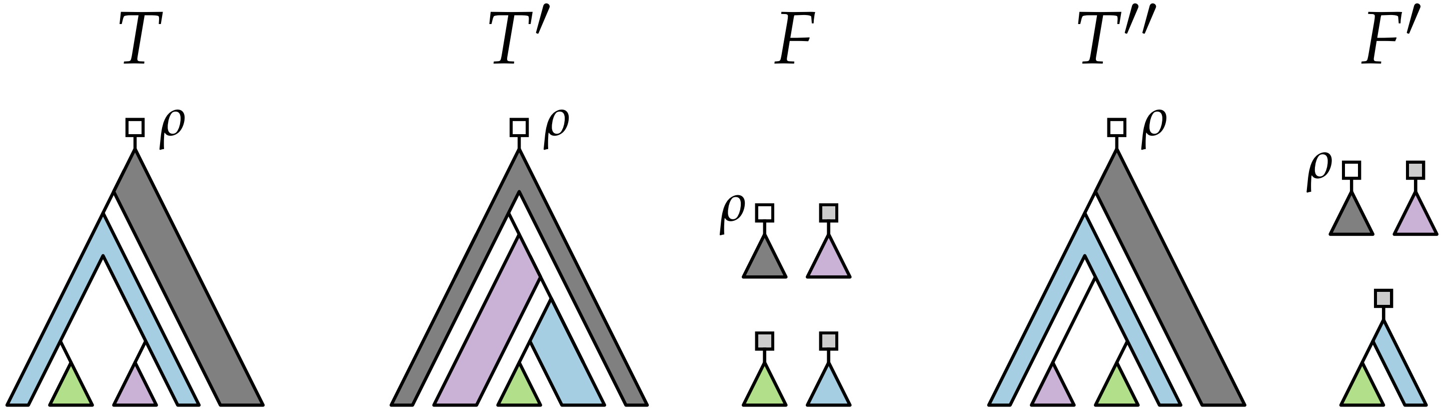
Let T and T' be two phylogenetic trees on X.
Let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of T and T'.
Define

$$\mathrm{m}(T, T') = k = |F| - 1.$$

Theorem 3. $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

Proof of "$\geq$" by induction on $d = \mathrm{m}(T, T')$.

- Let F be a MAF of T and T' of size $\ell + 2$.
- There exists a T_i that can be pruned in T.



- Regraft T_i according to the embedding of F into T' $\Rightarrow$ T'' & F'
- F' is an AF for T' and T''
- $\Rightarrow \mathrm{d}_{\mathrm{SPR}}(T'', T') \leq \ell$
- $\mathrm{d}_{\mathrm{SPR}}(T, T'') = 1$
- $\mathrm{d}_{\mathrm{SPR}}(T, T') \leq \ell + 1 = \mathrm{m}(T, T')$

Problem & Plan

Theorem 4. [HJWZ '96, BS '05]
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is NP-hard.

Proof is by reduction from Exact Cover by 3-Sets.

Problem & Plan

Theorem 4. [HJWZ '96, BS '05]
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is NP-hard.

Proof is by reduction from Exact Cover by 3-Sets.

Plan.

- Construct **kernel** of the problem.
  - Replace T and T' with smaller S and S'.
  - We should be able to get $\mathsf{d}_{\mathsf{SPR}}(T, T')$ from $\mathsf{d}_{\mathsf{SPR}}(S, S')$.

Problem & Plan

Theorem 4. [HJWZ '96, BS '05]
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is NP-hard.

Proof is by reduction from Exact Cover by 3-Sets.

Plan.

- Construct **kernel** of the problem.
  - Replace T and T' with smaller S and S'.
  - We should be able to get $\mathsf{d}_{\mathsf{SPR}}(T, T')$ from $\mathsf{d}_{\mathsf{SPR}}(S, S')$.
- Show that size of the kernel depends on $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Problem & Plan

Theorem 4. [HJWZ '96, BS '05]
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is NP-hard.

Proof is by reduction from Exact Cover by 3-Sets.

Plan.

- Construct **kernel** of the problem.
  - Replace T and T' with smaller S and S'.
  - We should be able to get $\mathsf{d}_{\mathsf{SPR}}(T, T')$ from $\mathsf{d}_{\mathsf{SPR}}(S, S')$.
- Show that size of the kernel depends on $\mathsf{d}_{\mathsf{SPR}}(T, T')$.
- Devise an fpt algorithm by computing $\mathsf{d}_{\mathsf{SPR}}$ for kernel.

Problem & Plan

Theorem 4. [HJWZ '96, BS '05]
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is NP-hard.

Proof is by reduction from Exact Cover by 3-Sets.

Plan.

- Construct **kernel** of the problem.
  - Replace T and T' with smaller S and S'.
  - We should be able to get $\mathsf{d}_{\mathsf{SPR}}(T, T')$ from $\mathsf{d}_{\mathsf{SPR}}(S, S')$.
- Show that size of the kernel depends on $\mathsf{d}_{\mathsf{SPR}}(T, T')$.
- Devise an fpt algorithm by computing $\mathsf{d}_{\mathsf{SPR}}$ for kernel.
- Devise an approximation algorithm.

Kernelisation – Subtrees

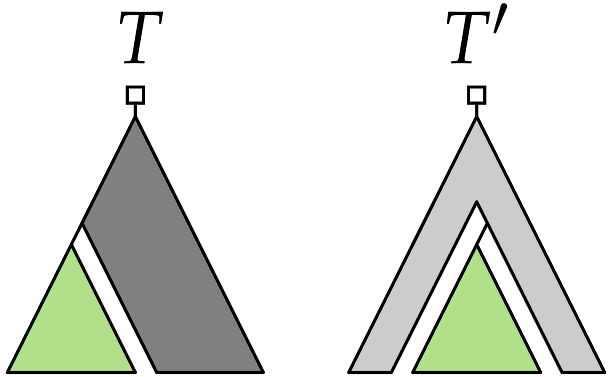
Common subtree reduction.

- Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.

Kernelisation – Subtrees

Common subtree reduction.

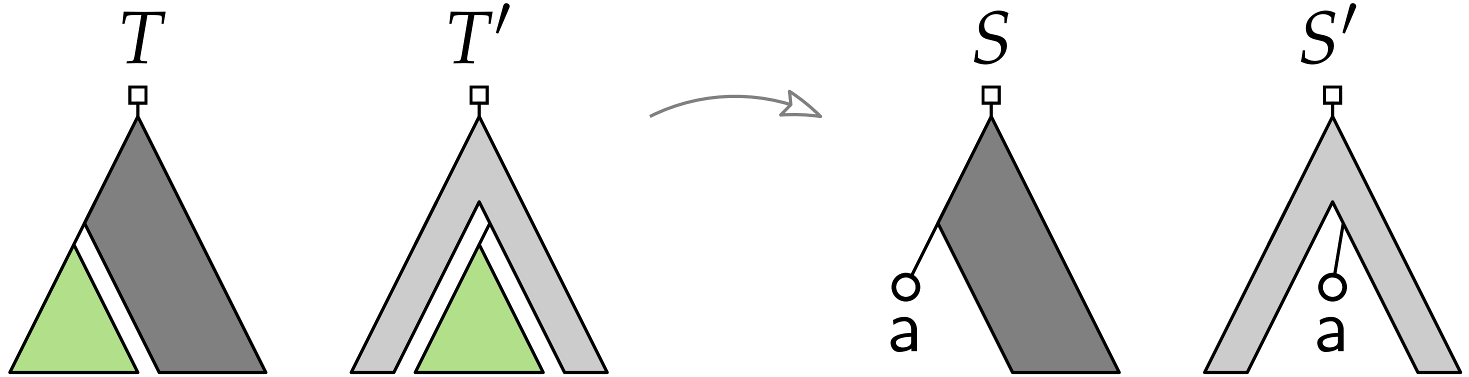
- Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.



Kernelisation – Subtrees

Common subtree reduction.

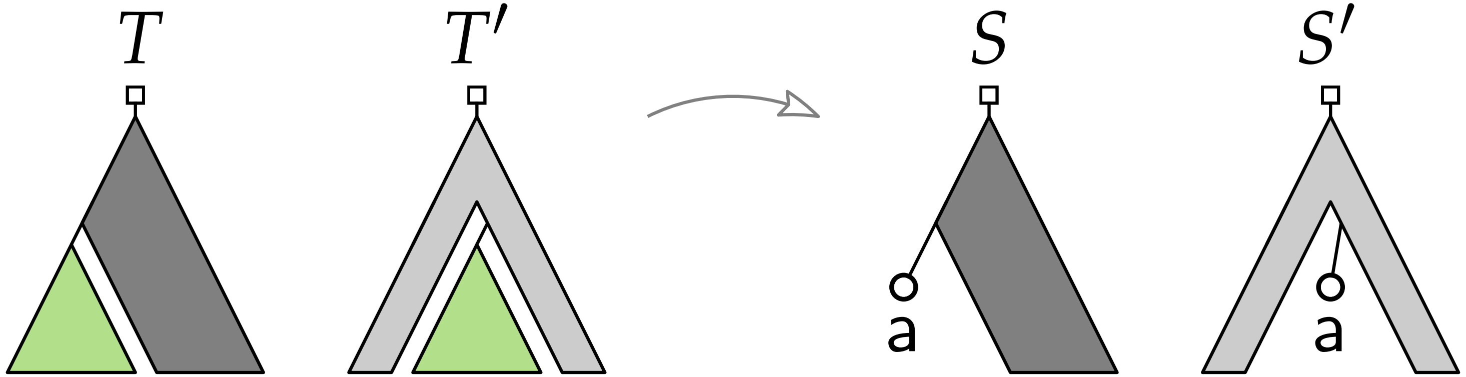
- Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.



Kernelisation – Subtrees

Common subtree reduction.

- Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.

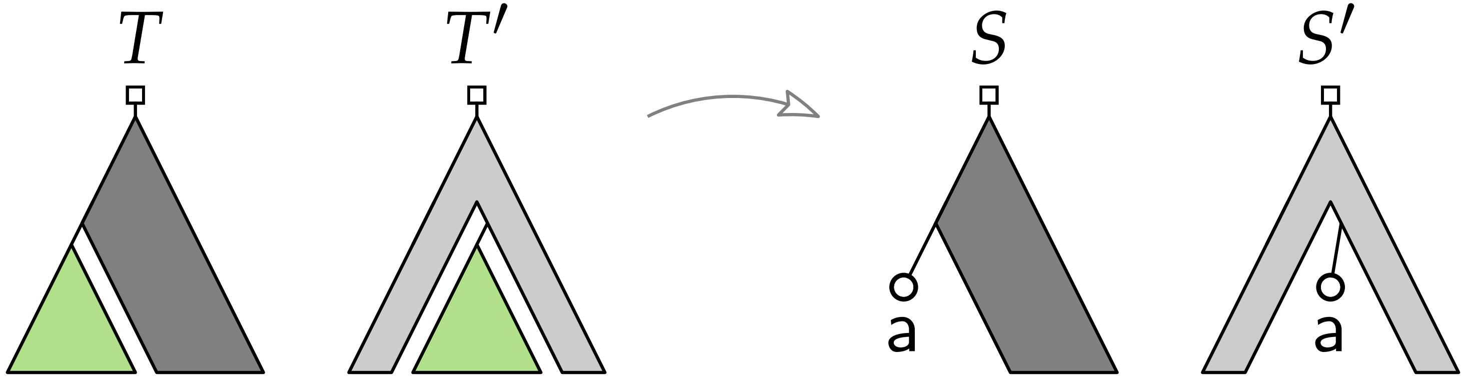


Lemma 5. Applying the common subtree reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

Kernelisation – Subtrees

Common subtree reduction.

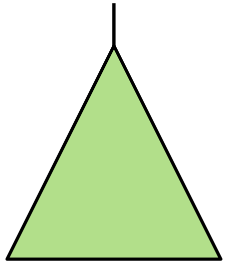
- Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.



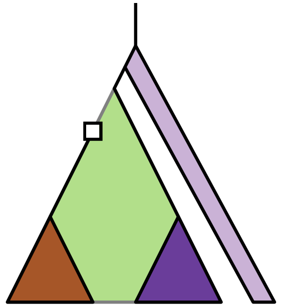
Lemma 5. Applying the common subtree reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

Proof.

Suppose



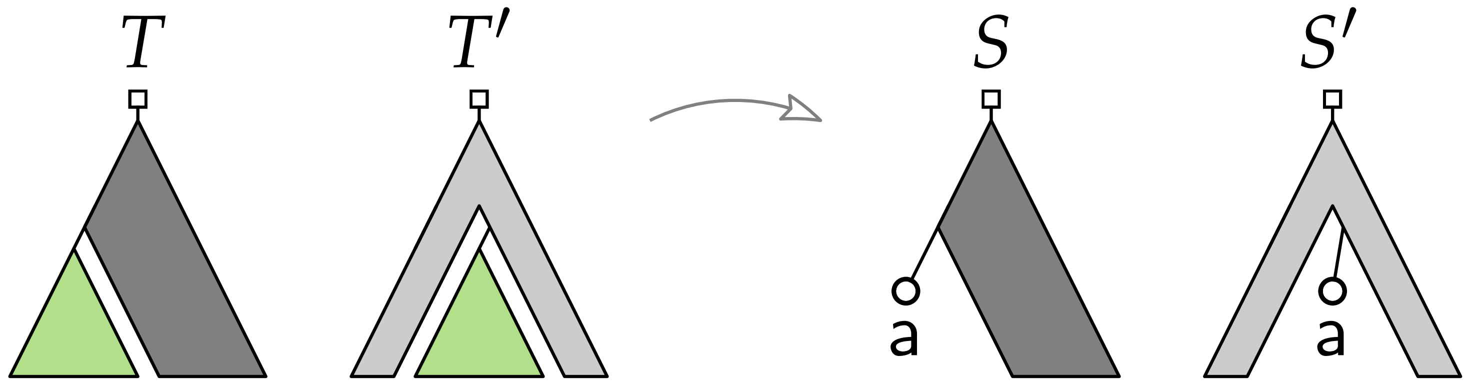
is covered by two trees of MAF



Kernelisation – Subtrees

Common subtree reduction.

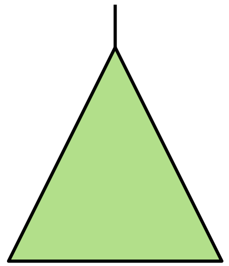
- Replace any pendant subtree that occurs identically in both trees by a single leaf with a new label.



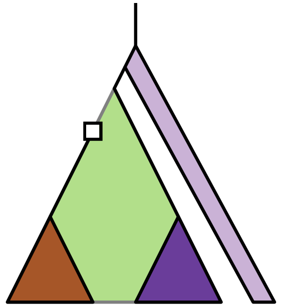
Lemma 5. Applying the common subtree reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

Proof.

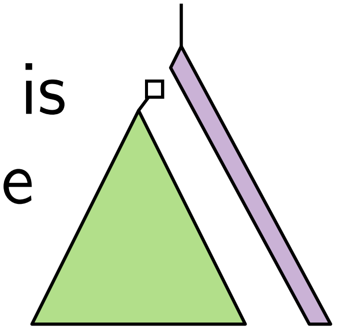
Suppose



is covered by two trees of MAF



then there is alternative MAF



Kernelisation – Chains

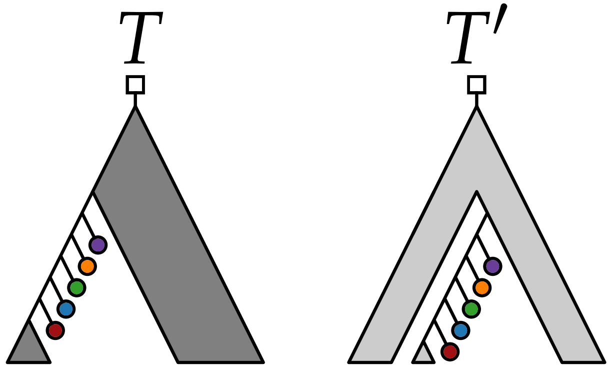
Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

Kernelisation – Chains

Chain reduction.

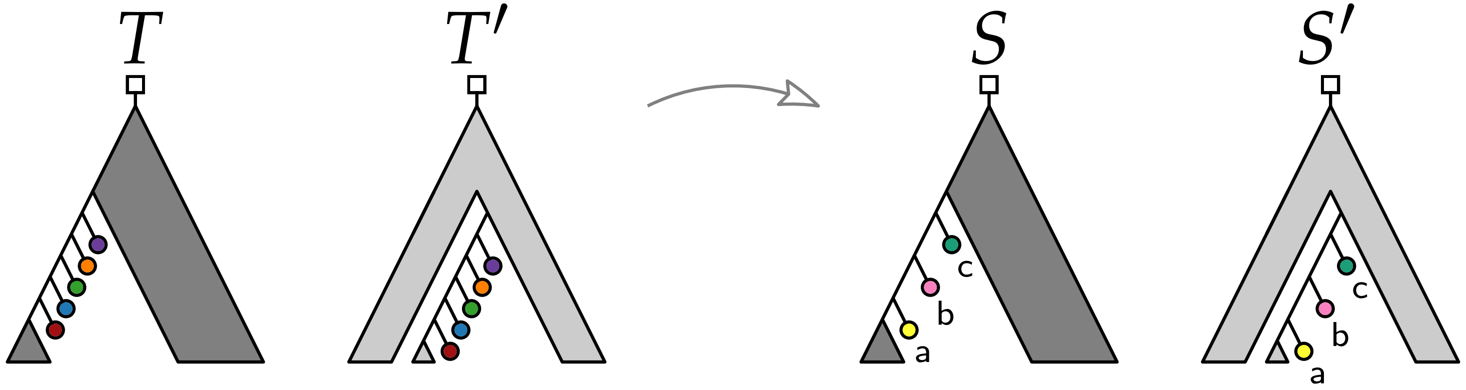
- Replace any chain of leaves that occurs identically in both trees by three new leaves.



Kernelisation – Chains

Chain reduction.

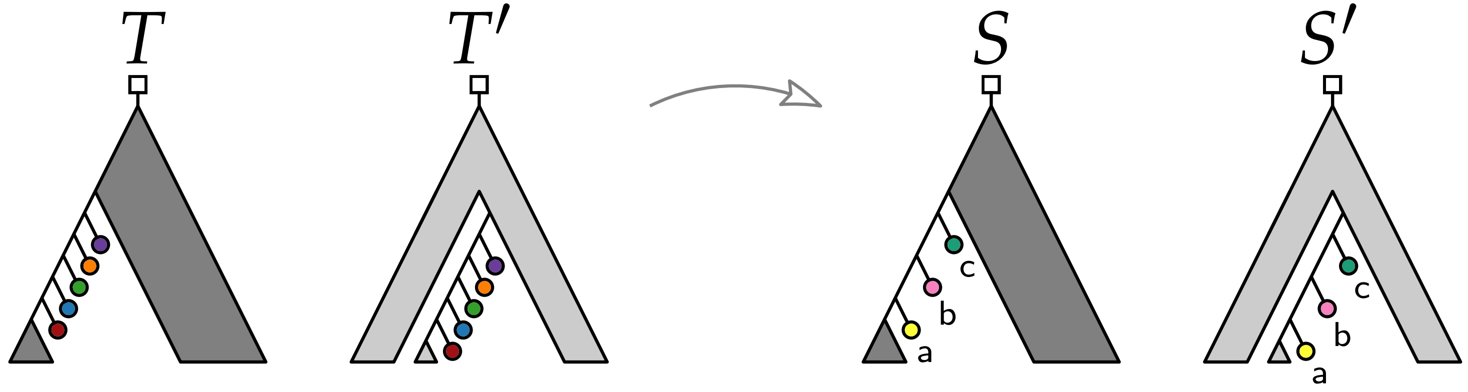
- Replace any chain of leaves that occurs identically in both trees by three new leaves.



Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

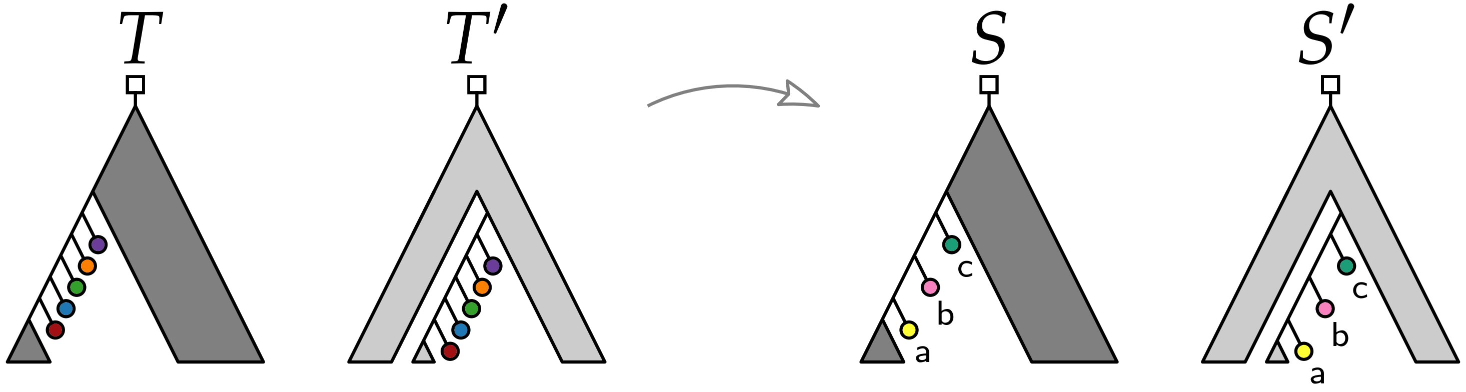


Lemma 6. Applying chain reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.



Lemma 6. Applying chain reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

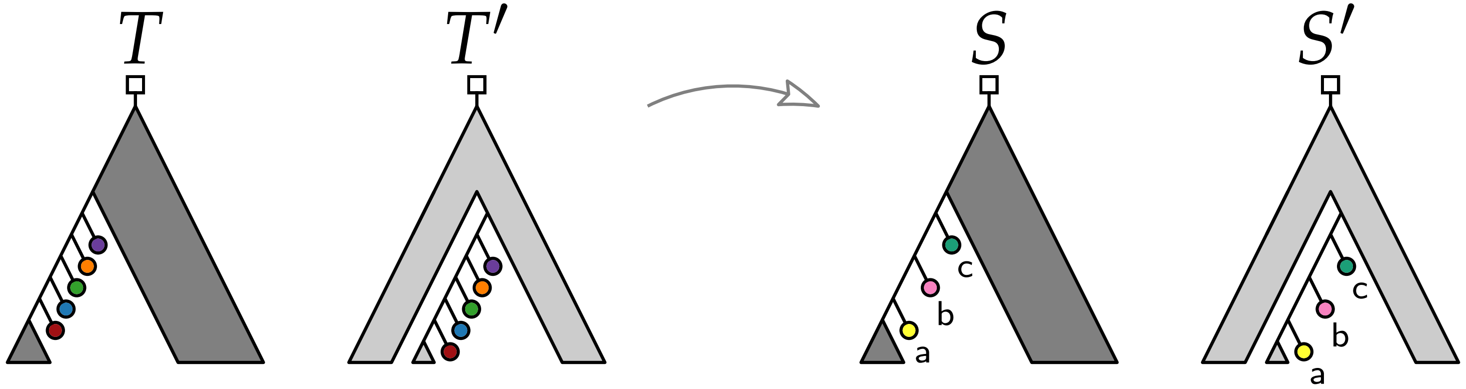
Proof.

- Show there is a tree with abc-chain in a MAF of S and S'.
- Swap abc-chain with original chain for MAF of T and T'.

Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

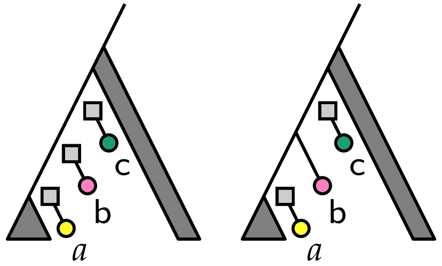


Lemma 6. Applying chain reduction is safe; i.e. $\mathrm{d}_{\mathsf{SPR}}(T, T') = \mathrm{d}_{\mathsf{SPR}}(S, S')$.

Proof.

- Consider embedding of a MAF F into S.

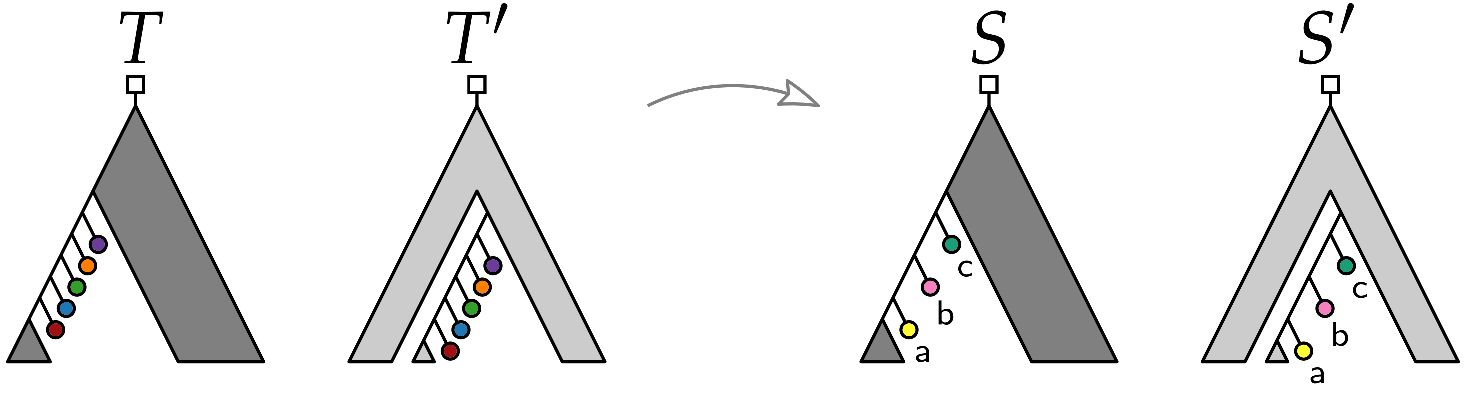
Case 1



Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

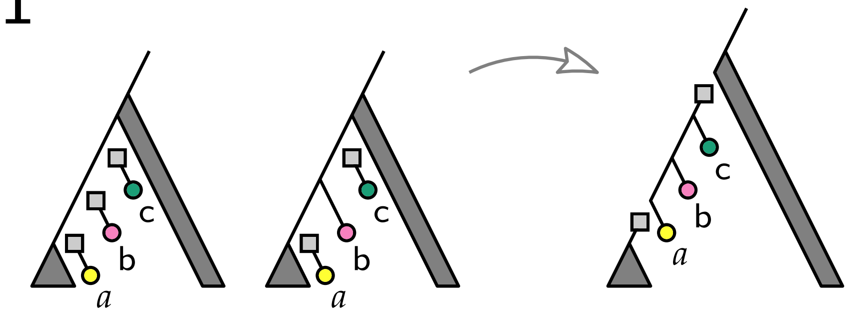


Lemma 6. Applying chain reduction is safe; i.e. $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

Proof.

- Consider embedding of a MAF F into S.

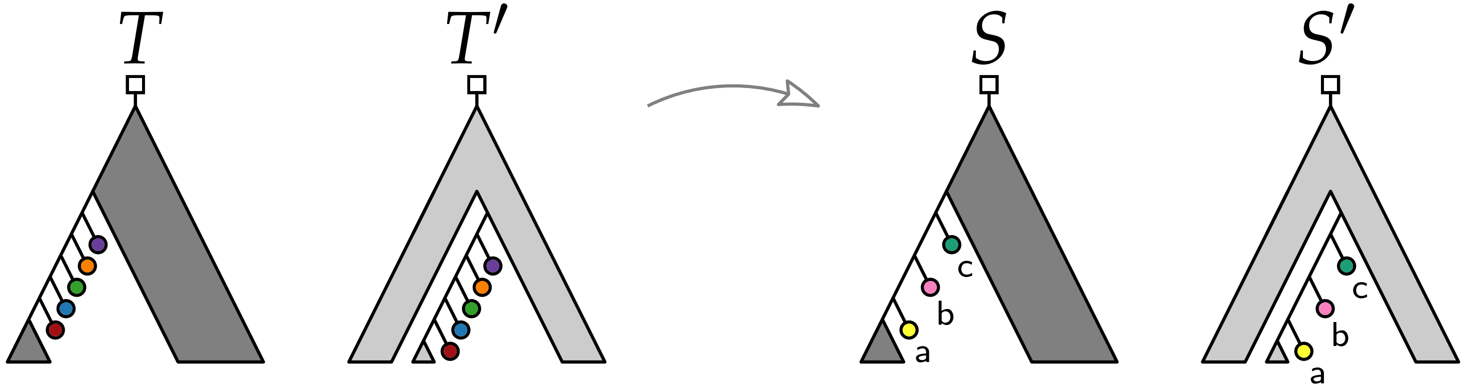
Case 1



Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

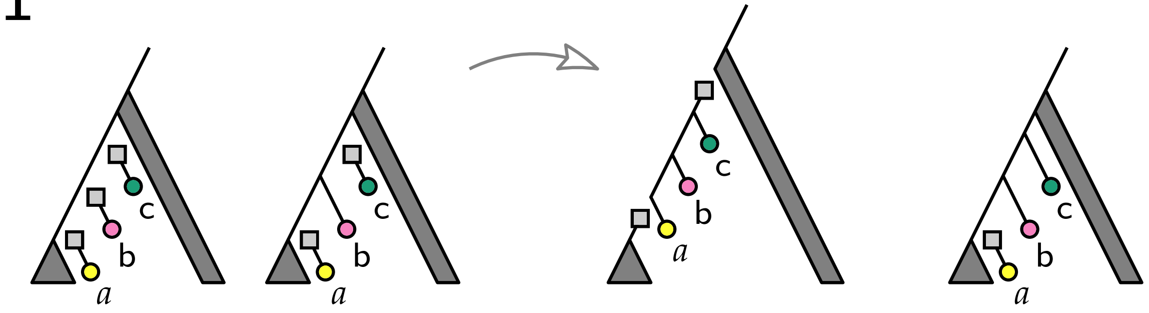


> **Lemma 6.** Applying chain reduction is safe; i.e. $\mathsf{d_{SPR}}(T, T') = \mathsf{d_{SPR}}(S, S')$.

Proof.

- Consider embedding of a MAF F into S.

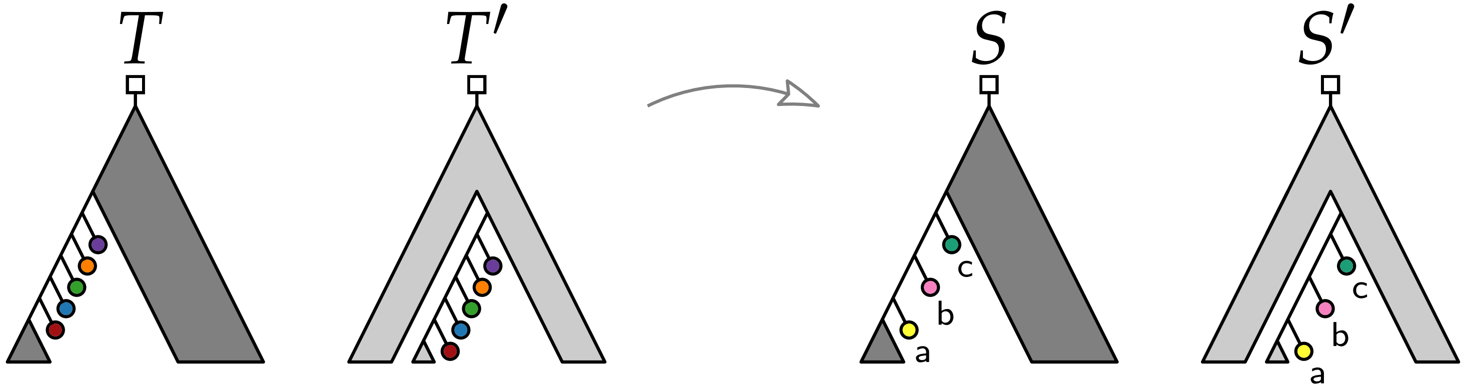
Case 1



Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

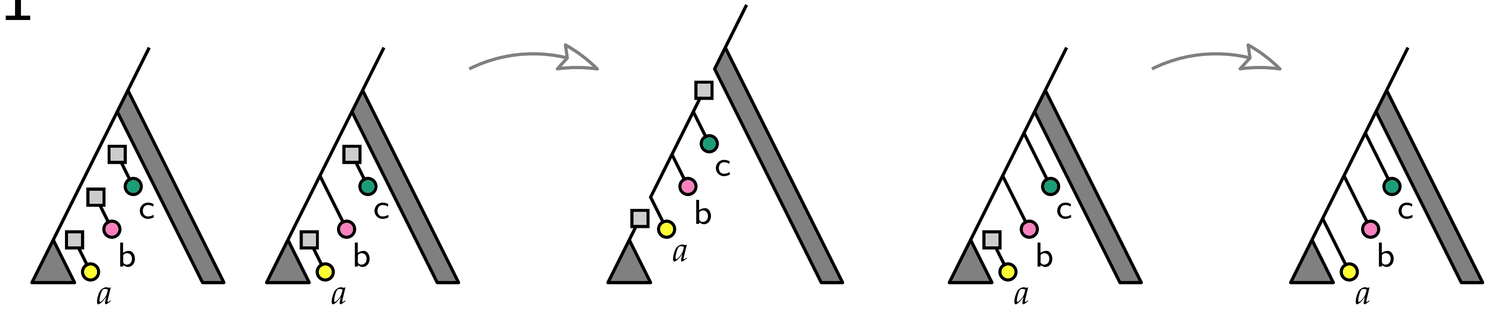


> **Lemma 6.** Applying chain reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

Proof.

- Consider embedding of a MAF F into S.

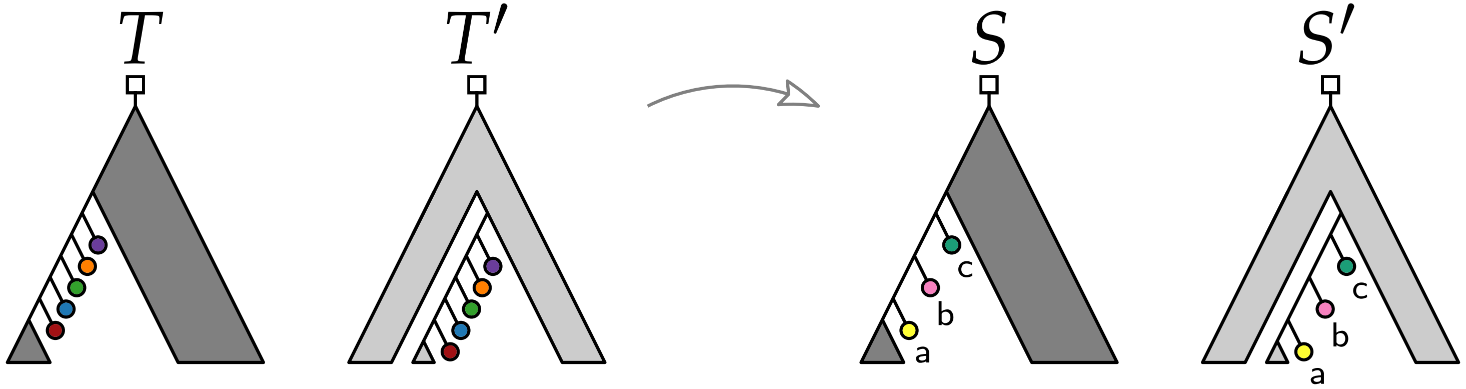
Case 1



Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

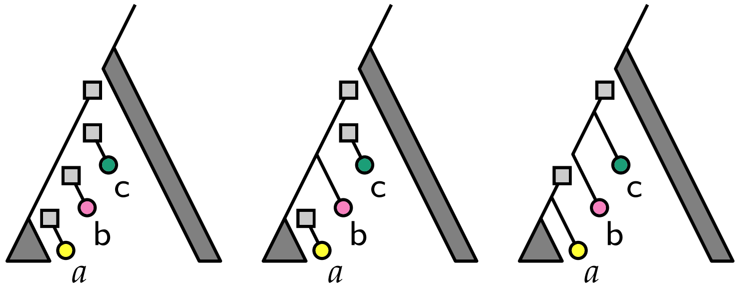


Lemma 6. Applying chain reduction is safe; i.e. $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

Proof.

- Consider embedding of a MAF F into S.

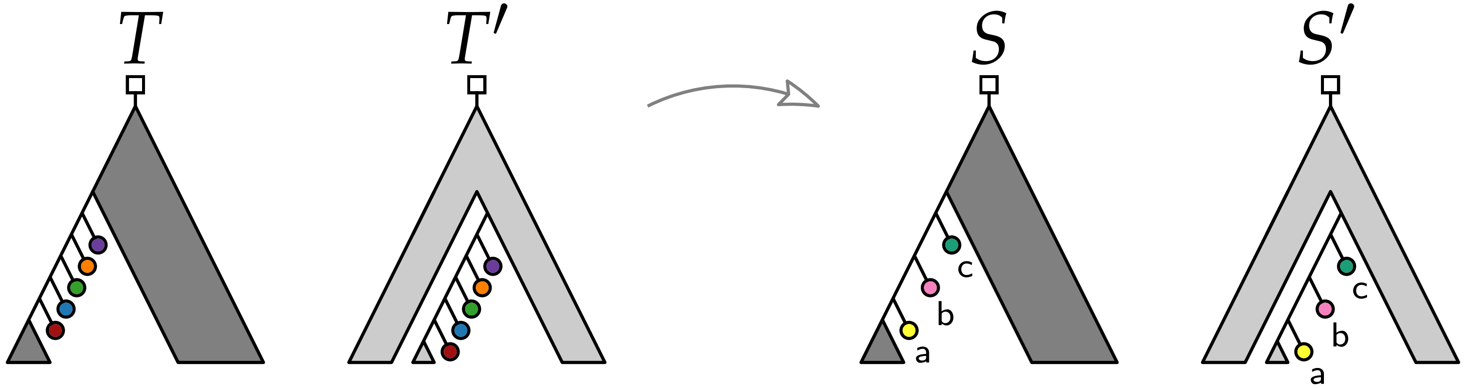
Case 2



Kernelisation – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically in both trees by three new leaves.

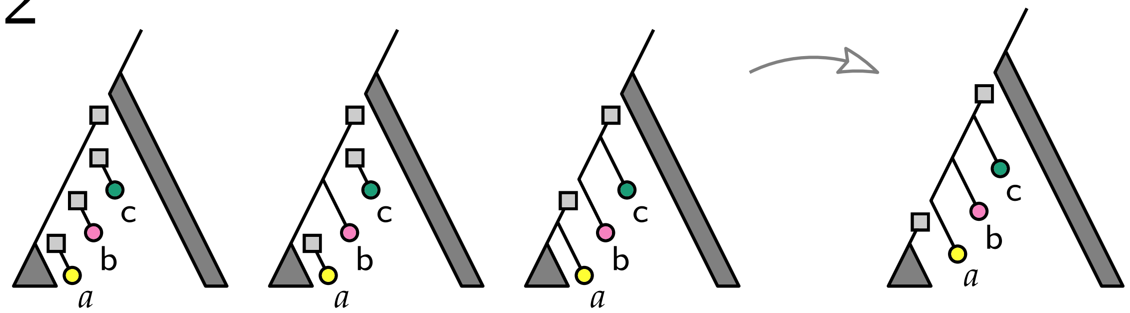


Lemma 6. Applying chain reduction is safe; i.e. $\mathsf{d}_{\mathsf{SPR}}(T,T') = \mathsf{d}_{\mathsf{SPR}}(S,S')$.

Proof.

- Consider embedding of a MAF F into S.

Case 2



Kernel size

Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then

$$|X'| \leq 28\, \mathsf{d}_{\mathsf{SPR}}(T, T').$$

Kernel size

Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then

$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be $\#$ of T_j that T_i overlaps with in embedding of F into S.

Kernel size

Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\, \mathrm{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be $\#$ of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k = 4\, \mathrm{d}_{\mathsf{SPR}}(T, T')$.

Kernel size

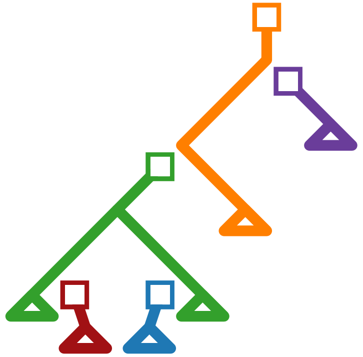
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then

$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.



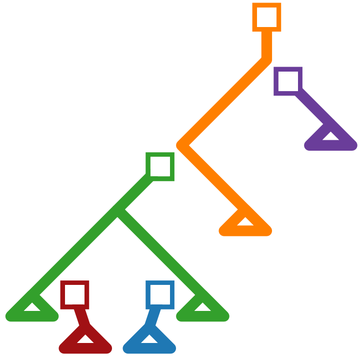
Kernel size

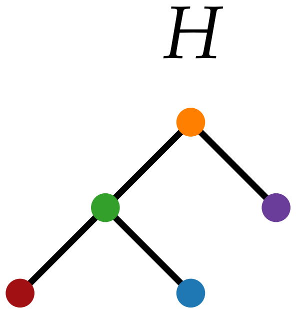
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.





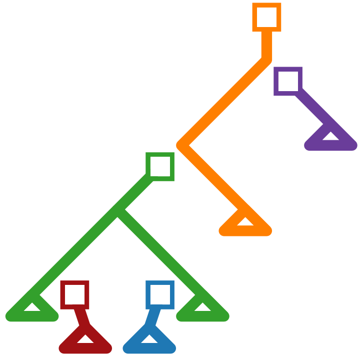
Kernel size

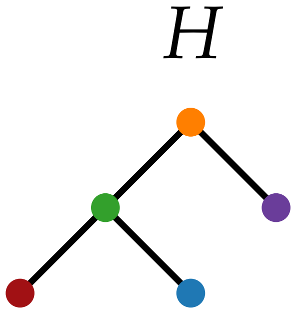
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.





$|V(H)| = k + 1$

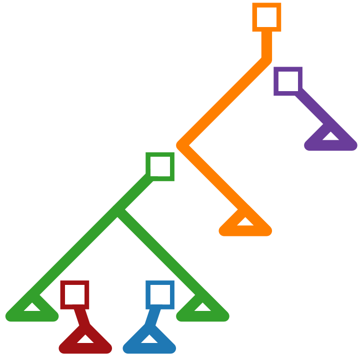
Kernel size

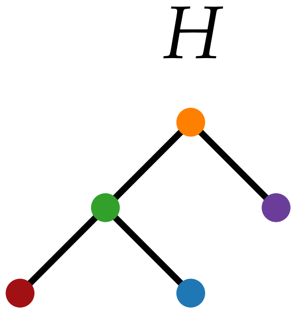
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'.
Then
$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T,T').$$

Proof. Let $F = \{T_\rho, T_1, \dots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T,T')$.





$$|V(H)| = k+1$$
$$= |E(H)| + 1$$

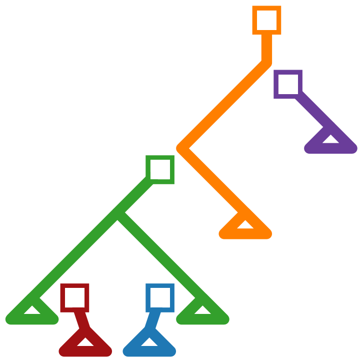
Kernel size

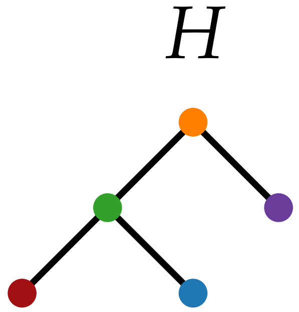
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \le 28\,\mathrm{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \dots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \le 4k = 4\,\mathrm{d}_{\mathsf{SPR}}(T, T')$.





$$\begin{aligned}|V(H)| &= k + 1\\ &= |E(H)| + 1\end{aligned}$$

$$\sum n(T_i) = 2|E(H)| \le 2k$$

Kernel size

Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then

$$|X'| \leq 28\,\mathrm{d_{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathrm{d_{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.

Kernel size

Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then

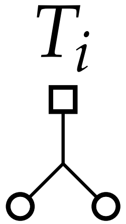
$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

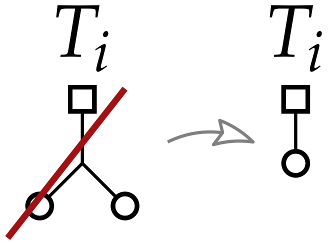
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'.
Then $$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

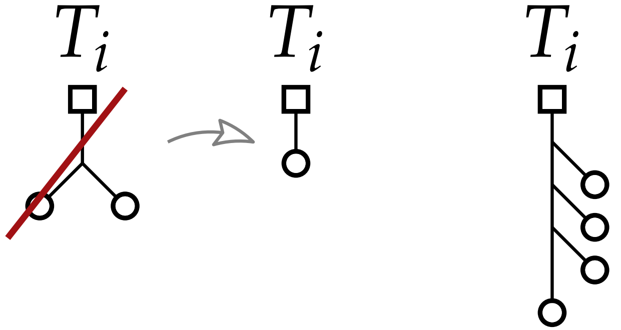
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

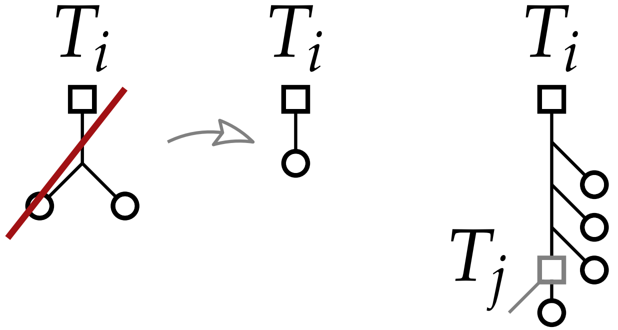
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathrm{d_{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathrm{d_{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

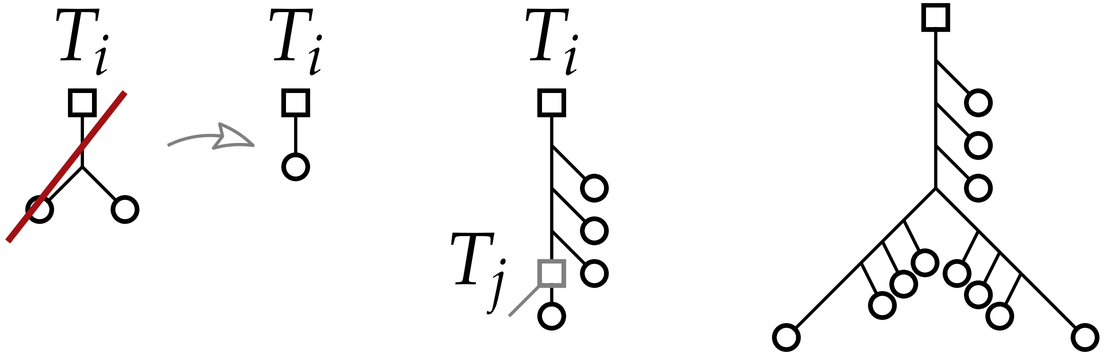
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathrm{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathrm{d}_{\mathsf{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

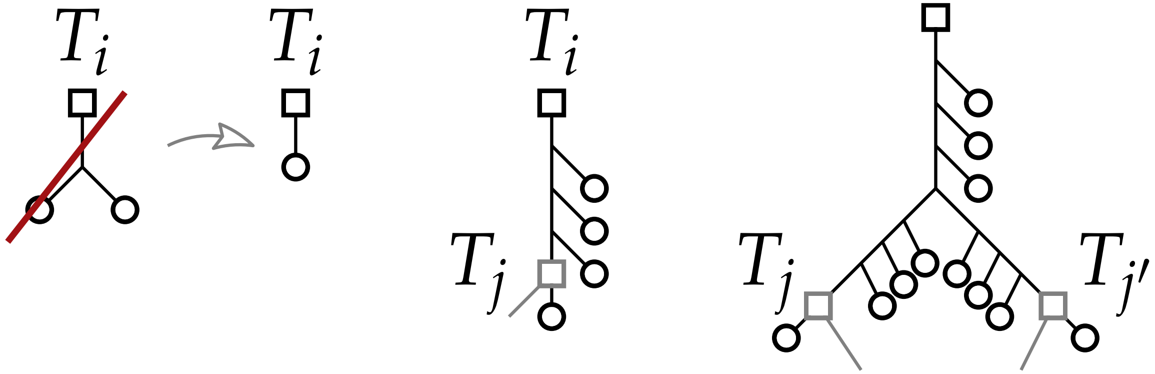
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathrm{d}_{\mathsf{SPR}}(T,T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathrm{d}_{\mathsf{SPR}}(T,T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then

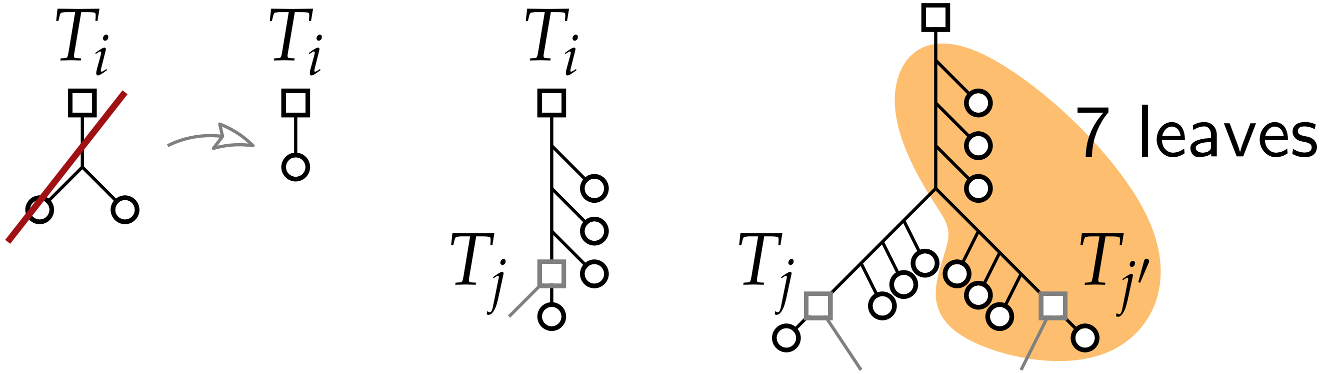
$$|X'| \leq 28\,\mathrm{d}_{\mathsf{SPR}}(T,T').$$

Proof. Let $F = \{T_\rho, T_1, \dots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathrm{d}_{\mathsf{SPR}}(T,T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



Kernel size

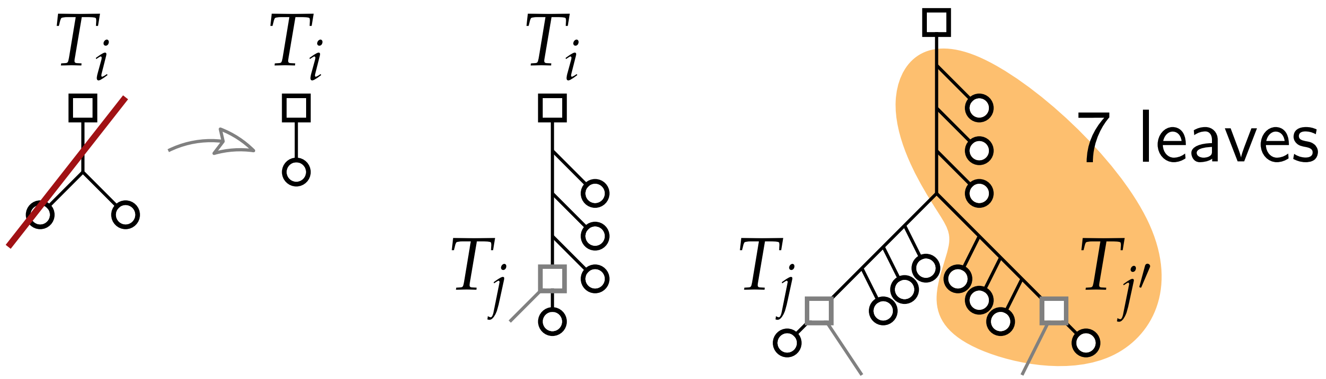
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathsf{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



$\sum_{i=\rho}^{k}$ # leaves of T_i

Kernel size

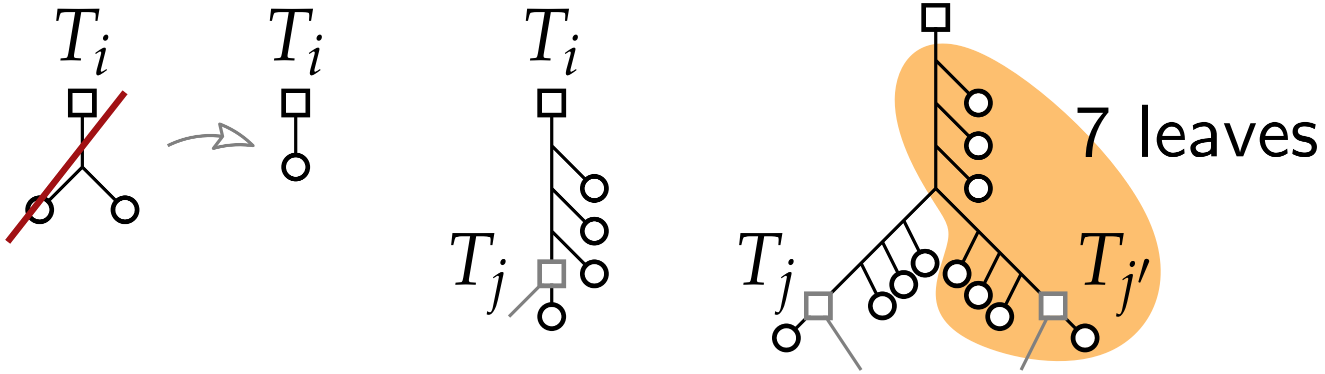
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathrm{d}_{\mathsf{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \dots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathrm{d}_{\mathsf{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



$\sum_{i=\rho}^{k}$ # leaves of T_i
$\leq \sum_{i=\rho}^{k} 7(n(T_i) + n'(T_i))$

Kernel size

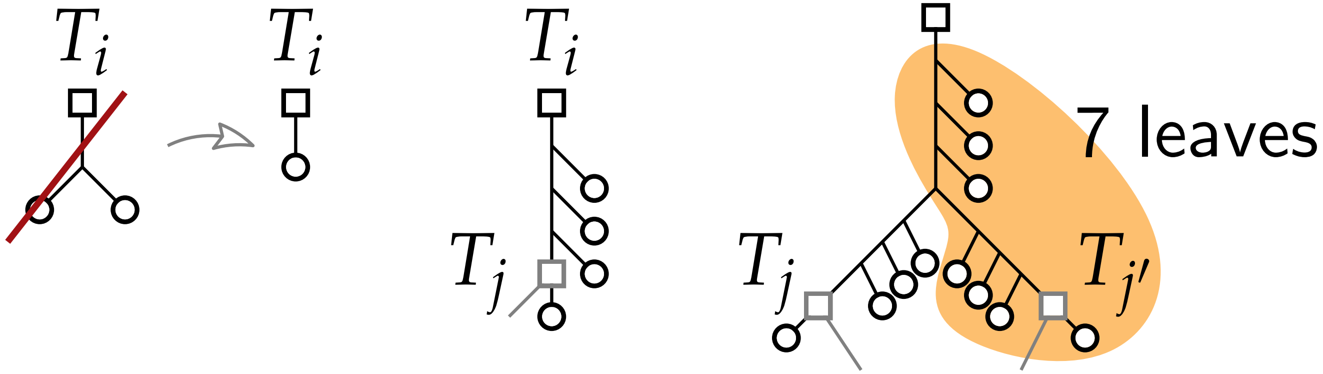
Theorem 7.
Reduce T and T' to S and S' by exhaustively applying the reduction rules. Let S and S' be on X'. Then
$$|X'| \leq 28\,\mathsf{d_{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for S and S'.

Let $n(T_i)$ be # of T_j that T_i overlaps with in embedding of F into S.

Claim 1. $\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k = 4\,\mathsf{d_{SPR}}(T, T')$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.



$\sum_{i=\rho}^{k}$ # leaves of T_i

$\leq \sum_{i=\rho}^{k} 7(n(T_i) + n'(T_i))$

$\leq 28k$

FPT algorithm

Theorem 8.
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

FPT algorithm

Theorem 8.
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Proof.

- Reduce T and T' to S and S' by exhaustively applying the reduction rules.
- Let S and S' be on X' and let $k = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

FPT algorithm

Theorem 8.
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Proof.

- Reduce T and T' to S and S' by exhaustively applying the reduction rules.
- Let S and S' be on X' and let $k = \mathsf{d}_{\mathsf{SPR}}(S, S')$.
- S has at most $4|X'|^2$ neighbours.

FPT algorithm

Theorem 8.
Computing $\mathrm{d}_{\mathsf{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathrm{d}_{\mathsf{SPR}}(T, T')$.

Proof.

- Reduce T and T' to S and S' by exhaustively applying the reduction rules.
- Let S and S' be on X' and let $k = \mathrm{d}_{\mathsf{SPR}}(S, S')$.
- S has at most $4|X'|^2$ neighbours.
  - S has less than $2|X'|$ edges to cut and to attach to.

FPT algorithm

Theorem 8.
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Proof.

- Reduce T and T' to S and S' by exhaustively applying the reduction rules.
- Let S and S' be on X' and let $k = \mathsf{d}_{\mathsf{SPR}}(S, S')$.
- S has at most $4|X'|^2$ neighbours.
  - S has less than $2|X'|$ edges to cut and to attach to.
- Length-k BFS from S visits at most $O\Big((4|X'|^2)^k\Big) = O((56k)^{2k})$ trees.

FPT algorithm

Theorem 8.
Computing $\mathrm{d}_{\mathrm{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathrm{d}_{\mathrm{SPR}}(T, T')$.

Proof.

- Reduce T and T' to S and S' by exhaustively applying the reduction rules.
- Let S and S' be on X' and let $k = \mathrm{d}_{\mathrm{SPR}}(S, S')$.
- S has at most $4|X'|^2$ neighbours.
  - S has less than $2|X'|$ edges to cut and to attach to.
- Length-k BFS from S visits at most $O\left(\left(4|X'|^2\right)^k\right) \overset{\text{by Theorem 7}}{=} O((56k)^{2k})$ trees.

FPT algorithm

Theorem 8.
Computing $\mathsf{d}_{\mathsf{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $\mathsf{d}_{\mathsf{SPR}}(T, T')$.

Proof.

- Reduce T and T' to S and S' by exhaustively applying the reduction rules.
- Let S and S' be on X' and let $k = \mathsf{d}_{\mathsf{SPR}}(S, S')$.
- S has at most $4|X'|^2$ neighbours.
  - S has less than $2|X'|$ edges to cut and to attach to.
- Length-k BFS from S visits at most $O\Big(\big(4|X'|^2\big)^k\Big) = O\big((56k)^{2k}\big)$ trees.

by Theorem 7

- Since $k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$, this yields an fpt algorithm.

Approximation algorithm

Idea.

- Given reduced trees T and T' we compute an agreement forest F by
- successively making "cuts" and "eliminations".
- This shrink T and T' further and further.
- Show that $|F|$ is at most $3|F'|$, where F' is a MAF of T and T'.

Approximation algorithm

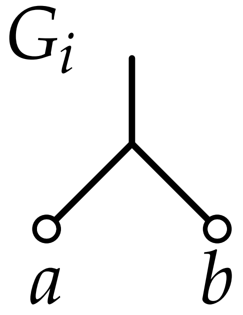
```
APPROXDSPR$(T, T')$
    $i \leftarrow 1$
    $G_i \leftarrow T$
    $H_i \leftarrow T'$
    while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do

    return $|H_i| - 1$
```

Approximation algorithm

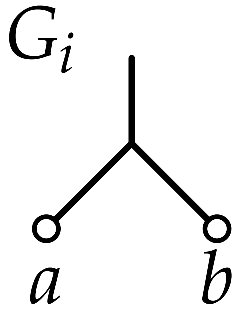
```
approxDSPR($T, T'$)
    $i \leftarrow 1$
    $G_i \leftarrow T$
    $H_i \leftarrow T'$
    while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do

    return $|H_i| - 1$
```



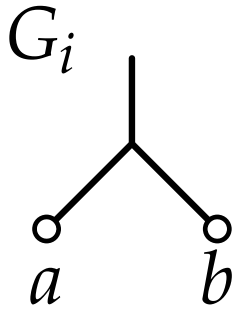
Approximation algorithm

```
$\textsc{approxDSPR}(T, T')$
    $i \leftarrow 1$
    $G_i \leftarrow T$
    $H_i \leftarrow T'$
    while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do
        find the case that applies to $a$ and $b$ in $H_i$
    return $|H_i| - 1$
```

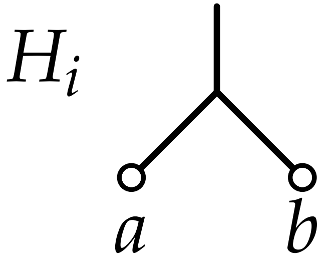


Approximation algorithm

```
$\textsc{approx}\mathrm{DSPR}(T, T')$
  $i \leftarrow 1$
  $G_i \leftarrow T$
  $H_i \leftarrow T'$
  while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do
      find the case that applies to $a$ and $b$ in $H_i$
  return $|H_i| - 1$
```

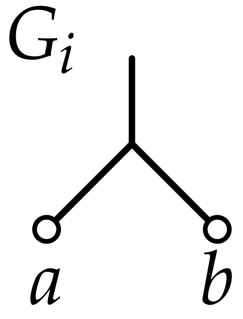


Case 1

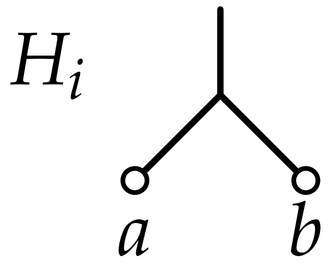


Approximation algorithm

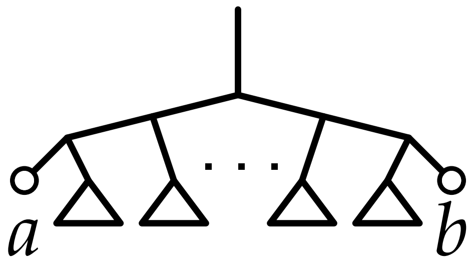
```
$\textsc{approx}\mathrm{DSPR}(T, T')$
    $i \leftarrow 1$
    $G_i \leftarrow T$
    $H_i \leftarrow T'$
    while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do
        find the case that applies to $a$ and $b$ in $H_i$
    return $|H_i| - 1$
```



Case 1

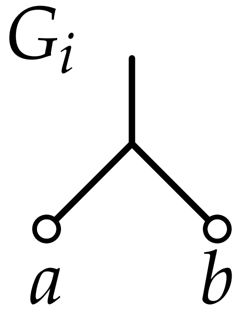


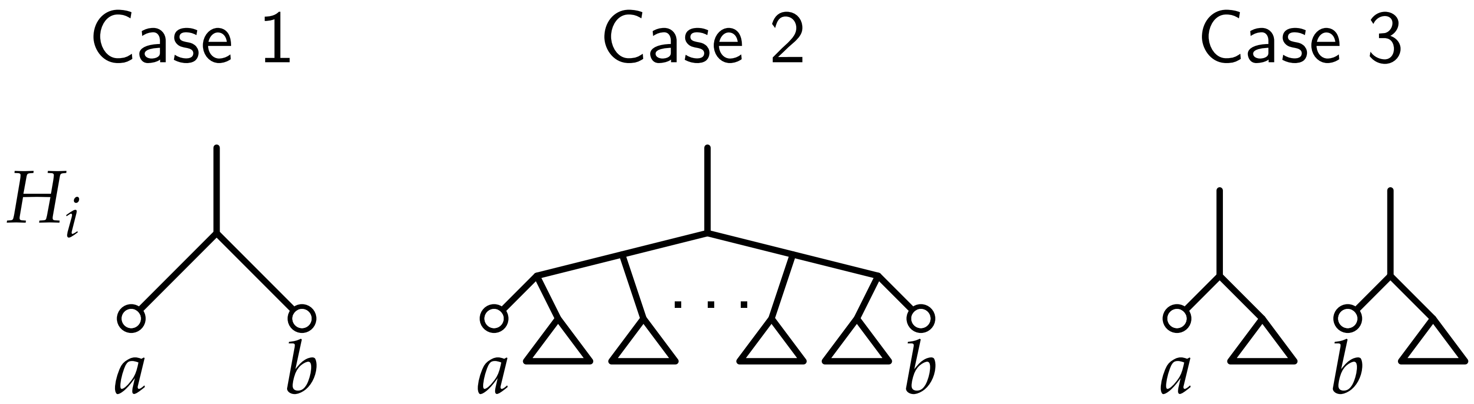
Case 2



Approximation algorithm

```
APPROXDSPR($T, T'$)
  $i \leftarrow 1$
  $G_i \leftarrow T$
  $H_i \leftarrow T'$
  while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do
      find the case that applies to $a$ and $b$ in $H_i$
  return $|H_i| - 1$
```



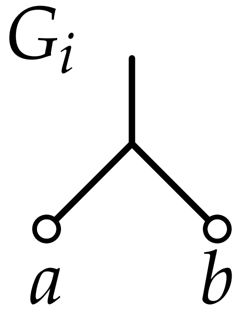

Case 1

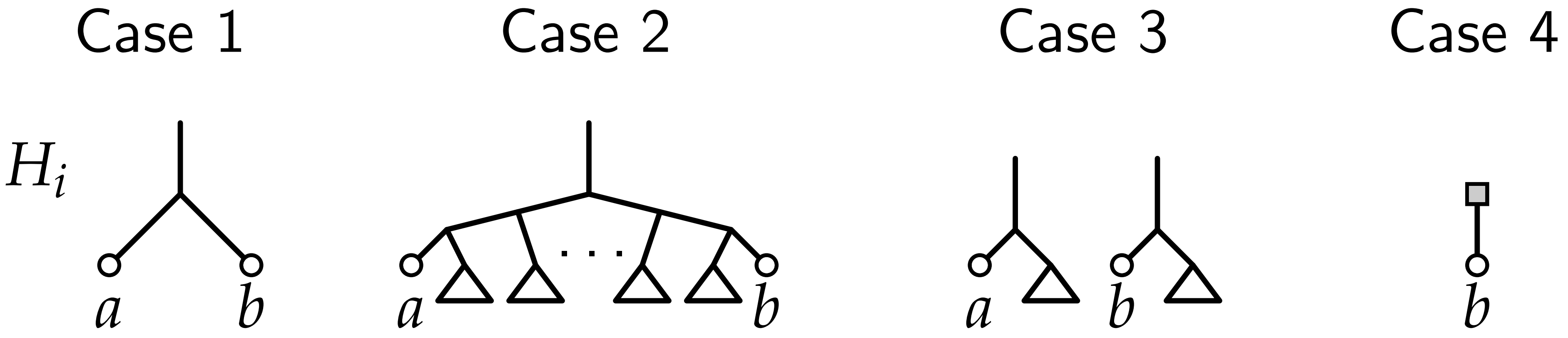
Case 2

Case 3

Approximation algorithm

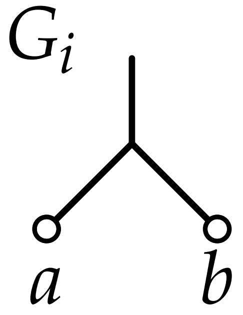
```
APPROXDSPR$(T, T')$
    $i \leftarrow 1$
    $G_i \leftarrow T$
    $H_i \leftarrow T'$
    while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do
        find the case that applies to $a$ and $b$ in $H_i$
    return $|H_i| - 1$
```

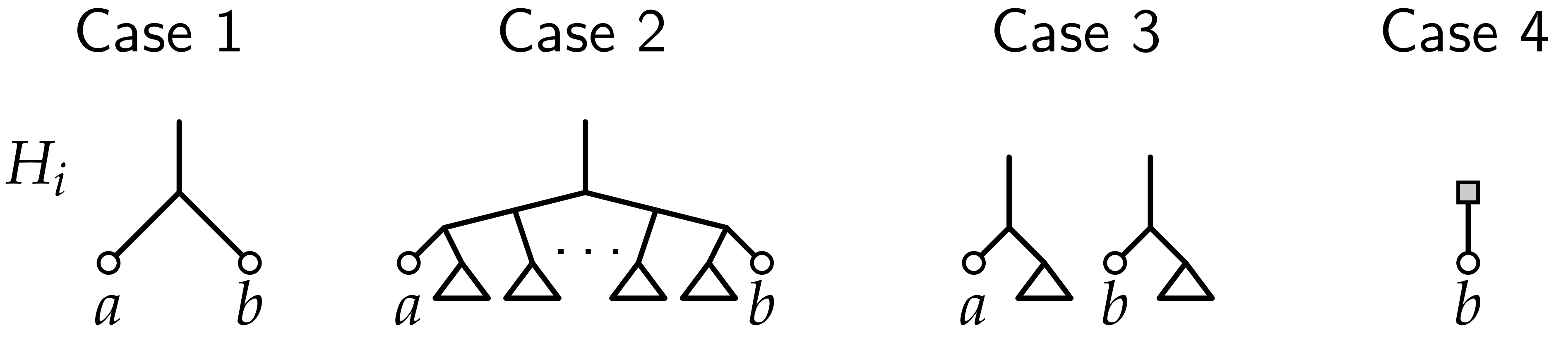




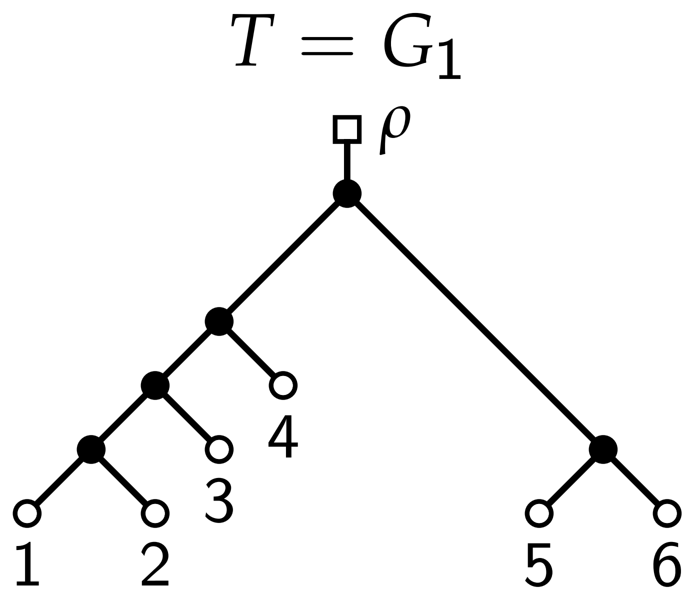
Approximation algorithm

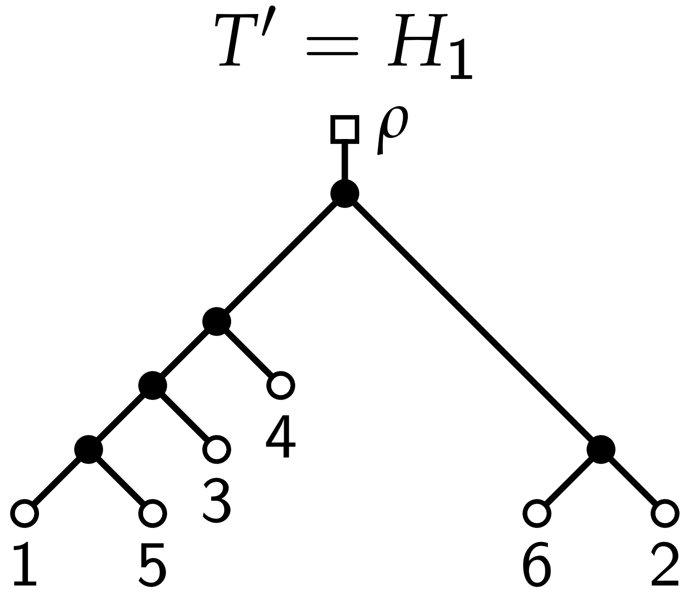
```
$\textsc{approx}\mathrm{DSPR}(T, T')$
  $i \leftarrow 1$
  $G_i \leftarrow T$
  $H_i \leftarrow T'$
  while $\exists$ pair of sibling leaves $a$ and $b$ in $G_i$ do
      find the case that applies to $a$ and $b$ in $H_i$
      apply the corresponding transaction
      to obtain $G_{i+1}$ from $G_i$ and $H_{i+1}$ from $H_i$
      $i++$
  return $|H_i| - 1$
```



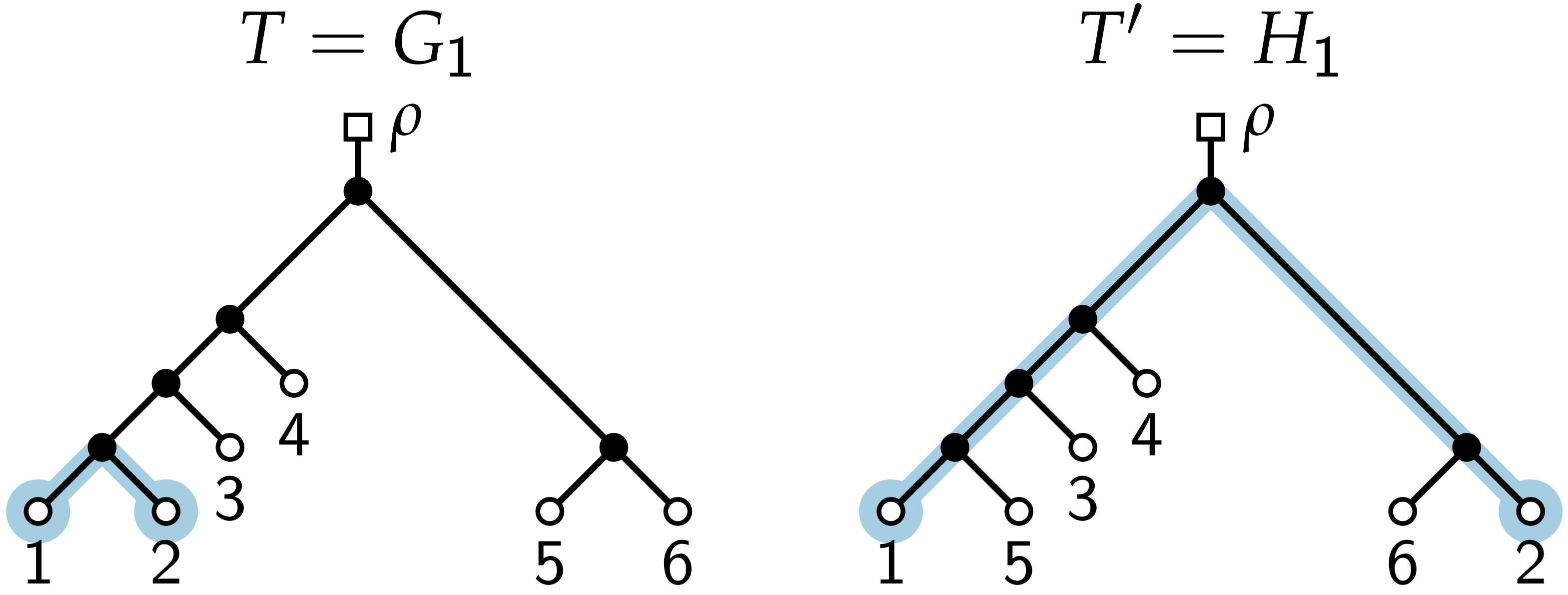


Approximation algorithm – example

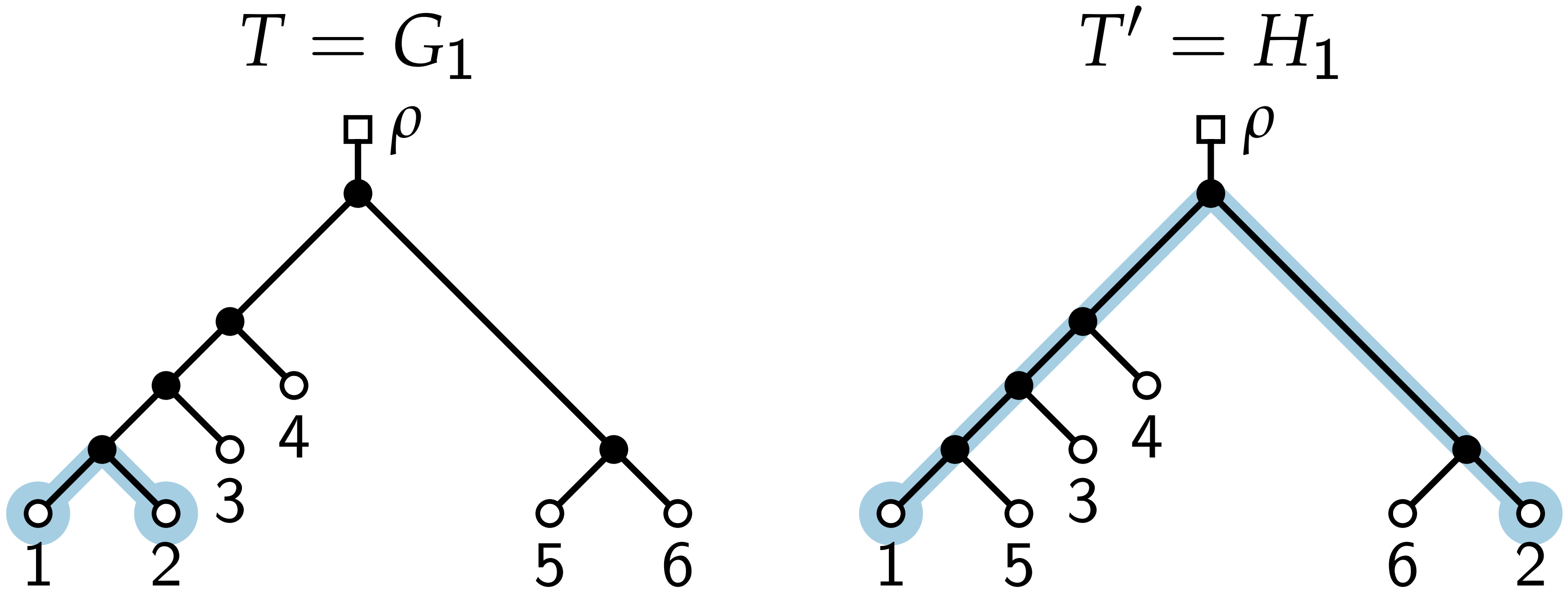




Approximation algorithm – example

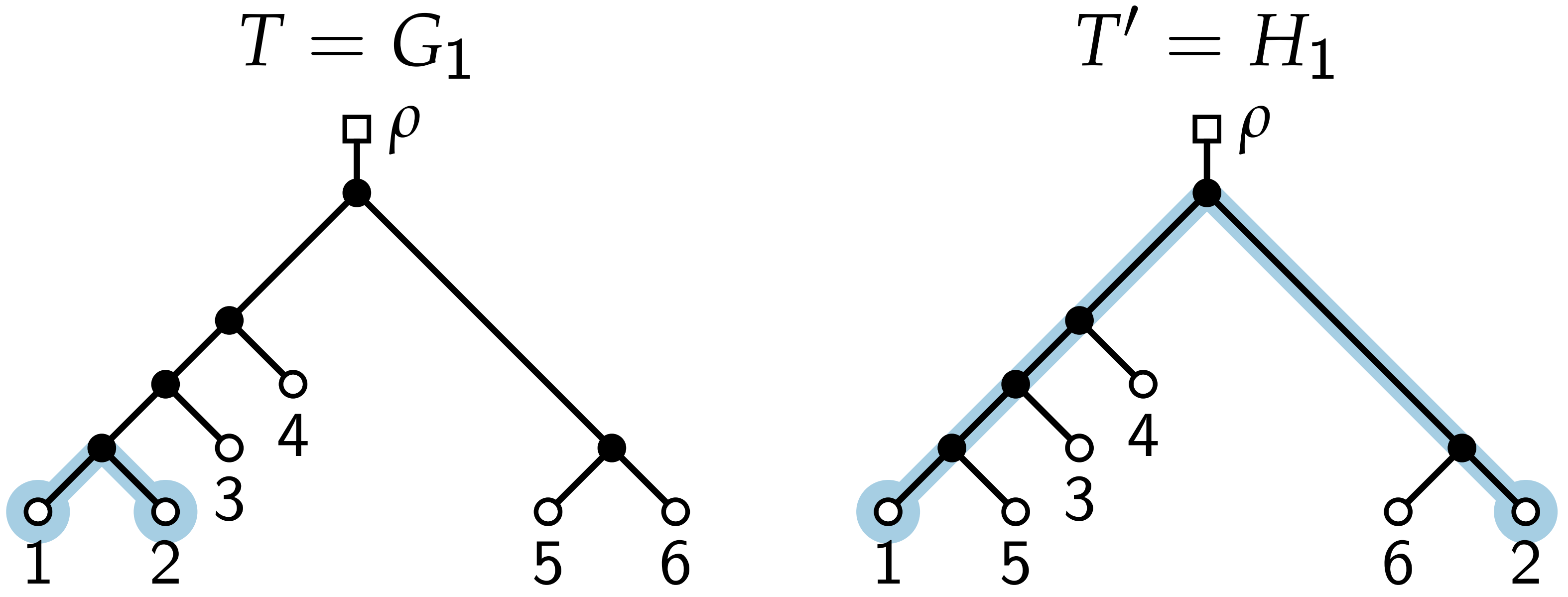


Approximation algorithm – example



- Should we cut of the leaves 1 or 2 or all in between them in H_1?

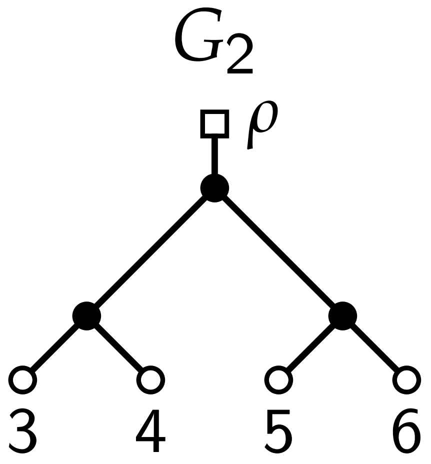
Approximation algorithm – example

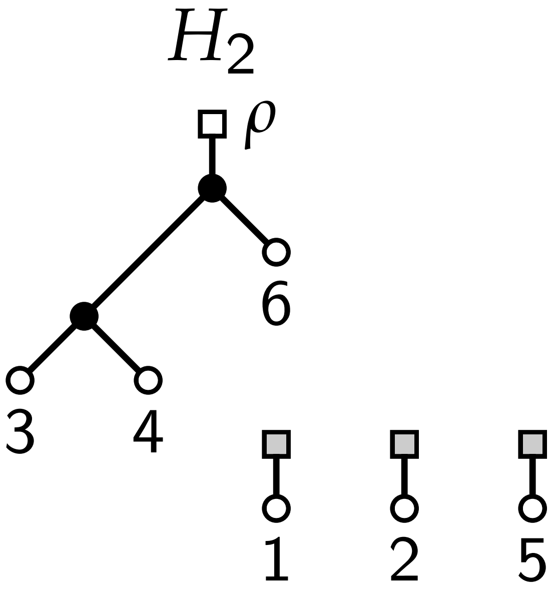


Case 2

- Should we cut of the leaves 1 or 2 or all in between them in H_1?
- Do parts of each!

Approximation algorithm – example

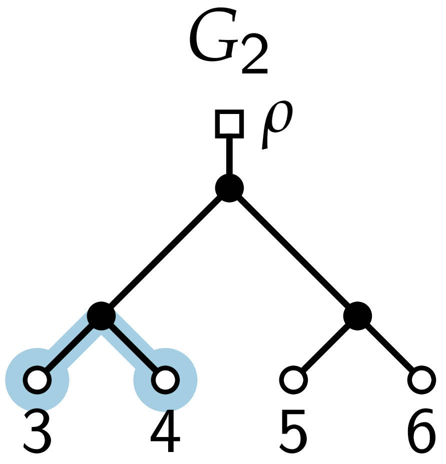


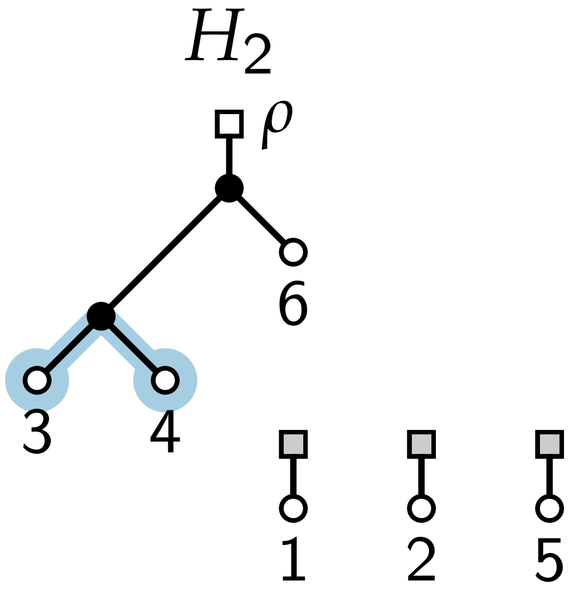


Case 2

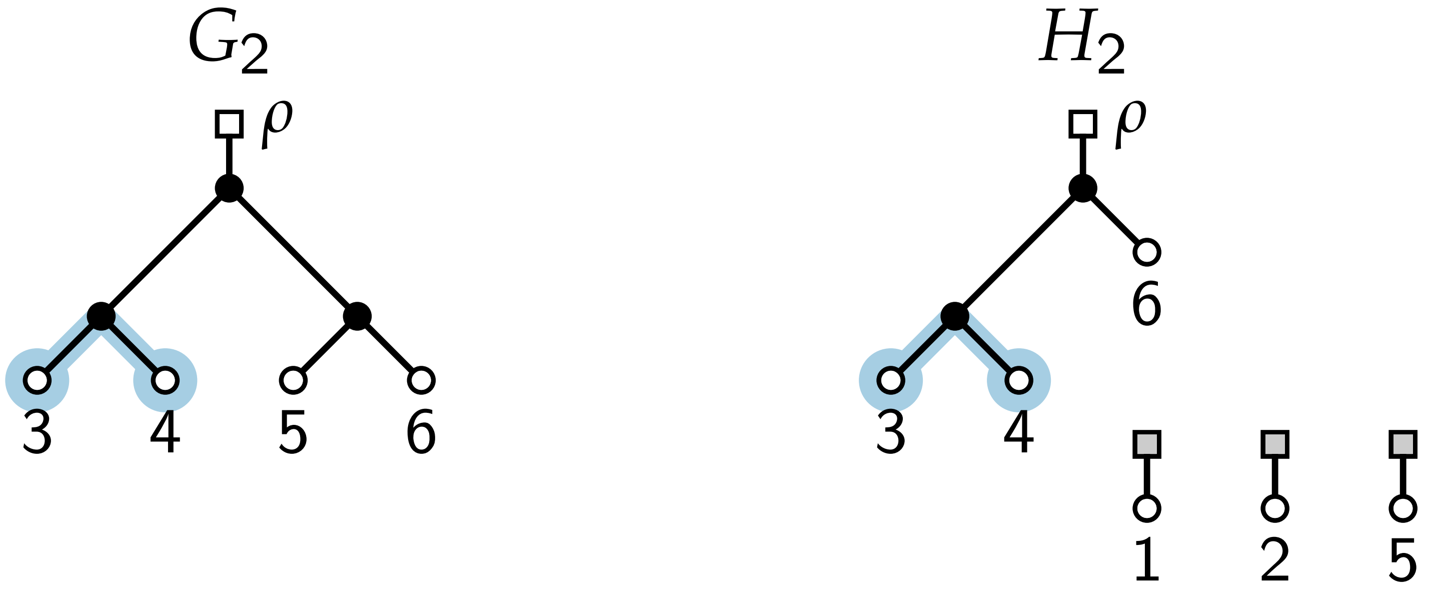
- Should we cut of the leaves 1 or 2 or all in between them in H_1?
- Do parts of each!

Approximation algorithm – example





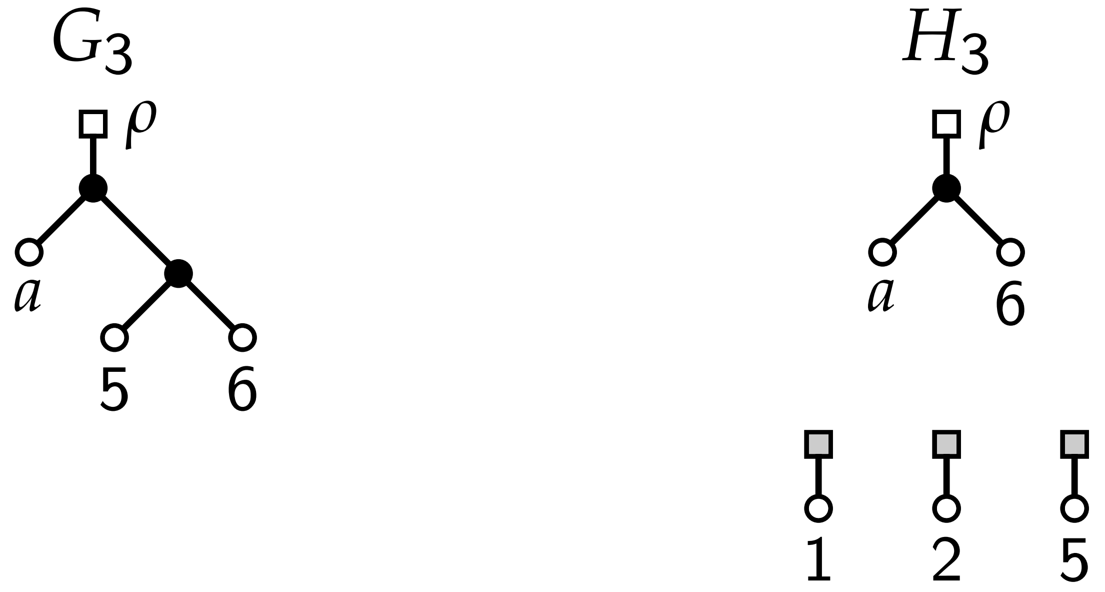
Approximation algorithm – example



Case 1

- If the same cherry occurs in H_i, we can simply reduce it.

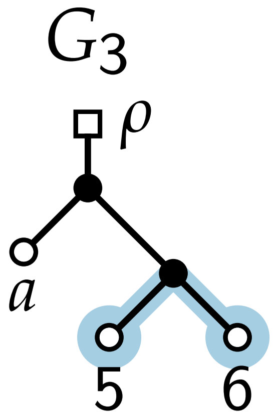
Approximation algorithm – example

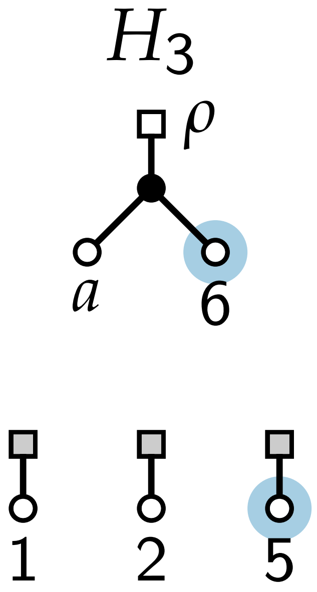


Case 1

- If the same cherry occurs in H_i, we can simply reduce it.

Approximation algorithm – example

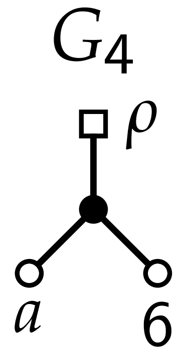


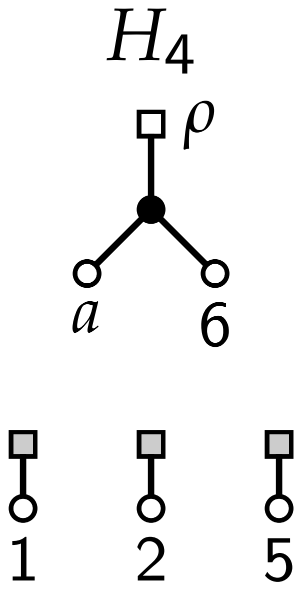


Case 4

- Leaf b is the only leaf of a tree in H_i.
- Cut off b in G_i.

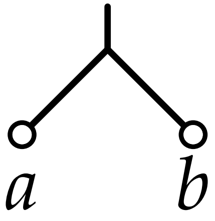
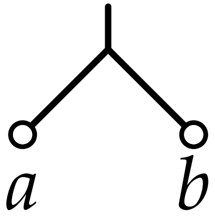
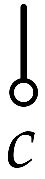
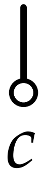
Approximation algorithm – example



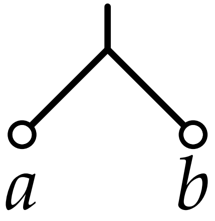
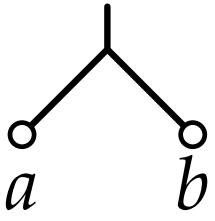
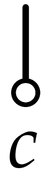
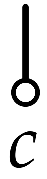


- Return 3.

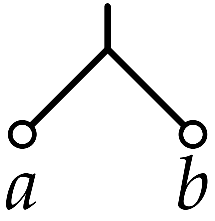
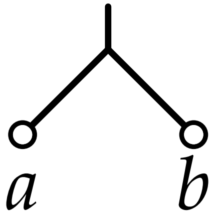
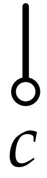
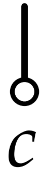
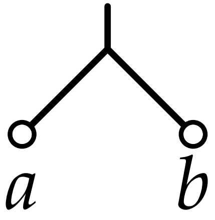
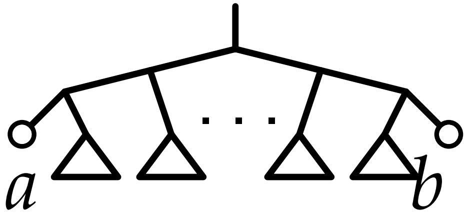
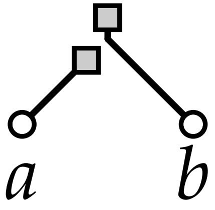
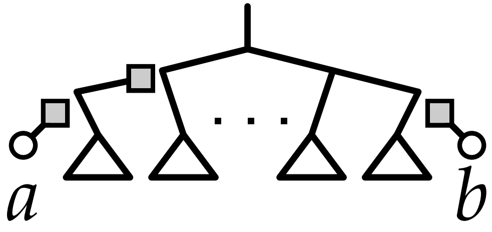
Approximation algorithm – analysis

Case	G_i	H_i		G_{i+1}	H_{i+1}	Cost
1						

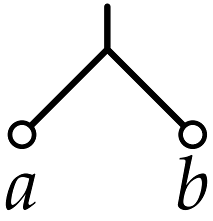
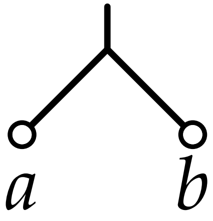
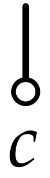
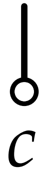
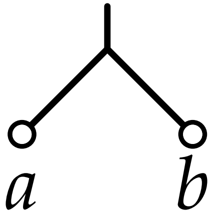
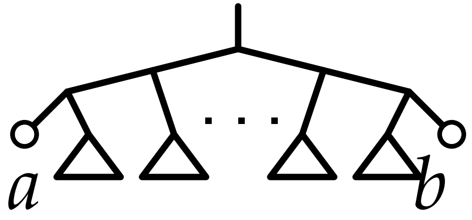
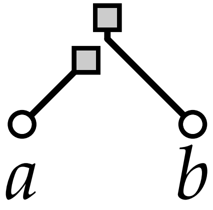
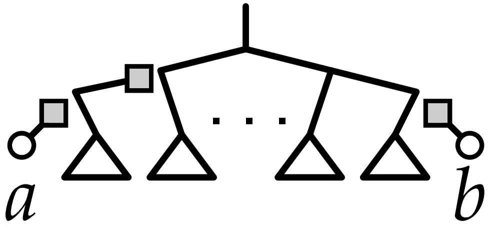
Approximation algorithm – analysis

Case	G_i	H_i	→	G_{i+1}	H_{i+1}	Cost
1						no mistake

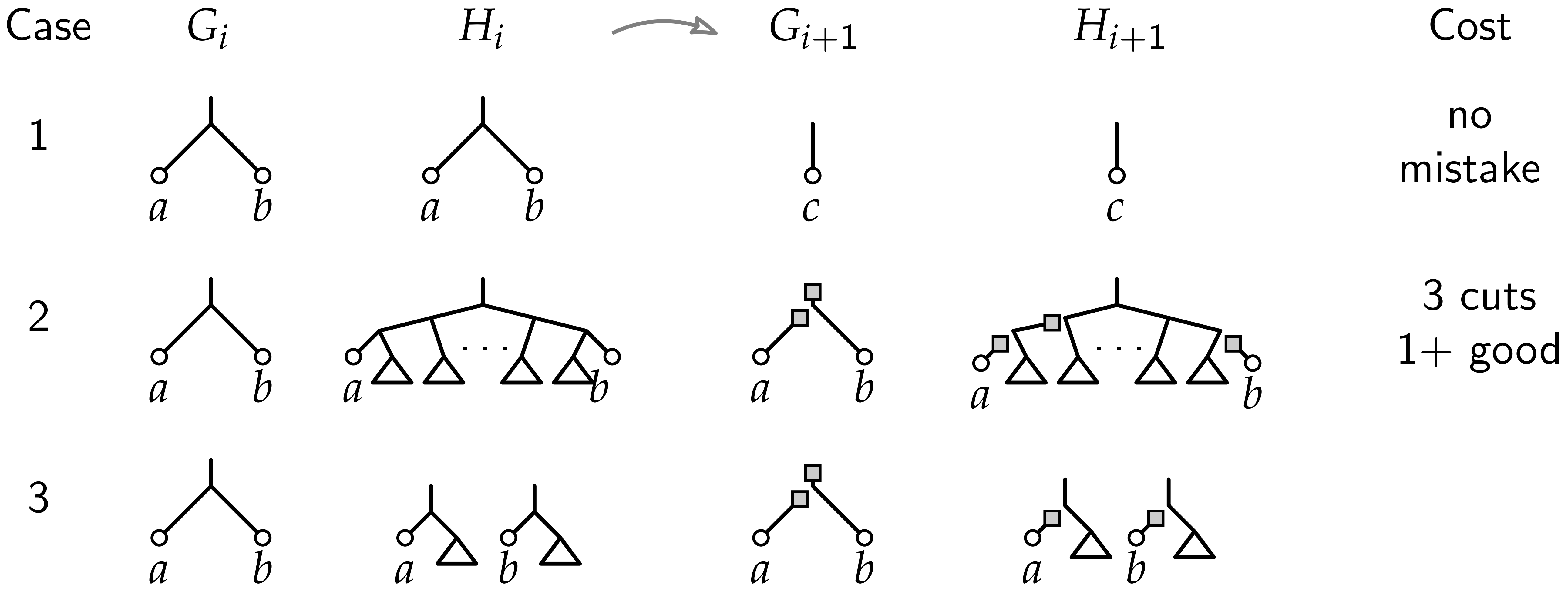
Approximation algorithm – analysis

Case	G_i	H_i	G_{i+1}	H_{i+1}	Cost
1					no mistake
2					

Approximation algorithm – analysis

Case	G_i	H_i	G_{i+1}	H_{i+1}	Cost
1					no mistake
2					3 cuts 1+ good

Approximation algorithm – analysis



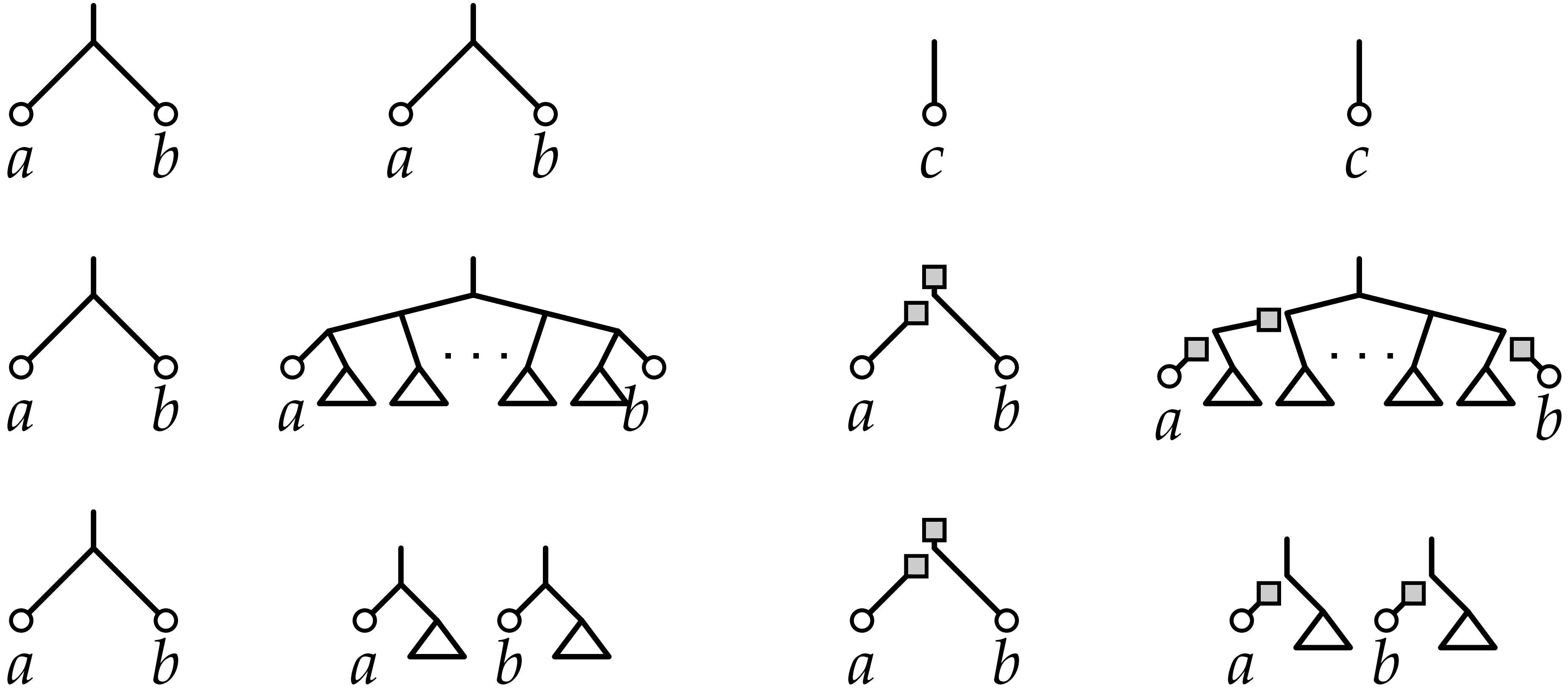
Case G_i H_i → G_{i+1} H_{i+1} Cost

1 no mistake

2 3 cuts 1+ good

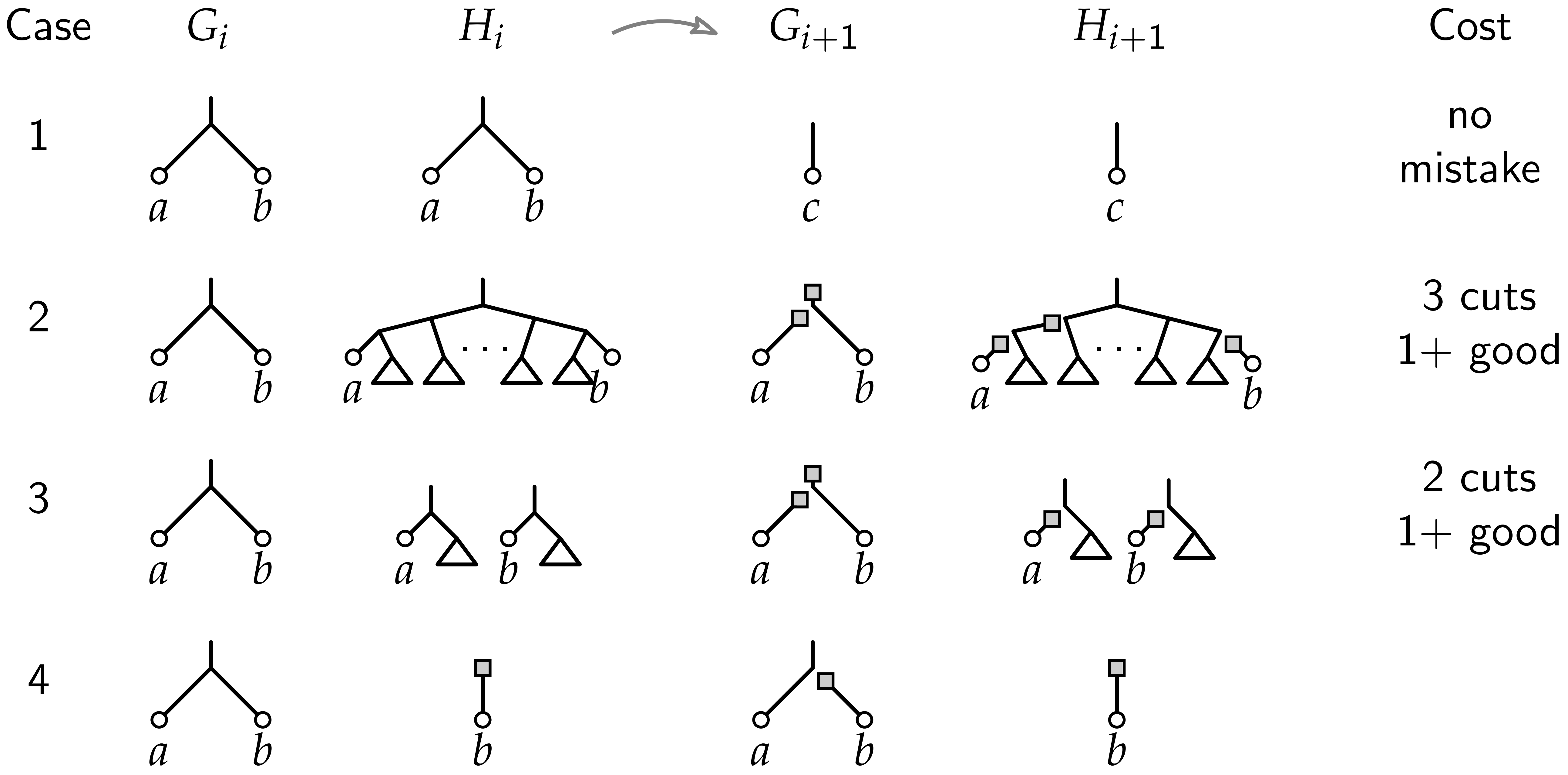
3

Approximation algorithm – analysis



Case	G_i	H_i	G_{i+1}	H_{i+1}	Cost
1					no mistake
2					3 cuts 1+ good
3					2 cuts 1+ good

Approximation algorithm – analysis



Case G_i H_i → G_{i+1} H_{i+1} Cost

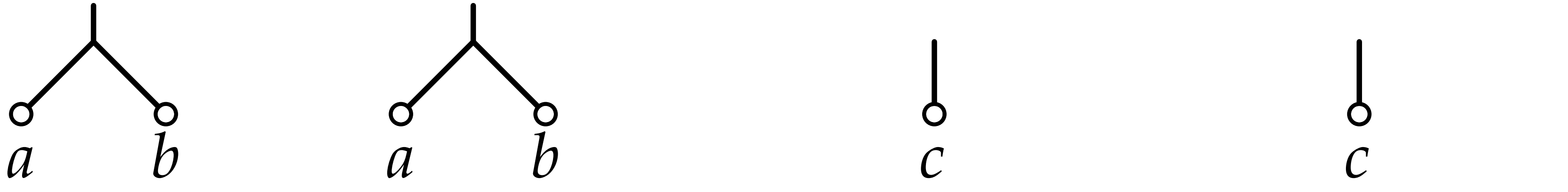
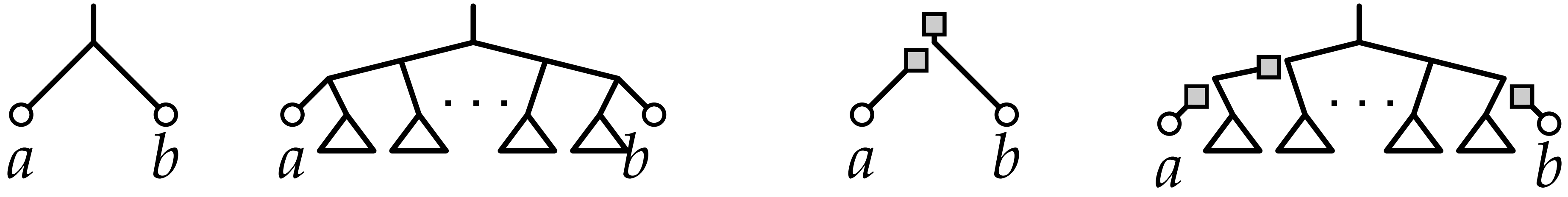
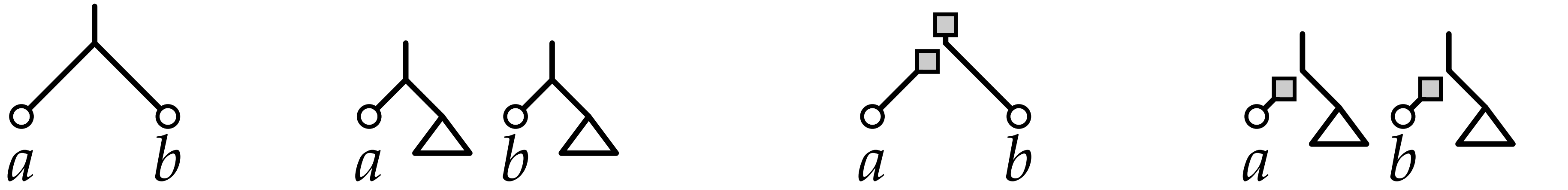
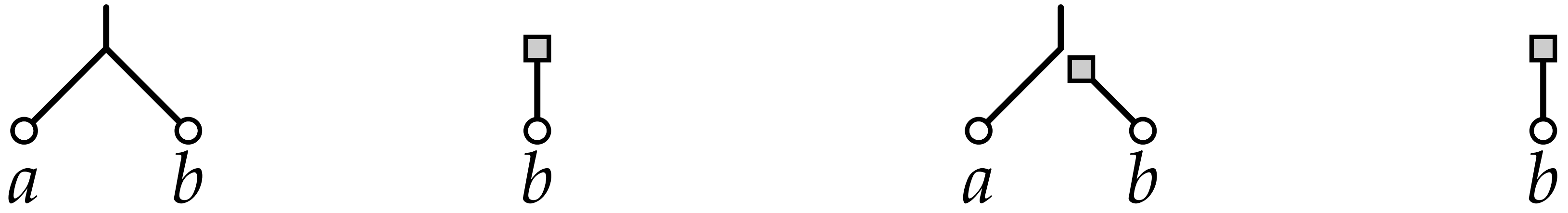
1 — no mistake

2 — 3 cuts 1+ good

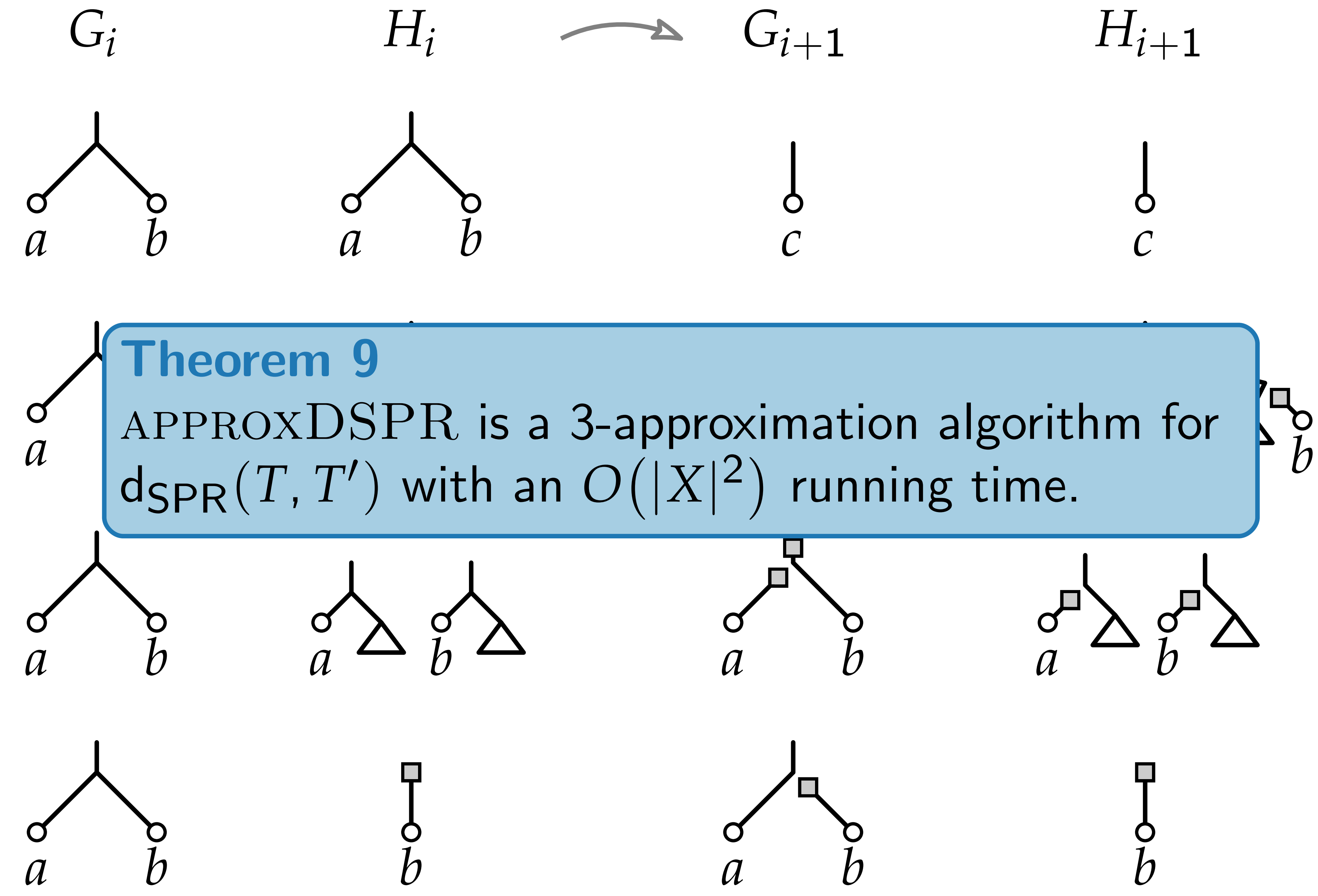
3 — 2 cuts 1+ good

4

Approximation algorithm – analysis

Case	G_i H_i → G_{i+1} H_{i+1}	Cost
1		no mistake
2		3 cuts 1+ good
3		2 cuts 1+ good
4		1 cut 1 good

Approximation algorithm – analysis



Case	G_i	H_i	G_{i+1}	H_{i+1}	Cost
1					no mistake
2					3 cuts 1+ good
3					2 cuts 1+ good
4					1 cut 1 good

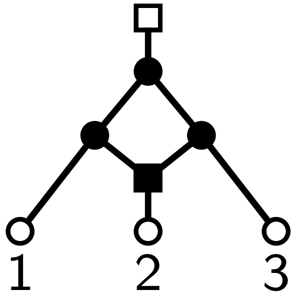
Theorem 9

$\textsc{approx}$DSPR is a 3-approximation algorithm for $d_{\mathsf{SPR}}(T, T')$ with an $O(|X|^2)$ running time.

Discussion

Phylogenetic trees.

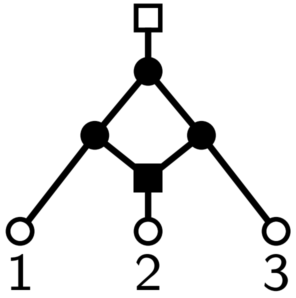
- There are other classes of phylogenetic trees: unrooted, non-binary, ranked, . . .
- Trees can be generalized to **phylogenetic networks**, which can also have indegree 2 outdegree 1 vertices.



Discussion

Phylogenetic trees.

- There are other classes of phylogenetic trees: unrooted, non-binary, ranked, . . .
- Trees can be generalized to **phylogenetic networks**, which can also have indegree 2 outdegree 1 vertices.



Maximum Agreement Forests.

- Reframing (characterising) a problem in a different way, can sometimes make your life a lot easier.
- MAF can be generalized to Maximum Agreement Graphs, but these don’t characterize the SPR-distance of networks anymore.

Discussion

Kernelization.

- Kernelization is an important technique to construct fpt algorithms.
- Result important since SPR-distance small in practice.
- Reduction rules actually give a kernel of size at most $15k - 9$.
- With further reduction rules can get size below $11k - 9$. [KL '18]
- Divide & conquer algorithm can (in practice) reduce further reduce problem sizes. [LS '11]

Discussion

Kernelization.

- Kernelization is an important technique to construct fpt algorithms.
- Result important since SPR-distance small in practice.
- Reduction rules actually give a kernel of size at most $15k - 9$.
- With further reduction rules can get size below $11k - 9$. [KL '18]
- Divide & conquer algorithm can (in practice) reduce further reduce problem sizes. [LS '11]

Approximation algorithm.

- There exist 2-approximation algorithms for the SPR-distance with a running time in $\mathcal{O}(n^3)$. [CHW '17]

Literature

Original papers:

- [BS ’05] “On the computational complexity of the rooted subtree prune and regraft distance” for SPR, MAF, characterisation, fpt, divide & conquer
- [RSW ’06] “The maximum agreement forest problem: Approximation algorithms and computational experiments”

Referenced papers:

- [HJWZ ’96] “On the complexity of comparing evolutionary trees” for NP-hardness proof
- [KL ’19] “New reduction rules for the tree bisection and reconnection distance”
- [CHW ’17] “A New 2-Approximation Algorithm for rSPR Distance”
- [LS11] “A cluster reduction for computing the subtree distance between phylogenies”