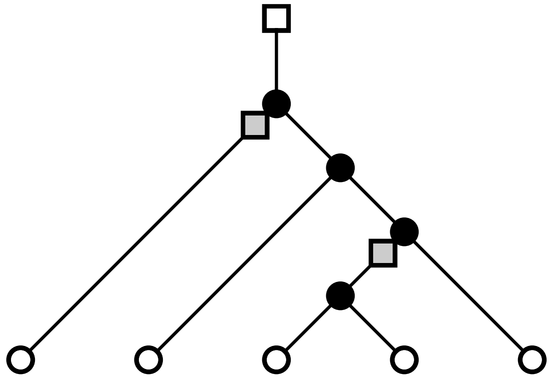
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

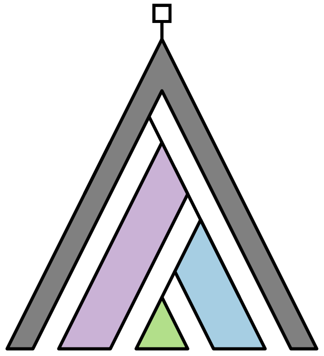
# Advanced Algorithms

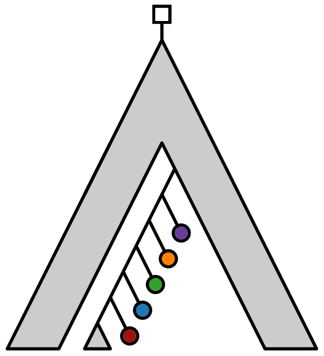
## Rearrangement Distance of Phylogenetic Trees

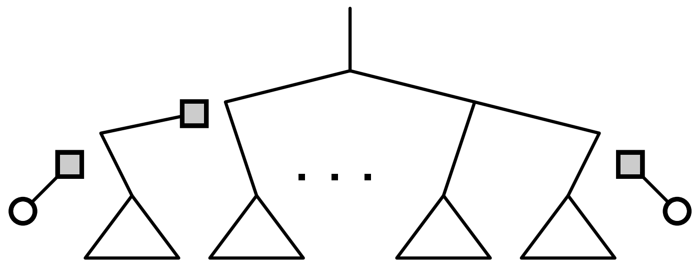
### Kernelization, FPT, Approximation Algorithm

Johannes Zink · WS22



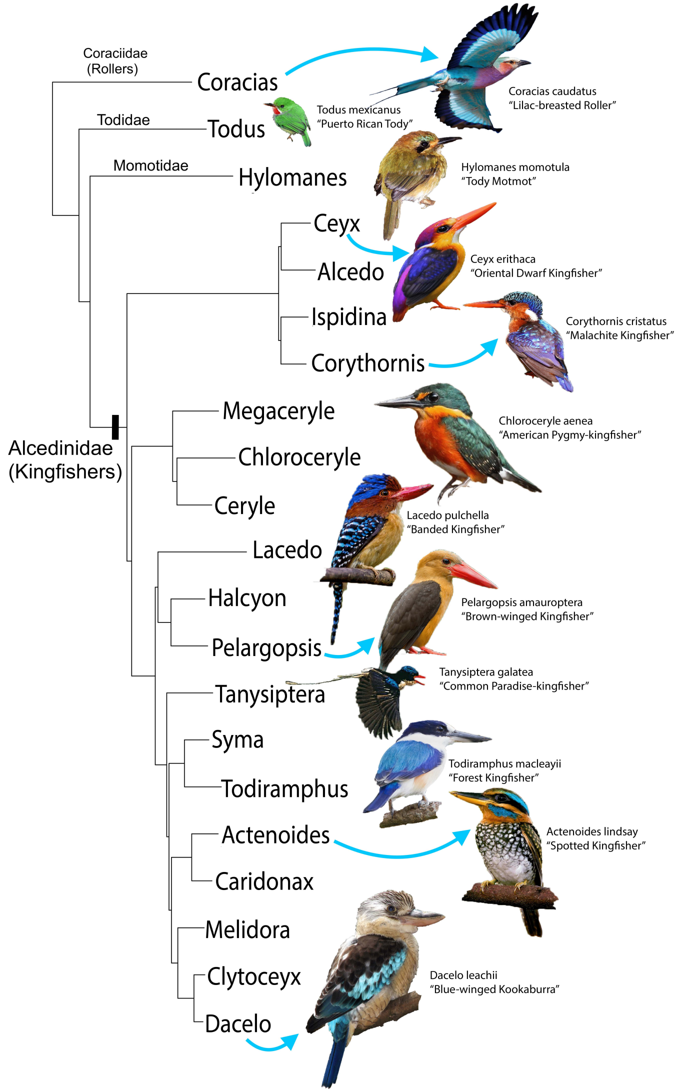






# Phylogenetic Trees

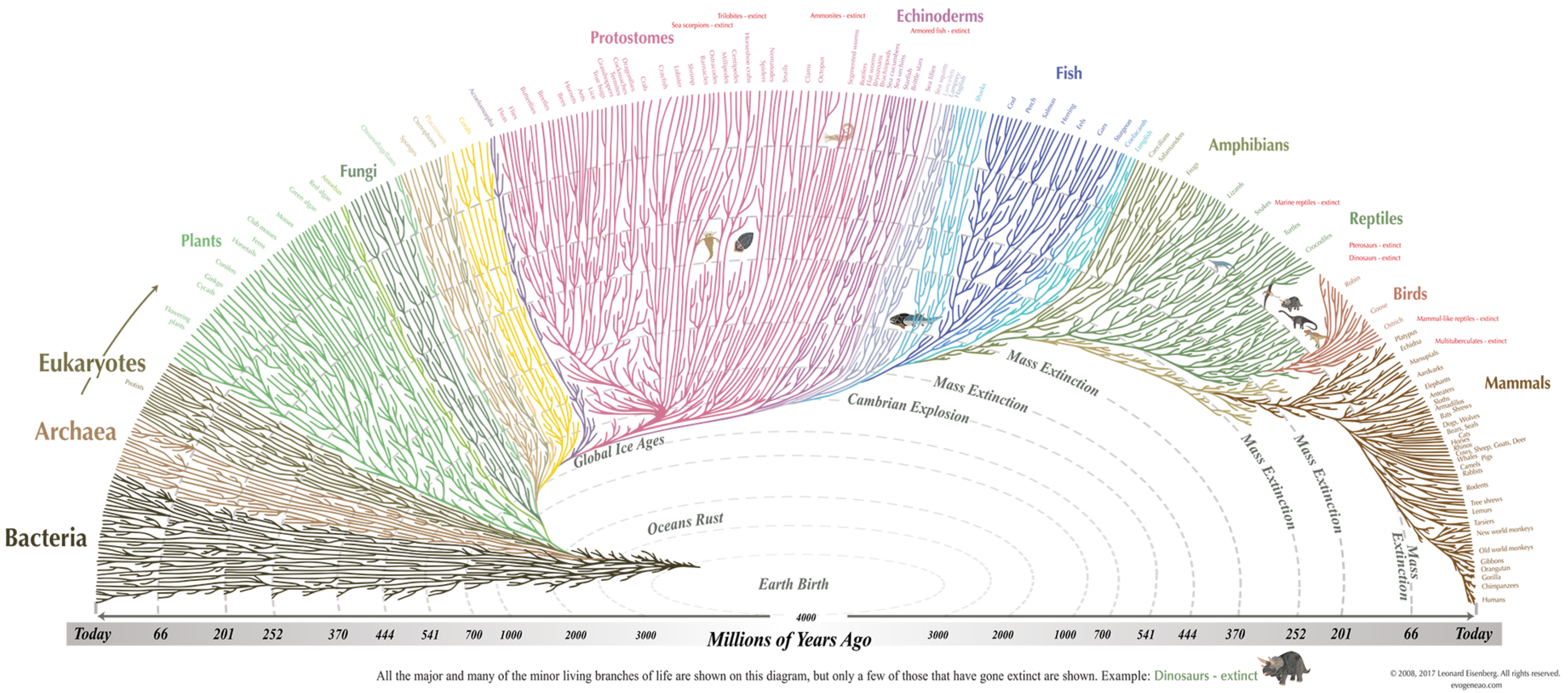
... represent the evolutionary history of a set of taxa.




Kingfishers (German: *Eisvögel*)
by McCullough et al. (2016)

# Phylogenetic Trees

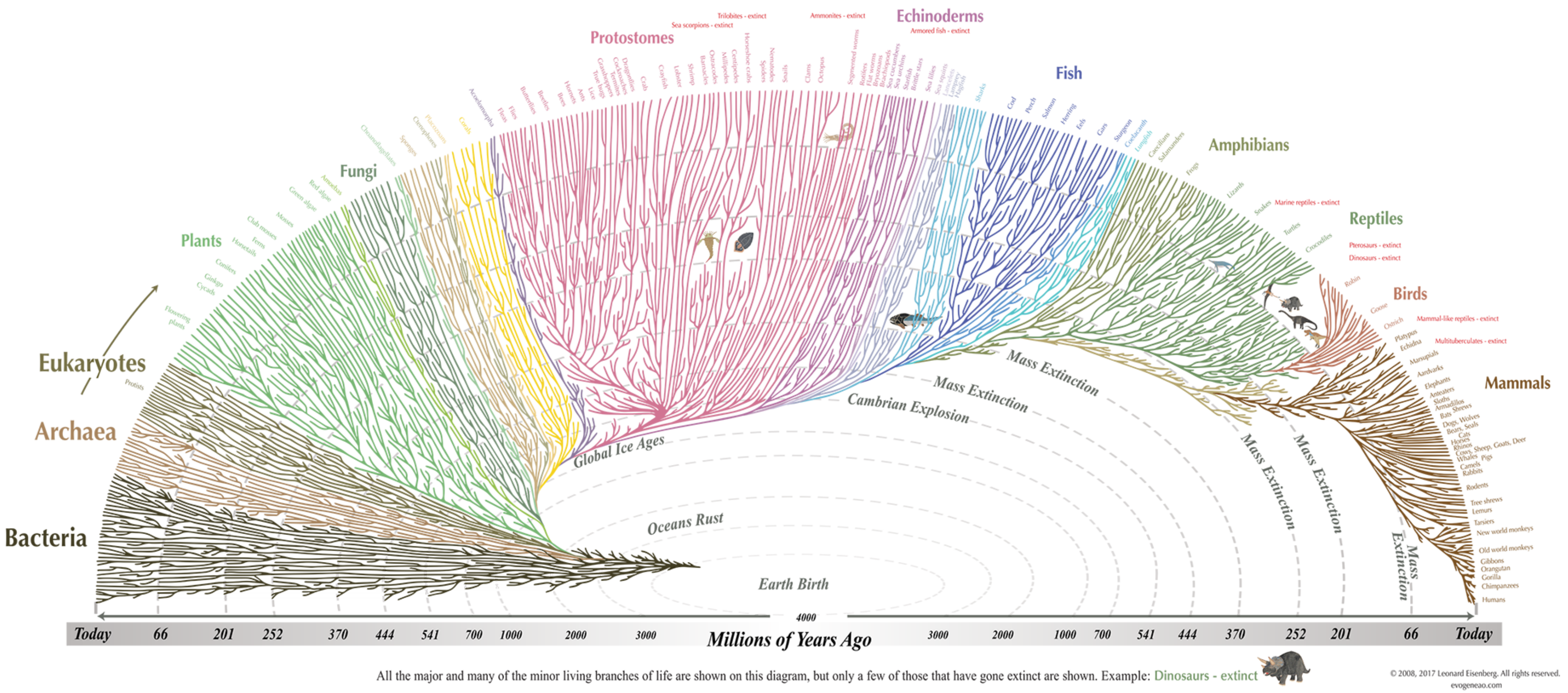
... represent the evolutionary history of a set of taxa.



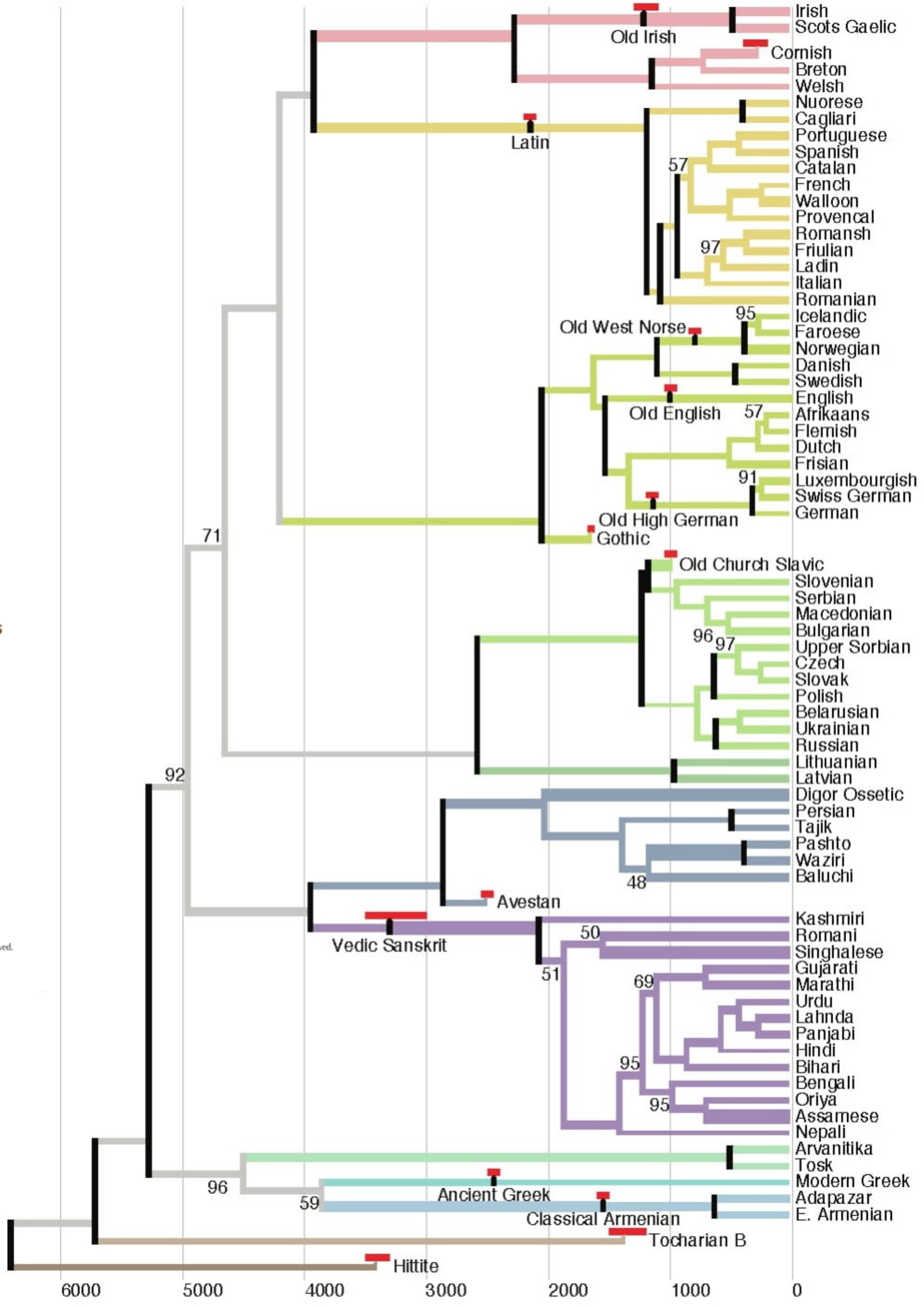

Tree of Life
www.evogeneao.com
(2017)

# Phylogenetic Trees

… represent the evolutionary history of a set of taxa.



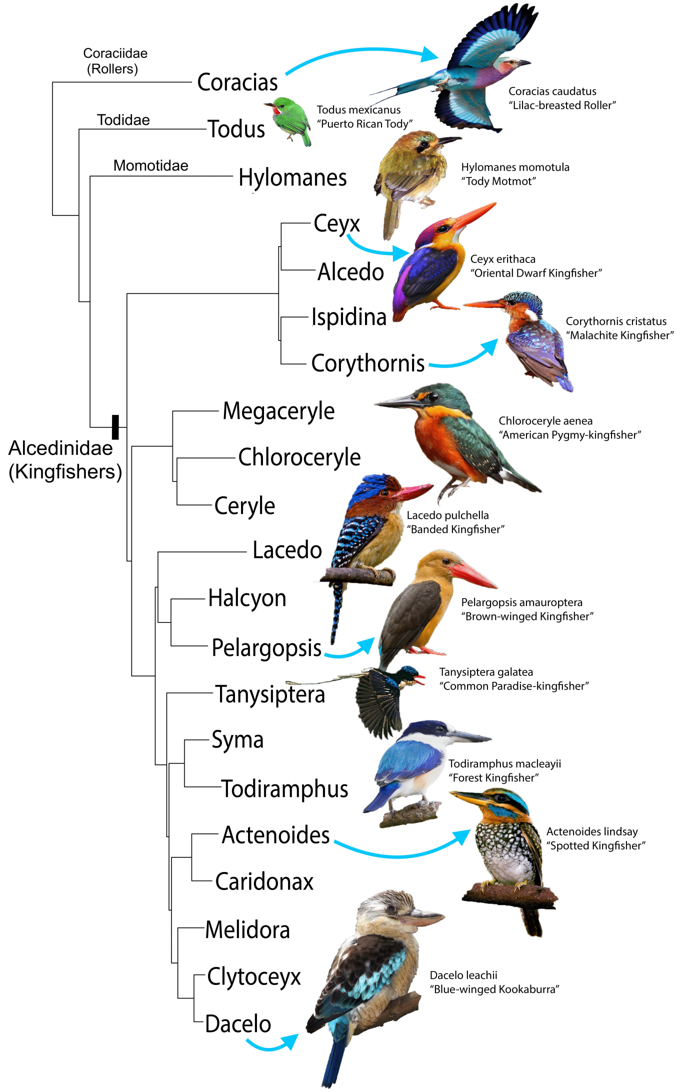

Tree of Life
www.evogeneao.com
(2017)




Phylogenetic tree of the Indo-European languages by Chang & Chundra (2015)

# Phylogenetic Trees

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

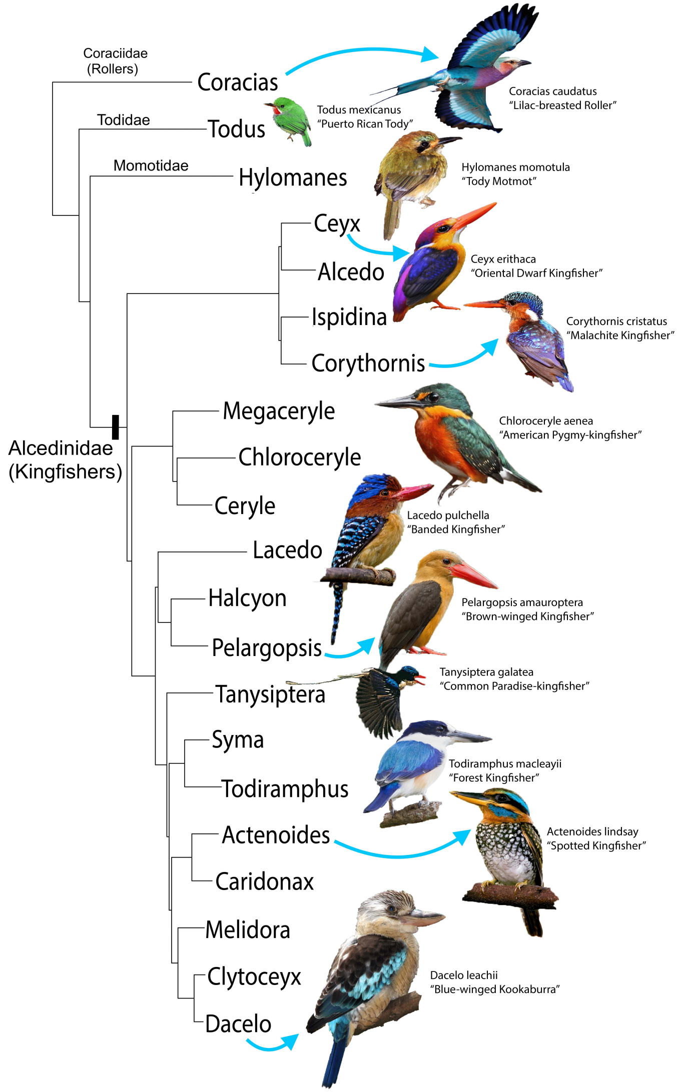



Kingfishers (German: *Eisvögel*)
by McCullough et al. (2016)

**Properties** (in the biological sense):

# Phylogenetic Trees

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



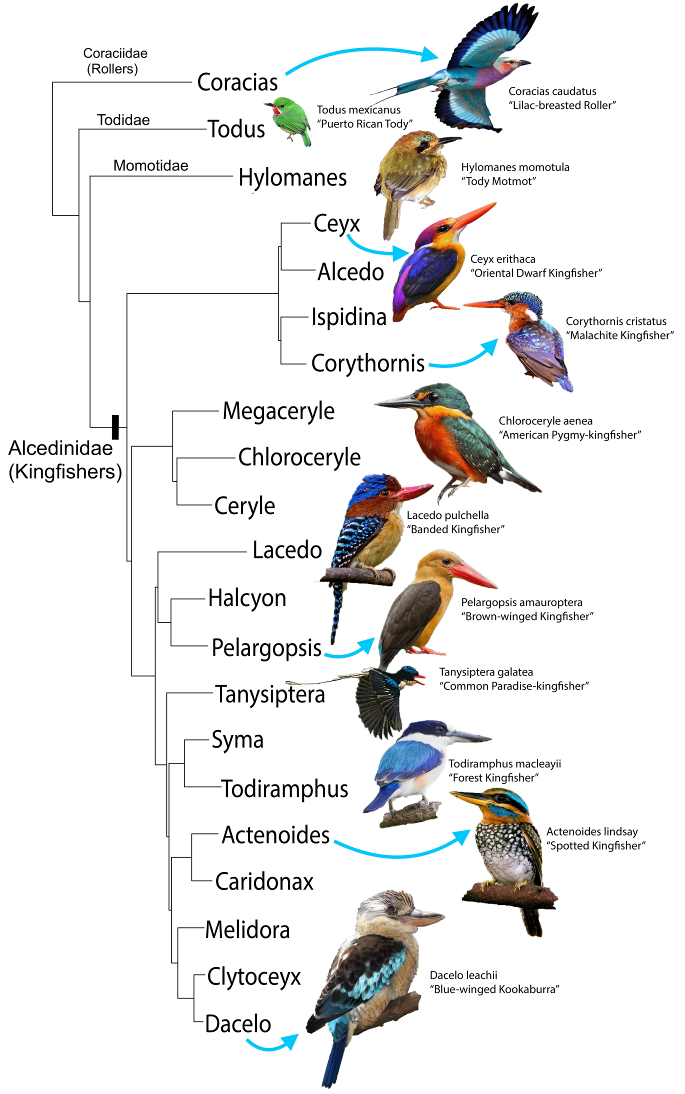

Kingfishers (German: *Eisvögel*)
by McCullough et al. (2016)

**Properties** (in the biological sense):

- 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.



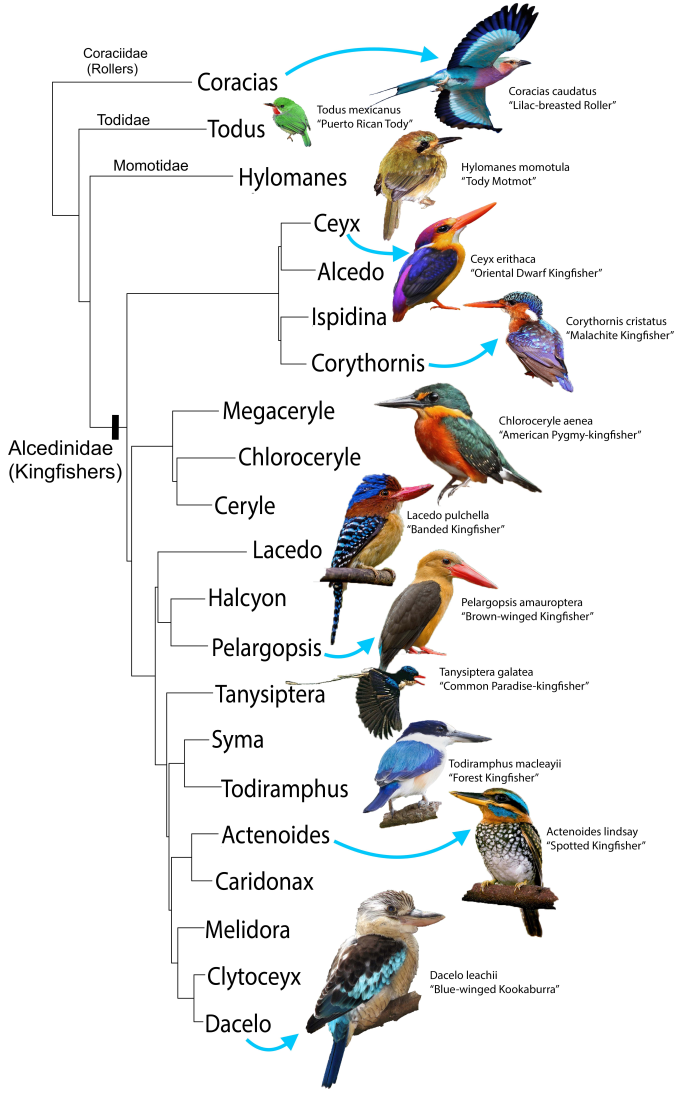

Kingfishers (German: *Eisvögel*) by McCullough et al. (2016)

**Properties** (in the biological sense):

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

# Phylogenetic Trees

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



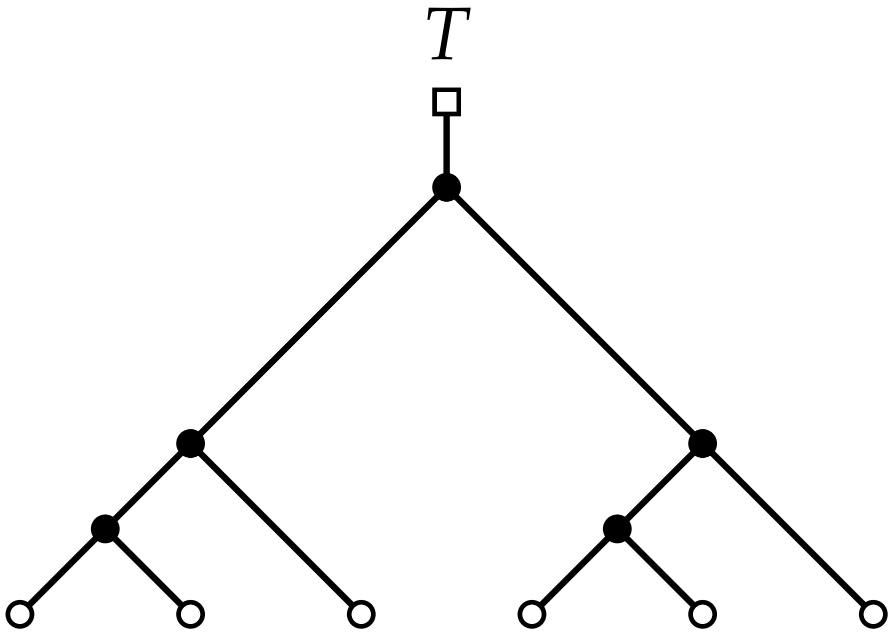

Kingfishers (German: *Eisvögel*)
by McCullough et al. (2016)

**Properties** (in the biological sense):

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

# Phylogenetic Trees

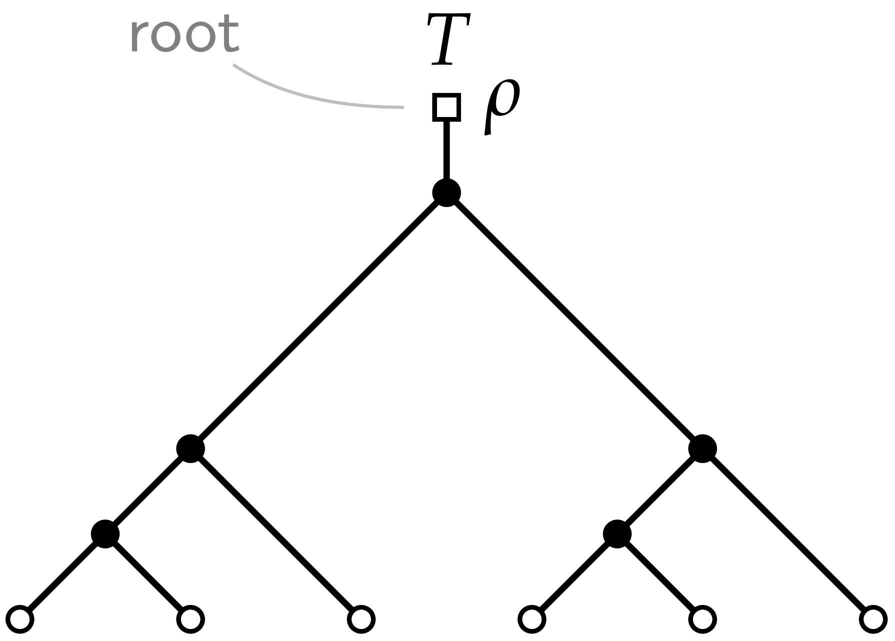
Let $X = \{1, 2, 3, \ldots, n\}$.
A **(rooted, binary) phylogenetic tree** $T$
is a rooted tree with the following
properties:



# Phylogenetic Trees

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

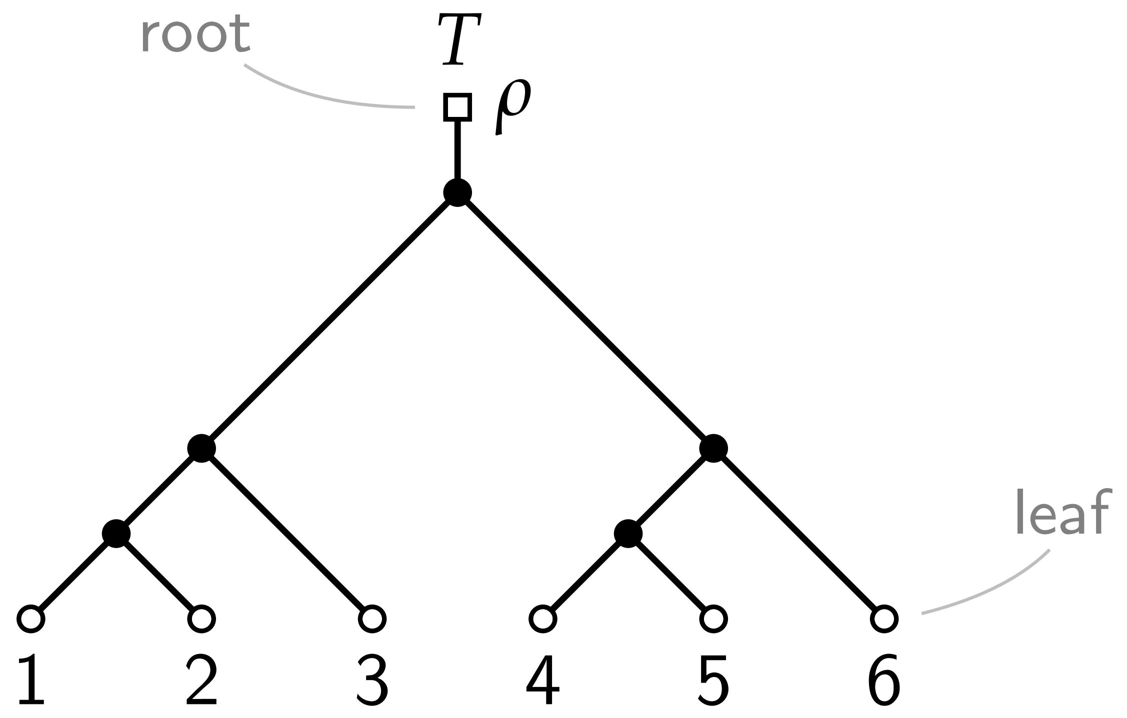
- The unique **root** is labeled $\rho$ and has outdegree 1.



# Phylogenetic Trees

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

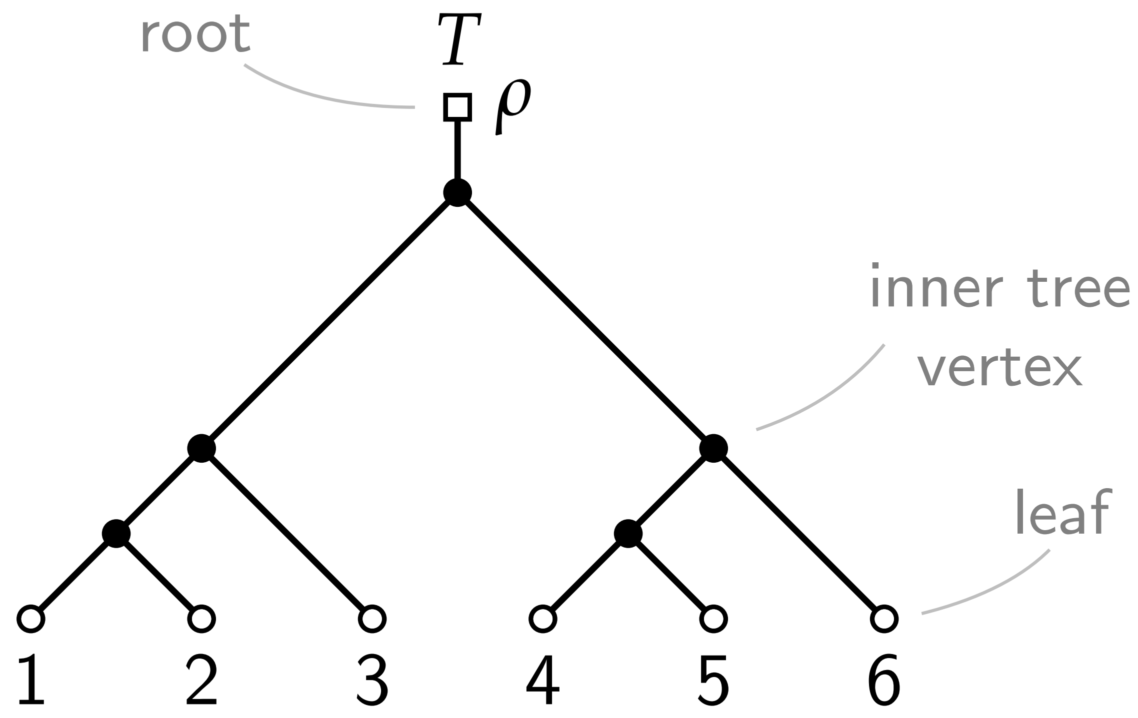
- The unique **root** is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.



# 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 $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
- All other vertices have indegree 1 and outdegree 2 (i.e., it is a *binary* tree).



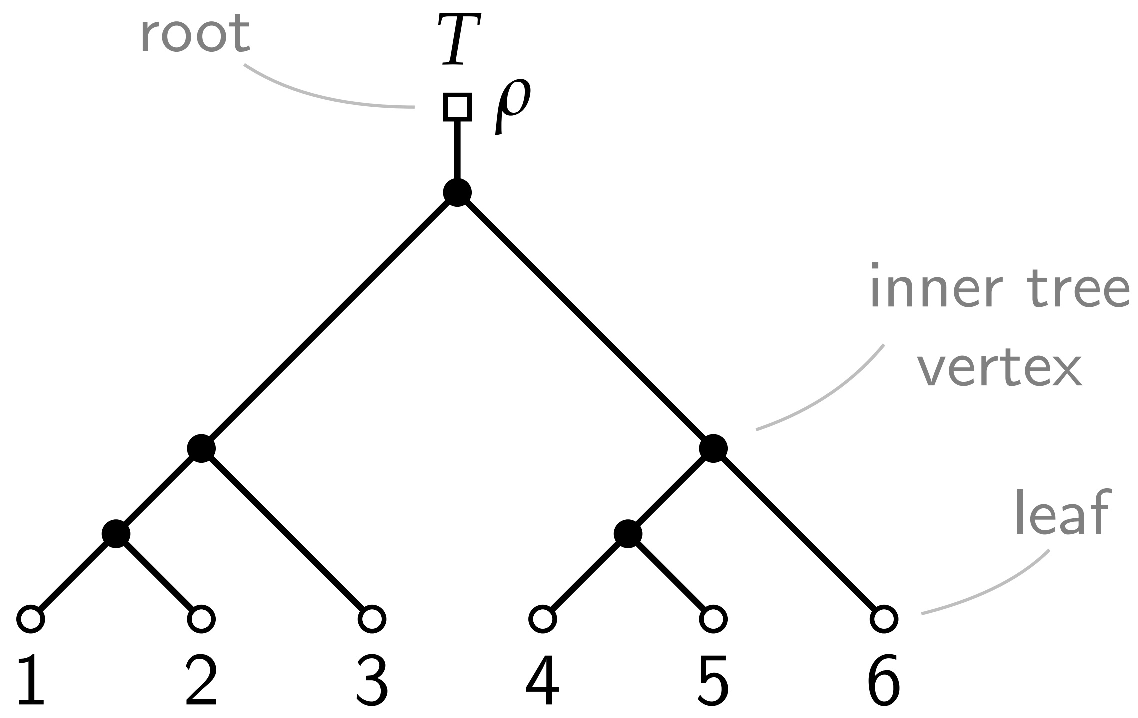
# Phylogenetic Trees

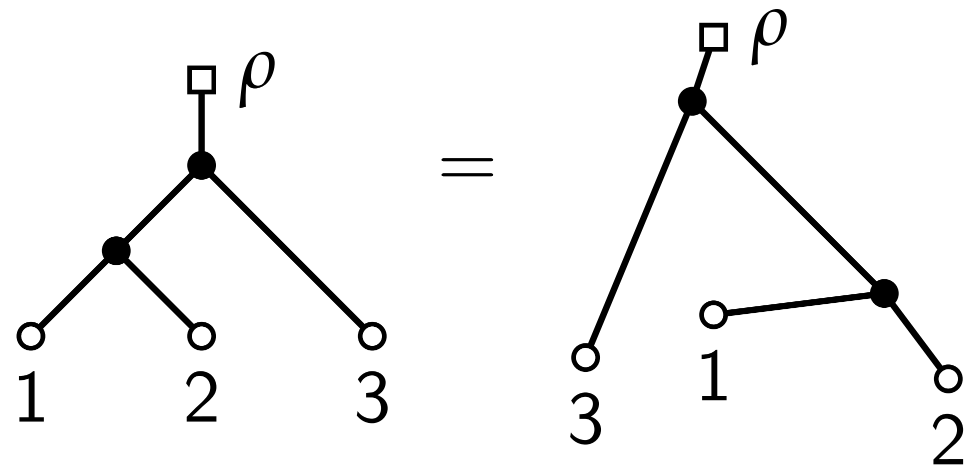
Let $X = \{1, 2, 3, \ldots, n\}$.
A **(rooted, binary) phylogenetic tree** $T$ is a rooted tree with the following properties:

- The unique **root** is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
- All other vertices have indegree 1 and outdegree 2 (i.e., it is a *binary* tree).

**Remarks.** Here, in our definition

- vertices have **no heights** and
- the order of the children of a vertex does not matter.

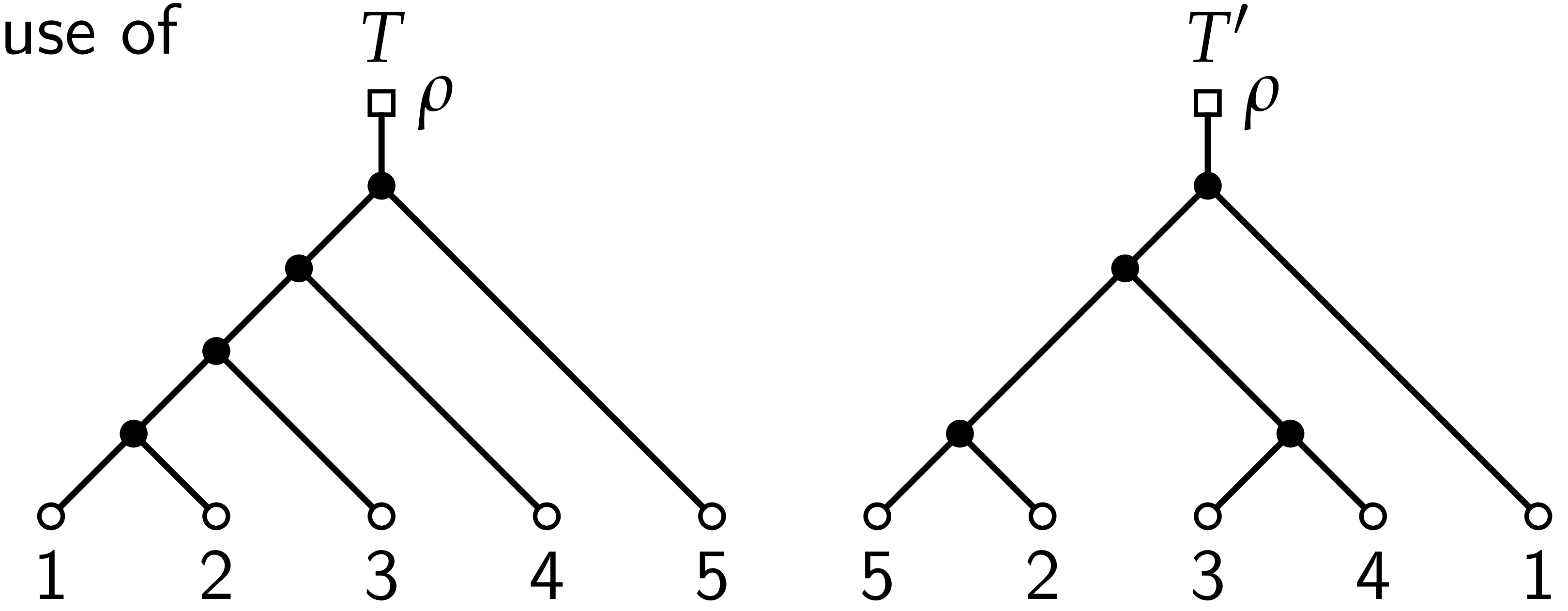


# Problem

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

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

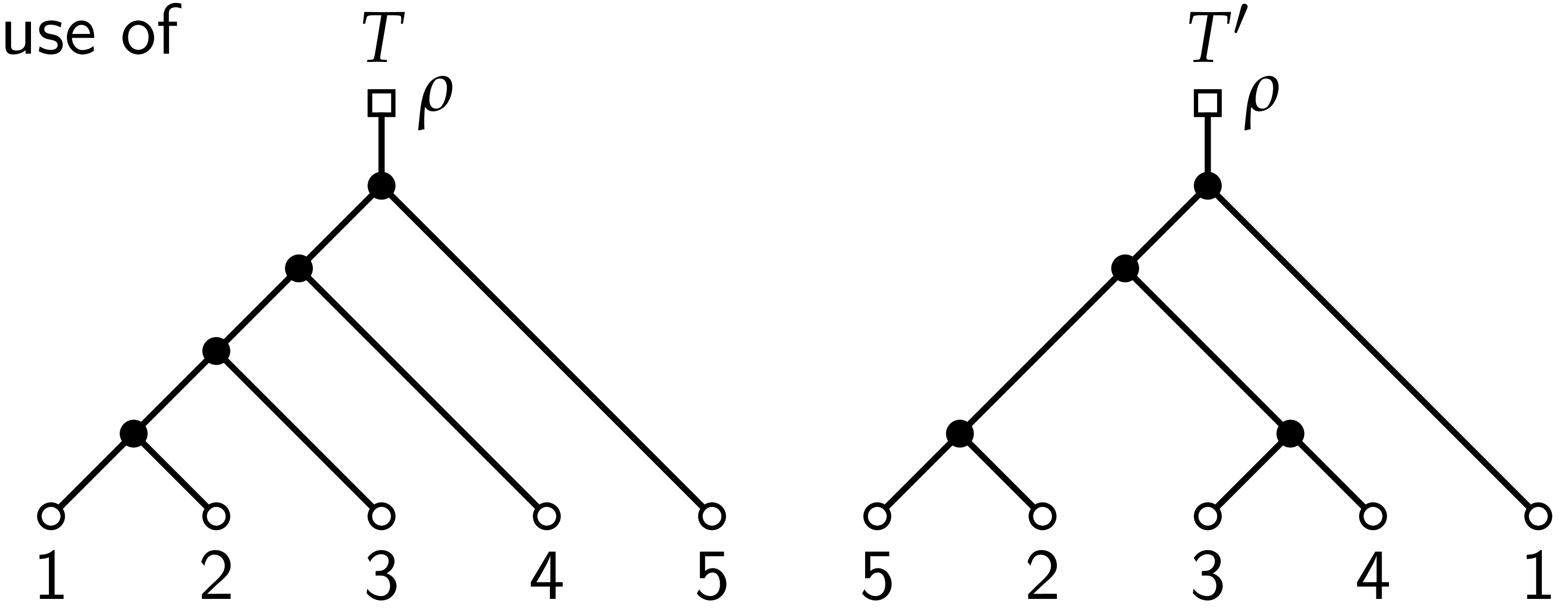


# Problem

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

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

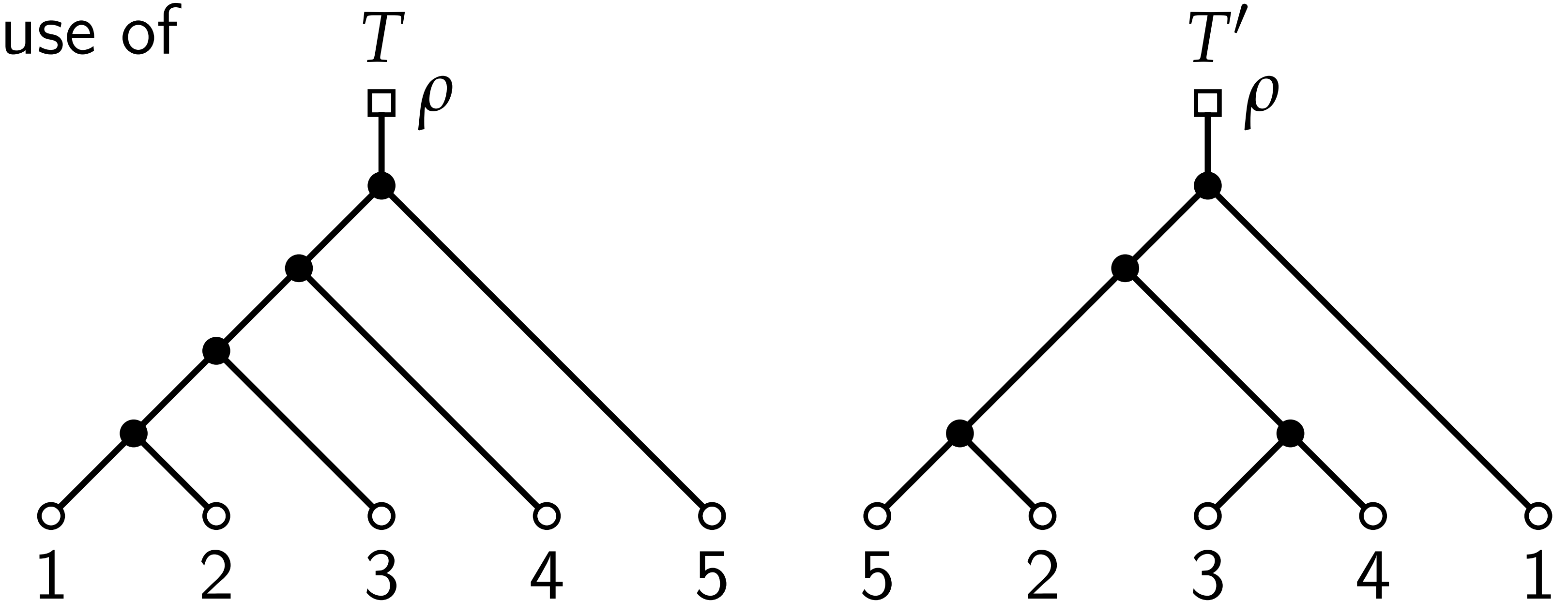
We want to be able to **compare** different phylogenetic trees.
How?



# 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** that specifies how similar two phylogenetic trees on the same set $X$ are and devise algorithms to compute it.

# Problem

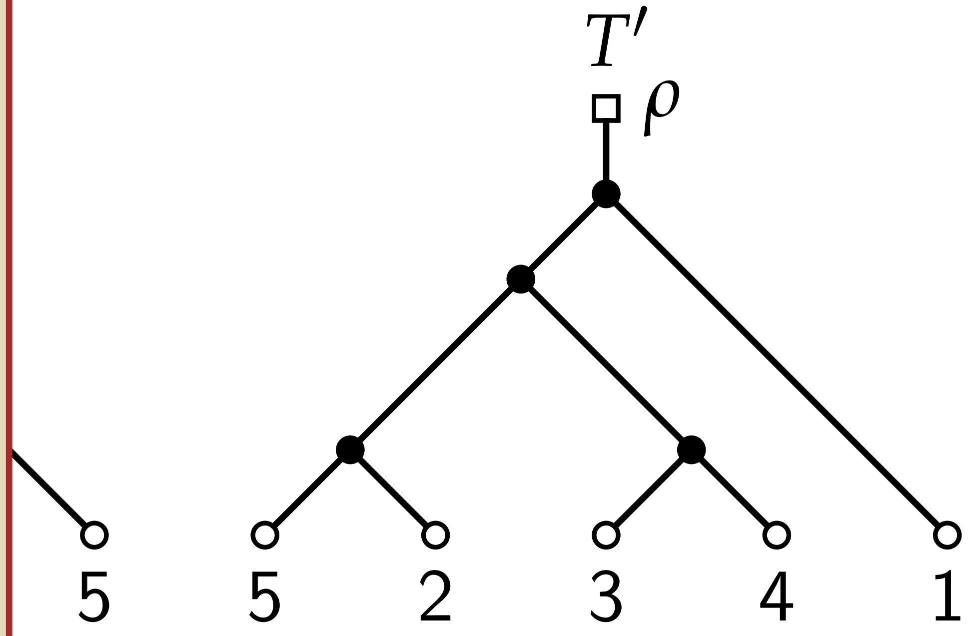
For the sar
phylogenet

- differen
- differen
- differen

We want to
different ph
How?

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x,x) = 0$ (no distance to itself)
- $d(x,y) > 0$ for $x \neq y$ (positive)
- $d(x,y) = d(y,x)$ (symmetric)
- $d(x,z) \leq d(x,y) + d(y,z)$ (triangle inequality holds)



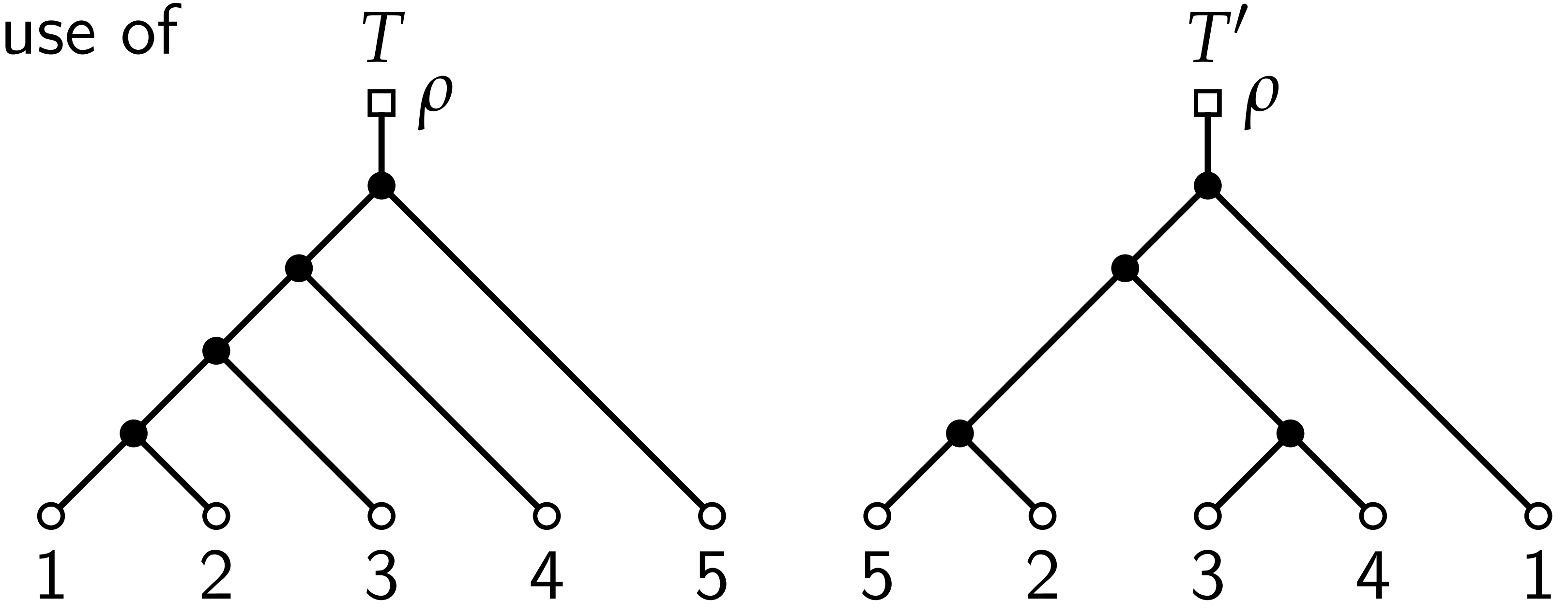

**Goal.**
Define a **metric** that specifies how similar two phylogenetic trees on the same set $X$ are and devise algorithms to compute it.

# 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?

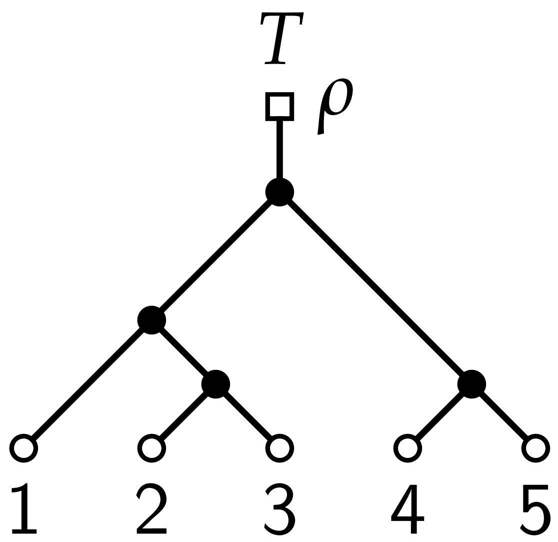



**Goal.**
Define a **metric** that specifies how similar two phylogenetic trees on the same set $X$ are and devise algorithms to compute it.

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

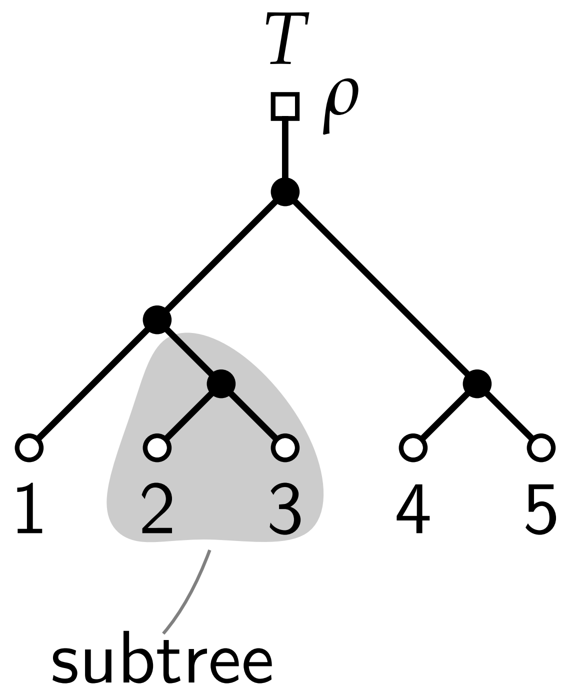
# Subtree Prune & Regraft (SPR)

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



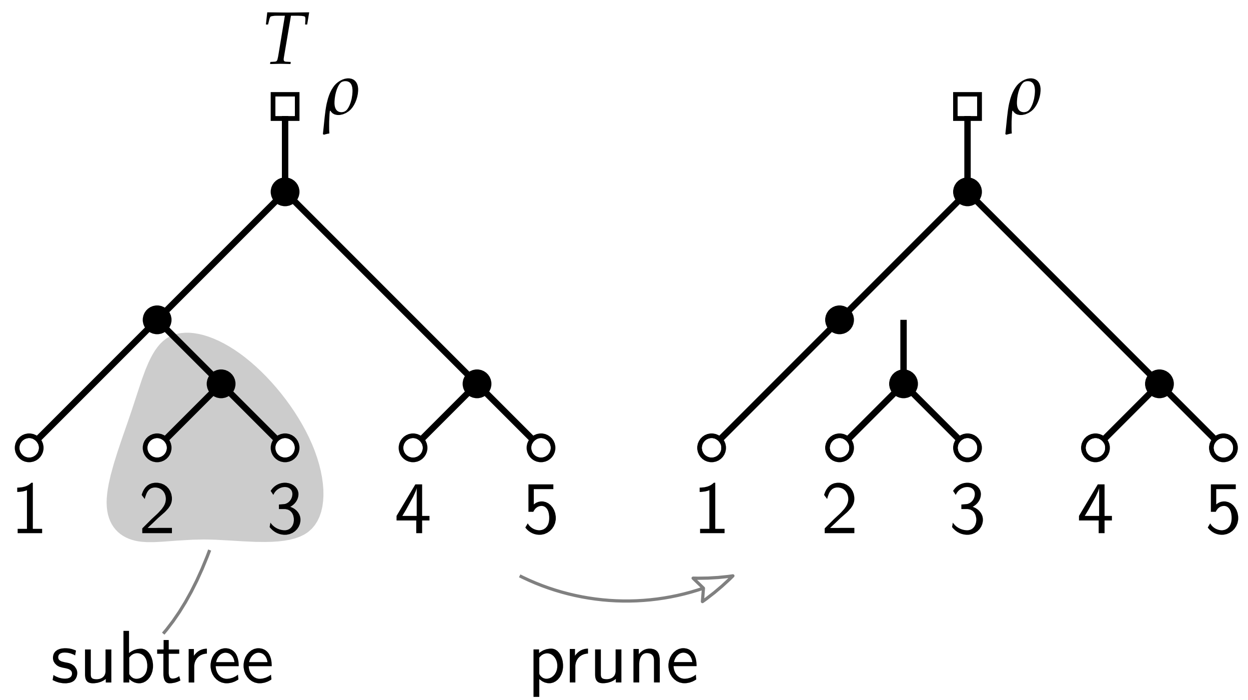
# Subtree Prune & Regraft (SPR)

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



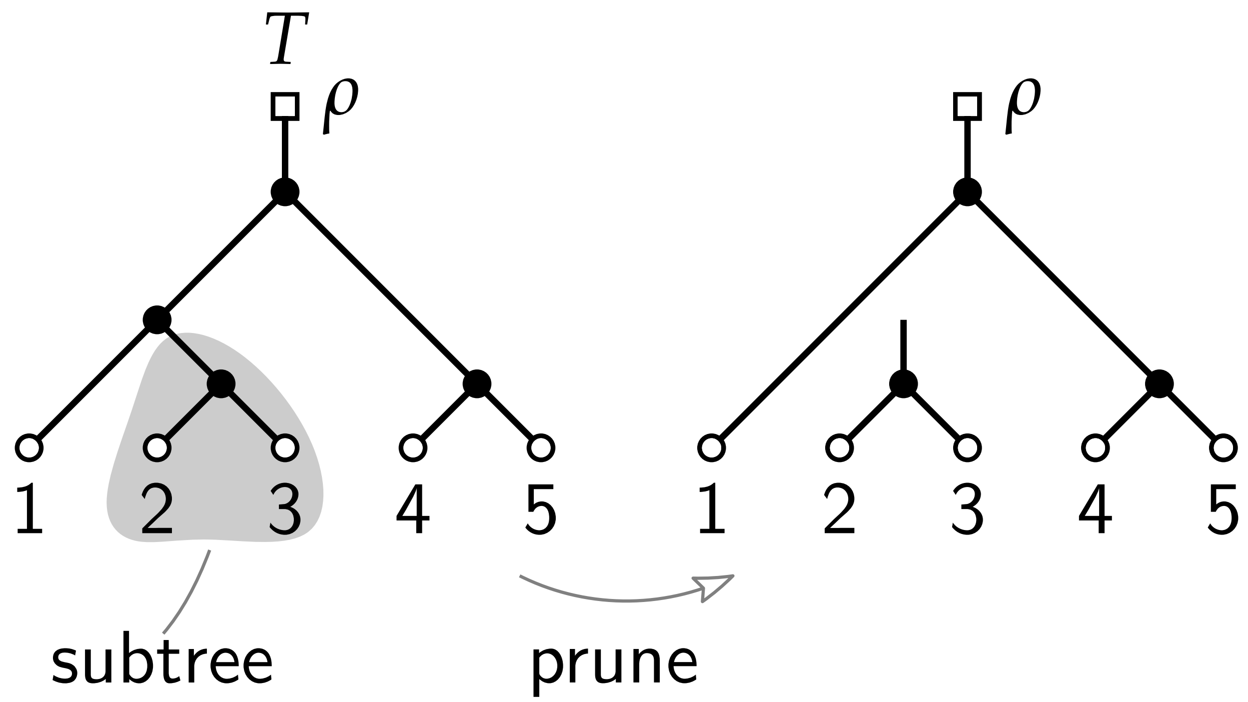
# Subtree Prune & Regraft (SPR)

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



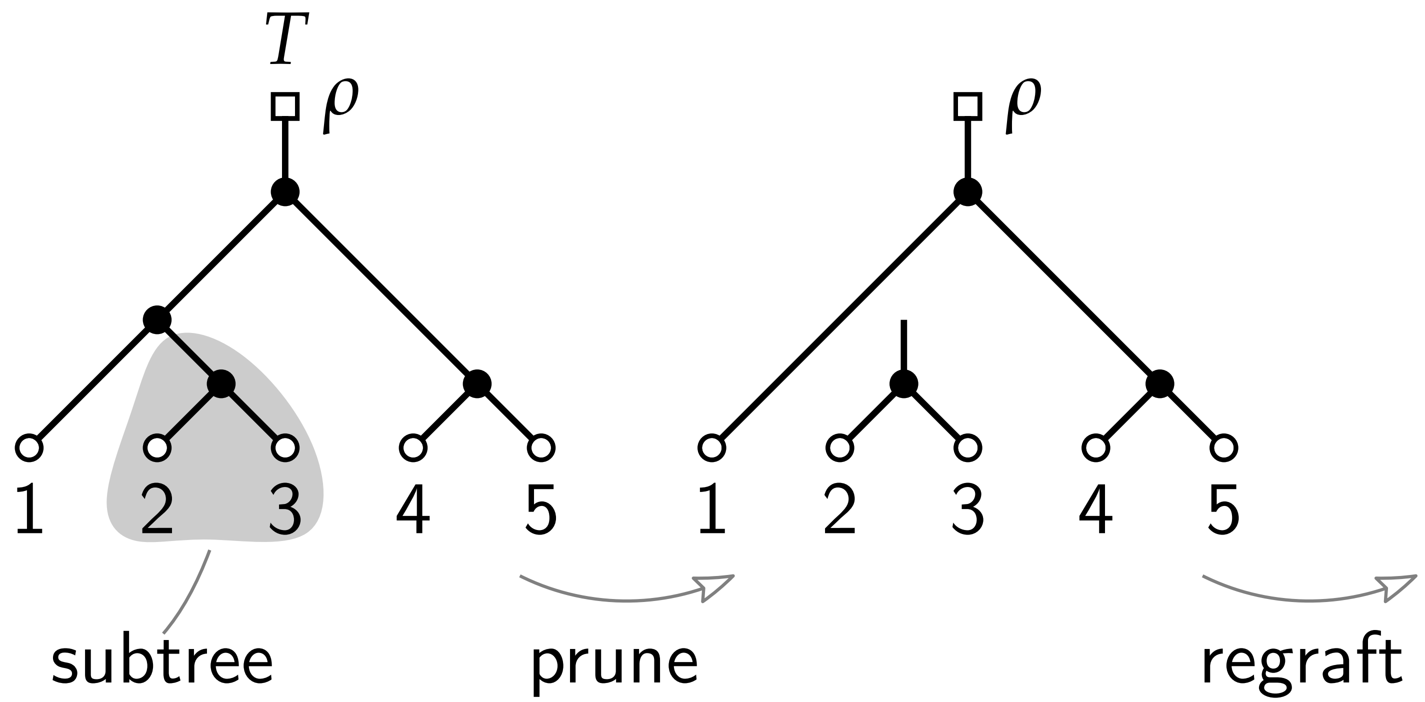
# Subtree Prune & Regraft (SPR)

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



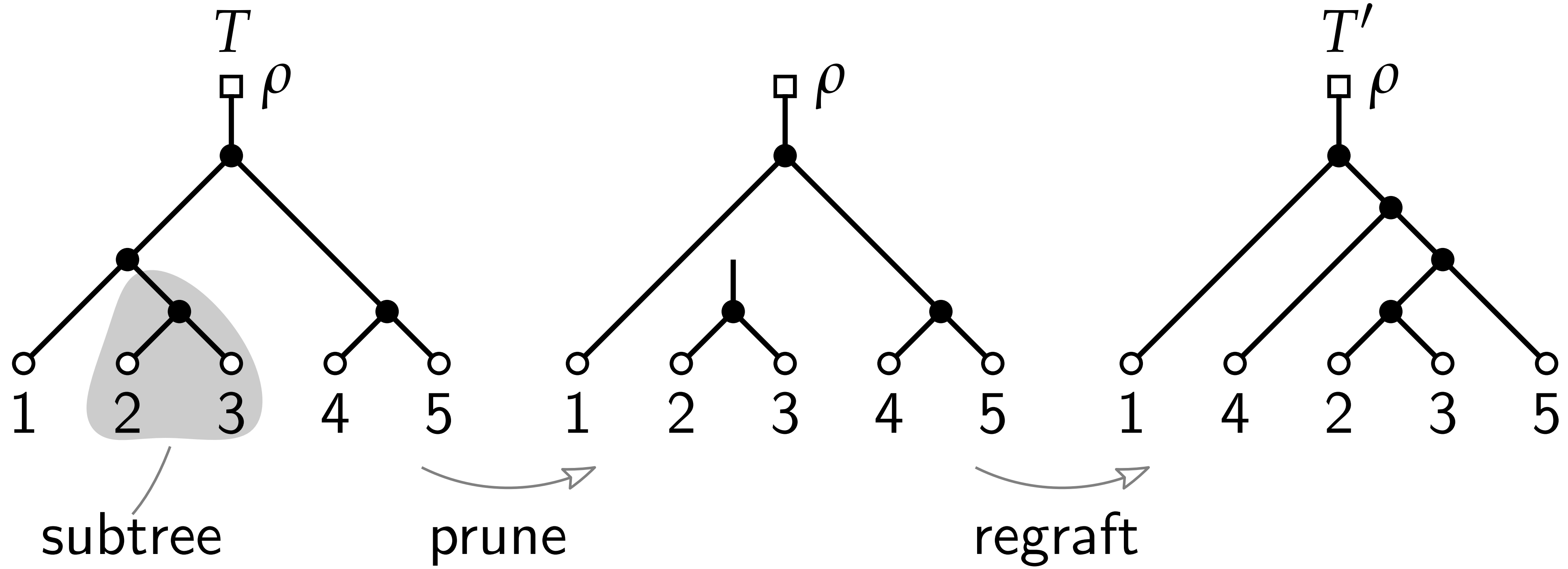
# Subtree Prune & Regraft (SPR)

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



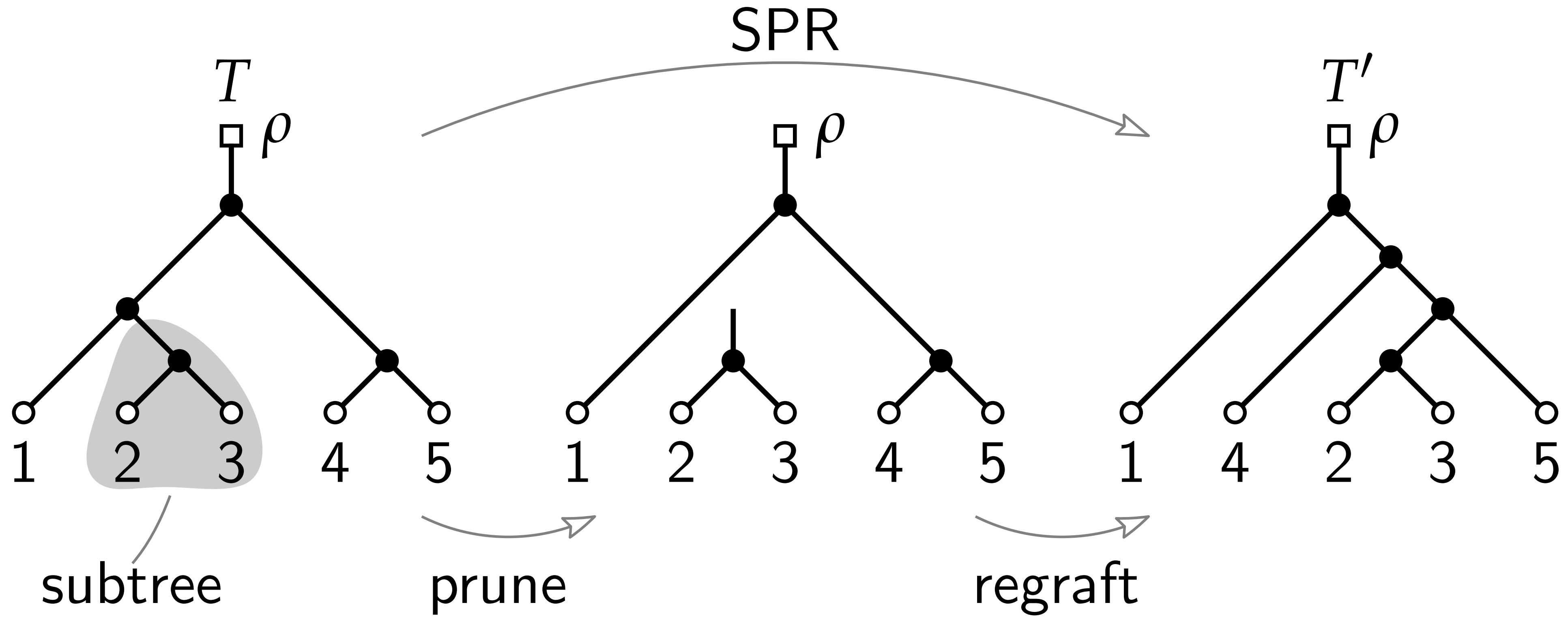
# Subtree Prune & Regraft (SPR)

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



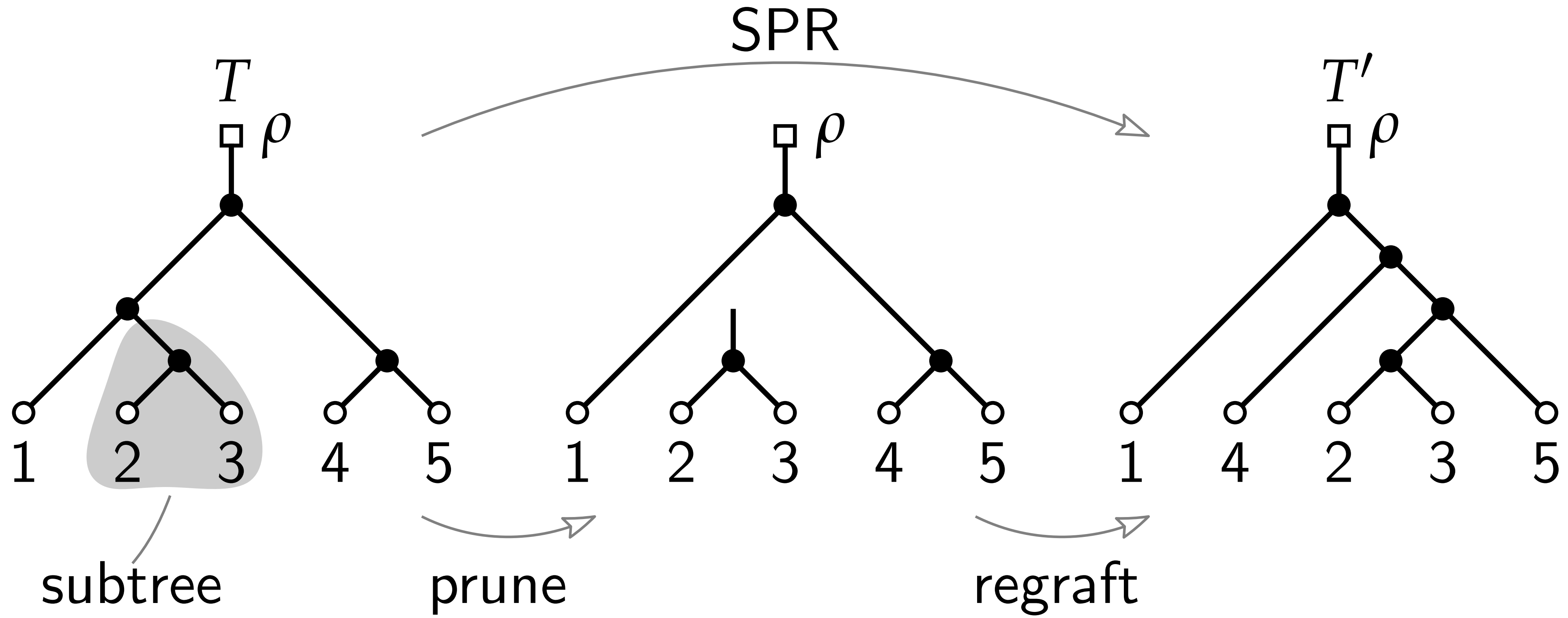
# Subtree Prune & Regraft (SPR)

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.

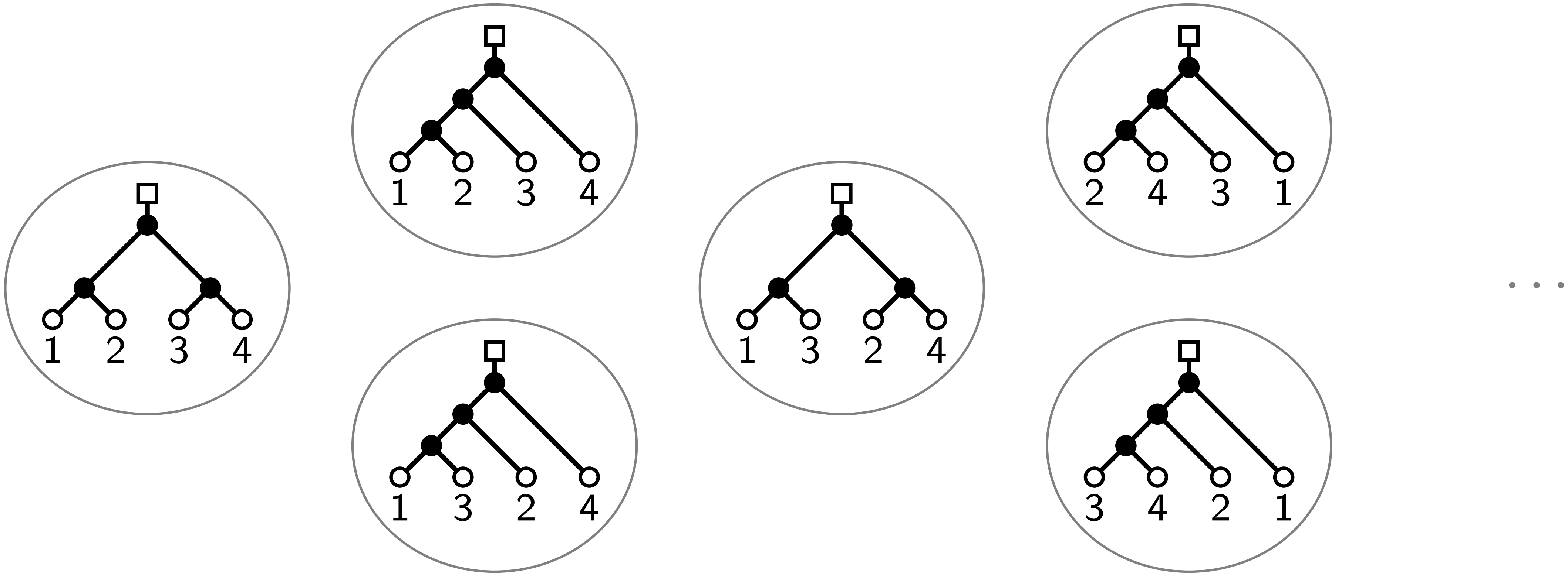
# SPR-Graph

The SPR operations induce the **SPR-graph** $G = (V, E)$ for a set $X$:

# SPR-Graph

The SPR operations induce the **SPR-graph** $G = (V, E)$ for a set $X$:

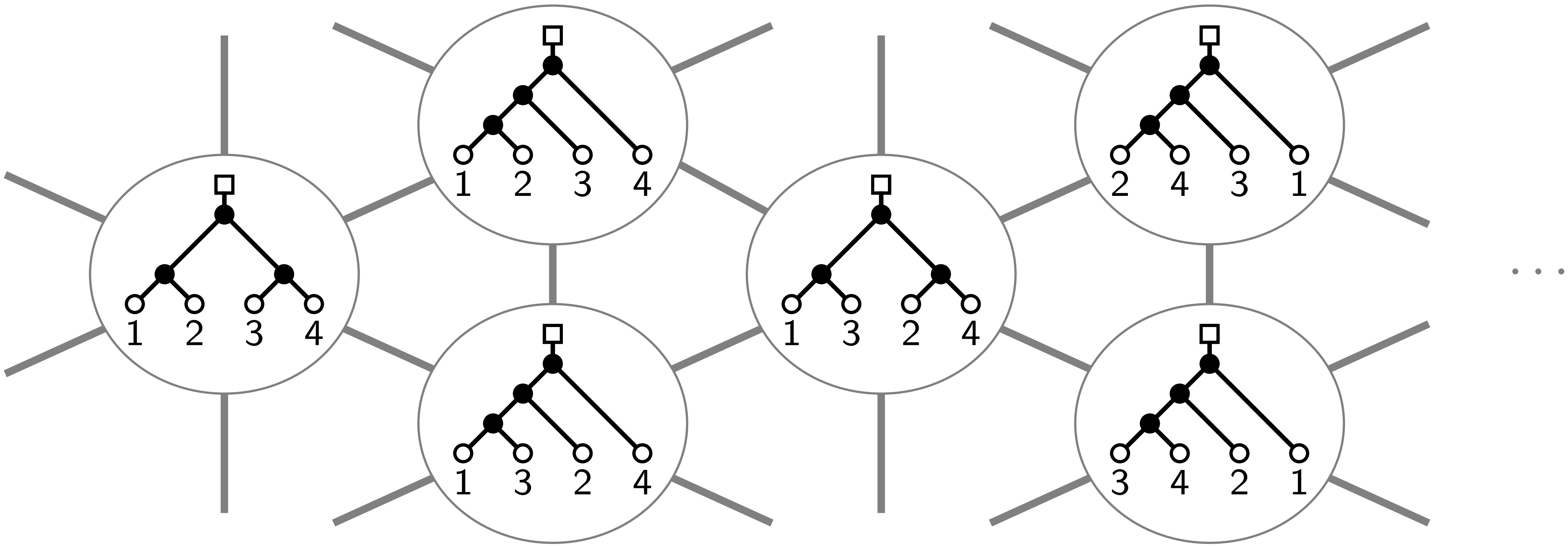
- $V = \{T \mid T$ is a phylogenetic tree on X$\}$



# SPR-Graph

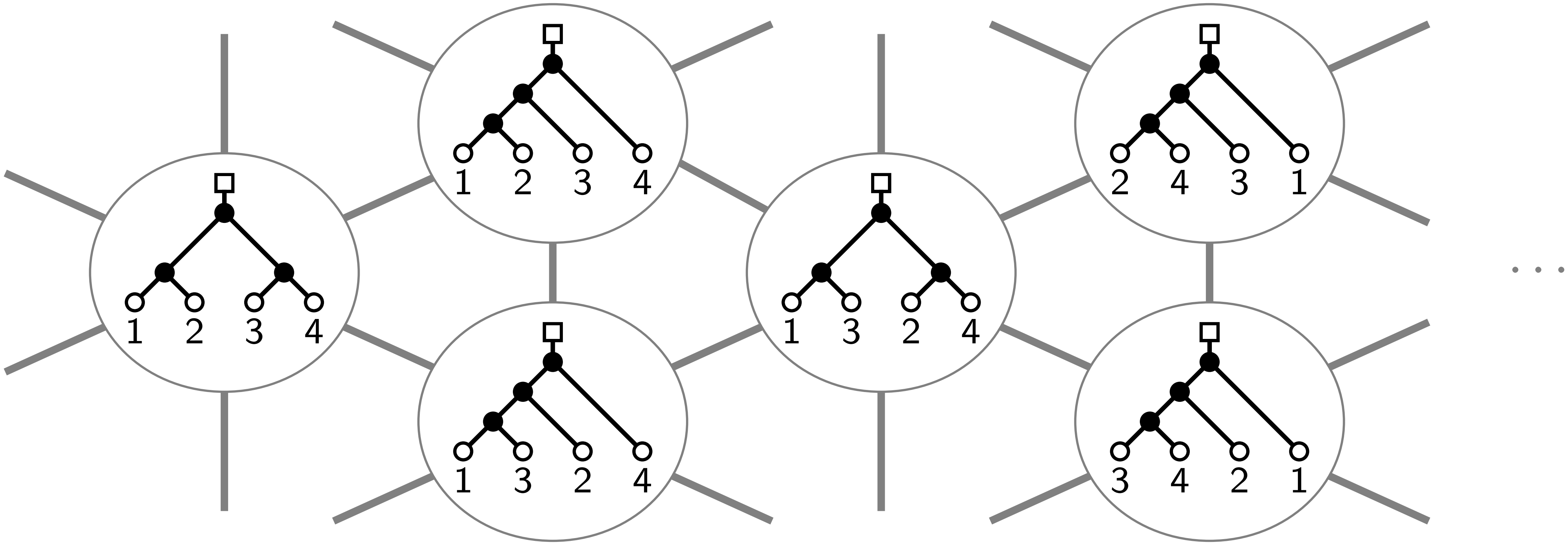
The SPR operations induce the **SPR-graph** $G = (V, E)$ for a set $X$:

- $V = \{T \mid T$ is a phylogenetic tree on X$\}$
- $E = \{\{T, T'\} \mid T$ 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** $\mathrm{d_{SPR}}(T, T')$ 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.

# SPR-Distance

The **SPR-distance** $\mathsf{d_{SPR}}(T, T')$ 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** exercise

# SPR-Distance

The **SPR-distance** $d_{SPR}(T, T')$ 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** exercise

**Lemma 2.**
The SPR-distance is a metric.

# 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.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

# 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.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

# 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.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself) ✓ trivial
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

# SPR-Distance

The **SPR-distance** $\mathrm{d}_{\mathrm{SPR}}(T, T')$ 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.** $G$ is connected and undirected.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself) ✓ trivial
- $d(x, y) > 0$ for $x \neq y$ (positive) ✓ shortest path exists because $G$ is connected
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

# 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.

**Lemma 2.**
The SPR-distance is a metric.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself) ✓ trivial
- $d(x, y) > 0$ for $x \neq y$ (positive) ✓ shortest path exists because $G$ is connected
- $d(x, y) = d(y, x)$ (symmetric) ✓ all paths can be reversed bc. $G$ is undirected
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

**Proof.** $G$ is connected and undirected.

# SPR-Distance

The **SPR-distance** $\mathrm{d}_{\mathrm{SPR}}(T, T')$ 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.** $G$ is connected and undirected.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself) ✓ trivial
- $d(x, y) > 0$ for $x \neq y$ (positive) ✓ shortest path exists because $G$ is connected
- $d(x, y) = d(y, x)$ (symmetric) ✓ all paths can be reversed bc. $G$ is undirected
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds) ✓ the triangle inequality holds because we can compose the path $x \rightsquigarrow z$ by $x \rightsquigarrow y \rightsquigarrow z$

# SPR-Distance

The **SPR-distance** $d_{SPR}(T, T')$ 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.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.
All properties of a metric follow. □

**Defintion:**
A *metric* $d$ is a function of two parameters such that:

- $d(x,x) = 0$ (no distance to itself) ✓ trivial
- $d(x,y) > 0$ for $x \neq y$ (positive) ✓ shortest path exists because $G$ is connected
- $d(x,y) = d(y,x)$ (symmetric) ✓ all paths can be reversed bc. $G$ is undirected
- $d(x,z) \leq d(x,y) + d(y,z)$ (triangle inequality holds) ✓ the triangle inequality holds because we can compose the path $x \rightsquigarrow z$ by $x \rightsquigarrow y \rightsquigarrow z$

# SPR-Distance

The **SPR-distance** $d_{SPR}(T, T')$ 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** exercise

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

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.
All properties of a metric follow.

$\square$

# SPR-Distance

The **SPR-distance** $\mathrm{d}_{\mathsf{SPR}}(T, T')$ 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** exercise

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

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.
All properties of a metric follow. □

... but $G$ is huge!

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

# SPR-Distance

The **SPR-distance** $\mathrm{d}_{\mathrm{SPR}}(T, T')$ 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** exercise

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

**Lemma 2.**
The SPR-distance is a metric.

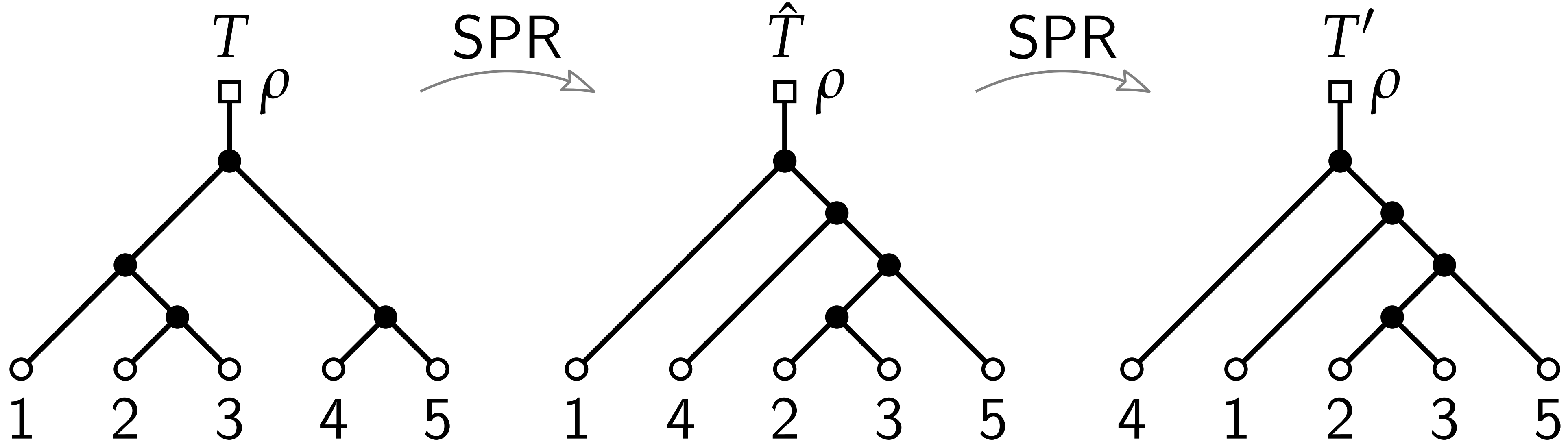
**Proof.** $G$ is connected and undirected.
All properties of a metric follow. □

... but $G$ is huge!

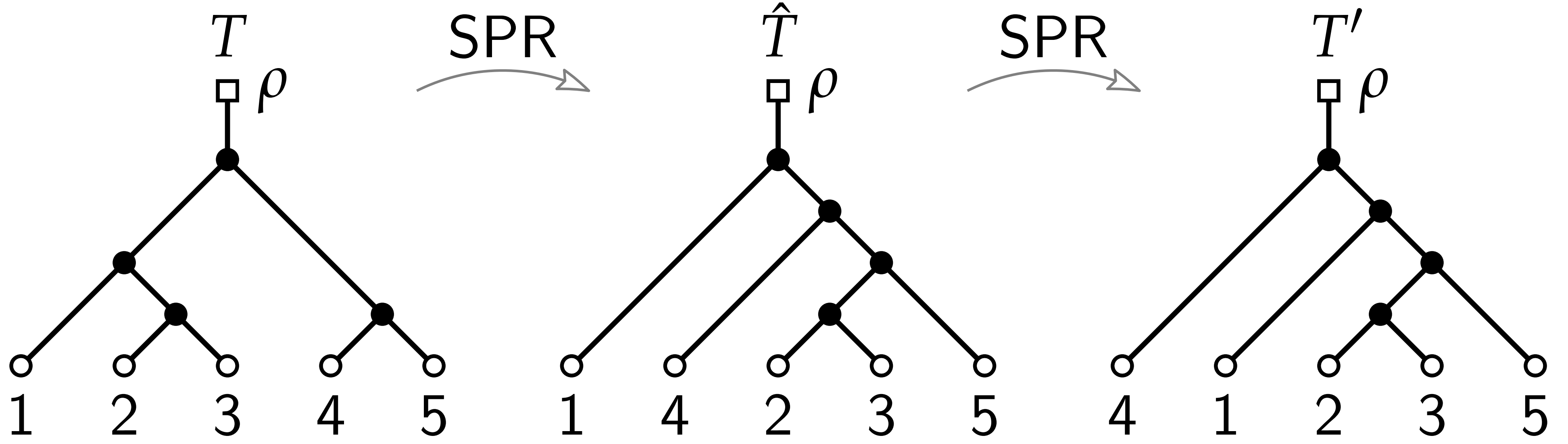
$$|V(G)| = (2n-3)!! = (2n-3) \cdot (2n-5) \cdot \ldots \cdot 5 \cdot 3$$

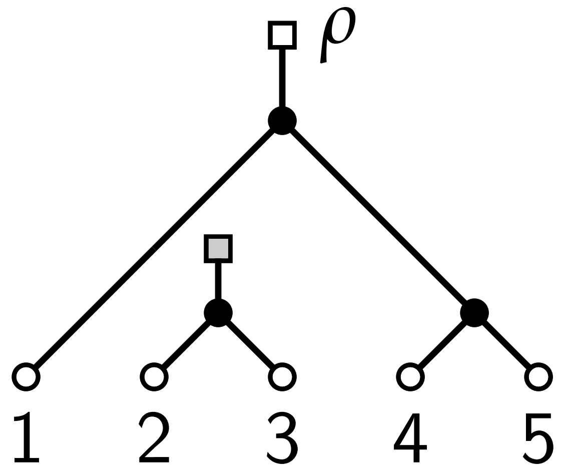
- Can we rephrase the problem?

# Maximum Agreement Forests

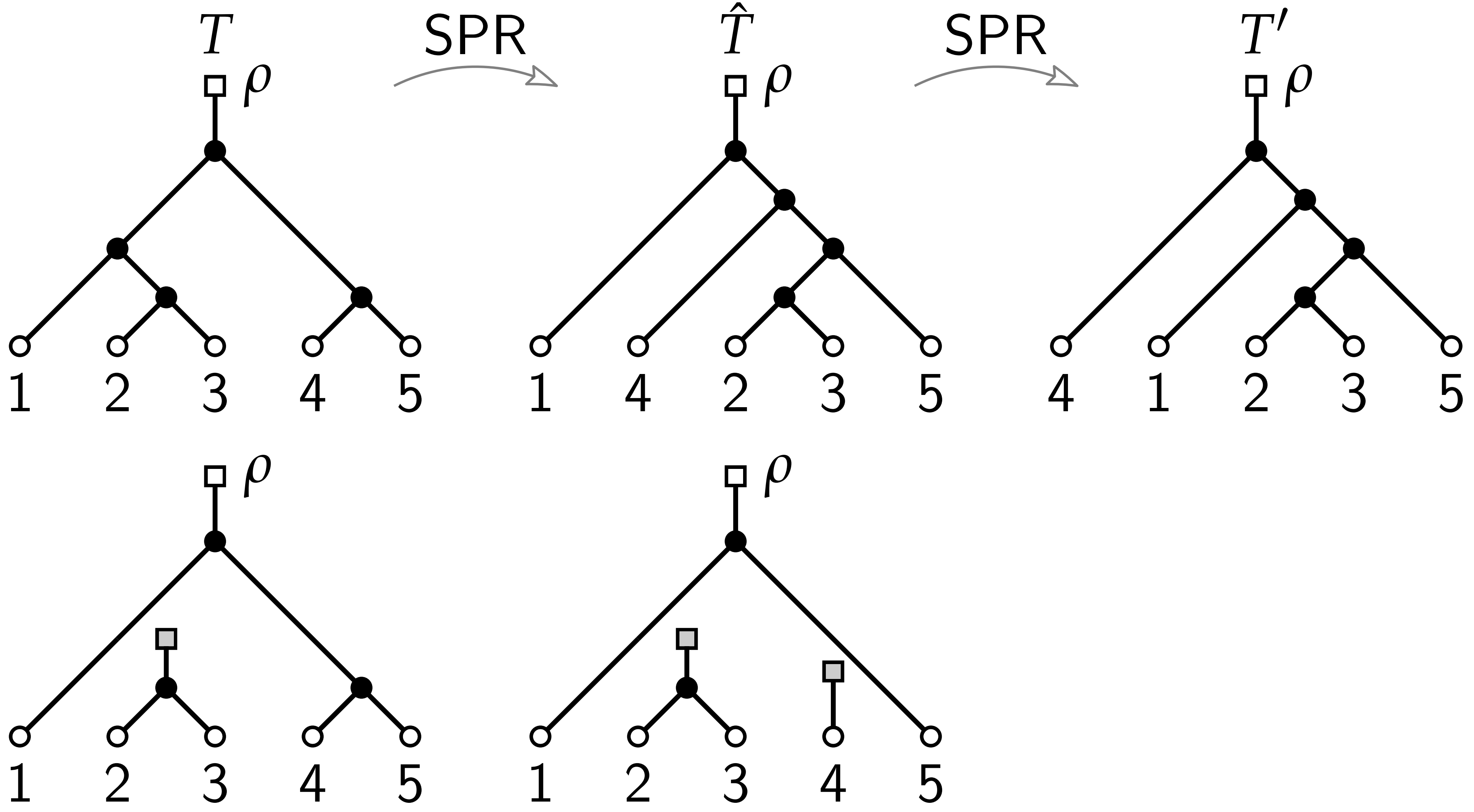


# Maximum Agreement Forests

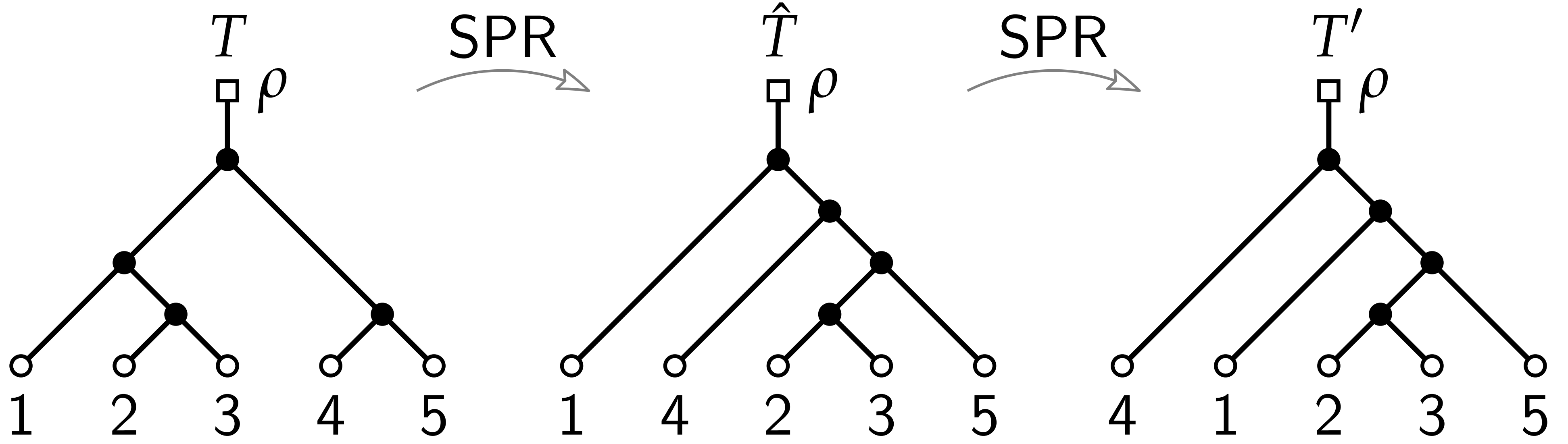


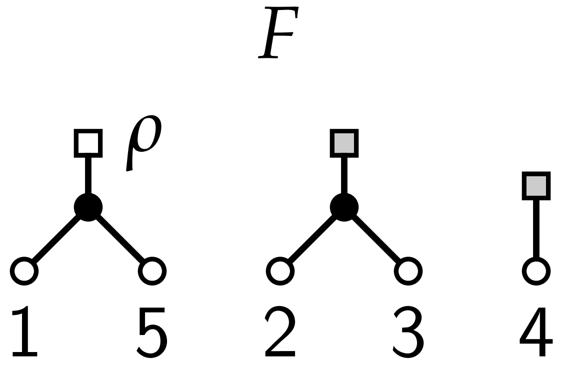



# Maximum Agreement Forests

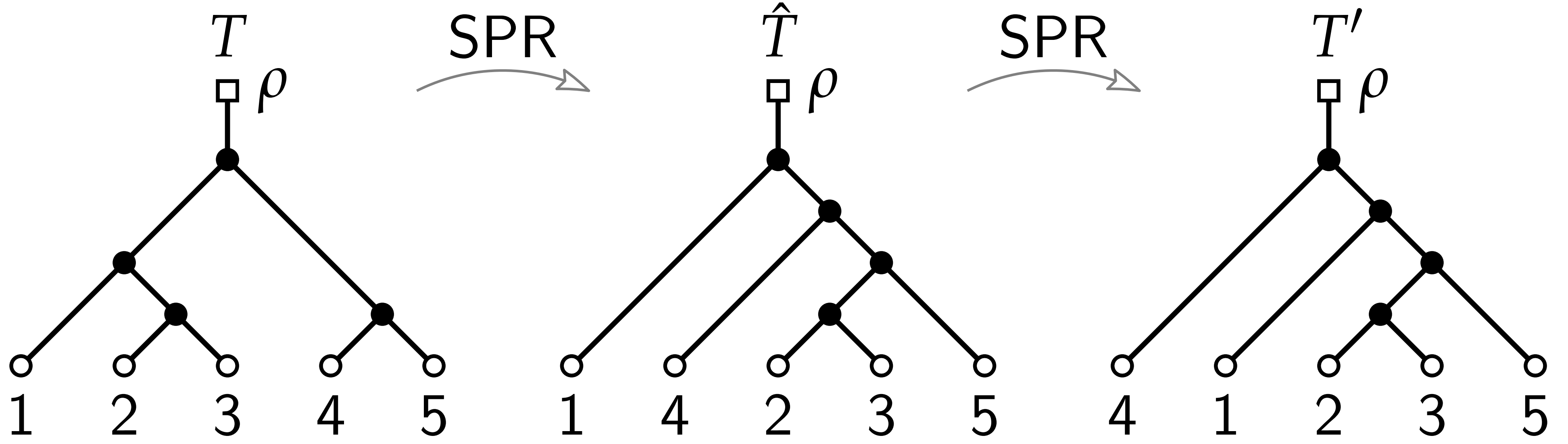


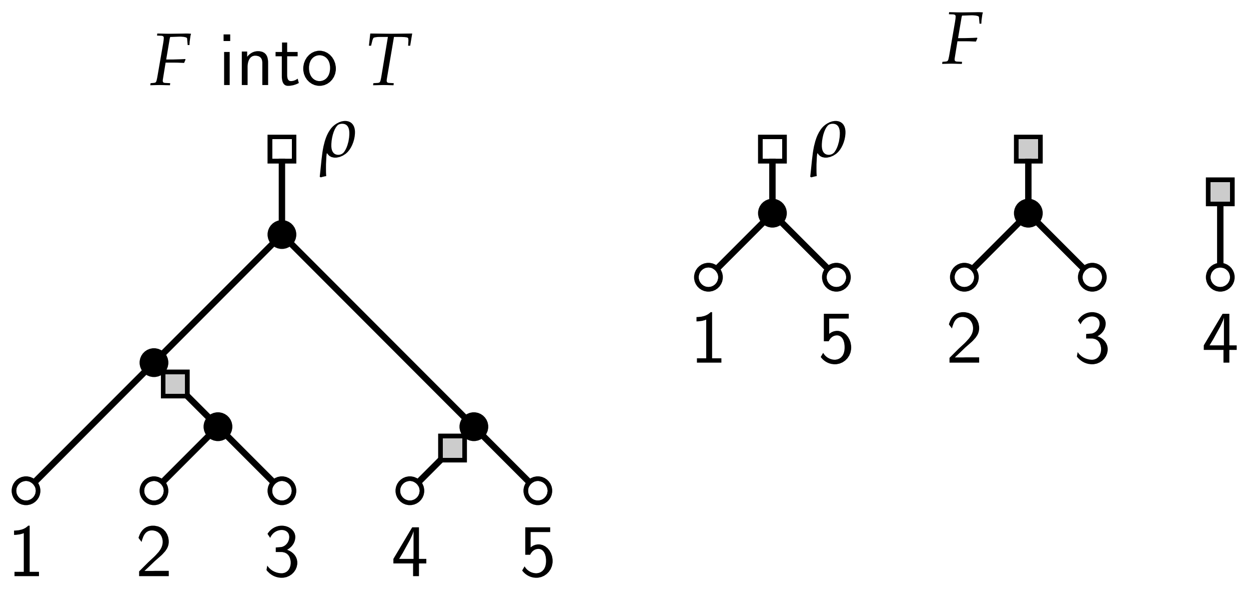
# Maximum Agreement Forests



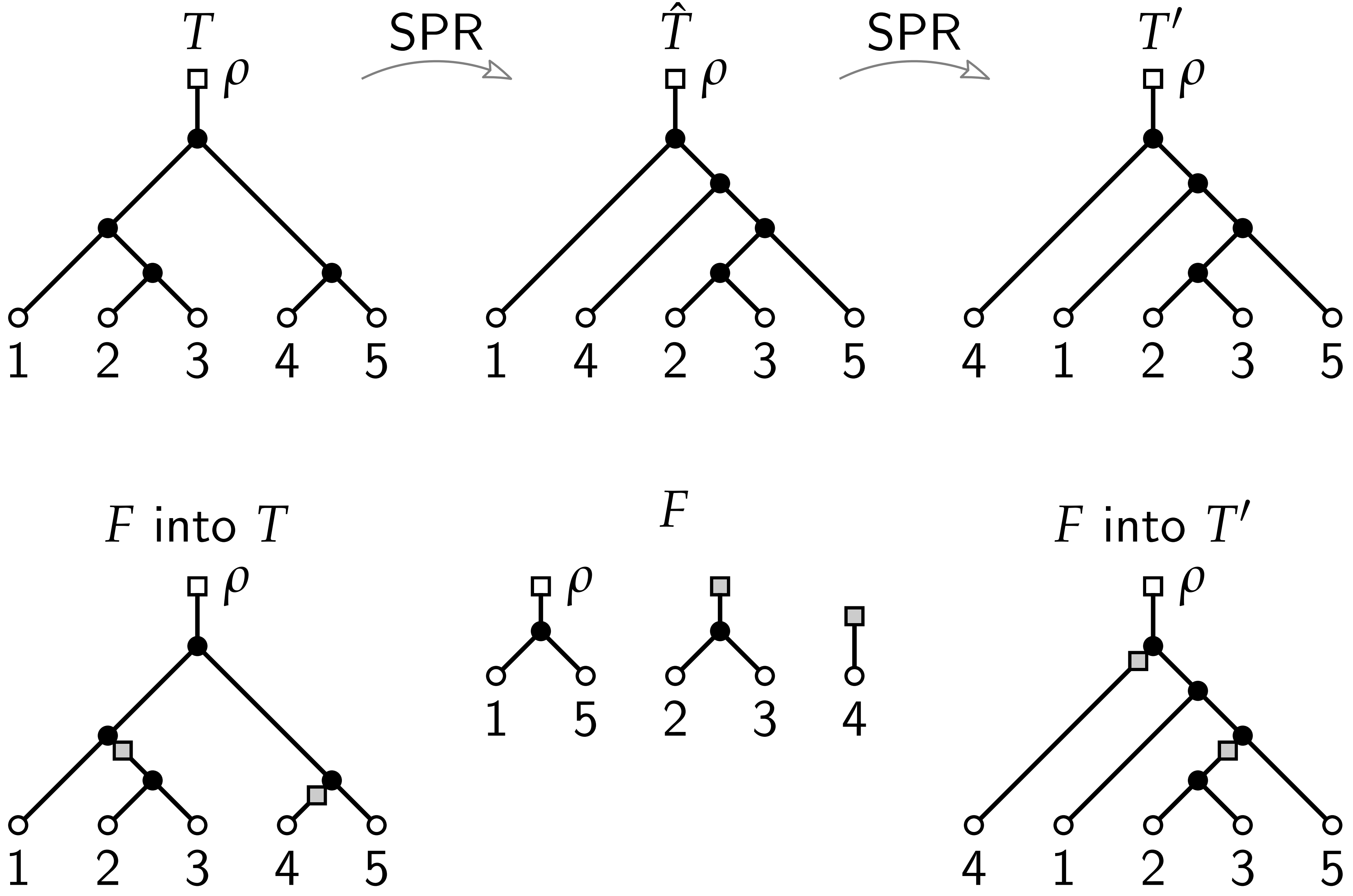



# Maximum Agreement Forests

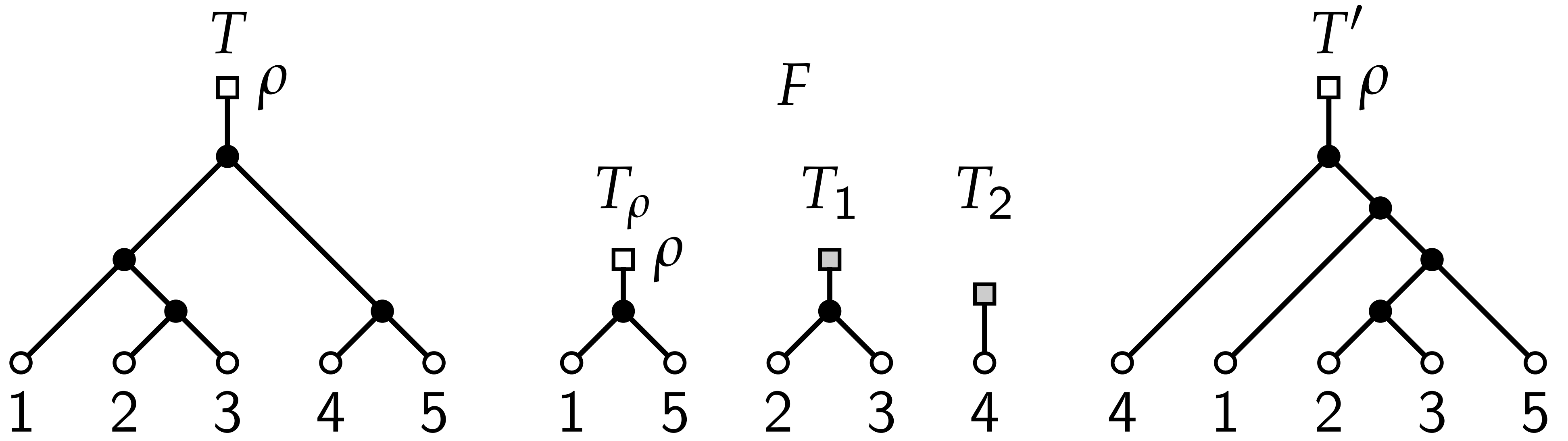


# Maximum Agreement Forests



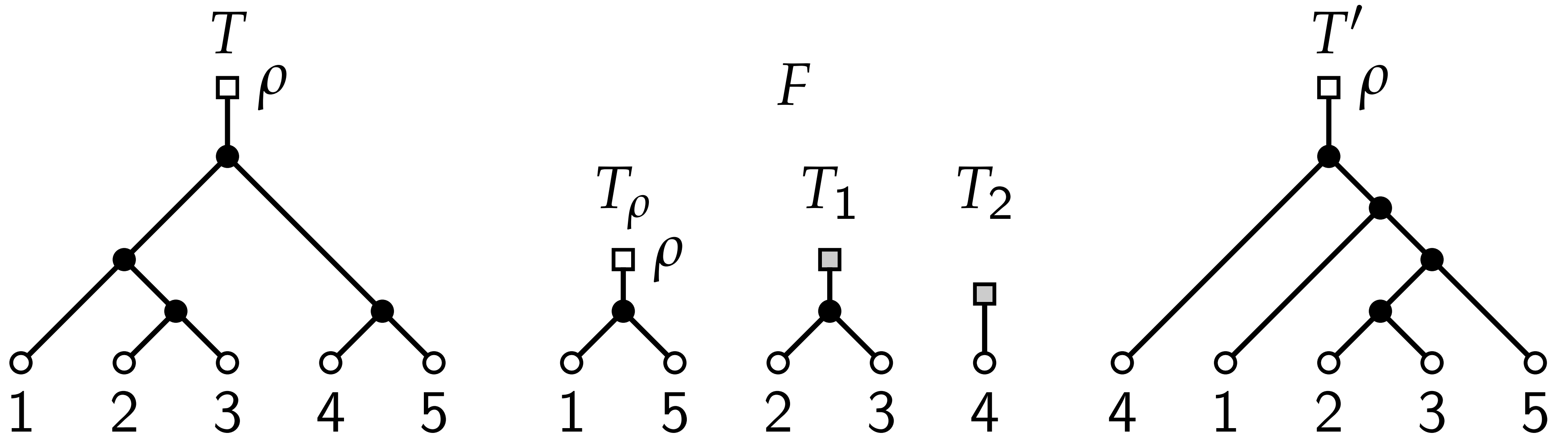
# Maximum Agreement Forests




An **agreement forest (AF)** $F$ 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 $X \cup \{\rho\}$,

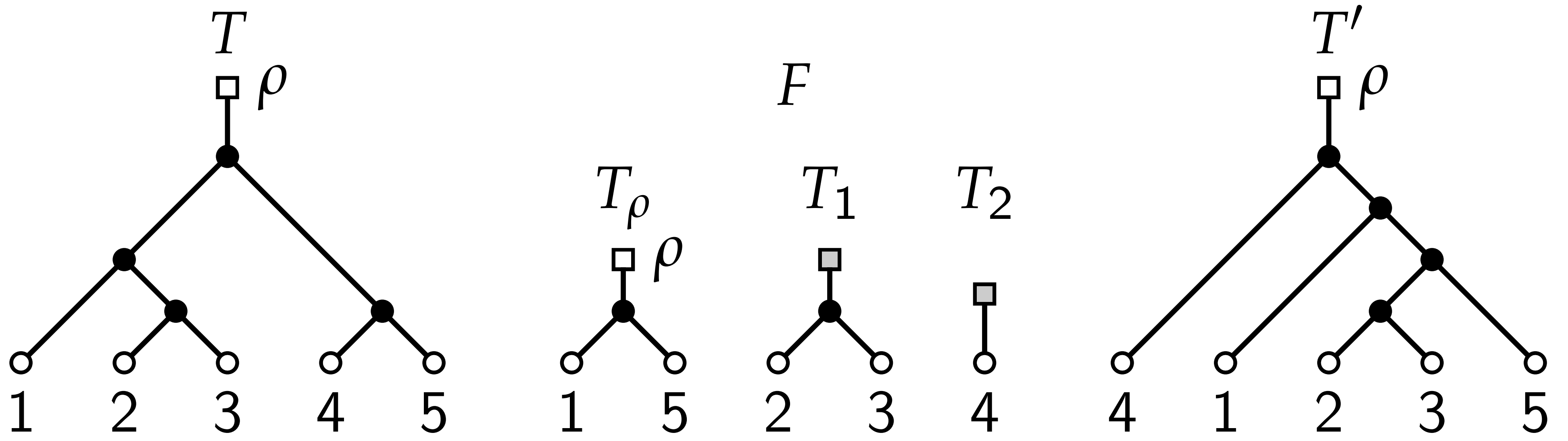
# Maximum Agreement Forests




An **agreement forest (AF)** $F$ 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 $X \cup \{\rho\}$,
- $\rho$ is in the label set of $T_\rho$, and

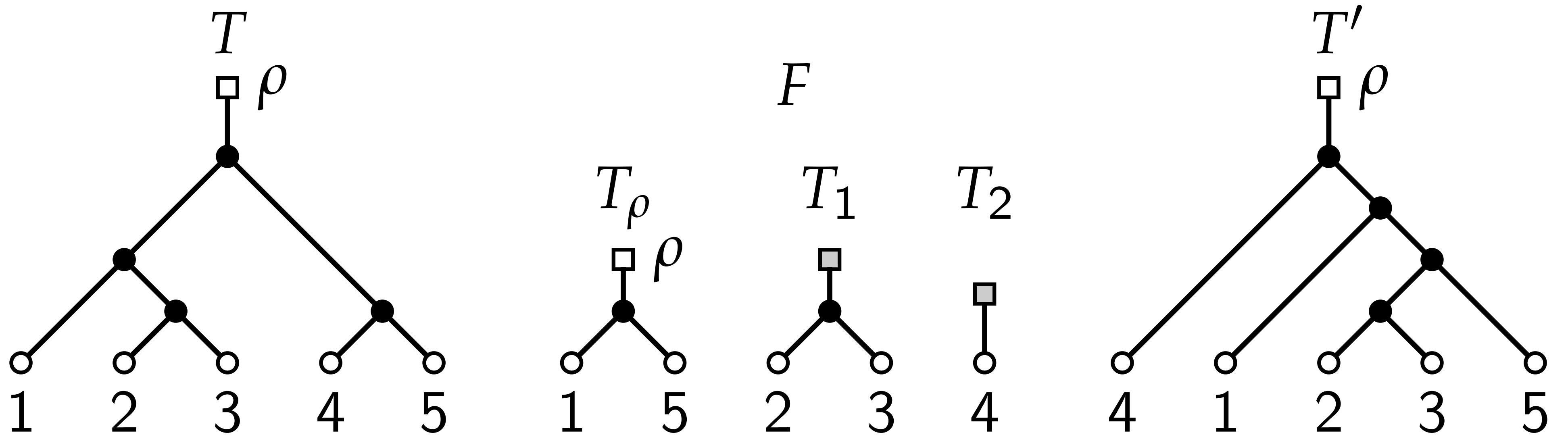
# Maximum Agreement Forests




An **agreement forest (AF)** $F$ 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 $X \cup \{\rho\}$,
- $\rho$ is in the label set of $T_\rho$, and
- there is an edge-disjoint embedding of the $T_i$s into $T$ and $T'$ where all edges of $T$ and $T'$ are covered. In other words, we can place all $T_i$s onto $T$ and $T'$ such that the $T_i$s do not overlap and every edge of $T$ and $T'$ lies under some $T_i$.

# Maximum Agreement Forests




An **agreement forest (AF)** $F$ 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 $X \cup \{\rho\}$,
- $\rho$ is in the label set of $T_\rho$, and
- there is an edge-disjoint embedding of the $T_i$s into $T$ and $T'$ where all edges of $T$ and $T'$ are covered. In other words, we can place all $T_i$s onto $T$ and $T'$ such that the $T_i$s do not overlap and every edge of $T$ and $T'$ lies under some $T_i$.

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

# Characterization

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

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

# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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.** $\mathrm{m}(T, T') = \mathrm{d}_{\mathrm{SPR}}(T, T')$

# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $d = \mathrm{d_{SPR}}(T, T')$.

# Characterization

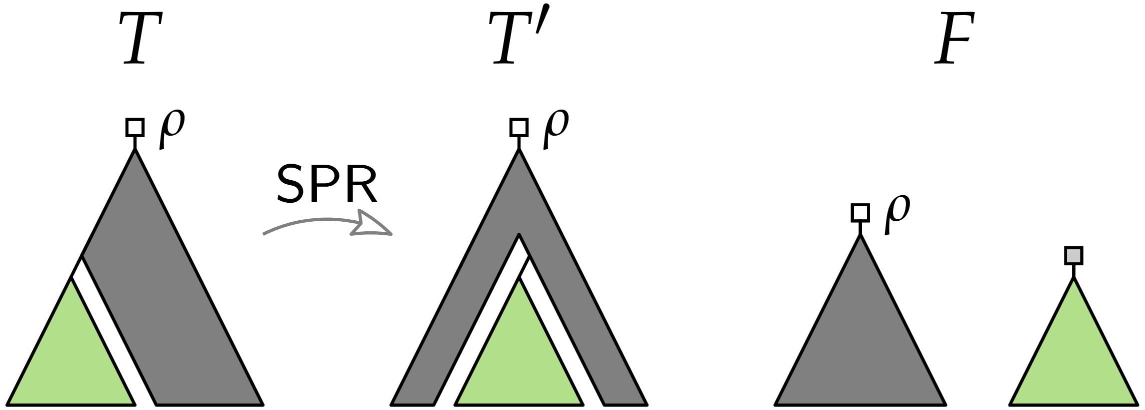
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define

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

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

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

- Case $d = 0$ is trivial and Case $d = 1$ is easy. ✓



# Characterization

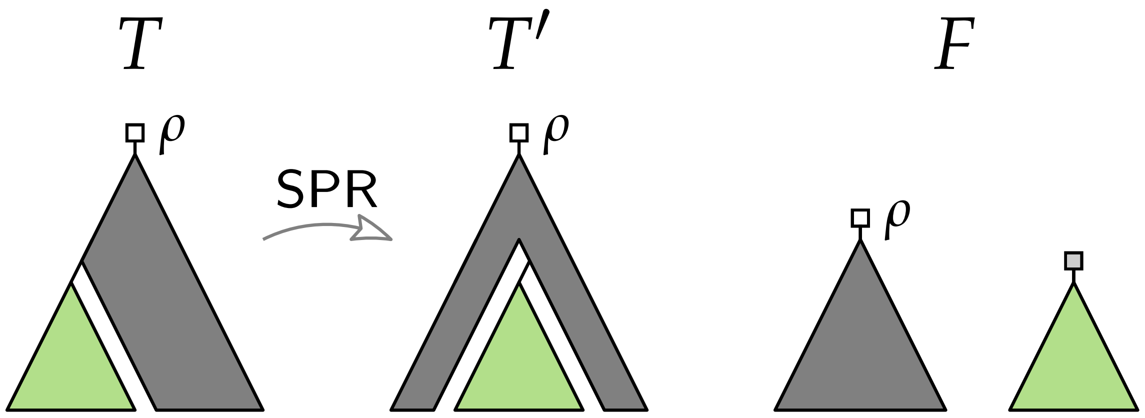
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of $T$ and $T'$.
Define

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

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

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

- Case $d = 0$ is trivial and Case $d = 1$ is easy. ✓
- Assume $\mathrm{m}(T, T') \leq \mathrm{d_{SPR}}(T, T')$ holds for all $d \leq \ell$.



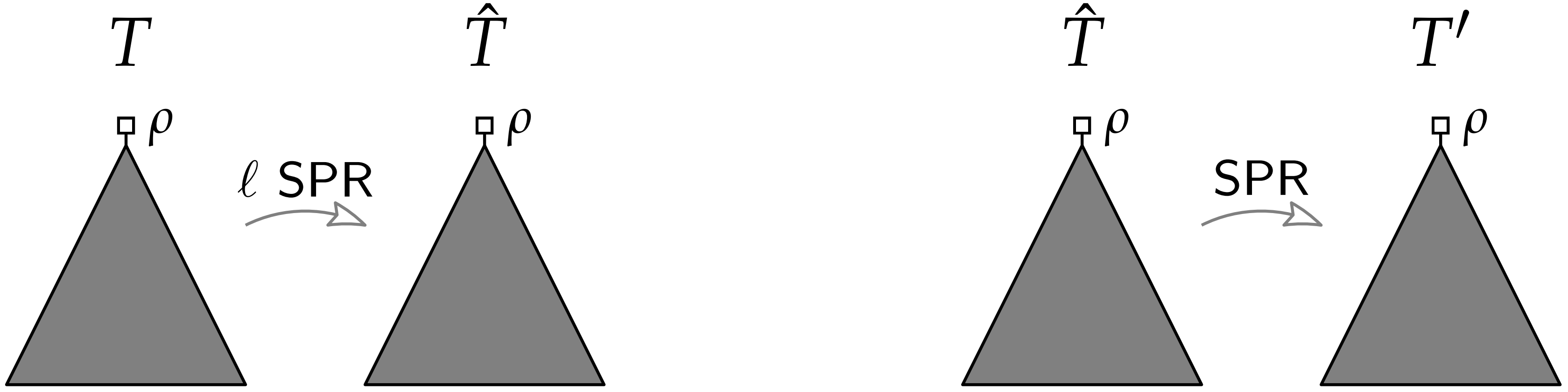
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $\hat{T}$ with $\mathrm{d_{SPR}}(T, \hat{T}) = \ell$ and $\mathrm{d_{SPR}}(\hat{T}, T') = 1$.



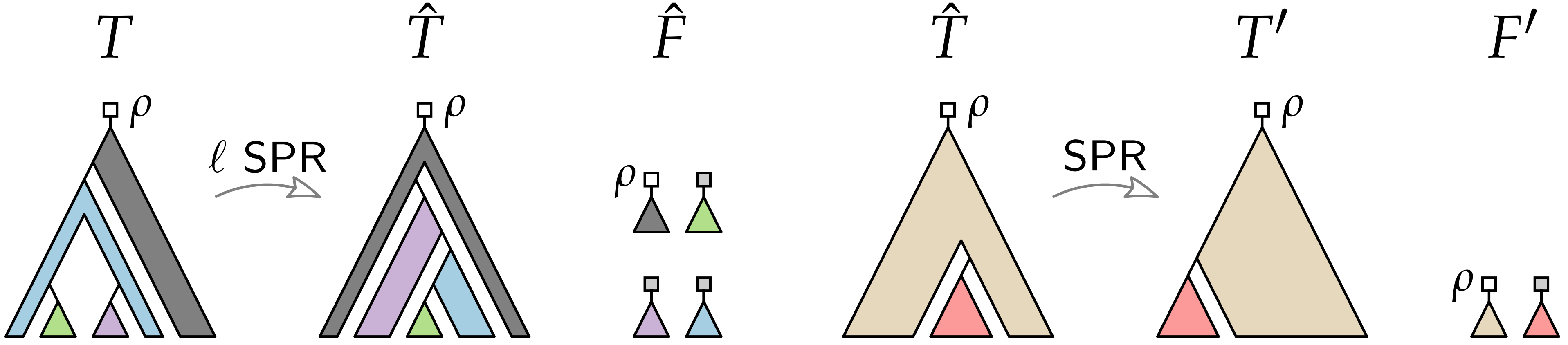
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $\hat{T}$ with $\mathrm{d_{SPR}}(T, \hat{T}) = \ell$ and $\mathrm{d_{SPR}}(\hat{T}, T') = 1$.
- $\exists$ MAF $\hat{F}$ for $T$ & $\hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T}$ & $T'$ of size 2.



# Characterization

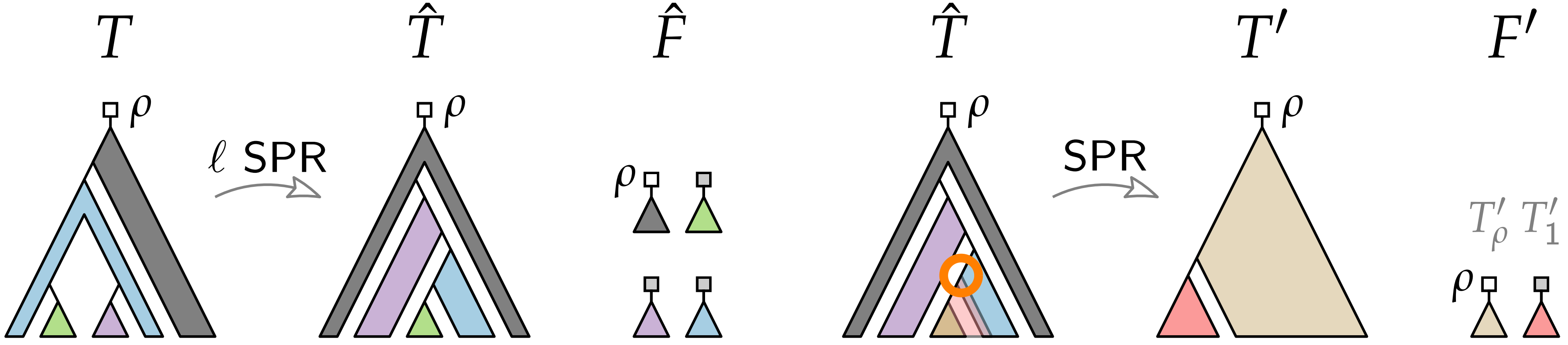
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of $T$ and $T'$. Define

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

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

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

- If $d = \ell + 1$, then there exists $\hat{T}$ with $\text{d}_{\text{SPR}}(T, \hat{T}) = \ell$ and $\text{d}_{\text{SPR}}(\hat{T}, T') = 1$.
- $\exists$ MAF $\hat{F}$ for $T$ & $\hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T}$ & $T'$ of size 2.
- Compose $\hat{T}$ by subtrees of $\hat{F}$. The subtree $T'_1$ of $F'$ is rooted at one edge of $\hat{T}$ within one subtree of $\hat{F}$.



# Characterization

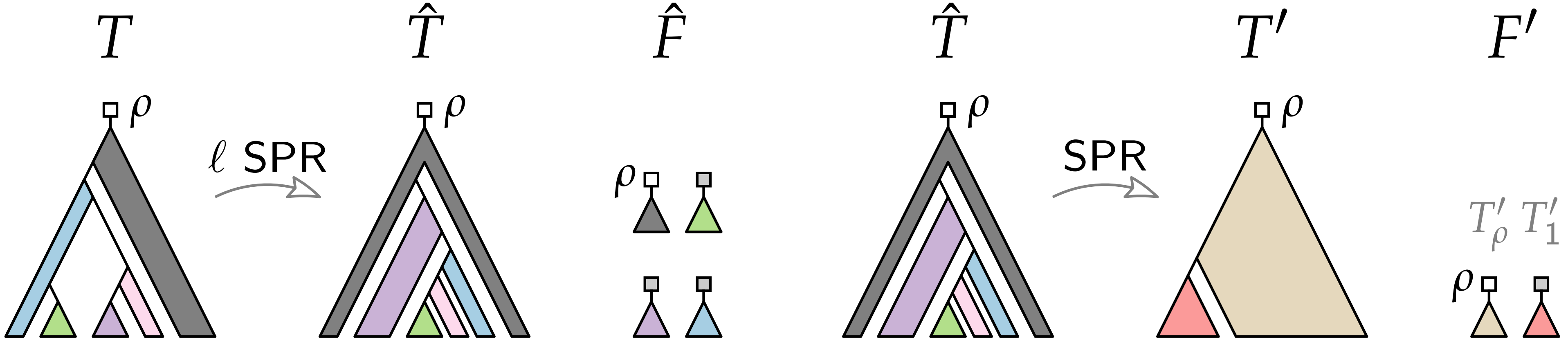
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$

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

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

- If $d = \ell + 1$, then there exists $\hat{T}$ with $d_{SPR}(T, \hat{T}) = \ell$ and $d_{SPR}(\hat{T}, T') = 1$.
- $\exists$ MAF $\hat{F}$ for $T$ & $\hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T}$ & $T'$ of size 2.
- Compose $\hat{T}$ by subtrees of $\hat{F}$. The subtree $T'_1$ of $F'$ is rooted at one edge of $\hat{T}$ within one subtree of $\hat{F}$.



# Characterization

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

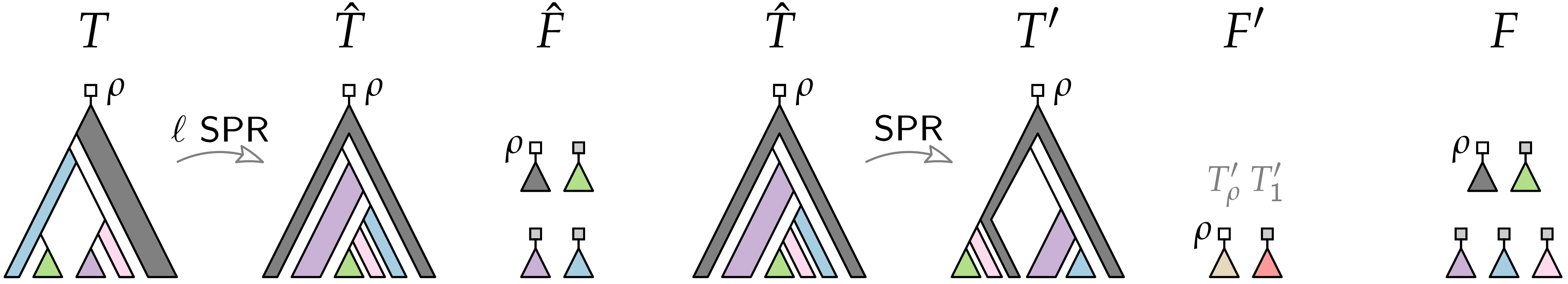
$$\mathrm{m}(T, T') = k = |F| - 1.$$

**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 $\hat{T}$ with $\mathrm{d_{SPR}}(T, \hat{T}) = \ell$ and $\mathrm{d_{SPR}}(\hat{T}, T') = 1$.

- $\exists$ MAF $\hat{F}$ for $T$ & $\hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T}$ & $T'$ of size 2.
- Compose $\hat{T}$ by subtrees of $\hat{F}$. The subtree $T'_1$ of $F'$ is rooted at one edge of $\hat{T}$ within one subtree of $\hat{F}$.
- Subdivide the corresponding tree to obtain $F$ from $\hat{F}$, which is an AF for $T$ and $T'$.



# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $m = \mathrm{m}(T, T')$.

# Characterization

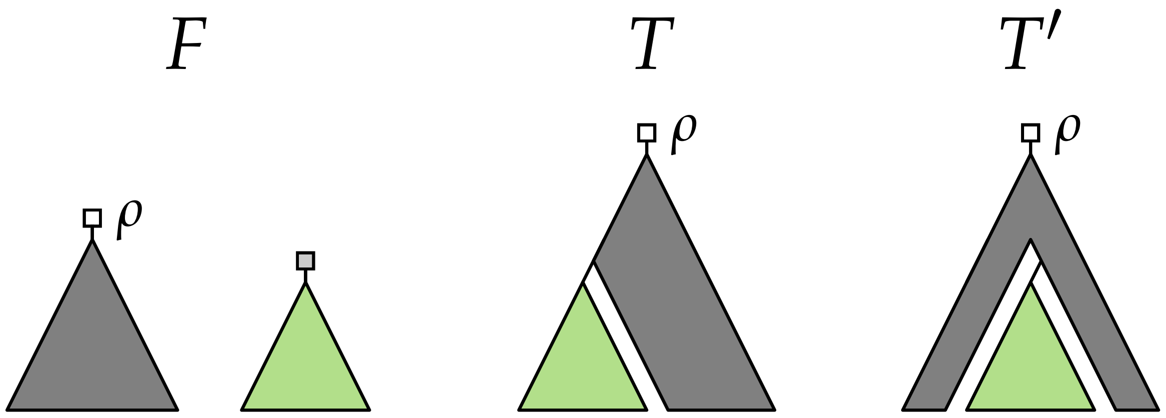
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define

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

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

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

- Case $m = 0$ is trivial and Case $m = 1$ is easy. ✓



# Characterization

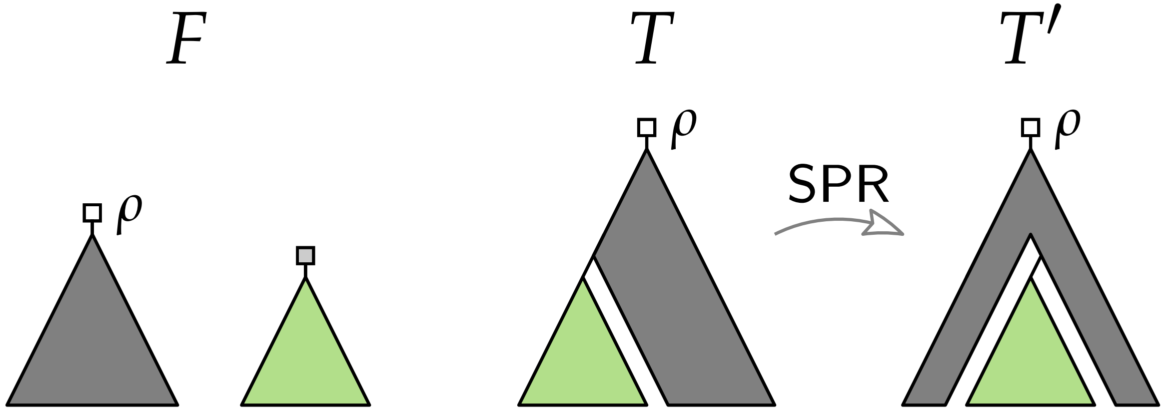
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of $T$ and $T'$.
Define

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

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

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

- Case $m = 0$ is trivial and Case $m = 1$ is easy. ✓



# Characterization

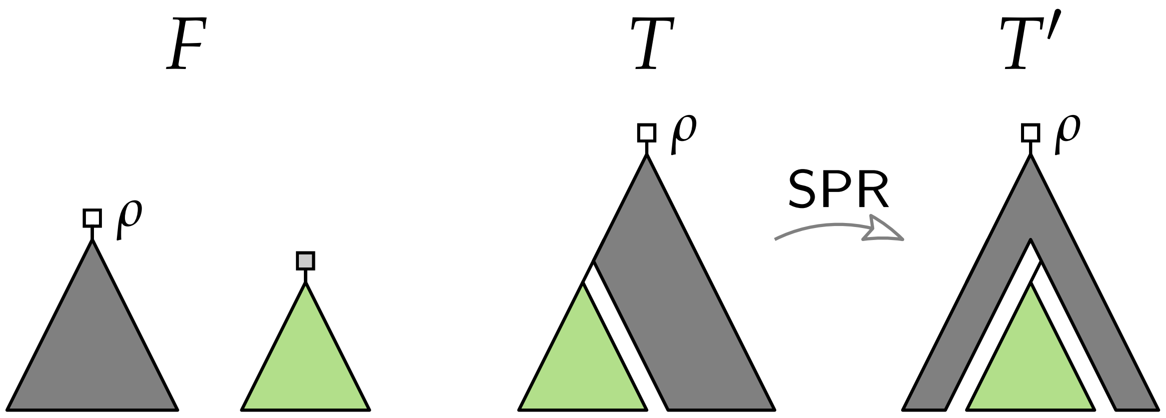
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of $T$ and $T'$.
Define

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

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

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

- Case $m = 0$ is trivial and Case $m = 1$ is easy. ✓
- Assume $\mathrm{m}(T, T') \geq \mathrm{d_{SPR}}(T, T')$ holds for all $m \leq \ell$.



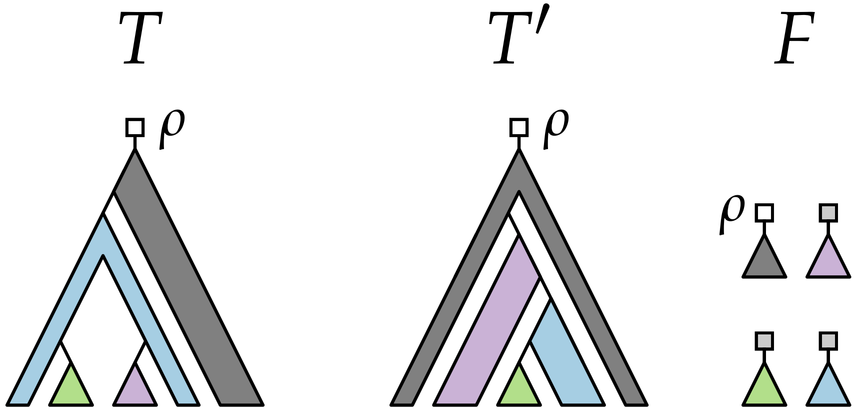
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $m = \mathrm{m}(T, T')$.

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



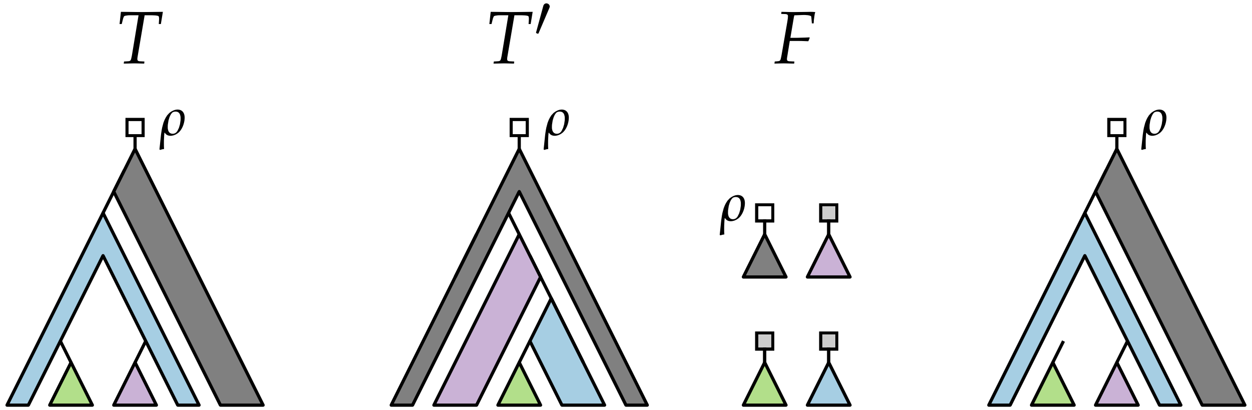
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $m = \mathrm{m}(T, T')$.

- 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 $T$ due to the nesting structure of subtrees.



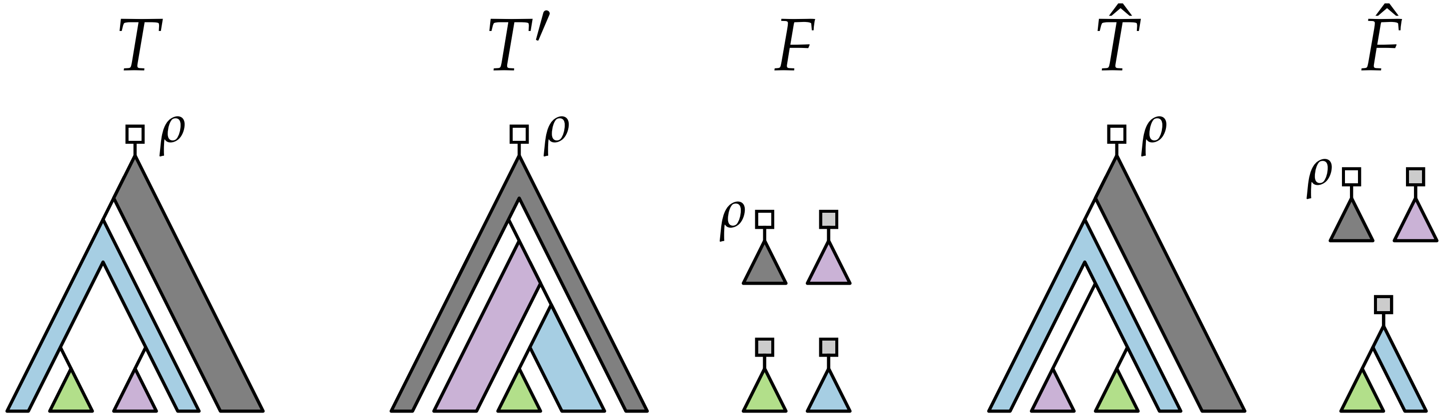
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $m = \mathrm{m}(T, T')$.

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



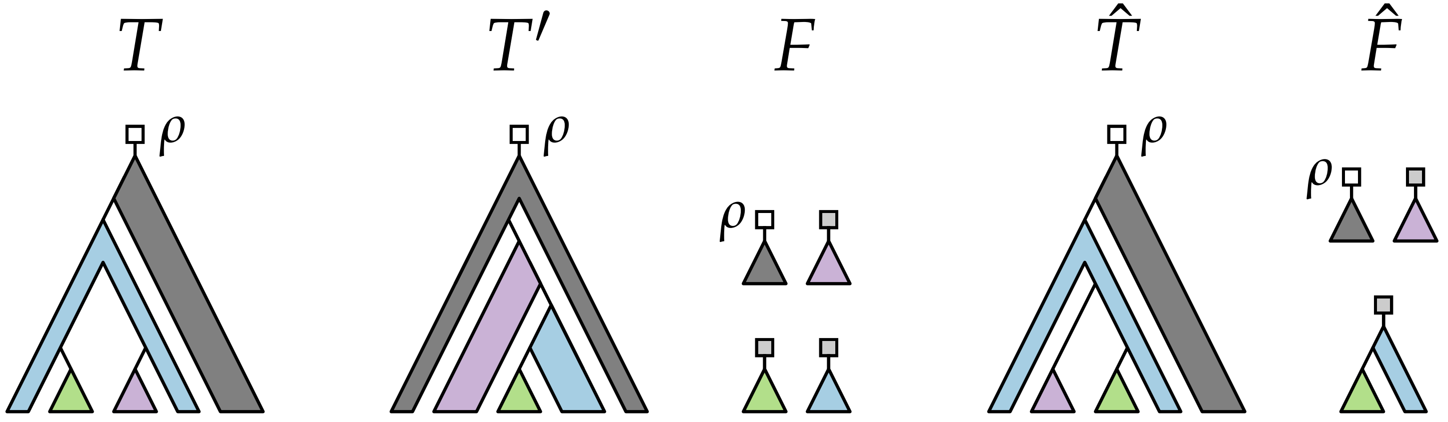
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $m = \mathrm{m}(T, T')$.

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




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

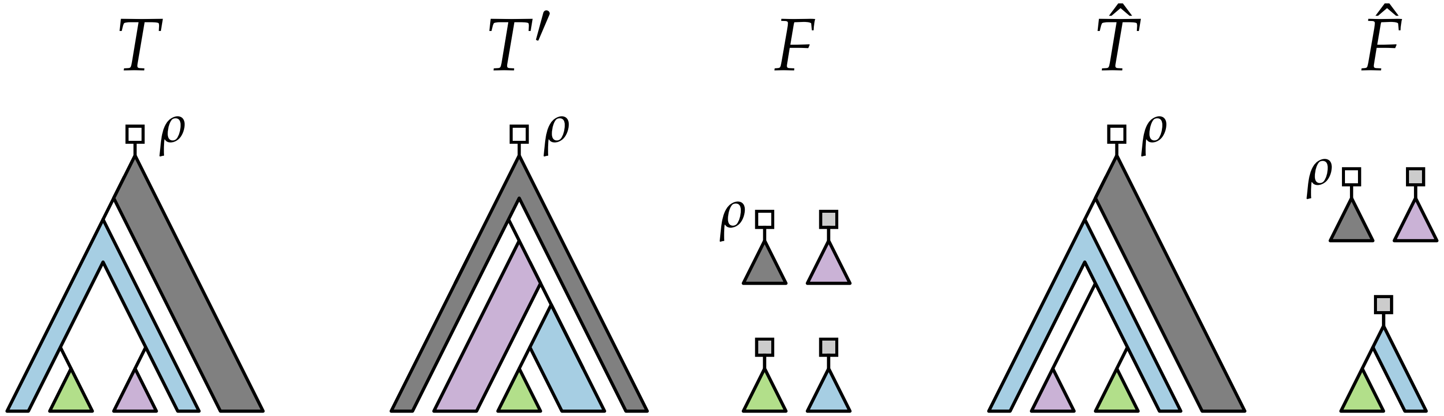
# Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and 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 $m = \mathrm{m}(T, T')$.

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




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

# Characterization

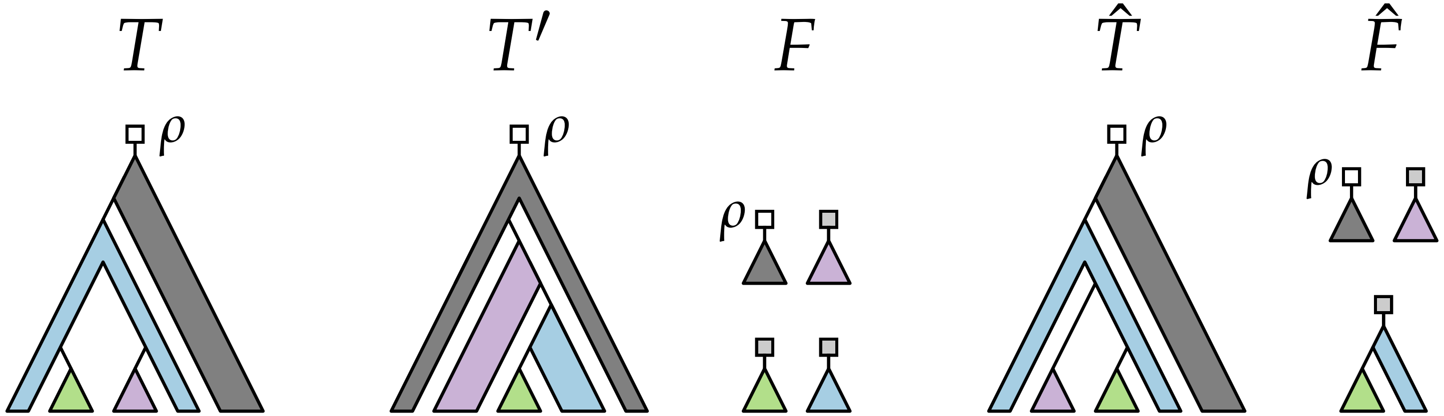
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \dots, T_k\}$ be a MAF of $T$ and $T'$.
Define

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

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

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

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




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

□

# Problem & Plan

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

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

# Problem & Plan

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

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

**Plan.**

- Construct **kernel** of the problem.
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  - Derive $\mathrm{d_{SPR}}(T, T')$ from $\mathrm{d_{SPR}}(S, S')$.

# Problem & Plan

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

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

**Plan.**

- Construct **kernel** of the problem.
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  - Derive $\mathrm{d_{SPR}}(T, T')$ from $\mathrm{d_{SPR}}(S, S')$.
- Show that the size of the kernel depends on $\mathrm{d_{SPR}}(T, T')$.

# Problem & Plan

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

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

**Plan.**

- Construct **kernel** of the problem.
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  - Derive $\mathrm{d}_{\mathrm{SPR}}(T, T')$ from $\mathrm{d}_{\mathrm{SPR}}(S, S')$.
- Show that the size of the kernel depends on $\mathrm{d}_{\mathrm{SPR}}(T, T')$.
- Devise an FPT algorithm with respect to $\mathrm{d}_{\mathrm{SPR}}$.

# Problem & Plan

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

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

**Plan.**

- Construct **kernel** of the problem.
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  - Derive $\mathrm{d}_{\mathrm{SPR}}(T, T')$ from $\mathrm{d}_{\mathrm{SPR}}(S, S')$.
- Show that the size of the kernel depends on $\mathrm{d}_{\mathrm{SPR}}(T, T')$.
- Devise an FPT algorithm with respect to $\mathrm{d}_{\mathrm{SPR}}$.
- Sketch an approximation algorithm.

# Kernelization – Subtrees

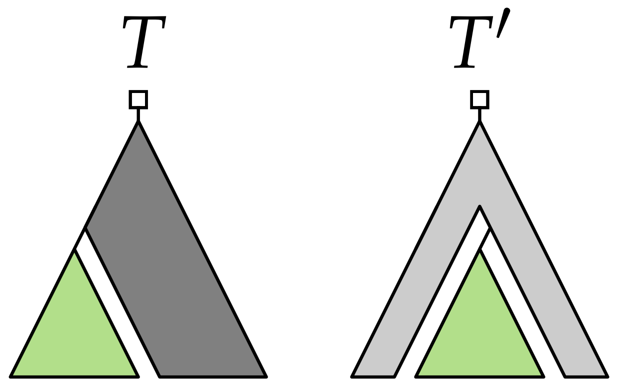
**Common subtree reduction.**

- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.

# Kernelization – Subtrees

**Common subtree reduction.**

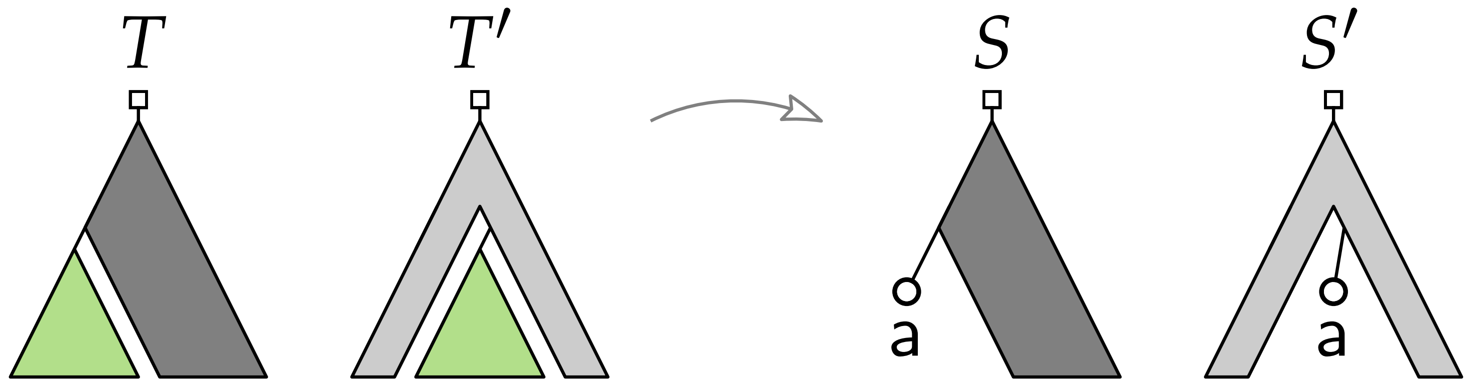
- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.



# Kernelization – Subtrees

**Common subtree reduction.**

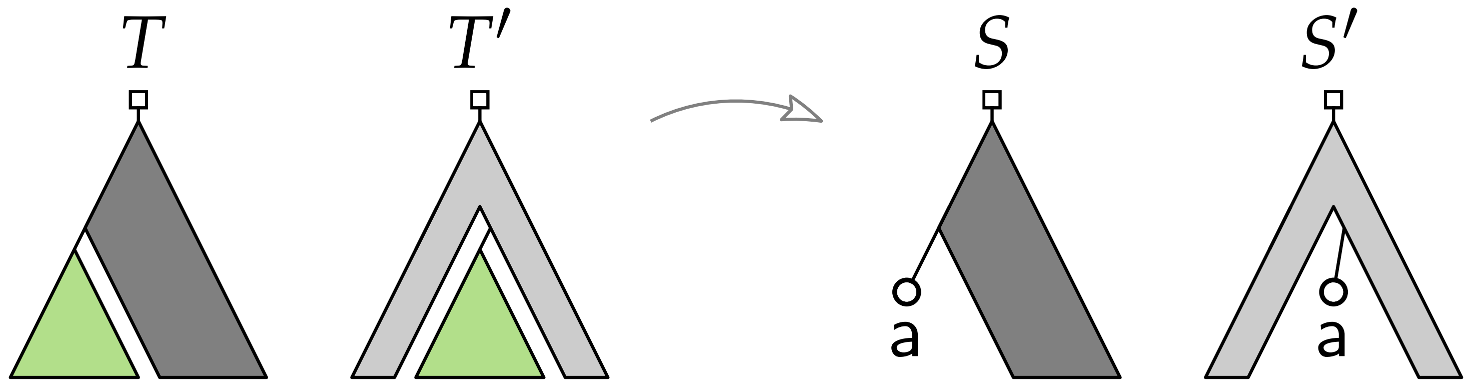
- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.



# Kernelization – Subtrees

**Common subtree reduction.**

- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.

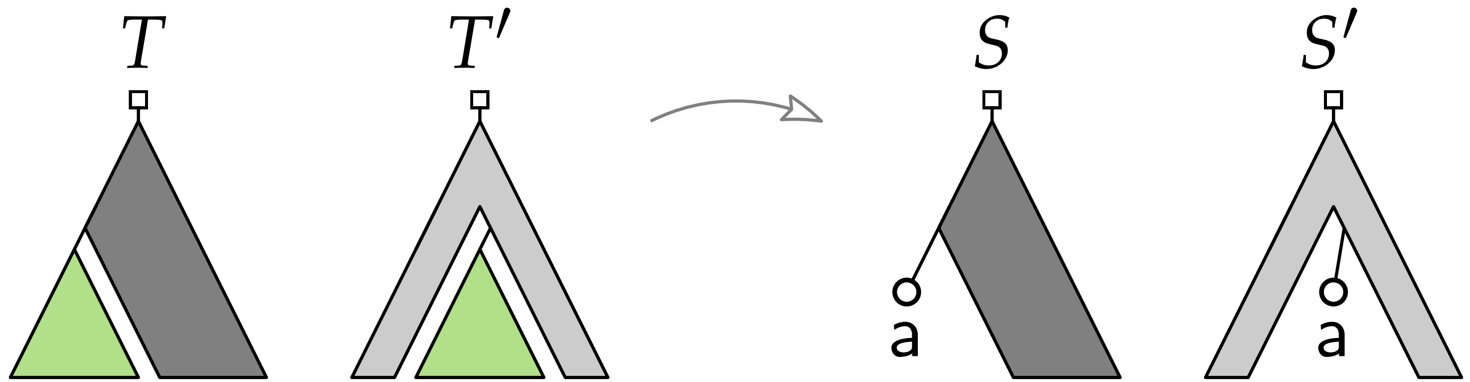



**Lemma 5.** Applying the common subtree reduction is safe, i.e., $\mathrm{d}_{\mathsf{SPR}}(T, T') = \mathrm{d}_{\mathsf{SPR}}(S, S')$.

# Kernelization – Subtrees

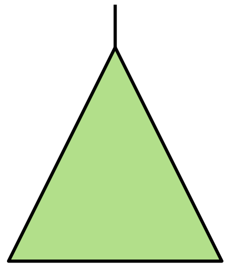
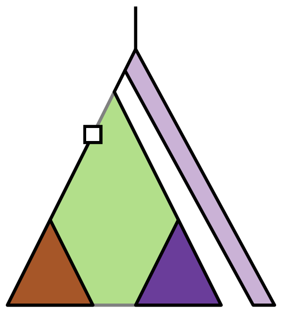
**Common subtree reduction.**

- Replace any subtree (with $\geq 2$ leaves) 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., $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

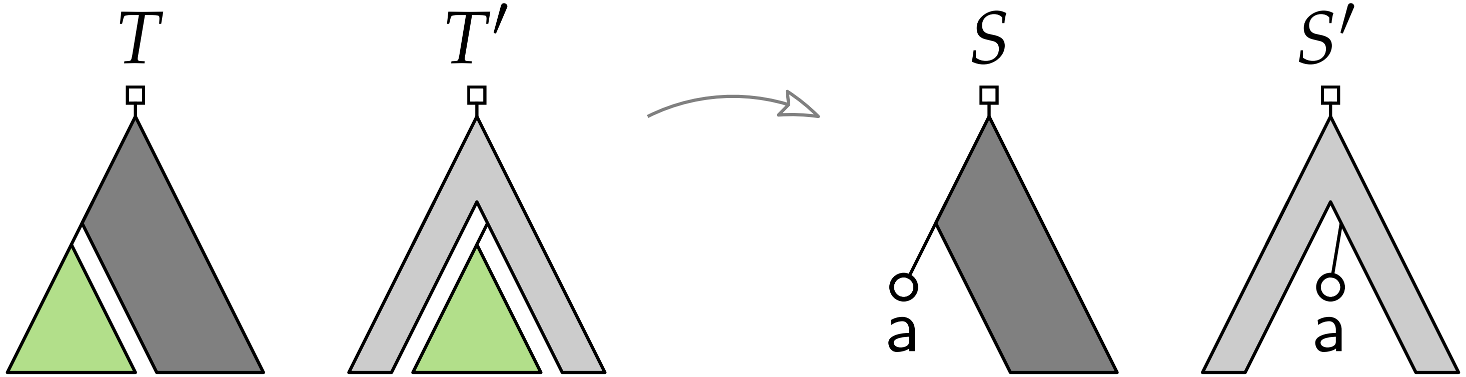
**Proof.**

Suppose  is covered by two trees of MAF 

# Kernelization – Subtrees

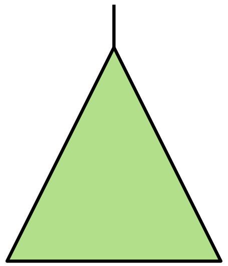
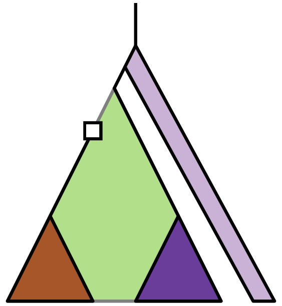
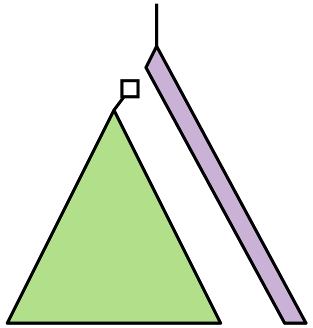
**Common subtree reduction.**

- Replace any subtree (with $\geq 2$ leaves) 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., $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

**Proof.**

Suppose  is covered by two trees of MAF  then there is an alternative MAF of the same size 

# Kernelization – Chains

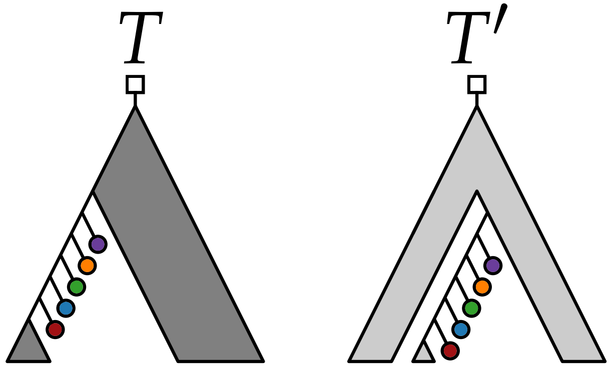
**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

# Kernelization – Chains

**Chain reduction.**

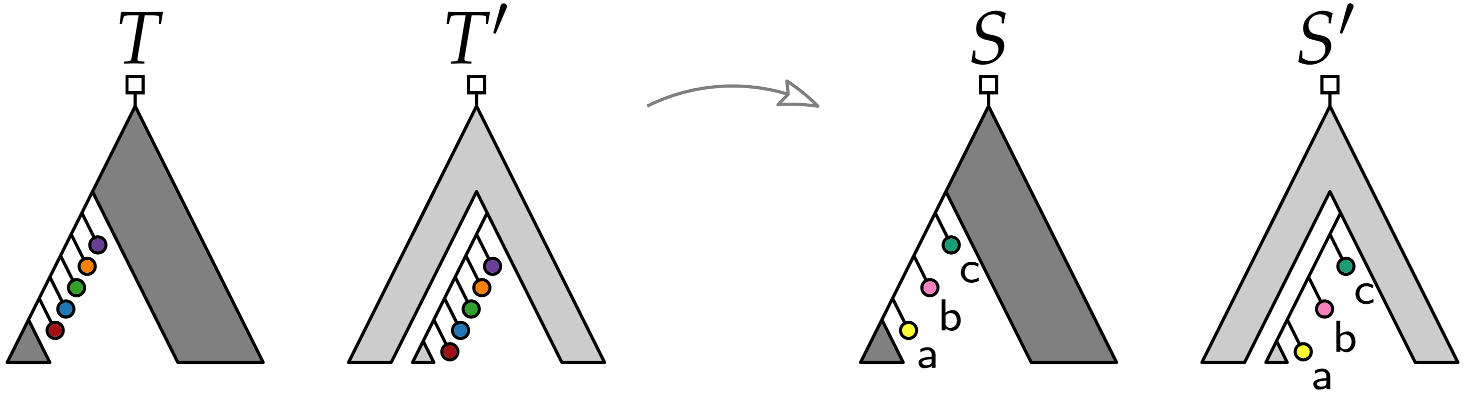
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.



# Kernelization – Chains

**Chain reduction.**

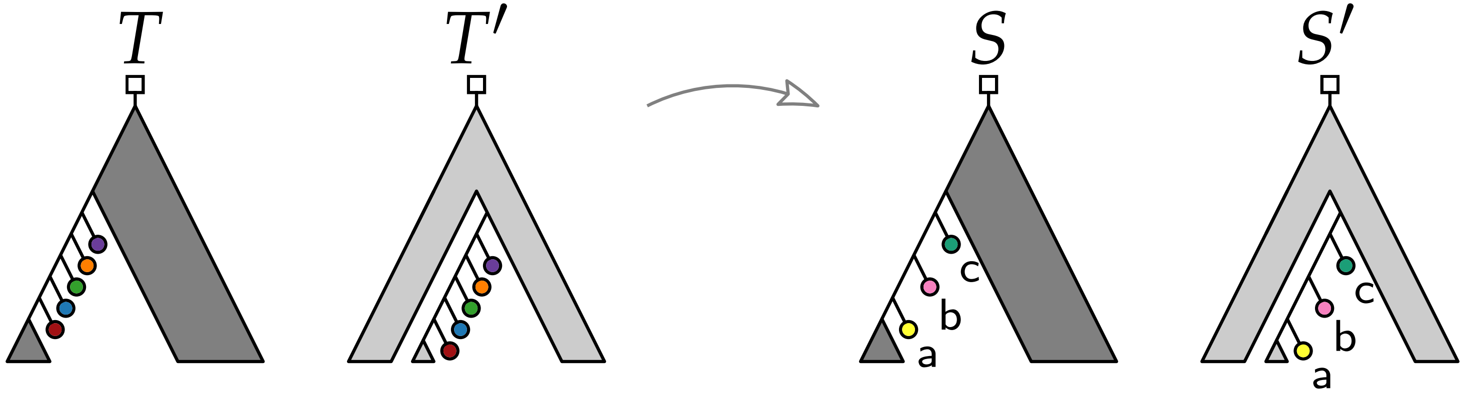
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.



# Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

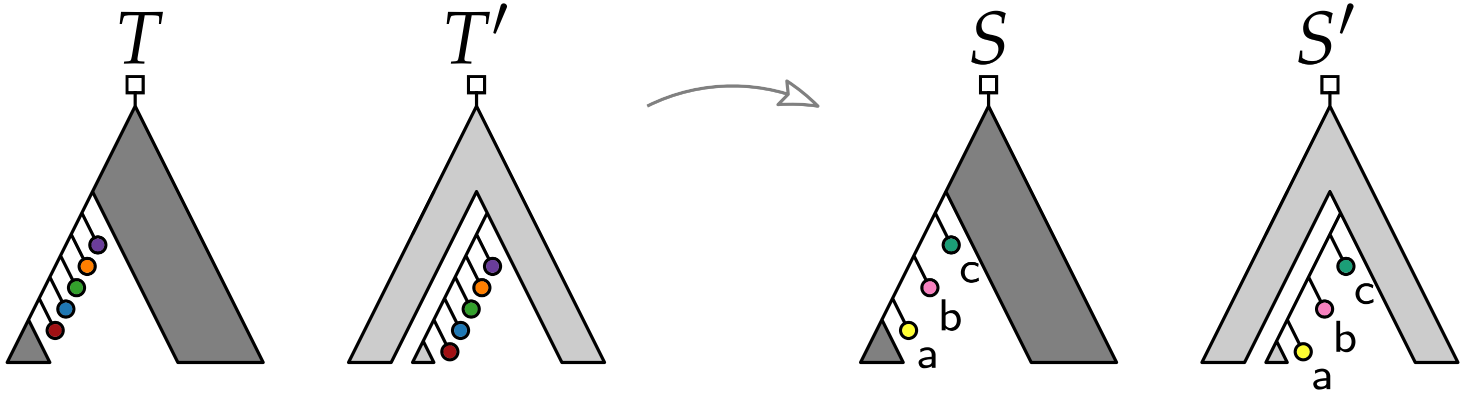



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathrm{d}_{\mathrm{SPR}}(T, T') = \mathrm{d}_{\mathrm{SPR}}(S, S')$.

# Kernelization – Chains

## Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.




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

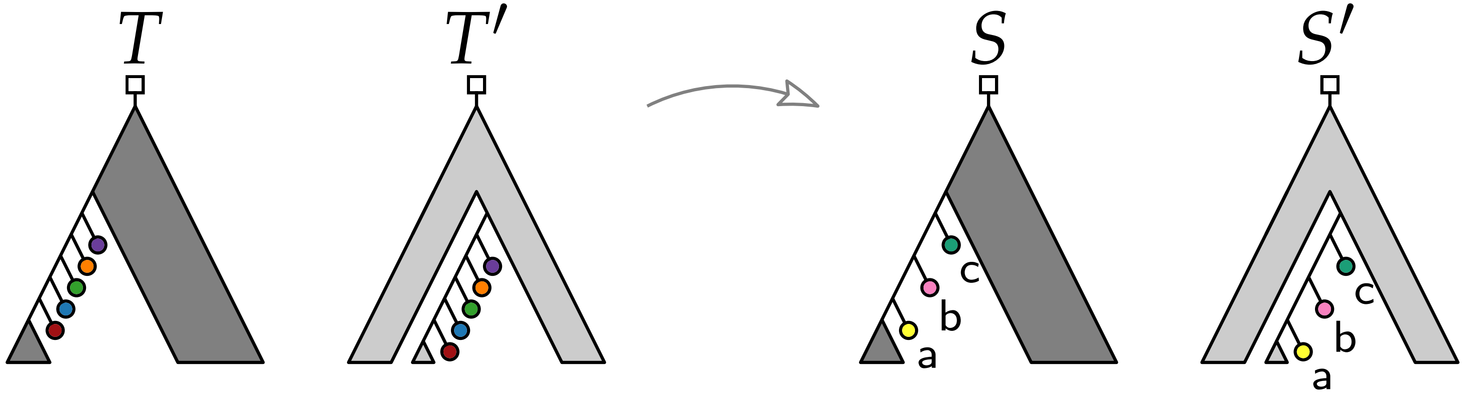
**Proof.**

- 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 $T'$.

# Kernelization – Chains

## Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

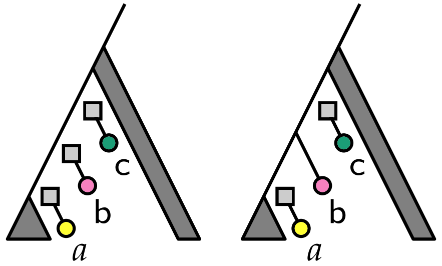



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

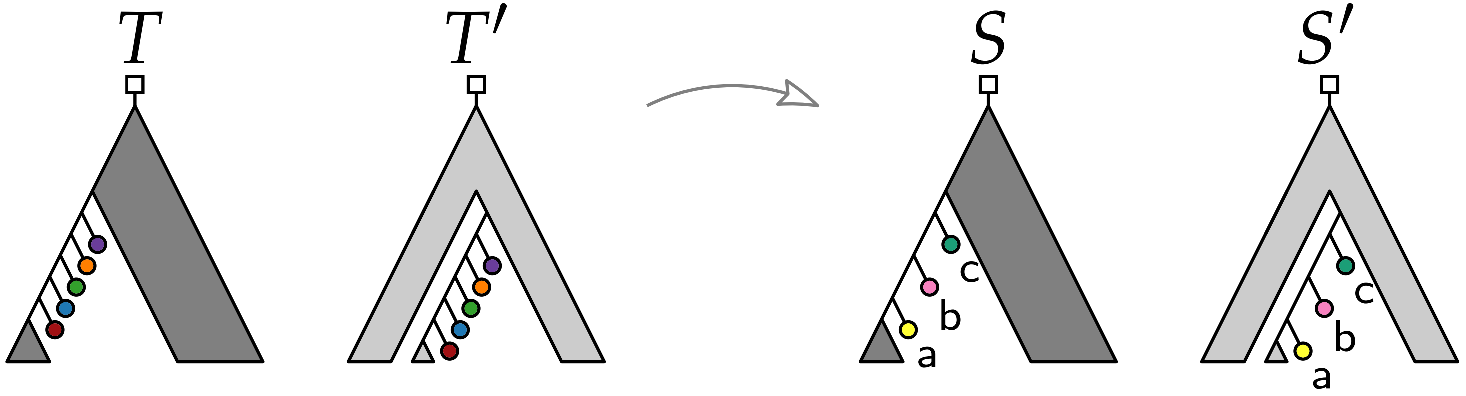
Case 1



# Kernelization – Chains

## Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

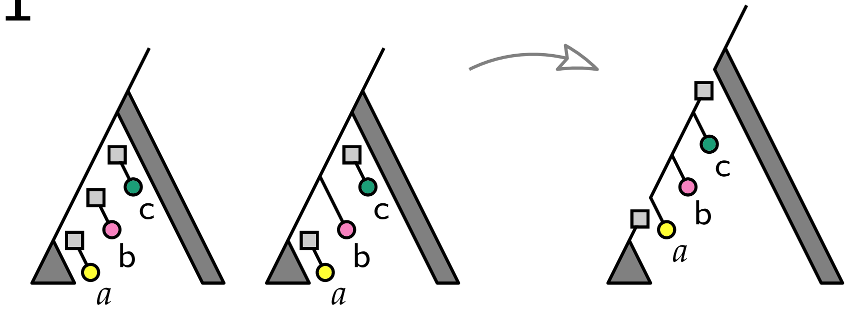



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

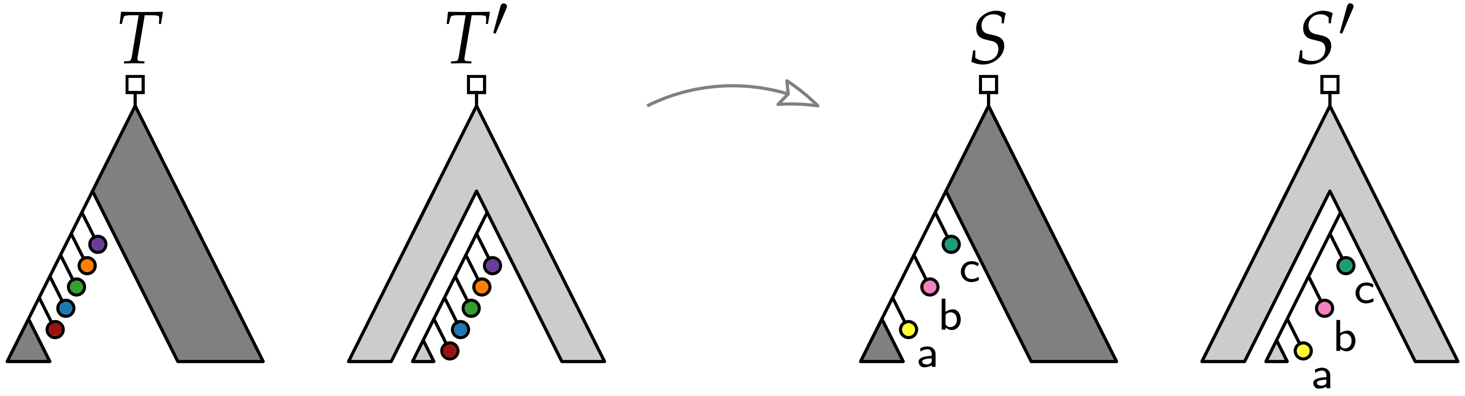
Case 1



# Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

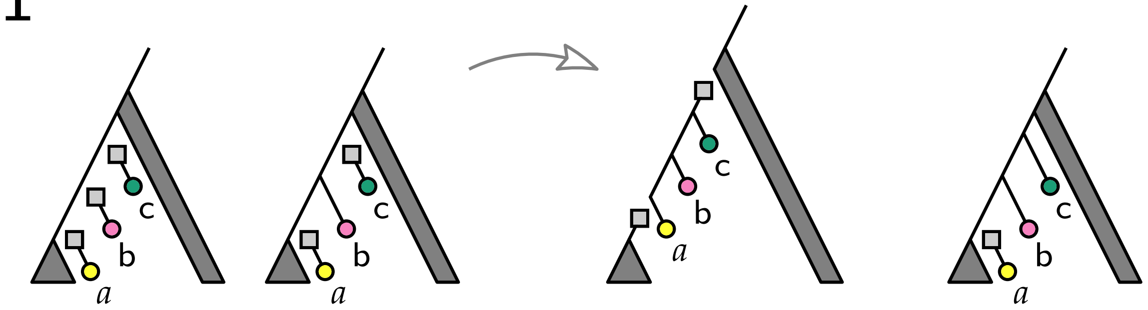



**Lemma 6.** Applying chain reduction is safe, i.e., $d_{\mathsf{SPR}}(T, T') = d_{\mathsf{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

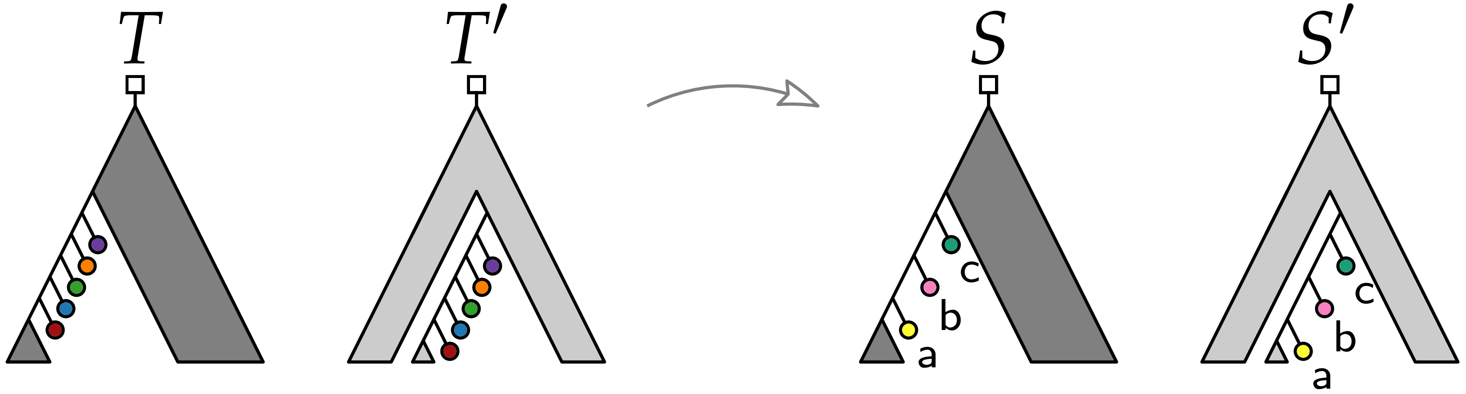
Case 1



# Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

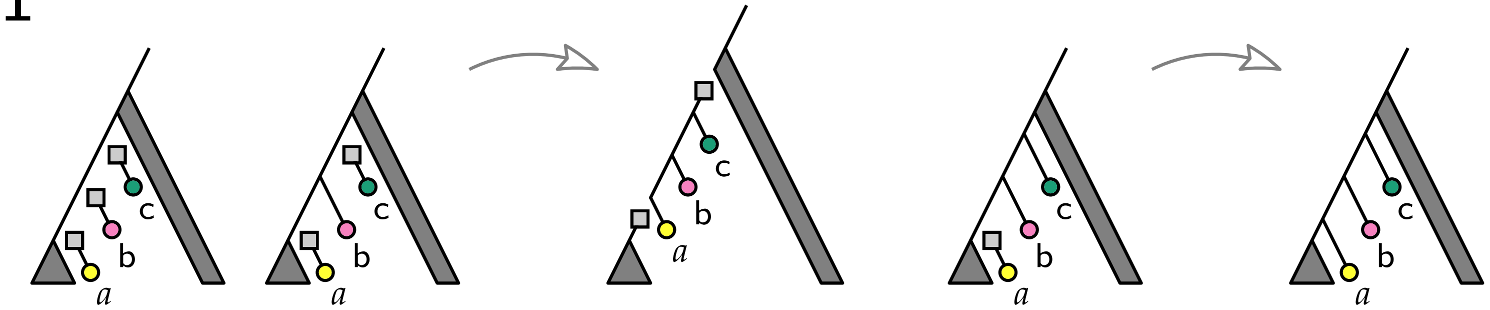



**Lemma 6.** Applying chain reduction is safe, i.e., $d_{\mathsf{SPR}}(T, T') = d_{\mathsf{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

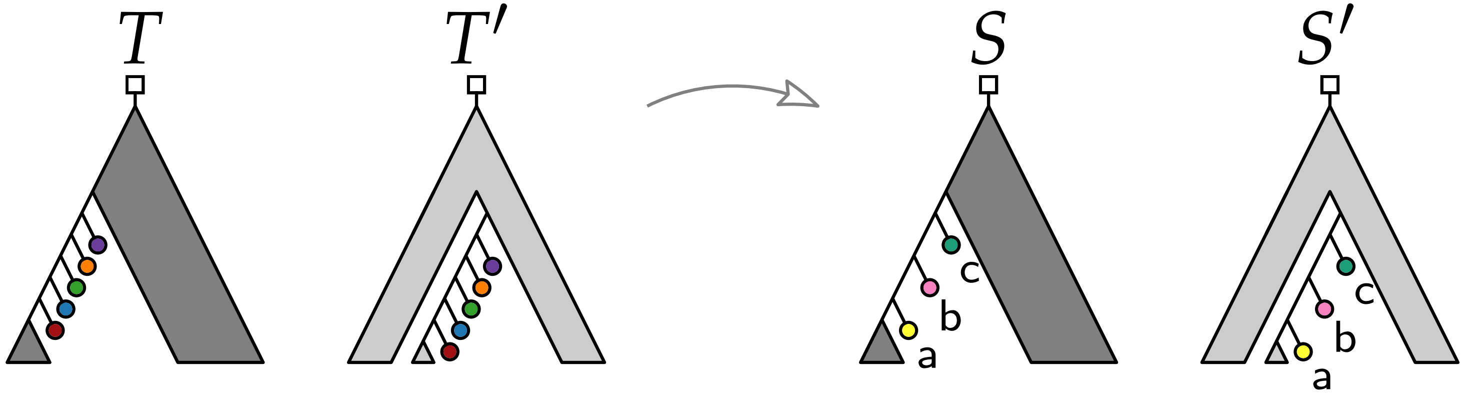
Case 1



# Kernelization – Chains

## Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

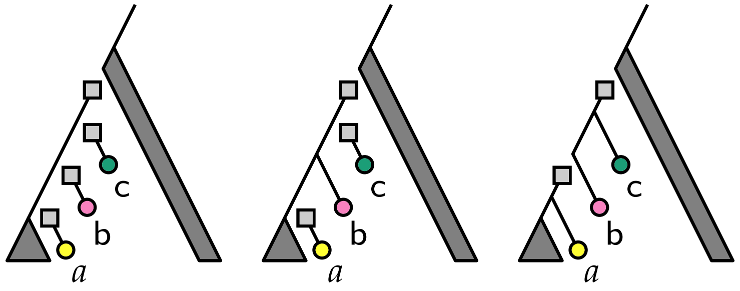



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathsf{d_{SPR}}(T, T') = \mathsf{d_{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

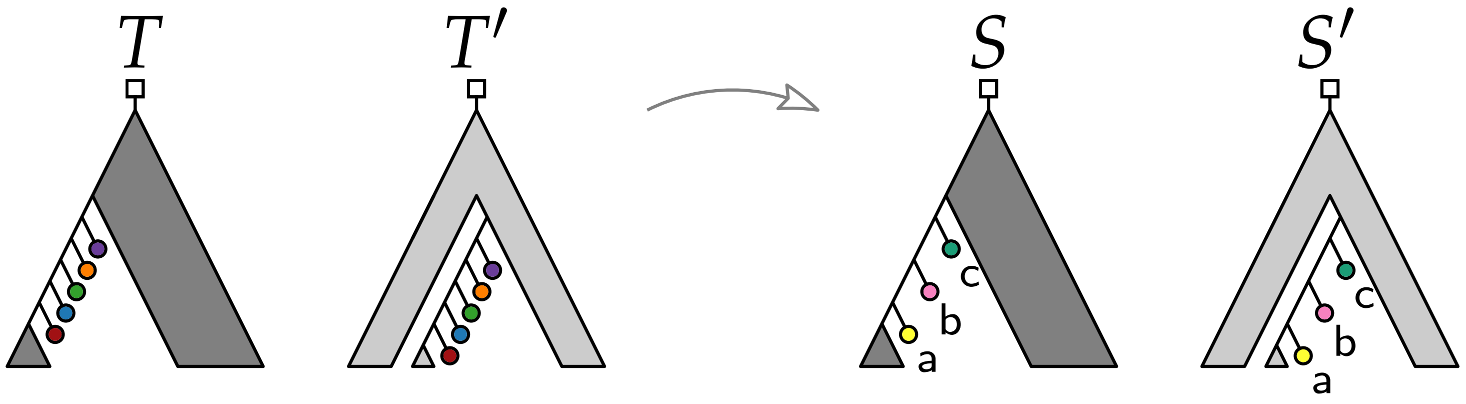
Case 2



# Kernelization – Chains

## Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

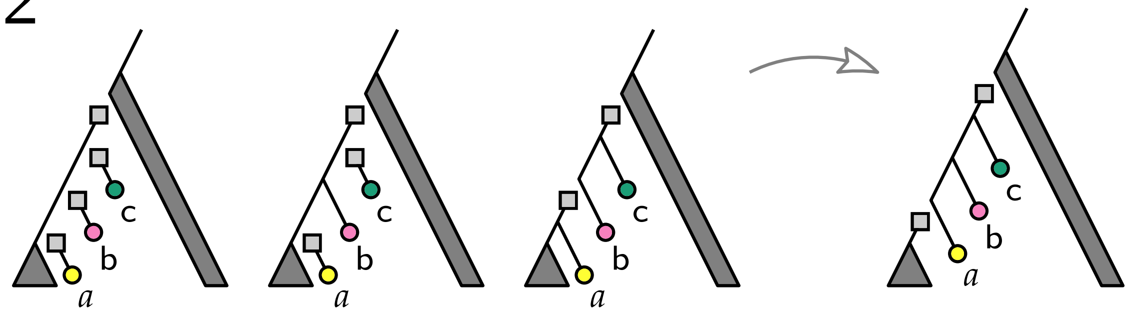



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

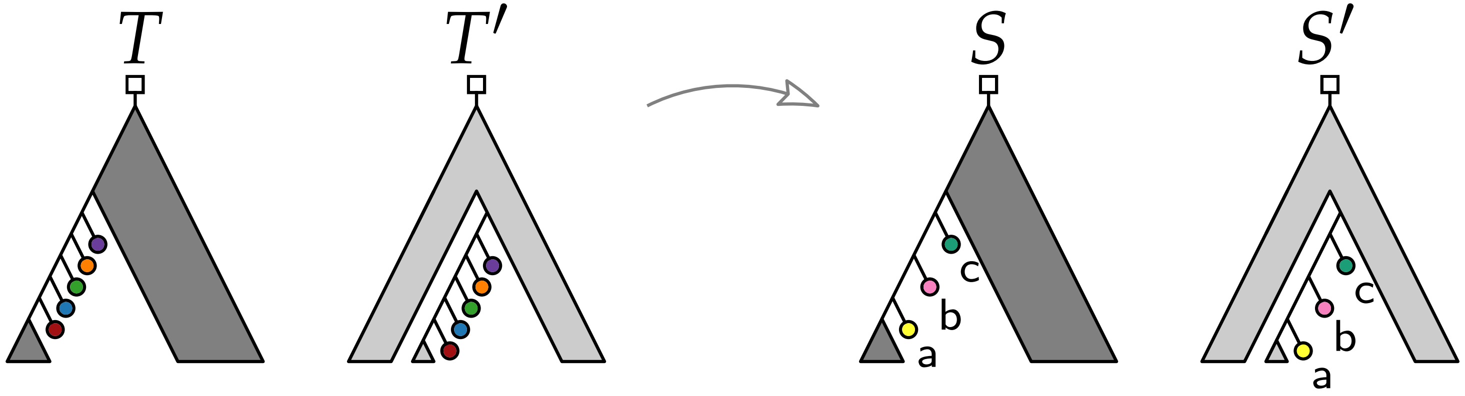
Case 2



# Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

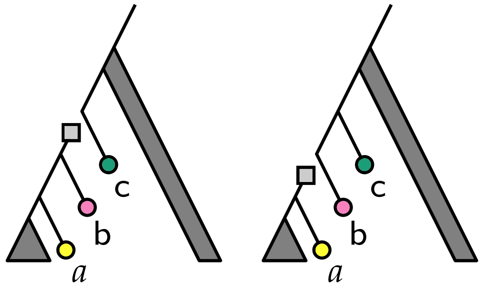



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

**Proof.**

- Consider embedding of a MAF $F$ into $S$.

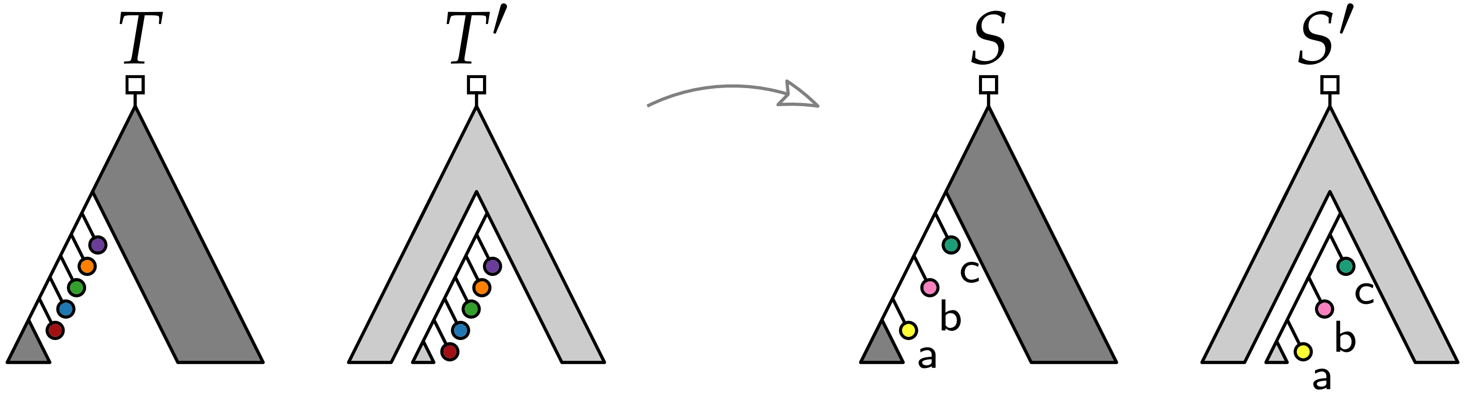
Case 3



# Kernelization – Chains

## Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

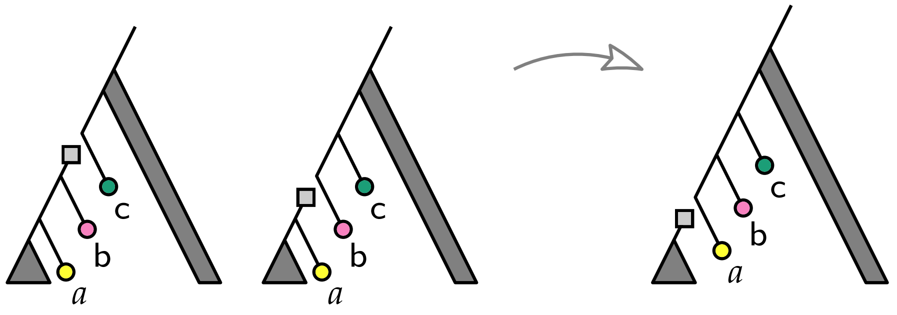



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathrm{d_{SPR}}(T, T') = \mathrm{d_{SPR}}(S, S')$.

## Proof.

- Consider embedding of a MAF $F$ into $S$.

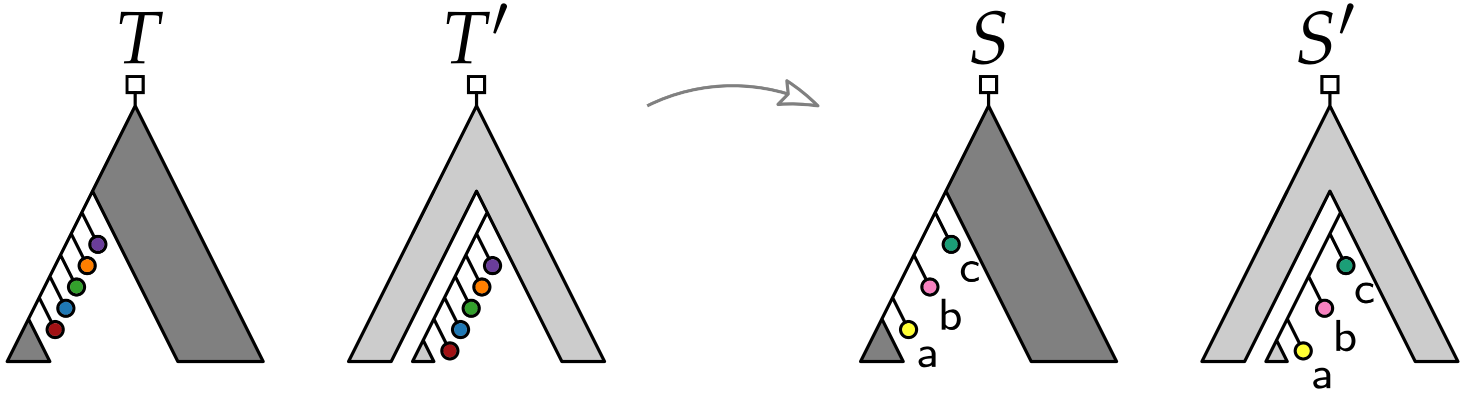
Case 3



# Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.



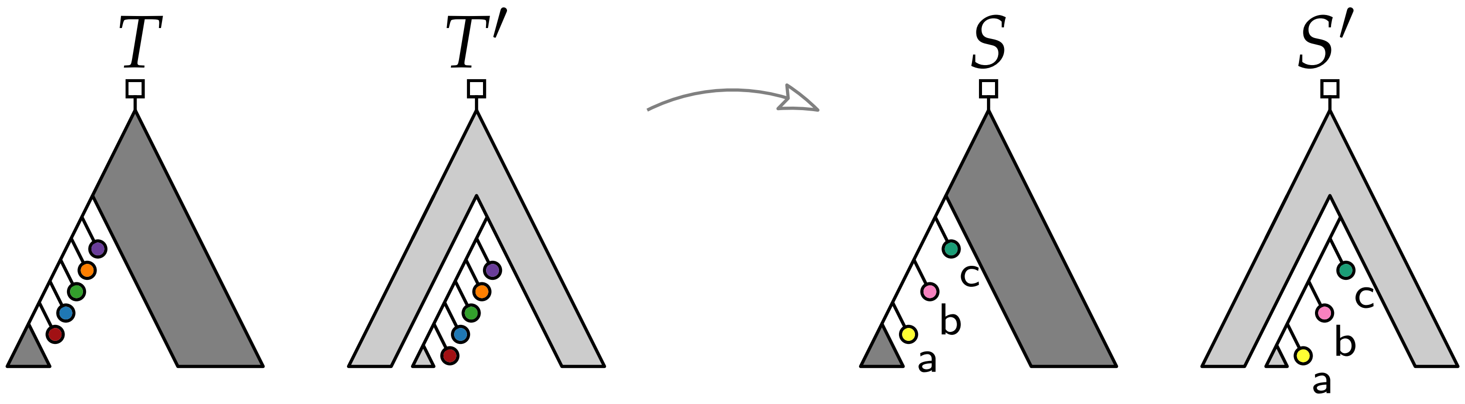

Why not using a chain of length $\leq 2$?

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

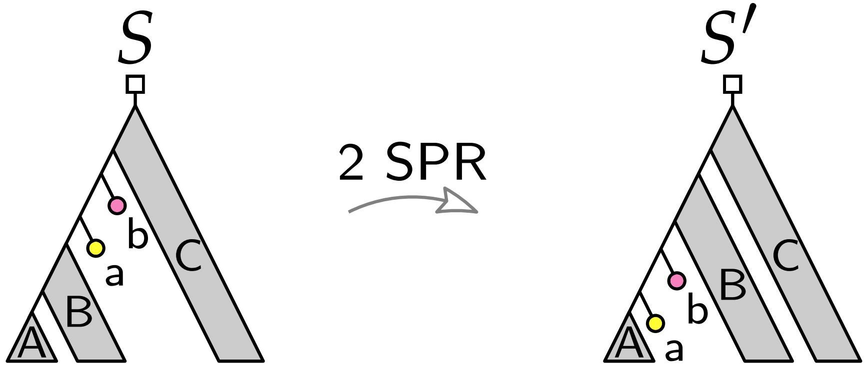
# Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.




Why not using a chain of length $\leq 2$?

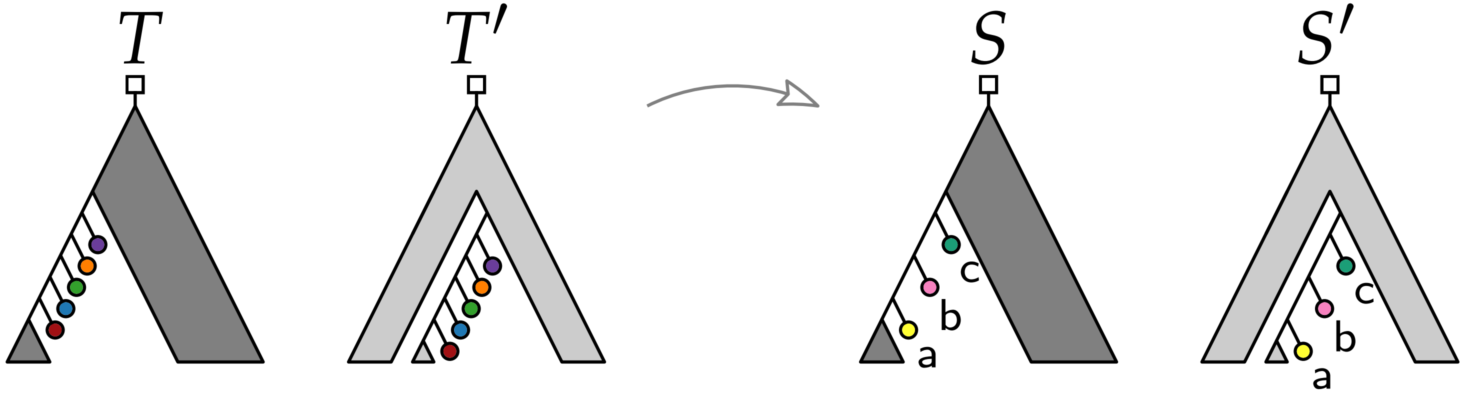



**Lemma 6.** Applying chain reduction is safe, i.e., $\mathsf{d}_{\mathsf{SPR}}(T, T') = \mathsf{d}_{\mathsf{SPR}}(S, S')$.

# Kernelization – Chains

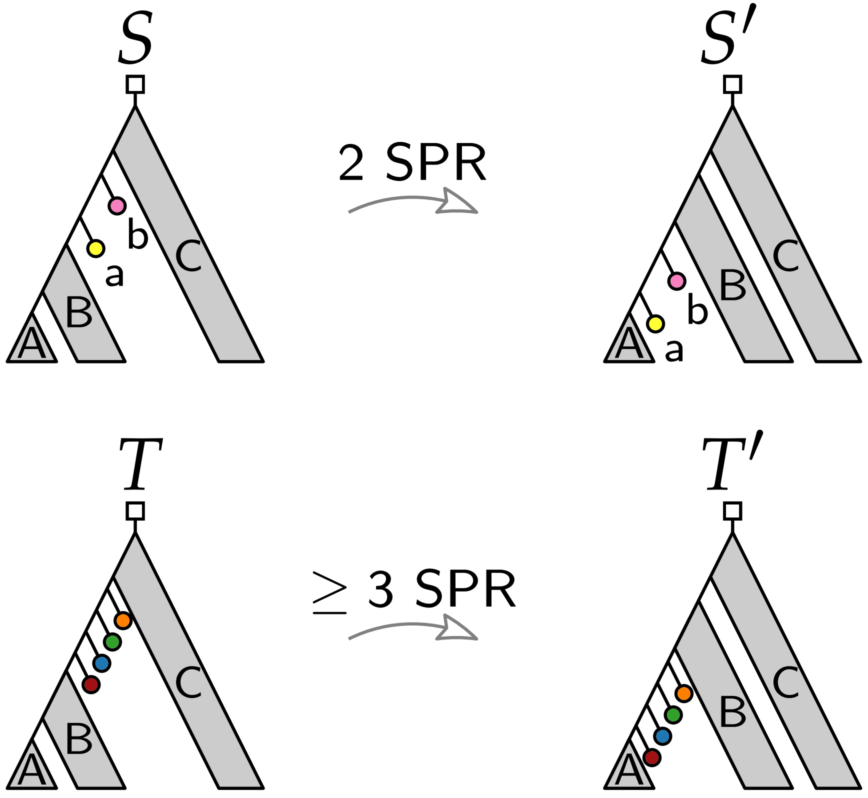
**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.




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

Why not using a chain of length $\leq 2$?



# Kernel Size

**Lemma 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

**Lemma 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 $S'$.

# Kernel Size

**Lemma 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 $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

# Kernel Size

**Lemma 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 $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

# Kernel Size

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

We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

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

Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k}(\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

# Kernel Size

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

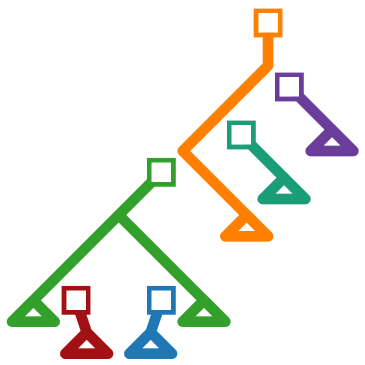
We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

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

Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.



# Kernel Size

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

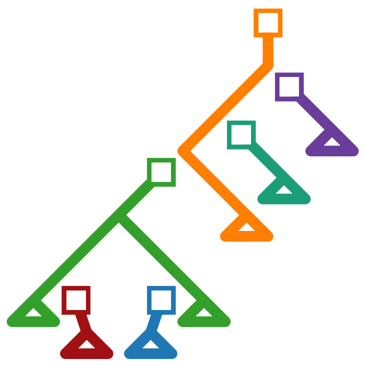
We know $k = \mathsf{d}_{\mathsf{SPR}}(S,S') = \mathsf{d}_{\mathsf{SPR}}(T,T')$.

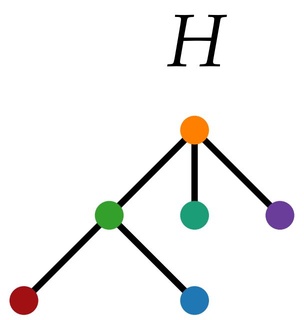
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k}(\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.





# Kernel Size

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

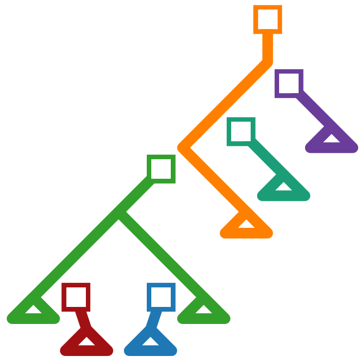
We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

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

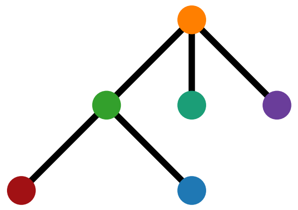
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.



$H$



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

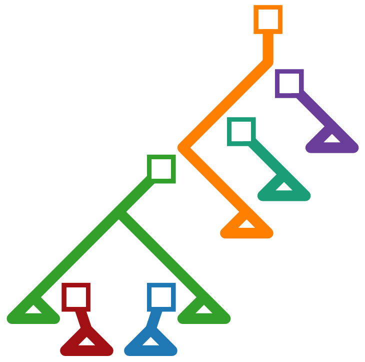
# Kernel Size

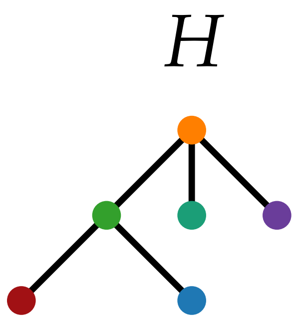
**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28\, \mathrm{d}_{\mathrm{SPR}}(T, T').$$

We know
$k = \mathrm{d}_{\mathrm{SPR}}(S, S') = \mathrm{d}_{\mathrm{SPR}}(T, T')$.

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.
Let $\mathrm{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.
Similarly, let $\mathrm{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k}(\mathrm{n}(T_i) + \mathrm{n}'(T_i)) \leq 4k$.






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

# Kernel Size

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

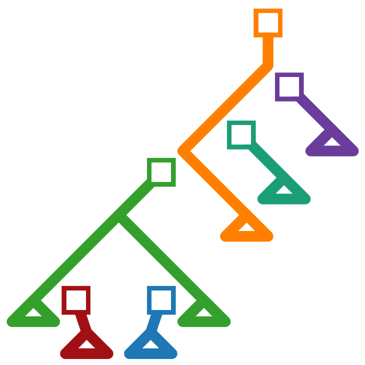
We know
$k = \mathrm{d_{SPR}}(S, S') = \mathrm{d_{SPR}}(T, T')$.

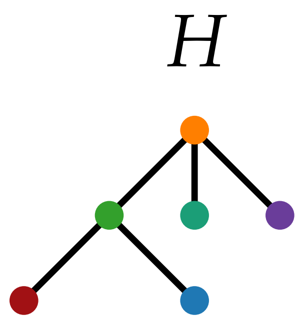
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $\mathrm{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathrm{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathrm{n}(T_i) + \mathrm{n}'(T_i)) \leq 4k$.






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

$$\sum_{i=\rho}^{k} \mathrm{n}(T_i) = 2|E(H)| \leq 2k$$

# Kernel Size

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

We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

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

Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k}(\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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

# Kernel Size

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

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

We know $k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

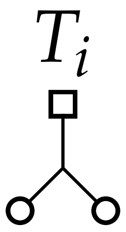
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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



# Kernel Size

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

We know
$k = \mathsf{d_{SPR}}(S, S') = \mathsf{d_{SPR}}(T, T')$.

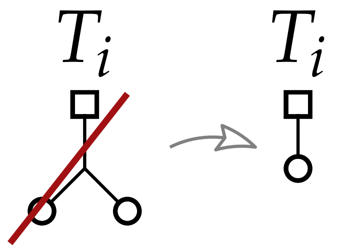
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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



# Kernel Size

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

We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

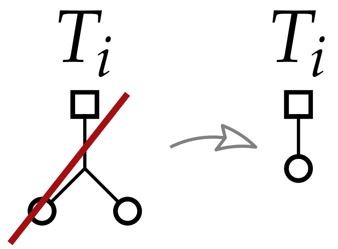
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

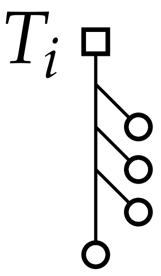
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i$ and $T_j$ touch in $S\}|$.

Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i$ and $T_j$ touch in $S'\}|$.

Claim 1. $\sum_{i=\rho}^{k}(\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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



# Kernel Size

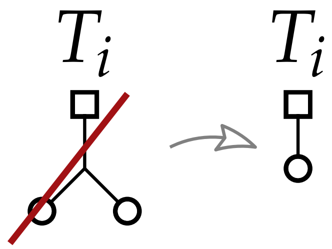
**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28\, \mathrm{d}_{\mathsf{SPR}}(T, T').$$

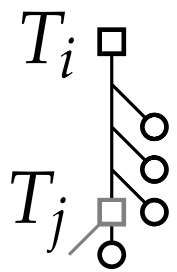
We know
$k = \mathrm{d}_{\mathsf{SPR}}(S, S') = \mathrm{d}_{\mathsf{SPR}}(T, T')$.

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.
Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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



# Kernel Size

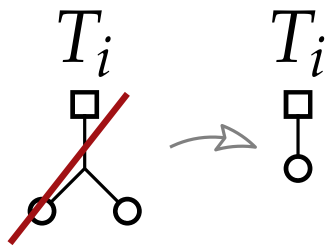
**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28\,\mathrm{d_{SPR}}(T, T').$$

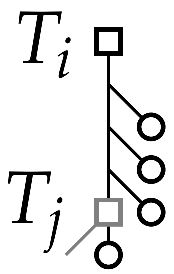
We know
$k = \mathrm{d_{SPR}}(S, S') = \mathrm{d_{SPR}}(T, T')$.

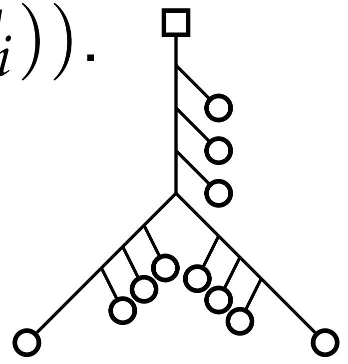
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.
Let $\mathrm{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.
Similarly, let $\mathrm{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathrm{n}(T_i) + \mathrm{n}'(T_i)) \leq 4k$.

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



# Kernel Size

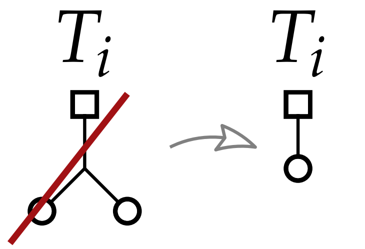
**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28\, \mathsf{d}_{\mathsf{SPR}}(T, T').$$

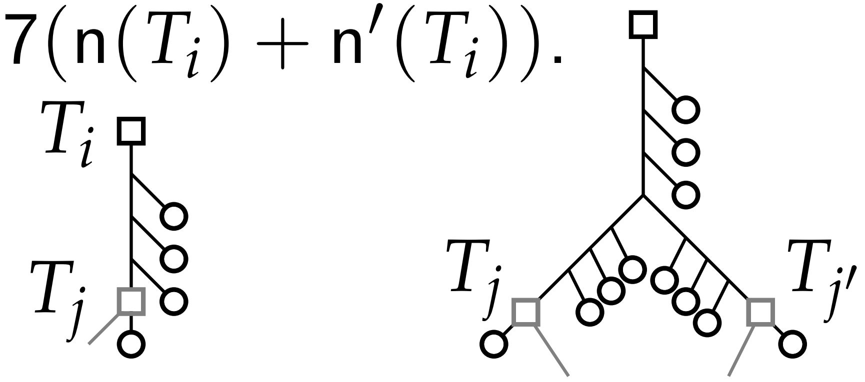
We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

**Proof.** Let $F = \{T_\rho, T_1, \dots, T_k\}$ be MAF for $S$ and $S'$.
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.
Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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



# Kernel Size

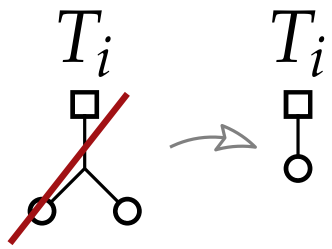
**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28\,\mathrm{d_{SPR}}(T, T').$$

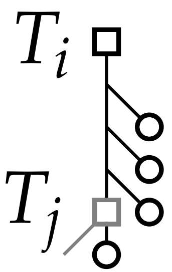
We know
$k = \mathrm{d_{SPR}}(S, S') = \mathrm{d_{SPR}}(T, T')$.

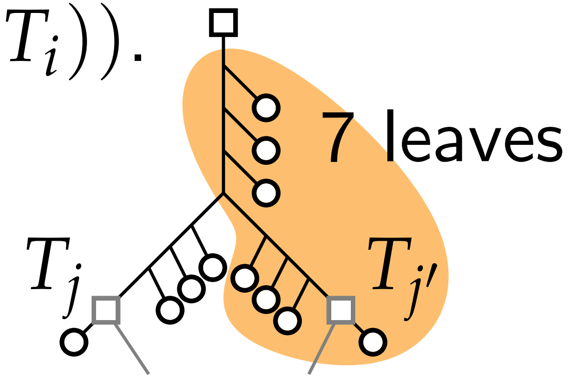
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.
Let $\mathrm{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.
Similarly, let $\mathrm{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k}(\mathrm{n}(T_i) + \mathrm{n}'(T_i)) \leq 4k$.

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



# Kernel Size

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

We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

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

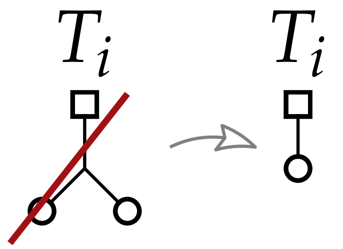
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

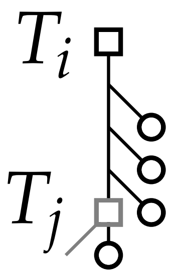
Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

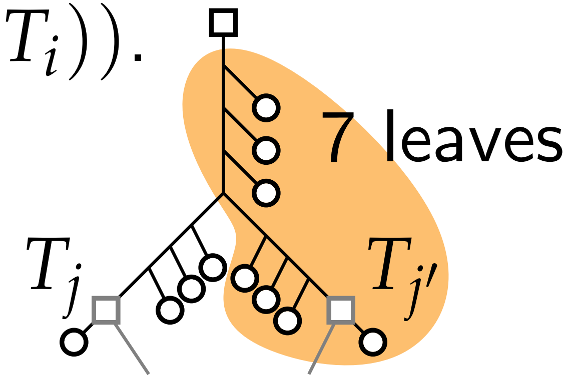
Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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

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



# Kernel Size

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

We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

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

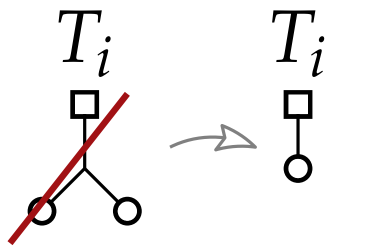
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.

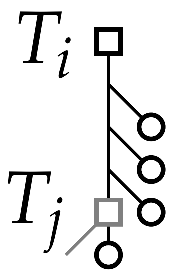
Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

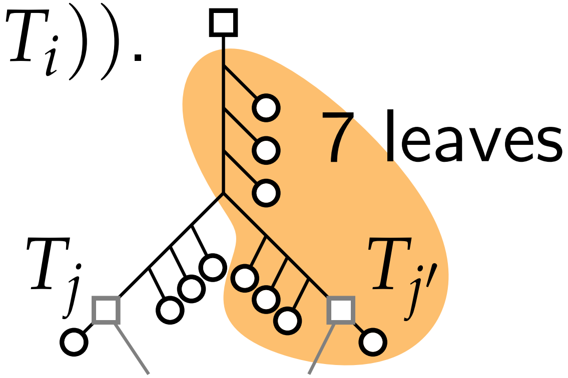
Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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

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



# Kernel Size

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

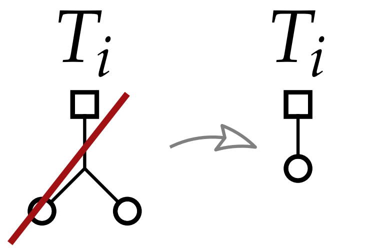
We know
$k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$.

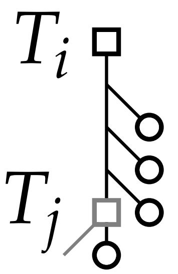
**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.
Let $\mathsf{n}(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S\}|$.
Similarly, let $\mathsf{n}'(T_i) := |\{T_j \mid T_j \in F \wedge T_i \text{ and } T_j \text{ touch in } S'\}|$.

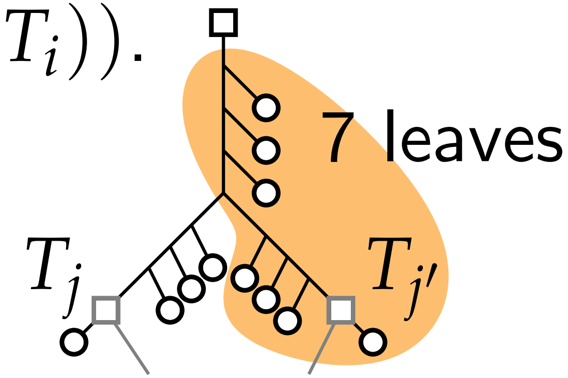
Claim 1. $\sum_{i=\rho}^{k} (\mathsf{n}(T_i) + \mathsf{n}'(T_i)) \leq 4k$.

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

$$|X'| = \sum_{i=\rho}^{k} \# \text{ leaves of } T_i$$
$$\leq \sum_{i=\rho}^{k} 7(\mathsf{n}(T_i) + \mathsf{n}'(T_i))$$
$$\leq 28k$$



# 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 $4|X'|^2$ neighbors in the SPR-graph $G$.

# 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 $4|X'|^2$ neighbors in the SPR-graph $G$.
  - $S$ has less than $2|X'|$ edges to cut and to attach to.

# 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 $4|X'|^2$ neighbors in the SPR-graph $G$.
  - $S$ has less than $2|X'|$ edges to cut and to attach to.
- Length-$k$ BFS from $S$ visits at most $O\left(\left(4|X'|^2\right)^k\right) = O\left((56k)^{2k}\right)$ trees.

# FPT Algorithm

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

**Proof.**

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

by Lemma 7

# 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 $4|X'|^2$ neighbors in the SPR-graph $G$.
  - $S$ has less than $2|X'|$ edges to cut and to attach to.
- Length-$k$ BFS from $S$ visits at most $O\left(\left(4|X'|^2\right)^k\right) = O\left((56k)^{2k}\right)$ trees. (by Lemma 7)
- Since $k = \mathsf{d}_{\mathsf{SPR}}(S, S') = \mathsf{d}_{\mathsf{SPR}}(T, T')$, this yields an FPT algorithm.

# Approximation Algorithm

**Idea.**

- Given trees $T$ and $T'$, which are reduced by the previous rules, we compute an agreement forest $F$ by
- successively making “cuts” and “eliminations”.
- These steps let $T$ and $T'$ shrink further and further.
- Show that $|F|$ is at most $3|F^*|$, where $F^*$ is a MAF of $T$ and $T'$.

# 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**

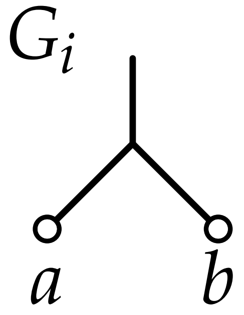
**return** $|H_i| - 1$

# 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**

**return** $|H_i| - 1$

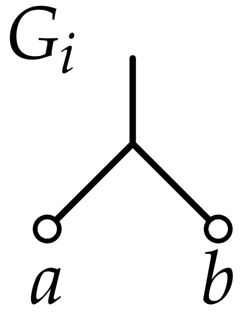


# Approximation Algorithm

$\textsc{approx}\mathrm{DSPR}(T, T')$

$i \leftarrow 1$
$G_i \leftarrow T$
$H_i \leftarrow T'$
**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** $|H_i| - 1$

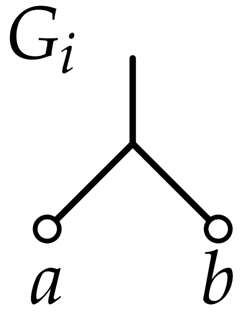


# Approximation Algorithm

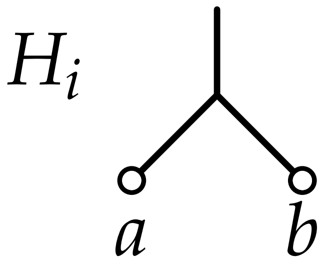
APPROXDSPR$(T, T')$

$i \leftarrow 1$
$G_i \leftarrow T$
$H_i \leftarrow T'$
**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

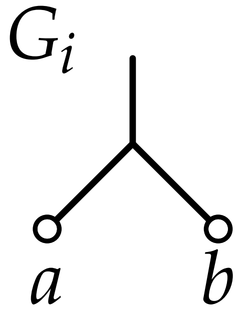


# Approximation Algorithm

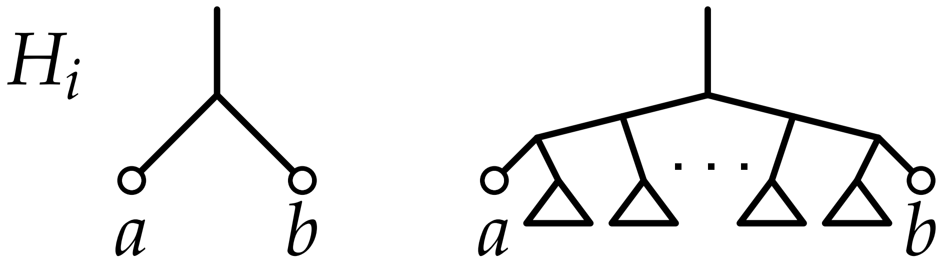
$\textsc{approx}\mathrm{DSPR}(T, T')$

$i \leftarrow 1$
$G_i \leftarrow T$
$H_i \leftarrow T'$
**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

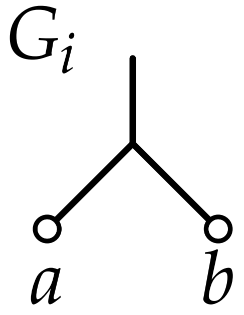


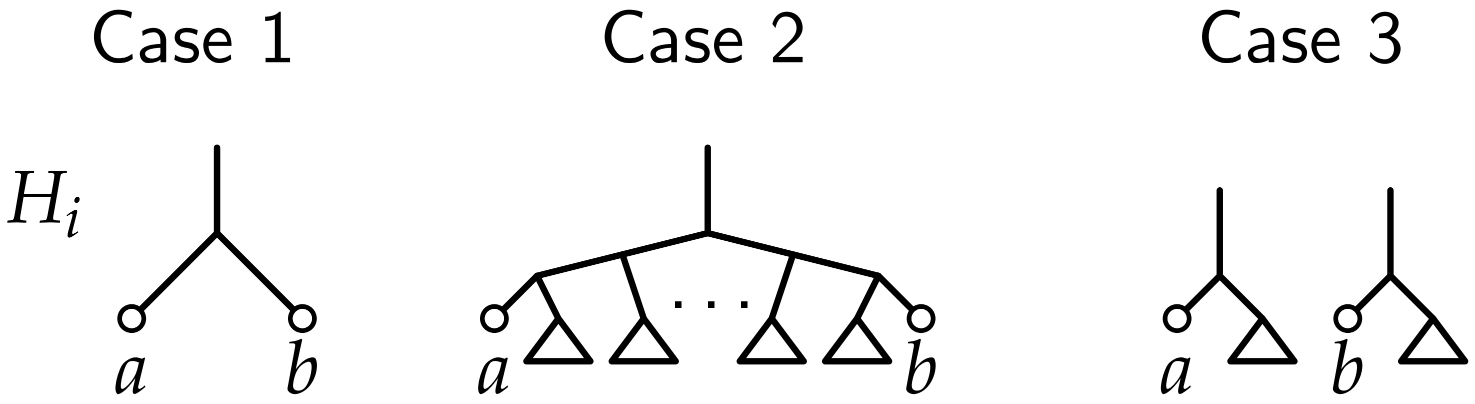
# Approximation Algorithm

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** $|H_i| - 1$



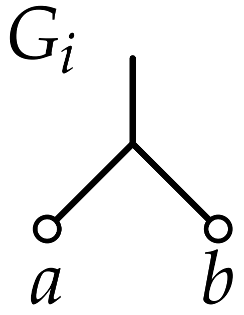



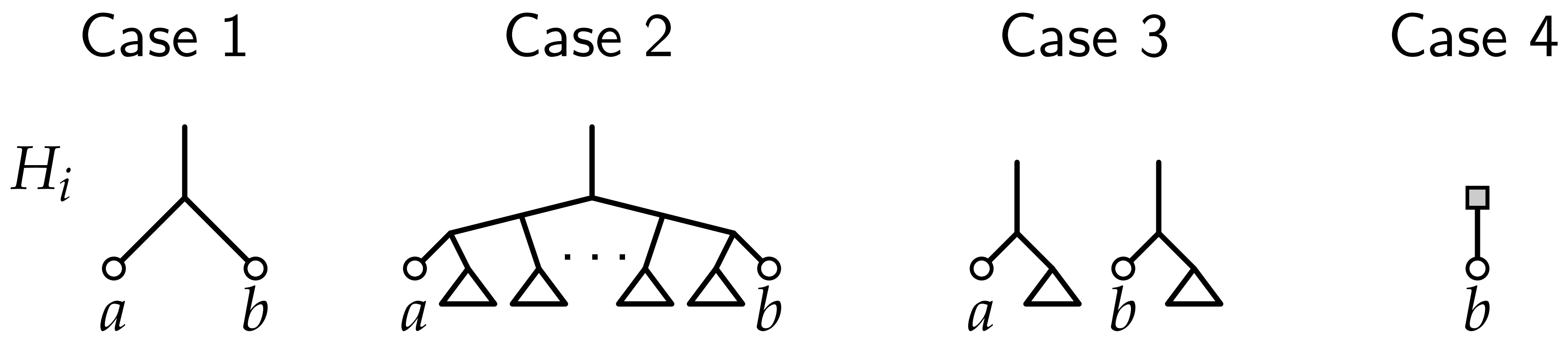
# Approximation Algorithm

$\textsc{approx}\mathrm{DSPR}(T, T')$

$i \leftarrow 1$
$G_i \leftarrow T$
$H_i \leftarrow T'$
**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** $|H_i| - 1$

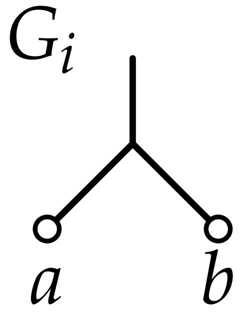


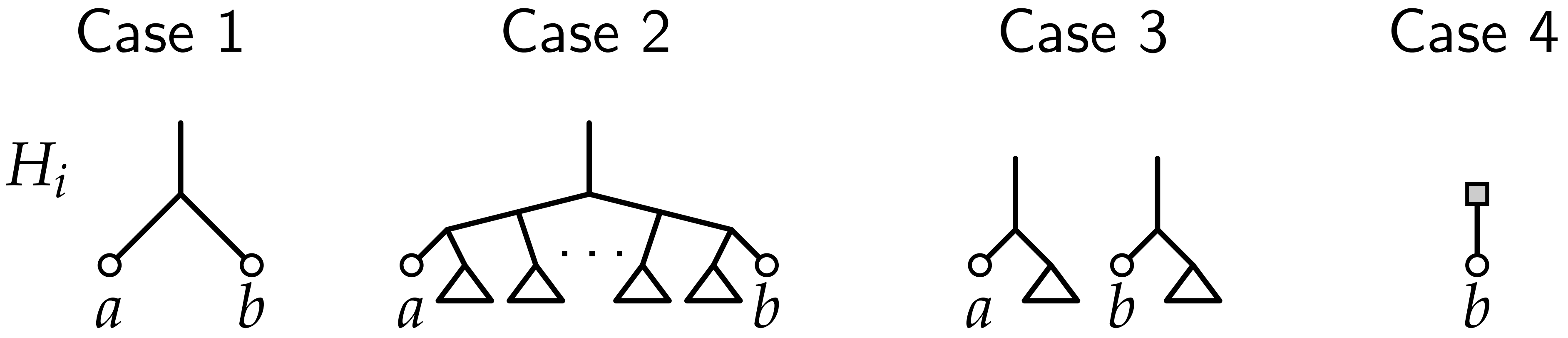



# Approximation Algorithm

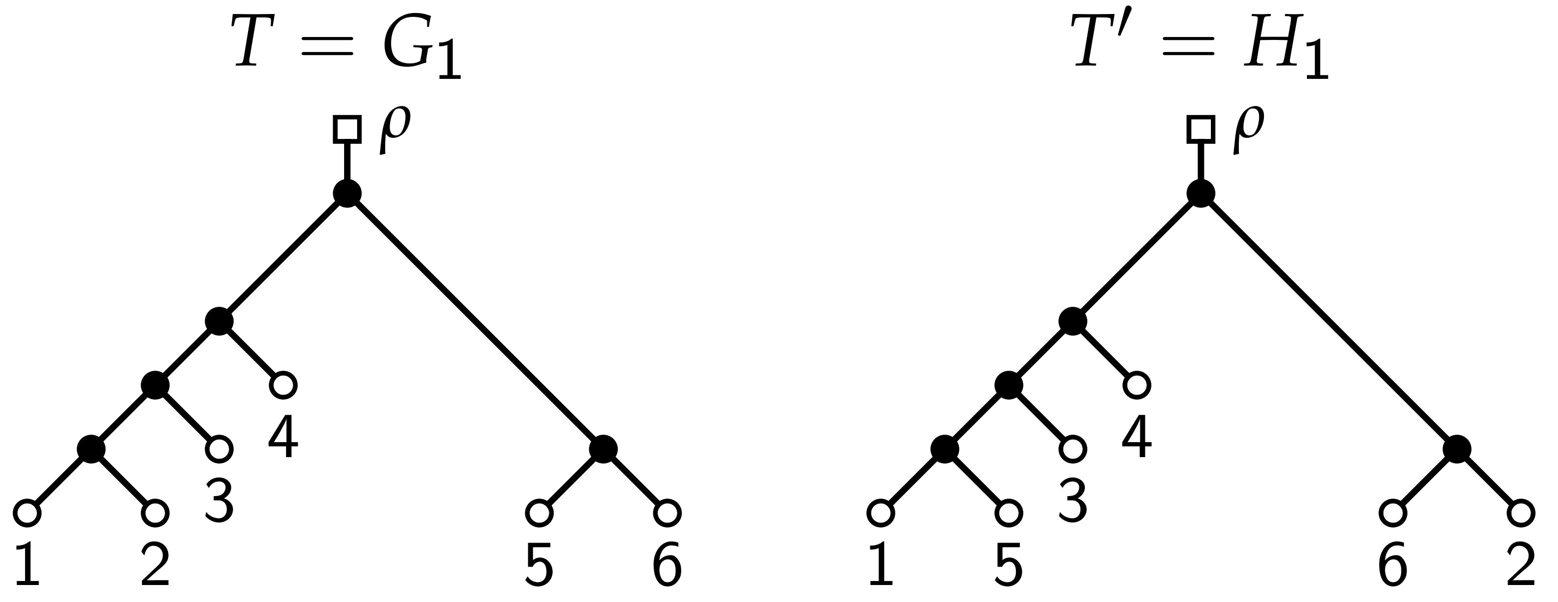
$\textsc{approx}\mathrm{DSPR}(T, T')$

$i \leftarrow 1$
$G_i \leftarrow T$
$H_i \leftarrow T'$
**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 modification
  to obtain $G_{i+1}$ from $G_i$ and $H_{i+1}$ from $H_i$
  $i++$
**return** $|H_i| - 1$

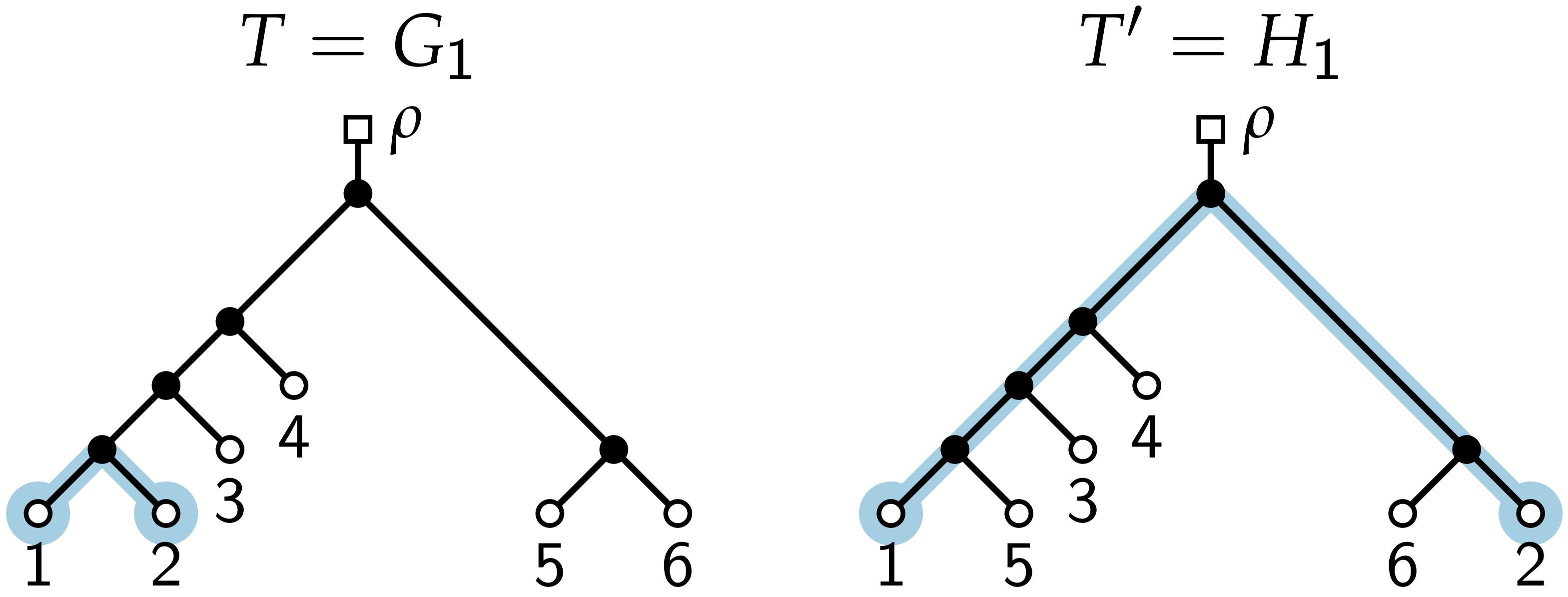


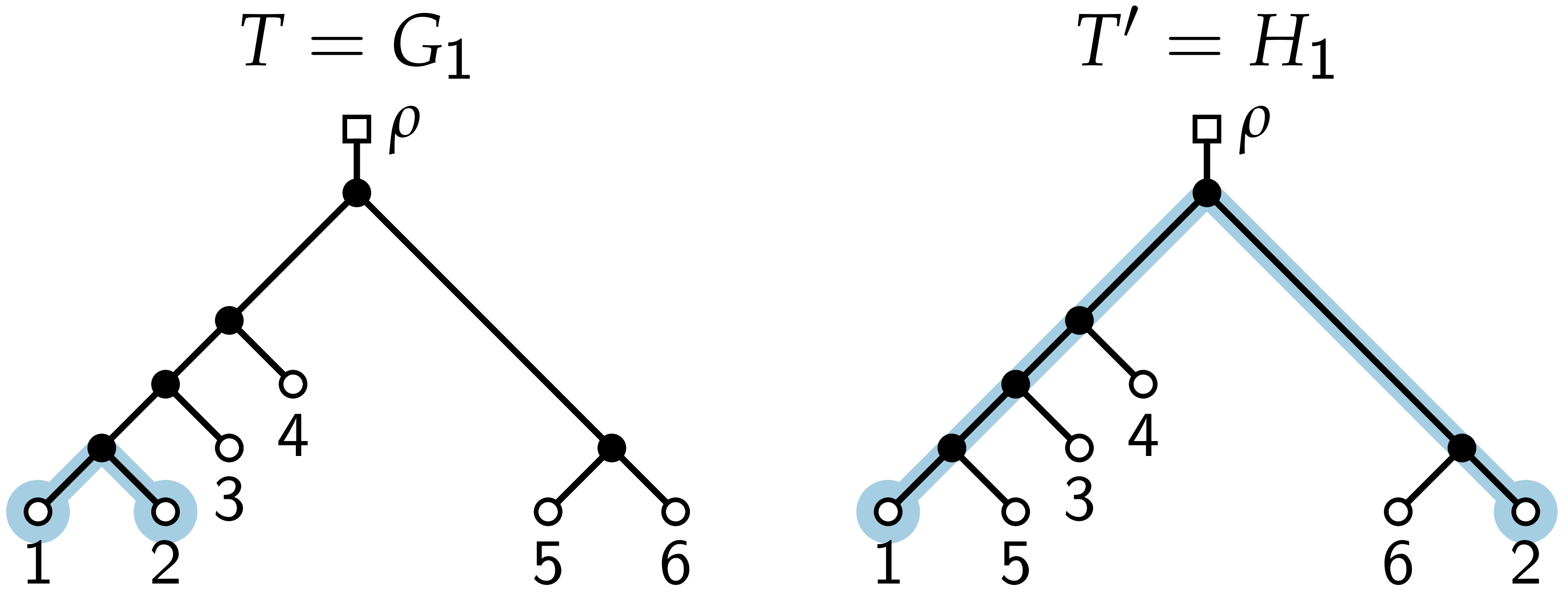
# Approximation Algorithm – Example



# Approximation Algorithm – Example

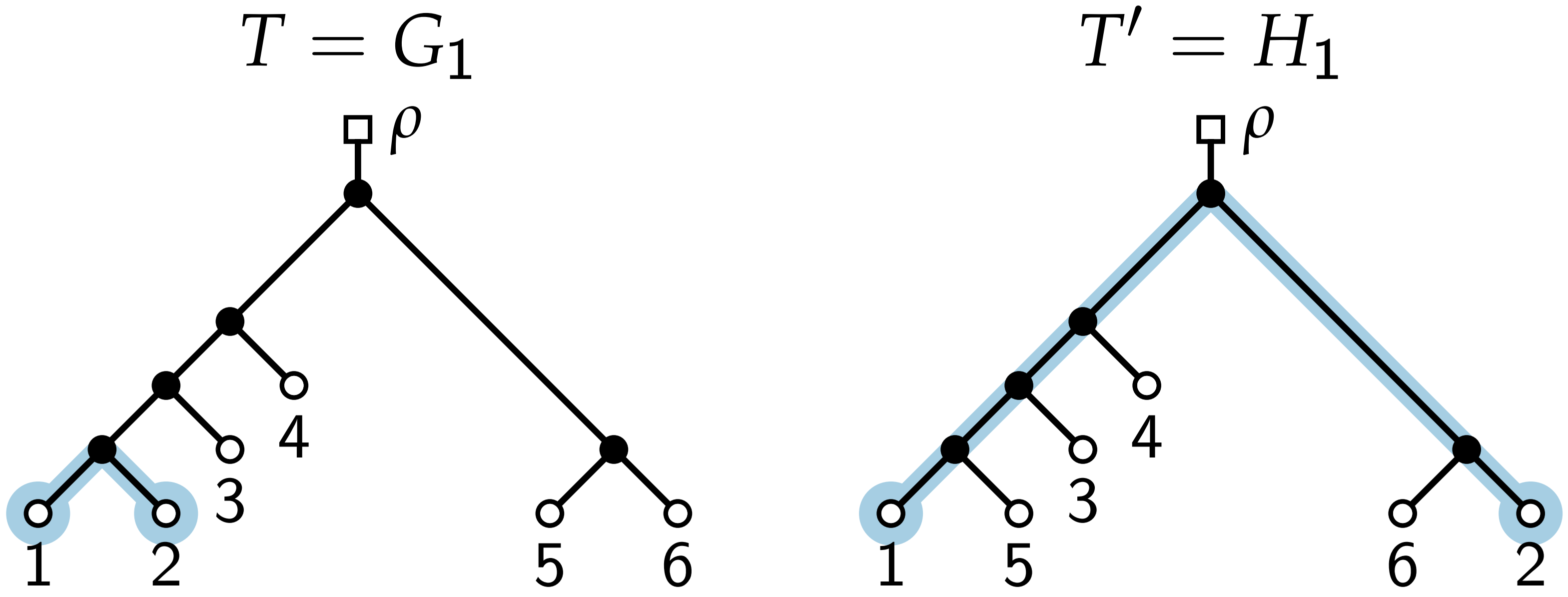


# Approximation Algorithm – Example




- Should we cut off leaf 1 or leaf 2 or everything between them in $H_1$?

# Approximation Algorithm – Example

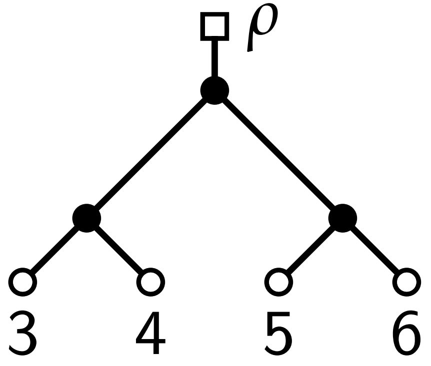



**Case 2**

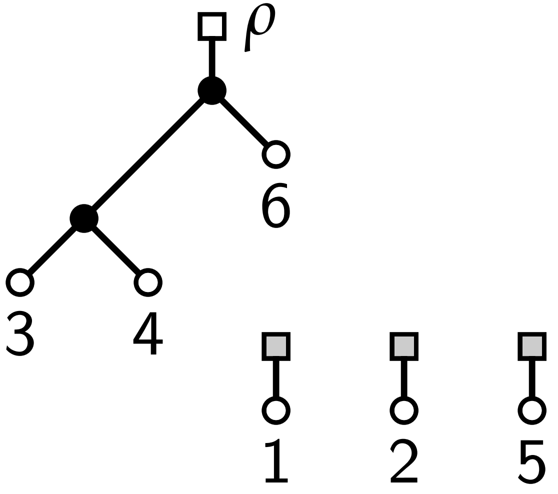
- Should we cut off leaf 1 or leaf 2 or everything between them in $H_1$?
- Do parts of each!

# Approximation Algorithm – Example

$G_2$



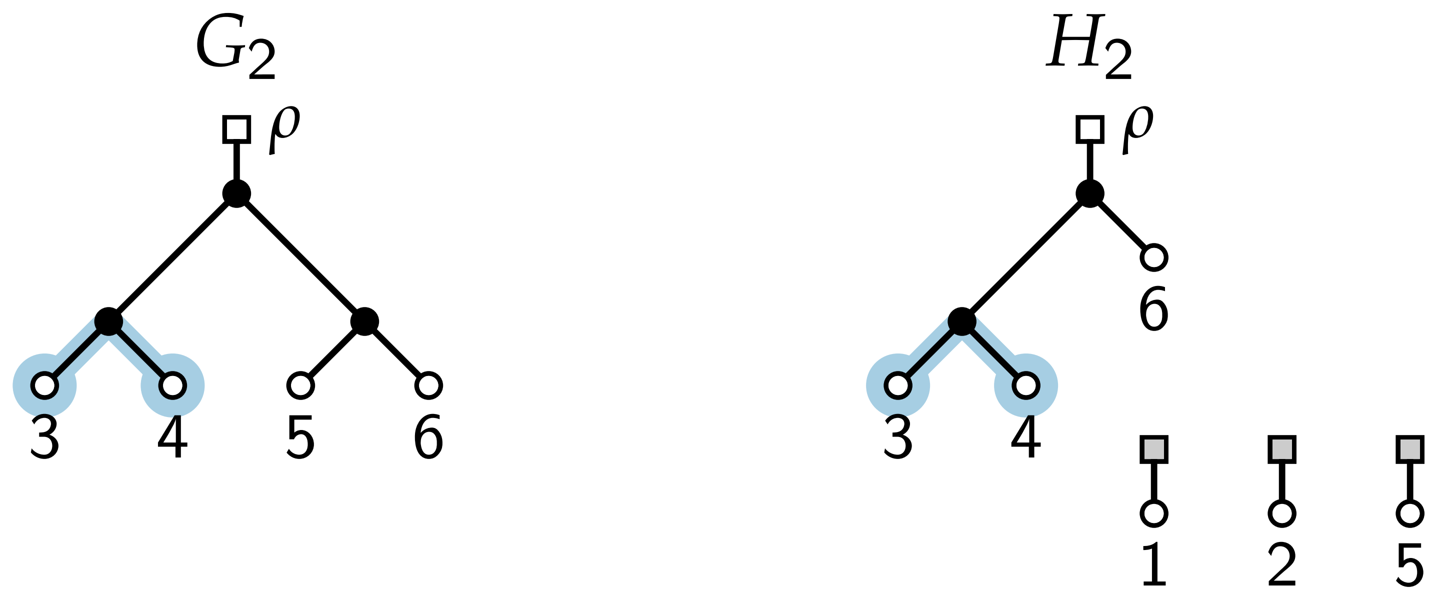

$H_2$



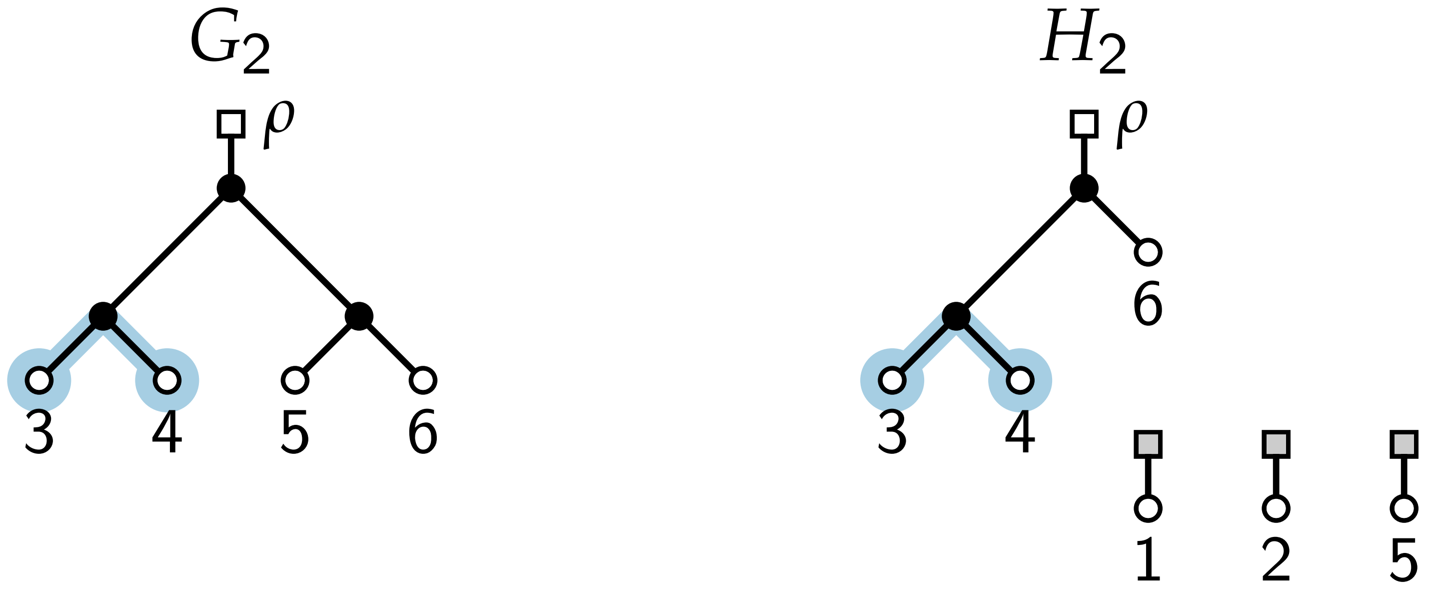

**Case 2**

- Should we cut off leaf 1 or leaf 2 or everything between them in $H_1$?
- Do parts of each!

# Approximation Algorithm – Example



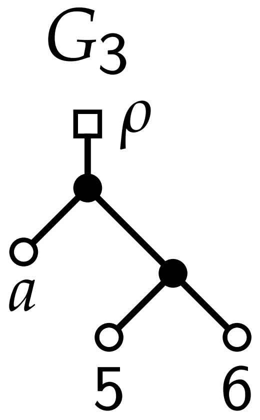
# Approximation Algorithm – Example

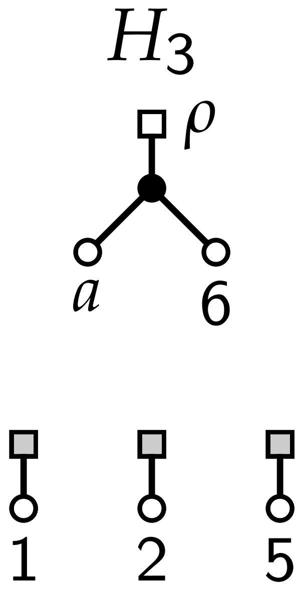



**Case 1**

- If the same "cherry" (i.e., pair of leaves) occurs in $G_i$ and $H_i$, we simply reduce it.

# Approximation Algorithm – Example

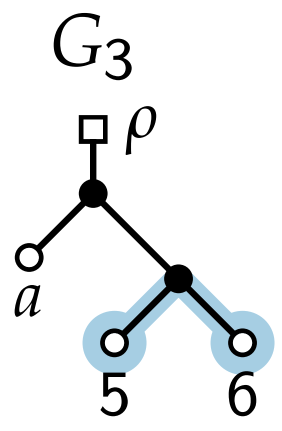


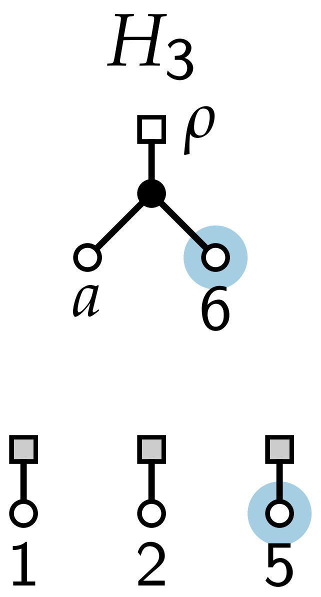




**Case 1**

- If the same "cherry" (i.e., pair of leaves) occurs in $G_i$ and $H_i$, we simply reduce it.

# Approximation Algorithm – Example

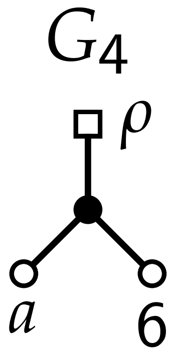


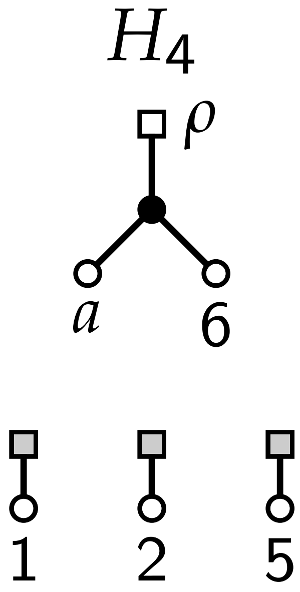




**Case 4**

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

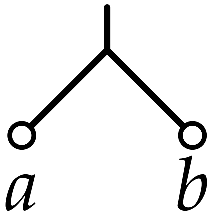
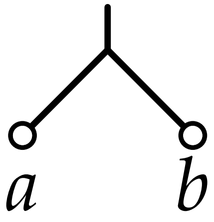
# Approximation Algorithm – Example



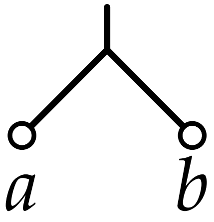
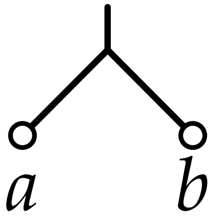
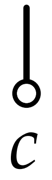




- Return 3.

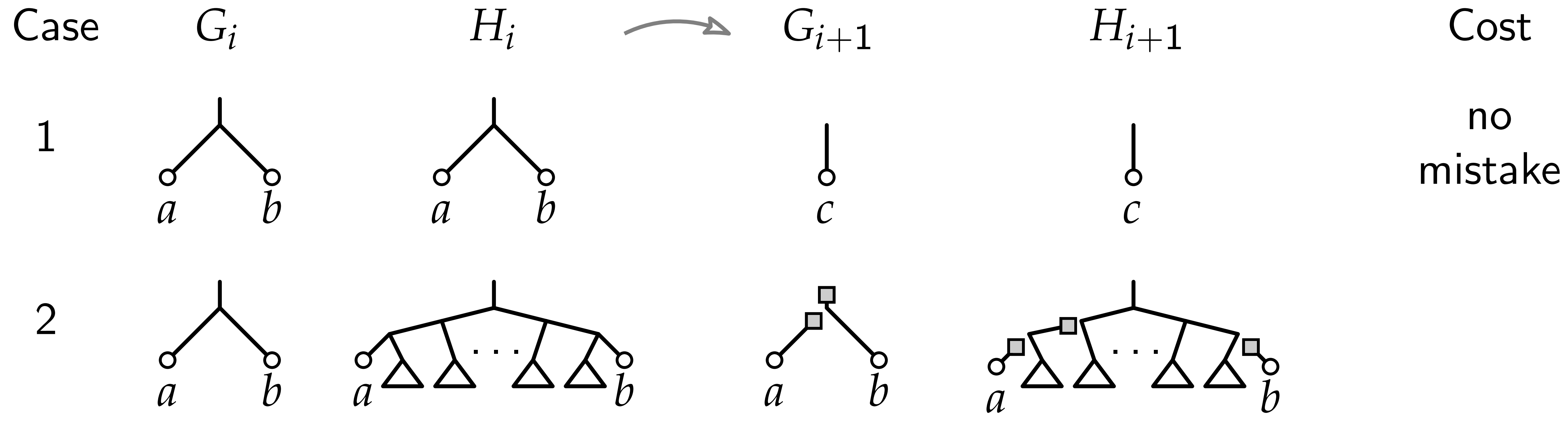
# Approximation Algorithm – Analysis

| Case | $G_i$ | $H_i$ | → | $G_{i+1}$ | $H_{i+1}$ | Cost |
|---|---|---|---|---|---|---|
| 1 |  |  | |  |  | |

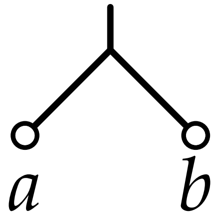
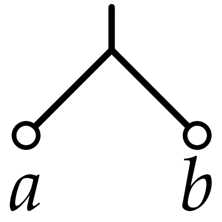
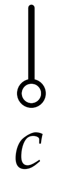
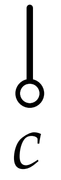
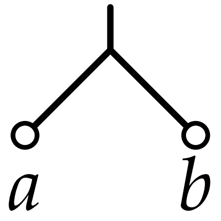
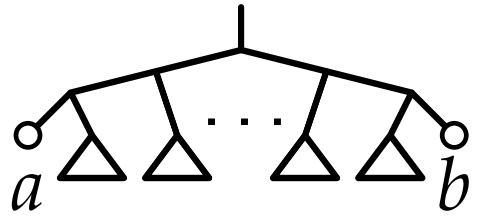
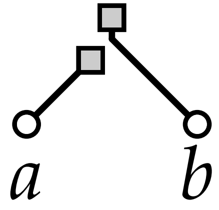
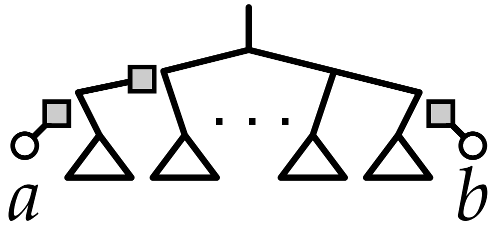
# Approximation Algorithm – Analysis

| Case | $G_i$ | $H_i$ | $G_{i+1}$ | $H_{i+1}$ | Cost |
|---|---|---|---|---|---|
| 1 |   |   |   |   | no mistake |

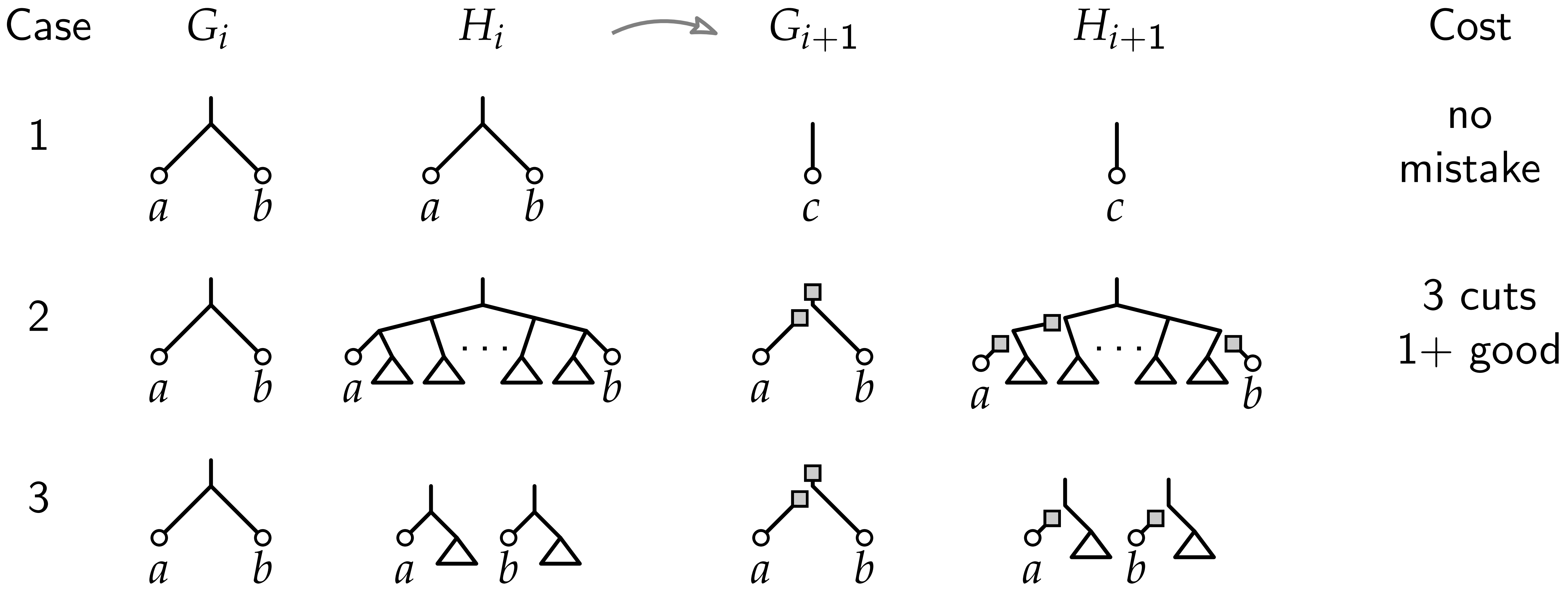
# Approximation Algorithm – Analysis



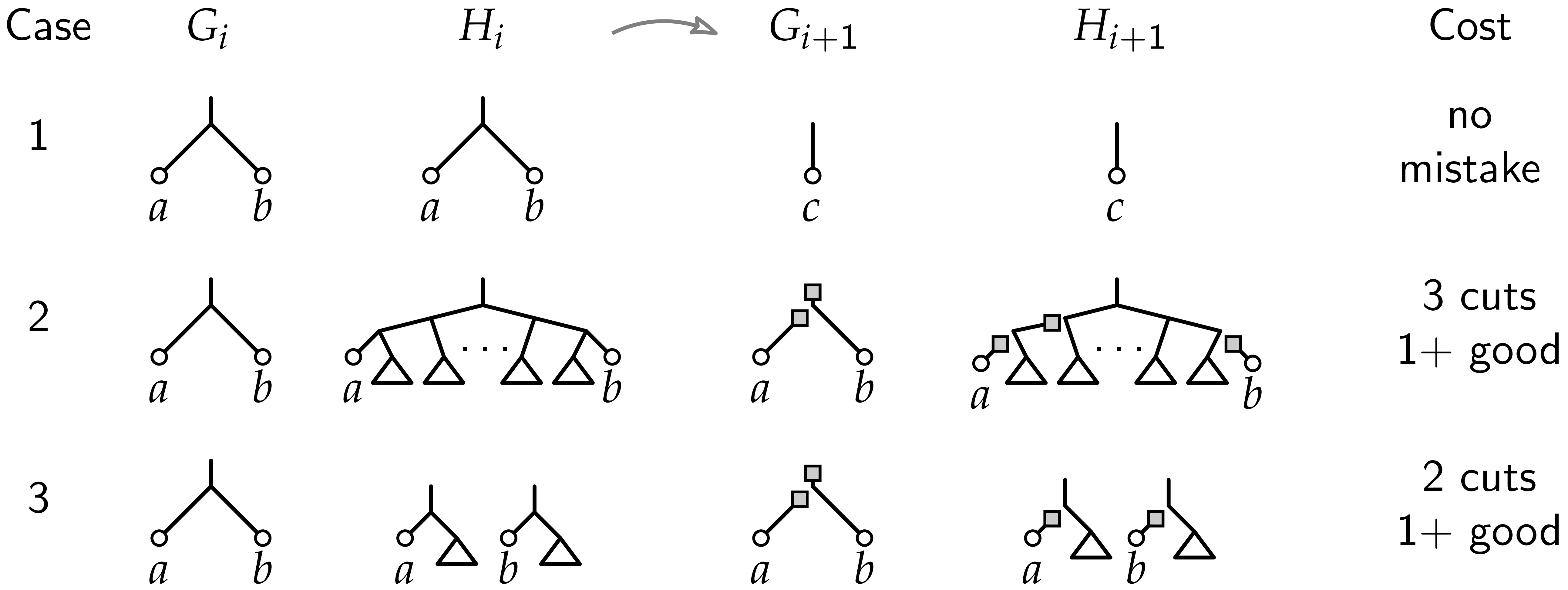
# Approximation Algorithm – Analysis

| Case | $G_i$ | $H_i$ | $G_{i+1}$ | $H_{i+1}$ | Cost |
|---|---|---|---|---|---|
| 1 |  |  |  |  | no mistake |
| 2 |  |  |  |  | 3 cuts<br>1+ good |

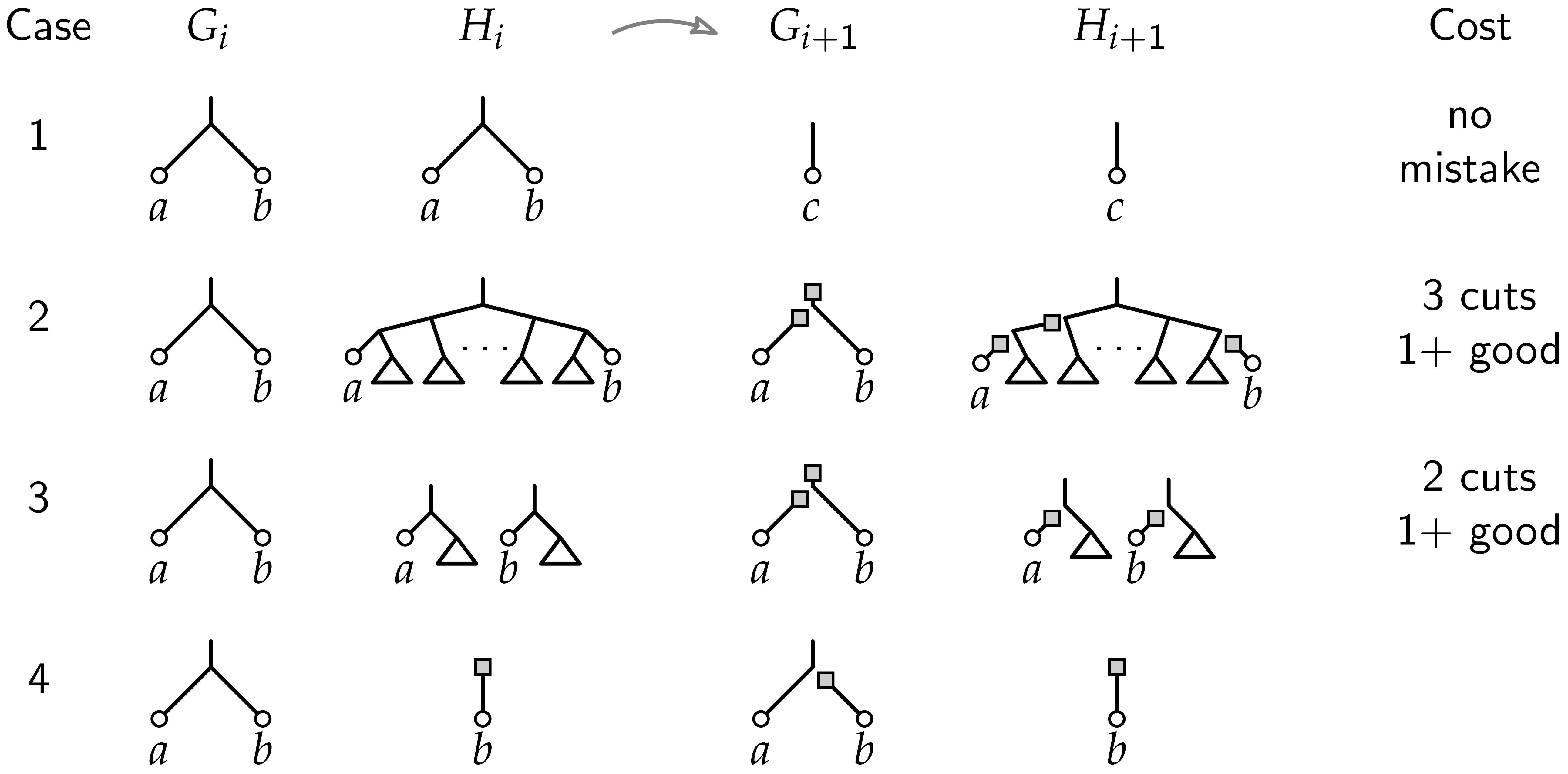
# Approximation Algorithm – Analysis



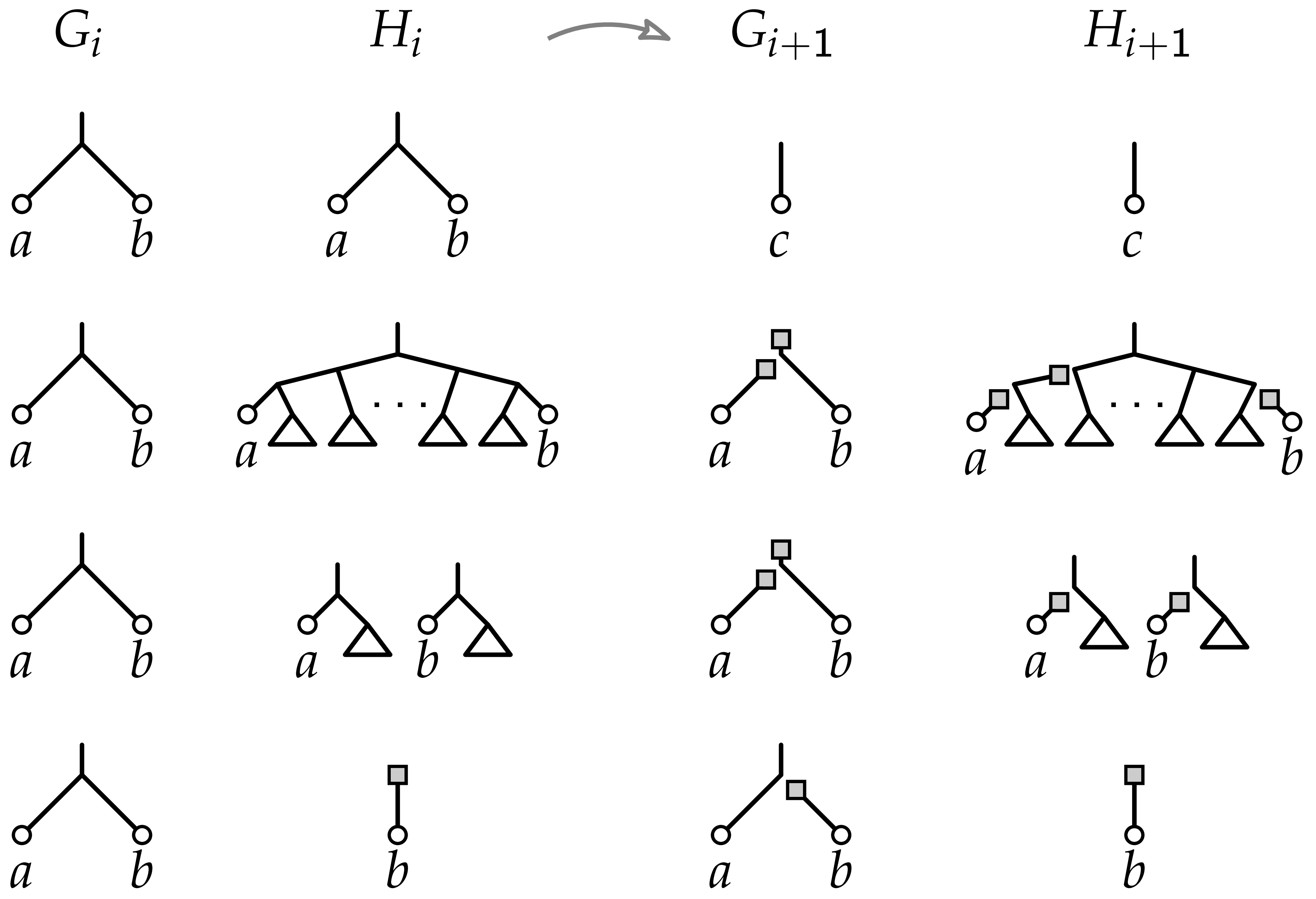
# Approximation Algorithm – Analysis



# Approximation Algorithm – Analysis

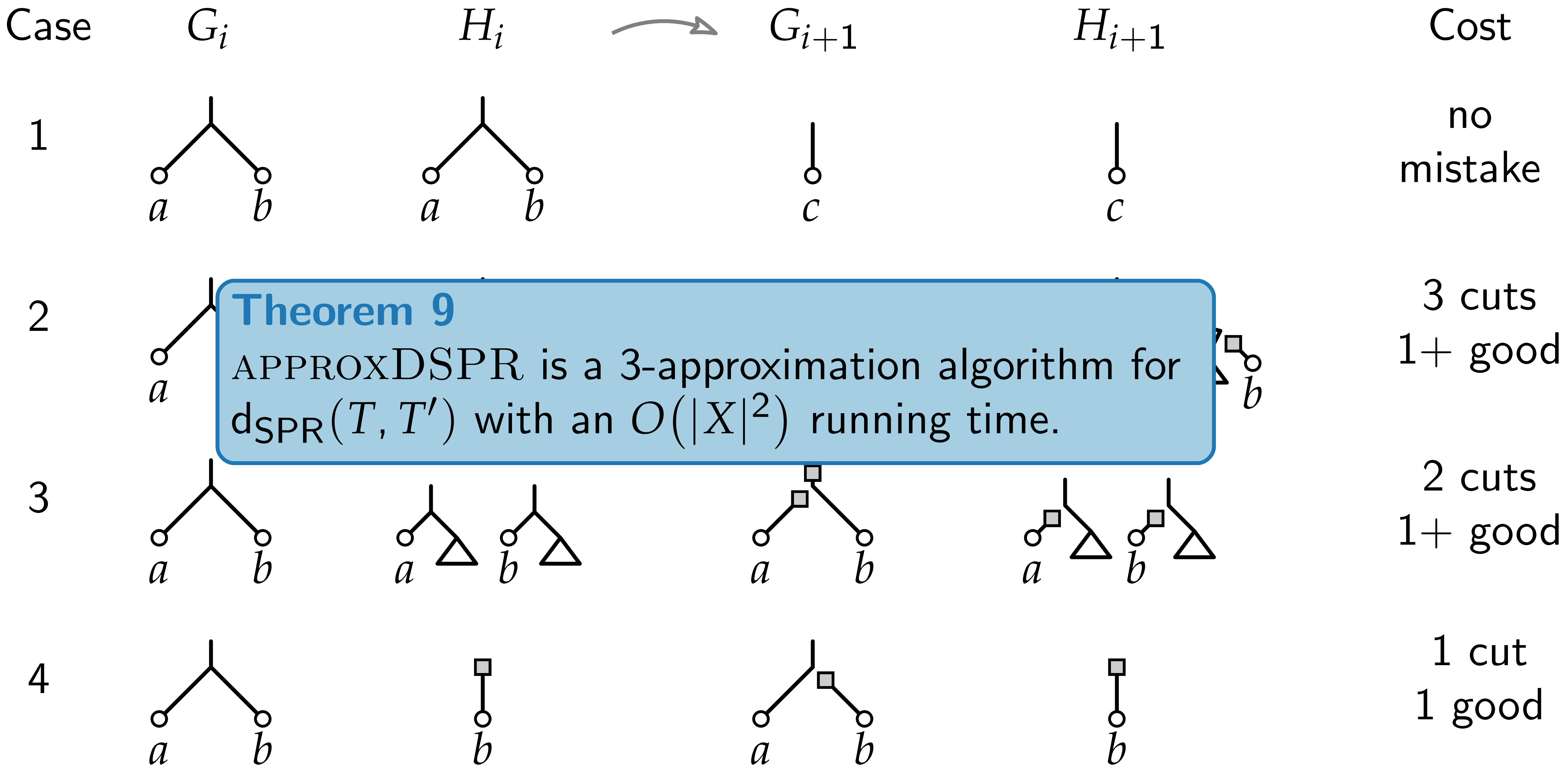


# Approximation Algorithm – Analysis




| Case | Cost |
|---|---|
| 1 | no mistake |
| 2 | 3 cuts<br>1+ good |
| 3 | 2 cuts<br>1+ good |
| 4 | 1 cut<br>1 good |

# Approximation Algorithm – Analysis




**Theorem 9**

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

# 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 $15k - 9$ (we have shown $28k$).
- With further reduction rules, we can get a size below $11k - 9$. [KL '18]
- Divide & conquer techniques can (in practice) further reduce the 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 $15k - 9$ (we have shown $28k$).
- With further reduction rules, we can get a size below $11k - 9$. [KL '18]
- Divide & conquer techniques can (in practice) further reduce the problem sizes. [LS '11]

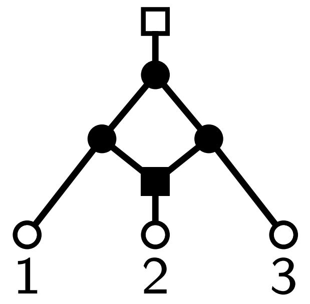
## Approximation algorithm.

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

# Discussion

## Phylogenetic trees.

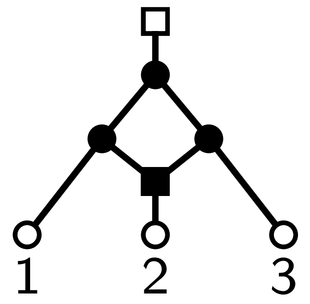
- There are other classes of phylogenetic trees: unrooted, non-binary, ranked, . . .
- Trees can be generalized to **phylogenetic networks**, which can also have indegree 2 outdegree 1 vertices.



# 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 (characterizing) a problem in a different way, can sometimes make your life a lot easier.
- MAF can be generalized to Maximum Agreement Graphs, but these do not characterize the SPR-distance of networks anymore. [K '20]

# Literature

Original papers:


- [BS '05] Semple C., Bordewich M.: *On the computational complexity of the rooted subtree prune and regraft distance* (for SPR, MAF, characterisation, fpt, divide & conquer)
- [HJWZ '96] Hein J., Jiang T., Wang L., Zhang K.: *On the complexity of comparing evolutionary trees* (for NP-hardness proof)
- [RSW '06] Rodrigues E. M., Sagot M.-F., Wakabayashi Y.: *The maximum agreement forest problem: Approximation algorithms and computational experiments* (for approx. algorithm)


Referenced papers:


- [CHW '17] Chen Z., Harada Y., Wang L.: *A new 2-approximation algorithm for rSPR distance*
- [K '20] Klawitter J.: *The agreement distance of unrooted phylogenetic networks*
- [KL '19] Kelk S., Linz. S.: *New reduction rules for the tree bisection and reconnection distance*
- [LS '11] Linz S., Semple C.: *A cluster reduction for computing the subtree distance between phylogenies*