Small phylogeny problem: character evolution trees

Arvind Gupta Ján Maňuch Ladislav Stacho and Chenchen Zhu

School of Computing Science and Department of Mathematics

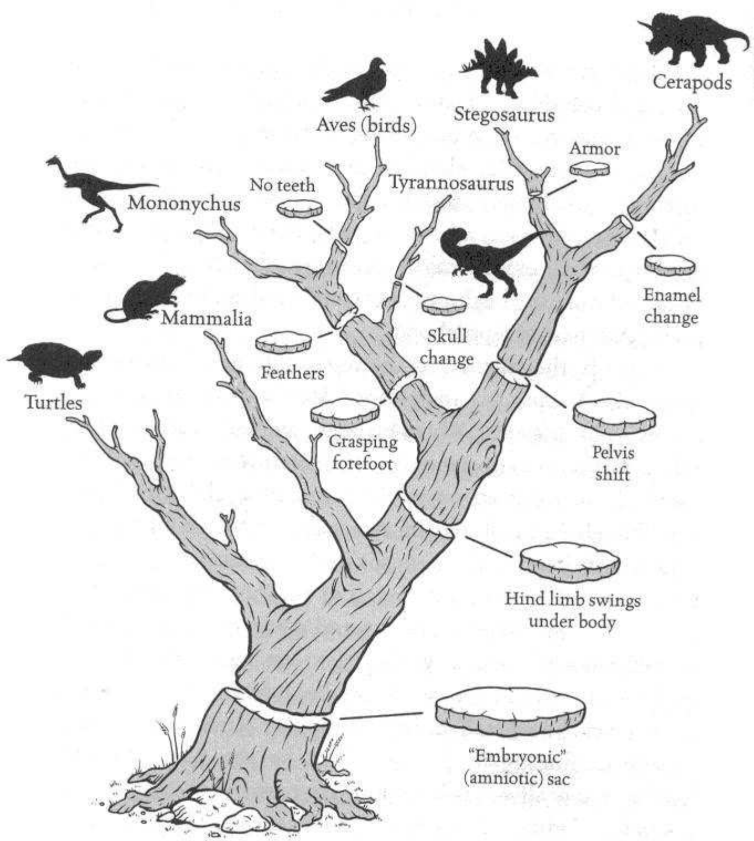
Simon Fraser University, Canada

Phylogenetics

- science determining ancestor/descendent relationships between species

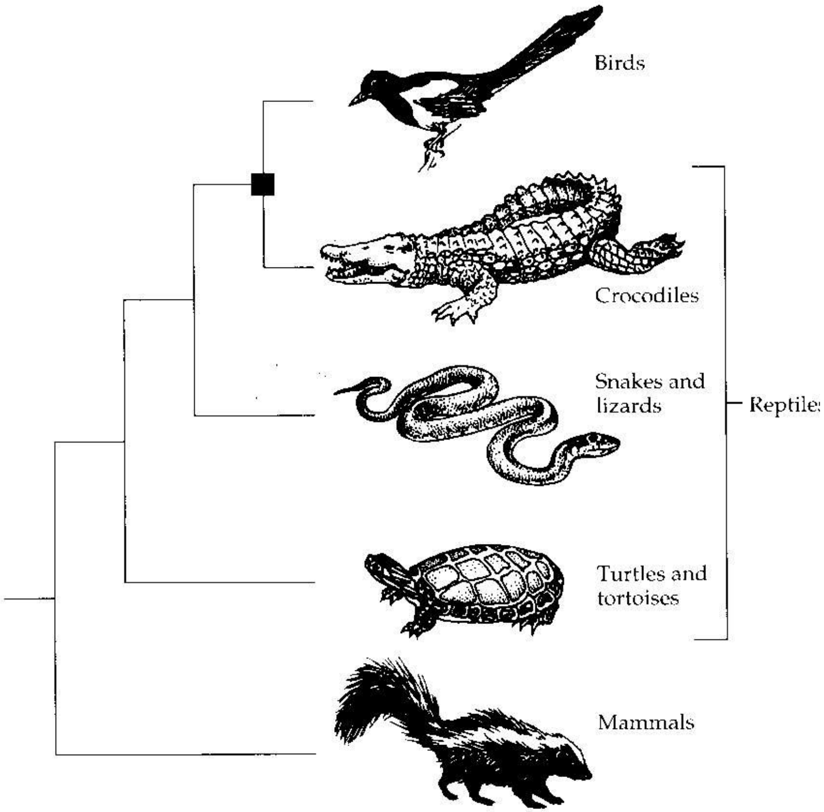
Phylogenetics

- science determining ancestor/descendent relationships between species
- usually expressed by phylogenetic trees



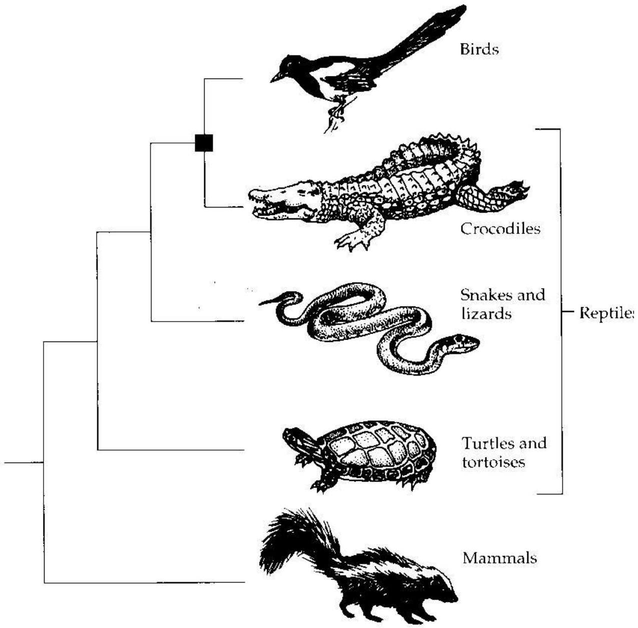
Phylogenetics

- science determining ancestor/descendent relationships between species
- usually expressed by phylogenetic trees



Phylogenetics

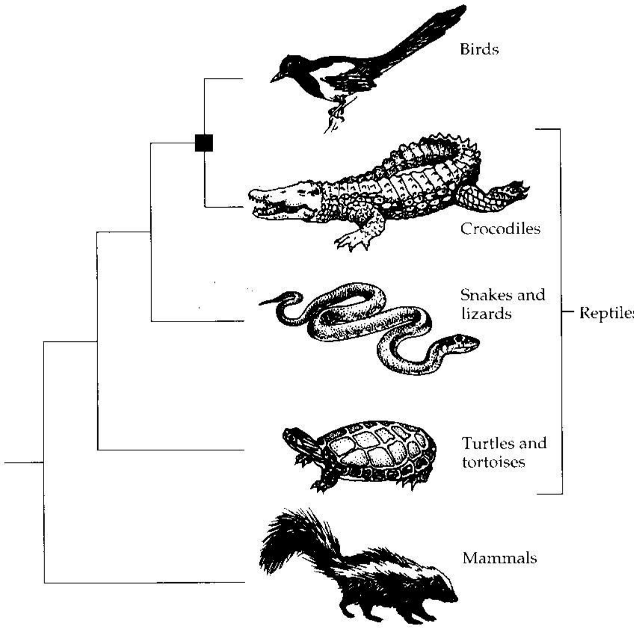
- science determining ancestor/descendent relationships between species
- usually expressed by phylogenetic trees

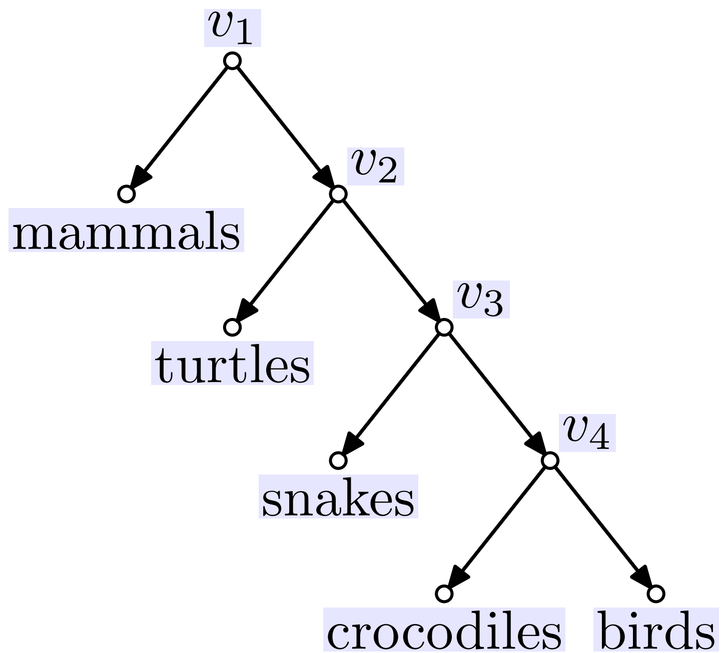


  - the leaves represent extant species
  - internal nodes hypothetical ancestors

Phylogenetics

- science determining ancestor/descendent relationships between species
- usually expressed by phylogenetic trees





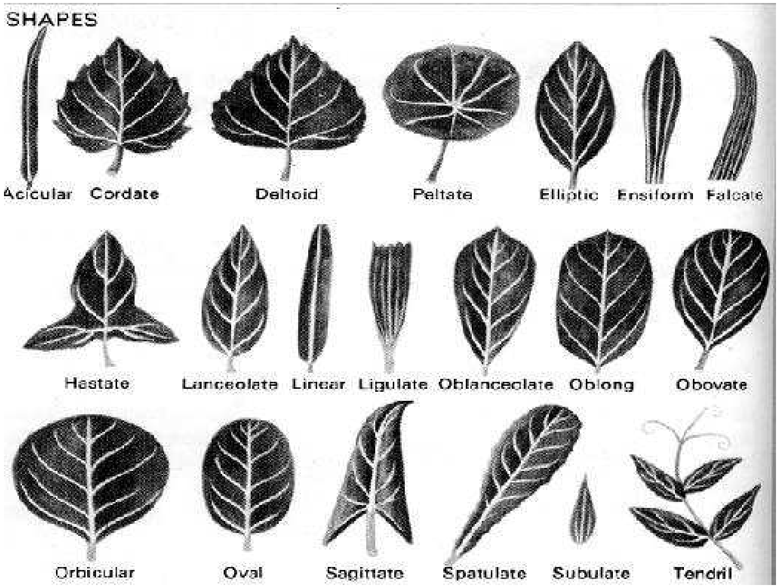
  - the leaves represent extant species
  - internal nodes hypothetical ancestors

Characters

- Principle of parsimony: the goal is to find the tree requiring the smallest number/score of evolutionary transitions (such as the loss of one character, or the modification or gain of another).

Characters

- Principle of parsimony: the goal is to find the tree requiring the smallest number/score of evolutionary transitions (such as the loss of one character, or the modification or gain of another).
- Each character (a morphological feature, a site in a DNA sequence, etc.) takes on one of a few possible states.



Characters

- Principle of parsimony: the goal is to find the tree requiring the smallest number/score of evolutionary transitions (such as the loss of one character, or the modification or gain of another).
- Each character (a morphological feature, a site in a DNA sequence, etc.) takes on one of a few possible states.
- Species can be modeled as vectors of states of a group of characters.

Large phylogeny problem

- **given:**
  - set of characters
  - set of states for each character
  - costs of transitions from one state to another
  - extant species (labeled with states for each character)

Large phylogeny problem

- **given:**
  - set of characters
  - set of states for each character
  - costs of transitions from one state to another
  - extant species (labeled with states for each character)
- **task:**
  - find a *phylogeny tree* and a *labeling of internal nodes* that minimizes cost over all evolutionary steps (*principle of parsimony*)

Large phylogeny problem

- **given:**
  - set of characters
  - set of states for each character
  - costs of transitions from one state to another
  - extant species (labeled with states for each character)
- **task:**
  - find a *phylogeny tree* and a *labeling of internal nodes* that minimizes cost over all evolutionary steps (*principle of parsimony*)
- This problem is **NP-hard** [Foulds, Graham (1982)].

Small phylogeny problem

- **given:**
  - set of characters
  - set of states for each character
  - costs of transitions from one state to another
  - extant species (labeled with states for each character)
  - structure of phylogeny tree (extant species are leaves of the tree)

Small phylogeny problem

- **given:**
  - set of characters
  - set of states for each character
  - costs of transitions from one state to another
  - extant species (labeled with states for each character)
  - structure of phylogeny tree (extant species are leaves of the tree)
- **task:**
  - find a *labeling of internal nodes* that minimizes cost over all evolutionary steps

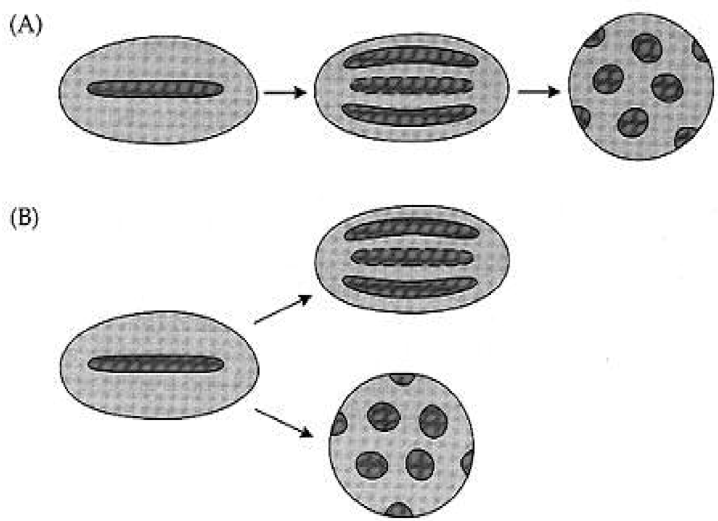
Small phylogeny problem

- **given:**
  - set of characters
  - set of states for each character
  - costs of transitions from one state to another
  - extant species (labeled with states for each character)
  - structure of phylogeny tree (extant species are leaves of the tree)
- **task:**
  - find a *labeling of internal nodes* that minimizes cost over all evolutionary steps
- There are polynomial algorithms: [Fitch (1971)] (uniform costs), [Sankoff (1975)] (non-uniform costs).

Character evolution tree

- So far we assumed that during one evolutionary step one state of a character can change to any other state. However, for many characters **character state order** and **character state polarity** can be observed.

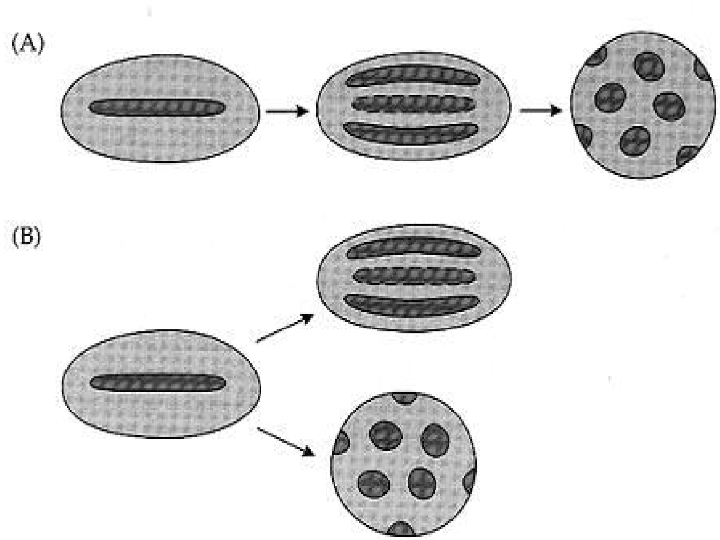
  Example: character evolution trees for pollen

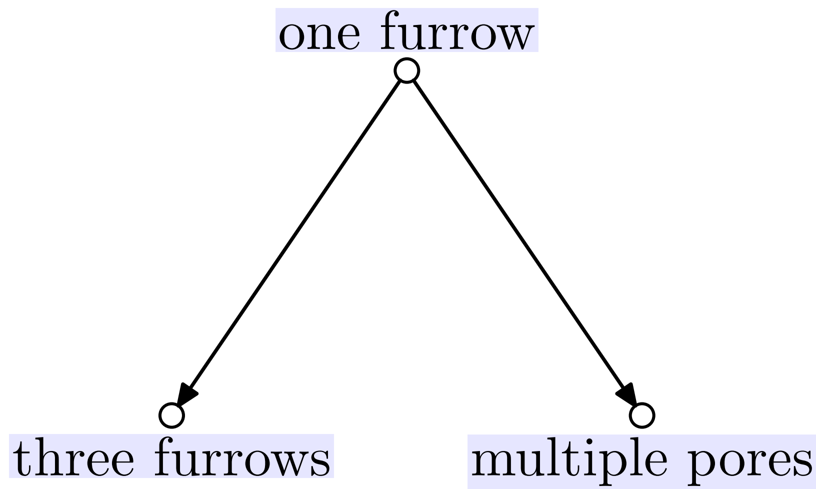


Character evolution tree

- So far we assumed that during one evolutionary step one state of a character can change to any other state. However, for many characters **character state order** and **character state polarity** can be observed.

  Example: character evolution trees for pollen

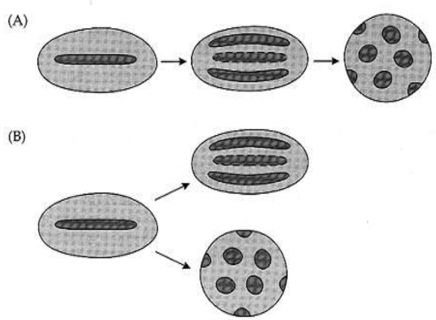


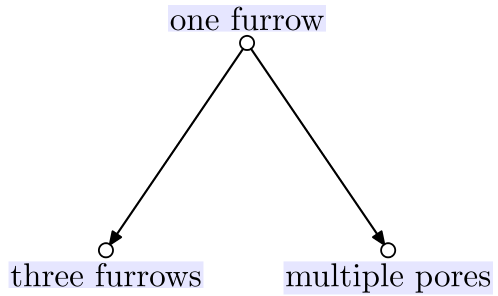


Character evolution tree

- So far we assumed that during one evolutionary step one state of a character can change to any other state. However, for many characters **character state order** and **character state polarity** can be observed.

  Example: character evolution trees for pollen

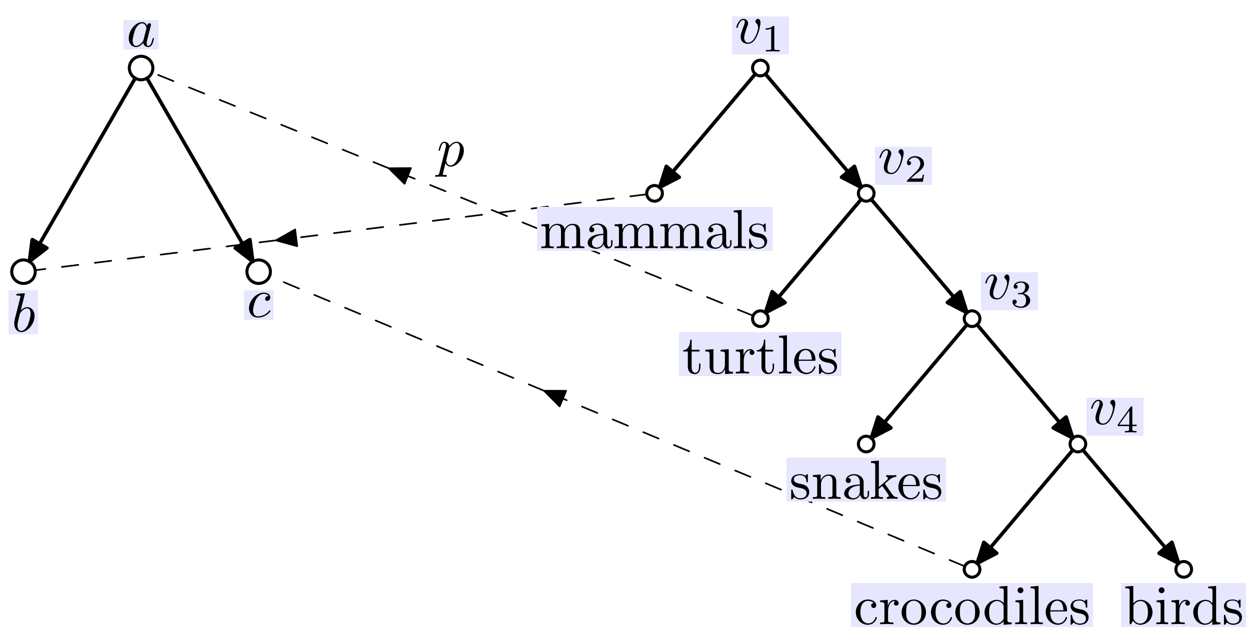




- Our goal is to find a method of directly comparing a character evolution trees with a phylogenetic trees.

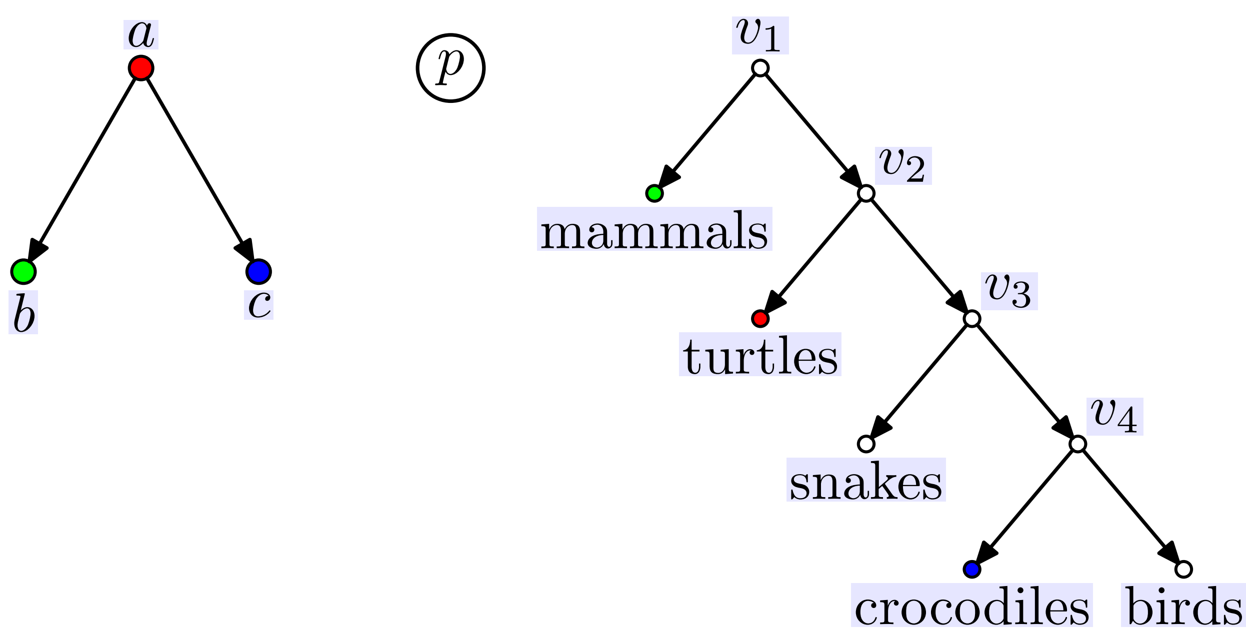
Small phylogeny with character evolution

- **given:**
  - character evolution tree H_h with $V(H)$ being states of the character
  - a phylogeny tree G_g with leaves $L(G)$ being extant species
  - a *leaf labeling* $p : L(G) \to V(H)$ (a partial function)



Small phylogeny with character evolution

- **given:**
  - character evolution tree H_h with $V(H)$ being states of the character
  - a phylogeny tree G_g with leaves $L(G)$ being extant species
  - a *leaf labeling* $p: L(G) \to V(H)$ (a partial function)

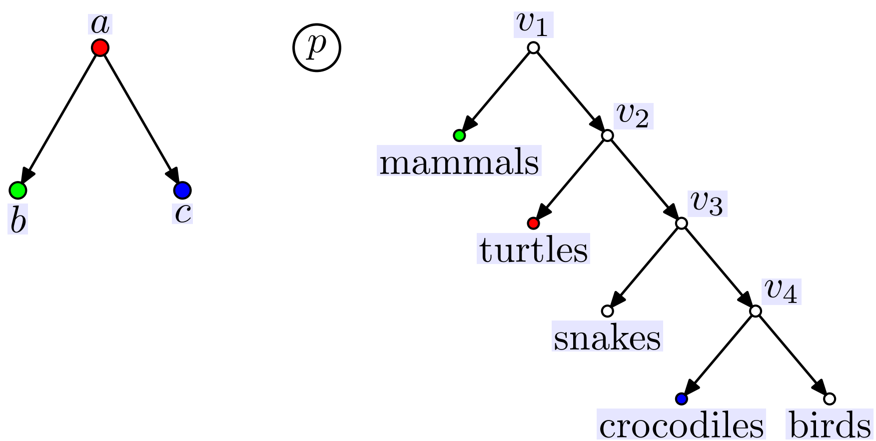


Small phylogeny with character evolution

- **given:**
  - character evolution tree H_h with $V(H)$ being states of the character
  - a phylogeny tree G_g with leaves $L(G)$ being extant species
  - a *leaf labeling* $p: L(G) \rightarrow V(H)$ (a partial function)
- **task:**
  - find a *labeling* $l: V(G) \rightarrow V(H)$ which is:
    - p-constrained

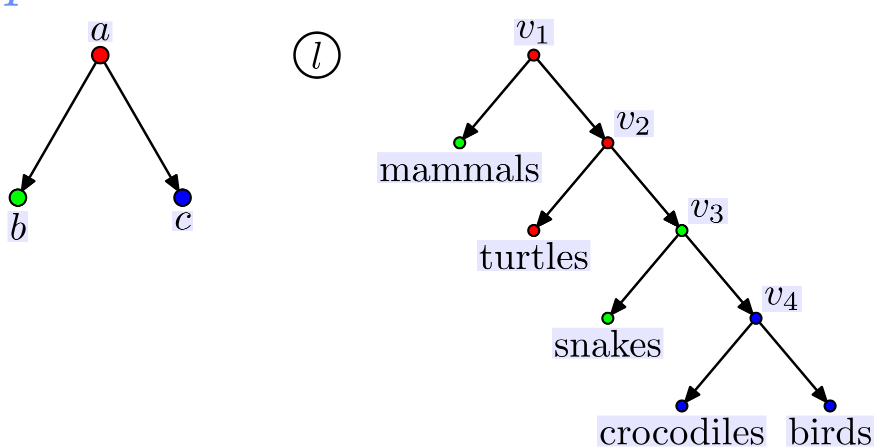
Small phylogeny with character evolution

- **given:**
  - character evolution tree H_h with $V(H)$ being states of the character
  - a phylogeny tree G_g with leaves $L(G)$ being extant species
  - a *leaf labeling* $p: L(G) \to V(H)$ (a partial function)
- **task:**
  - find a *labeling* $l: V(G) \to V(H)$ which is:
    - p-constrained



Small phylogeny with character evolution

- **given:**
  - character evolution tree H_h with $V(H)$ being states of the character
  - a phylogeny tree G_g with leaves $L(G)$ being extant species
  - a *leaf labeling* $p: L(G) \to V(H)$ (a partial function)
- **task:**
  - find a *labeling* $l: V(G) \to V(H)$ which is:
    - p-constrained

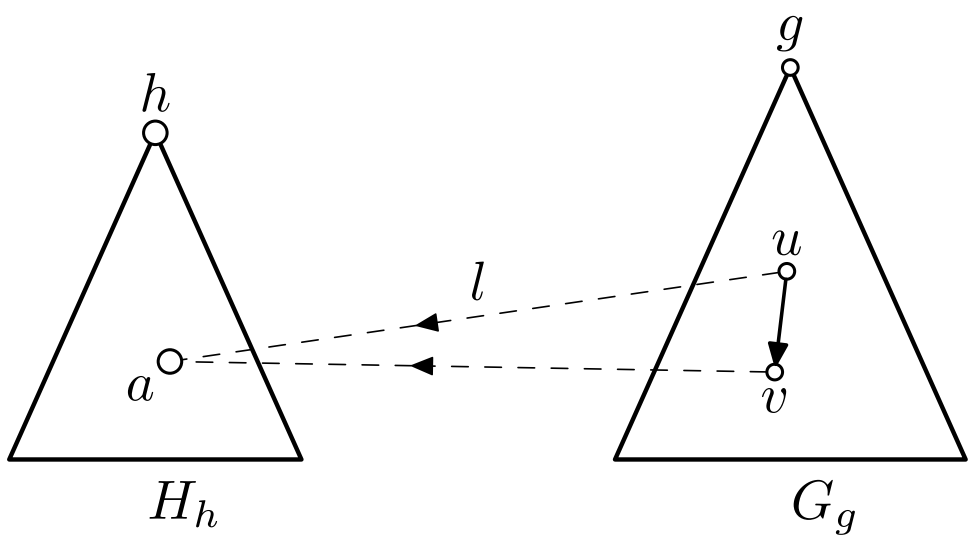


Small phylogeny with character evolution

- **given:**
  - character evolution tree H_h with $V(H)$ being states of the character
  - a phylogeny tree G_g with leaves $L(G)$ being extant species
  - a *leaf labeling* $p: L(G) \rightarrow V(H)$ (a partial function)
- **task:**
  - find a *labeling* $l: V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$

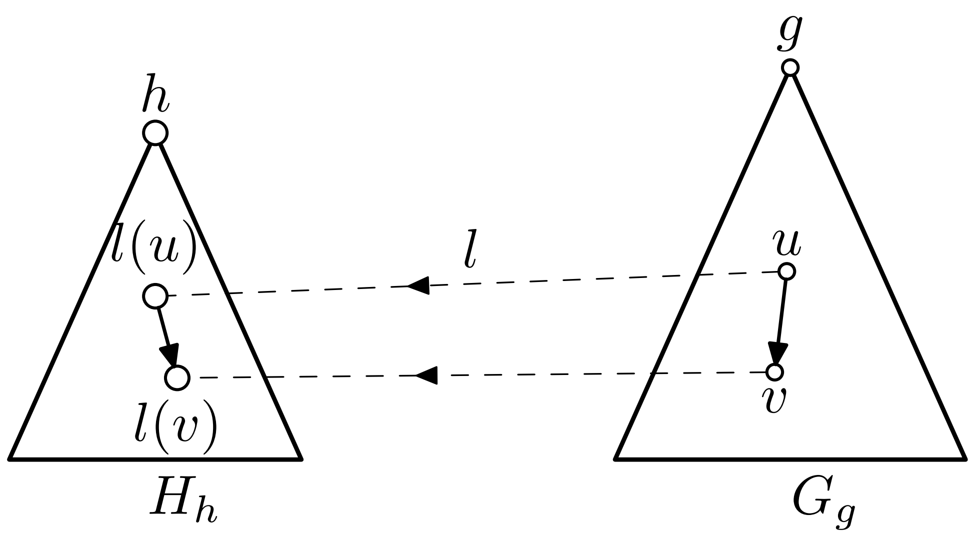
Small phylogeny with character evolution

- **task:**
  - find a *labeling* $l : V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$



Small phylogeny with character evolution

- **task:**
  - find a *labeling* $l : V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$

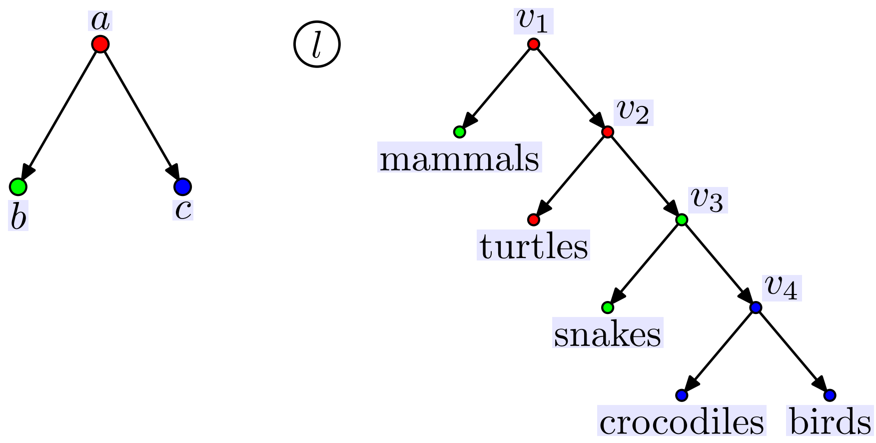


Small phylogeny with character evolution

- **task:**
  - find a *labeling* $l : V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$
    - does not allow "scattering" [Lipscomb (1992)]

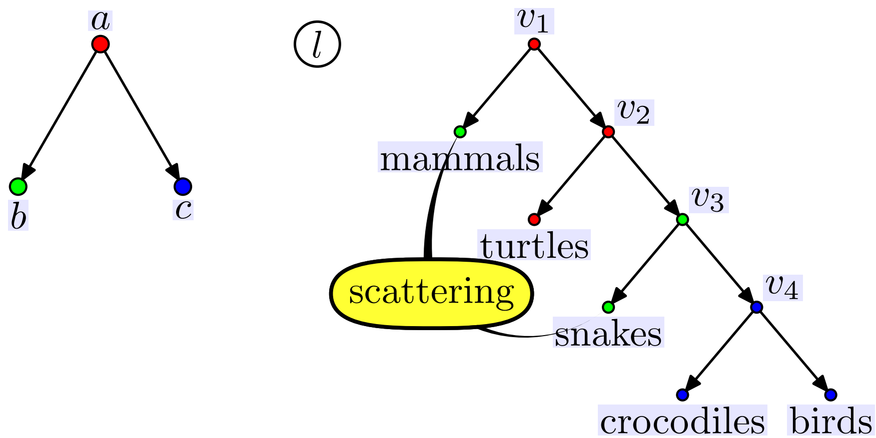
Small phylogeny with character evolution

- **task:**
  - find a *labeling* $l : V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$
    - does not allow "scattering" [Lipscomb (1992)]



Small phylogeny with character evolution

- **task:**
  - find a *labeling* $l : V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$
    - does not allow "scattering" [Lipscomb (1992)]



Small phylogeny with character evolution

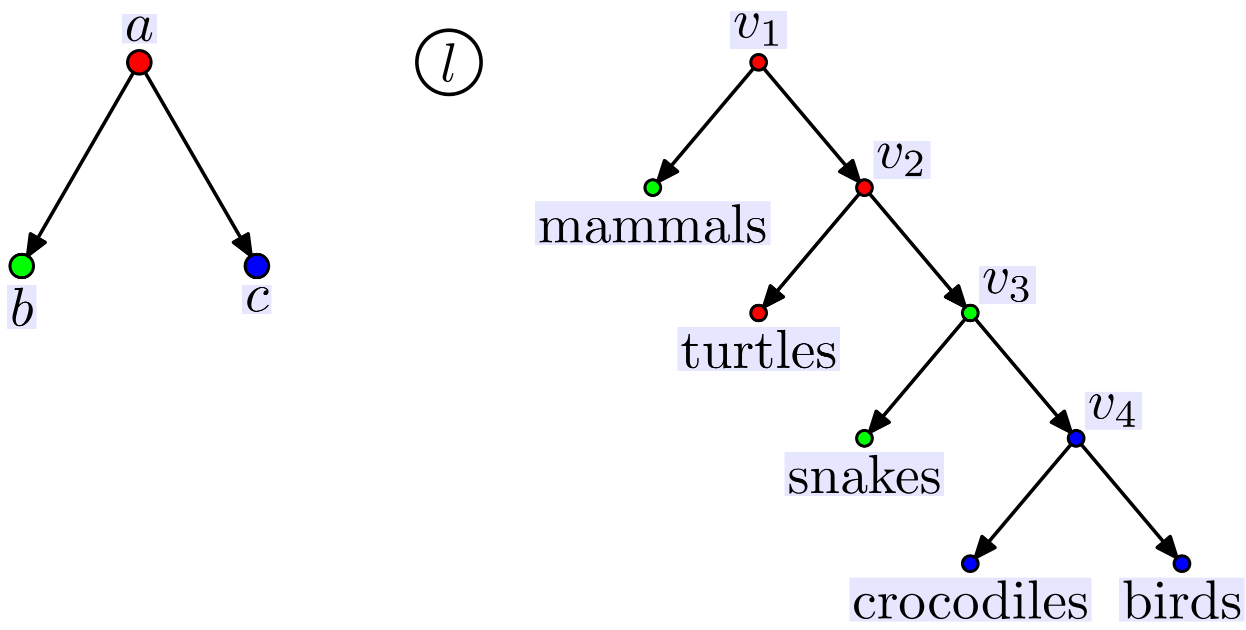
- **task:**
  - find a *labeling* $l : V(G) \rightarrow V(H)$ which is:
    - p-constrained
    - if a species v is a child of a species u then the character state $l(v)$ is either equivalent to, or a child of the character state $l(u)$
    - does not allow "scattering" [Lipscomb (1992)]
- existence of such labeling l is very close to graph-theoretical notion of graph minors

Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$

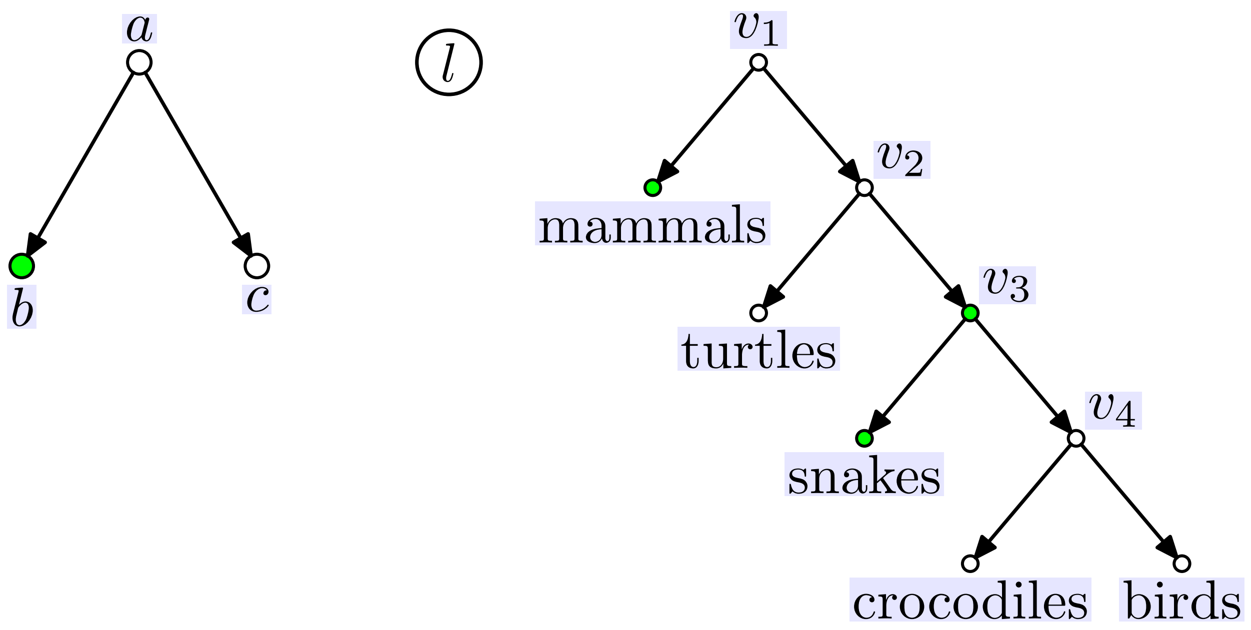
Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$



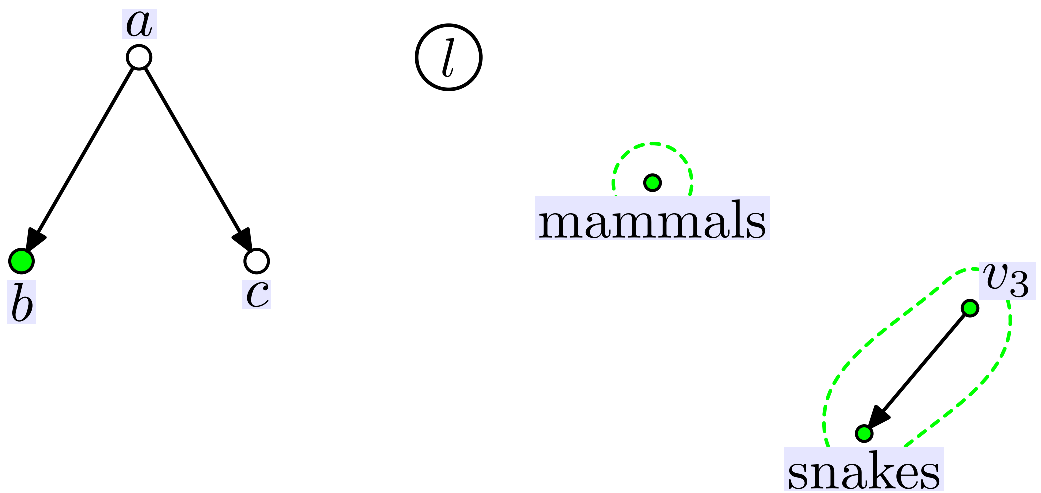
Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$



Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$

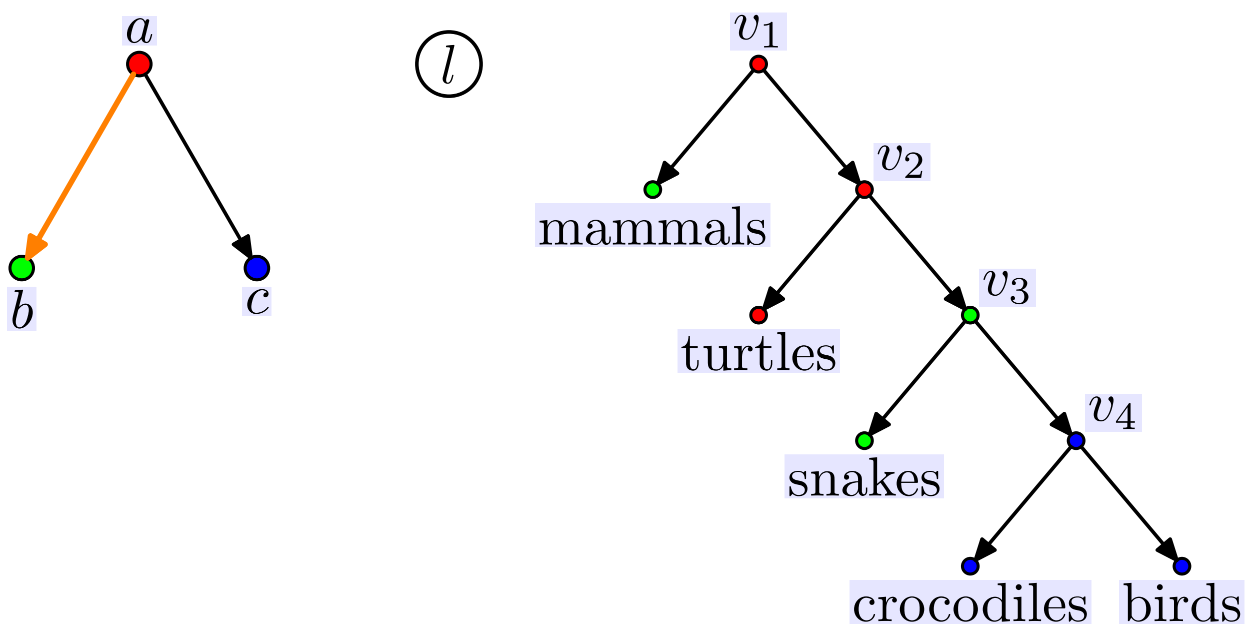


Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$
- evolutionary step $\langle u, v \rangle$ is a realization of an evolutionary transition $\langle a, b \rangle$ if $l(u) = a$ and $l(v) = b$

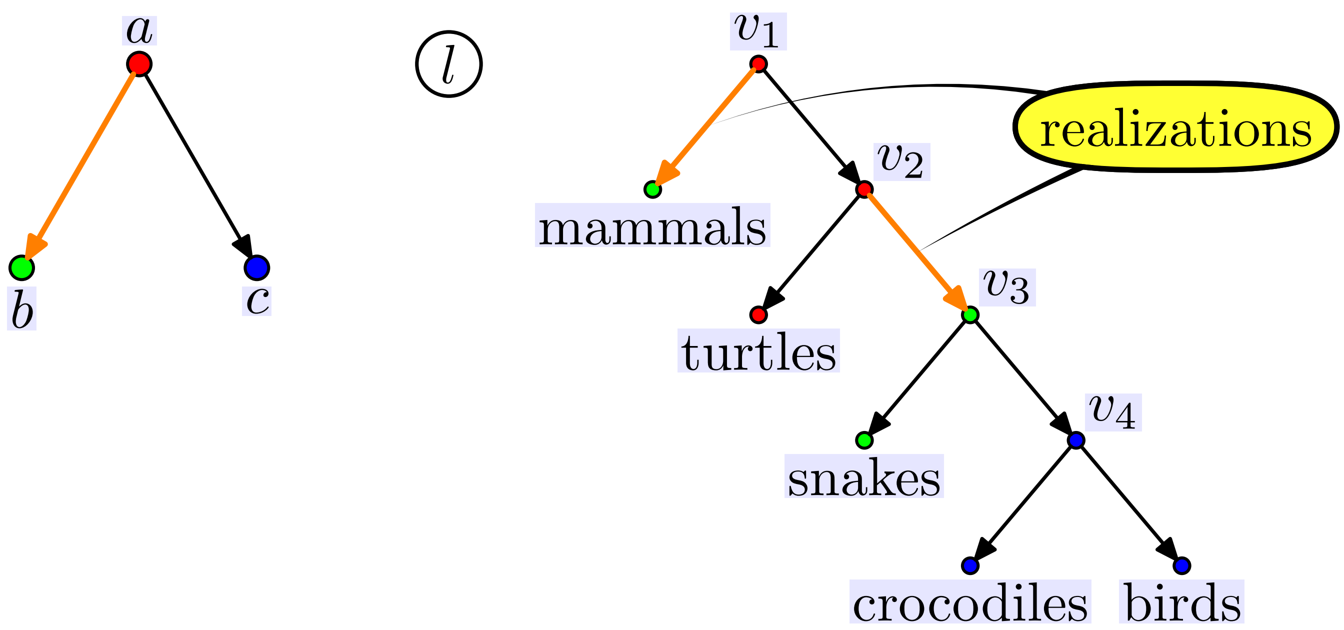
Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$
- evolutionary step $\langle u, v \rangle$ is a realization of an evolutionary transition $\langle a, b \rangle$ if $l(u) = a$ and $l(v) = b$



Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$
- evolutionary step $\langle u, v\rangle$ is a realization of an evolutionary transition $\langle a, b\rangle$ if $l(u) = a$ and $l(v) = b$



Rooted-tree minor

- bag-set of a state a: the set of connected components (called bags) of the subgraph of G_g induced by vertices in $l^{-1}(a)$
- evolutionary step $\langle u, v \rangle$ is a realization of an evolutionary transition $\langle a, b \rangle$ if $l(u) = a$ and $l(v) = b$
- We say that H_h is a *rooted-tree minor of* G_g, if there exists a labeling $l : V(G) \to V(H)$ satisfying:

(1) for each character state a, the bag-set of a contains **exactly one** component; and

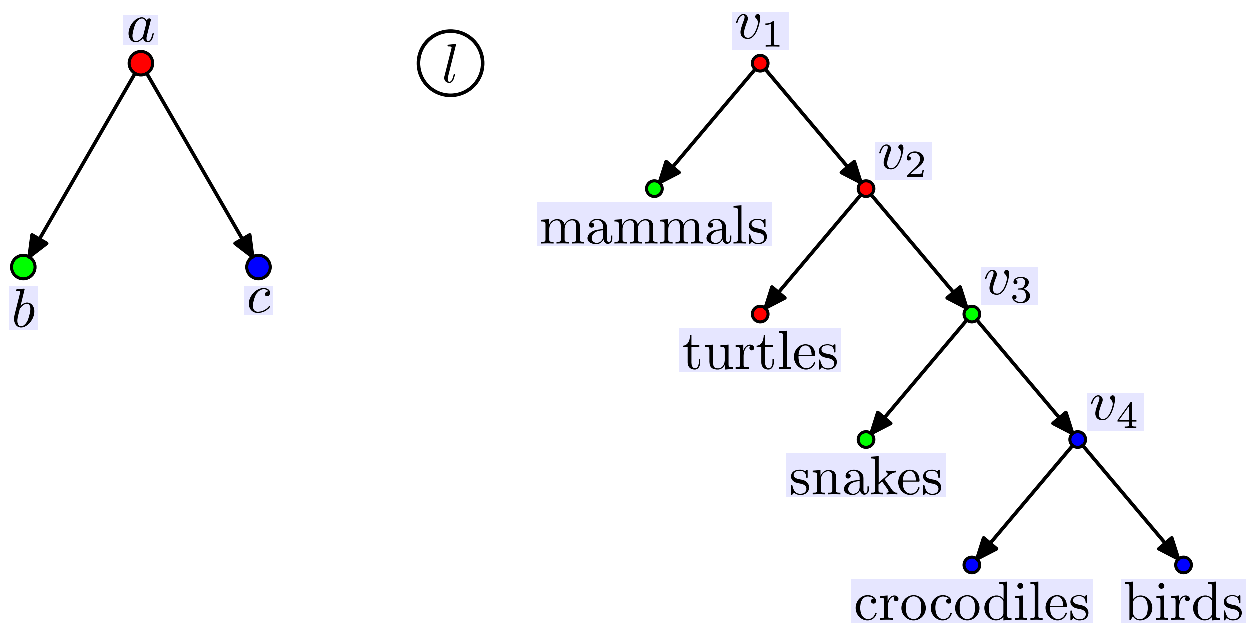
(2) each evolutionary transition has a realization.

Rooted-tree minor

- We say that H_h is a *rooted-tree minor of* G_g, if there exists a labeling $l : V(G) \to V(H)$ satisfying:

(1) for each character state a, the bag-set of a contains **exactly one** component; and

(2) each evolutionary transition has a realization.



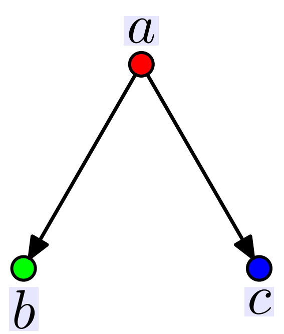
Rotated-tree minor

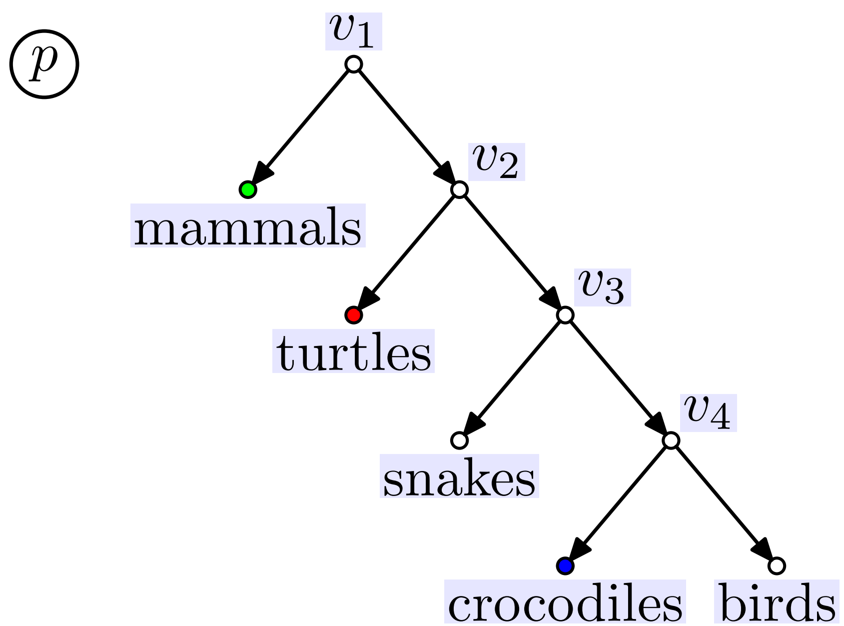
- We say that H_h is a *rooted-tree minor of* G_g, if there exists a labeling $l: V(G) \rightarrow V(H)$ satisfying:

(1) for each character state a, the bag-set of a contains **exactly one** component; and

(2) each evolutionary transition has a realization.

however, there exists such p-constrained labeling l

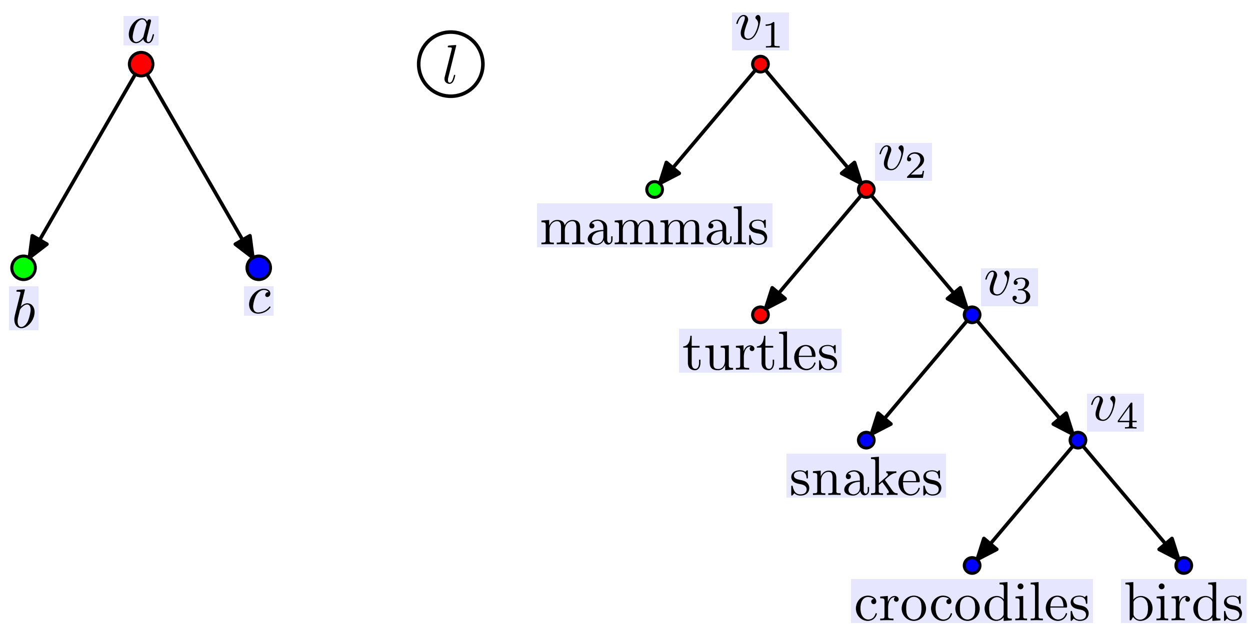




Rooted-tree minor

- We say that H_h is a *rooted-tree minor of* G_g, if there exists a labeling $l : V(G) \rightarrow V(H)$ satisfying:
  (1) for each character state a, the bag-set of a contains **exactly one** component; and
  (2) each evolutionary transition has a realization.

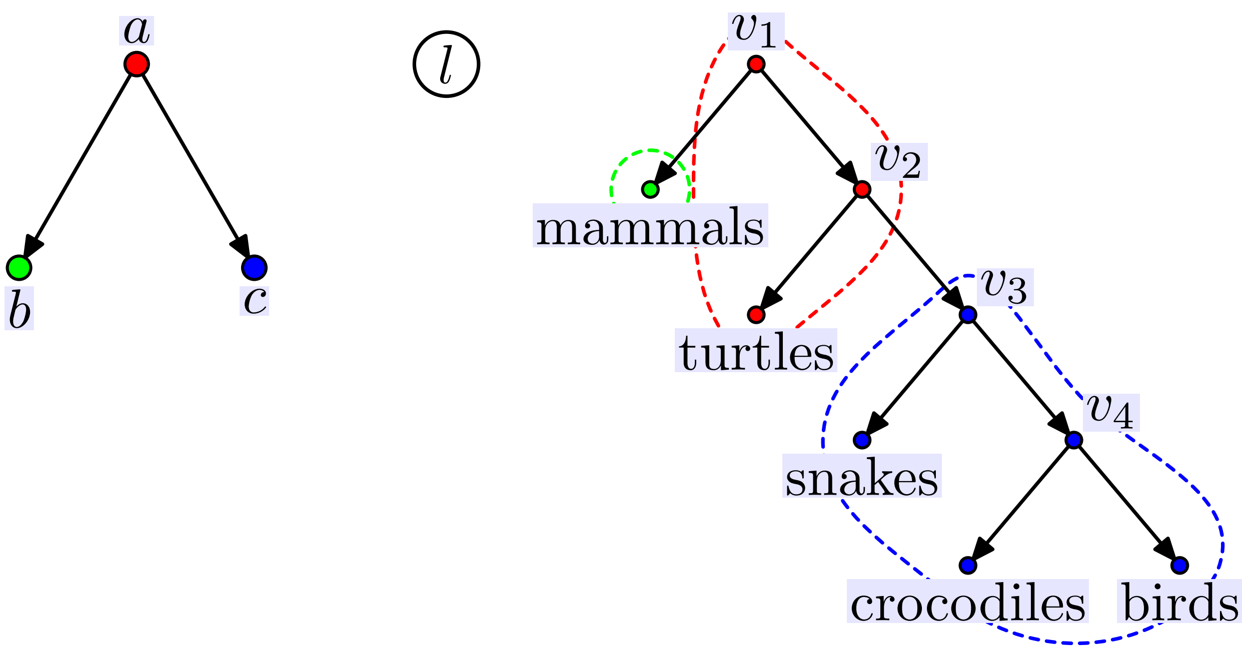
however, there exists such p-constrained labeling l



Rooted-tree minor

- We say that H_h is a *rooted-tree minor of* G_g, if there exists a labeling $l : V(G) \rightarrow V(H)$ satisfying:
  (1) for each character state a, the bag-set of a contains **exactly one** component; and
  (2) each evolutionary transition has a realization.

however, there exists such p-constrained labeling l



Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \to V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \rightarrow V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Tree minor problem** is NP-hard. [Matousek, Thomas (1992)]

Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \rightarrow V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

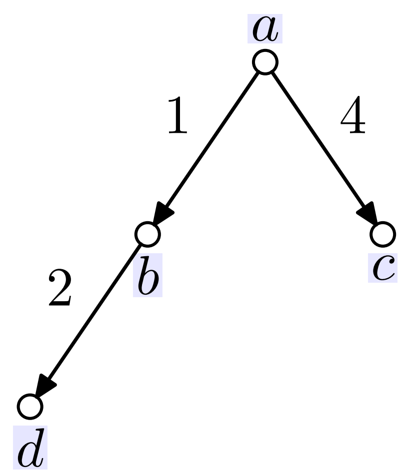
- **Tree minor problem** is NP-hard. [Matousek, Thomas (1992)]
- *Tree minor problem* can be converted to *Rooted-tree minor problem*, hence *Rooted-tree minor problem* is also NP-hard (when p is the empty leaf labeling).

Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \rightarrow V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Tree minor problem** is NP-hard. [Matousek, Thomas (1992)]
- *Tree minor problem* can be converted to *Rooted-tree minor problem*, hence *Rooted-tree minor problem* is also NP-hard (when p is the empty leaf labeling).

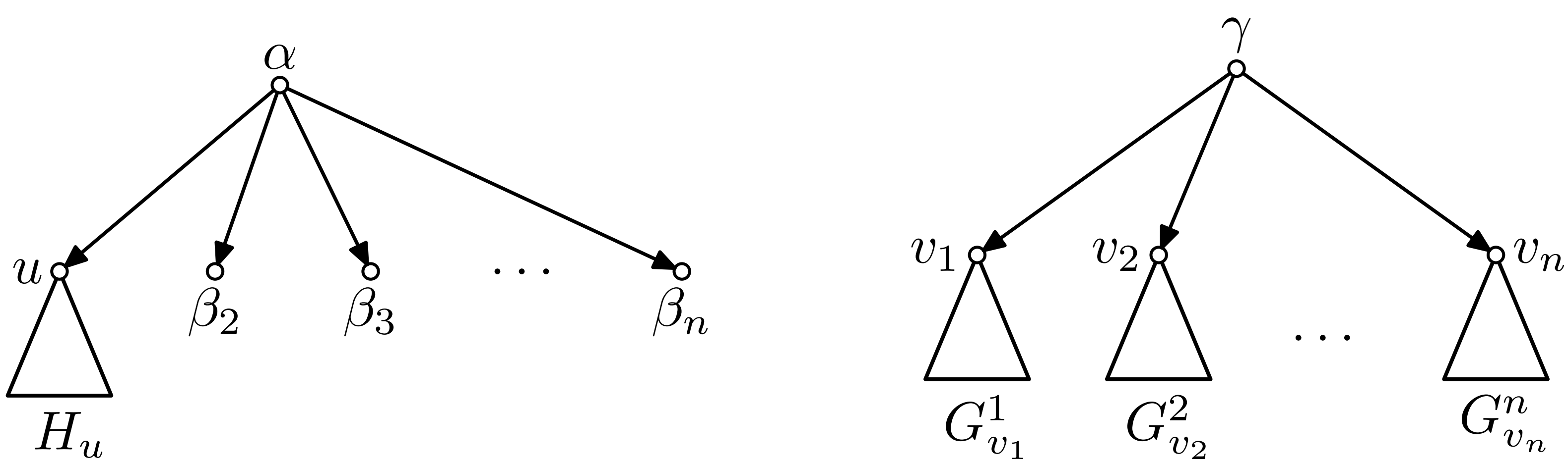
Consider an instance of Tree minor problem:



Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \to V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Tree minor problem** is NP-hard. [Matousek, Thomas (1992)]
- *Tree minor problem* can be converted to *Rooted-tree minor problem*, hence *Rooted-tree minor problem* is also NP-hard (when p is the empty leaf labeling).



Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \rightarrow V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Theorem 1.** *Rooted-tree minor problem* is NP-hard.

Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \to V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Theorem 1.** *Rooted-tree minor problem* is NP-hard.
- **Theorem 2.** If the leaf labeling p is complete, then *Rooted-tree minor problem* can be decided in **linear time**.

Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p: L(G_g) \rightarrow V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Theorem 1.** *Rooted-tree minor problem* is NP-hard.
- **Theorem 2.** If the leaf labeling p is complete, then *Rooted-tree minor problem* can be decided in **linear time**.
  - Compute the LCA-tree.
    (The label of each species is the least common ancestor of labels of its children.)
    This can be done in linear time (requires preprocessing on the character evolution tree [Harel, Tarjan (1984)]).

Rooted-tree minor problem

Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \to V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

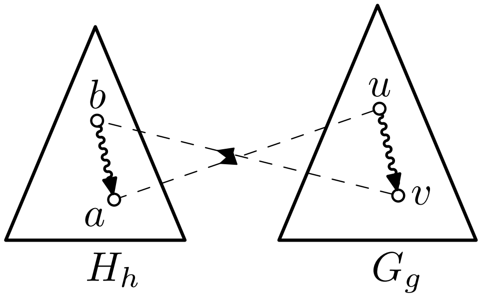
- **Theorem 1.** *Rooted-tree minor problem* is NP-hard.
- **Theorem 2.** If the leaf labeling p is complete, then *Rooted-tree minor problem* can be decided in **linear time**.
  - Compute the LCA-tree.
  - Fix labels of inner vertices of all single branch paths in G_g (if possible).
    Crucial lemma: the ends of single branch paths are already fixed correctly for any labeling l satisfying the definition of *Rooted-tree minor*.

Rooted-tree minor problem

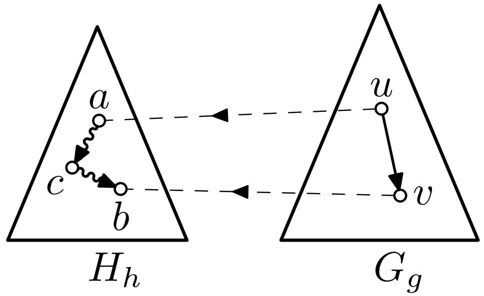
Rooted-tree minor problem. Given two rooted trees H_h and G_g, and a leaf labeling $p : L(G_g) \to V(H)$. Decide whether H_h is a rooted-tree minor of G_g with respect to p.

- **Theorem 1.** *Rooted-tree minor problem* is NP-hard.
- **Theorem 2.** If the leaf labeling p is complete, then *Rooted-tree minor problem* can be decided in **linear time**.
  - Compute the LCA-tree.
  - Fix labels of inner vertices of all single branch paths in G_g (if possible).
  - If each evolutionary transition has exactly one realization accept the input.

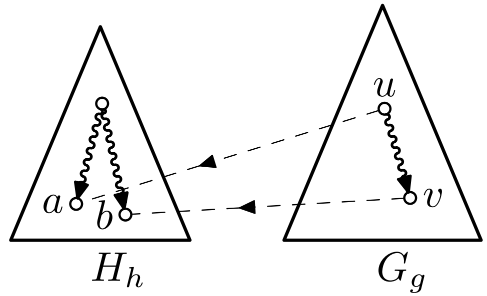
Incongruences



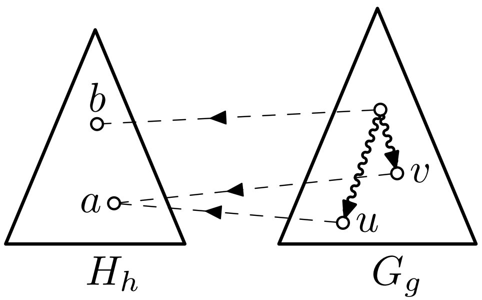
Inversion



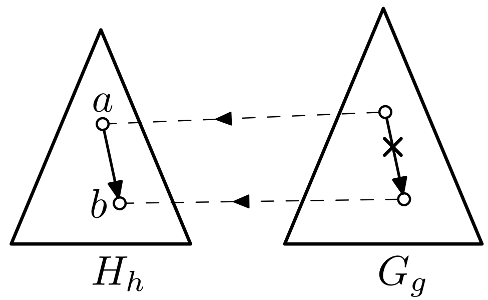
Transitivity



Addition



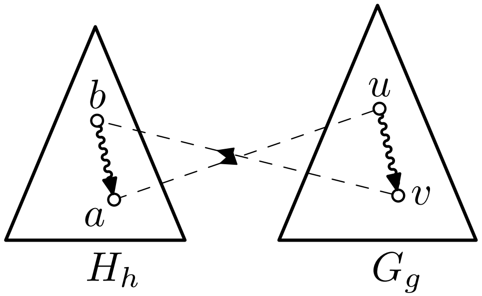
Separation



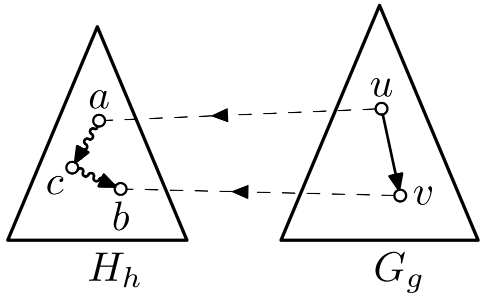
Negligence

Remark: solid lines represent arcs, wavy lines directed paths of length at least one.

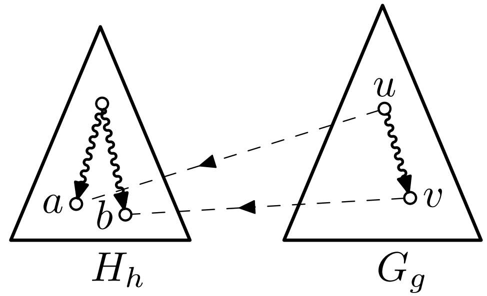
Incongruences



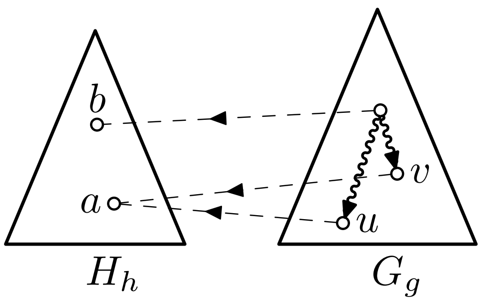
Inversion



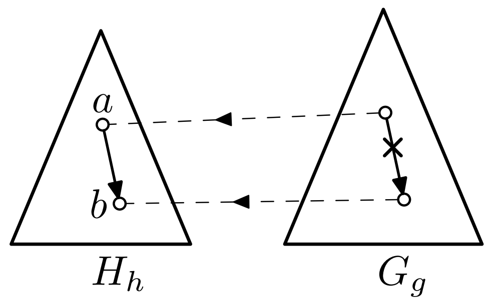
Transitivity



Addition



Separation



Negligence

- find labeling of nodes in phylogenetic tree minimizing the number of incongruences

Parsimony criteria

Given two rooted trees H_h and G_g with a labeling $l : G \rightarrow H$ and a weight function d on H_h, the *arc cost* and the *bag cost* of l are defined as follows:

$$\text{arccost}(H_h, G_g, l) := \sum_{\langle u,v \rangle \in A(G_g)} d(l(u), l(v)),$$

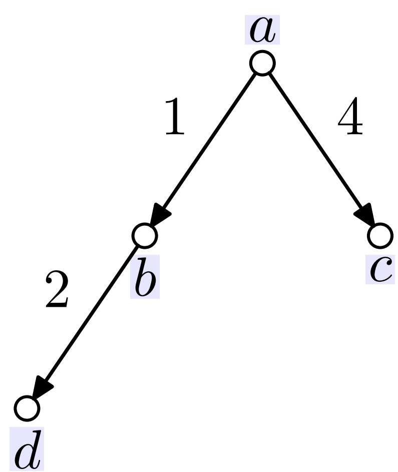
$$\text{bagcost}(H_h, G_g, l) := \sum_{v \in V(H)} \text{size of the bag-set of } v.$$

Parsimony criteria

Given two rooted trees H_h and G_g with a labeling $l : G \rightarrow H$ and a weight function d on H_h, the *arc cost* and the *bag cost* of l are defined as follows:

$$\mathrm{arccost}(H_h, G_g, l) := \sum_{\langle u,v\rangle \in A(G_g)} d(l(u), l(v)),$$

$$\mathrm{bagcost}(H_h, G_g, l) := \sum_{v \in V(H)} \text{size of the bag-set of } v.$$

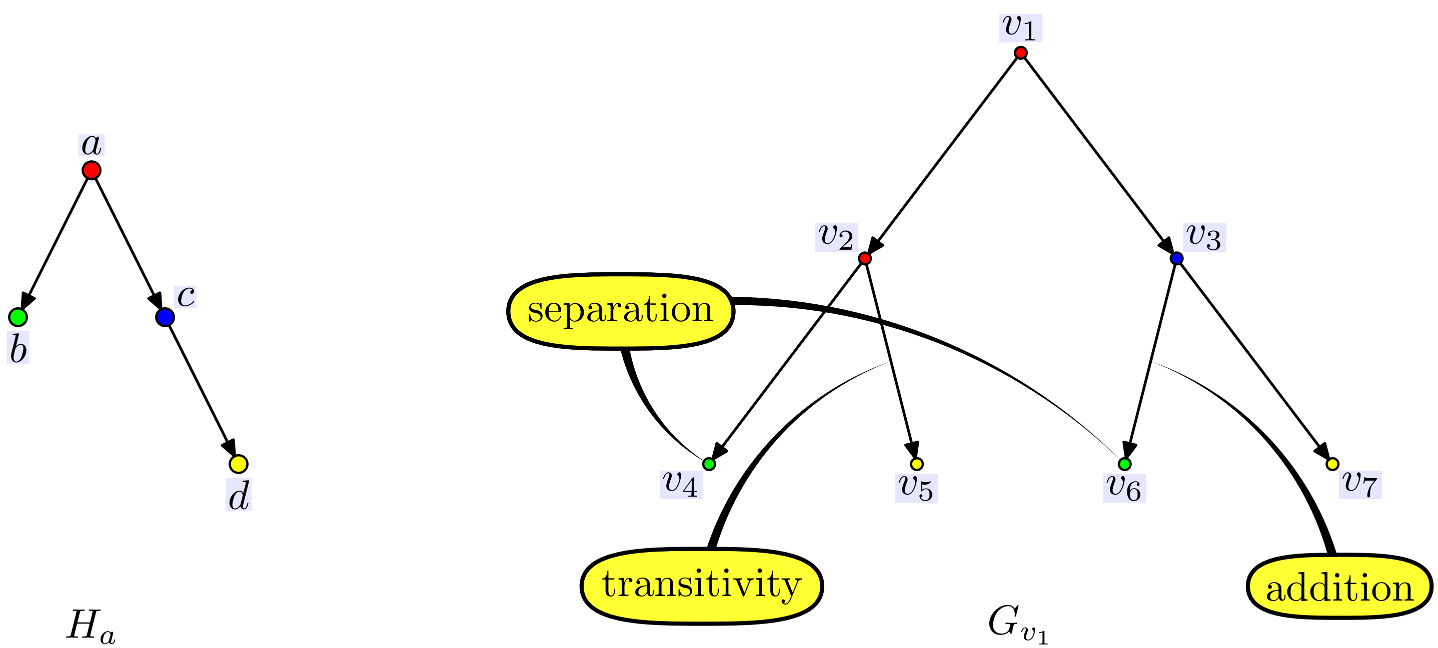


Relaxations of Rooted-tree minor

- **Relax-minor:** Given two rooted trees H_h and G_g with a leaf labeling p, we say that H_h is a *relax-minor* of G_g with respect to p if there exists a p-constrained labeling function $l : G \rightarrow H$ satisfying the following two conditions:
  - each evolutionary transition has a realization (no negligence); and
  - if u is an ancestor of v, then $l(v)$ cannot be a proper ancestor of $l(u)$ (no inversion).

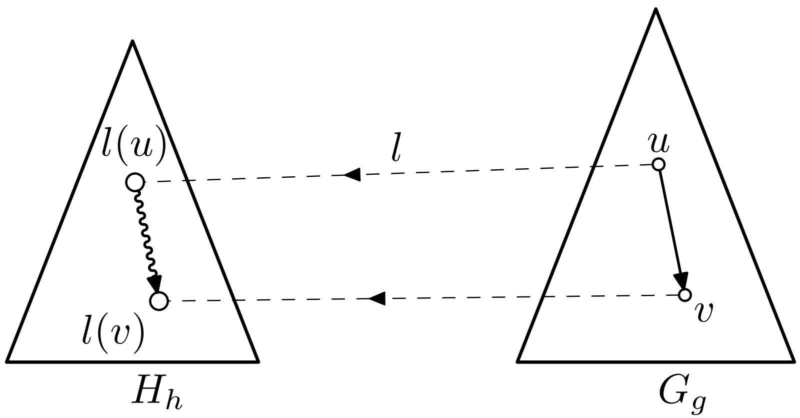
Relaxations of Rooted-tree minor

- **Relax-minor:**
  - each evolutionary transition has a realization (no negligence); and
  - if u is an ancestor of v, then $l(v)$ cannot be a proper ancestor of $l(u)$ (no inversion).
- **Example** of other incongruences:



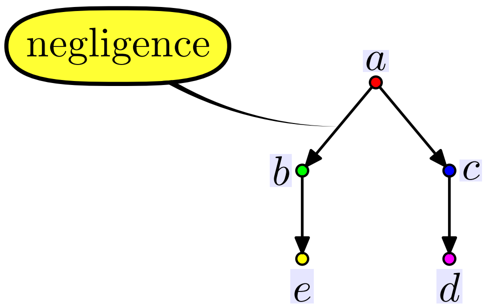
Relaxations of Rooted-tree minor

- Given two rooted trees H_h and G_g with a leaf labeling p, we say that H_h is a *pseudo-minor* of G_g with respect to p if there exists a p-constrained labeling function $l : G \rightarrow H$ such that
  - for every evolutionary step $\langle u, v \rangle$, $l(u)$ is an ancestor of $l(v)$ (no addition and inversion).

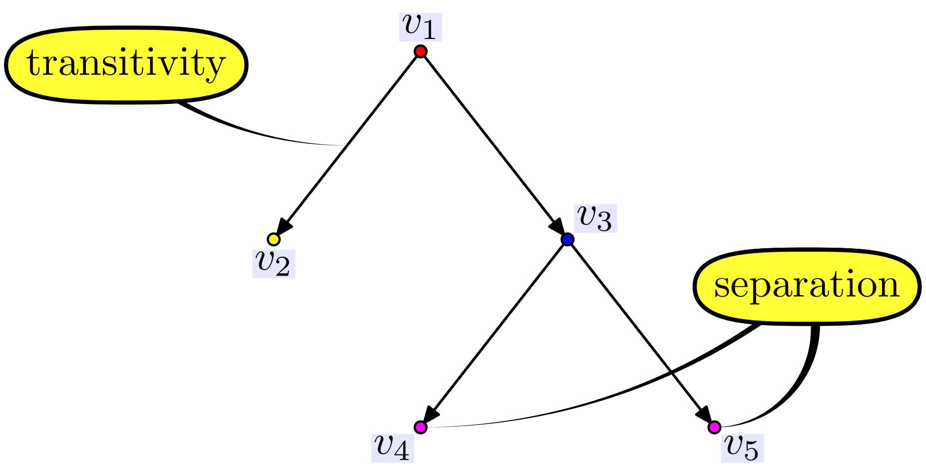


Relaxations of Rooted-tree minor

- Given two rooted trees H_h and G_g with a leaf labeling p, we say that H_h is a *pseudo-minor* of G_g with respect to p if there exists a p-constrained labeling function $l : G \rightarrow H$ such that
  - for every evolutionary step $\langle u, v \rangle$, $l(u)$ is an ancestor of $l(v)$ (no addition and inversion).
- **Example** of other incongruences:



H_a



G_{v_1}

Relaxations of Rooted-tree minor

- summary:

	rooted minor	*relax-minor*	*pseudo-minor*
inversion	N	N	N
transitivity	N	Y	Y
addition	N	Y	N
separation	N	Y	Y
negligence	N	N	Y

Complexities of Relax-minor problems

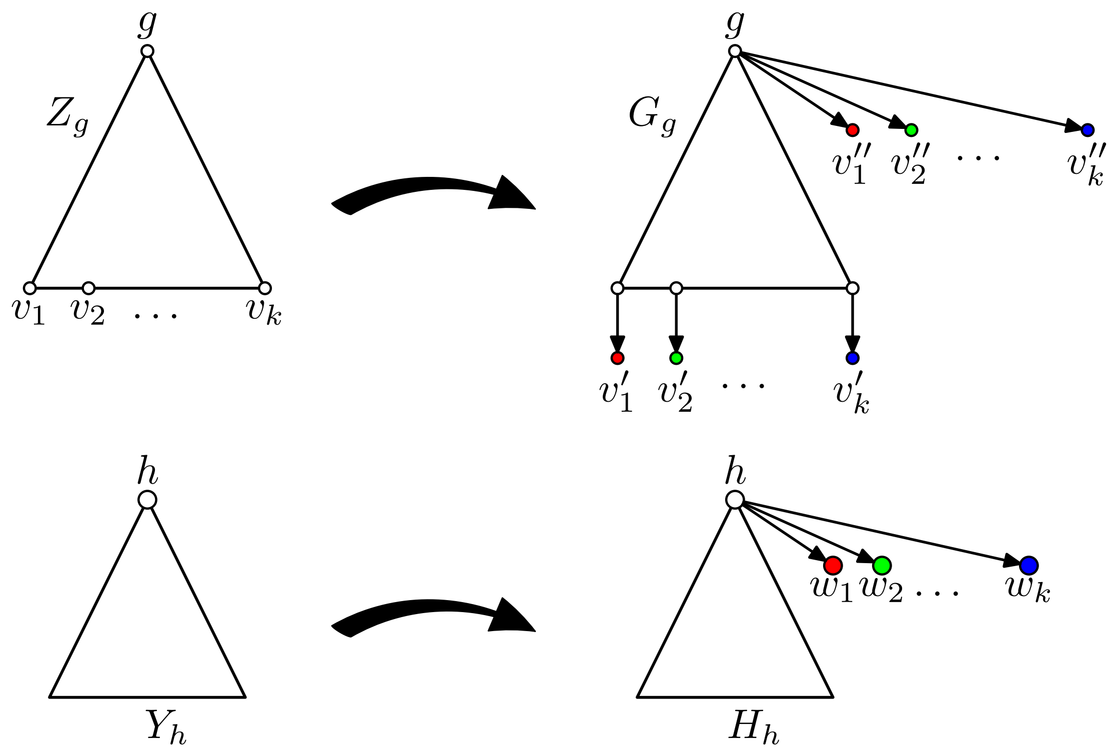
- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.

Complexities of Relax-minor problems

- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.
  - NP-hard if p is the empty leaf labeling.

Complexities of Relax-minor problems

- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.
  - NP-hard if p is the empty leaf labeling.
  - NP-hard even when p is a complete leaf labeling.



Complexities of Relax-minor problems

- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.
  - NP-hard if p is the empty leaf labeling.
  - NP-hard even when p is a complete leaf labeling.
- **Problem 2.** Find a relax-minor labeling with the minimal arc-cost.

Complexities of Relax-minor problems

- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.
  - NP-hard if p is the empty leaf labeling.
  - NP-hard even when p is a complete leaf labeling.
- **Problem 2.** Find a relax-minor labeling with the minimal arc-cost.
  - Since the relax-minor allows addition, the arc-cost is not always finite.

Complexities of Relax-minor problems

- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.
  - NP-hard if p is the empty leaf labeling.
  - NP-hard even when p is a complete leaf labeling.
- **Problem 2.** Find a relax-minor labeling with the minimal arc-cost.
  - Since the relax-minor allows addition, the arc-cost is not always finite.
  - NP-hard if p is the empty leaf labeling.

Complexities of Relax-minor problems

- **Problem 1.** Find a relax-minor labeling with the minimal bag-cost.
  - NP-hard if p is the empty leaf labeling.
  - NP-hard even when p is a complete leaf labeling.
- **Problem 2.** Find a relax-minor labeling with the minimal arc-cost.
  - Since the relax-minor allows addition, the arc-cost is not always finite.
  - NP-hard if p is the empty leaf labeling.
  - **Open problems.**
    - Is it possible to solve Problem 2 in polynomial time when p is a complete leaf labeling?
    - Is it possible to decide whether there is a relax-minor labeling with finite arc-cost in P time?

Complexities of Pseudo-minor problems

- **Problem 3.** Find a pseudo-minor labeling with the minimal bag-cost.

Complexities of Pseudo-minor problems

- **Problem 3.** Find a pseudo-minor labeling with the minimal bag-cost.
  - Can be done in linear time:
    - compute the LCA-tree;
    - if a species does not belong to a bag containing a leaf, change its label to the label of the root.

Complexities of Pseudo-minor problems

- **Problem 3.** Find a pseudo-minor labeling with the minimal bag-cost.
  - Can be done in linear time:
    - compute the LCA-tree;
    - if a species does not belong to a bag containing a leaf, change its label to the label of the root.
- **Problem 3.** Find a pseudo-minor labeling with the minimal arc-cost.

Complexities of Pseudo-minor problems

- **Problem 3.** Find a pseudo-minor labeling with the minimal bag-cost.
  - Can be done in linear time:
    - compute the LCA-tree;
    - if a species does not belong to a bag containing a leaf, change its label to the label of the root.
- **Problem 3.** Find a pseudo-minor labeling with the minimal arc-cost.
  - Can be done in linear time:
    - compute the LCA-tree.