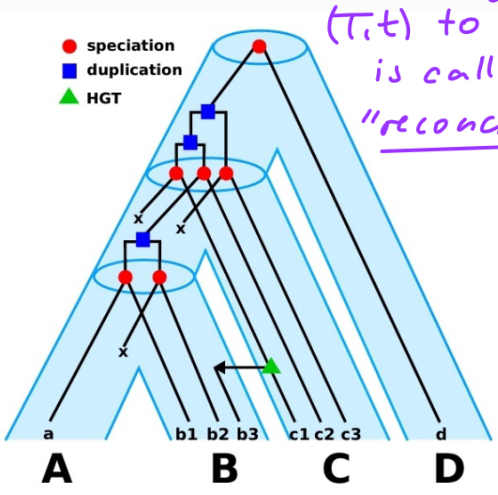
Homology - Relations

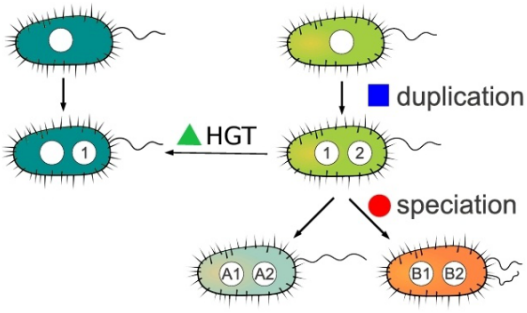
Detailed Evolutionary Scenarios:

- species are characterized by its genome:
  a "bag of genes"
- "Genes" evolve along a *rooted* tree with unique coloring
  $t : V^0 \to M = \{\bullet, \blacksquare, \blacktriangle\}$ $\hat{=}$ (T, t)
- "×" = gene loss

Embedding of (T,t) to S is called "reconciliation"



- Gene duplication : an offspring has two copies of a single gene of its ancestor
- Speciation : two offspring species inherit the entire genome of their common ancestor
- HGT : transfer of genes between organisms in a manner other than traditional reproduction and across different species



- Two genes are homologous if they share common ancestor.

- **Homology-Relations** are binary relations between genes.

[are important: gene function,
find new genes,
mechanism that act on genes]

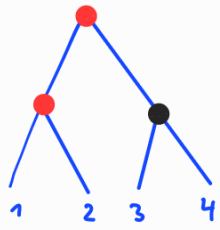
Here we investigate in more detail the structure of 2 Homology relations: ORTHOLOGY
XENOLOGY

ORTHOLOGY

- Two genes x, y are <u>orthologous</u> if they were separated by speciation event.

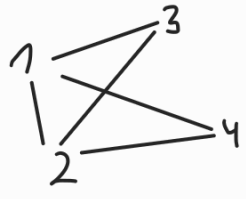
$\Longleftrightarrow$ in given gene tree (T, t) with $t : V \setminus L \to \{\bullet, \bullet\}$ we have $t(lca_T(x, y)) = \bullet$ = speciation.

$R_\bullet$ = binary relation that comprises all pairs of orthologous genes.



$R_\bullet = \{(1,2), (2,1), (1,3), (3,1), (1,4), (4,1), (2,3), (3,2), (2,4), (4,2)\}$.

<u>As graph:</u> (undirected)



2 classical ways to infer orthology:

Tree-based inference

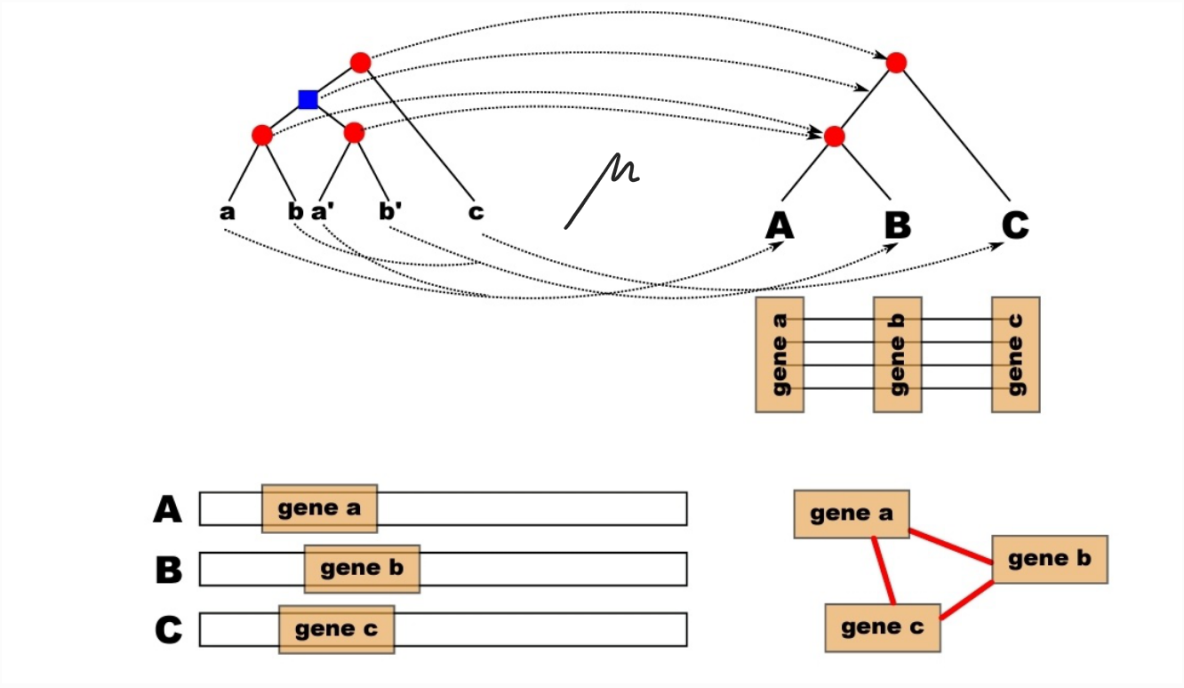
- construct gene and species trees and find reconciliation map μ between them
- based on the placing of vertices in gene tree to species tree on infers speciation events

Graph-based inference

Typically run in two phases:

- a graph construction phase, in which pairs of orthologous genes are inferred and connected by edges
- a clustering/clean-up phase, in which (groups of) orthologous genes are constructed/extracted based on the structure of the graph

TREE-BASED (IDEA)



1 **Compute Species Tree:**

- Find 1:1-orthologs
  = collection of genes such that from each species one gene and each gene is ortholog to all other genes in this collection
  - Select families of genes that rarely exhibit duplications (e.g. rRNAs, ribosomal proteins)
- Alignments of protein or DNA sequences and standart techniques yield gene tree with speciation-events only
  This history is believed to be congruent to that of the respective species.

Compute Gene Tree *without events*:

- Alignments of protein or DNA sequences and standart techniques

2 **Compute Gene Tree *without events*:**

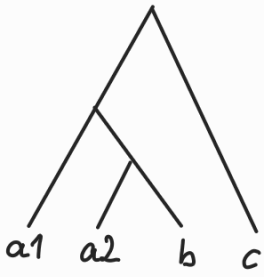
- Alignments of protein or DNA sequences and standart techniques

3 **Compute Events of Gene Tree:**

- Find reconciliation map μ w.r.t. certain optimization criteria (e.g. minimize number of losses and duplications)

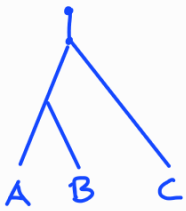
2

T (gene tree)



1

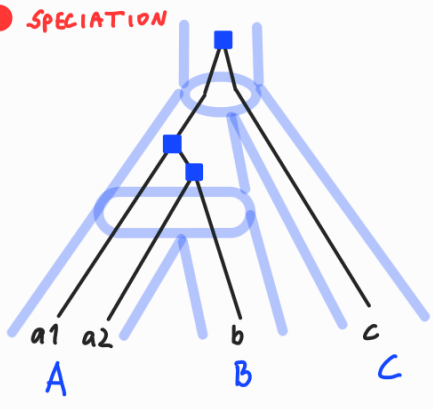
S (species tree)

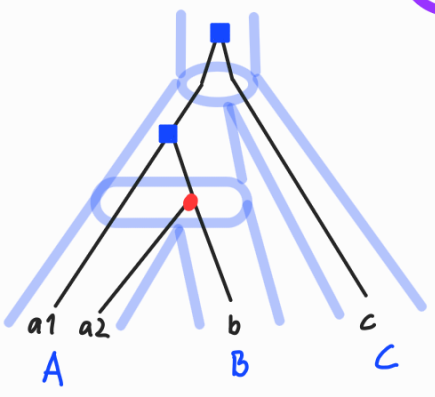


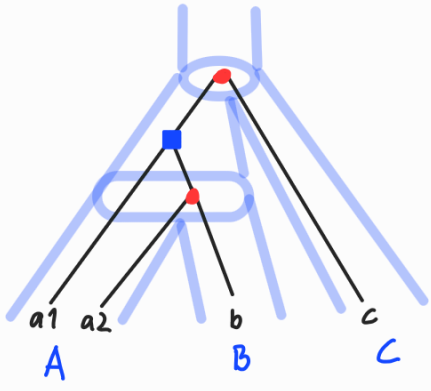
possible reconciliations

3

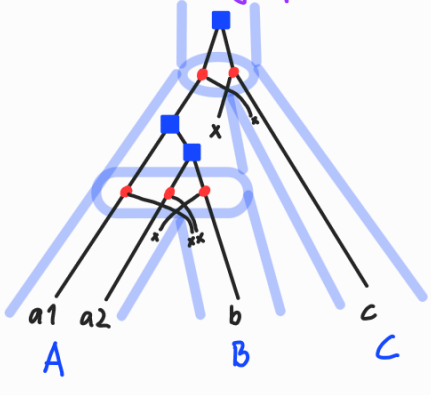
■ DUPLICATION
● SPECIATION



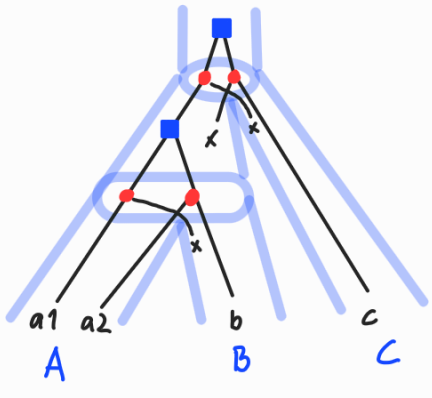




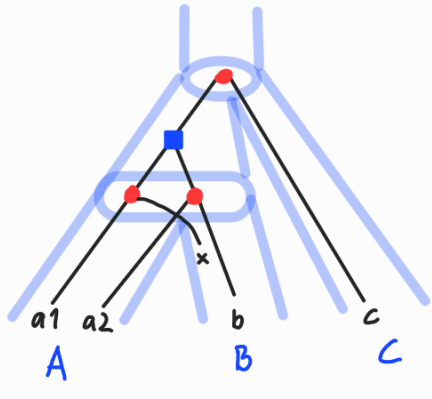
Resulting putative true evol. histories:



3 Duplic. / 5 losses



2 Dupl / 3 losses



1 Dupl / 1 loss

In practice: Find most parsimony reconciliation, i.e. one with a min. nr of losses & duplications

Observation (TREE-BASED)

- **Compute Species Tree**
  - some orthologs must already be known!
  - since only 1:1 orthologs are used, $\sim$ 90% of the genetic sequence material remains unused
- **Compute Gene Tree + Reconciliation**
  - Methods that allow to reconstruct the history of arbitrary genes rely on "restrictive" evolutionary models (e.g. event probabilities, maximum parsimony)

This reveals a circular problem:
Reconstruction of species trees requires identifying events of the family evolution
Reconstruction of event-labeled gene trees requires a known species trees

Accuracy strongly depends on the predicted gene tree and the used methods (together with underlying evolutionary model) to reconcile gene and species tree.

=> alternative!

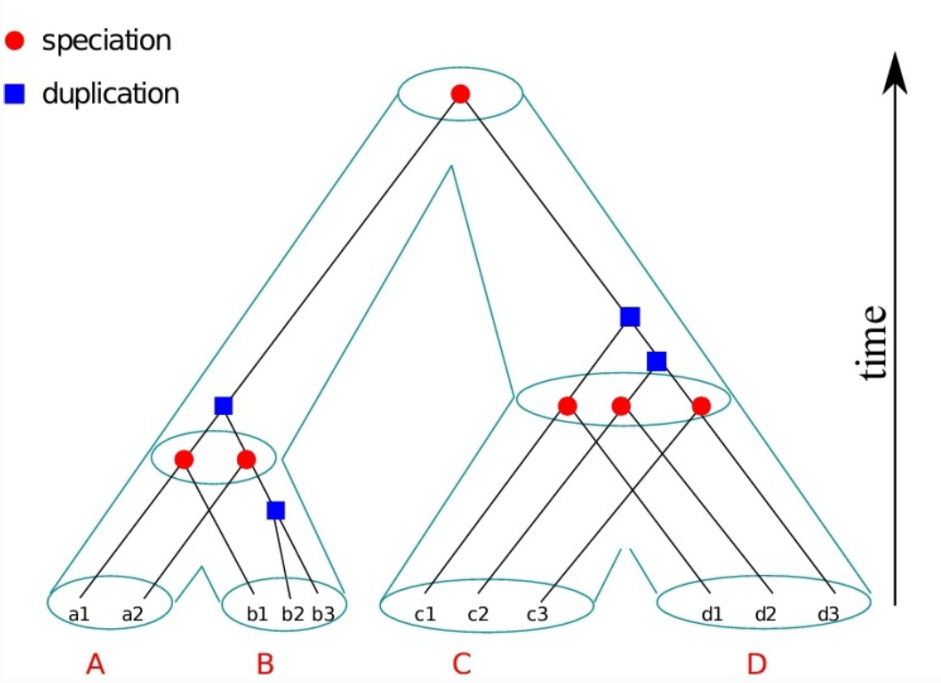
GRAPH-BASED (IDEA)

2 Phases:

1. a graph construction phase, in which pairs of orthologous genes are inferred and connected by edges

2. a clustering/clean-up phase, in which (groups of) orthologous genes are constructed/extracted based on the structure of the graph

① Assume we know true history [no HGT]



Observation:

Orthologs tend to be the homologs that diverged least. Why?

If no HGT, orthologs branched by definition at the latest possible time point—the speciation between the two genomes in question.

IDEA:

- T gene tree, S species tree
- $t_S(X, Y)$ = divergence time of species X, Y.
- $y \in Y$ is orthologous to $x \in X$, if
  1. $X \neq Y$, orthologs are never found in the same species
  2. $t_T(x, y) \simeq t_S(X, Y)$, divergence time of x and y in $T \simeq t_S(X, Y)$.

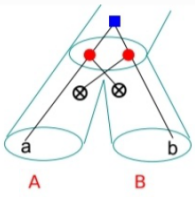
Don't know divergence time!
But can measure sequence similarities

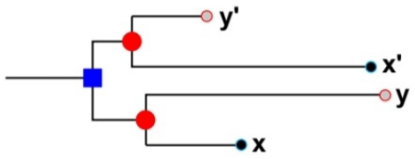
- T gene tree, S species tree
- ~~$t_S(X, Y)$ = divergence time of species X, Y.~~
- $y \in Y$ is a candidate orth. of $x \in X$, if
  1. $X \neq Y$, orthologs are never found in the same species
  2. $sim(x, y) \gtrsim sim(x, y') \; \forall \; y' \in Y$ and $sim(x, y) \gtrsim sim(x', y) \; \forall \; x' \in X$.

  if x and y are orthologs, then they do not have (much) closer relatives in the two species.

Not too weired mutation rates: $t_T(x,y) \leq t_T(x,y') \iff sim(x,y) \geq sim(x,y')$
"closer related, higher similarity"

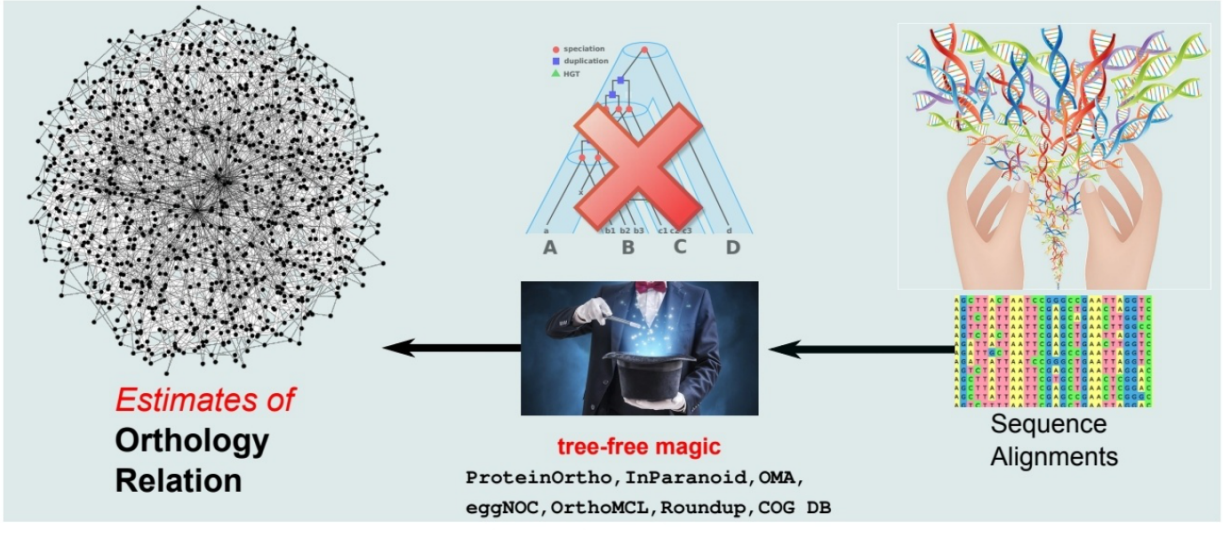
This cannot work perfectly:





But we can "approximate" Orthology from such sequence similarities

AKA: (Reciprocal) Best Hits/ Best matches.



How can we trust such estimates $\widehat{R}_\bullet$ of the *true* $R_\bullet$?

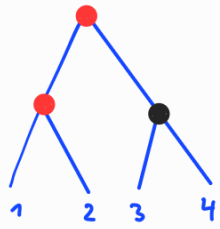
The least task we can do:
Ask for an event-labeled gene tree that supports our observation.

An estimated orthology relation $\widehat{R}_\bullet$ is **feasible** if
there is a tree $T = (V, E)$ with coloring $t\colon V^0 \to \{\bullet, \bullet\}$ such that

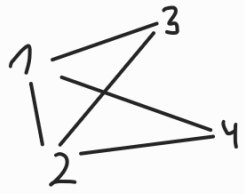
$$t(\mathrm{lca}_T(x,y)) = \bullet \iff (x,y) \in \widehat{R}_\bullet \text{ for all distinct } x, y \in X.$$

Can we mathematically characterize **feasible** estimates $\widehat{R}_\bullet$?

The structure of Orthology



As graph: (undirected)



$R_\bullet = \{ (1,2), (2,1), (1,3), (3,1), (1,4), (4,1), (2,3), (3,2), (2,4), (4,2) \}$.

$R_\bullet$ is symmetric ($lca(xy) = lca(yx)$)
but not transitive (eg. 3 & 1 are orthol.
1 & 4 are orthol.
But 3 & 4 are not orthol.)

we can represent $R_\bullet$ as an undirected graph $G = (V, E)$

$\{x, y\} \in E \Leftrightarrow (xy), (yx) \in R_\bullet$

To understand $R_\bullet$ we can investigate its graph structure.

Thus we consider the equivalent problem:

Given a graph G.

Is there a tree (T, t) with leaf set $L(T) = V(G)$ & labeling $t: V^0 \to \{0,1\}$ with $V^0 := V(T) \setminus L(T)$

s.t.

$$\{x,y\} \in E(G) \iff t(lca(x,y)) = 1$$

In the following we characterize feasible graphs

To this end, we need some extra notation:

DEF [complement]: $\overline{G}$ of G: $V(\overline{G}) = V(G)$

$$E(\overline{G}) = \{\{x,y\} \mid \{x,y\} \notin E(G), x \neq y\}$$

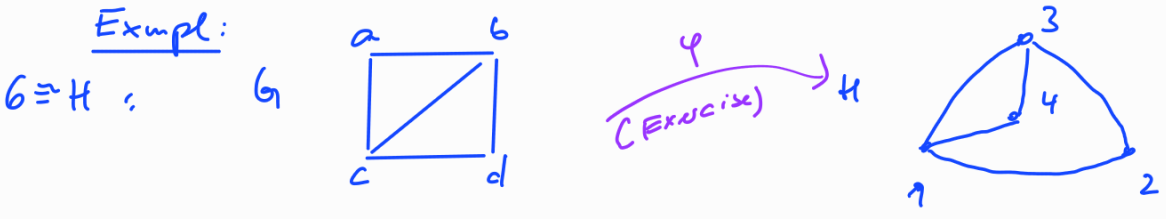
DEF (isomorphy)

2 graphs G, H are isomorphic (in symbols $G \cong H$) if $\exists$ bijective map $\varphi: V(G) \to V(H)$ s.t.

$$\{x,y\} \in E(G) \iff \{\varphi(x), \varphi(y)\} \in E(H)$$

Exmpl:

$G \cong H$:



DEF [induced subgraph]

Given $G = (V, E)$ & $W \subseteq V$. The subgraph $G[W]$ induced by W has vertex set W & edges $\{x,y\}$ for all $x, y \in W$ & $\{x,y\} \in E$

DEF [path]

A path $P \subseteq G$ is subgraph of the form:

[formal: $V(P) = \{v_1 \dots v_n\}$, $n \geq 1$ [i.e. $v_i \neq v_j$ $\forall i \neq j$]
$E(P) = \{\{v_i, v_{i+1}\} \mid 1 \leq i < n\}$]

length of path $P = |V(P)|$

Path of length n are denoted by P_n.

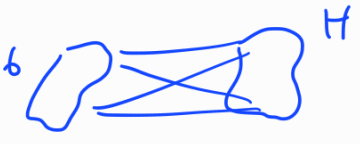
DEF [disjoint union / join]

Given graphs G, H with $V(G) \cap V(H) = \emptyset$.

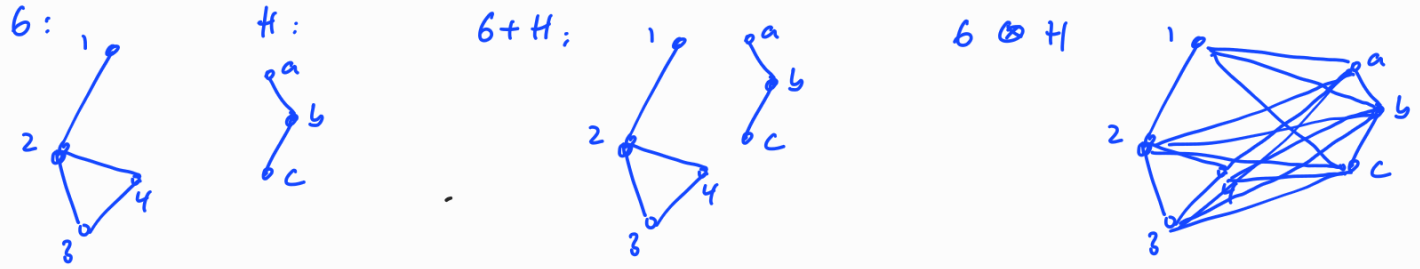
Their disjoint union $G + H$:

$$V(G+H) := V(G) \dot\cup V(H)$$
$$E(G+H) := E(G) \dot\cup E(H).$$

Their join $G \otimes H$:



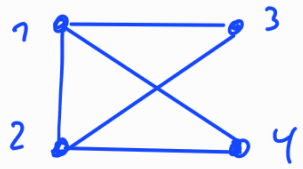
$$V(G \otimes H) := V(G) \dot\cup V(H)$$
$$E(G \otimes H) := E(G+H) \cup \{\{x,y\} : x \in V(G), y \in V(H)\}$$



DEF: Cograph recursively defined:

1) K_1 is cograph ($K_1 \cong$ • single vertex graph)
2) if G_1 & G_2 cograph
   $\Rightarrow$ $G_1 + G_2$ & $G_1 \otimes G_2$ cograph.

Is this graph a cograph?

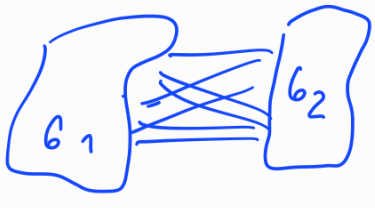


[Exercise start will if G cograph
$\Rightarrow G = G_1 \otimes G_2$ / $G_1 \cup G_2$
what is then G_1 & G_2?
recurse on G_1 & G_2]

Obs. if G cograph $\Rightarrow$ $G \cong K_1$ or either G or $\overline{G}$ disconnected.)
$\Rightarrow$ G cograph implies $\overline{G}$ cograph.

Why? $G = G_1 + G_2$ disconn. ✓

$G = G_1 \otimes G_2$



$\Rightarrow$ G connected
& $\overline{G}$:

$\overline{G} = \overline{G_1} + \overline{G_2}$

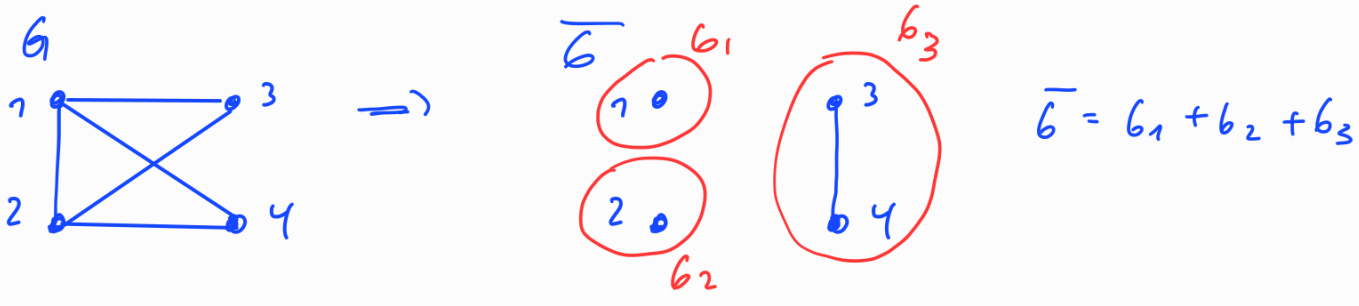
$\Rightarrow$ IF G cograph : $G = G_1 + G_2$

or $G = G_1 \otimes G_2 \Leftrightarrow \overline{G} = \overline{G_1 \otimes G_2} = \overline{G}_1 + \overline{G}_2$

with G_1 / G_2 cographs.

& either G or $\overline{G}$ disjonnected.

$\Rightarrow$ check components of disconnected graph in $\{G, \overline{G}\}$ whether they are cographs!



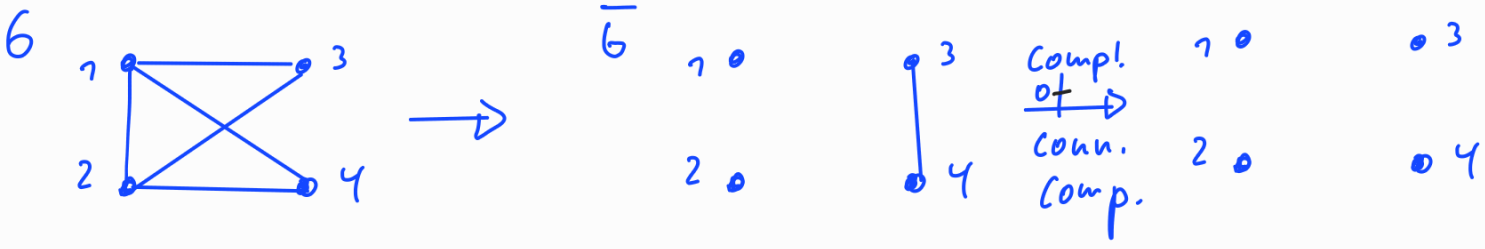
$\overline{G} = G_1 + G_2 + G_3$

since $G_1 \cong G_2 \cong K_1$ they are cographs

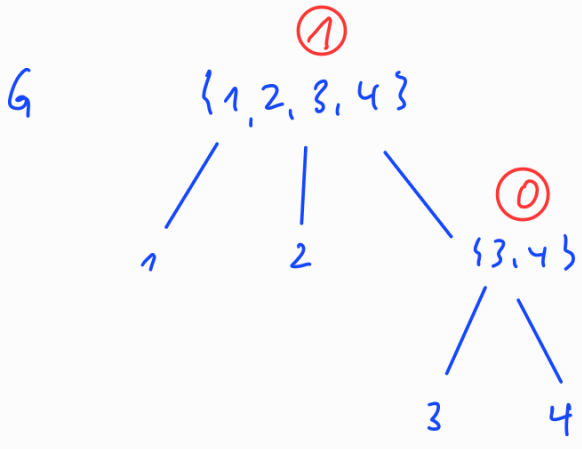
For $G_3 =$ 3 $\otimes$ 4 cograph by def.

$\Rightarrow G = (1 \otimes 2) \otimes (3 + 4)$

cograph = complement reducible graph,
i.e. stepwise complementation
of conn. components yield
complete edge-less graph.



In this way one can compute tree for cographs:



$t(lca(xy)) = 1 \iff \{x,y\} \in E(G)$.

ALGO (Input cograph G)

init: Add root s to empty tree T

IF (G connected)

$t(s) = 1$, $B = 0$, call COTREE($\overline{G}, s, B$)

ELSE $t(s) = 0$, $B = 1$, call COTREE(G, s, B)

COTREE(G, v, B)

IF ($|V(G)| = 1$)

return rooted tree "$\bullet x$" // $V(G) = \{x\}$

ELSE // G disconnected since cograph

Let $C_1 \dots C_k$ conn. of G

add vertices $v_1 \dots v_k$ to T

add edges $\{v, v_i\}$ to to T, $1 \le i \le k$.

Label $t(v_i) = B$

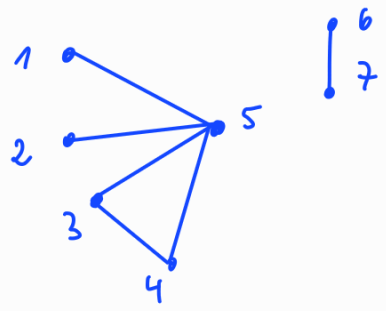
FOR ($i = 1 \dots k$)

call COTREE($\overline{G[C_i]}, v_i, \overline{B}$) // $\overline{B} = \begin{cases} 1, B=0 \\ 0, B=1 \end{cases}$

Exampl:

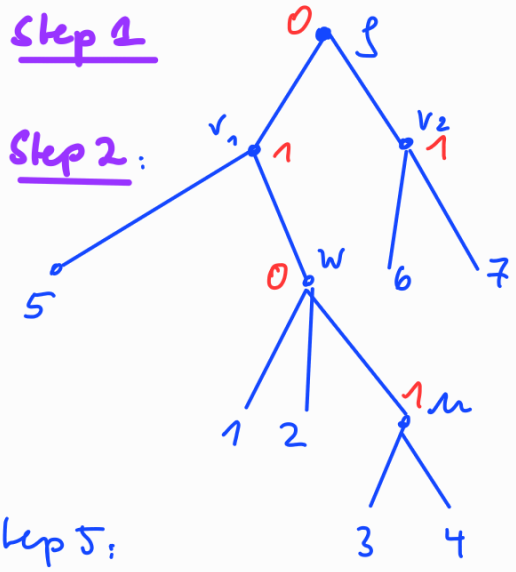
label

G:



Step 1

Step 2:



Step 3

Step 4:

Step 5:

Step 1 init

call $(G, s, B=1)$

Step 2 G disconnected

with 2 conn. comp.

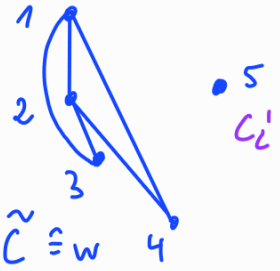
$C_1 = \{1 \dots 5\}$

$C_2 = \{6, 7\}$

→ call $(\overline{G[C_1]}, v_1, 0)$ // $\overline{G[C_i]}$ must be disconnected!

call $(\overline{G[C_2]}, v_2, 0)$

Step 3 $\overline{G[C_1]}$



$\overline{G[C_2]} =$ 

call $(\overline{G[\tilde{C}]}, w, 1)$, call $(\overline{G[\{5\}]}, 5, 1)$

$(\overline{G[\{6\}]}, 6, 1)$

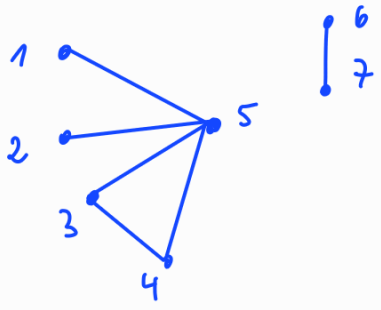
$(\overline{G[\{7\}]}, 7, 1)$

} Stop in next iteration

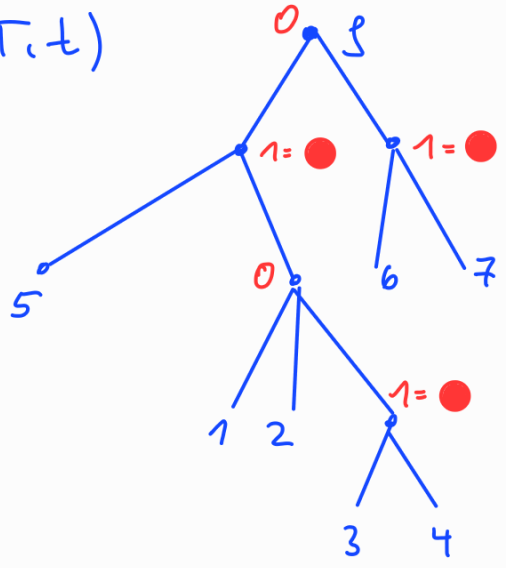
Step 4

→ step 5 ... and so on.

G



(T, t)



Leaves of T are vertices of G.

By construction:

$$\{x,y\} \in E(G) \iff t(lca_T(xy)) = 1$$

"G is explained by (T,t)"

Thm: [without proof but idea clear]

G cograph $\iff$ $\exists\ (T,t)$ that explains G

Recap: $\{x,y\} \in G_\Theta \iff t(lca(xy)) = \bullet$

Thm [2013] G_Θ valid $\iff$ G_Θ is cograph.

⊛ If G cograph $\Rightarrow$ every induced subgraph $H \subseteq G$ is cograph

[exercise, hint look at underlying tree structure]

Thm: G cograph $\Leftrightarrow$ $\nexists$ induced P_4.

proof: P_4 

$\overline{P_4} \cong P_4$ 

both conn. $\Rightarrow$ no cograph.

$\circledast$ $\Rightarrow$ IF G cograph then it cannot contain induced P_4.

Now assume G does not contain induced P_4.

By induction on $|V(G)|$ with $|V(G)| \leq 3 \Rightarrow G$ cograph.
Assume statement holds for all G with $|V(G)| < n$, ie: every graph with less than n vertices & without induced P_4 is cograph.

Consider now G with $|V(G)| = n \geq 4$ & no induced P_4.

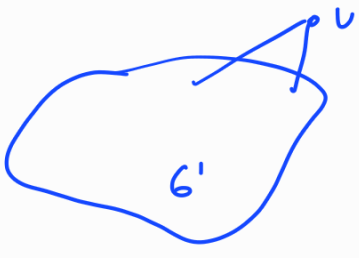
IF G disconnected $\Rightarrow$ G can be written as $G = G_1 \dot\cup G_2$

$\Rightarrow |V(G_1)|, |V(G_2)| < n$

$\overset{\text{Ind hyp}}{\Rightarrow}$ G_1, G_2 cogr. $\overset{\text{Def.}}{\Rightarrow}$ $G_1 \dot\cup G_2 = G$ cogr.

IF G connected, then consider $G' = G - v$

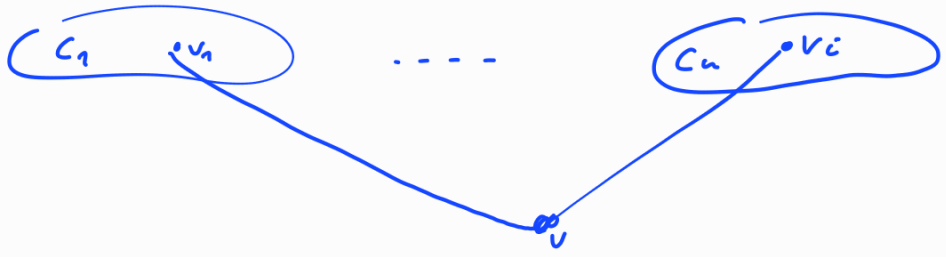
(remove v & all incid. edgs from G)



G' is cograph (by Ind hyp).

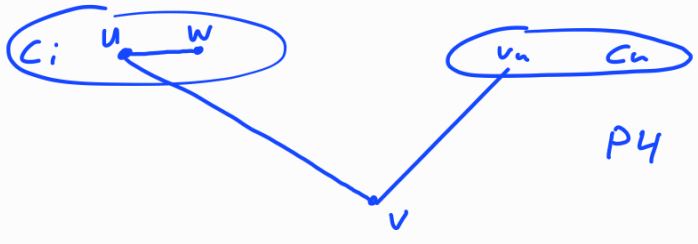
a) if G' disconnected it has conn. comp. $C_1 .. C_n$

Since G connected $\Rightarrow$ v must be incident to some $v_i \in C_i$ $\forall$ $i \in \{1..n\}$.



Assume, for contradiction, v not adj to <u>all</u> $w \in V(G')$.

$\Rightarrow$ $\exists$ conn. comp C_i of G' & vertices w, u s.t.



P_4 $\not\subseteq$ since G contains no P_4

$\Rightarrow$ $G = G' \otimes K_1$ cograph per Def.

b) IF G' connected.

since G' cograph $\Rightarrow$ $\overline{G'}$ disconn. & cograph.

$\Rightarrow$ $\overline{G'}$ P_4-free

now apply analogous arguments as in case a) for $\overline{G}$

/13

Summary

Thm: R • feasible $\Leftrightarrow$ its graph representation is a cograph $\Leftrightarrow \nexists$ induced P_4 

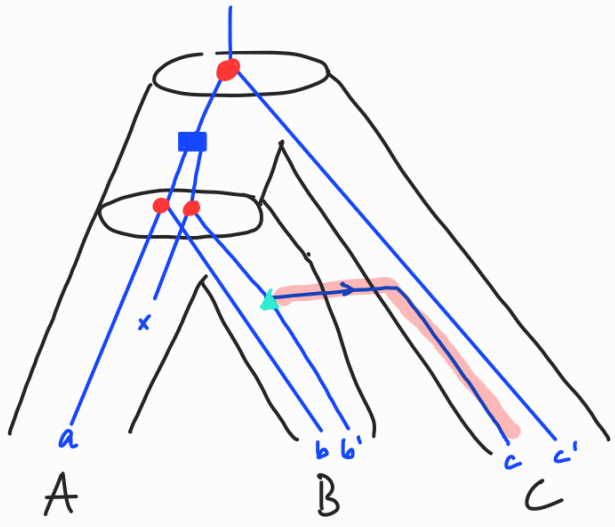
$\Rightarrow$ estimated ontology that violates the "cograph property" must be corrected

[cograph-editing is NP hard].

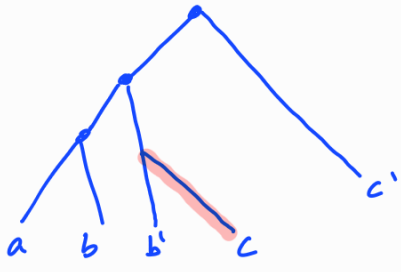
Xenology

Examples for HGT
& IDEA of inference methods
↗ slides.

HGT



- losses cannot be observed!
- Given tree with specified HGT-edges

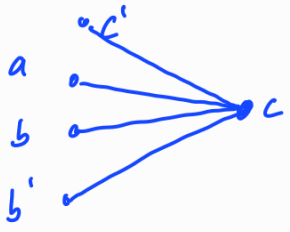


(T, λ)

$\lambda : E \to \{0,1\}$, $\lambda(e) = 1 \Leftrightarrow e$ is HGT edge.

$\mathcal{F}(T, \lambda) = (V, E)$ with

$V := L(T)$, $E := \{ |xy| \mid \exists$ edge e with $\lambda(e) = 1$ along unique xy-path in $T \}$

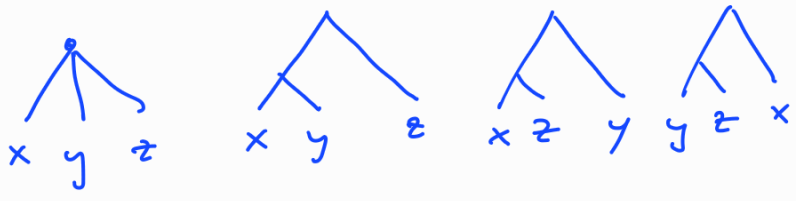


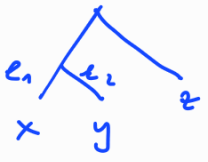
Given G, when exist (T, λ) s.t. $G \cong \mathcal{F}(T, \lambda)$?
In this case, (T, λ) explains G.

Lemma: $F(T,\lambda)$ does not contain $K_1 + K_2 \hat{=}$ "$\cdot\ \ \cdot$" as an induced subgraph.

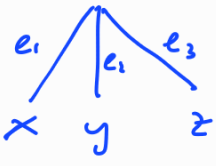
Proof: Assume, for contradiction, it contains $K_1 + K_2$ as induced subgraph.

possible trees:



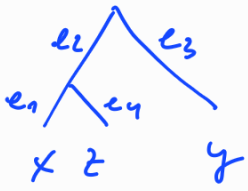


or



$\Rightarrow$ at least one of e_1 or e_2 has label 1

$\Rightarrow \exists$ edge $\{xz\}$ or $\{yz\}$ $\lightning$



$\Rightarrow \lambda(e_4) = 0$ [since z not adj to y & x]

$\lambda(e_1) = 0$ [since z not adj to x]

$\lambda(e_2) = \lambda(e_3) = 0$ [since z not adj to y]

$\Rightarrow$ no HbT edges $\Rightarrow$ $\lightning$ to $\{xy\} \in E$

[analog case 

]

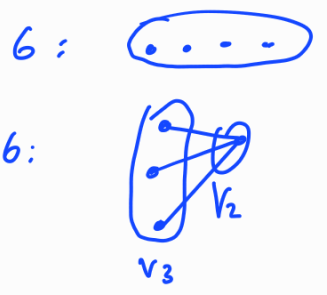
$\Rightarrow$ $F(T,\lambda)$ cannot contain $K_1 + K_2$ $\square$

DEF: $G = (V, E)$ is complete multipartite, if exist partition $V_1 \dots V_k$, $k \geq 1$ of V s.t.

$G[V_i]$ contains no edges, $1 \leq i \leq k$

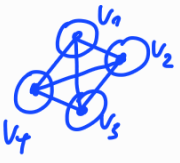
$\forall x \in V_i, y \in V_j, i \neq j : \{x y\} \in E$.

Exmpl:

G: 

G: 

G:

G: 

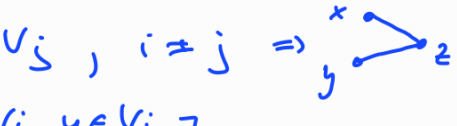
Lemma G complete multipartite $\Leftrightarrow$ does not contain $K_1 + K_2$ as an induced subgraph

proof: „$\Rightarrow$" Let $V_1 \dots V_k$, $k \geq 1$ be partition of V

Take 3 vertices $x, y, z \in V$.

cases: $x, y, z \in V_i \Rightarrow G[V_i]$ edgeless

$\Rightarrow G[\{x, y, z\}] \cong \cdot\ \cdot$

$x, y \in V_i$, $z \in V_j$, $i \neq j \Rightarrow$ 

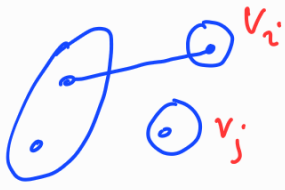
[analog if $xz \in V_i$, $y \in V_j$
$yz \in V_i$, $z \in V_j$]

$x \in V_i$, $y \in V_j$, $z \in V_k$, i, j, k pairw. distinct

$\Rightarrow$ 

$\Rightarrow$ no $K_2 + K_1$ V

"$\Leftarrow$" Let $V_1 \dots V_k$ be partition of V s.t. $G[V_i]$ is inclusion-maximal wrt to "edge"-less, $1 \le i \le k$

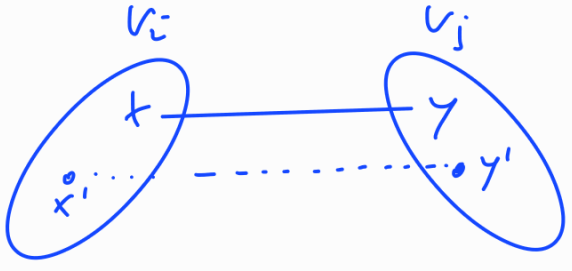


not inclusion-max
[can combine V_i & V_j]

inclusion-max:  or

$\Rightarrow$ between distinct V_i & V_j there must be at least one edge $\{xy\}$, $x \in V_i$, $y \in V_j$

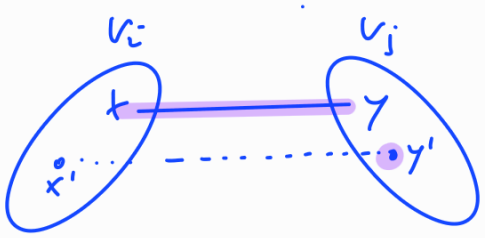
Assume $\exists\, x' \in V_i$ s.t. x' not adj. to all $y' \in V_j$

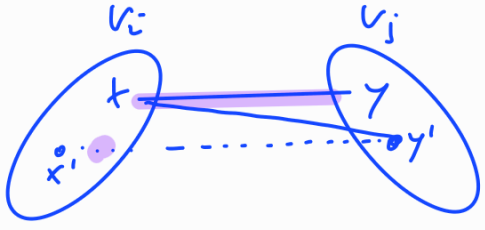


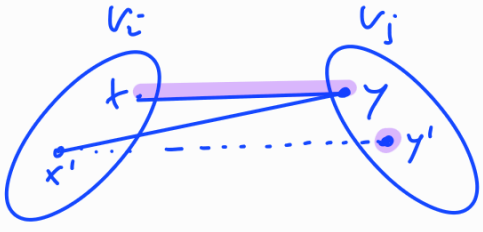
"…." no edge

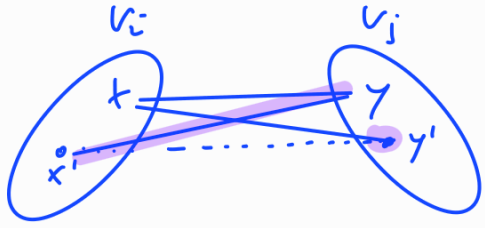
Subcases:

$K_2 + K_1$









$\Rightarrow$ Contains $K_1 + K_2$ ↯

$\Rightarrow \forall x \in V_i, y \in V_j : \{xy\} \in E \Rightarrow$ G complete multipartite graph

COR : $F(T,d)$ is complete multipartite graph.

Theorem [2018]

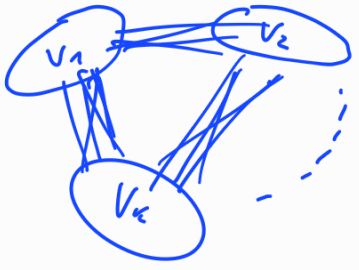
G can be explained by (T,d), i.e. $G = F(T,d)$ $\iff$ G is complete multipartite graph.

proof:

"$\Rightarrow$" $G = F(T,d) \overset{COR}{\Rightarrow}$ G complete multip.

"$\Leftarrow$" G complete multipartite.

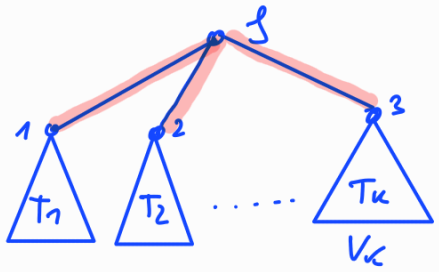
$\Rightarrow$



if $k = 1$ $\Rightarrow$ G edge-less

with no HGT edges

$k > 1$ $\Rightarrow$



HGT edge

$T_i \cong$

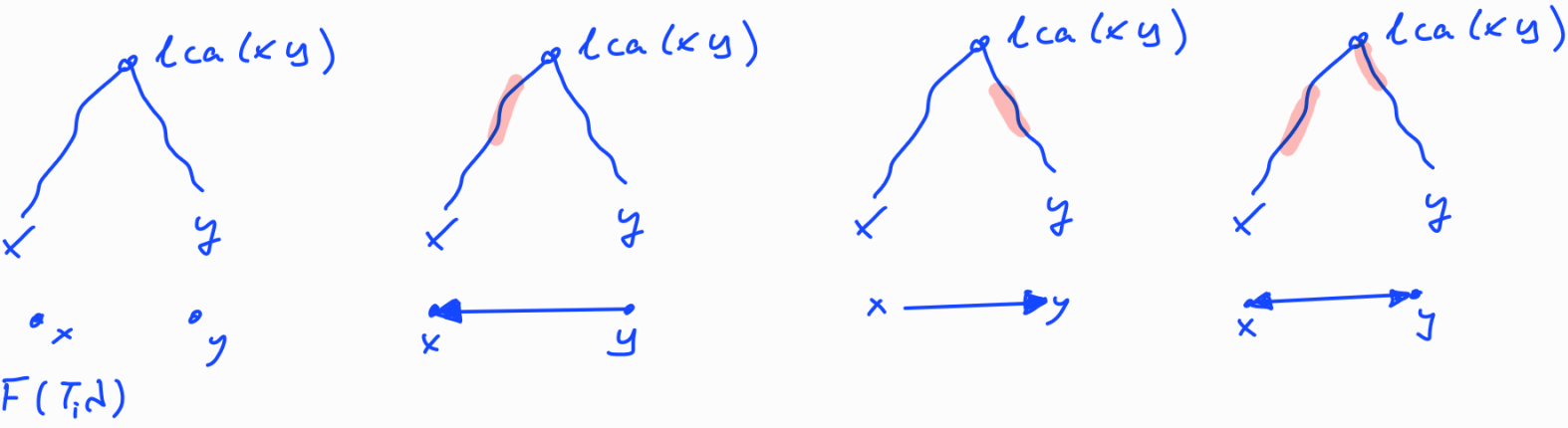
no HGT edges

□

Outlook:

- The latter characterization helps us to find trees & if there is no tree ($\exists$ forbidden subgraphs) we must edit graphs st no forb. subgraphs exist (NP hard) heuristics.

▶ what if direction known?



F(T,d)

"directed version"

→ we end in further "forbidden subgraph"

▶ what if orthologs & paralogs are known? when do they fit in common tree & do they constrain each other?

▶ what if evolution is "network" like?