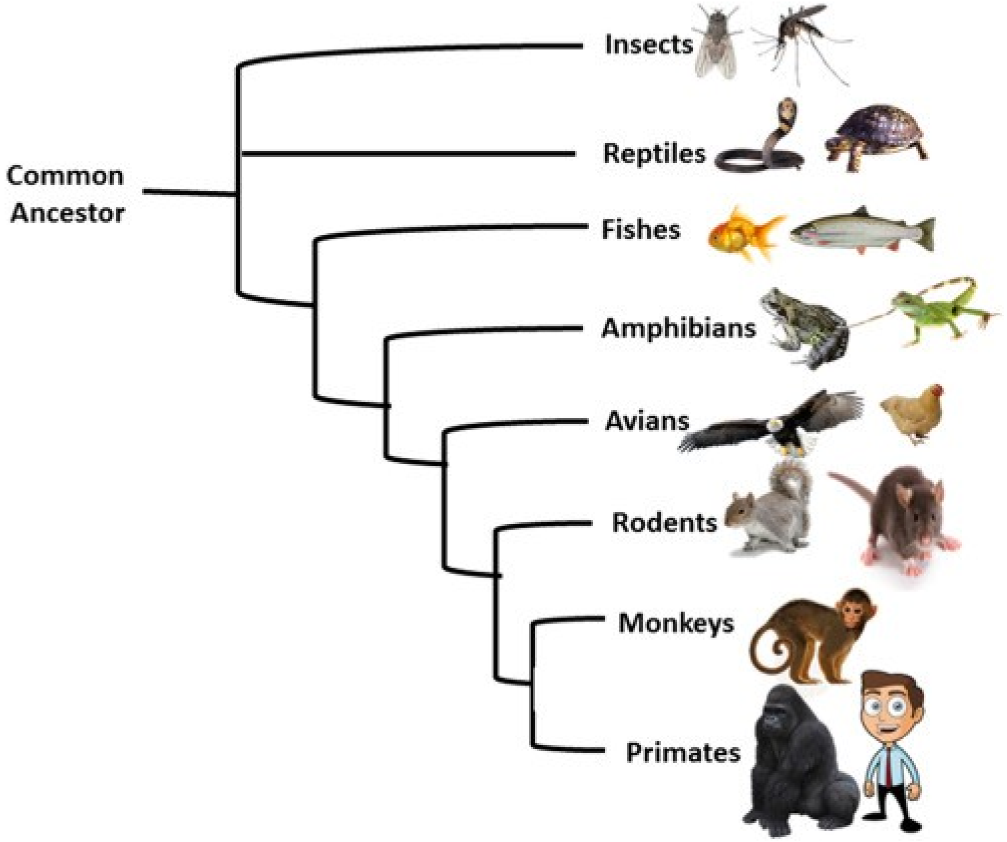
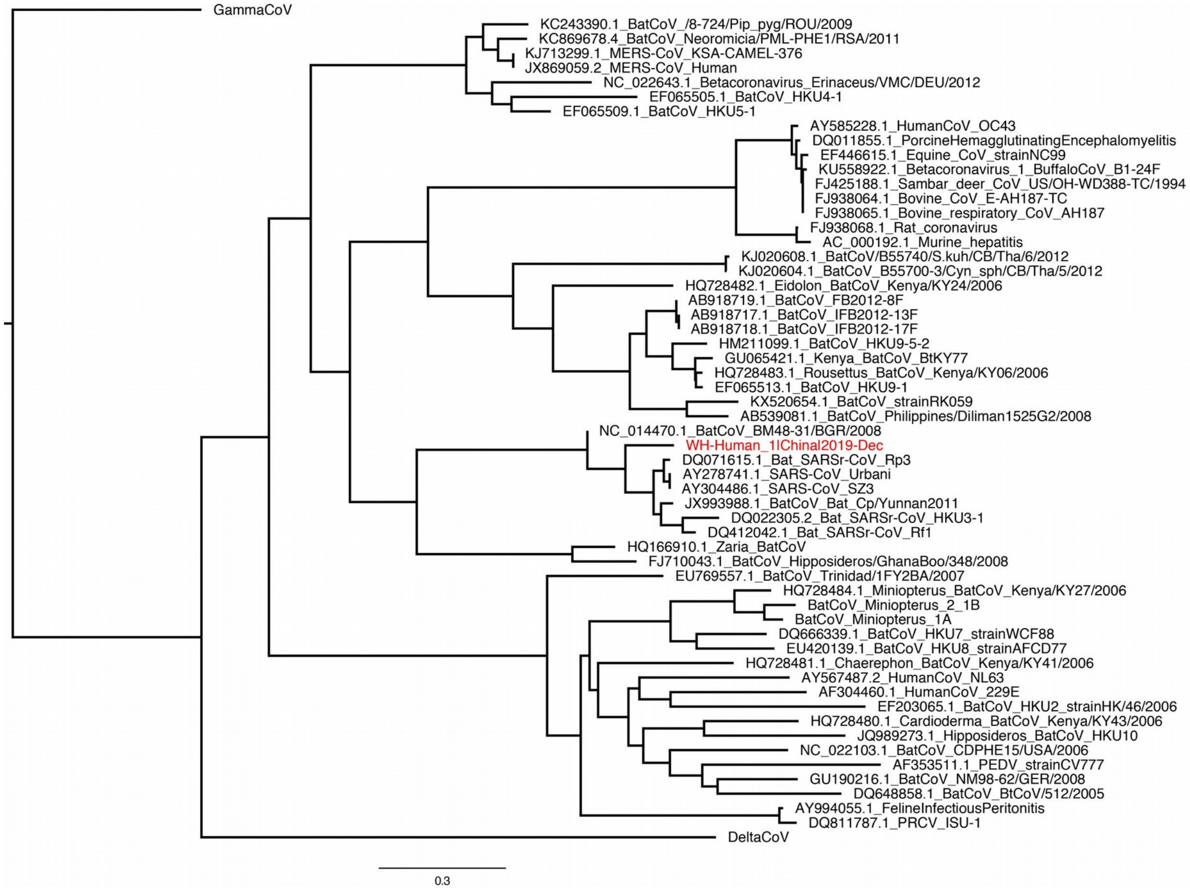
On the Elusiveness of Clusters

Steven M. Kelk   Celine Scornavacca   Leo van Iersel

Lukas Friedlos 22.04.21

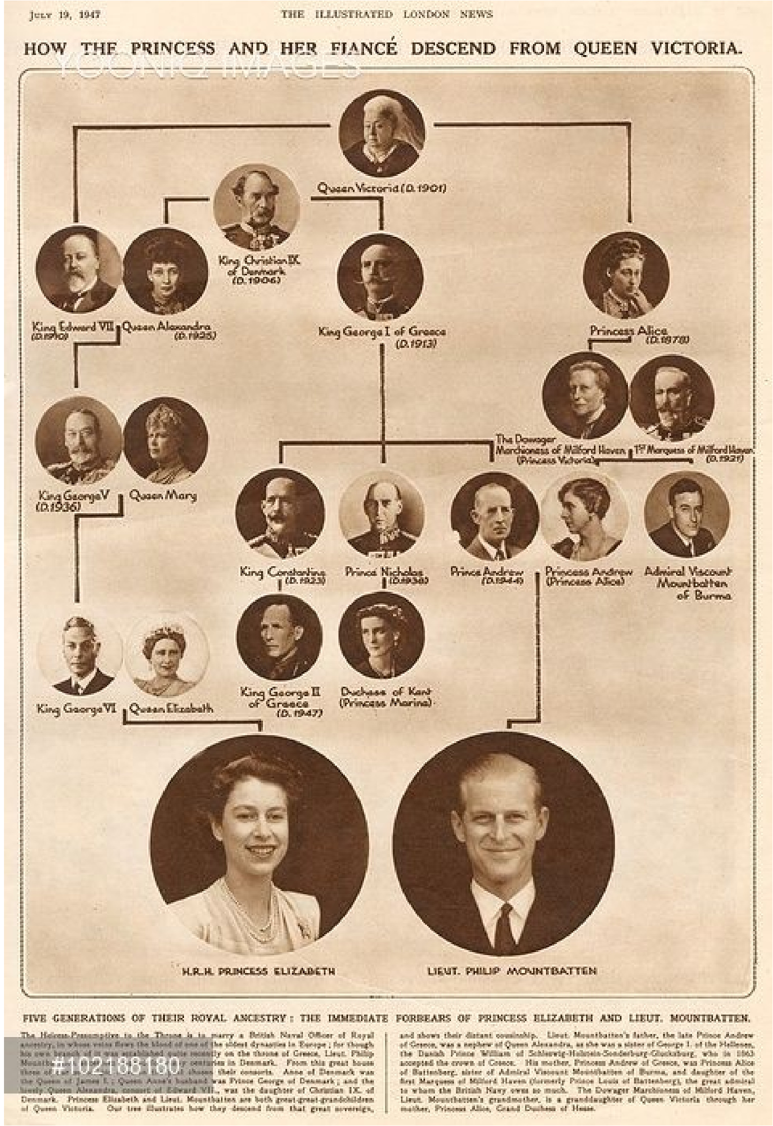

Common Ancestor
Insects
Reptiles
Fishes
Amphibians
Avians
Rodents
Monkeys
Primates



Preliminary maximum likelihood phylogenetic analysis of novel Wuhan, China human CoV in red, GenBank (accession MN908947)
Novel CoV seq data from: http://virological.org/t/initial-genome-release-of-novel-coronavirus/319. The Shanghai Public Health Clinical Center & School of Public Health, in collaboration with the Central Hospital of Wuhan, Huazhong University of Science and Technology, the Wuhan Center for Disease Control and Prevention, the National Institute for Communicable Disease Control and Prevention, Chinese Center for Disease Control, and the University of Sydney, Sydney, Australia.

PhyML tree based on partial RdRp gene sequence (410bp), aligned with representative human and animal CoV sequences from Genbank compiled by Alice Latinne; tree by Kevin Olival.
Analysis by EcoHealth Alliance - 11 Jan 2020 (12:30pm EST)

EcoHealth Alliance



July 19, 1947 THE ILLUSTRATED LONDON NEWS

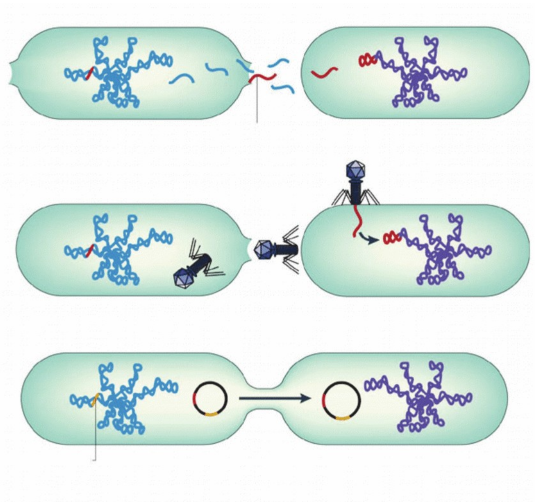
HOW THE PRINCESS AND HER FIANCÉ DESCEND FROM QUEEN VICTORIA.

FIVE GENERATIONS OF THEIR ROYAL ANCESTRY: THE IMMEDIATE FORBEARS OF PRINCESS ELIZABETH AND LIEUT. MOUNTBATTEN.

The Heiress-Presumptive to the Throne is to marry a British Naval Officer of Royal ancestry, in whose veins flows the blood of one of the oldest dynasties in Europe; for though his own branch of it was established quite recently on the throne of Greece, Lieut. Philip Mountbatten [illegible] centuries in Denmark. From this great house three of [illegible] chosen their consorts. Anne of Denmark was the Queen of James I.; Queen Anne's husband was Prince George of Denmark; and the lovely Queen Alexandra, consort of Edward VII., was the daughter of Christian IX. of Denmark. Princess Elizabeth and Lieut. Mountbatten are both great-great-grandchildren of Queen Victoria. Our tree illustrates how they descend from that great sovereign, and shows their distant cousinship. Lieut. Mountbatten's father, the late Prince Andrew of Greece, was a nephew of Queen Alexandra, as she was a sister of George I. of the Hellenes, the Danish Prince William of Schleswig-Holstein-Sonderburg-Glucksburg, who in 1863 accepted the crown of Greece. His mother, Princess Andrew of Greece, was Princess Alice of Battenberg, sister of Admiral Viscount Mountbatten of Burma, and daughter of the first Marquess of Milford Haven (formerly Prince Louis of Battenberg), the great admiral to whom the British Navy owes so much. The Dowager Marchioness of Milford Haven, Lieut. Mountbatten's grandmother, is a granddaughter of Queen Victoria through her mother, Princess Alice, Grand Duchess of Hesse.

Cross-Hybridisation

- Hybrid speciation (through sexual reproduction)
  - Homo Sapiens ↔ Neanderthal
  - Polar Bear ↔ Brown Bear
  - Most common in plants
- Horizontal gene transfer



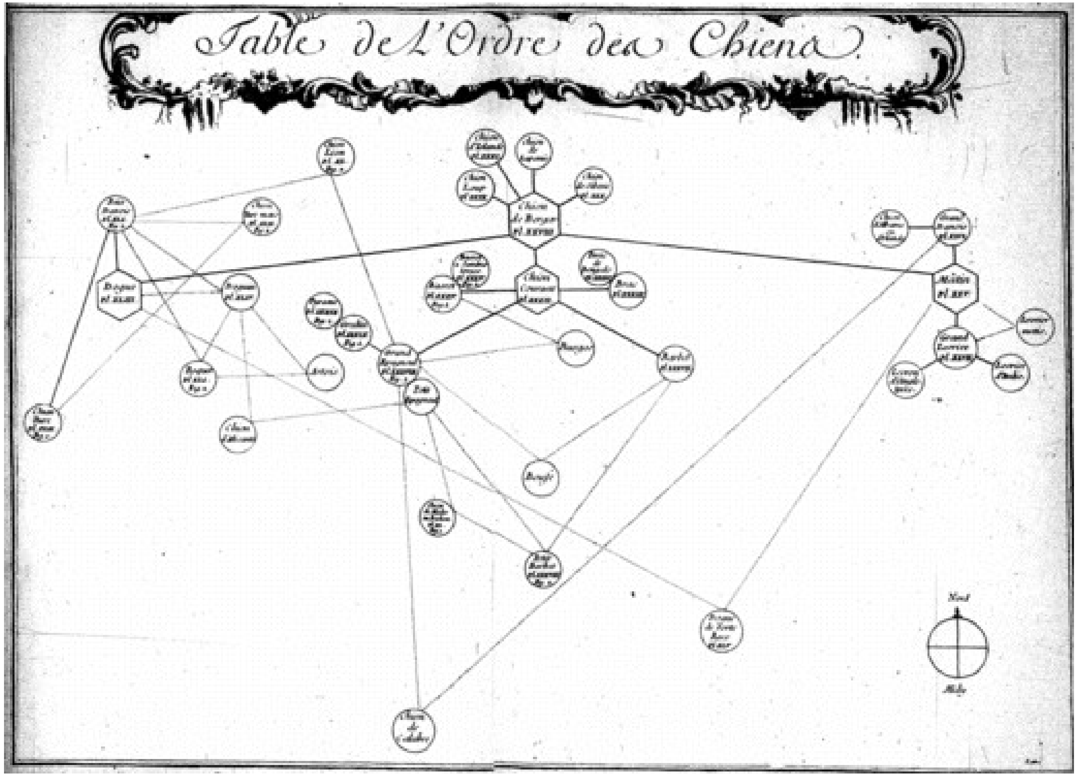
Horizontal gene transfer in bacteria

Heliconius heurippa

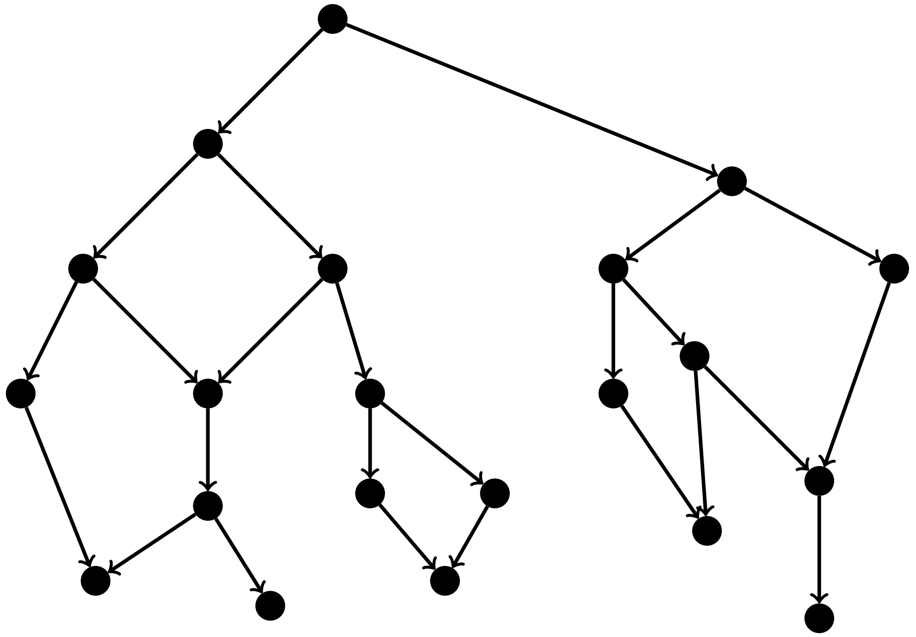
Jaguma

Prizzly

Phylogenetic Networks



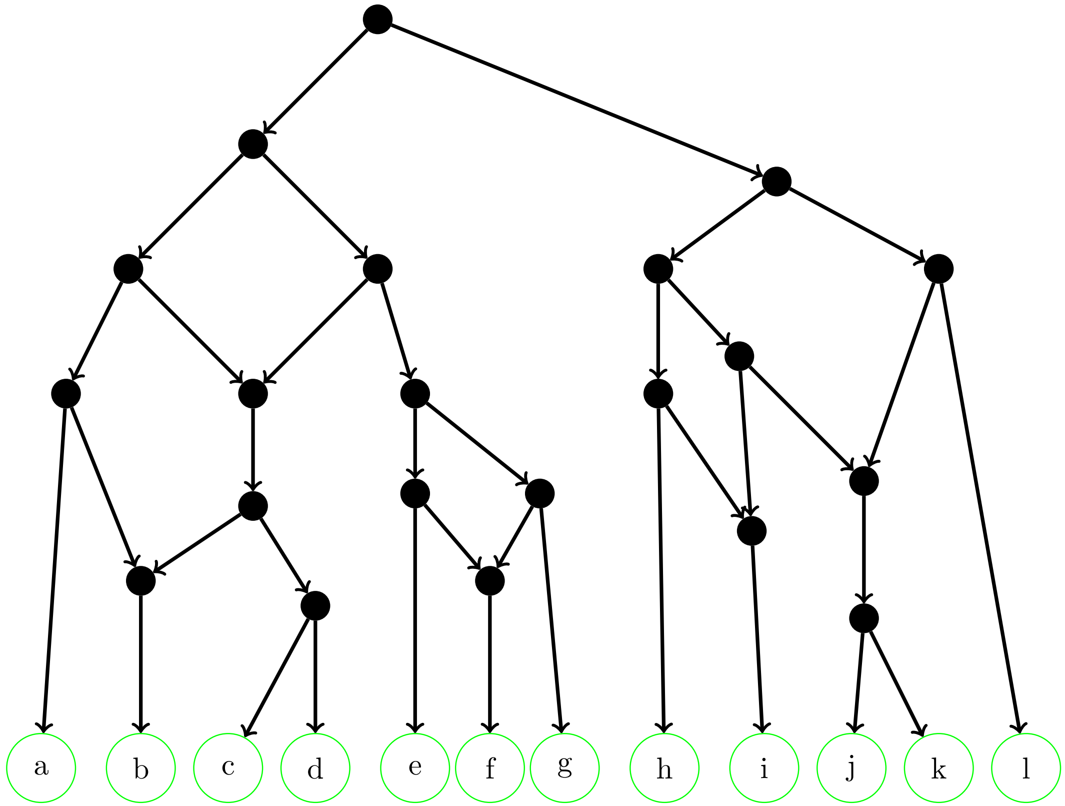
Phylogenetic Networks



Definition

- rooted directed acyclic graph
  Henceforth network N

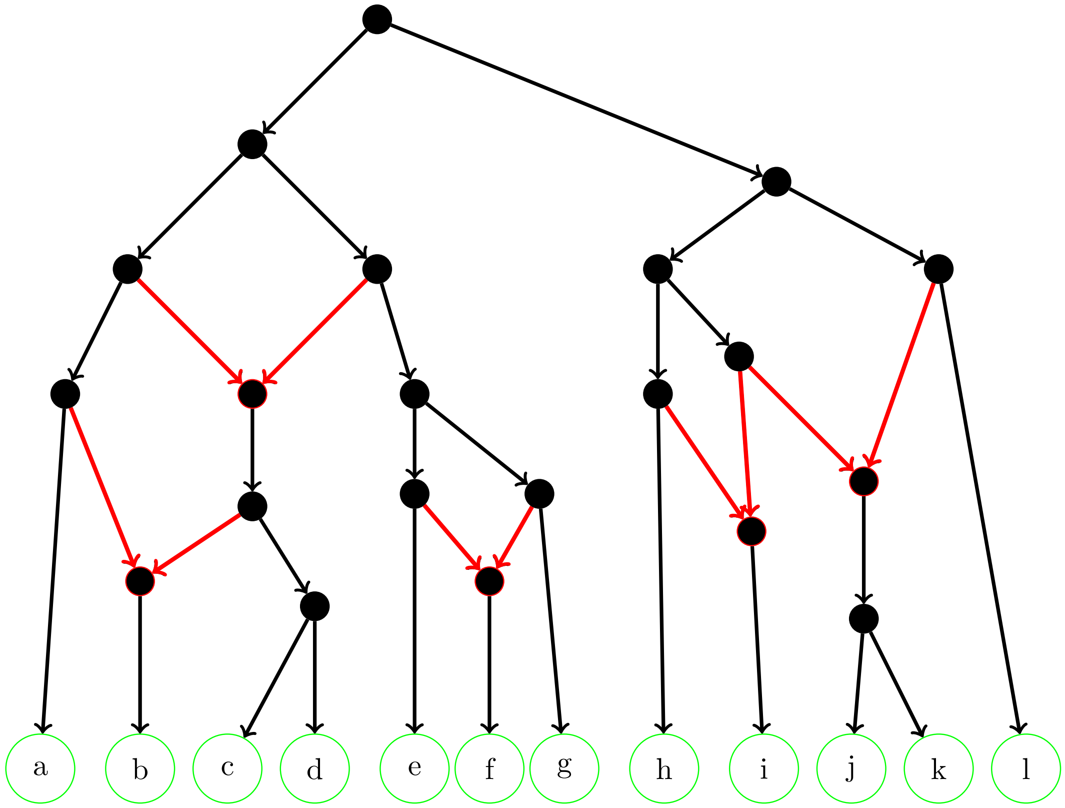
Phylogenetic Networks



Definition

- rooted directed acyclic graph
  Henceforth network N
- leaves representing a set of taxa
  $\mathcal{X} = \{a, b, c, d, e, f, g, h, i, j, k, l\}$

Phylogenetic Networks



Definition

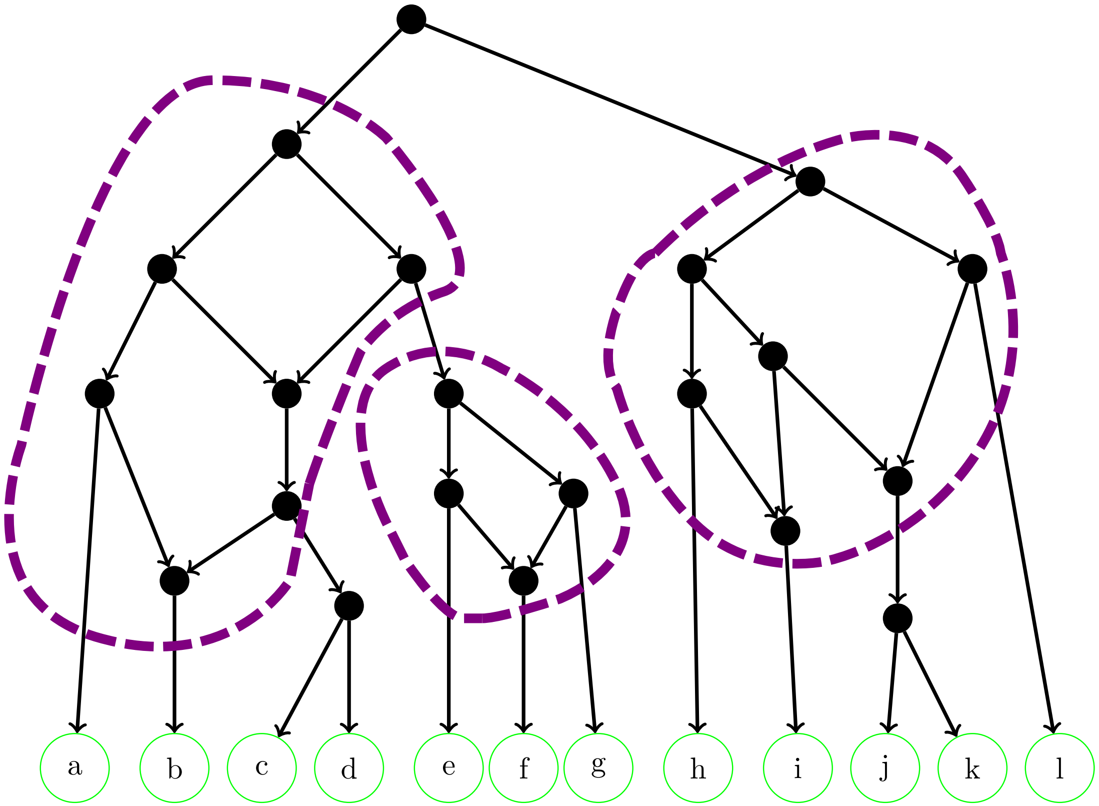
- rooted directed acyclic graph
  Henceforth network N
- leaves representing a set of taxa
  $\mathcal{X} = \{a, b, c, d, e, f, g, h, i, j, k, l\}$

Terminology

- reticulation

$$\text{in-deg}(v) \geq 2$$

Phylogenetic Networks



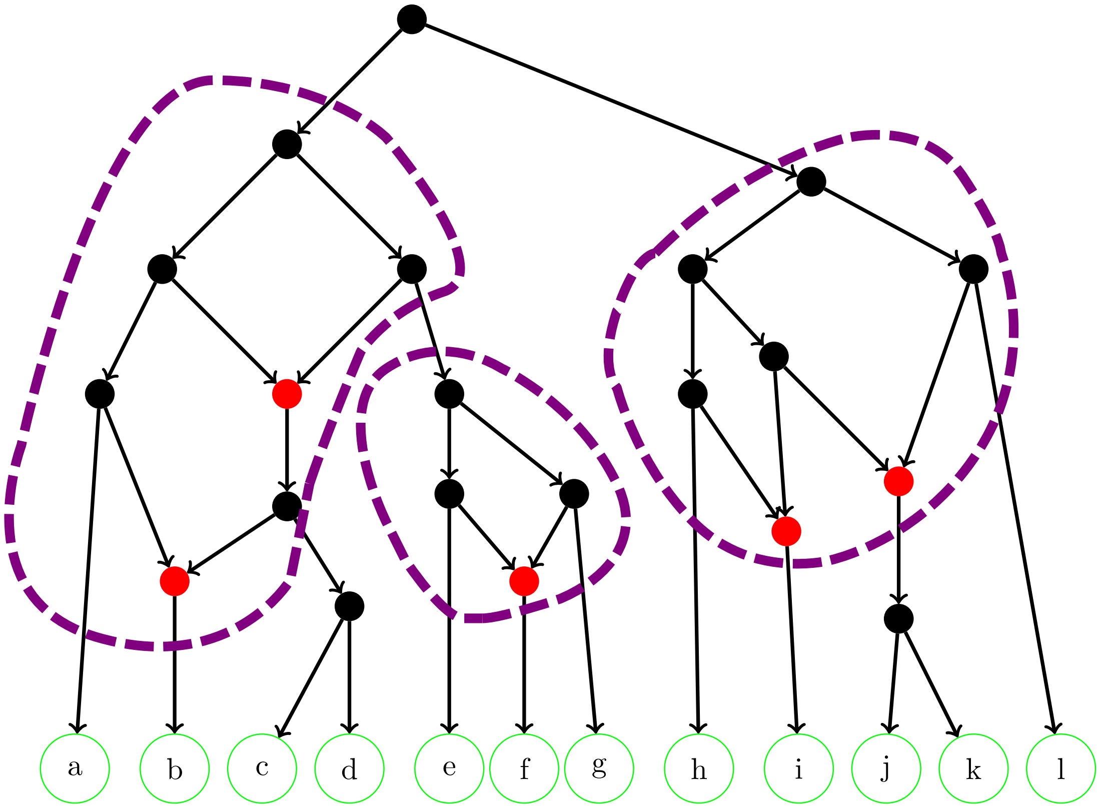
Definition

- rooted directed acyclic graph
  Henceforth network N
- leaves representing a set of taxa
  $\mathcal{X} = \{a, b, c, d, e, f, g, h, i, j, k, l\}$

Terminology

- reticulation
  $$\text{in-deg}(v) \geq 2$$
- biconnected components

Phylogenetic Networks



Definition

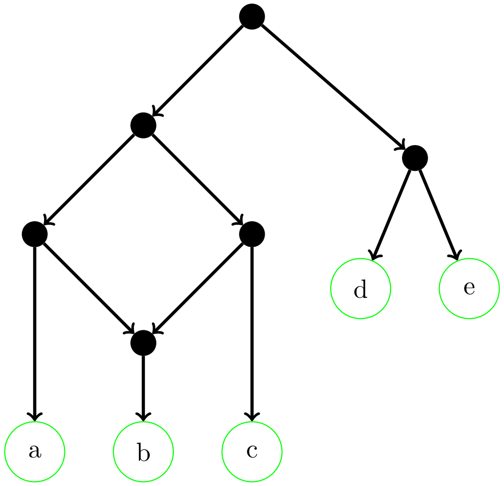
- rooted directed acyclic graph
  Henceforth network N
- leaves representing a set of taxa
  $\mathcal{X} = \{a, b, c, d, e, f, g, h, i, j, k, l\}$

Terminology

- reticulation
  $$\text{in-deg}(v) \geq 2$$
- biconnected components
- k -level

Network ‘Displaying’ a Tree

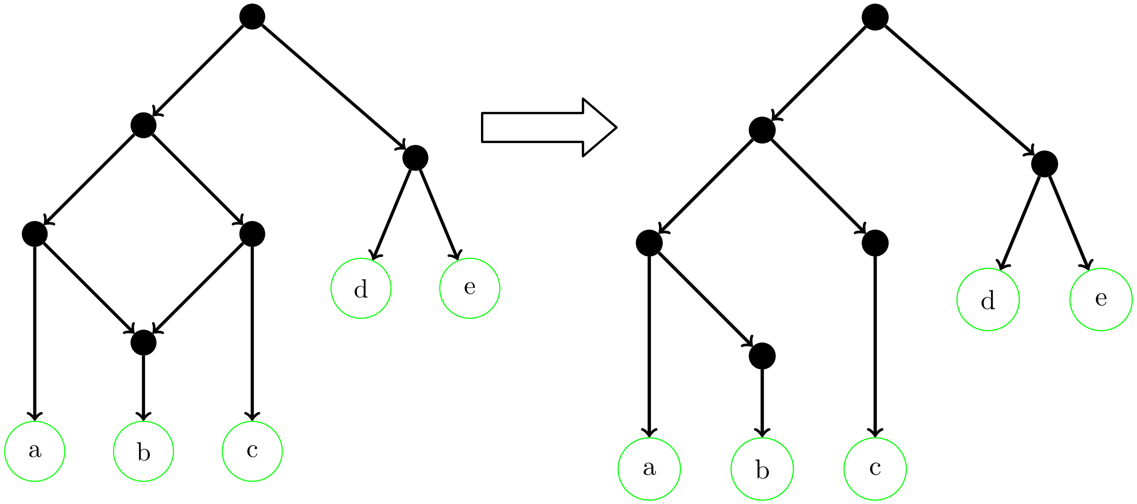
A Network N *displays* a tree T, if it can be obtained from N via



Network ‘Displaying’ a Tree

A Network N *displays* a tree T, if it can be obtained from N via

Node and/or Edge deletions

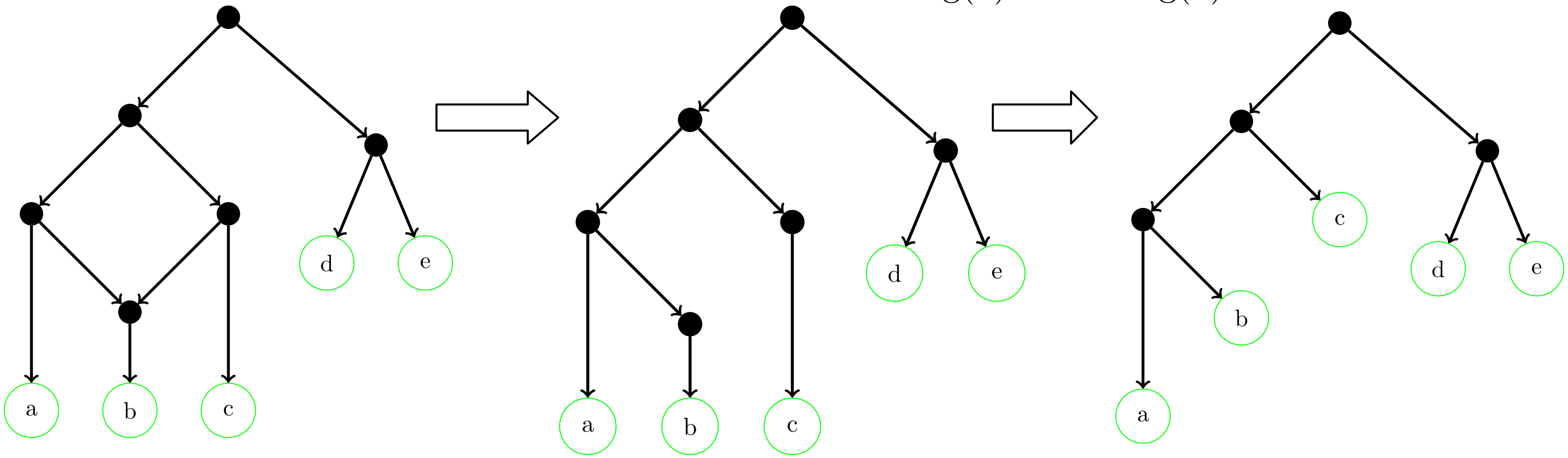


Network ‘Displaying’ a Tree

A Network N *displays* a tree T, if it can be obtained from N via

Node and/or Edge deletions

Suppressing Nodes with $\text{in-deg}(v) = \text{out-deg}(v) = 1$



Clusters

Given a set of taxa $\mathcal{X} = \{a, b, c, d, e\}$

A cluster $C \subset \mathcal{X}$ is a proper subset of all taxa, e.g. $\{a, b, c\}$

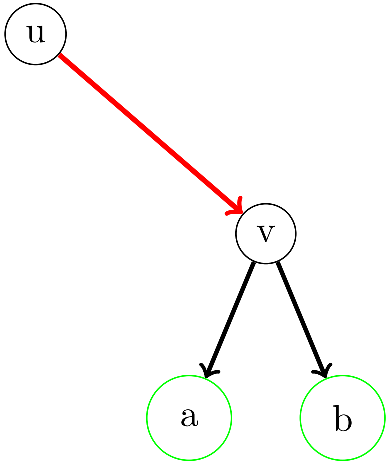
Taxa are in a cluster $\Longrightarrow$ Taxa have a common ancestor

A set of clusters $\mathcal{C}$ on a set of taxa, e.g. $\Big\{\{a, b, c\}, \{a, b\}, \{d, e\}\Big\}$

Networks represent Clusters

Representing Clusters

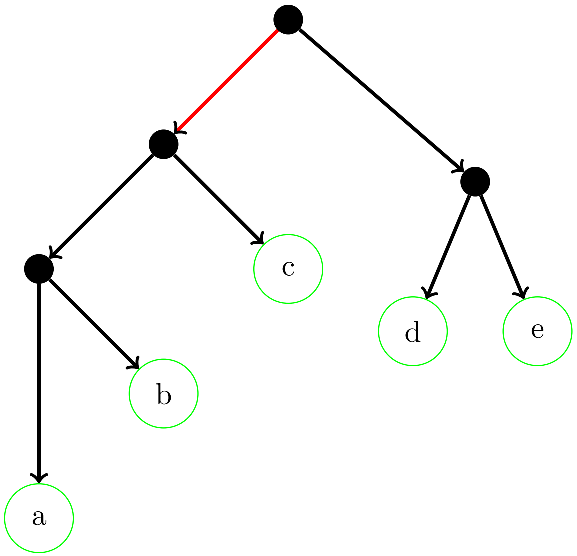
We say that an **edge** (u, v) *represents* a cluster C if C is the set of leaf descendants of v.



$$C = \{a, b\}$$

Representing Clusters

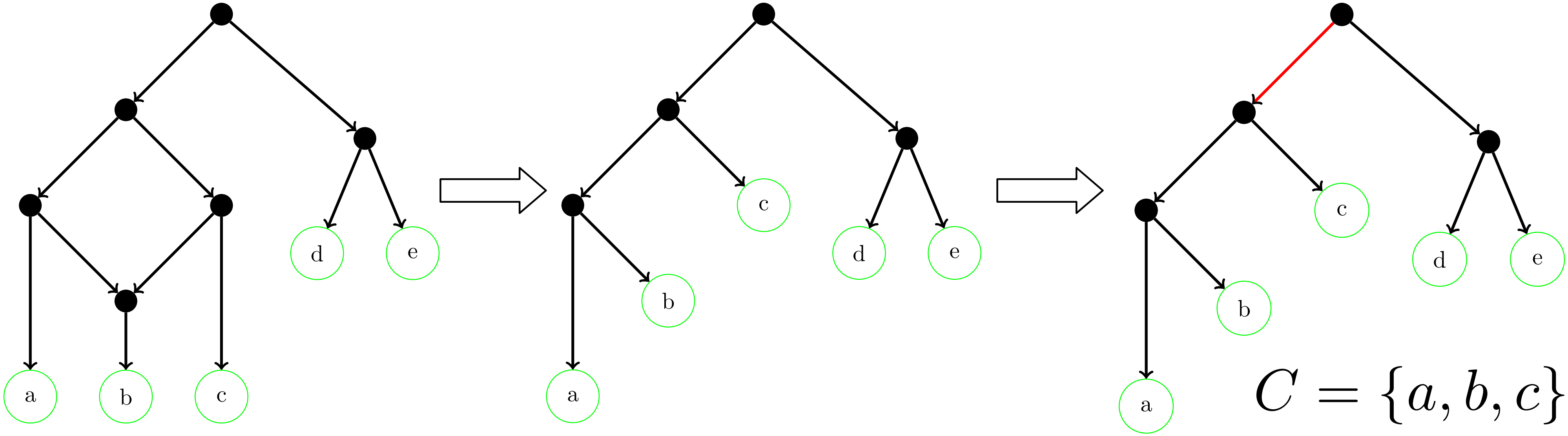
We say that a tree T *represents* a cluster C if T has an edge that represents C.



$$C = \{a, b, c\}$$

Representing Clusters

We say that a network N *represents* a cluster C if N displays a tree that represents C.



A Theoretical Polynomial-Time Algorithm for Constructing Level-k Networks

Goal

Given:

- set of clusters $\mathcal{C}$
- fixed $k \geq 0$

Goal

Given:
- set of clusters $\mathcal{C}$
- fixed $k \geq 0$

⇒

Construct level-k network N representing $\mathcal{C}$, if such a network exists.

Goal

Given:
- set of clusters $\mathcal{C}$
- fixed $k \geq 0$

$\Rightarrow$

Construct level-k network N representing $\mathcal{C}$, if such a network exists.

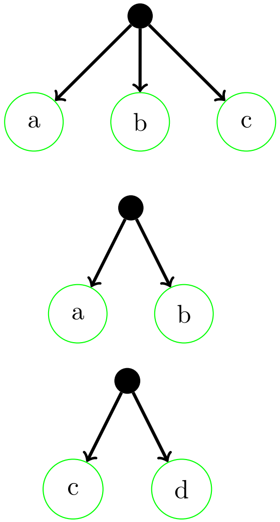
Possible in Polynomial-Time!

From Clusters to Networks (Trivial)

E.g. $\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \big\{\{a, b, c\}, \{a, b\}, \{c, d\}\big\}$

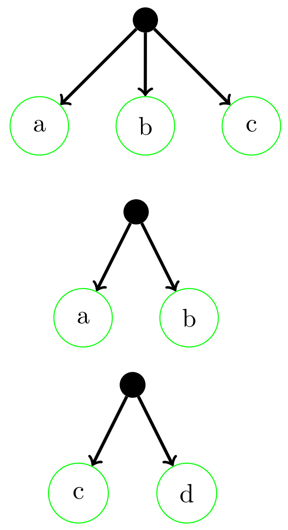
From Clusters to Networks (Trivial)

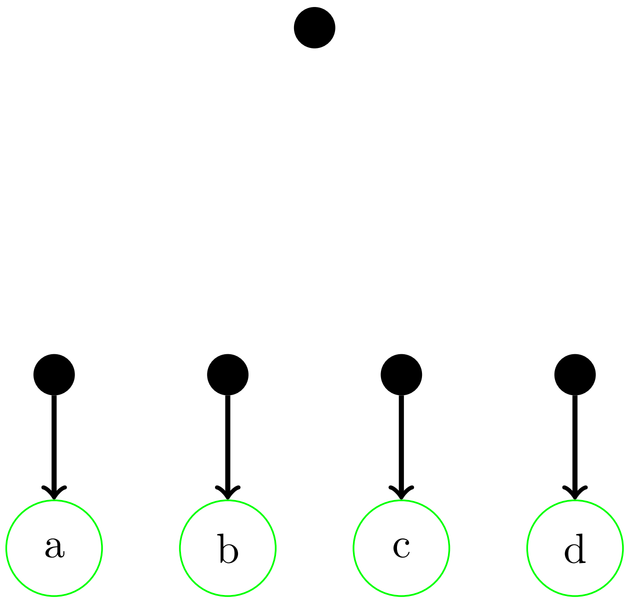
E.g. $\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$



From Clusters to Networks (Trivial)

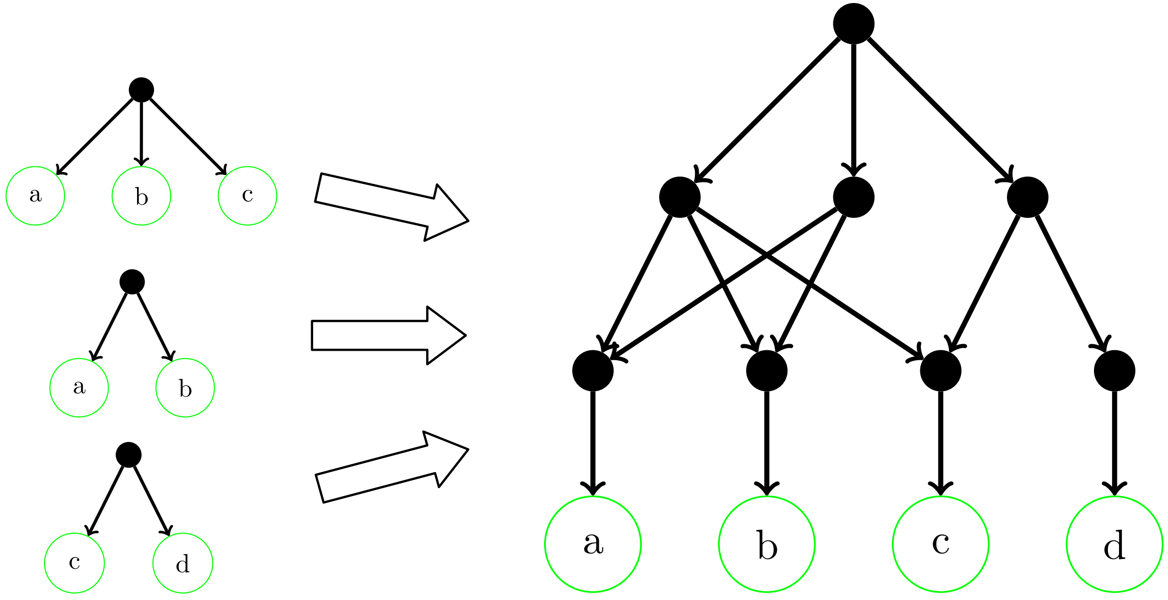
E.g. $\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$





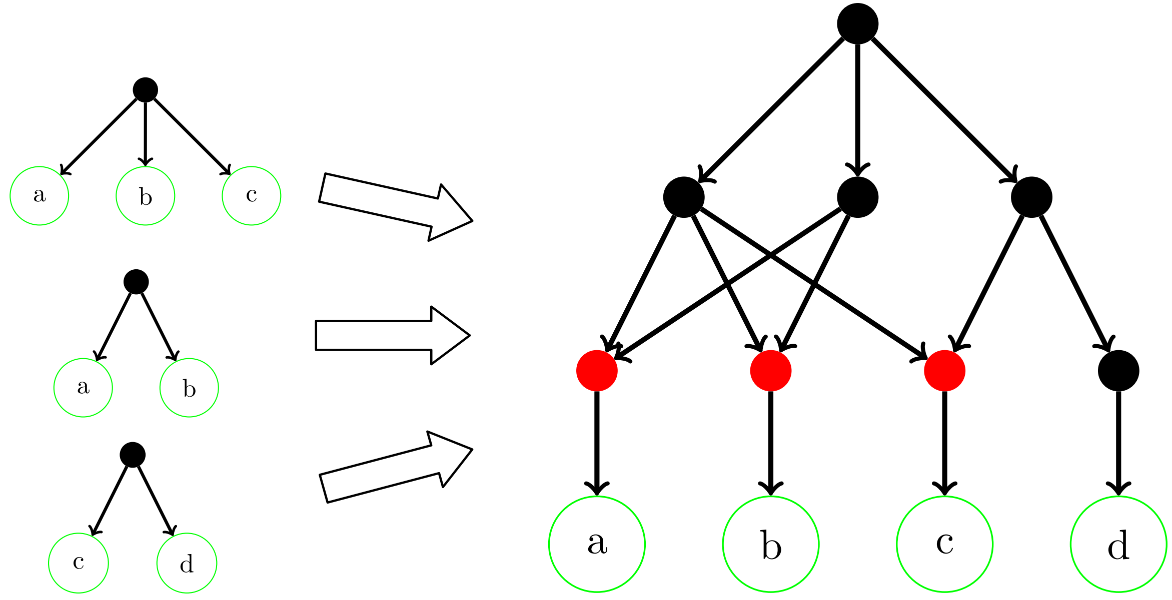
From Clusters to Networks (Trivial)

E.g. $\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$



From Clusters to Networks (Trivial)

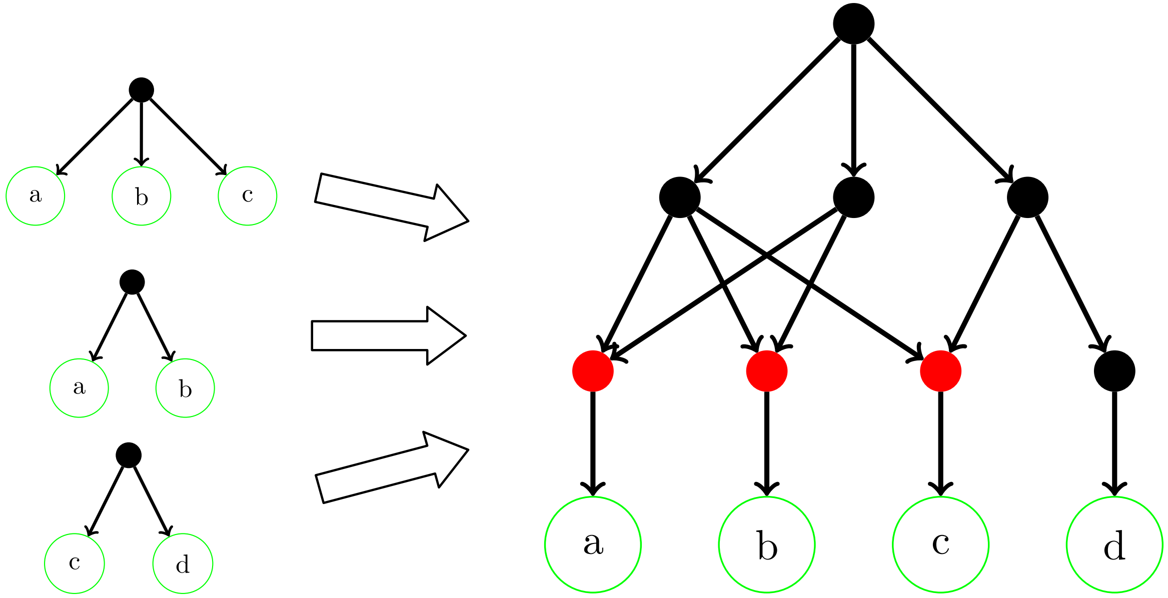
E.g. $\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$



$k = 3$

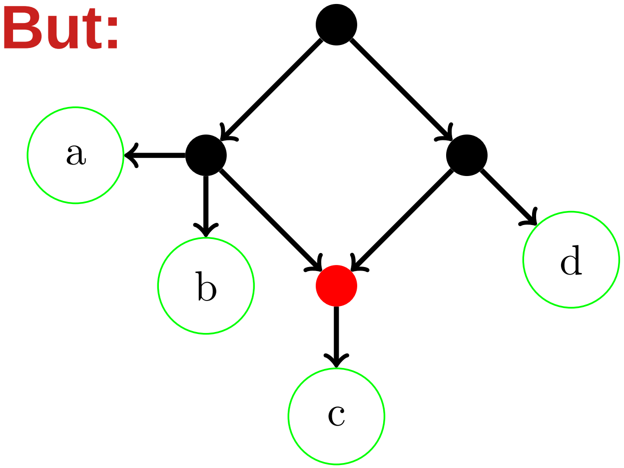
From Clusters to Networks (Trivial)

E.g. $\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$



$k = 3$

But:



$k = 1$

From Clusters to Networks

- Constructing a network representing clusters is trivial

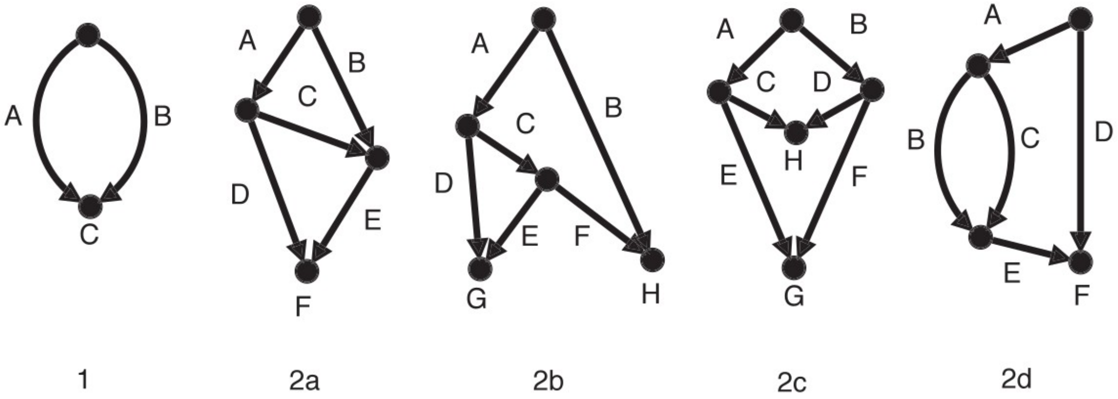
From Clusters to Networks

- Constructing a network representing clusters is trivial

- Constructing a network representing clusters with minimal level is NP-hard

Generators

Any level-k network can be reduced to a level-k *generator*



Edges and nodes with $\text{out-deg} = 0$ are called *sides*

Algorithm Outline

- Guess a generator for the network
- Build the network from the generator with guesses

Disclaimer: This algorithm is purely theoretical and doesn’t lend itself for practical implementation

Algorithm

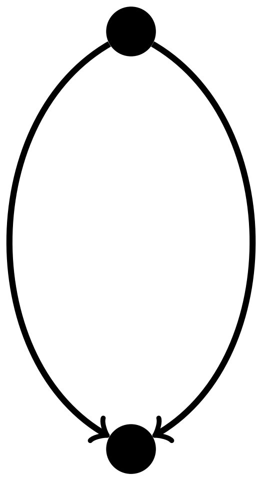
$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \Big\{\{a, b, c\}, \{a, b\}, \{c, d\}\Big\}$, $k = 1$

1. Guess a generator

Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \big\{\{a, b, c\}, \{a, b\}, \{c, d\}\big\}$, $k = 1$

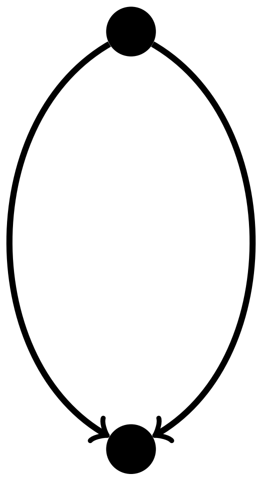
1. Guess a generator



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \Big\{\{a, b, c\}, \{a, b\}, \{c, d\}\Big\}$, $k = 1$

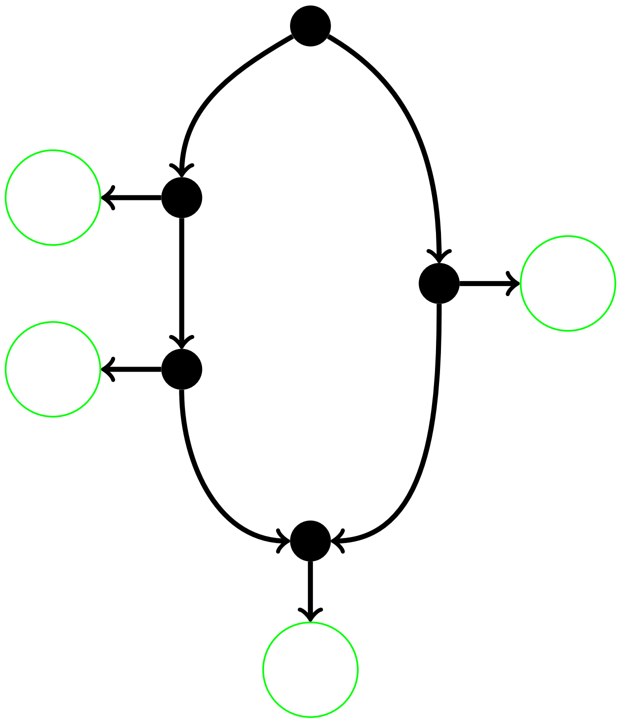
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$, $k = 1$

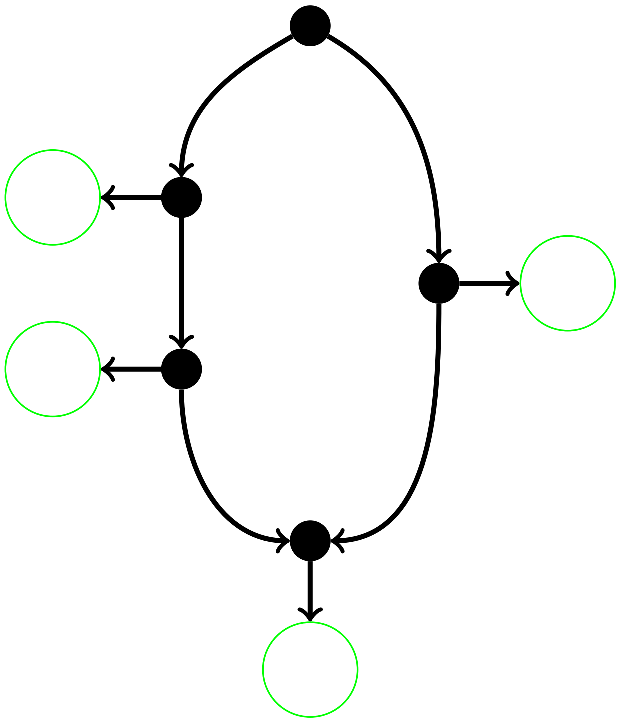
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$, $k = 1$

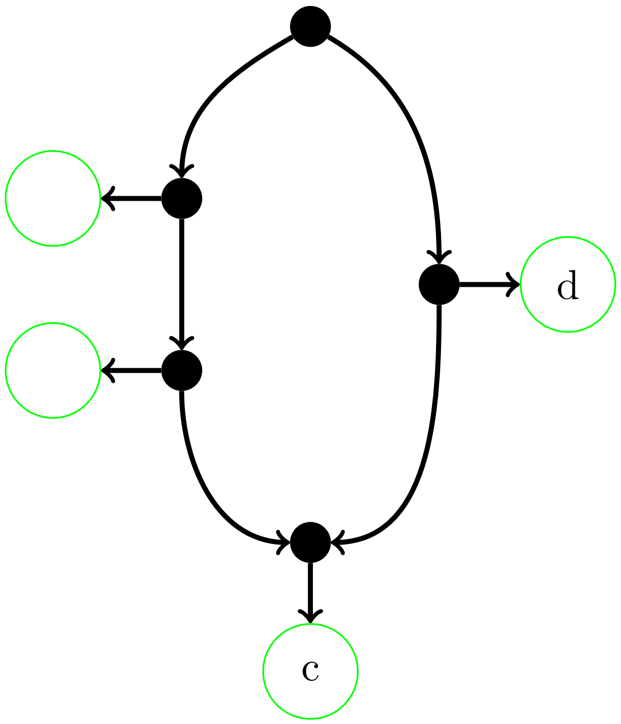
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side
3. Guess leaves of the sides with 1 leaf



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$, $k = 1$

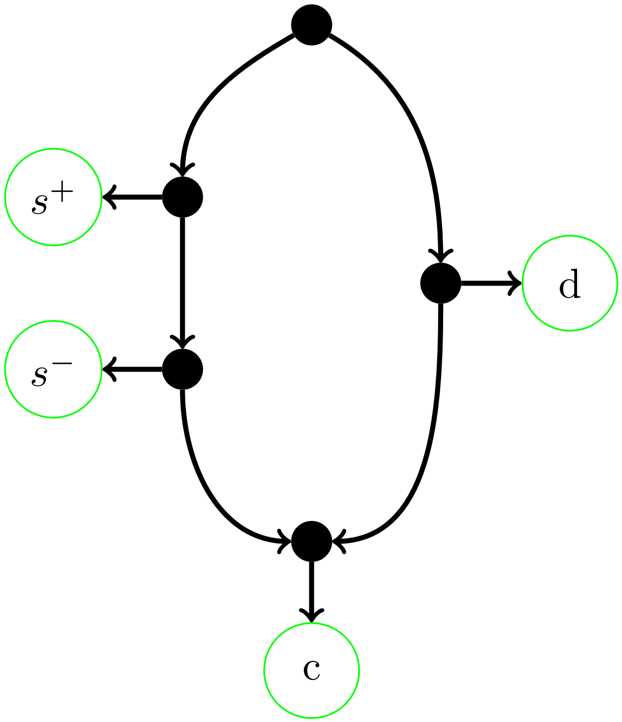
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side
3. Guess leaves of the sides with 1 leaf



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$, $k = 1$

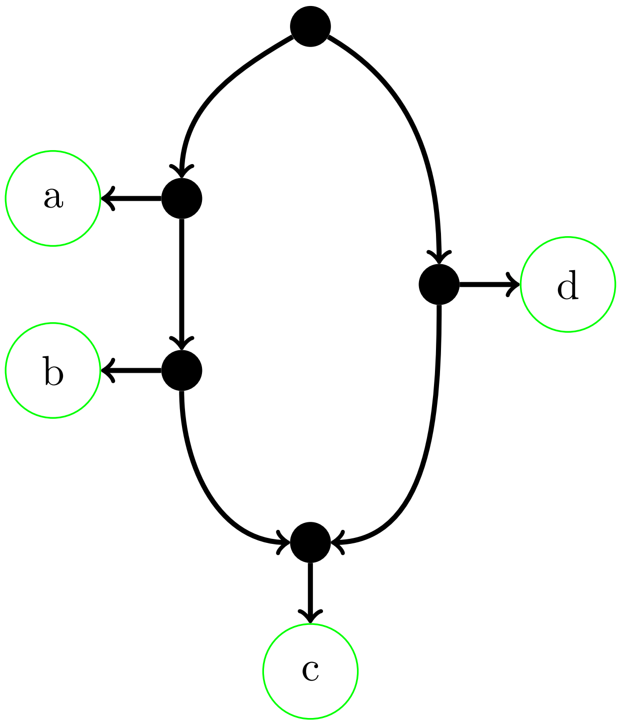
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side
3. Guess leaves of the sides with 1 leaf
4. Guess top s^+ and bottom s^- of the sides with ≥ 2 leaves



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$, $k = 1$

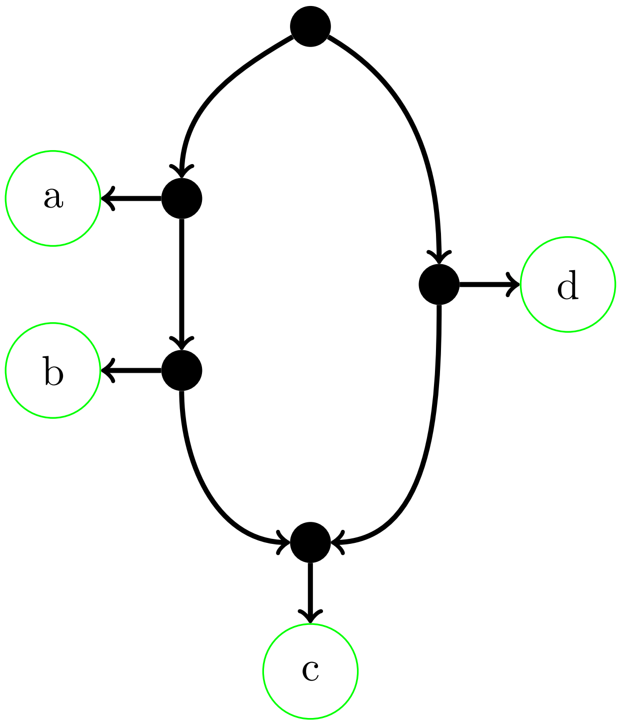
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side
3. Guess leaves of the sides with 1 leaf
4. Guess top s^+ and bottom s^- of the sides with ≥ 2 leaves



Algorithm

$\mathcal{X} = \{a, b, c, d\}$, $\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\}$, $k = 1$

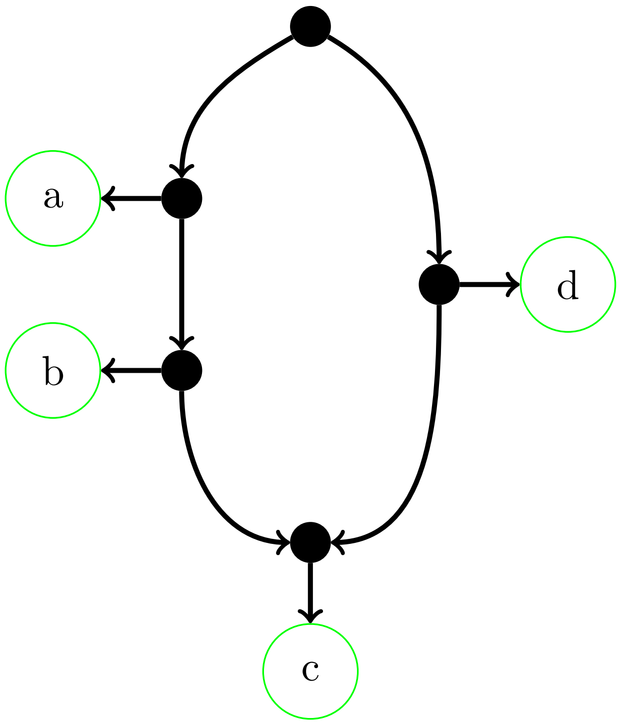
1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side
3. Guess leaves of the sides with 1 leaf
4. Guess top s^+ and bottom s^- of the sides with ≥ 2 leaves
5. Add remaining leaves



Algorithm

$$\mathcal{X} = \{a, b, c, d\},\ \mathcal{C} = \big\{\{a, b, c\}, \{a, b\}, \{c, d\}\big\},\ k = 1$$

1. Guess a generator
2. For each side, guess if it has 0, 1, 2, or more leaves on its side
3. Guess leaves of the sides with 1 leaf
4. Guess top s^+ and bottom s^- of the sides with ≥ 2 leaves
5. Add remaining leaves
6. Verify that the network actually represents $\mathcal{C}$



Adding Remaining Leaves

$$\mathcal{C} = \big\{\{a, b, c\}, \{a, b\}, \{c, d\}\big\}$$

Adding Remaining Leaves

$$\mathcal{C} = \big\{\{a,b,c\},\{a,b\},\{c,d\}\big\} \implies \begin{array}{l} a \to b \\ b \to a \\ c \to d \\ d \to c \end{array}$$

$a \to b$

=

"Every cluster containing a, also contains b"

Adding Remaining Leaves

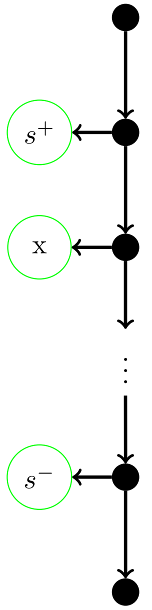
$$\mathcal{C} = \{\{a, b, c\}, \{a, b\}, \{c, d\}\} \Longrightarrow$$

$$a \to b$$
$$b \to a$$
$$c \to d$$
$$d \to c$$

$a \to b$

=

“Every cluster containing a , also contains b ”



Start with the lowest side

Based on s^+, s^- and the relationships from above, you can determine the leaf to be inserted in polynomial time

$s^+ \to x \to s^-$ must hold

Adding Remaining Leaves

$$\mathcal{C} = \{\{a,b,c\},\{a,b\},\{c,d\}\} \Longrightarrow \begin{array}{l} a \rightarrow b \\ b \rightarrow a \\ c \rightarrow d \\ d \rightarrow c \end{array}$$

$a \rightarrow b$

=

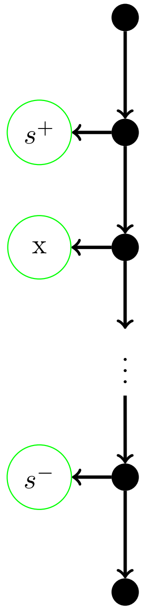
"Every cluster containing a , also contains b "

Start with the lowest side

Based on s^+, s^- and the relationships from above, you can determine the leaf to be inserted in polynomial time

$s^+ \rightarrow x \rightarrow s^-$ must hold

If no leaf can be added $\Longrightarrow$ Start new



Complexity Recap

- Guess from a **constant** number of generators
- For a **constant** number of sides, guess how many leaves from 4 options
- $\mathcal{O}(n^2)$ guesses for top and bottom
- Polynomial time for the remaining leaves
- Polynomial time to verify

⇒ **Very polynomial-time algorithm!**

Issues

Issues

- # generators explodes

Issues

- # generators explodes

$$[2^{k-1}, k!^2 50^k]$$

k	generators
1	1
2	4
3	65
4	1993
5	91454

Issues

- # generators explodes

$$[2^{k-1}, k!^2 50^k]$$

k	generators
1	1
2	4
3	65
4	1993
5	91454

- # edges (and hence sides) in the generators grows linearly in k

Issues

- # generators explodes

$$[2^{k-1}, k!^2 50^k]$$

k	generators
1	1
2	4
3	65
4	1993
5	91454

- # edges (and hence sides) in the generators grows linearly in k
- The need to guess $0, 1, 2, > 2$ leaves per side

Issues

- # generators explodes

$$[2^{k-1}, k!^2 50^k]$$

k	generators
1	1
2	4
3	65
4	1993
5	91454

- # edges (and hence sides) in the generators grows linearly in k
- The need to guess $0, 1, 2, > 2$ leaves per side
- The need to make up to $\mathcal{O}(n^2)$ guesses per side to guess s^+/s^-

Issues

- # generators explodes

$$[2^{k-1}, k!^2 50^k]$$

k	generators
1	1
2	4
3	65
4	1993
5	91454

- # edges (and hence sides) in the generators grows linearly in k
- The need to guess $0, 1, 2, > 2$ leaves per side
- The need to make up to $\mathcal{O}(n^2)$ guesses per side to guess s^+/s^-

In sum this makes the algorithm practically unfeasible :(

The End

Thank you for your attention!