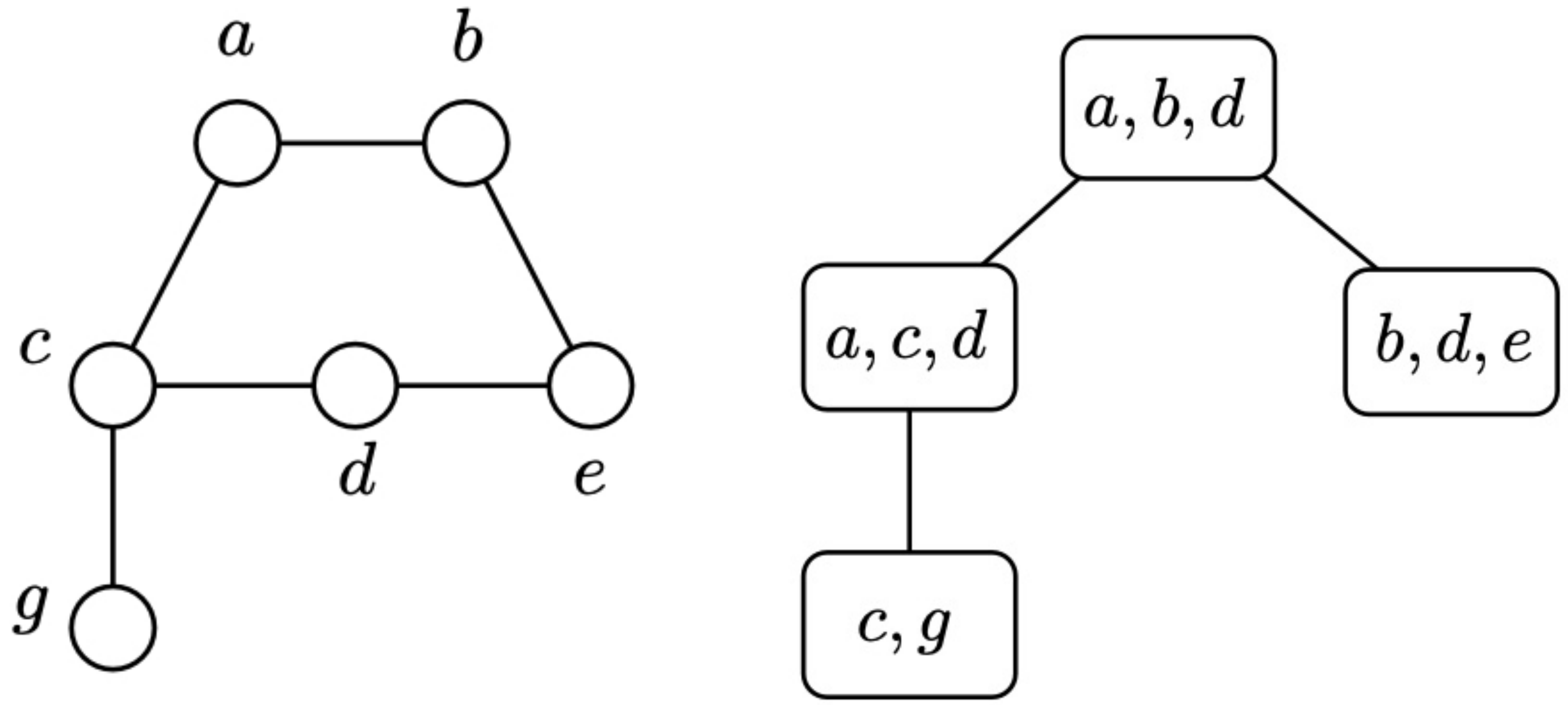
A meta-theorem for distributed certification

Distributed certification for “$\mathrm{tw} \leq k$ + MSO property” using $O(\log^2 n)$ bits – SIROCCO 2022

Pierre Fraigniaud (IRIF – CNRS), Pedro Montealegre (U. Adolfo Ibañez, Santiago), Ivan Rapaport (CMM – U. Chile), Ioan Todinca (LIFO – U. Orléans)

ANR ESTATE & DUCAT meeting, Cap Hornu, March 17, 2022

Outline



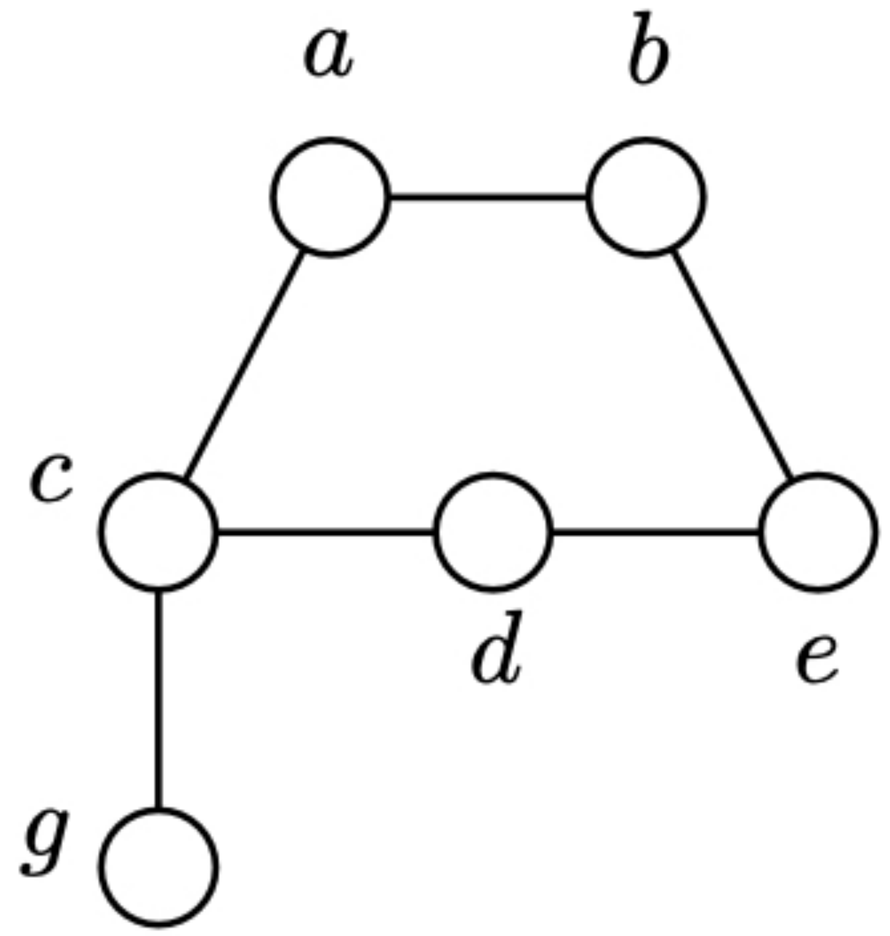
1. Tree decompositions, treewidth & Courcelle's theorem
2. **Distributed certification for small treewidth… approximation**
3. **Distributed certification for** "tw $\leq k$ + **MSO property**"
4. Conclusion

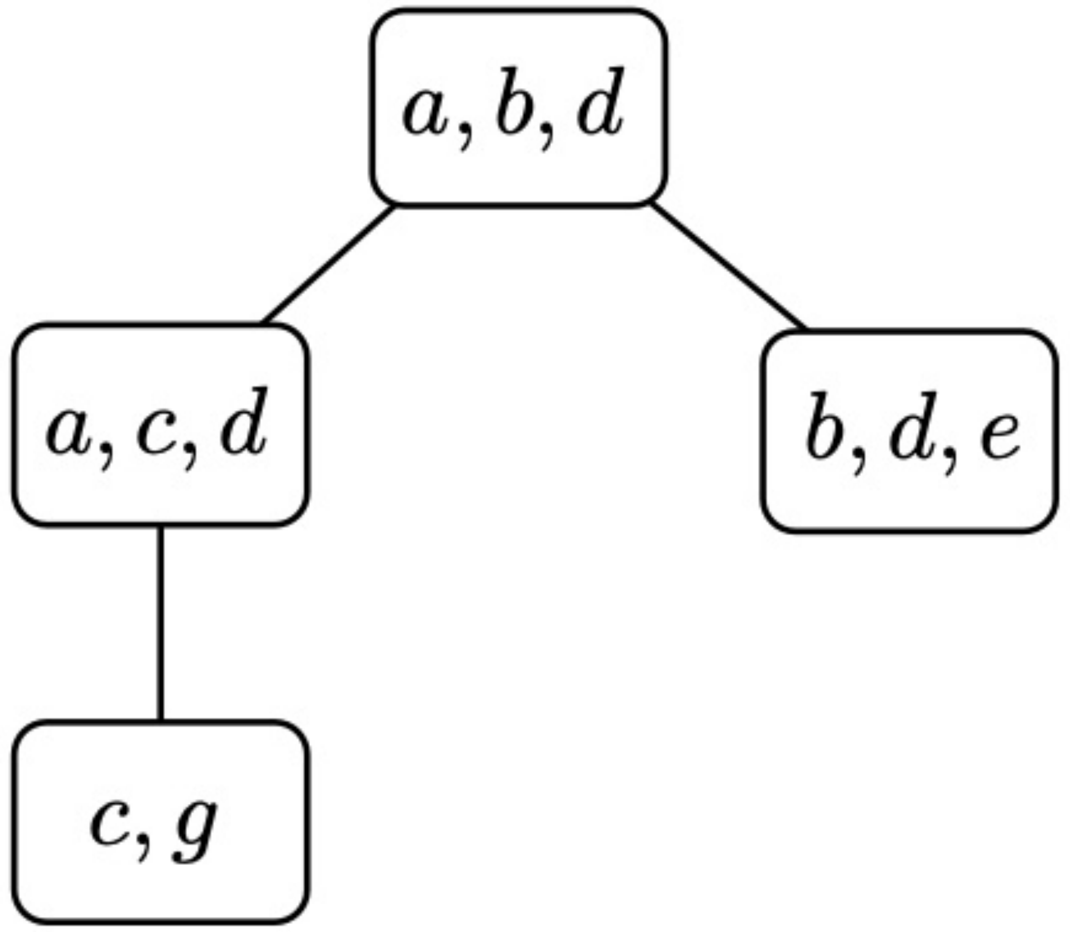
Related work: Bousquet, Feuilloley, Pierron '21 (arXiv): Local certification of MSO properties for bounded treedepth graphs

Tree decompositions and treewidth

Tree decomposition of $G = (V, E)$:

- A tree together with a bag (vertex subset of G) associated to each of its nodes
- Each vertex and each edge of G must be in some bag
- For each vertex of *G*, the bags containing it form a connected subtree





Treewidth $\text{tw}(G)$: the minimum k such that G has a tree decomposition with bags of size $\leq k+1$

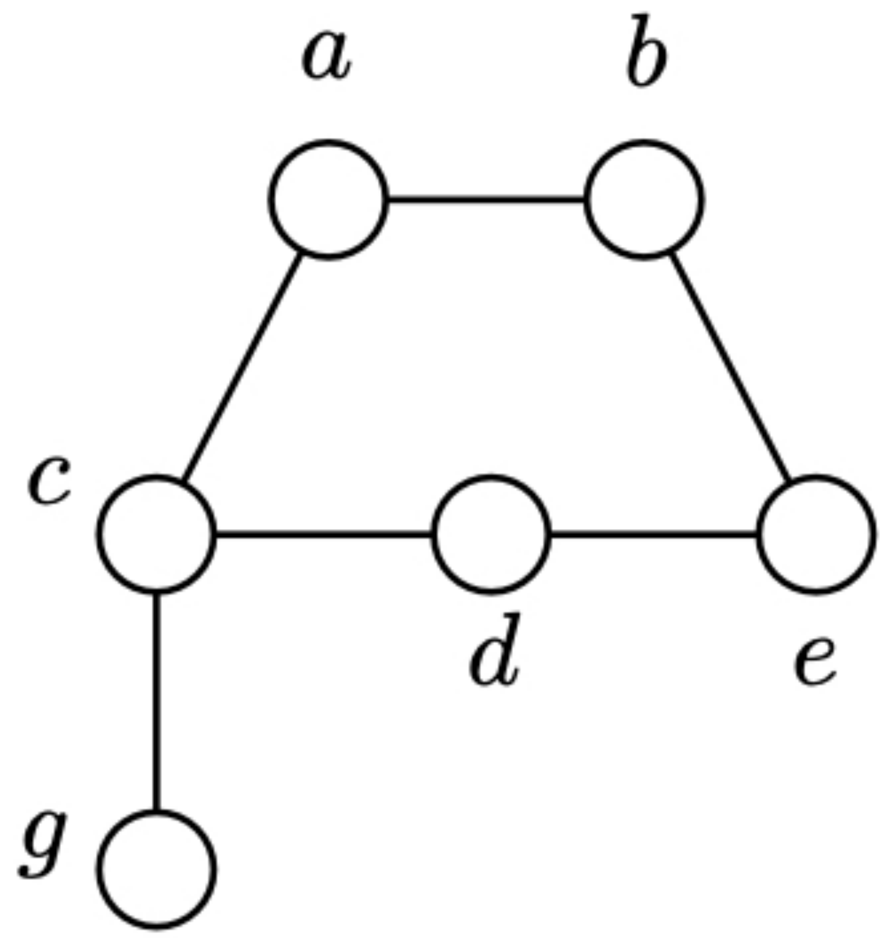
Why is treewidth important ?

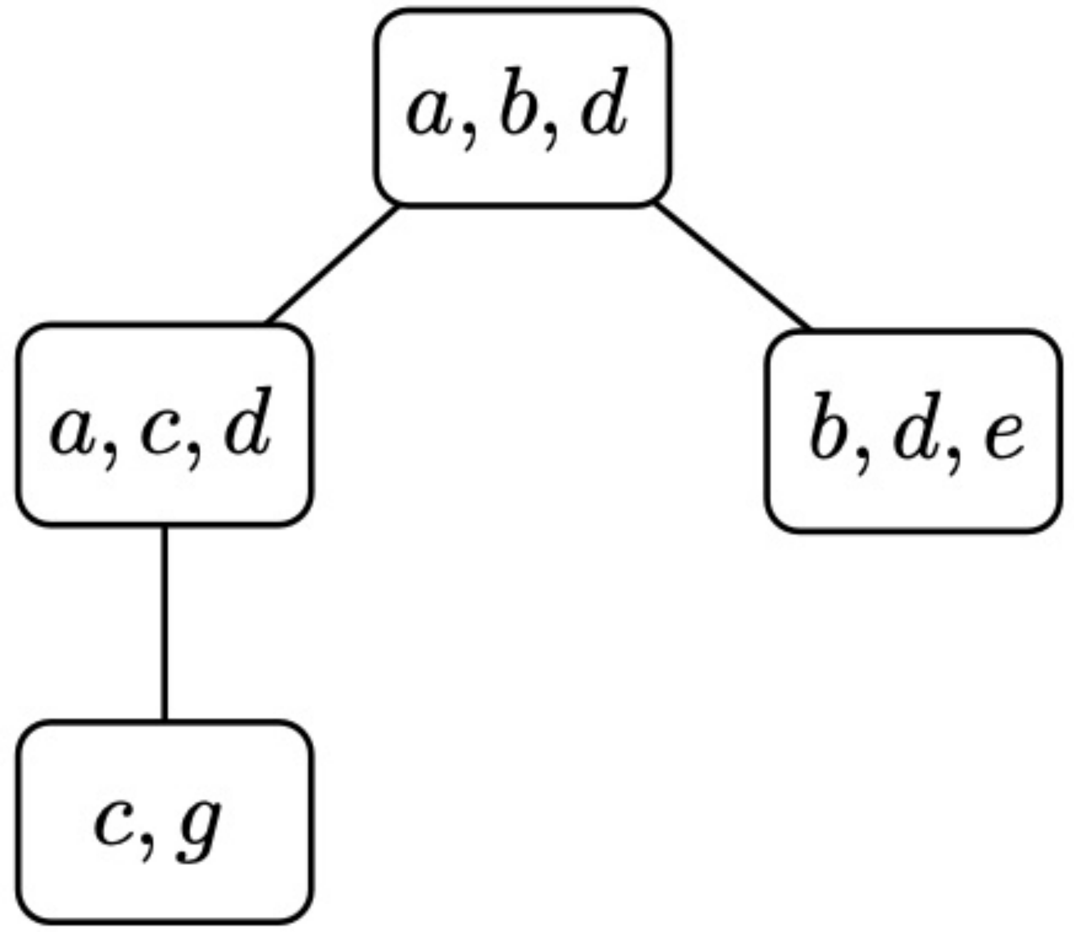
[A personal point of view]

- At the heart of the graph minors project (Robertson & Seymour) and a major starting point for parameterized algorithms (Downey & Fellows...).
- *Courcelle's (meta) theorem: every property expressible in monadic second order logic can be decided in $O(n)$ time on bounded treewidth graphs. Actually, $O(f(k, \varphi) \cdot n)$ time.*

$$
\begin{aligned}
&\exists Red, Greed, Blue \subseteq V : \\
&\quad (\forall x \in V, x \in Red \vee x \in Green \vee x \in Blue) \\
&\quad \wedge [\forall x, y \in E, (x \in Red \wedge y \in Red) \Rightarrow \neg \mathrm{adj}(x, y)] \\
&\quad \wedge [\forall x, y \in E, (x \in Green \wedge y \in Green) \Rightarrow \neg \mathrm{adj}(x, y)] \\
&\quad \wedge [\forall x, y \in E, (x \in Blue \wedge y \in Blue) \Rightarrow \neg \mathrm{adj}(x, y)]
\end{aligned}
$$

- Win-win techniques: parameterized algorithms for the disjoint paths problem on arbitrary graphs (R&S), parameters of planar graphs (bidimensionality — Demaine, Fomin, Hajiaghayi, Thilikos).

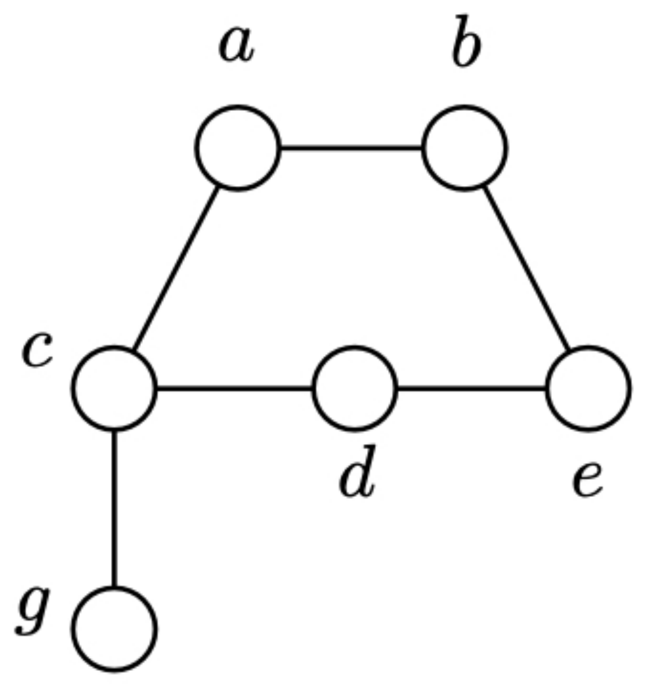


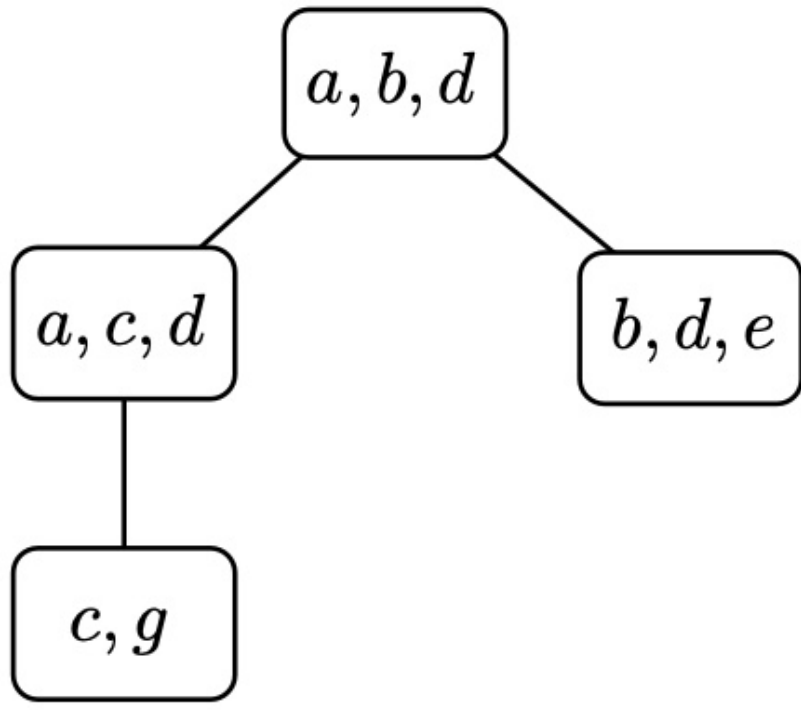


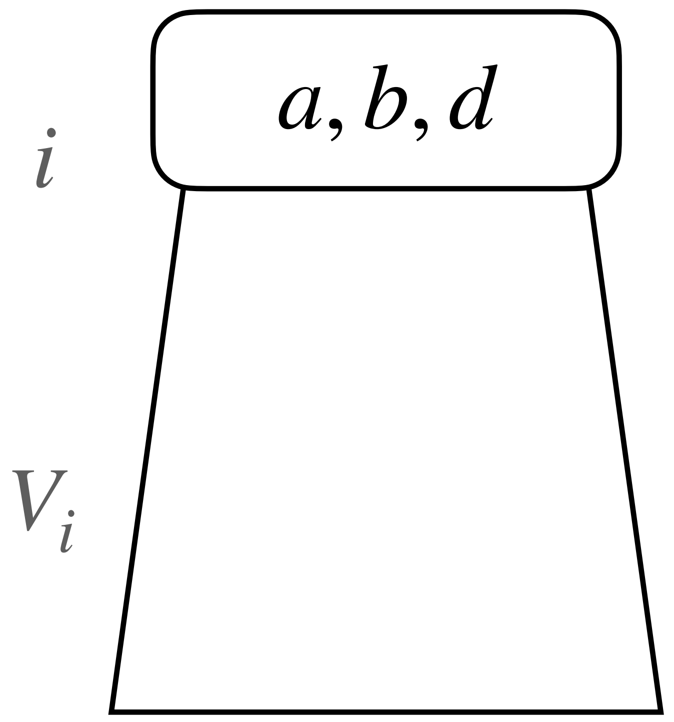
Courcelle's theorem

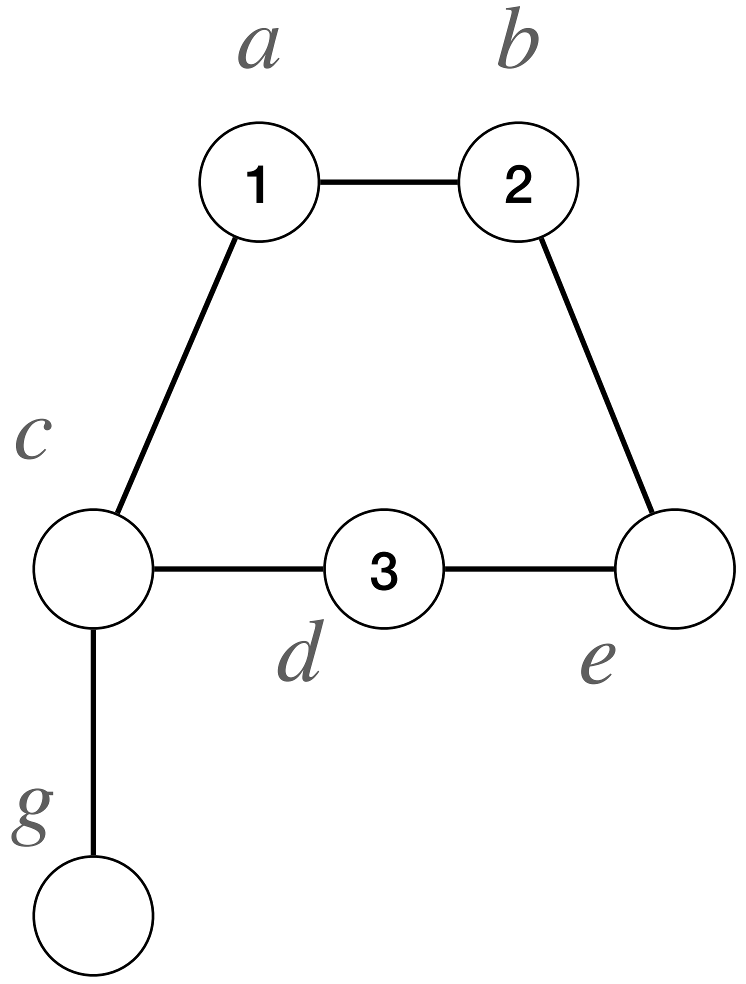
Every property $\mathscr{P}$ expressible in monadic second order logic can be decided in $O(n)$ time on bounded treewidth graphs.

- dynamic programming [Borie, Parker, Tovey '92]
- at each node i, store only the *homomorphism class* of property $\mathscr{P}$ for $G[V_i]$ and bag B_i
- the number of classes is bounded by a constant, depending on the property and on tw
- for leaf nodes, the homomorphism class is computed directly
- for other nodes i, the class is deduced from the ones of its children, and on the glueings of the children bags









3Colorability : the class is formed by all 3-partitions of the bag that can be extended into 3-colourings

Distributed certification for property $\mathscr{P}$

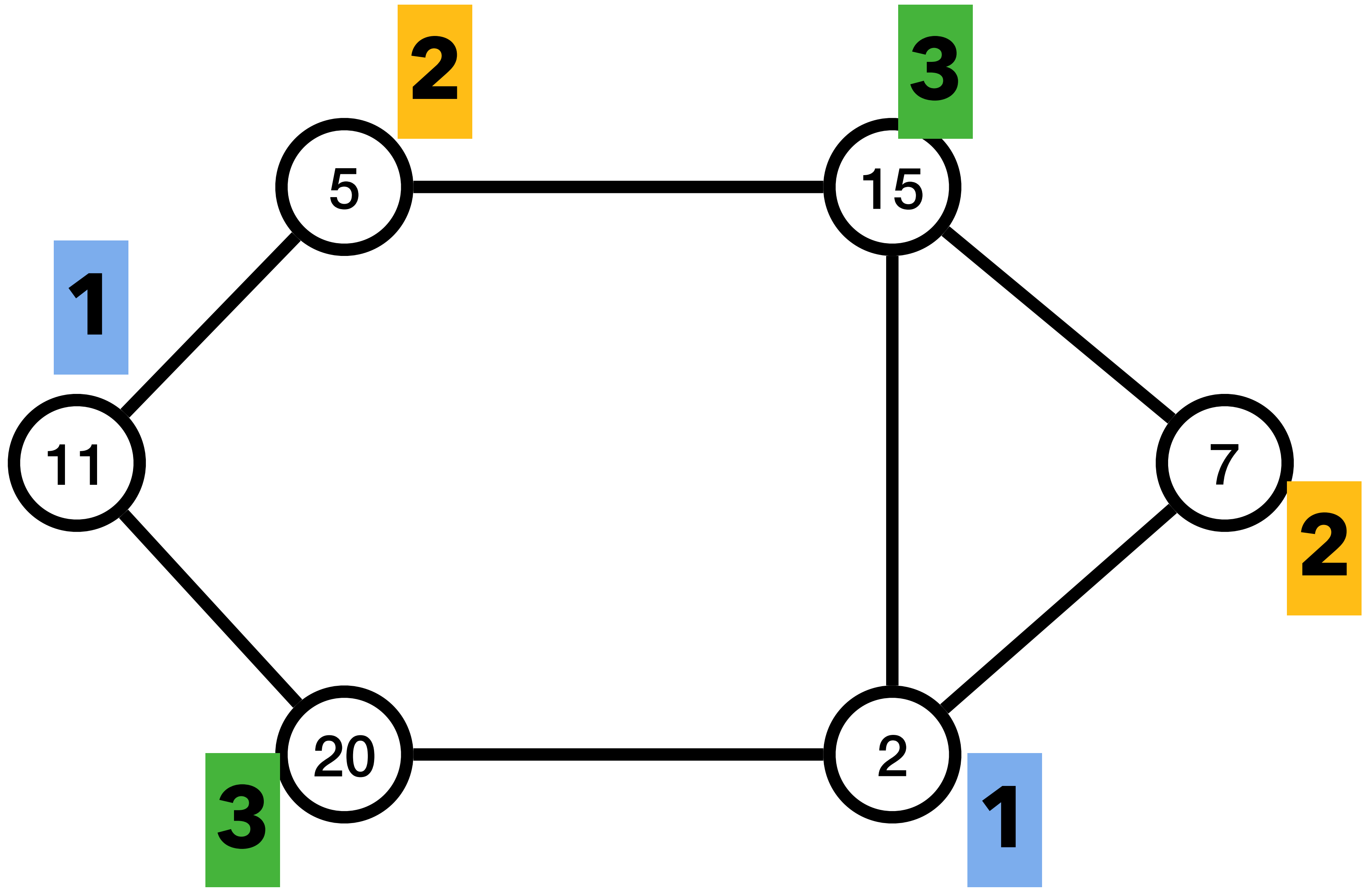
many variants… here, one round, determinist protocol

Centralized prover: knows the whole graph, assigns a (small) **certificate** to each node.

Distributed verifier: each nodes exchanges small messages with its neighbours (as in CONGEST), then accepts or rejects.

The prover is not trustable.

- **Completeness**: if $\mathscr{P}$ is true, there must exist a set of certificates s.t. all nodes accept;
- **Soundness**: if $\mathscr{P}$ is false, for any set of certificates, at least one node rejects.



3Colorability: easy, certificates of 2 bits. Non-3Colorability: hard...

Distributed certification for spanning tree

$O(\log n)$ **certificates**

Centralized prover to each vertex v: $(r = root_T, parent_T(v), distRoot_T(v))$

Distributed verifier for vertex v :

- if $distRoot_T(v) \neq 0$ check that $distRoot_T(parent_T(v)) = distRoot_T(v) - 1$
  → detects cycles or incoherences

- check that all neighbours got the same r

- if $distRoot_T(v) = 0$ check that $v = r$
  → ensure that T has a unique connected component

Property tw $\leq k$**: certifying a 3-approximation**

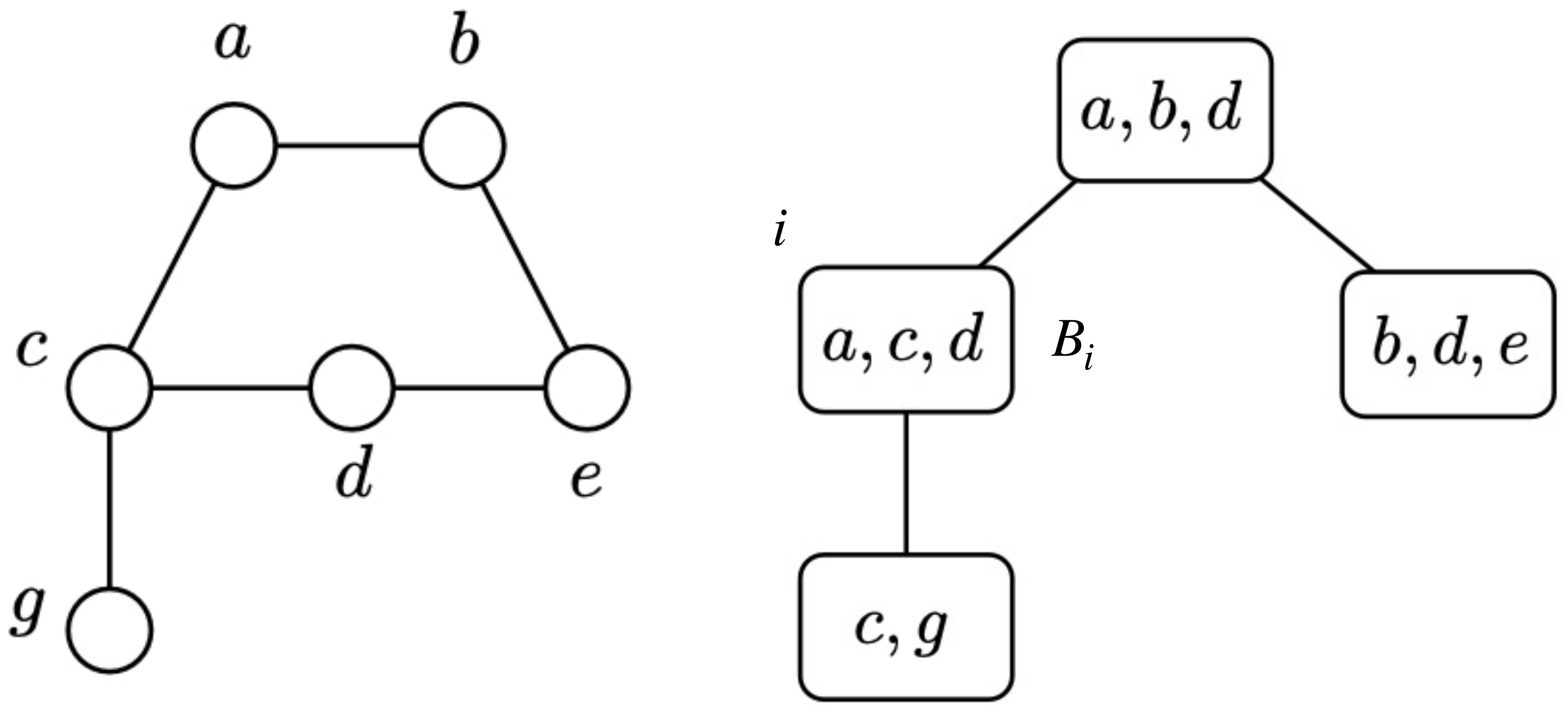
certificates & messages of size $O(k^2 \log^2 n)$

tw $\leq k \Rightarrow$ there exists a certificate assignment s.t. all vertices accept

tw $> 3k + 2 \Rightarrow$ for any certificate assignment, at least one vertex rejects

Graphs of tw $\leq k$ have **coherent tree decompositions** [Bodlaender '88]

1. decomposition tree of **depth** $O(\log n)$,
2. bags of **size** $\leq 3k + 3$,
3. **connectivity** of $G[V_i \backslash B_{p(i)}]$ for all i.



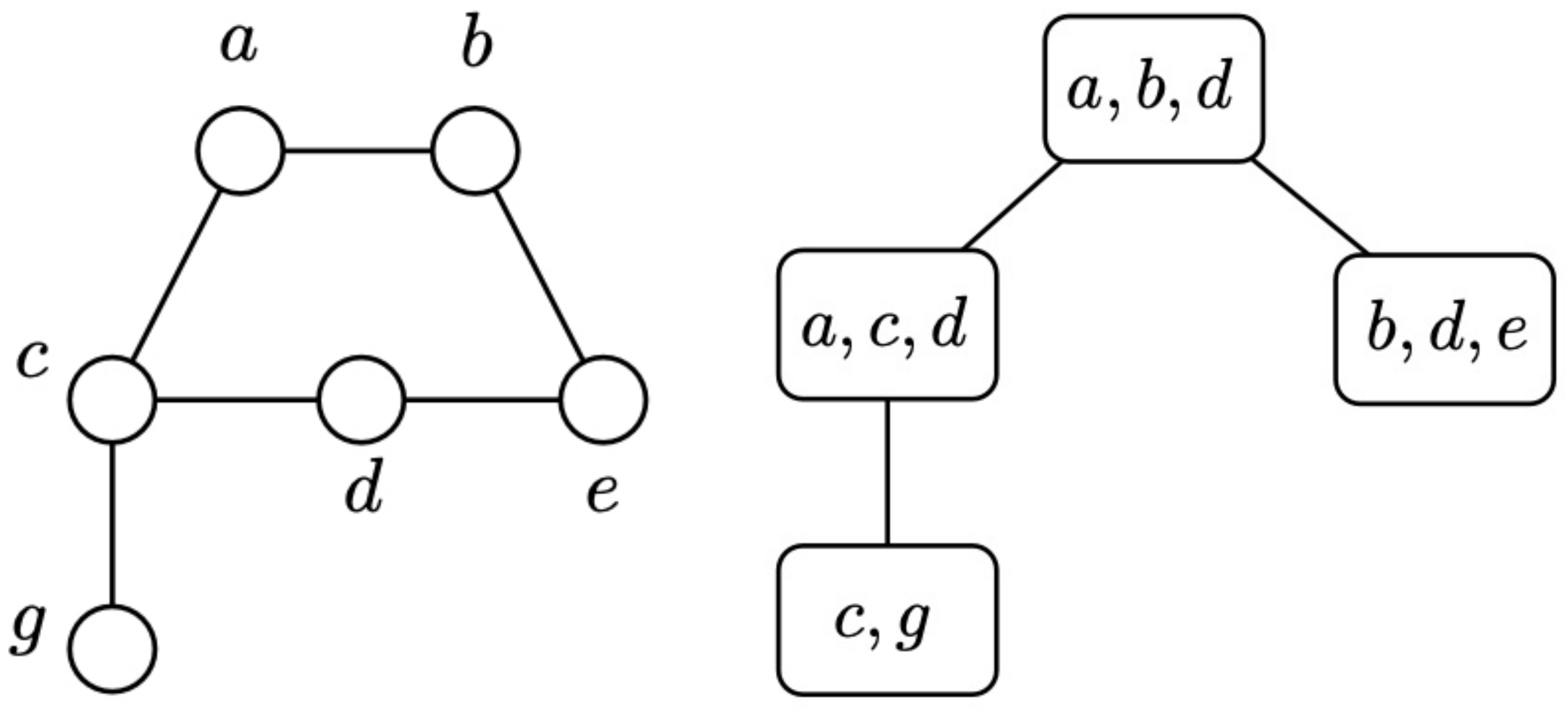
Property $\mathrm{tw} \leq k$: certifying a 3-approximation

certificates & messages of size $O(k^2 \log^2 n)$

Certificate for vertex v:

1. $d(v)$ the depth of its topmost appearance in the decomposition tree,
2. $\mathscr{B}(v) = (B_d(v), B_{d-1}(v), \ldots, B_1(v))$, the bags from $B_d(v)$ to the root
3. ... plus auxiliary messages to check that all vertices of $F(v) = B_d(v) \setminus B_{d-1}(v)$ got the same certificate

The last item uses a spanning tree of $V_{B_d} \setminus B_{d-1}$; congestion $O(\log n)$.



$\mathscr{B}(a) = \mathscr{B}(b) = \mathscr{B}(d) = (\{a, b, d\})$

$\mathscr{B}(c) = (\{a, c, d\}, \{a, b, d\})$

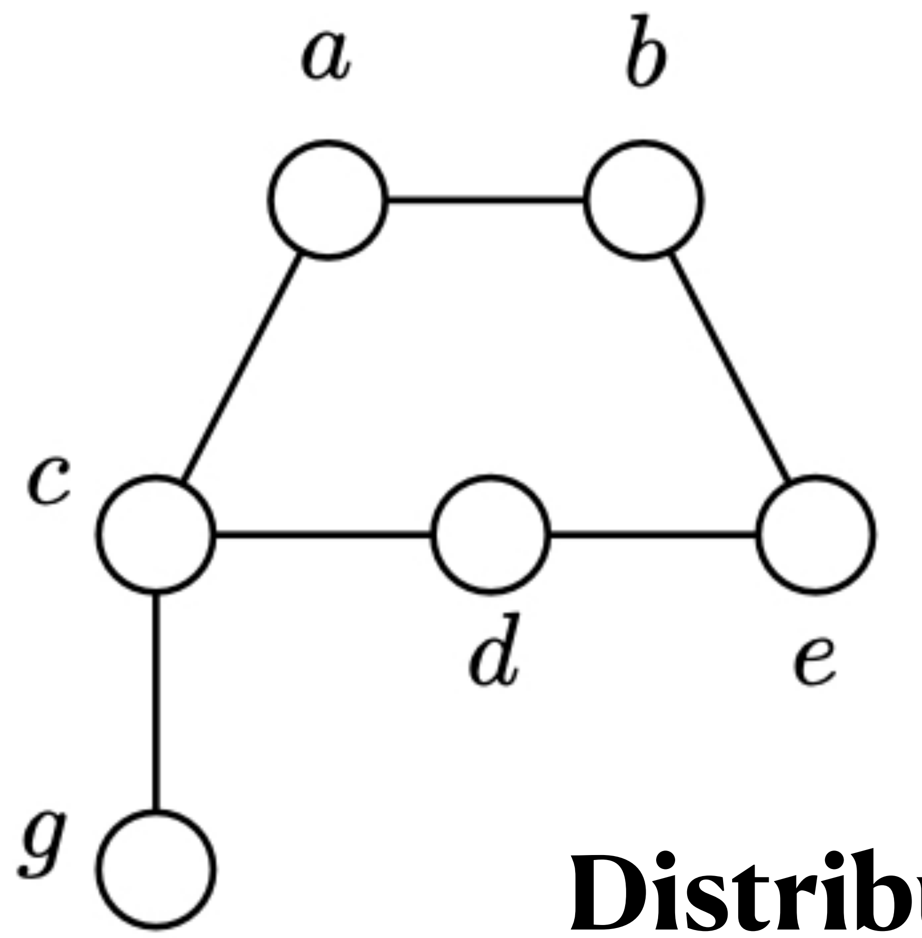
$\mathscr{B}(g) = (\{c, g\}, \{a, c, d\}, \{a, b, d\})$

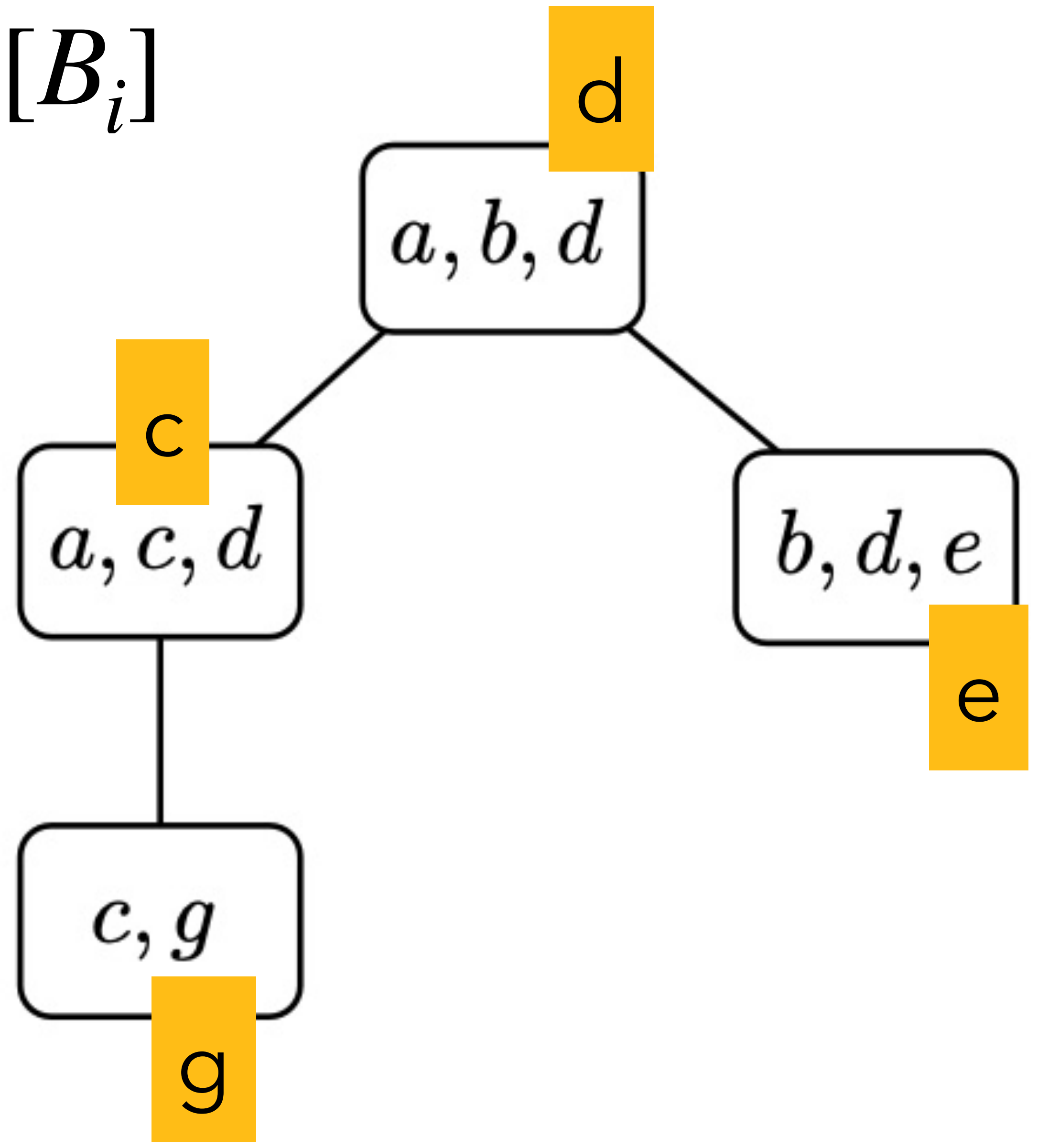
Distributed certification of $\text{tw} \leq k + \text{MSO}$

wishful thinking…

Prover certificates

- A 3-approximation for $\text{tw} \leq k$
- Bag B_i: choose a leader $v \in B_i \backslash B_{p(i)}$
- send to v the homomorphism class of $G[V_i]$
- … and graph $G[B_i]$



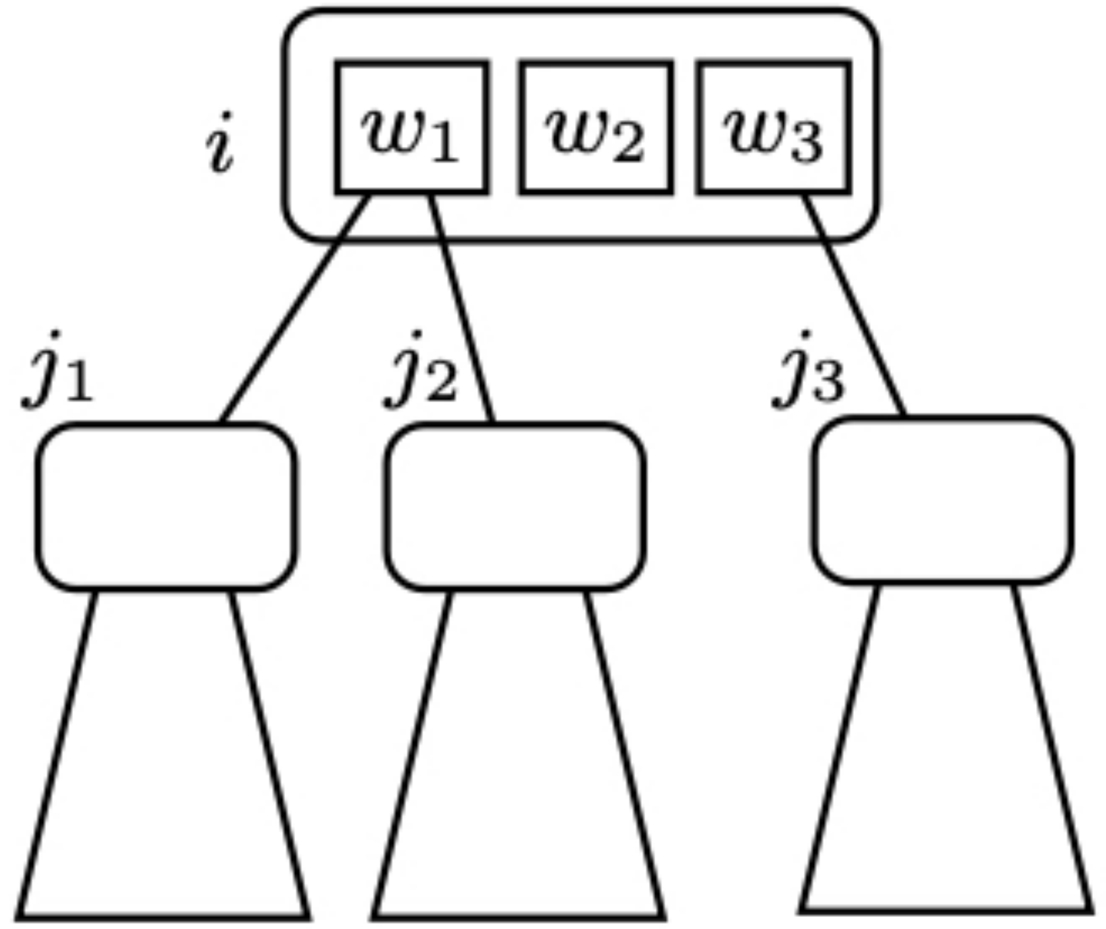


Distributed verifier

- the leader of bag i retrieves the certificates from its children bags $j_1, \ldots, j_p$
- *leader*(i) knows how $G[V_i]$ was obtained by glueing graphs $G[V_j]$ and the bag $G[B_i]$
- … so it checks that the homomorphism classes are coherent with the glueing

Distributed certification of $\mathrm{tw} \leq k$ + MSO

- **not that straightforward**, the leader of bag i may not see its children bags $j_1, \ldots, j_p$
- for each each child j of i we choose an "exit vertex" in $G[V_j \backslash B_i]$ adjacent to some node $w \in B_i \backslash B_{p(i)}$
- that w is responsible for several children nodes
- w gets the homomorphism class of $G^+[w]$ obtained by glueing $G[B_i]$ and all $G[V_j]$ for children j attached to w
- w is in charge of checking the consistency between $h(G^+[w])$ and all corresponding classes $h(G[V_j])$
- and $leader(i)$ ends the job.



w_1 is in charge of children j_1, j_2
w_3 is in charge of j_3

Conclusion

Pisco Sour

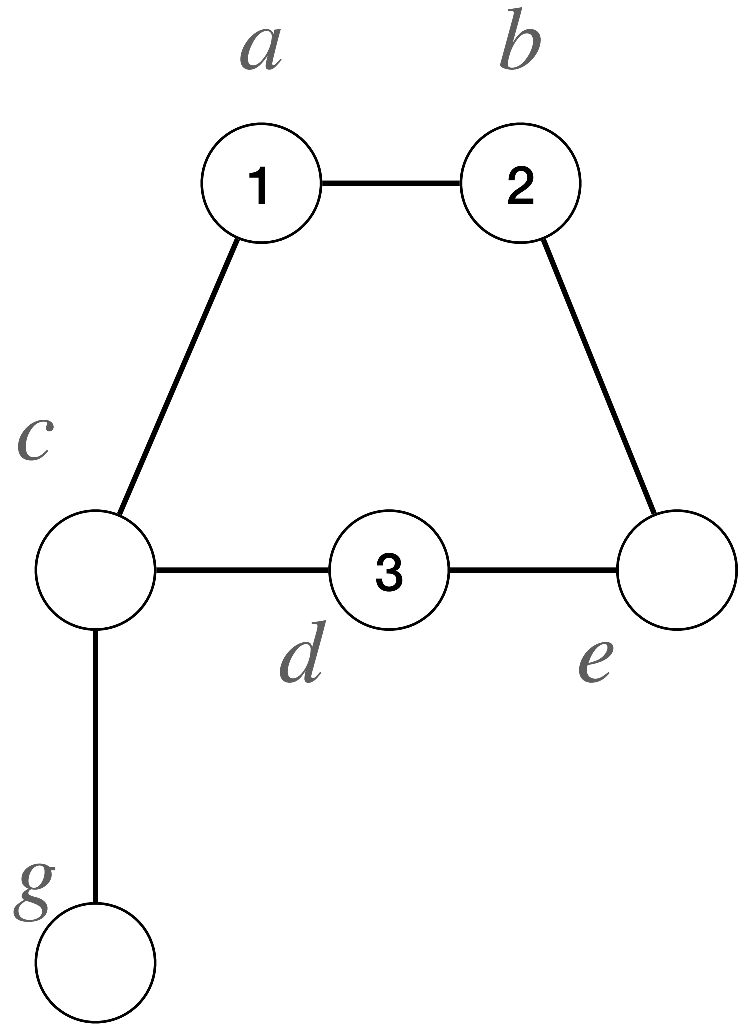
- Distributed certification for “tw $\leq k$ + MSO property”
- Deterministic, one round, uses $O(\log^2 n)$ bits
- Extends to optimisation problems, e.g., “tw $\leq k$ + MaxIndependentSet”
- Hides large constants in k, even for “tw $\leq k$”

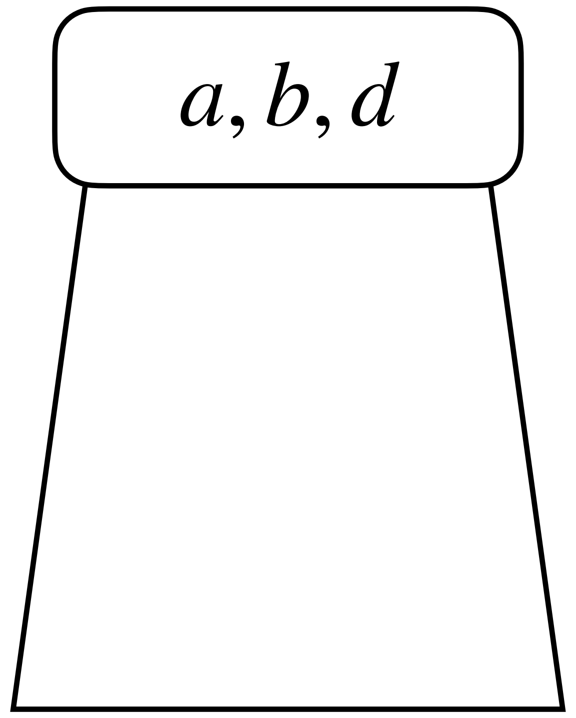
- What about $O(\log n)$ certificates — as for tree-depth, [Bousquet, Feuilloley, Pierron ’21]?
- Distributed certification? Done for planarity/bounded genus, chordal graphs…
- Distributed algorithmic meta-theorems?

More on MSO on bounded tw: regular properties

Courcelle's theorem in the version of [Borie, Parker, Tovey '92]

Informally: (1) regular properties are defined to have a dynamic programming scheme with tables of constant size, and (2) MSO properties are regular.

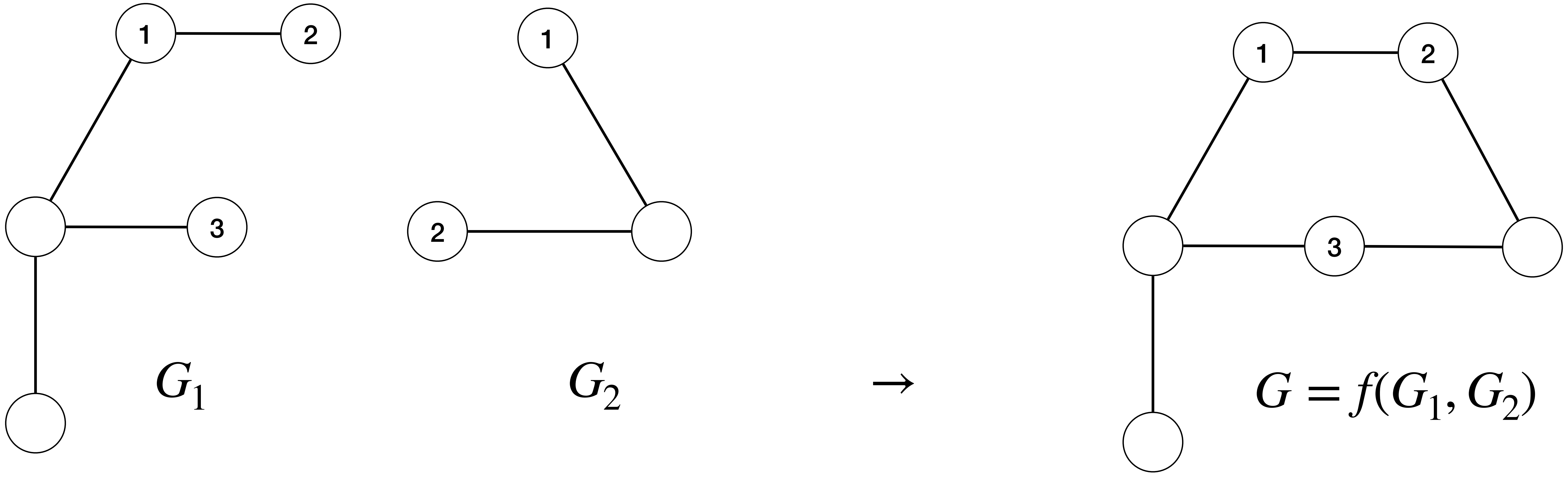




Graphs of tw $\leq k$ defined by a graph grammar on $k+1$-terminal graphs, i.e., having $k+1$ distinguished, numbered vertices (the root bag)

- a binary "glueing" operation
- a unary "forget" operation

Glueing operation for $k+1$-terminal graphs

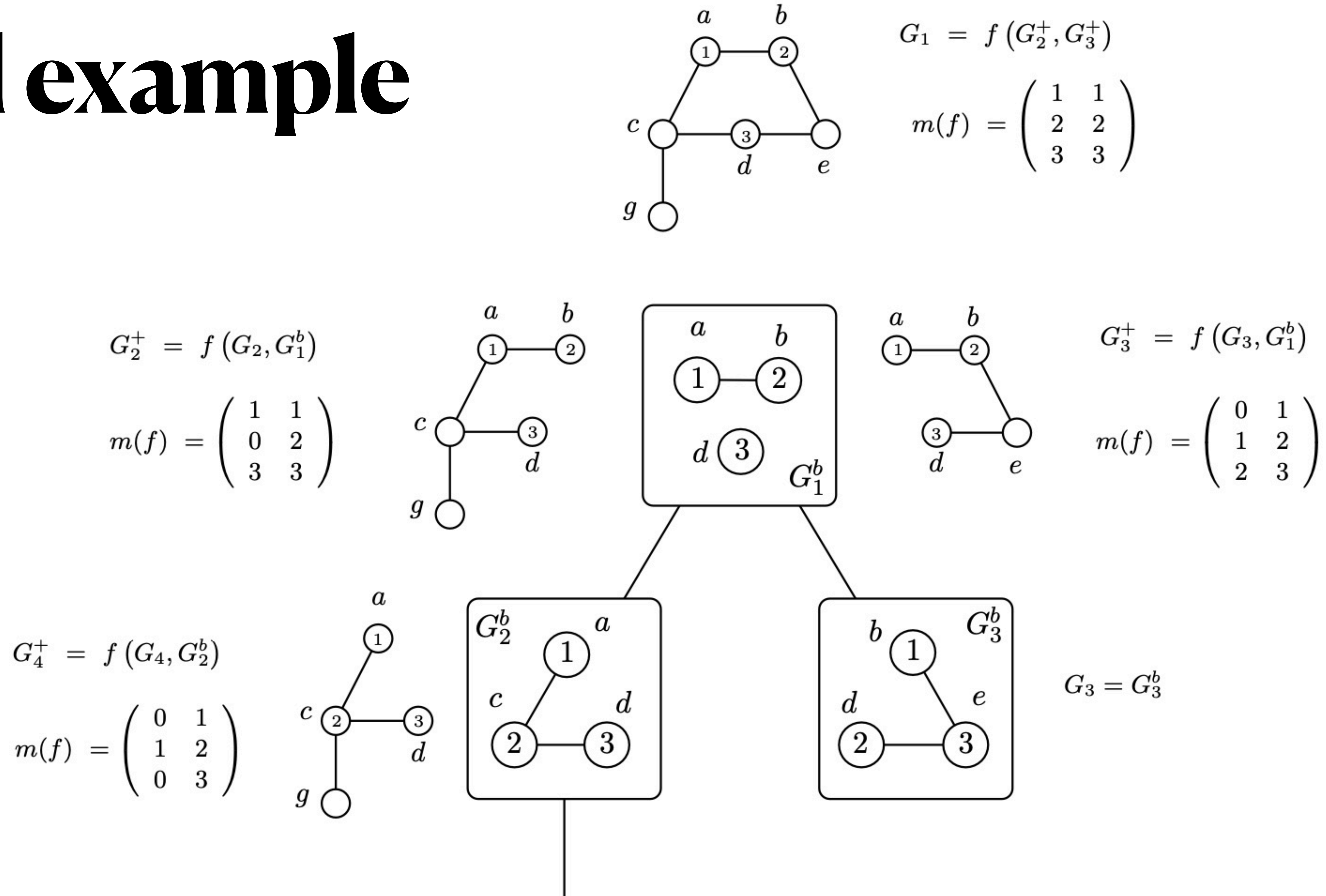


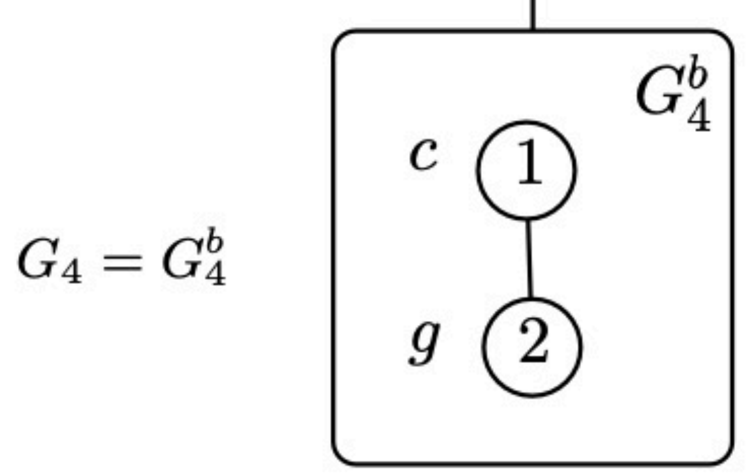
f described by matrix $m_f = \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 2 \end{bmatrix}$ with two columns, $k+1$ rows;

$m_f(i, c)$ is the terminal of G_c mapped on terminal number i of G

- a similar unary operation with only one column
- base graphs: only terminals (at most $k+1$ vertices)

Full example





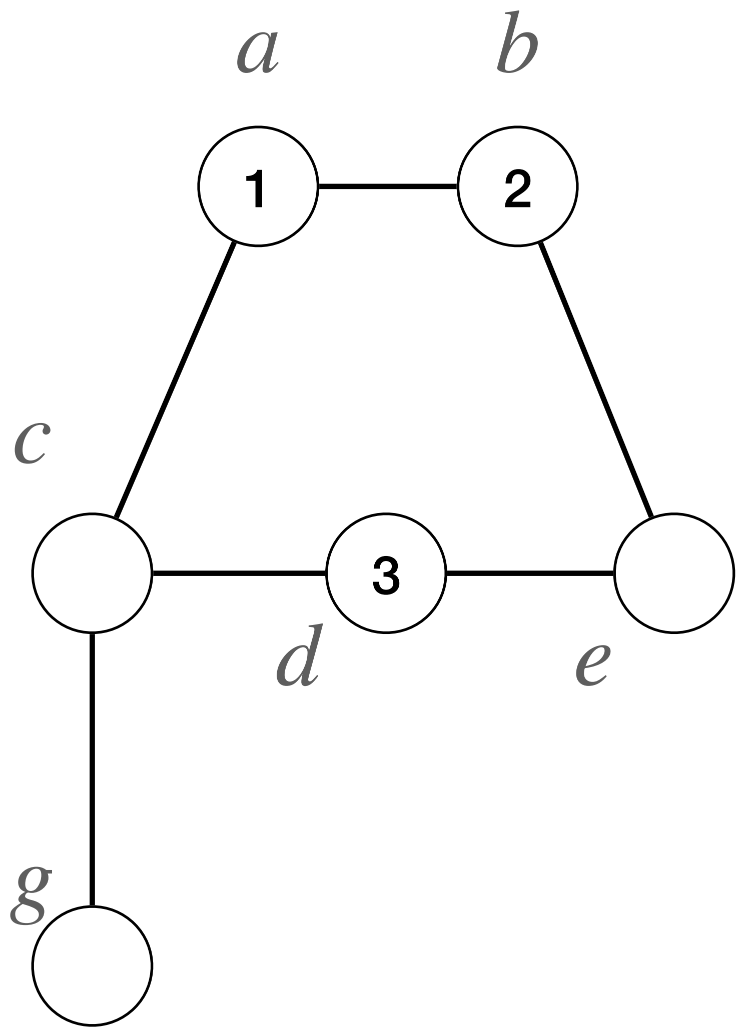
Graphs of $tw \leq k$ are exactly the $k + 1$ terminal recursive graphs. See e.g. [Bodlaender '98] — arboretum.

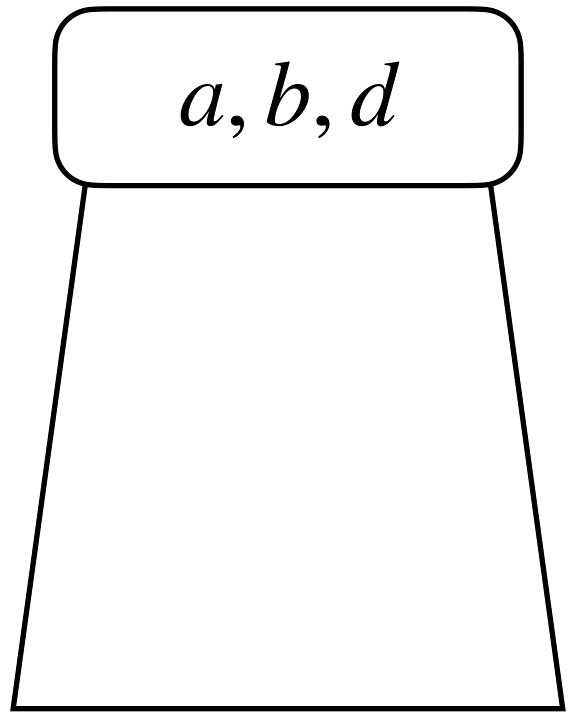
Regular properties on terminal recursive graphs

Courcelle's theorem in the version of [Borie, Parker, Tovey '92]

Property $\mathscr{P}$ is **regular** if we can associate homomorphism classes to $k+1$-terminal recursive graphs

$h : G = (V, E, T) \to \mathscr{C}_{k+1}$ such that:



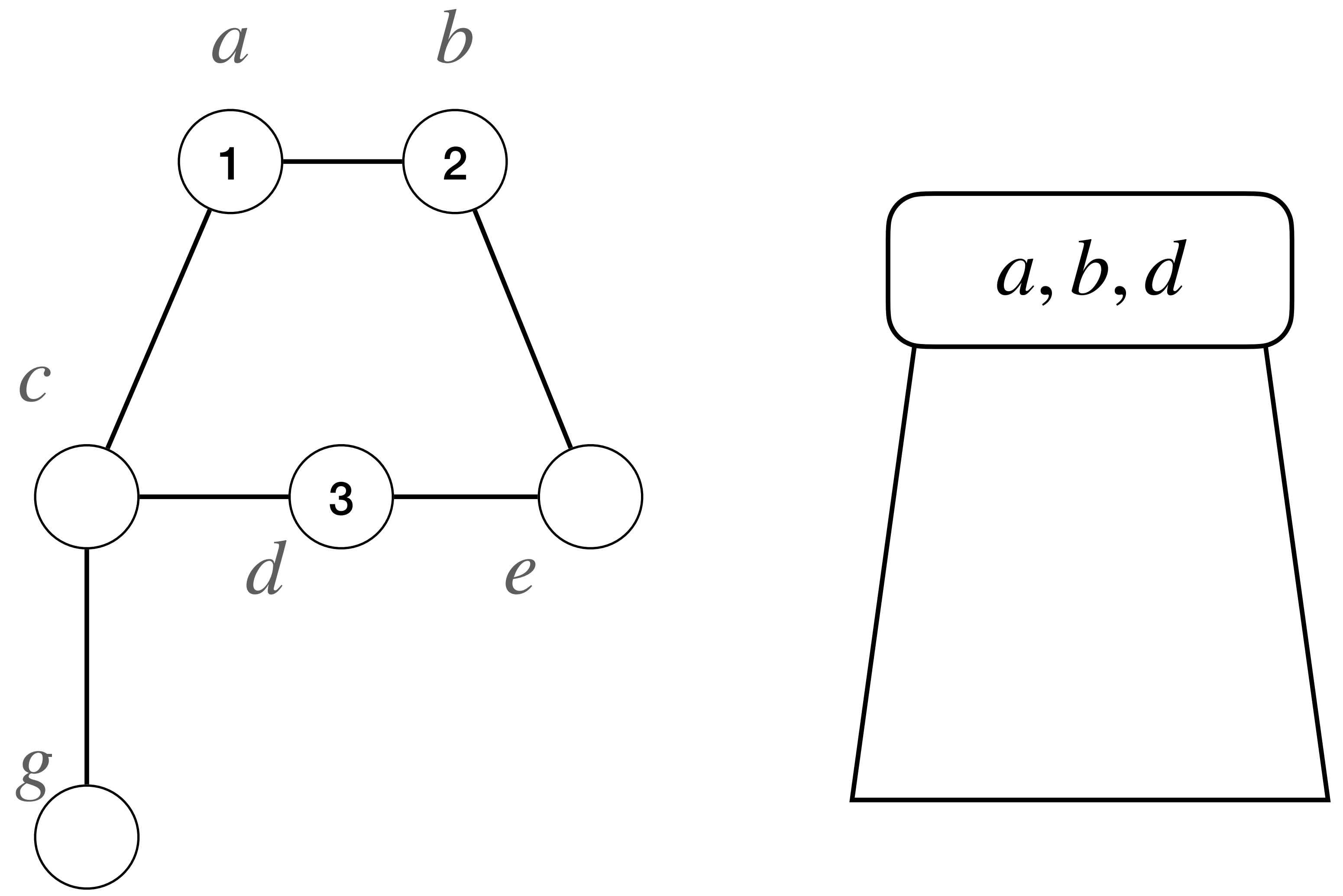


- $h(G_1) = h(G_2) \Rightarrow \mathscr{P}(G_1) = \mathscr{P}(G_2)$
- if $G = f(G_1, G_2)$ then $h(G)$ only depends on $h(G_1), h(G_2)$ and m_f
- same for unary operations f

Example: $\mathscr{P}$ = 3Colorability. Take as $h(G)$ all 3-partitions (R, G, B) of $\{1,\ldots,k+1\}$ such that $T \cap R, T \cap G, T \cap B$ can be extended into a colouring of G.

MSO properties are regular

Theorem [Borie, Parker, Tovey '92]. MSO properties are regular. Given formula φ and k, one can compute homomorphism classes for property $\mathscr{P}_\varphi$ for base graphs, and update tables for composition operations f.



- $h(G_1) = h(G_2) \Rightarrow \mathscr{P}(G_1) = \mathscr{P}(G_2)$
- if $G = f(G_1, G_2)$ then $h(G)$ only depends on $h(G_1)$, $h(G_2)$ and m_f
- same for unary operations f

Bottom-up dynamic programming to compute the homomorphism class of $G[V_i]$. Decision at the root. Also works for properties on graphs and vertex/edge subsets.