The Parameterized Complexity of SAT

Stefan Szeider

TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

ALGORITHMS AND COMPLEXITY GROUP

PCCR Workshop @ FLoC 2022

Propositional satisfiability (SAT)

- SAT (or CNF-SAT) is the following problem:
  - Instance: a propositional formula in conjunctive normal form
  - Question: is the formula satisfiable?

$F = \{C_1, \ldots, C_5\}$

$C_1 = \{u, \bar{v}, y\}, C_2 = \{\bar{u}, z, \bar{y}\}, C_3 = \{v, \overline{w}\}, C_4 = \{w, \bar{x}\}, C_5 = \{x, y, \bar{z}\}$

satisfied by setting $y = 1, u = 0, v = 1, x = 0$

define literal, clause, occurrence, truth assignment

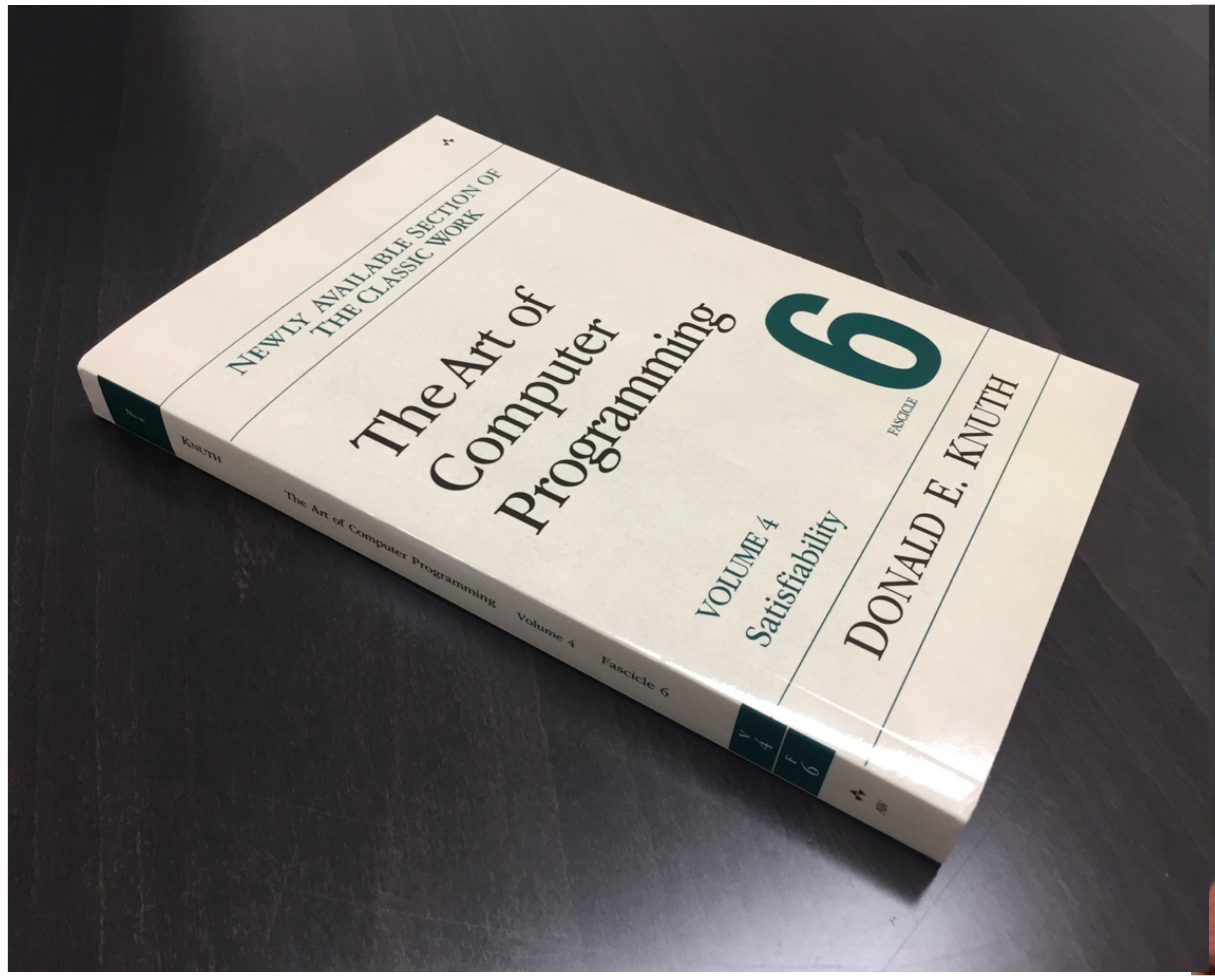
just a simple problem…

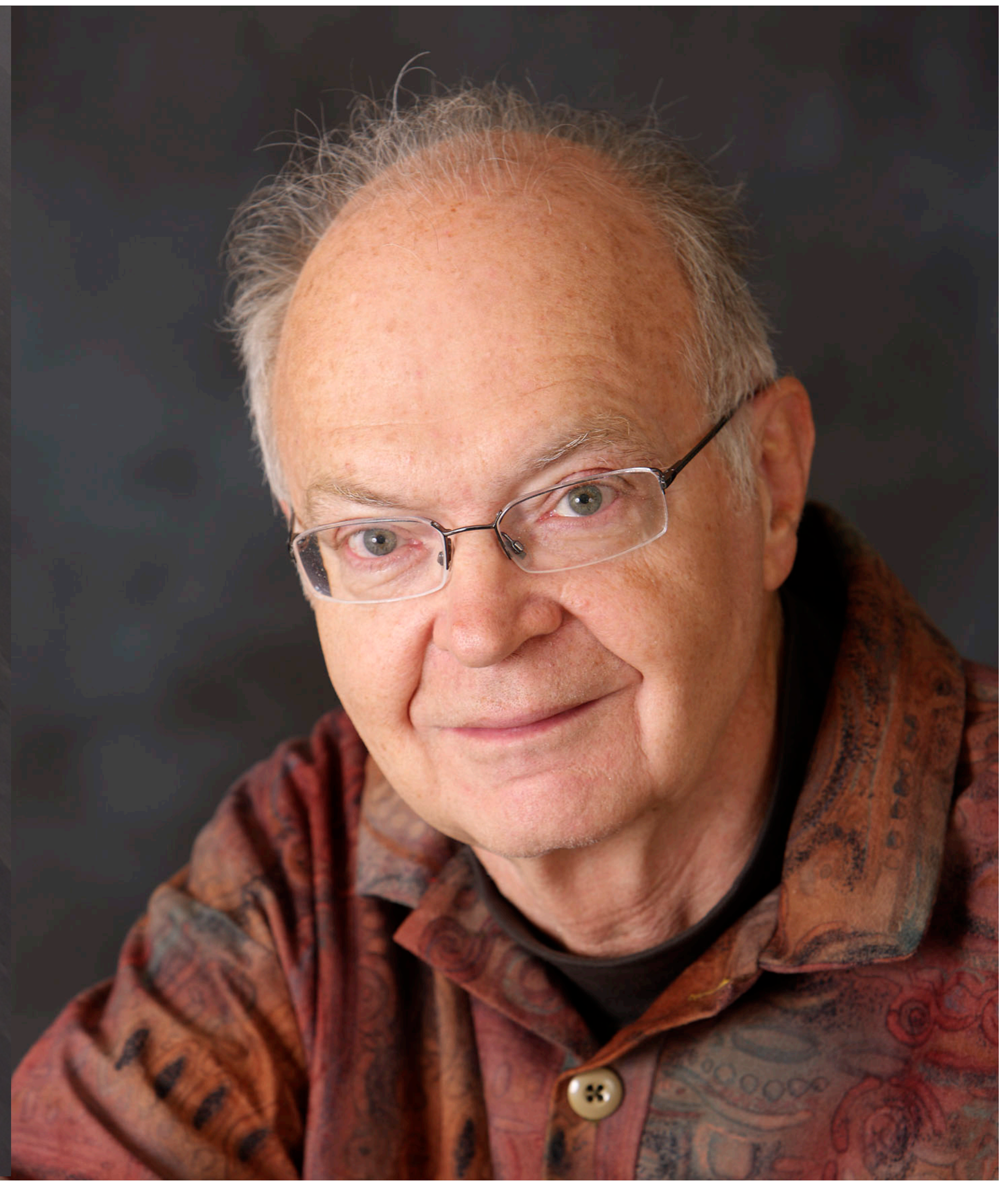
- Donald E Knuth wrote a 300+ page chapter on SAT in his TAOCP.

"The SAT problem is evidently a killer app, because it is key to the solution of so many other problems."

FLoC 2022

Knuth: Wed, 9:00





The silent (R)evolution of SAT

[Fichte, Hecher, Leberre, Sz. CACM 2022, to appear]

- The Pre-Revolution (< 2000)
  - DPLL 1960s, Variable selection heuristics 1990s, DIMACS SAT Challenges
- The Revolution (≈ 2000)
  - Solvers GRASP, Chaff, Conflict-driven Clause learning (CDCL), Watched Literal data structure, etc
- The Evolution (> 2000)
  - Efficient encodings, incremental solving, in/preprocessing, parallelization, proofs, cube and conquer, open source

Time Leap Challenge [Fichte, Hecher, Sz. CP 2020]

new computer

old computer new

	Grasp (1996)	zChaff (2001)	siege_v3 (2003)	Glucose (2016)	CaDiCal (2019)	Maple (2019)
					Team SW	
old HW (1999)	73	48	37	106	98	77
		Team HW				
new HW (2019)	76	71	93	188	190	195

“Hidden Structure” in SAT instances

- SAT solvers routinely solve industrial instances with millions of clauses and variables (today’s solvers use the CDCL approach which is closely linked to the resolution proof system)
- For classical TCS approaches, SAT is hard

  (PPSZ: 1.364^{200} = 2 x age of universe in nanoseconds, (S)ETH)
- Theory-practice gap
- Common insight: real-world SAT-instances contain some kind of “hidden structure” which is implicitly utilized by solvers
- Can we utilize the structure also in theory?

Two Approaches

Correlation

Try to capture structure in a way that statistically correlates with CDCL-solving time
community structure, modularity, centrality, …
features for hardness prediction
[Ansótegui, Bonet, Giráldez-Cru, Levy, Simon JAIR'19]
[Li, Chung, Mukherjee, Vinyals, Fleming, Kolokolova, Ganesh SAT'21]

Causation

Try to capture structure in a way that provides worst-case performance guarantees for SAT algorithms
decomposability, backdoors, ….

Community Structure in Industrial SAT Instances

- **modularity** of G is $\max q(\mathcal{C})$ over all partitions $\mathcal{C}$ of V [Newman, Girvan 2004]

$$q(\mathcal{C}) = \sum_{C \in \mathcal{C}} \left[\frac{|E(C)|}{m} - \left(\frac{\sum_{v \in C} deg(v)}{2m} \right)^2 \right]$$

- In general, industrial formulas have a exceptionally high modularity, greater than 0.8 in many cases. Notice that in other kind of networks, values greater than 0.7 are rare [Ansotegui et al JAIR 2019]

Algorithmic use of modularity?

- It is easy to construct a class of formulas of arbitrarily large modularity for which SAT decision remains NP-hard.
- [Ganian, Sz, AIJ 2021] (we'll come back to this a bit later …)

FPT-SAT

Parameterized Complexity

- For causal models, parameterized complexity provides and ideal framework
- We can develop different parameters that capture different properties of SAT instances
- Compare parameters by their generality

Almost 20 years ago!

On Fixed-Parameter Tractable Parameterizations of SAT

Stefan Szeider*

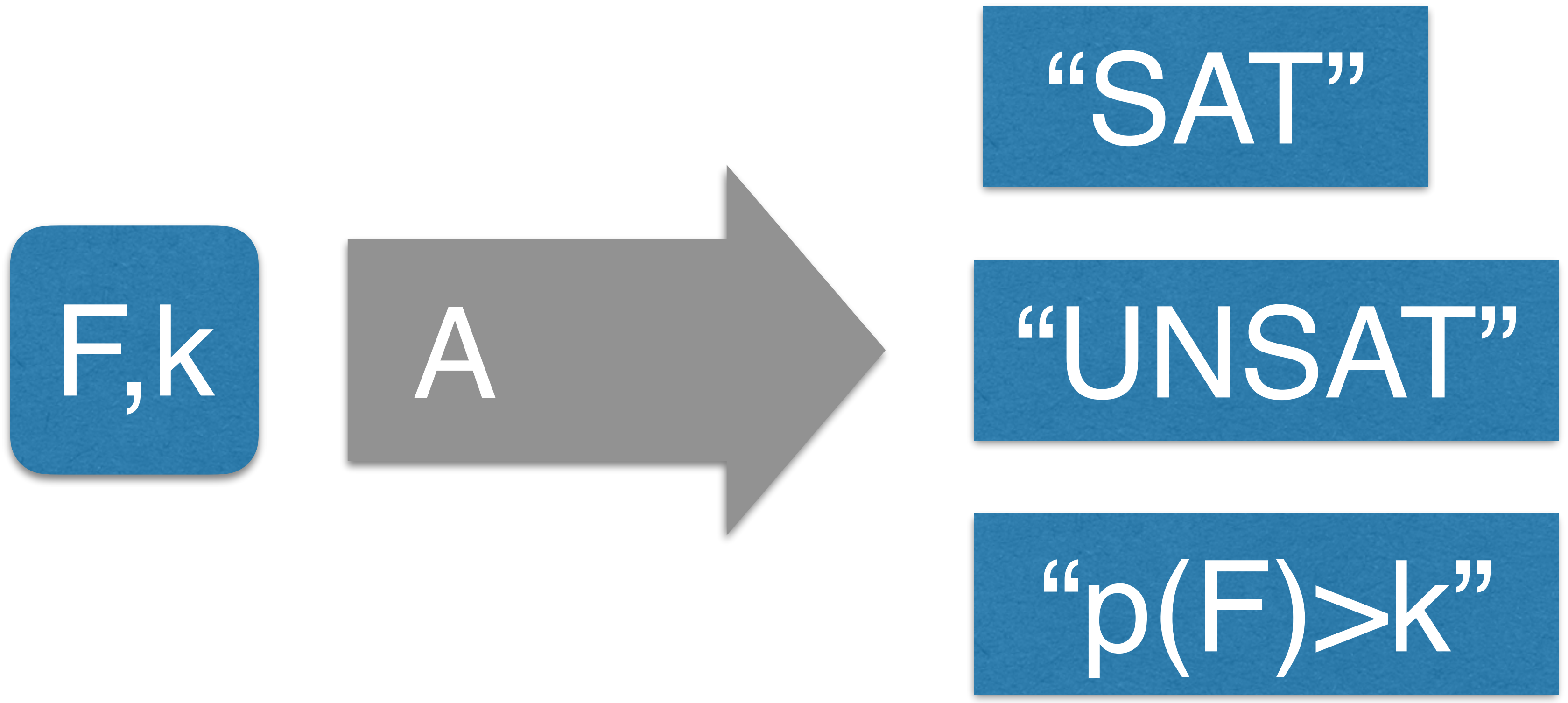
Department of Computer Science, University of Toronto,
M5S 3G4 Toronto, Ontario, Canada
szeider@cs.toronto.edu

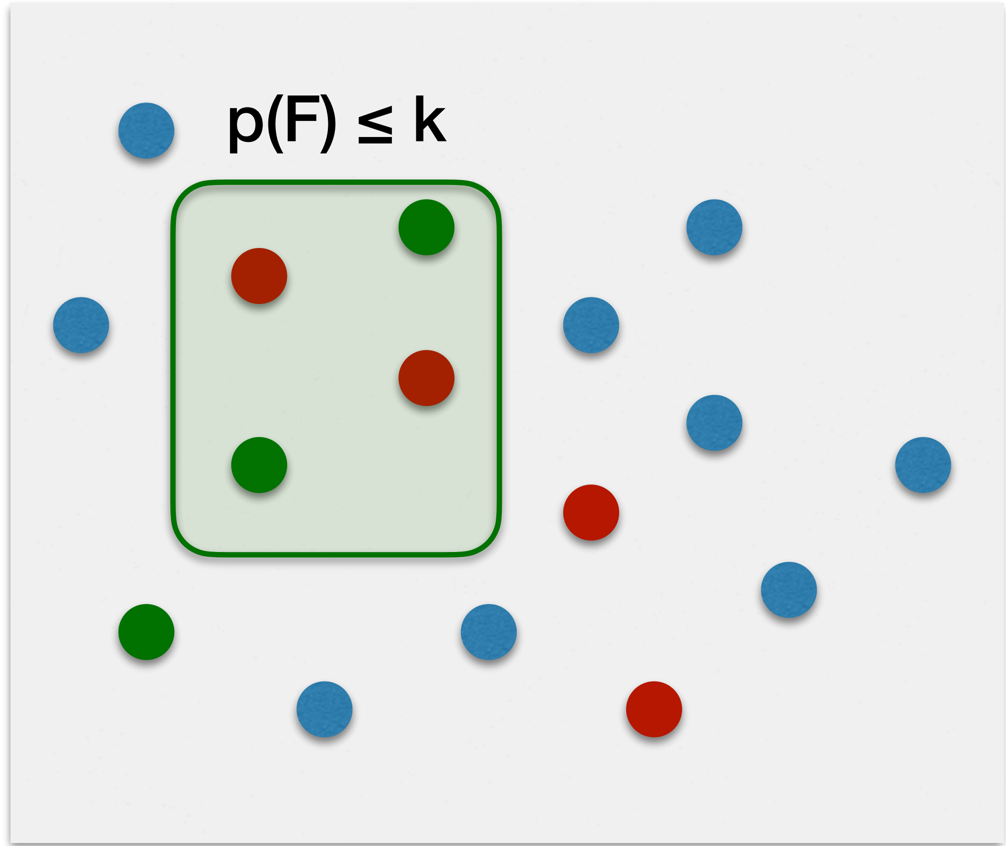
Abstract. We survey and compare parameterizations of the propositional satisfiability problem (SAT) in the framework of Parameterized Complexity (Downey and Fellows, 1999). In particular, we consider (a) parameters based on structural graph decompositions (tree-width, branch-width, and clique-width), (b) a parameter emerging from matching theory (maximum deficiency), and (c) a parameter defined by translating clause-sets into certain implicational formulas (falsum number).

E. Giunchiglia and A. Tacchella (Eds.): SAT 2003, LNCS 2919, pp. 188–202, 2004.

FPT-SAT

“permissive” or “robust” approach





two-phases approach

explicit
k-structure

“p(F) ≤ f(k)” FPT-approx

“p(F) > k”

Comparison of SAT-parameters

p dominates q if there is a function f such that for all F it holds that $p(F) \leq f(q(F))$

- **General research program:** come up with stronger and stronger parameters, and draw a detailed map of SAT-parameters and their mutual dominance

1) Graphical Structure
2) Syntactical Structure
3) Hybrid Models

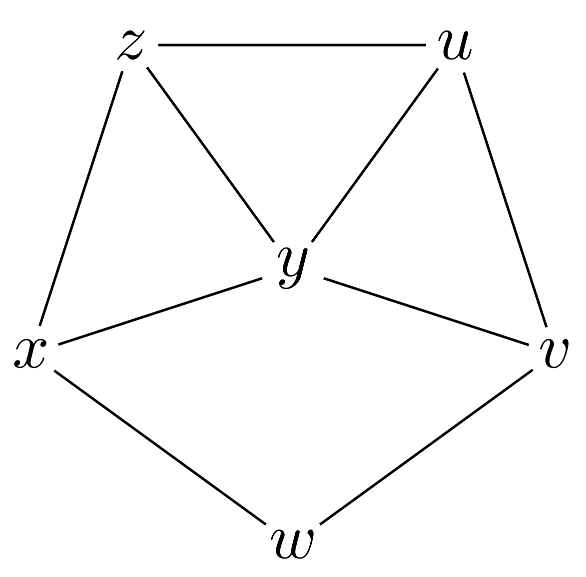
Graphical Structure

Common Graphs

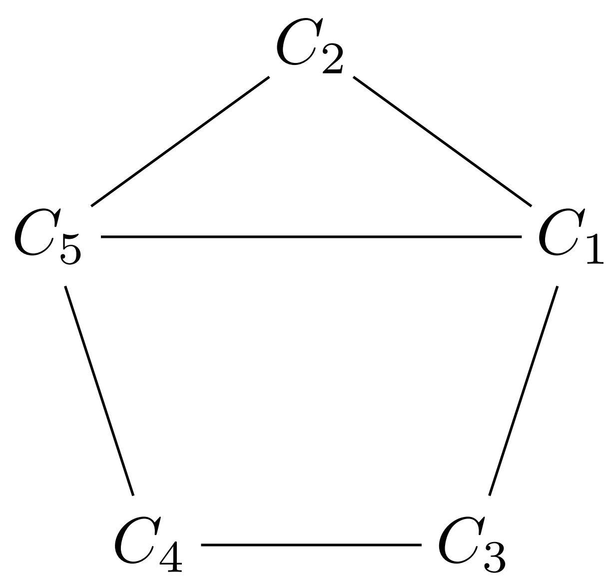
$F = \{C_1, \ldots, C_5\}$

$C_1 = \{u, \overline{v}, y\}, C_2 = \{\overline{u}, z, \overline{y}\}, C_3 = \{v, \overline{w}\}, C_4 = \{w, \overline{x}\}, C_5 = \{x, y, \overline{z}\}$

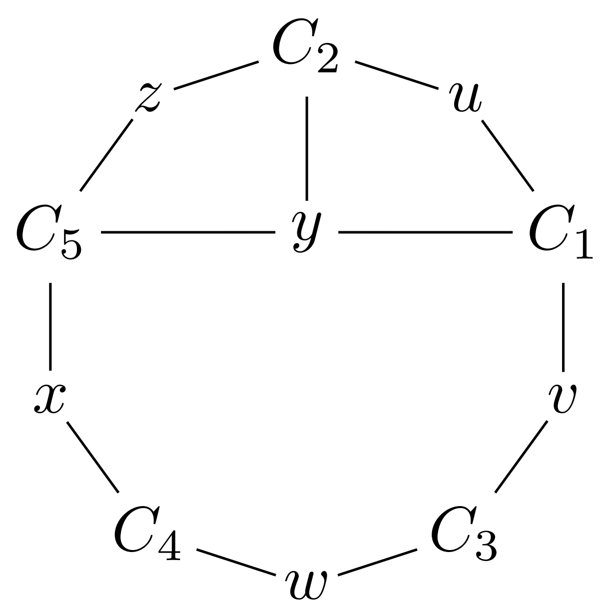
primal aka VIG



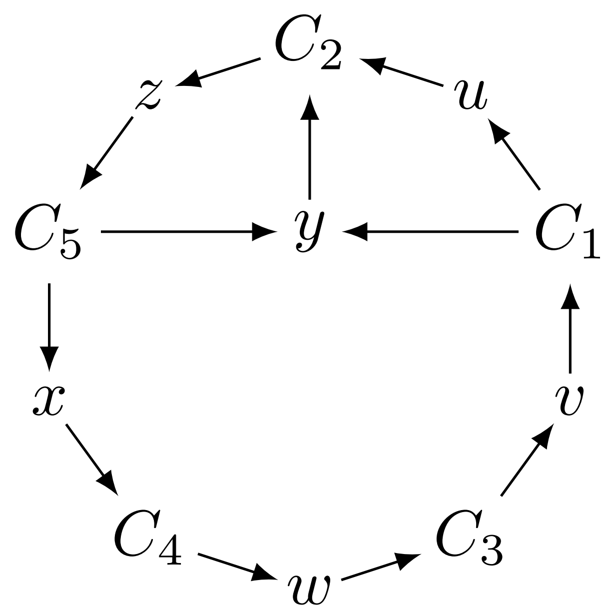
dual aka CIG

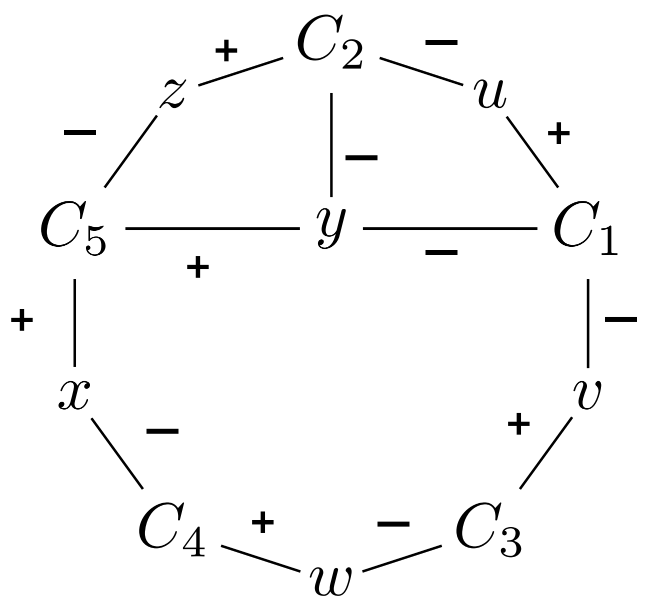


incidence aka CVIG

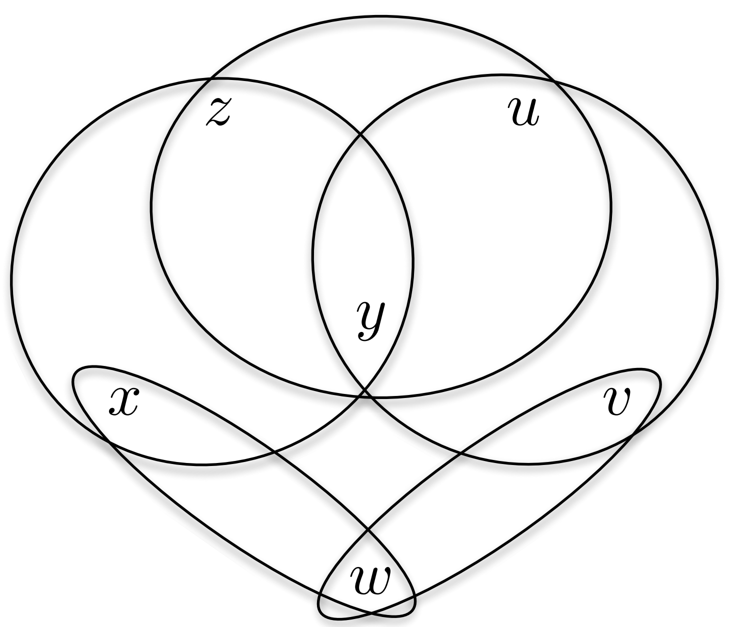


directed/signed inc

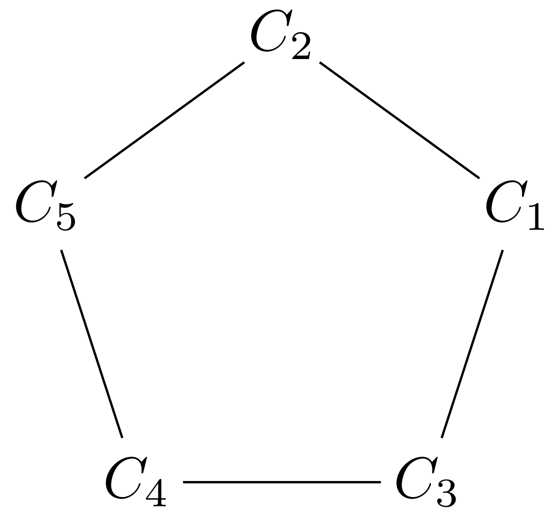




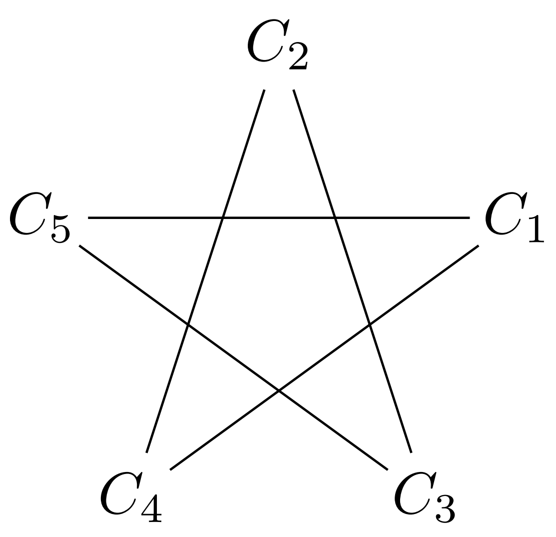
hypergraph



conflict

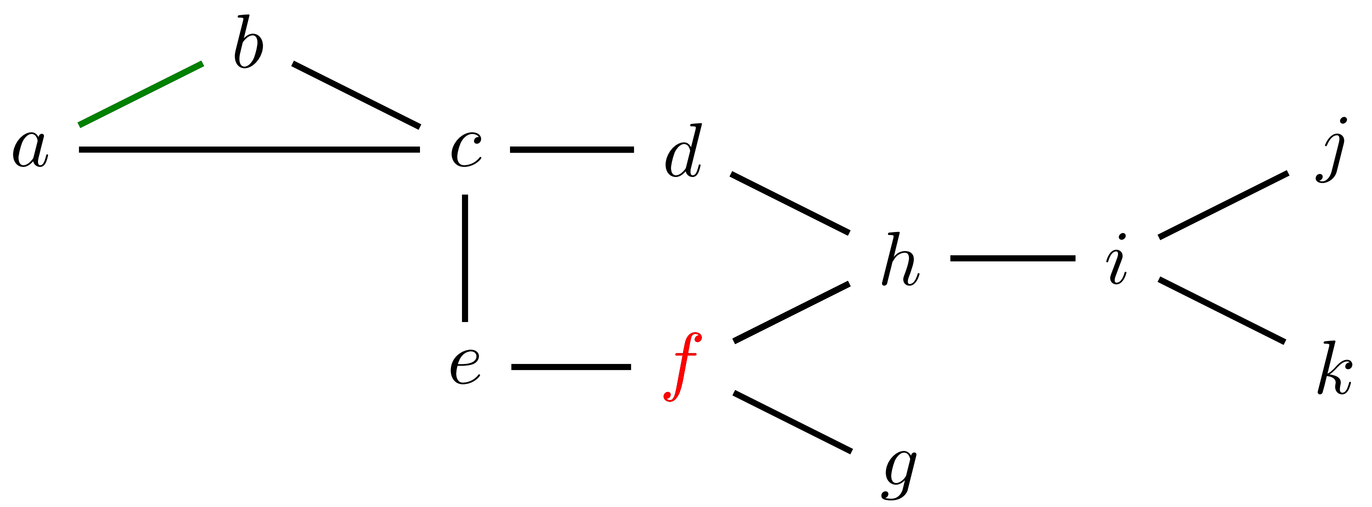


consensus

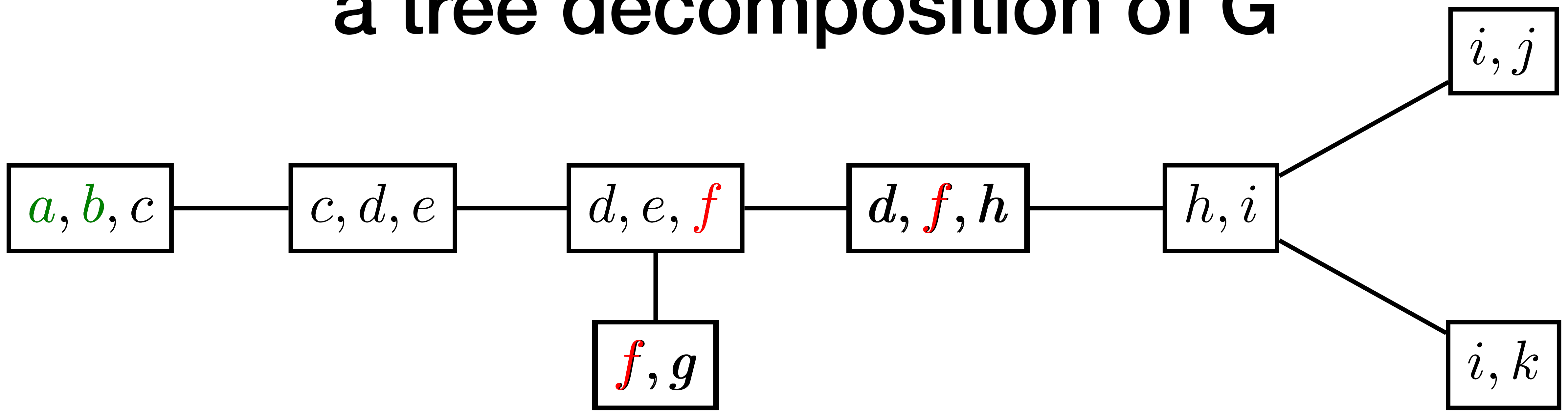


Graph Decompositions and Width Parms

a graph G



a tree decomposition of G

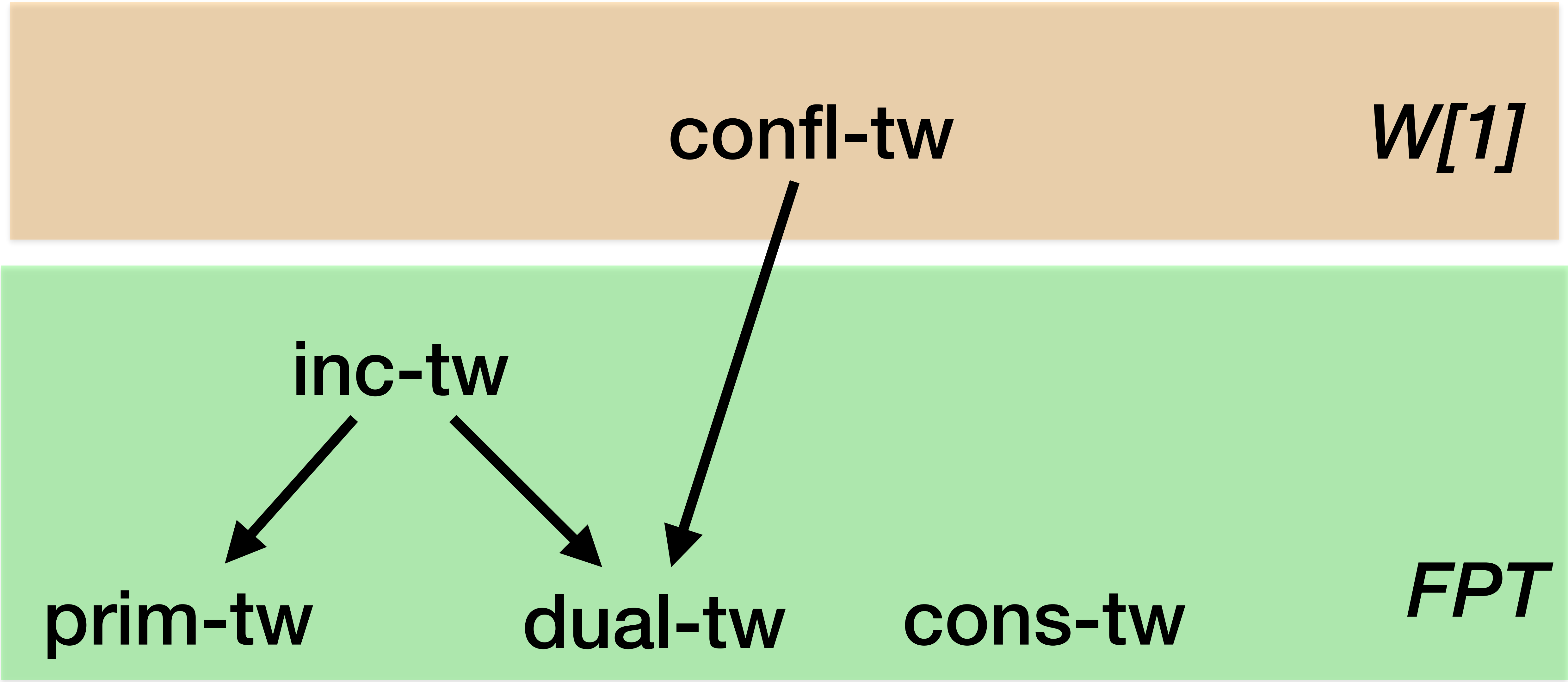


width = size of largest bag -1

- tw(G)=min width over all its tree decompositions
- checking tw(G) ≤ k is FPT

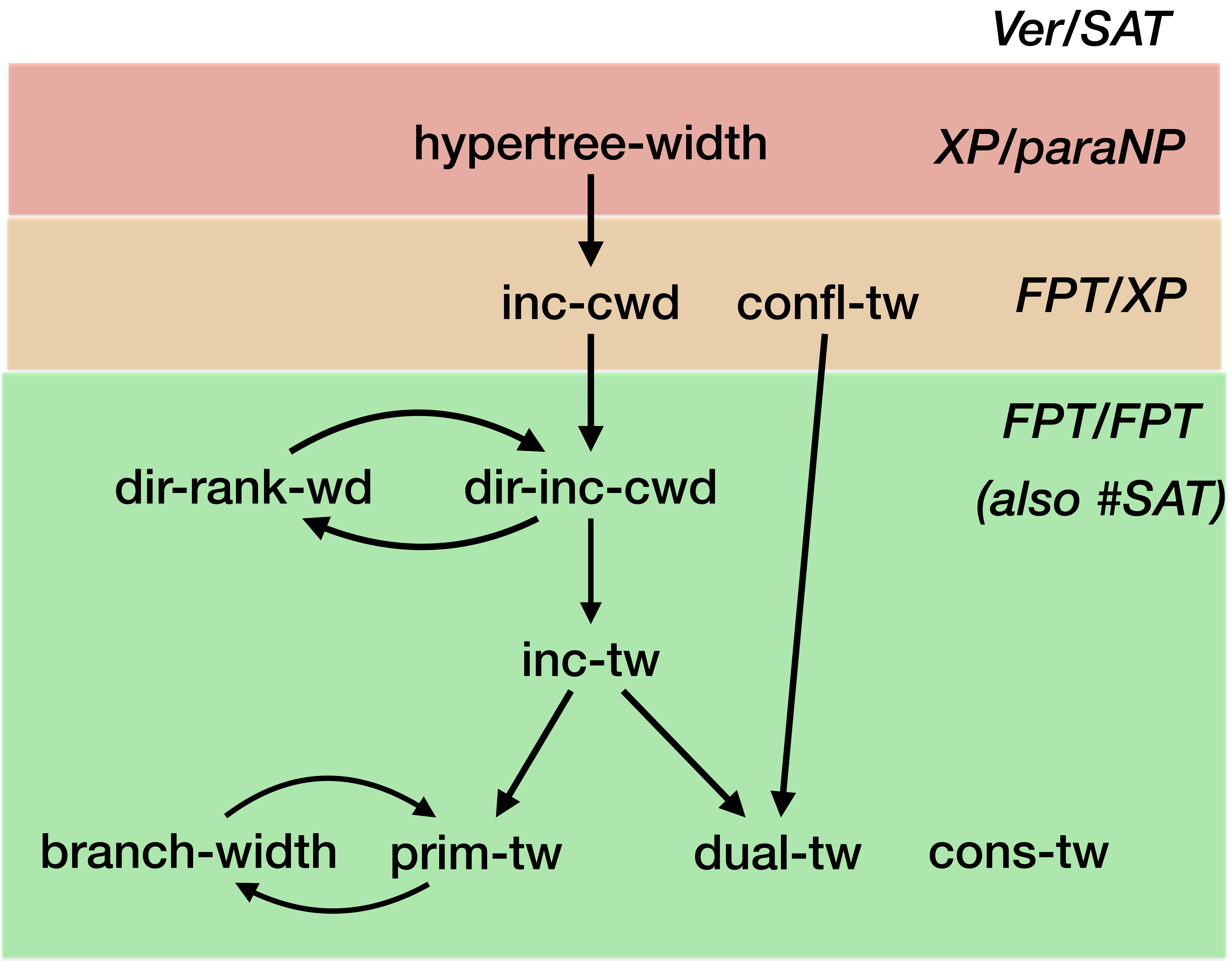
Treewidth of Formulas

- prim-tw(F), dual-tw(F), inc-tw(F), cons-tw(F), conf-tw(F)
- SAT is FPT parameterized by all the above parameters, except for confl-tw.



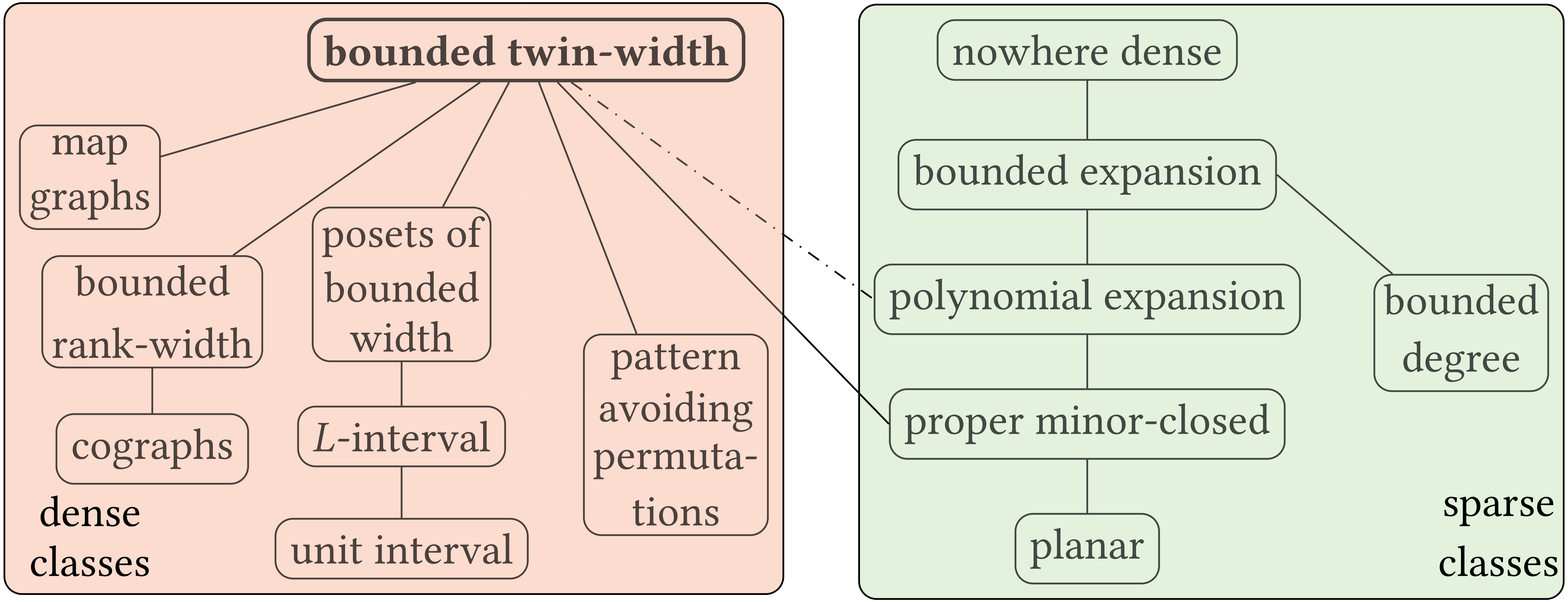
Improvement of $O^*(4^k) \Rightarrow O^*(2^k)$ for inc-tw using covering products [Slivovsky, Sz SAT 2020]

Width Parameter Zoo



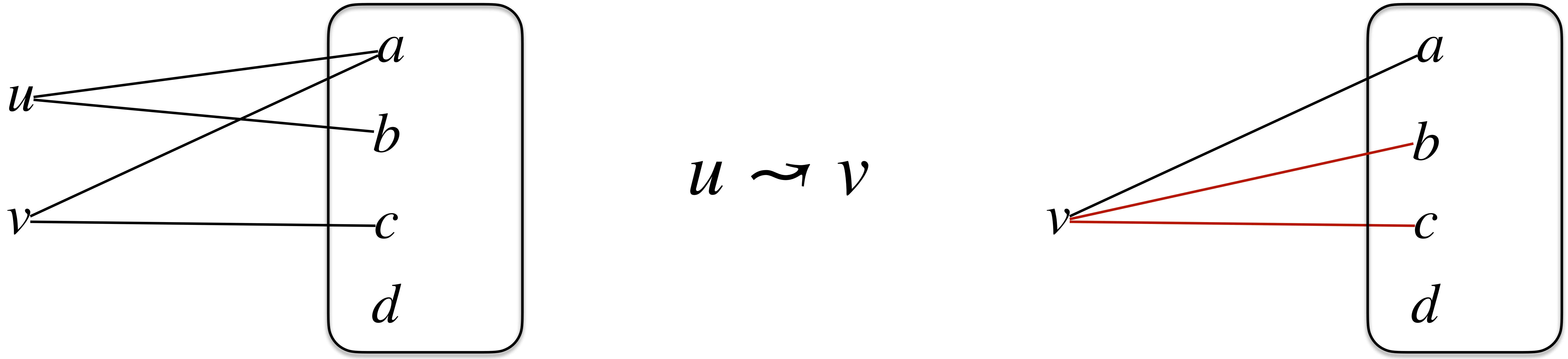
Twin-Width

Diagram of Graph Classes [Bonnet et al. JACM 2022]



Twin-Width of Graphs

- Reduce a given Graph to a single vertex by a sequence of contractions.
- Each contraction removes a vertex u by contracting it to one of the remaining vertices v. In symbols $u \rightsquigarrow v$.
- If u, v are twins, then the contraction is perfect.
- if u, v are not twins, record the error by coloring edges red.
- red edges remain red in subsequent steps

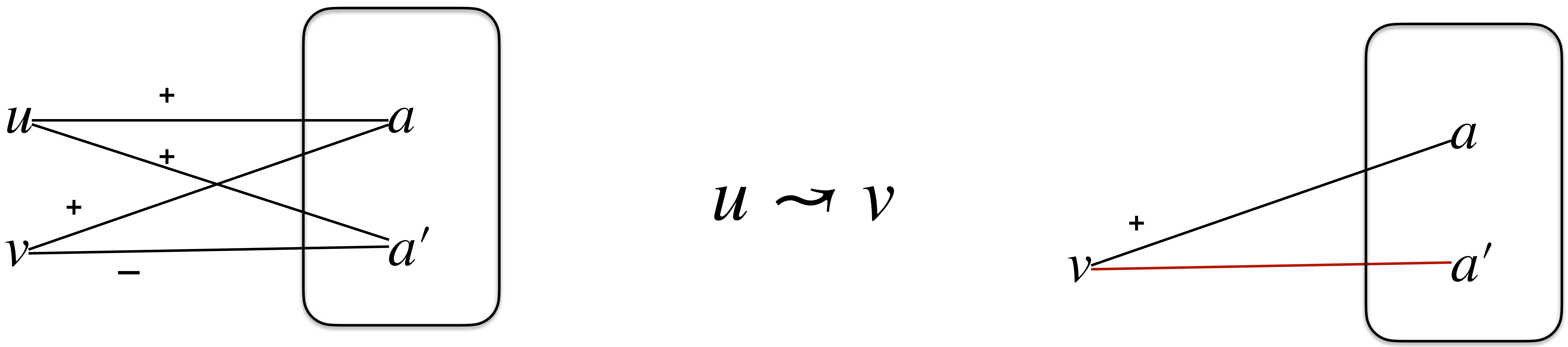


Twin-width of Graphs

- A **d-contraction sequence** of a graph contracts all vertices step-by-step to a single vertex graph, such that each intermediate graph has **red degree** at most d.
- $G = G_n \rightsquigarrow G_{n-1} \rightsquigarrow G_{n-2} \rightsquigarrow \cdots \rightsquigarrow G_1$
- The **twin-width** of a graph is the smallest d such that it admits a d-contraction sequence.

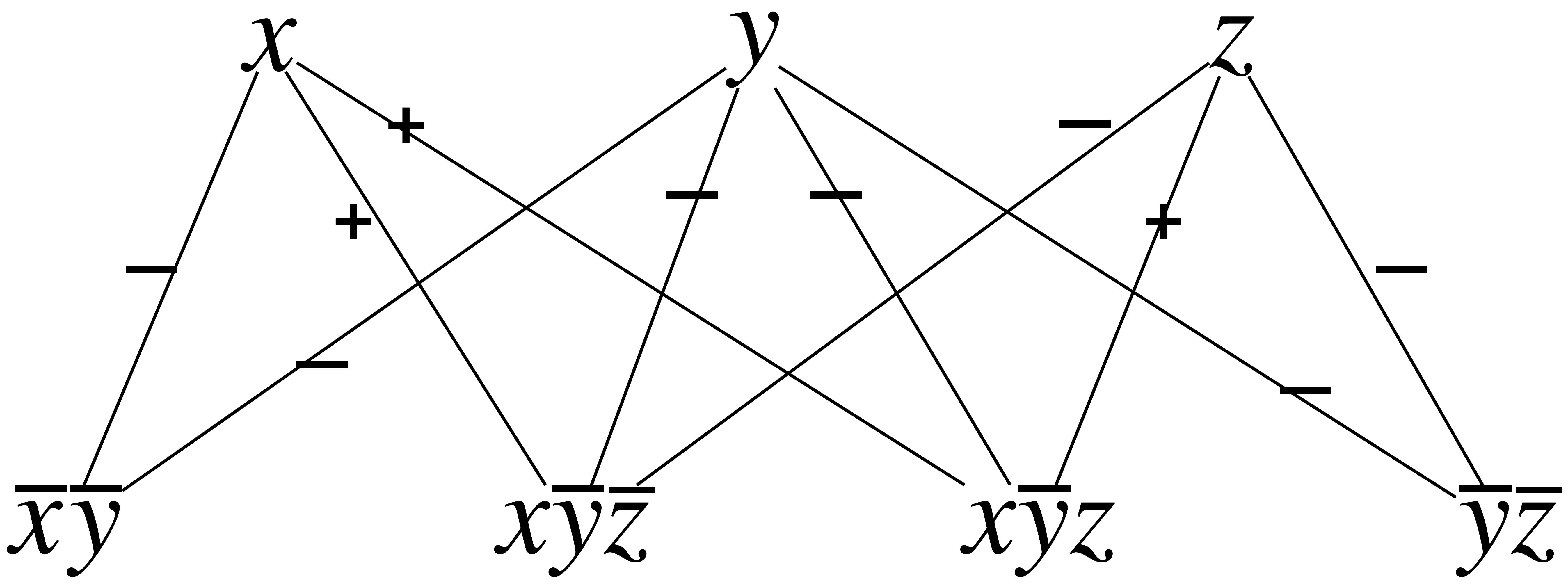
Signed twin-width

- The given graph G is **signed**, i.e., all edges are labeled + or -.
- A d-contraction sequence is defined as before, except that contracting black edges of different signs become red as well.

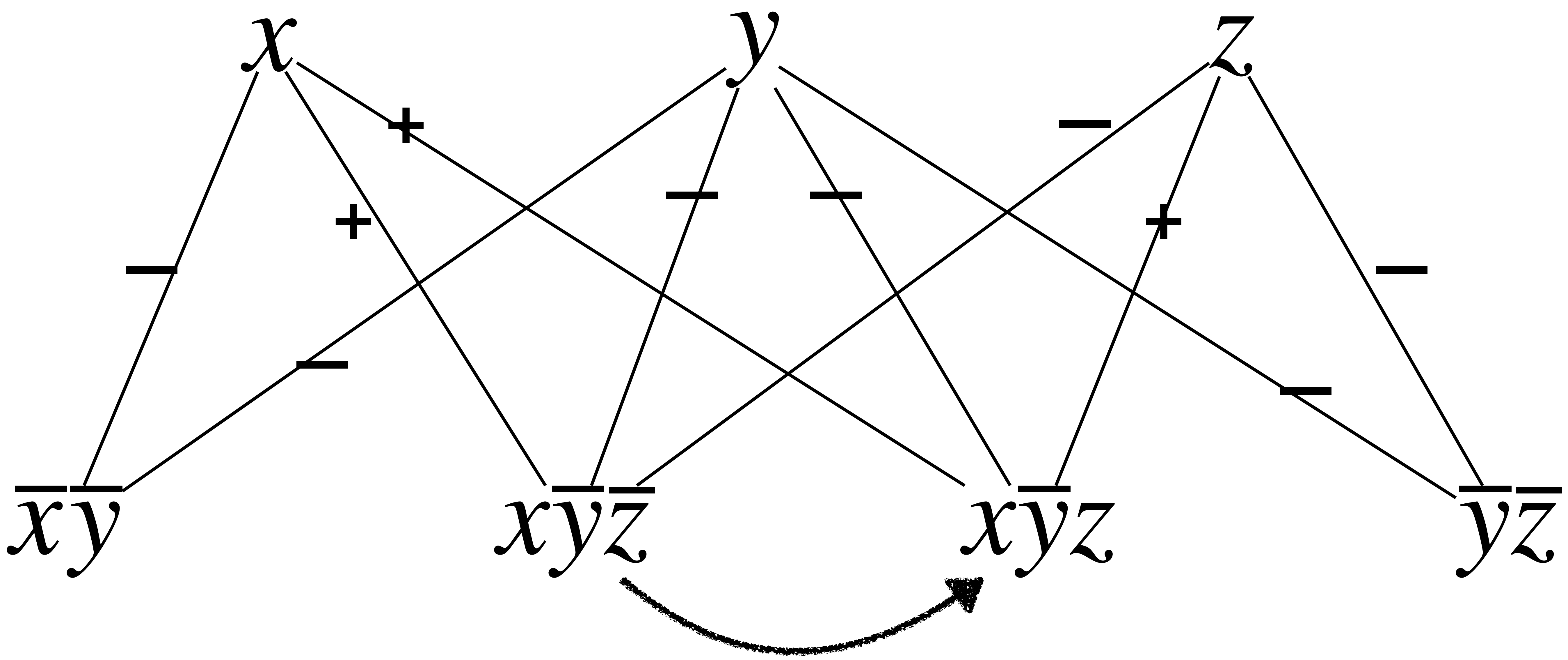


- For bipartite signed graphs, we can assume that we always contract vertices that belong to the same side of the partition. We can show that the tww does only change by a small constant factor if we implement this assumption.

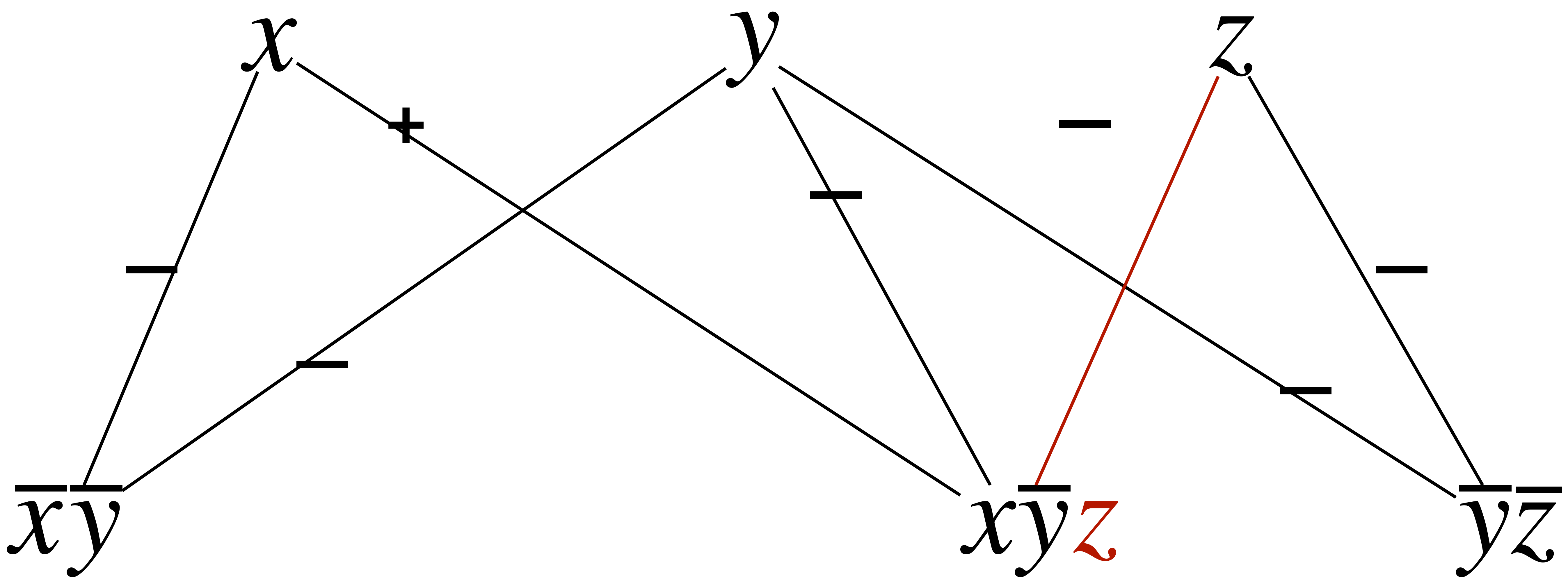
Example


x
y
z
+
+
–
–
–
–
–
+
–
–
$\overline{x}\overline{y}$
$x\overline{y}\overline{z}$
$x\overline{y}z$
$\overline{y}\overline{z}$

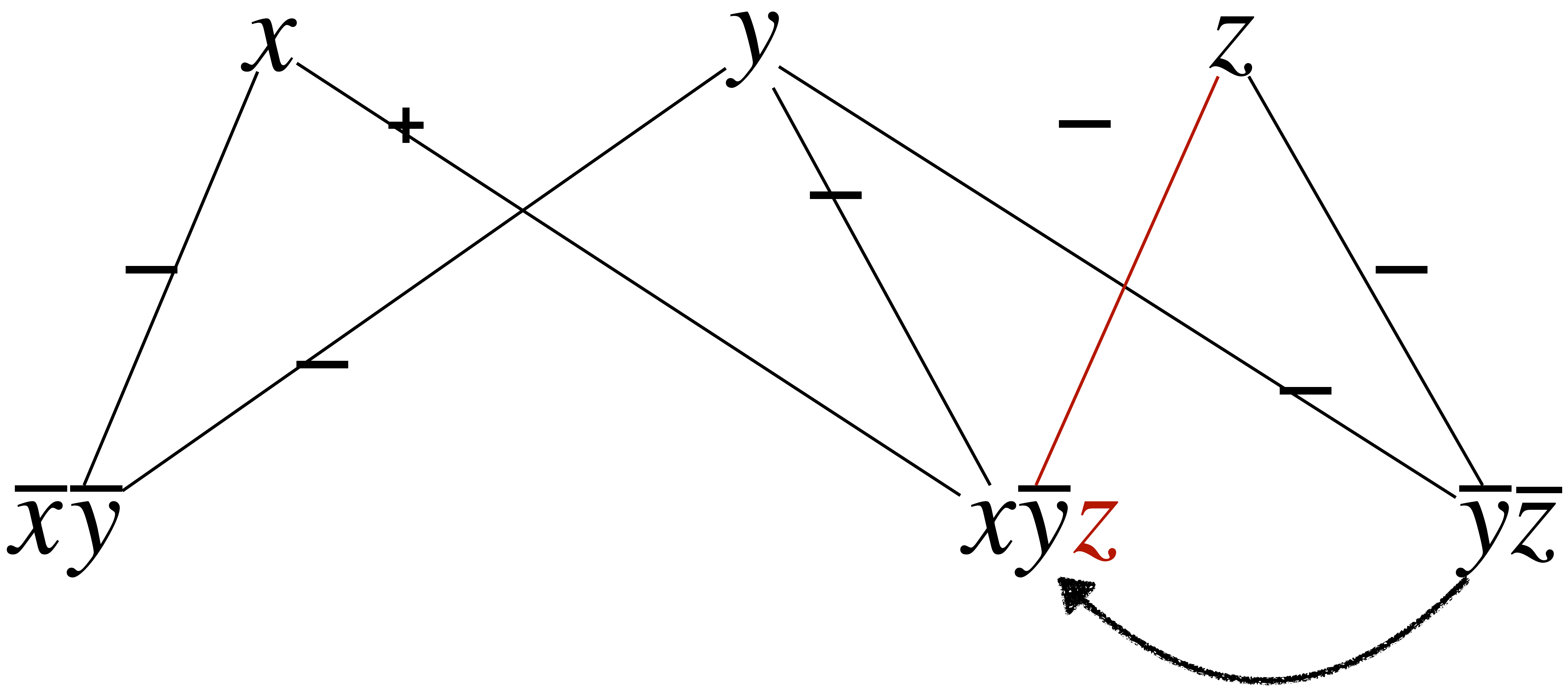
Example



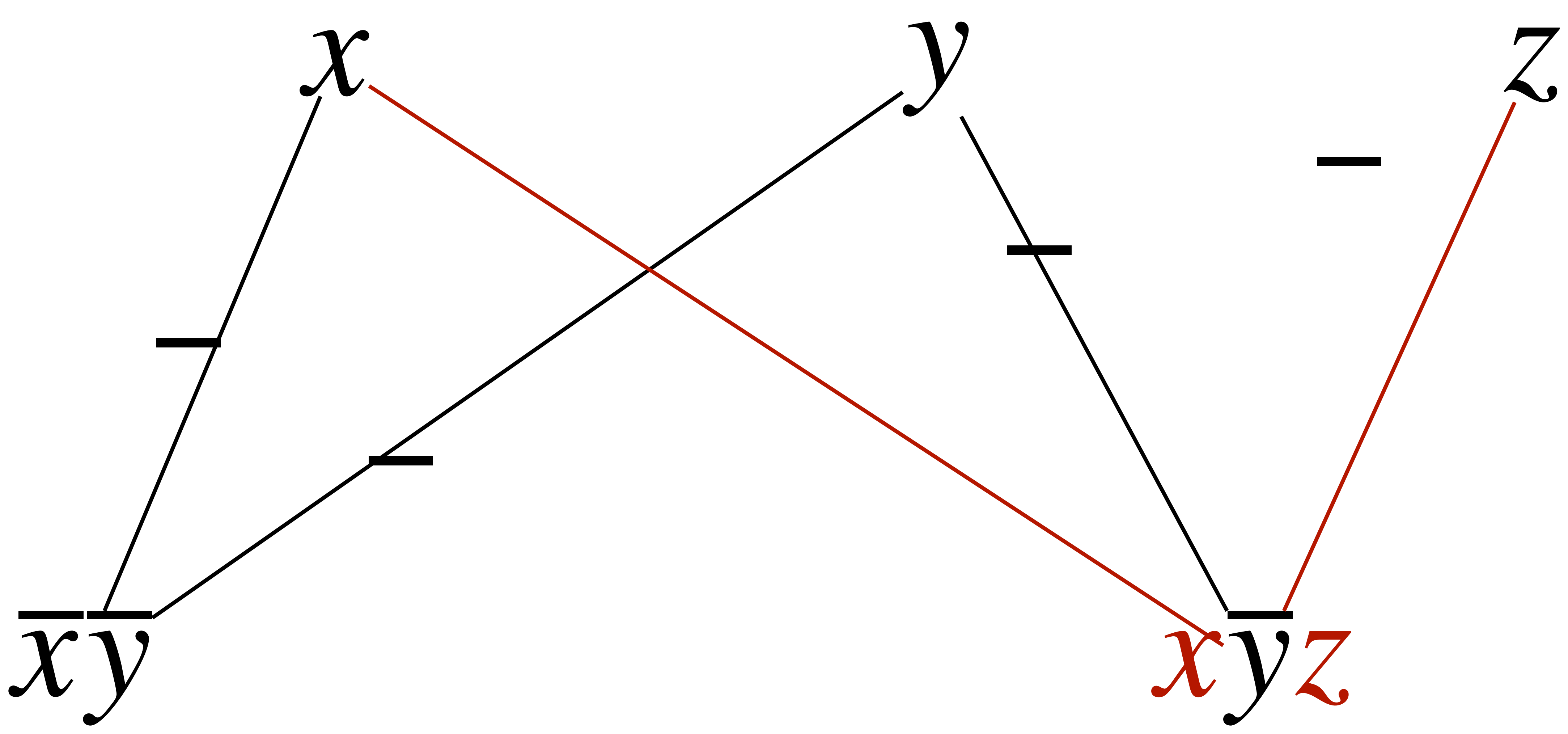
Example



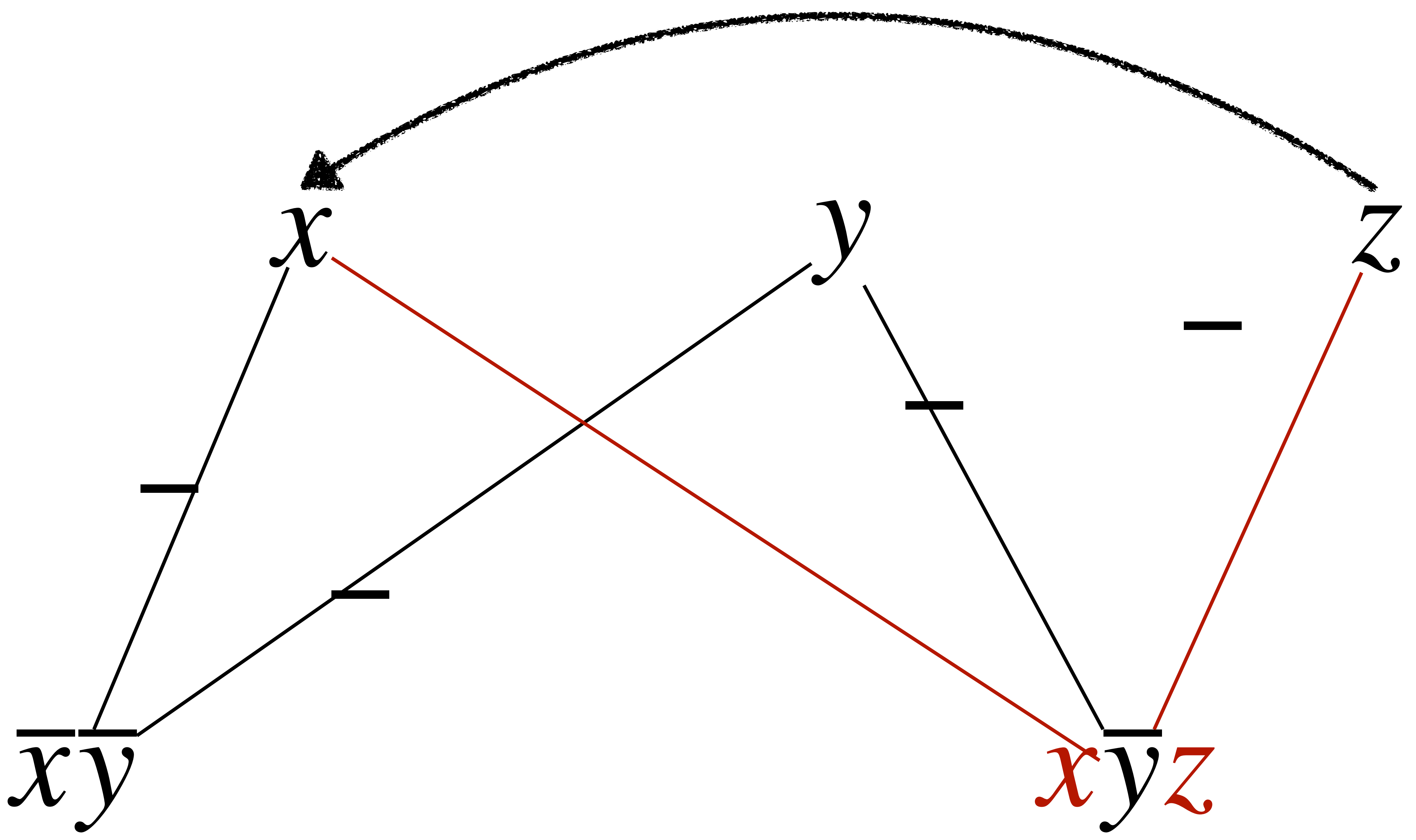
Example



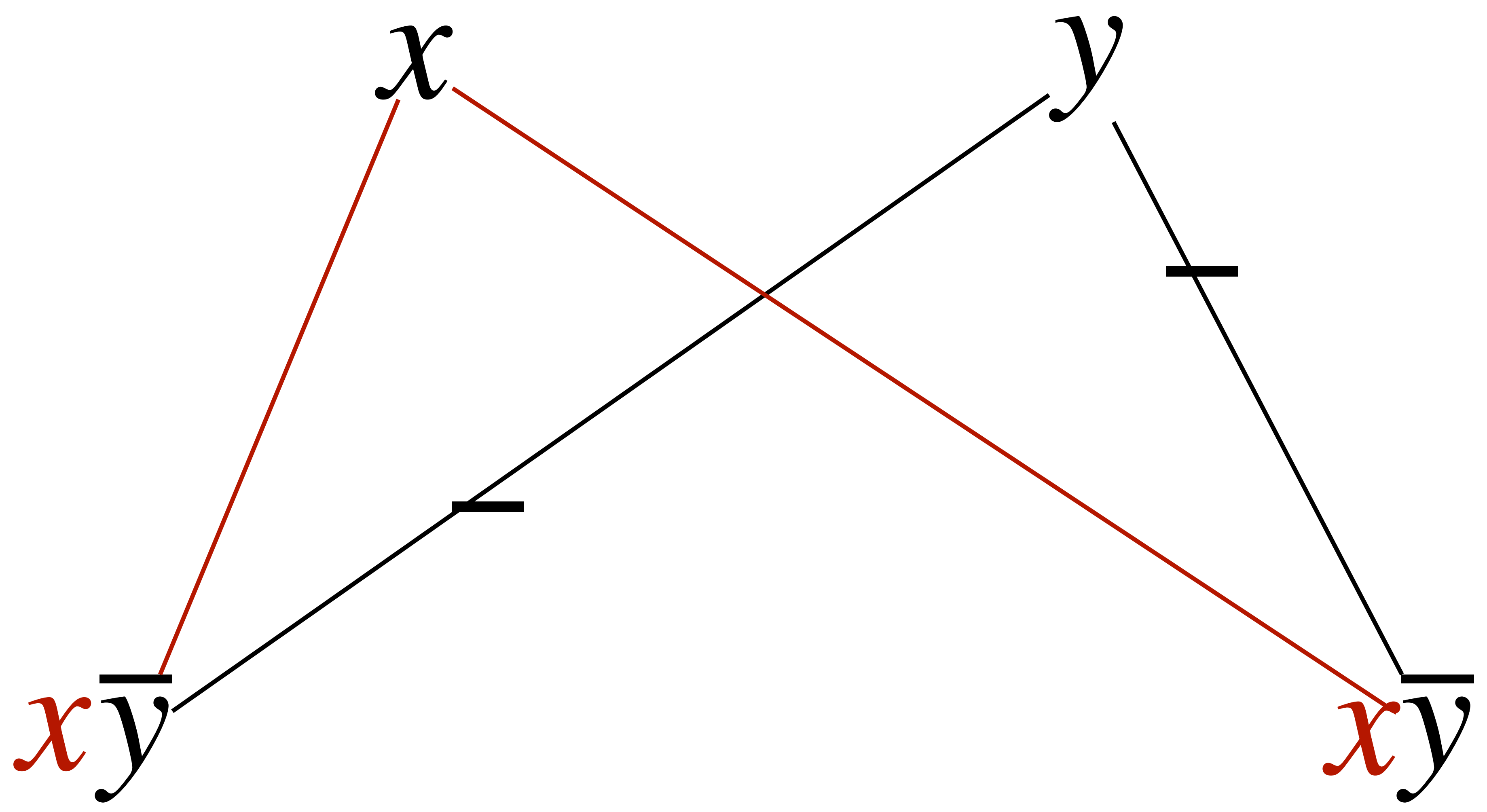
Example



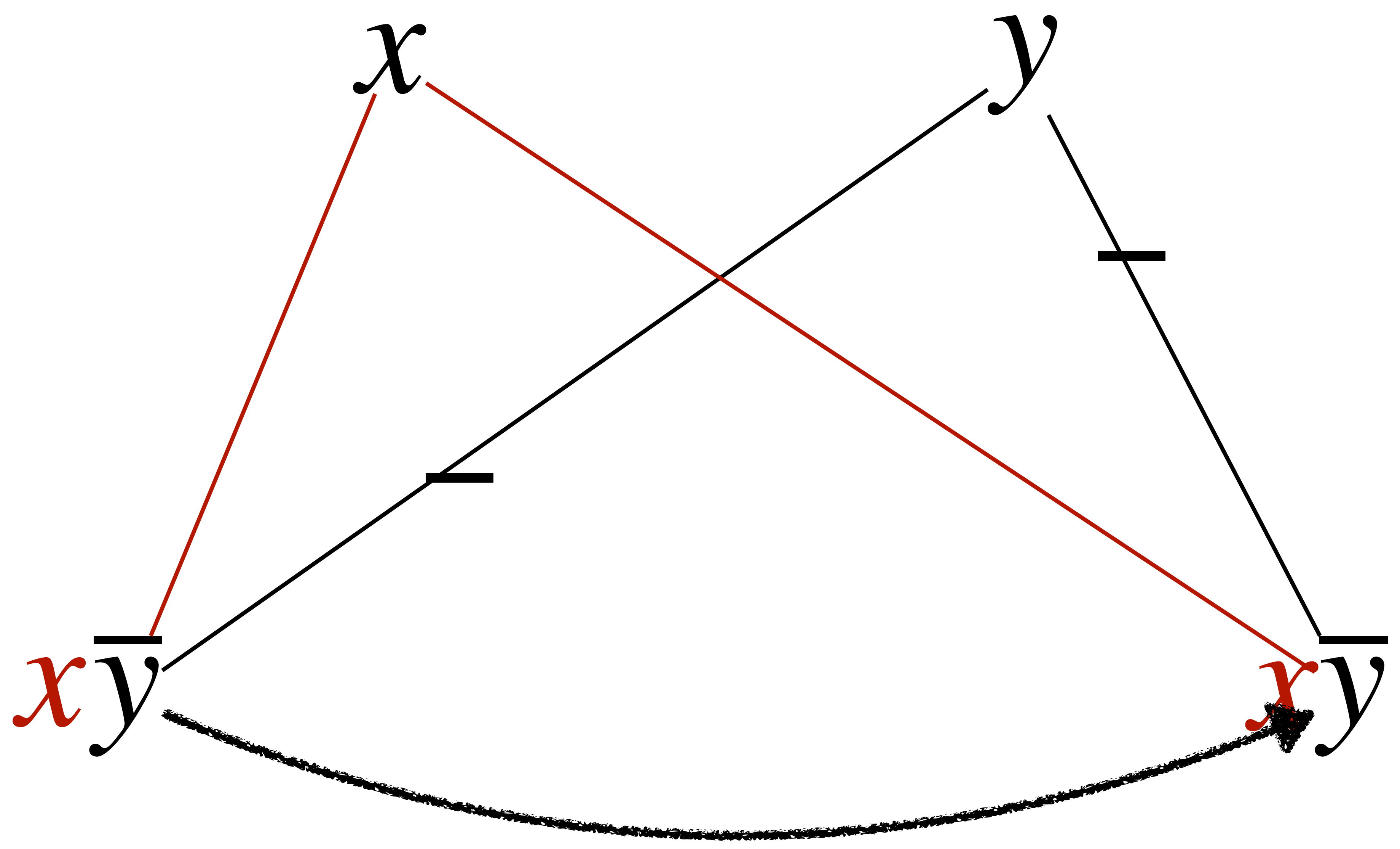
Example



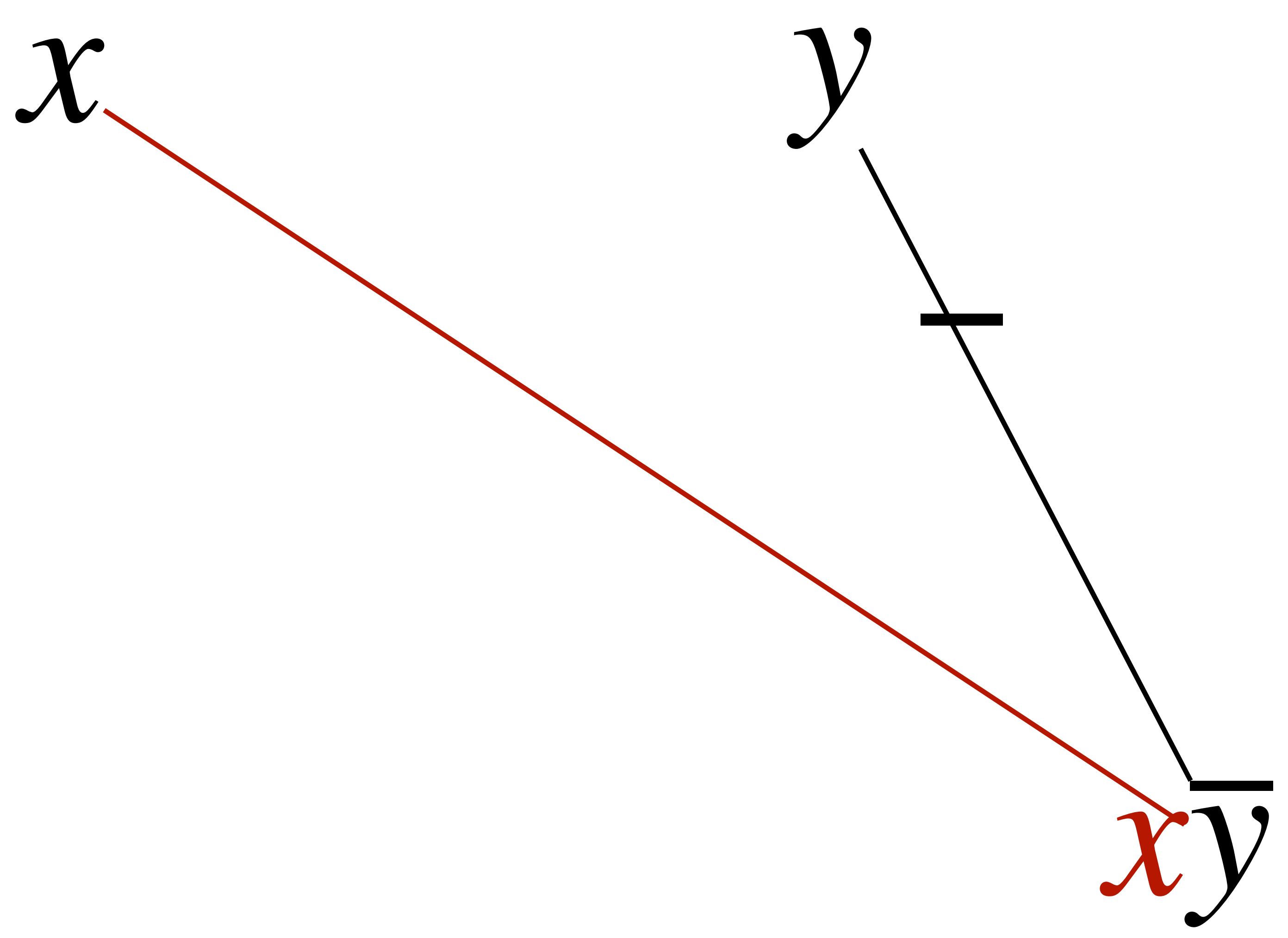
Example



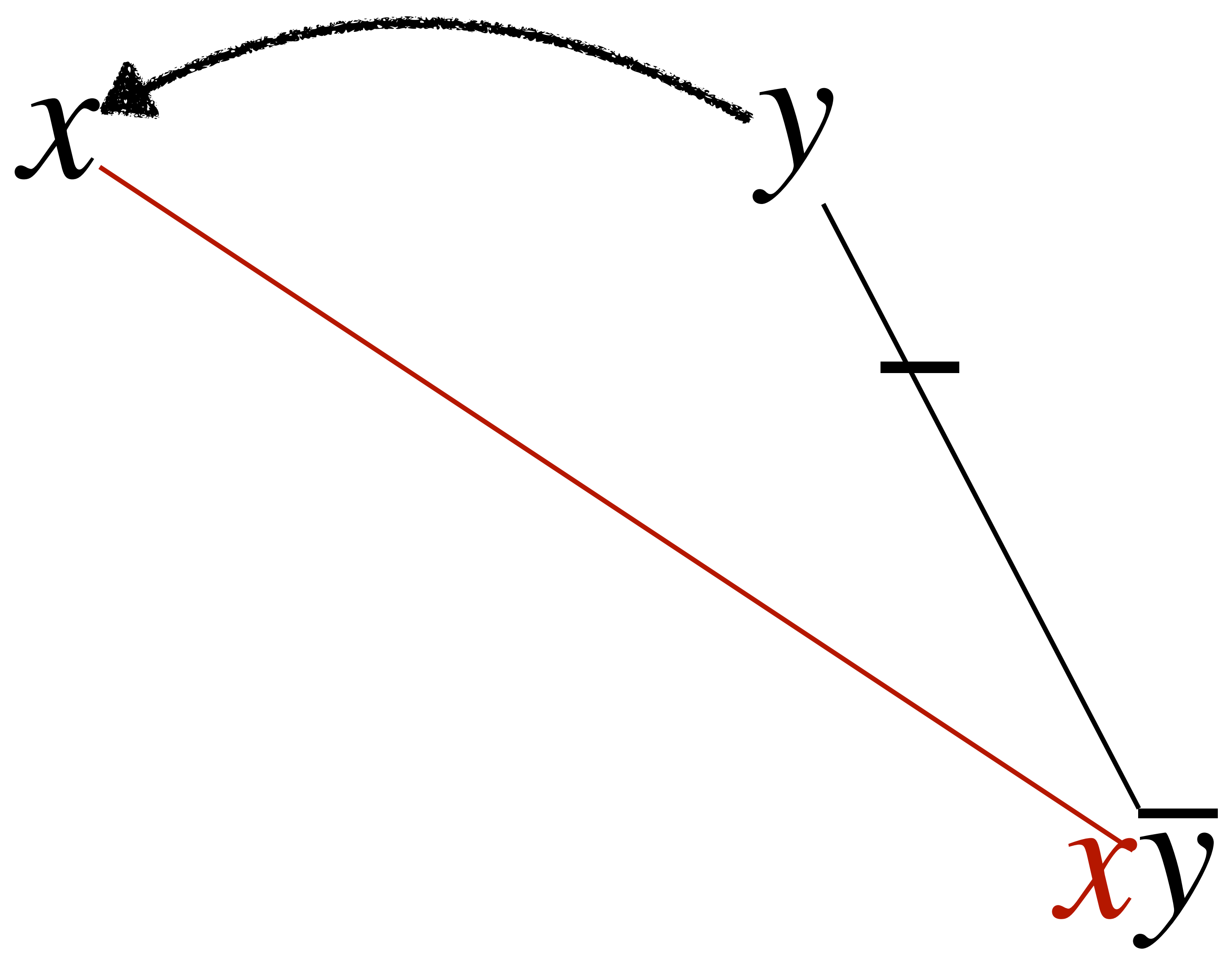
Example



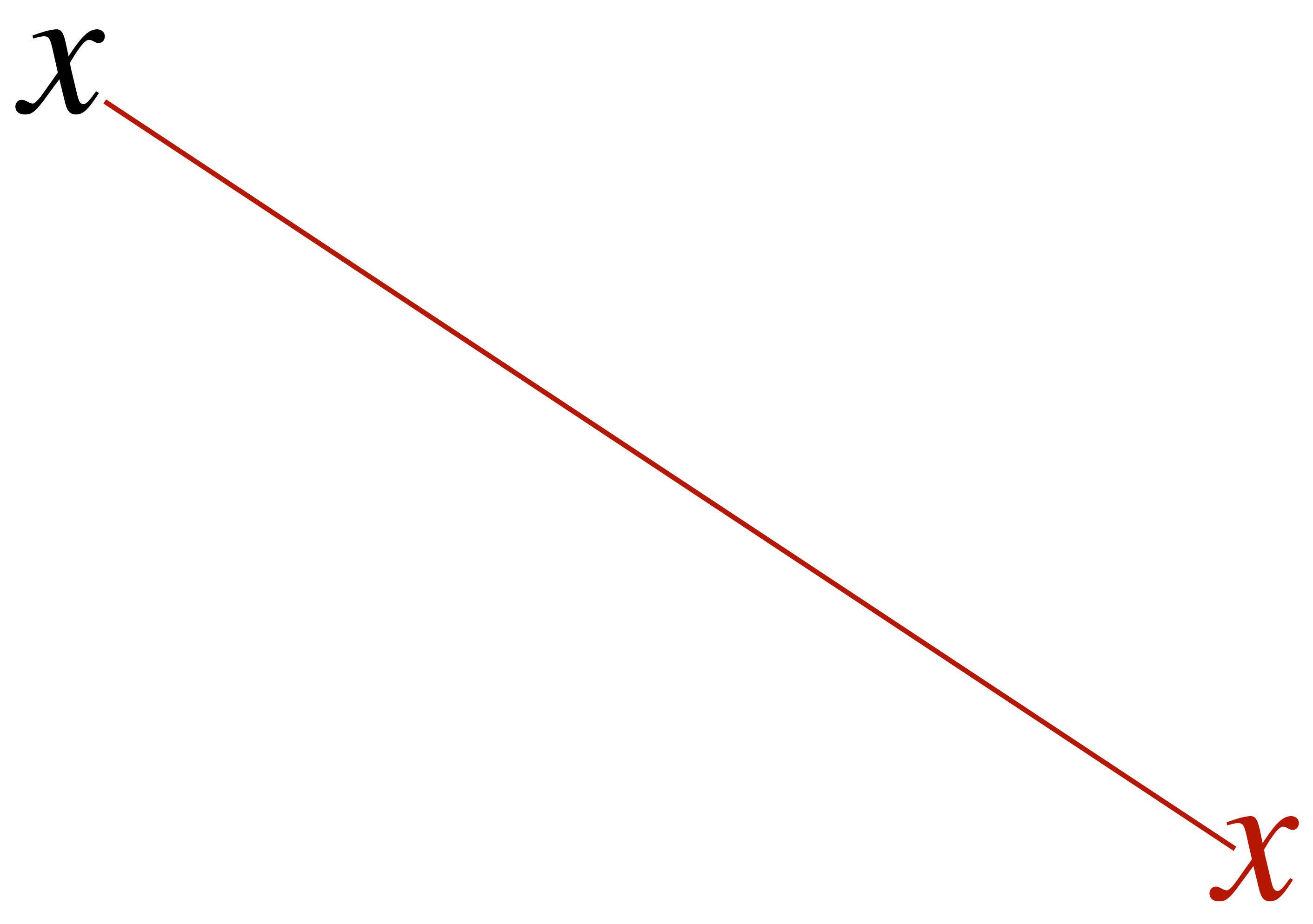
Example



Example



Example



Main Result [Ganian, et al. SAT 2022]

Bounded-ones Weighted Model Counting (BWMC)

Input: a CNF formula F where each literal is weighted $w(\ell) \in \mathbb{R}$, an integer $k \geq 0$

Task: compute the sum of weights for all satisfying assignments of F that set at most k variables to true.

BWMC generalizes WMC and SAT by setting k=|vars(F)|

Theorem: BWMC is fixed-parameter tractable parameterised by the certified signed tww of F and k.

None of the restrictions can be dropped.

Tightness

By reduction from Partitioned Clique.
Gadgets with k classes of clauses, in each class clauses are over the same vars.

	signed tww	tww	primal tww
k is parmeter	FPT	W[1]-hard	W[2]-hard
k is unrestricted	para-NP-hard	para-NP-hard	para-NP-hard

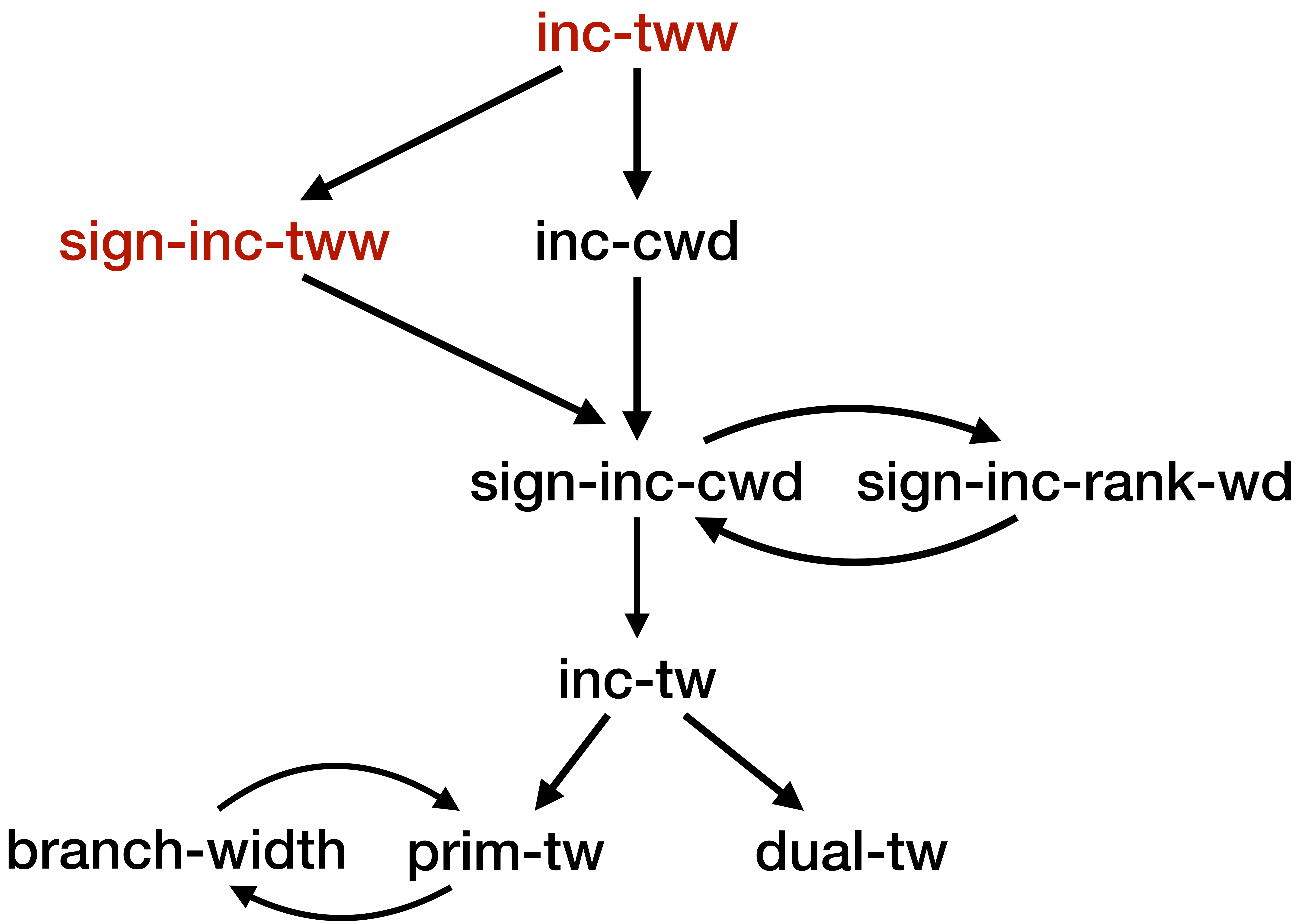
By reduction from Hitting Set.
We can make tww=0 by adding a dummy large clause.

By reduction from SAT.
We can make tww=0 by adding a dummy large clause.

Planar signed graphs have bounded tww.
SAT remains NP-hard for planar formulas [Lichtenstein 1982]

- All hardness results hold even if an optimal contraction sequence is provided, and only SAT decision is queried.

Relation to other width parameters

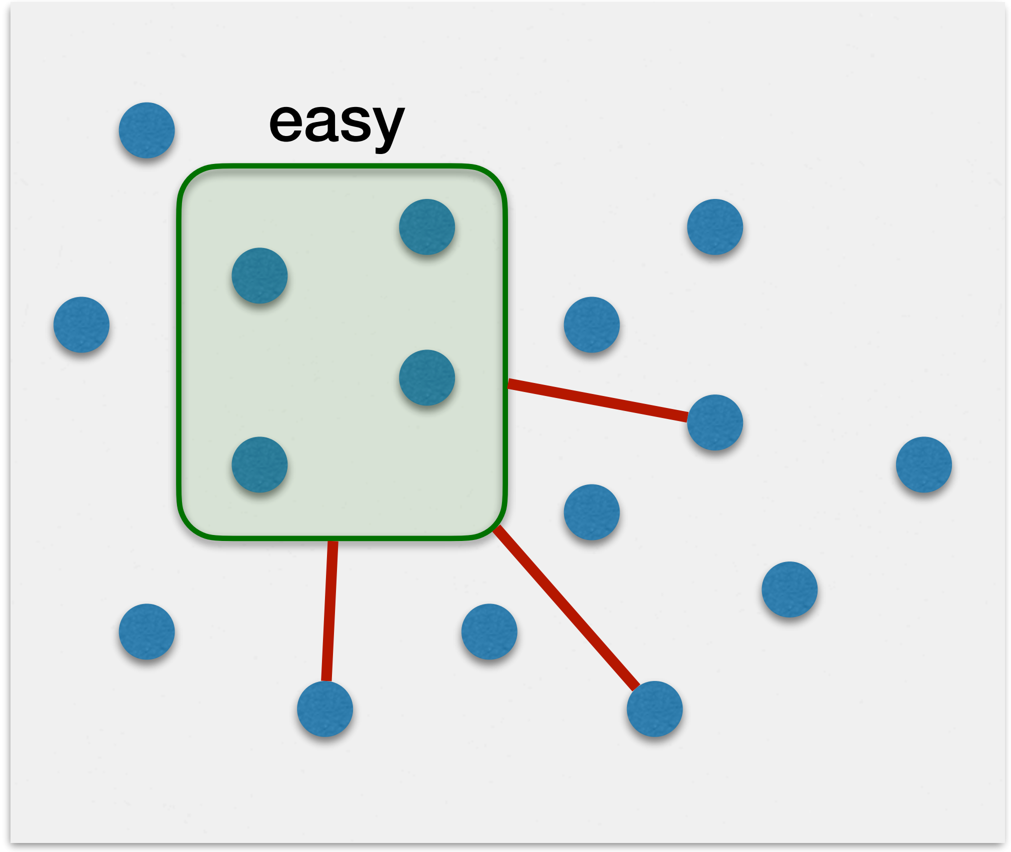


FLoC 2022

SAT:
Wed Aug 3, 17:00

Syntactic Structure

Tractable Classes or Islands of Tractability

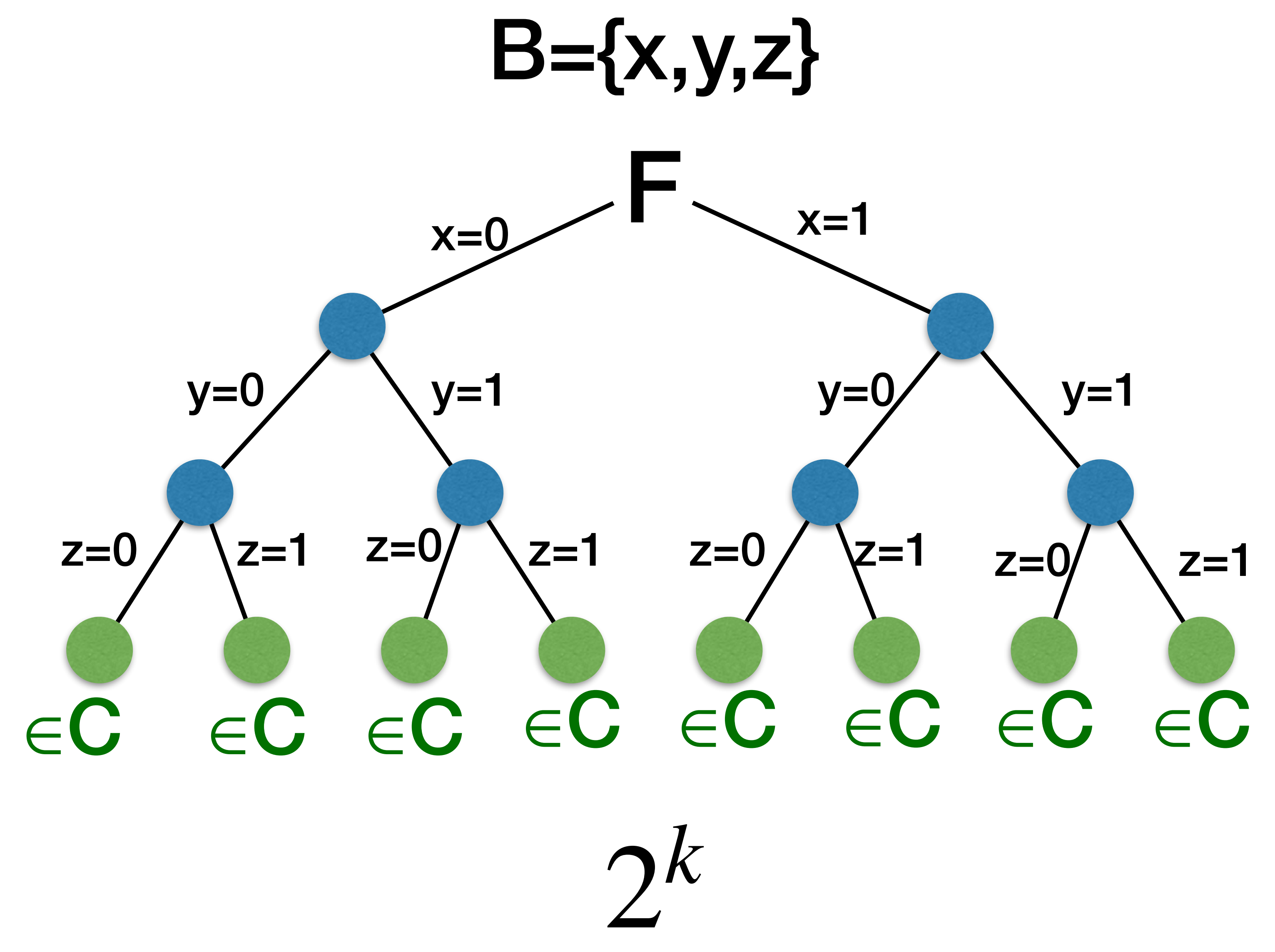


Parameterize by the **distance to a class**

where the class is syntactical defined

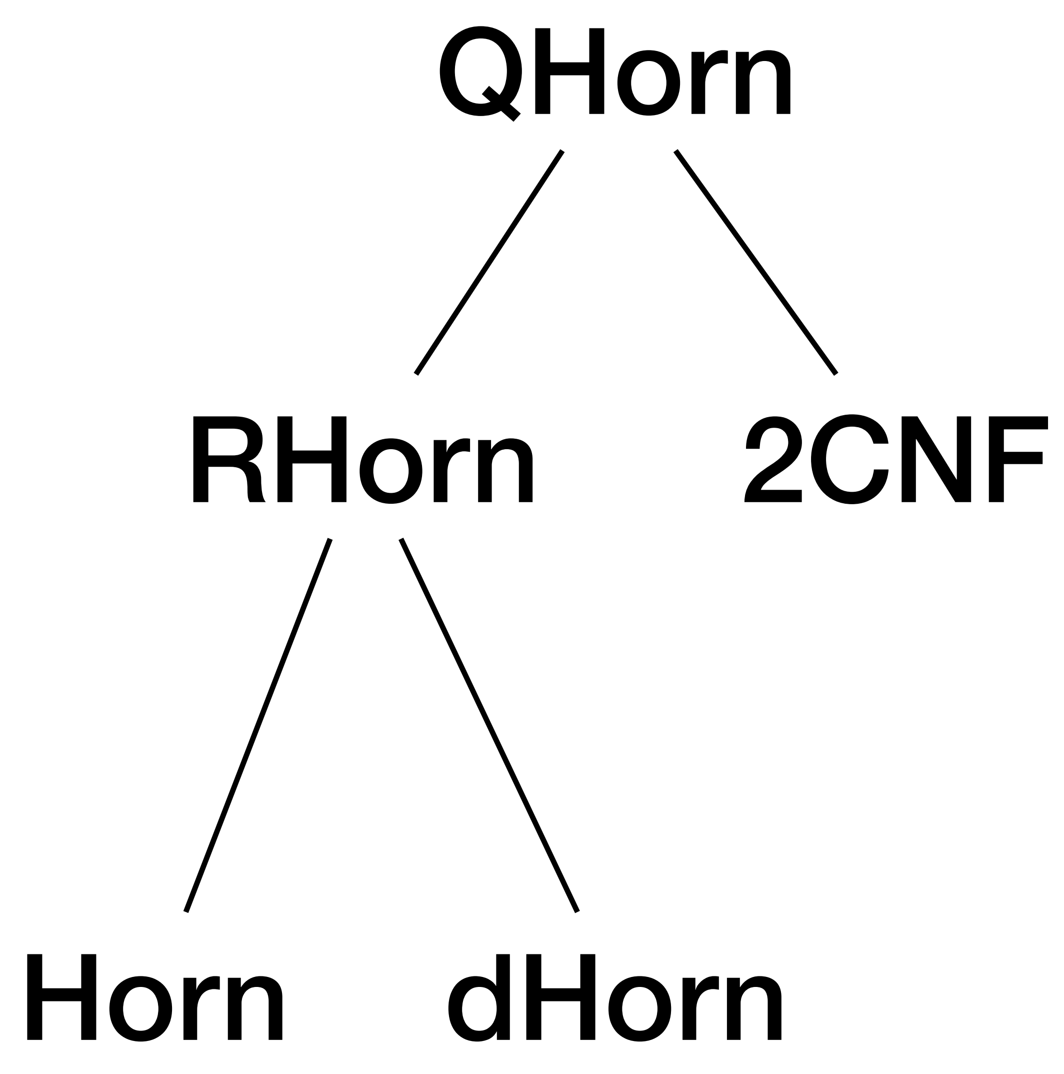
Distance = size of smallest strong backdoor set

- Fix a base class C (e.g., Horn)
- **B is a strong C-backdoor of F** if for all assignments t:B →{0,1} we have F[t] ∈ C.
- F[t] is obtained from F by removing clauses from F which contain a literal that t sets to 1, and removing from the remaining clauses all literals that t sets to 0



Syntactic Base Classes

- **Horn:** each clause contains at most one positive literal
- **dual Horn:** each clause contains at most one negative literal
- **2CNF** (or Krom): each clause contains at most 2 literals
- **RHorn:** can be made Horn by consistently flipping literals
- **QHorn:** there exists a function $v : \mathrm{var}(F) \rightarrow [0,1]$ such that $v(x) + v(\bar{x}) = 1$ and $\sum_{x \in C} v(x) \leq 1$ for all clauses C of F.

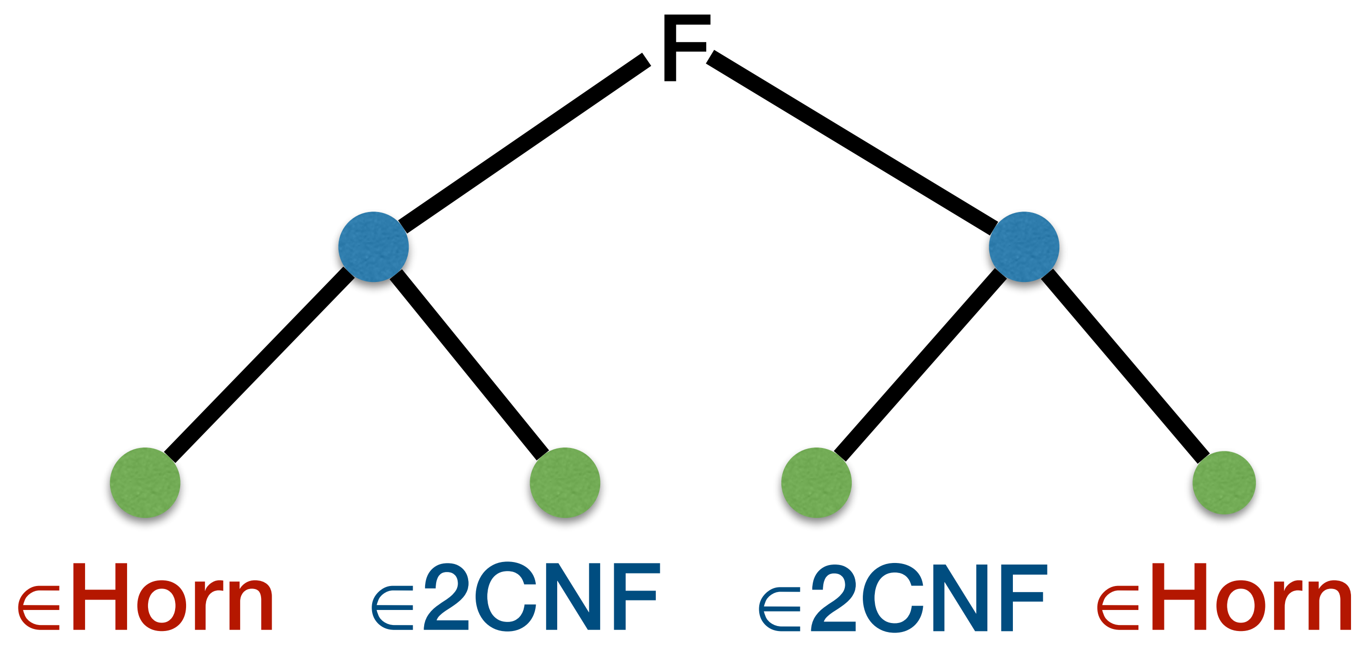


Other base classes

- **HIT:** any two clauses of the forma contain a complementary pair of literals
- **CLU:** variable-disjoint union of HIT formulas
- **W[t]:** formulas of incidence treewidth at most t.
- From base classes C and D we can form
  - the **heterogeneous** base class C ∪ D and
  - the **scattered** base class C ⊕ D

A hitting formula is unsatisfiable if

$$\sum_{C \in F} 2^{-|C|} = 1$$

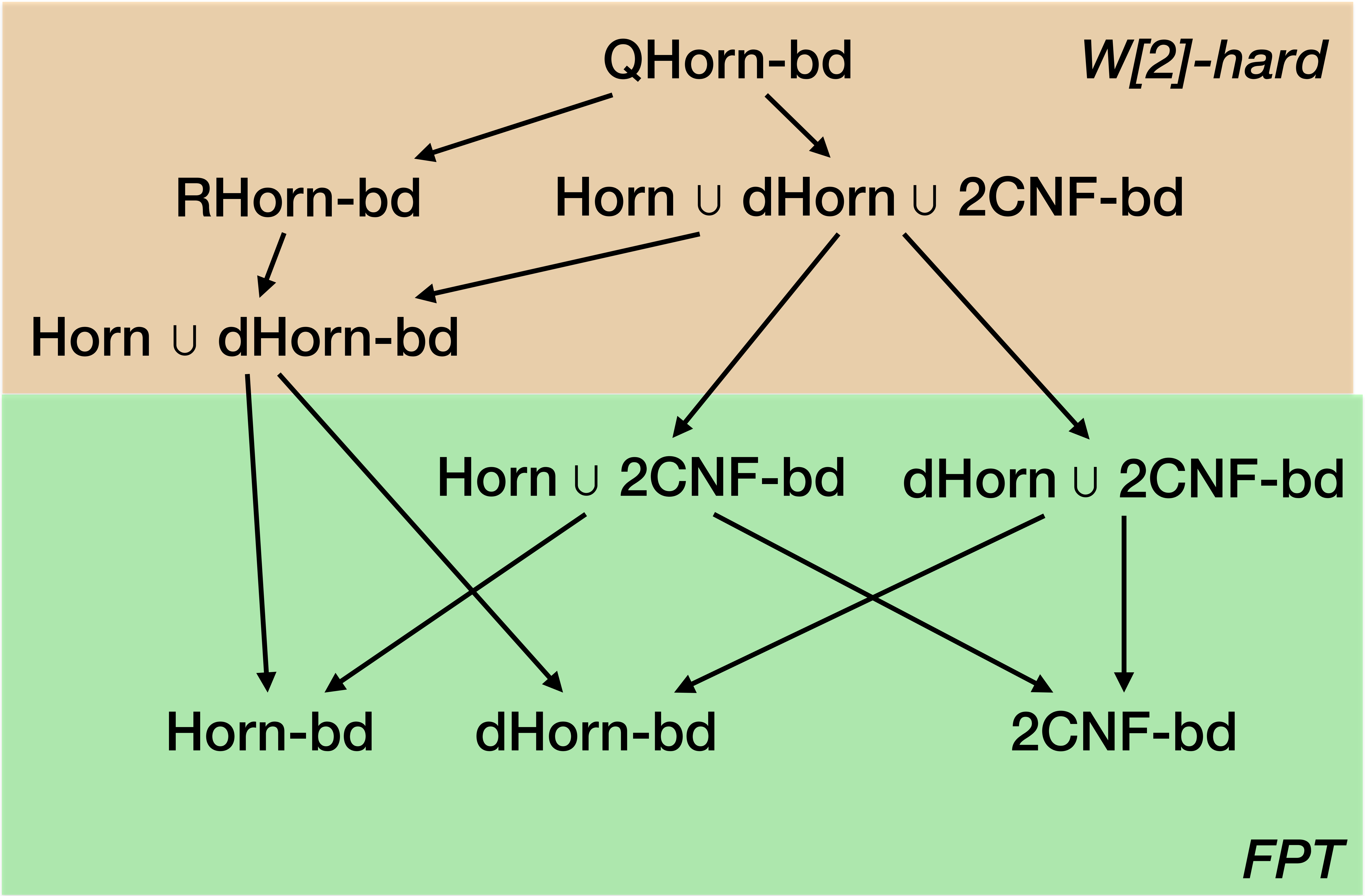


heterogeneous base classes

h-modularity

- Any parameter that resembles modularity but gives runtime guarantees?
- h-modularity [Ganian, Sz. AIJ 2021]
    - partition clauses into clusters of HIT formulas
    - contract each cluster into a single vertex
    - take the treewidth of the resulting graph
    - h-modularity: smallest tw over all possible partitions

Backdoor Parameter Zoo



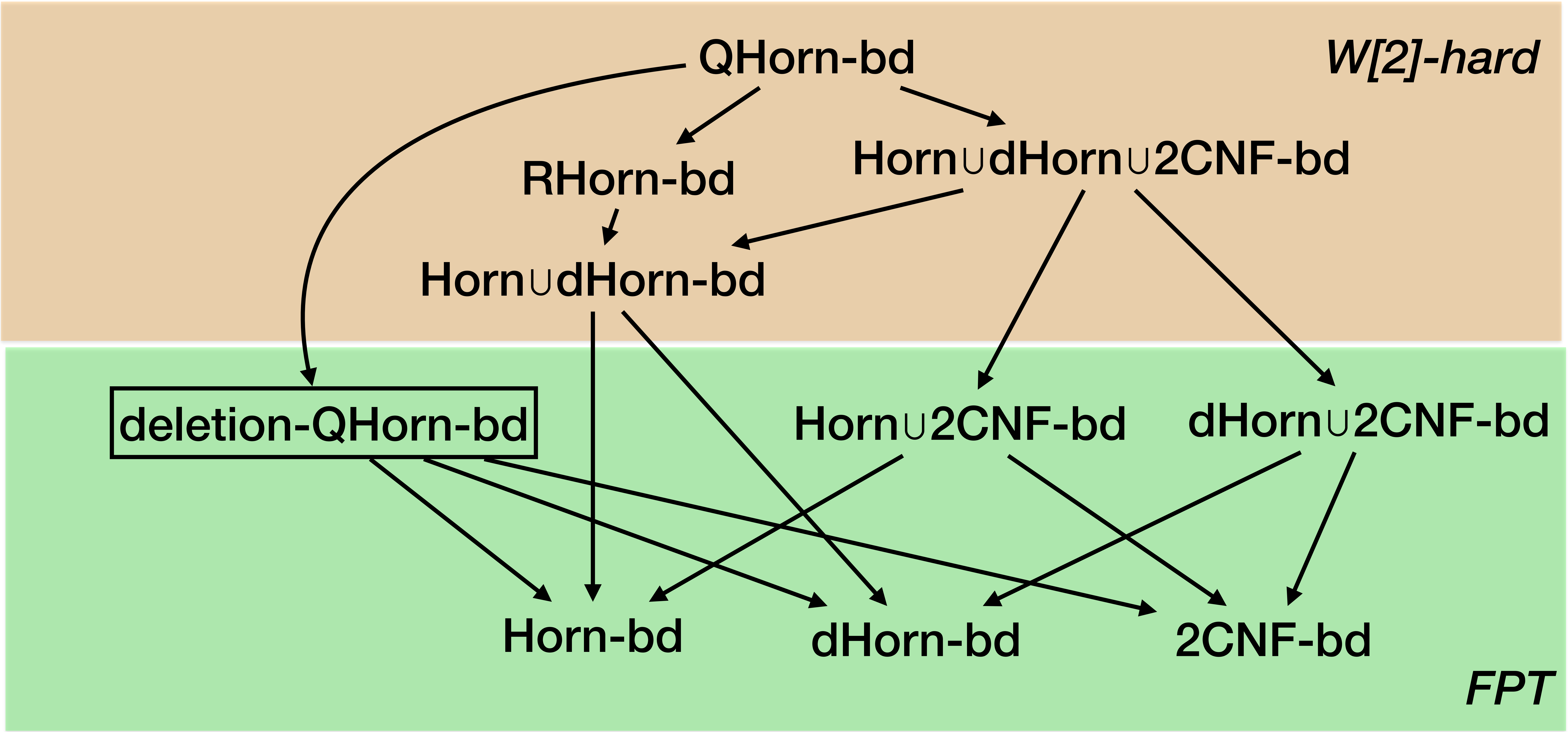
Deletion backdoor sets

- B is a **deletion backdoor** if $F - B \in C$.

- Instead of looking at all partial assignments $t : B \to \{0,1\}$ we delete the backdoor variables from F (notation $F - B$)

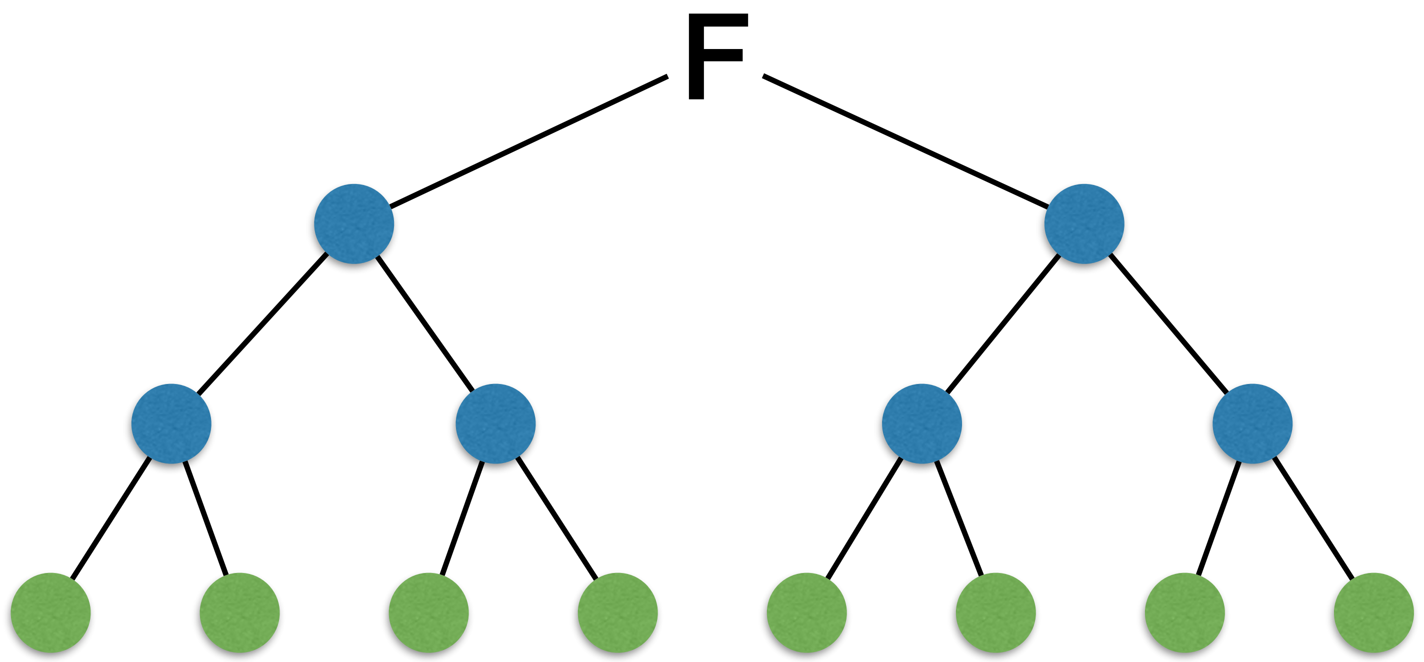
- Fact: if C is clause-induced ($F' \subseteq F, F \in C \Rightarrow F' \in C$) then each deletion backdoor set is also a backdoor set (but not necessarily the other way around)

Deletion Backdoor Sets

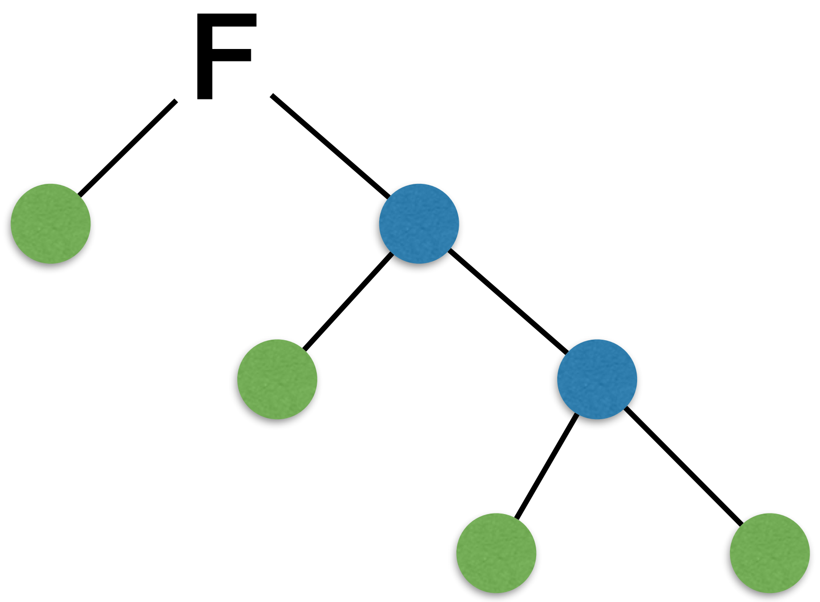


[Samer, Sz. AAAI 2008], [Ordyniak, Sz. IJCAI 2021]

Avoid the 2^k assignments: Backdoor Trees



2^k



$k+1$

size of backdoor tree = number of leaves

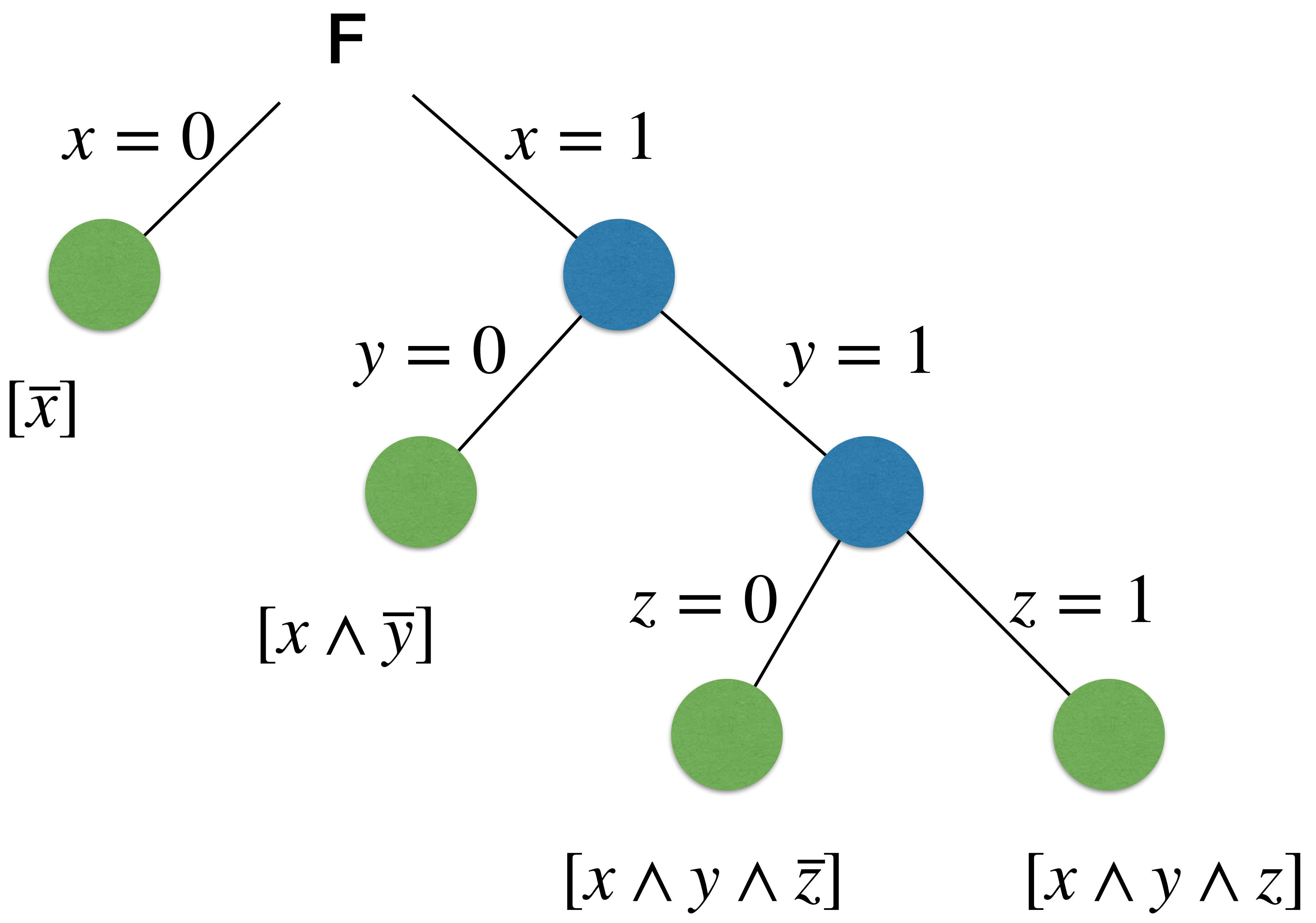
- smallest backdoor sets ≠ backdoor trees with smallest number of leaves!
- subset-minimal backdoor sets ≠ backdoor trees with smallest number of leaves

Finding backdoor trees with k leaves is FPT for Horn, dHorn, and 2CNF

even heterogeneous base class Horn ∪ 2CNF

Avoid the 2^k assignments: Backdoor DNFs

- Partial assignments at the leaves of a backdoor tree give rise to a DNF
- The DNF is a tautology

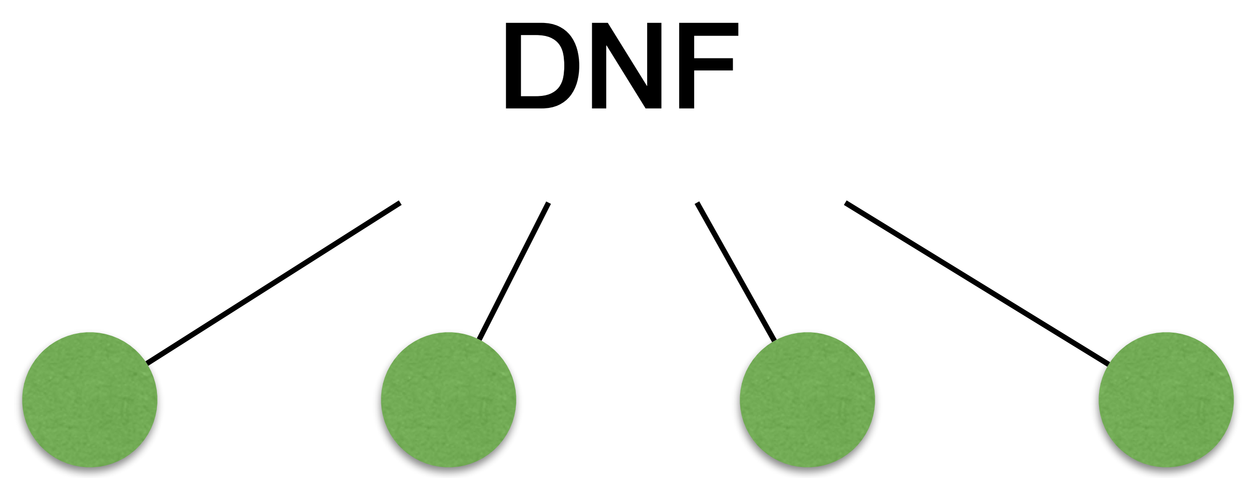


$$[\bar{x}] \vee [x \wedge \bar{y}] \vee [x \wedge y \wedge \bar{z}] \vee [x \wedge y \wedge z]$$

[Ordyniak, Sz. IJCAI 2021]

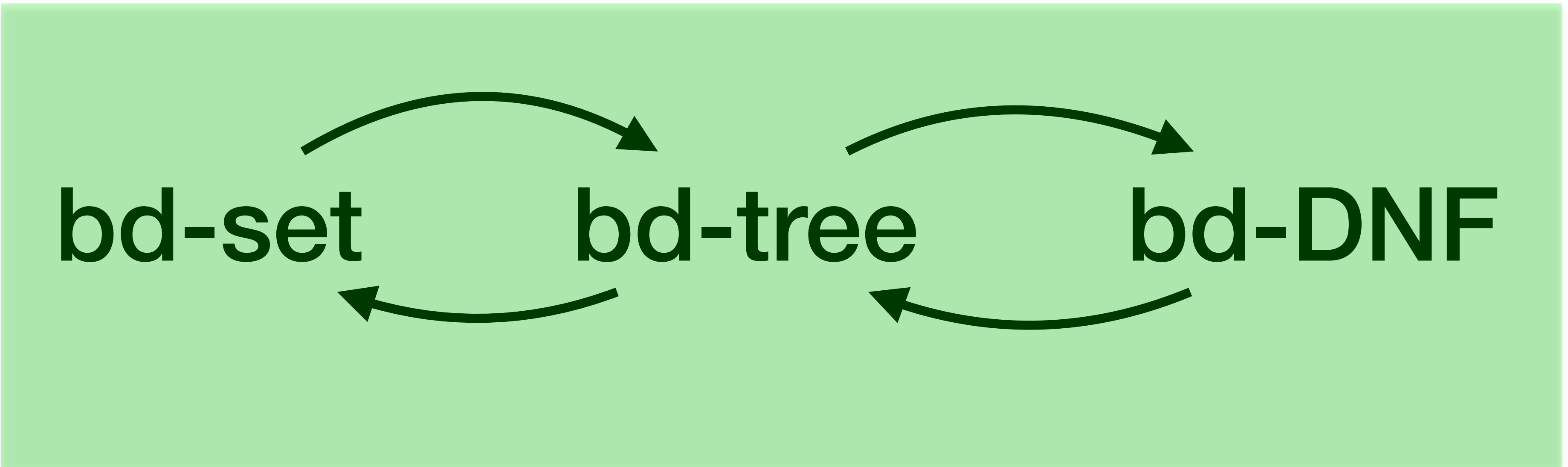
Avoid the 2^k assignments: Backdoor DNFs

- Partial assignments at the leaves of a backdoor tree give rise to a DNF
- The DNF is a tautology
- **Backdoor DNF**: take **any** such tautological DNF
- Backdoor DNFs are more succinct than backdoor trees



Finding backdoor DNFs with k terms is FPT for Horn, dHorn, and 2CNF

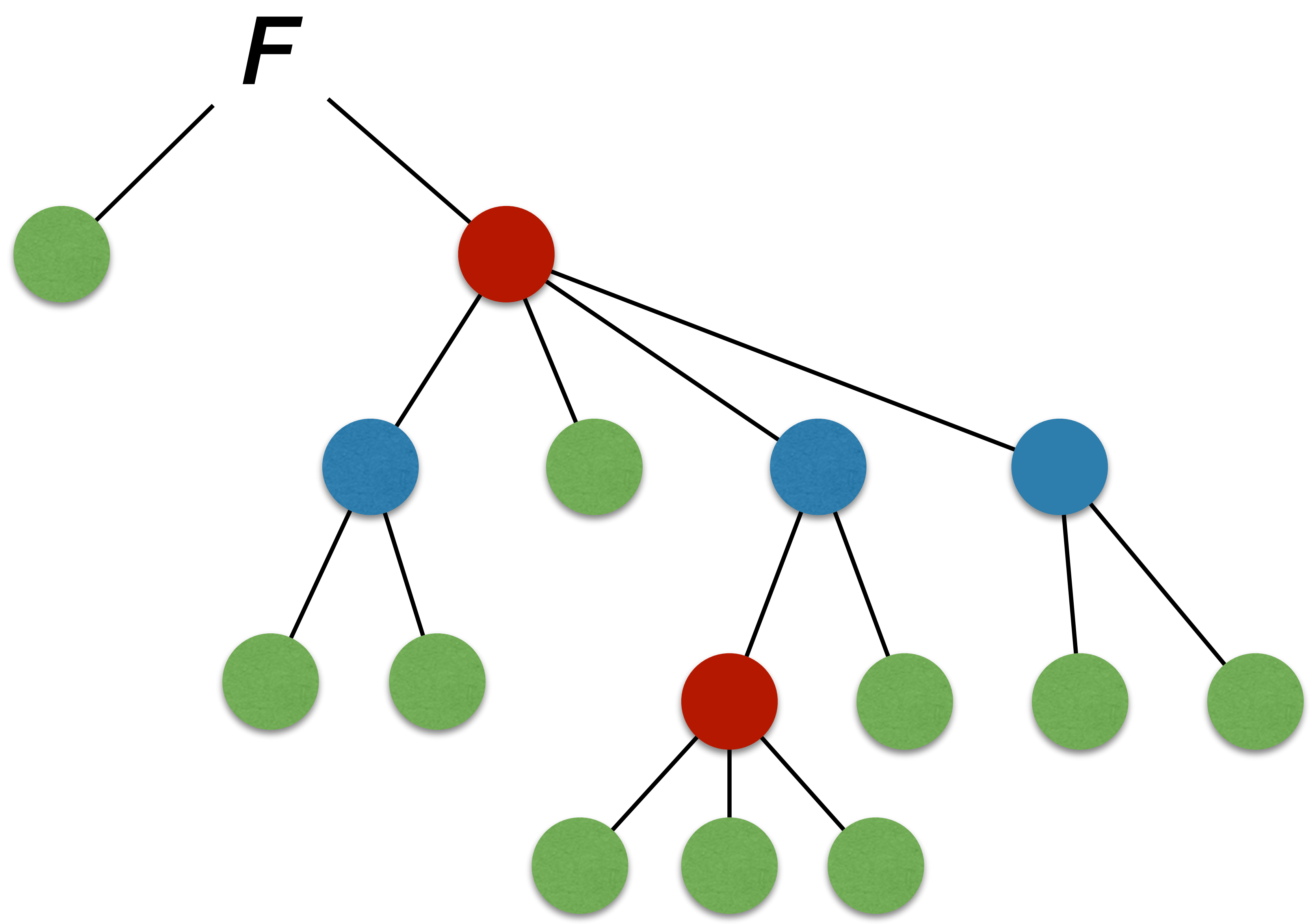
one can even mix Horn with 2CNF (or dHorn with 2CNF)



Backdoor Depth

[Mählmann, Siebertz, Vigny, MFCS 2021]

Component backdoor trees

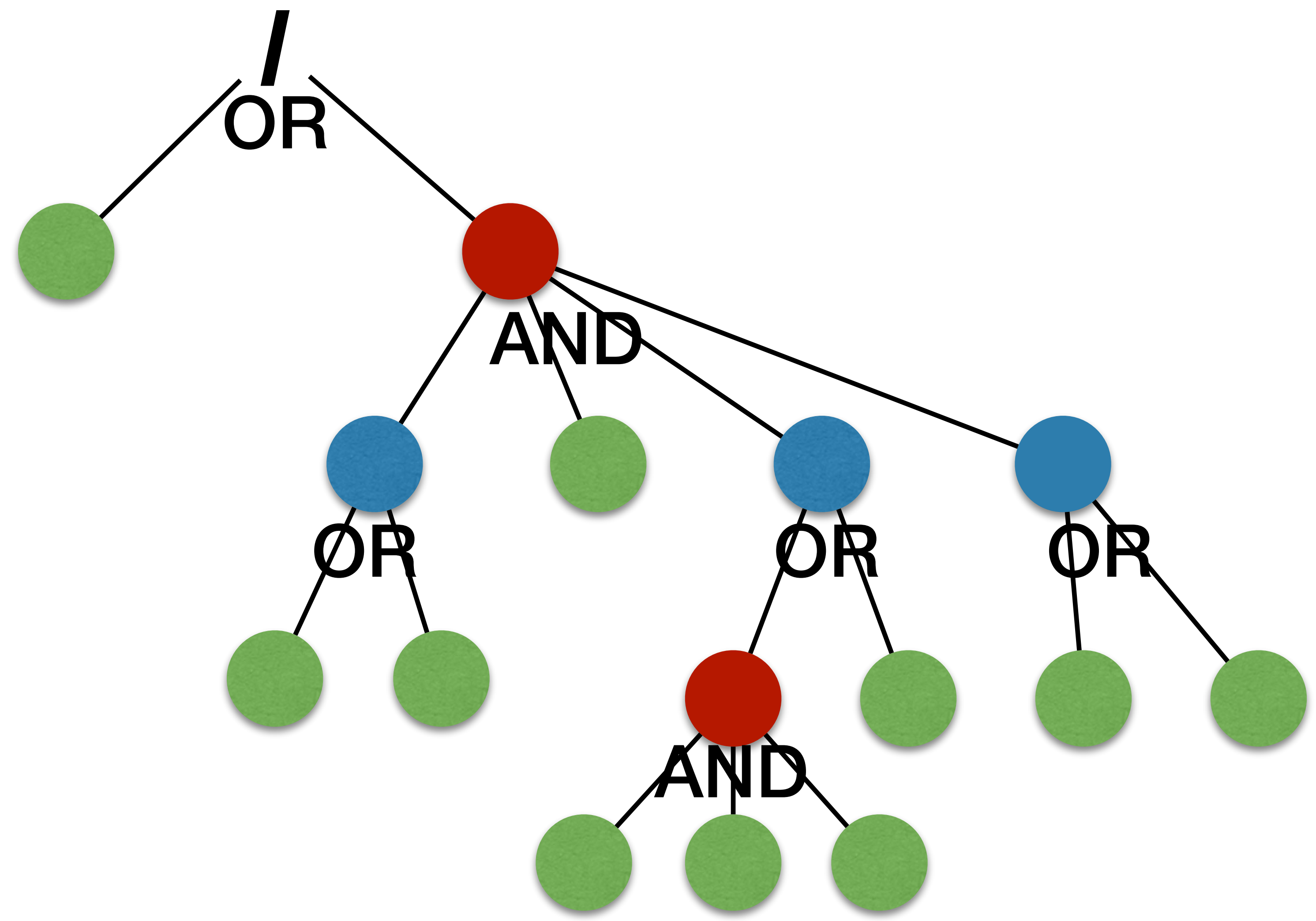


component nodes (red) split instance into connected components.

- backdoor depth: smallest depth of any component backdoor tree
- for fixed depth, number of variables in the backdoor is unbounded!

Component backdoor Trees

- Backdoor depth is significantly better parameter than backdoor size or number of backdoor tree leaves
- Definition motivated by treedepth [Nesetril, Ossona de Mendez 2006]
- Once we have a component backdoor tree that witnesses the backdoor depth of a given instance, we can decide the instance quickly
- Algorithmically challenging problem: find a component backdoor tree of small depth

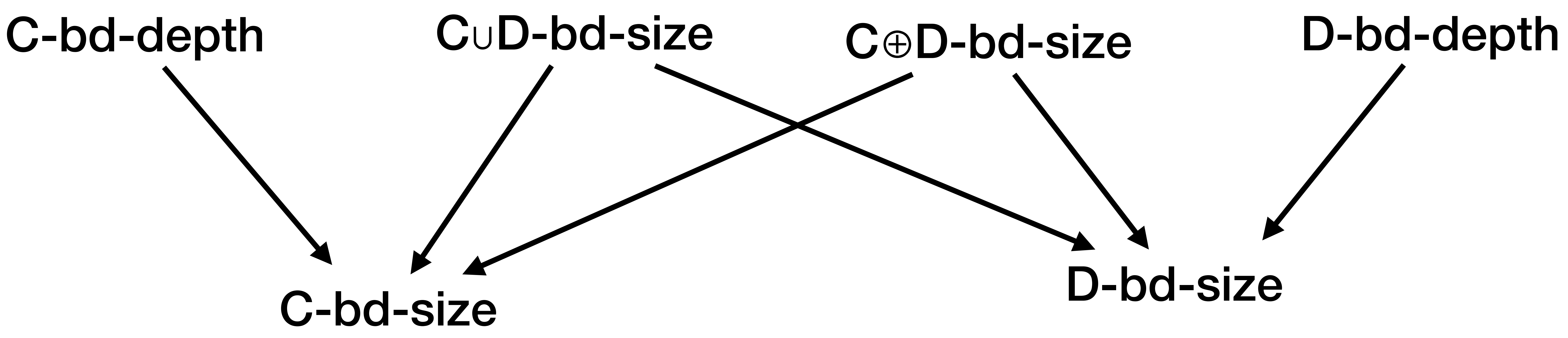


FPT-approximating backdoor depth

- FPT approximation for base class NULL [Mählmann, Siebertz, Vigny, MFCS 2021]
- FPT approximation for the base classes Horn and 2CNF [Dreier, Ordyniak, Sz. ESA 2022]
    - starting point: obstruction trees from Mählmann et al.
    - Separator obstructions can separate obstruction trees containing an unbounded number of variables from all potential future obstruction trees.
    - Use game theoretic framework for specifying the algorithm

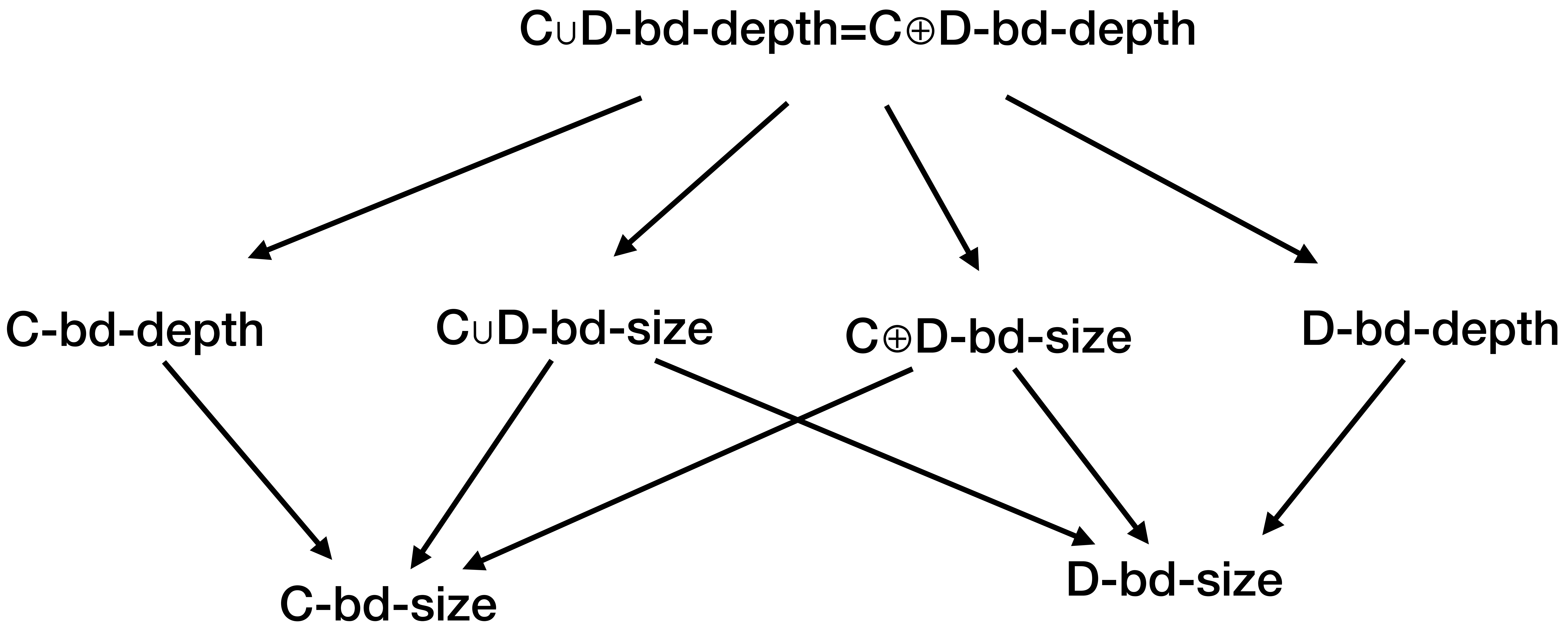
FLoC 2022
CP:
Tue Aug 2, 9:00

Comparison Summary



What's next?

FPT?



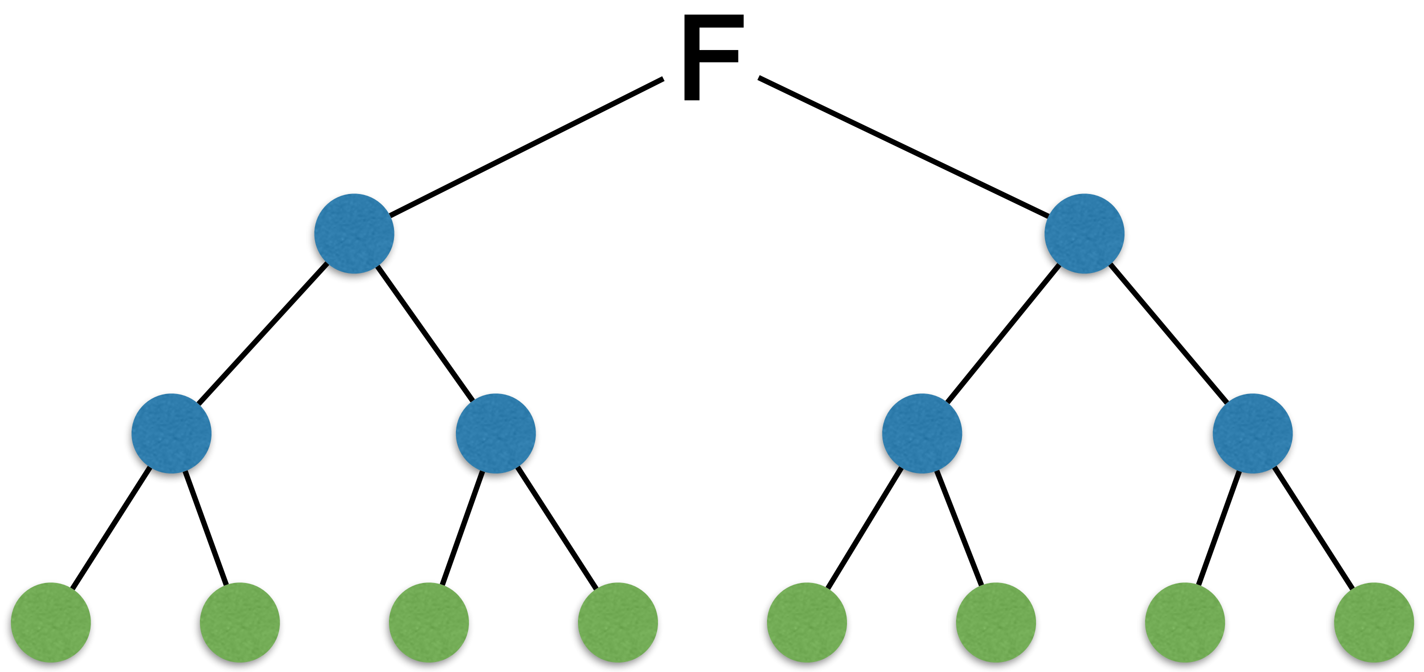
Hybrid parameters

large incidence treewidth
constant Horn-bd size

INCOMPARABLE

large Horn-bd size
constant incidence treewidth

(A) Backdoors into bounded treewidth



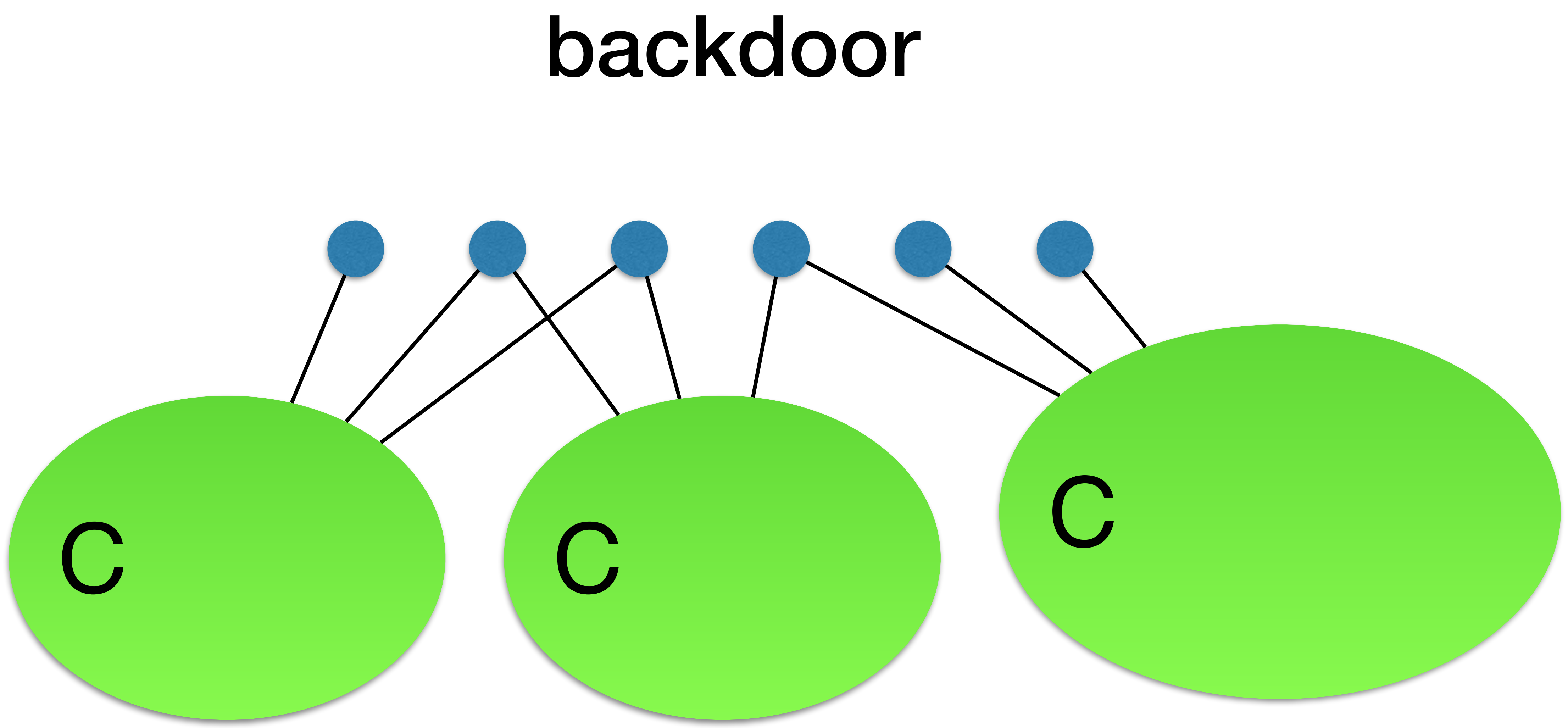
TW[t]

TW[t] = {F | tw(I(F)) ≤ t }

- deletion backdoors are not interesting, but strong backdoors are!

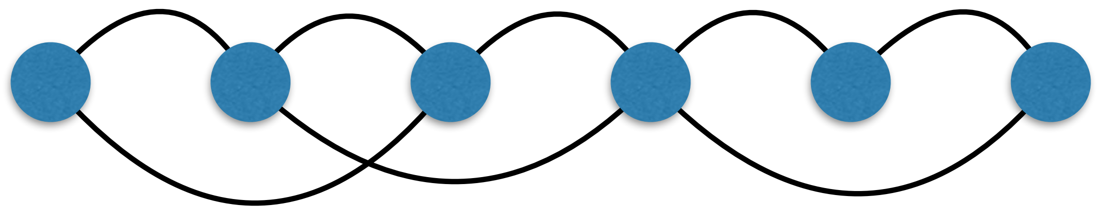
For each constant t, TW[t]-backdoor detection is FPT-approx.

(B) backdoor treewidth



(B) backdoor treewidth

backdoor



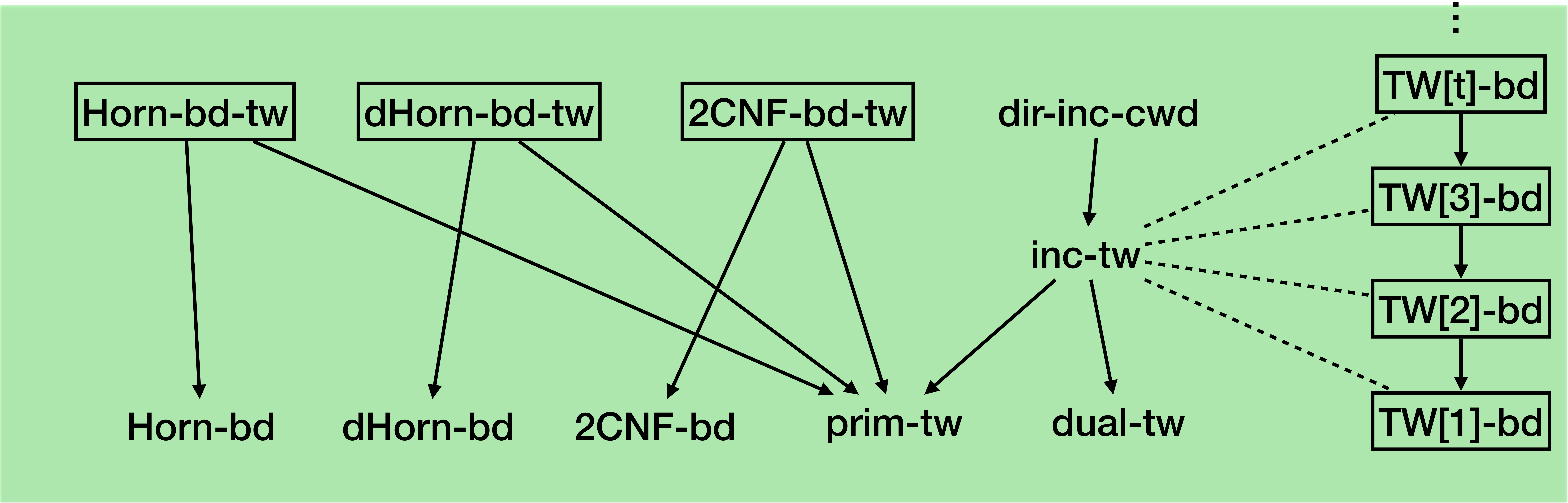
torso graph

- **C-backdoor treewidth** is the minimum treewidth over the torso graphs of all the C-backdoors.
- C-backdoor treewidth
  ≤ min{ primal treewidth, C-backdoor size}

C-backdoor treewidth is FPT
for C ∈ {Horn,dHorn,2CNF}

FLoC 2022
PCCR:
Mon Aug 1, 14:00

Parameter Zoo



Resolution

Resolution: proofs of unsatisfiability

- To certify that a formula is satisfiable, just provide a satisfying assignment
- To certify that a formula is unsatisfiable, we need a proof.
- There are many proof systems, resolution is the most fundamental one.
- Idea: consider all clauses of the input formula as axioms.
- From two clauses already obtained and they contain a pair of closing literals, obtain their resolvent as new clause.
- When you derive the empty clause, you can stop.

$$\frac{\{u, \overline{v}, w\} \qquad \{\overline{x}, y, \overline{u}\}}{\{\overline{v}, w, \overline{x}, y\}}$$

tree-like vs sag-like

Resolution and SAT-solvers

- Fact: a formula is unsatisfiable if and only if it has a resolution proof
- DAG-like resolution is exponentially more succinct then tree-like resolution
- CDCL SAT solver runs on unsatisfiable formulas can be interpreted as dag-like resolution proofs.

Resolution and FPT algorithms

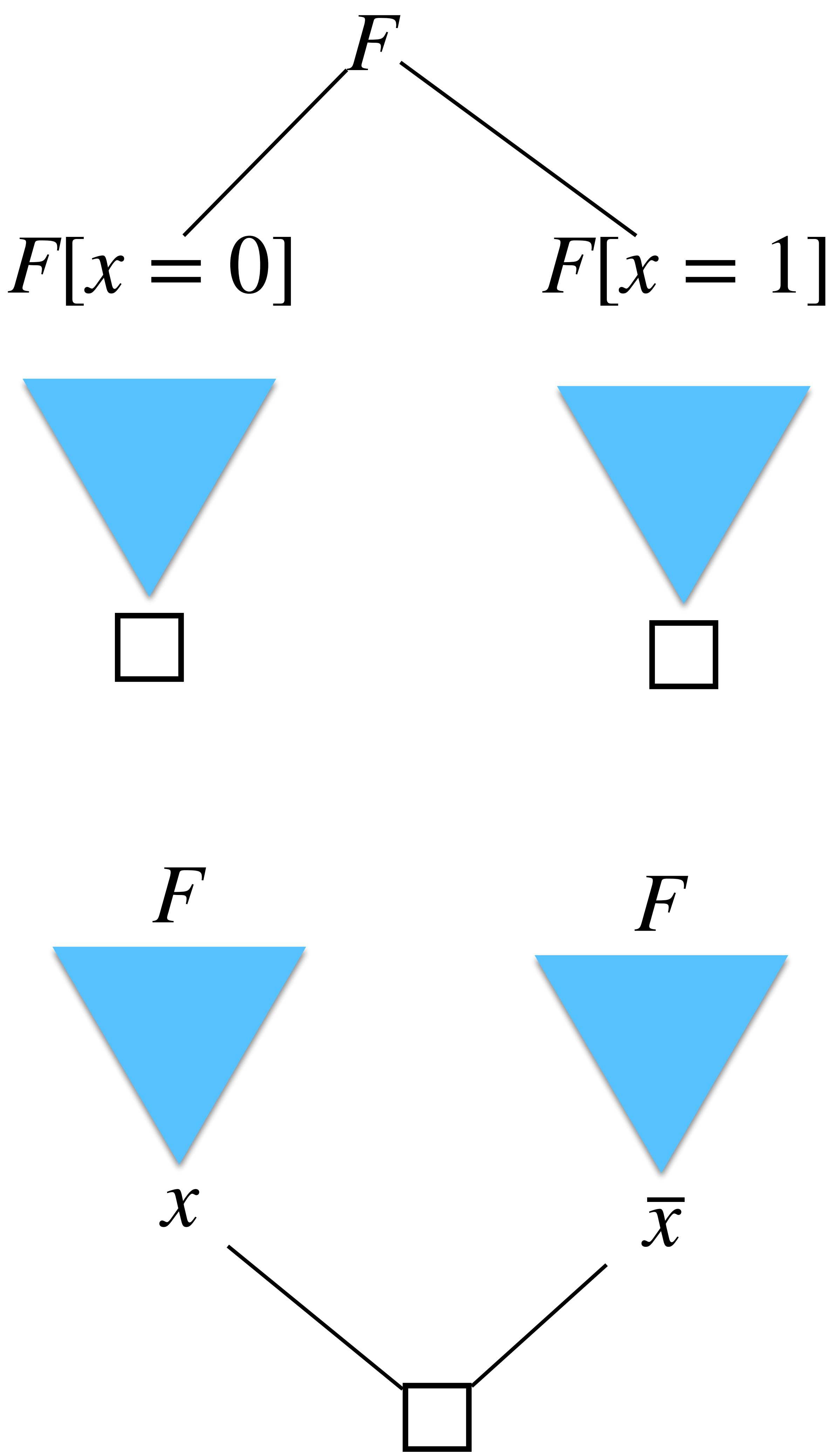
- Question: are there parameters that admit FPT SAT decision, but where not always an FPT-size resolution proof exists?
- Let's look at some of the parameters from above.

Treewidth

- primal-treewidth admits FPT-size resolution proofs
- incidence-pathwidth admits FPT-size resolution proofs [Imanishi, WALCOM 2017]
- incidence-treewidth admits XP-size resolution proofs (unknown whether FPT)
- incidence-treewidth after preprocessing admits FPT-size resolution proofs [Samer, Sz. JCSS 2010]

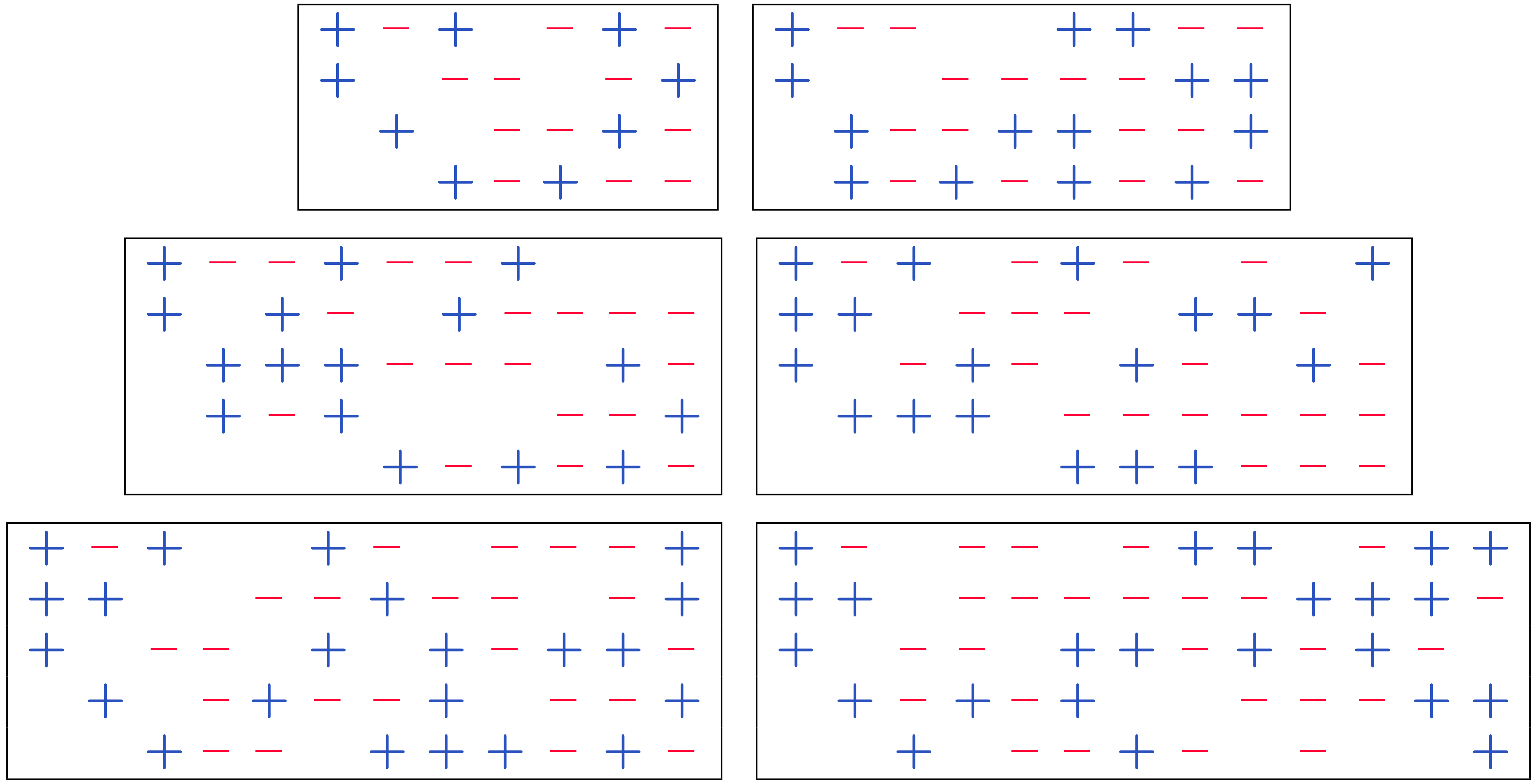
Backdoors

- If formulas in the base class C do have poly-size resolution proofs, then strong backdoor size into C admits FPT-size resolution proofs.
- This also holds for backdoor depth.
- Poly-size resolution proofs are known for Horn, 2CNF, QHorn
- Interesting open case: HIT



HIT and Resolution

- We can construct large HIT formulas from smaller ones, but the resulting formulas don't have a significantly larger resolution complexity.
- In fact, it is not known whether there exist infinitely many irreducible HIT formulas.
- [Peitl, Sz Arxiv 2022] conducted computer search for hard HIT formulas.



FLoC 2022

PC :
Mon Aug 1, 17:00

Handbook of Satisfiability, 2nd Edition

http://www.ac.tuwien.ac.at/files/tr/ac-tr-21-004.pdf

Extended and revised chapter 17
“Fixed-parameter Tractability”