The Parameterized Complexity of SAT

Stefan Szeider

TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

ALGORITHMS AND COMPLEXITY GROUP

Boolean Seminar
Liblice, Sept 2023

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).

20 years ago!

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

Purpose of Talk

- Briefly discuss background SAT theory vs SAT practice
- Outline the parameterized complexity framework
- Discuss ways of parametrising SAT (decompositions, backdoors, hybrid parameters)
- Focus on more recent progress (tww, bd depth, bd tw)
- Not a technical talk, rather give a general picture and discuss what questions can be asked

Propositional satisfiability (SAT)

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

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

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

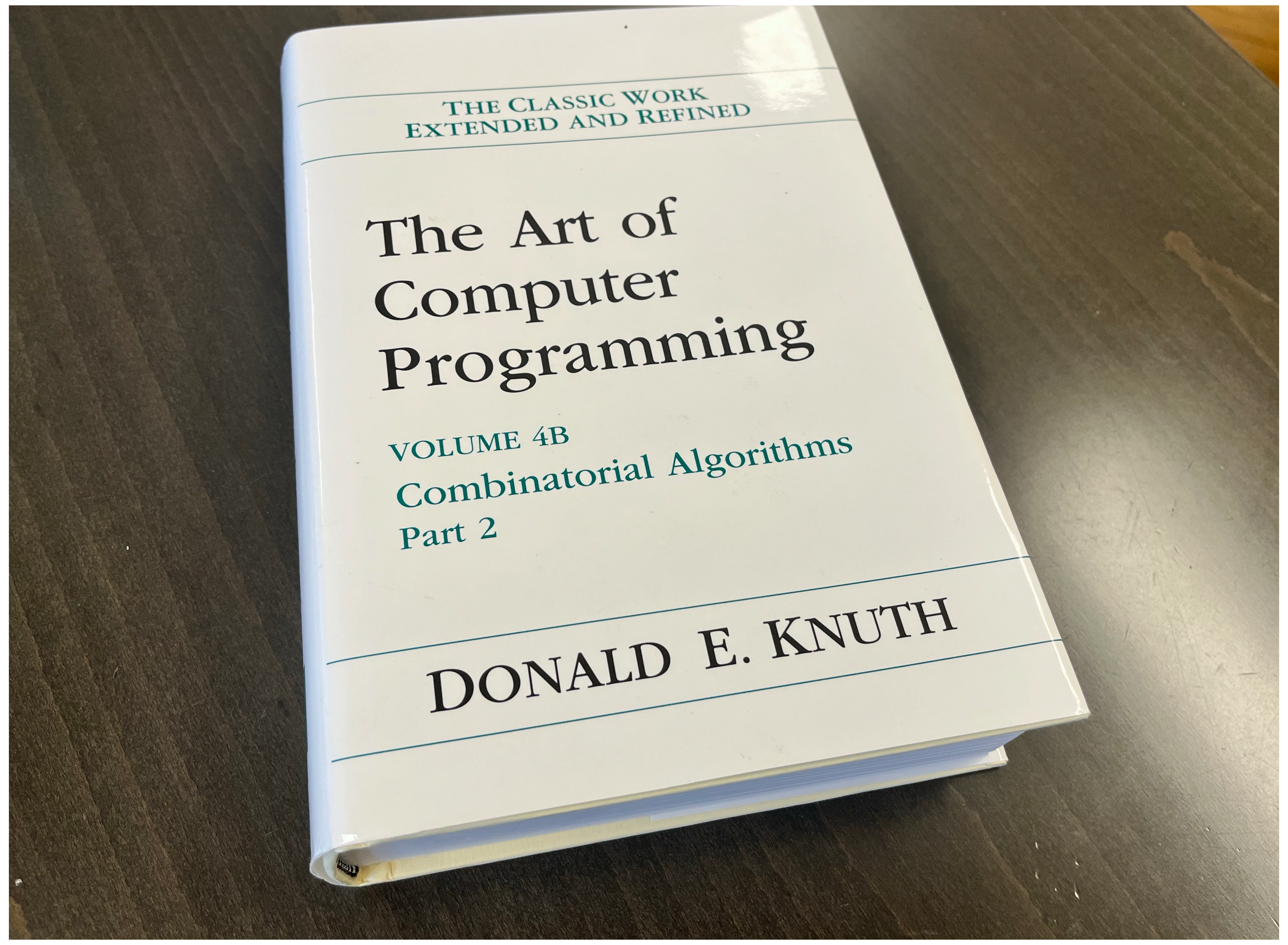
satisfied by setting $y = 1, u = 0, v = 1, x = 0$

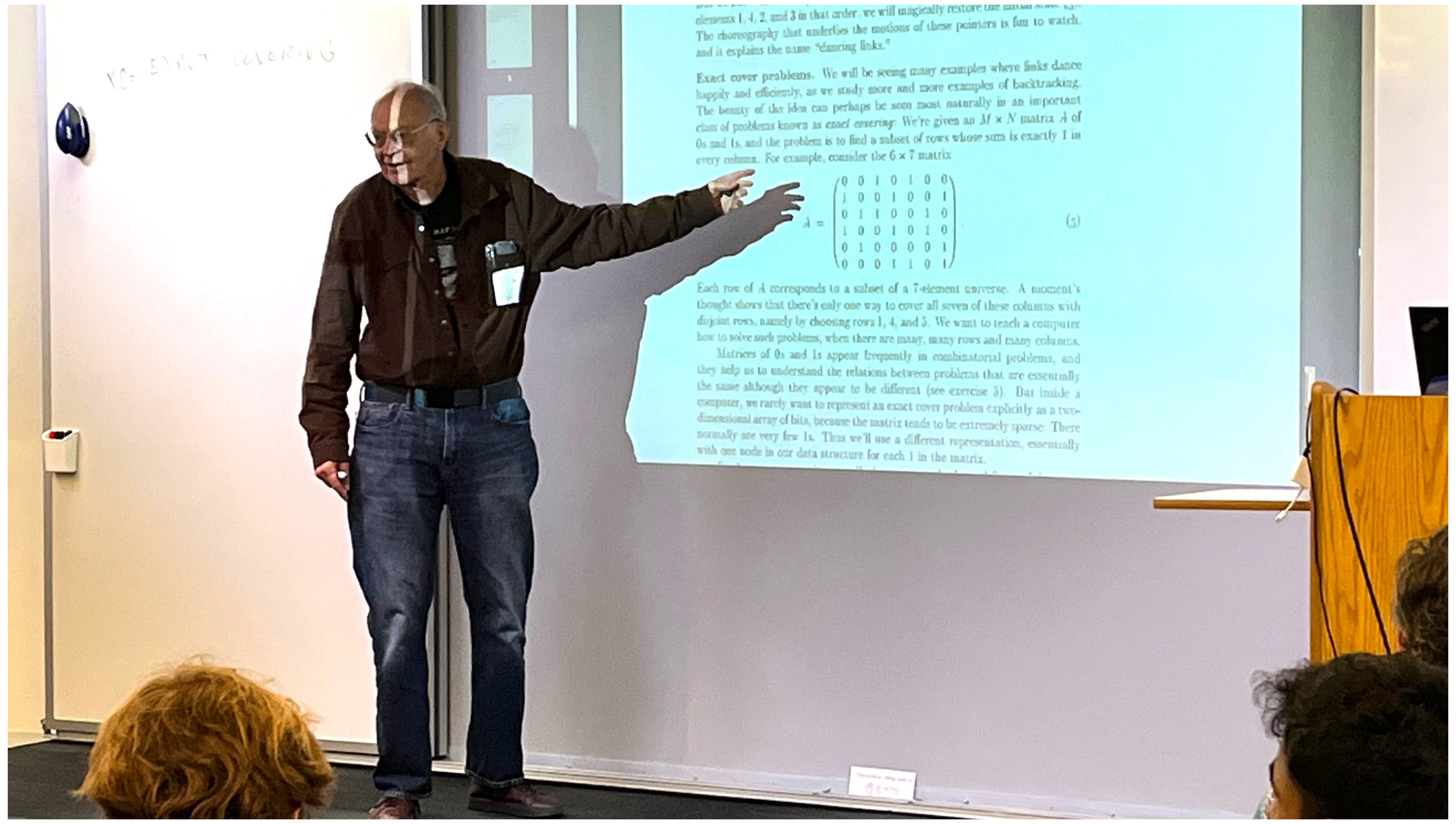
define literal, clause, occurrence, truth assignment, applying partial assignment $F[t]$

TAOCP

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

"The story of satisfiability is the tale of a triumph of software engineering, blended with rich doses of beautiful mathematics."






Berkeley, April 2023

Classical 3SAT Time Bounds—*SAT is hard*

2^n	trivial
1.3333^n	1999 (Schöning)
1.3302^n	2002
1.3290^n	2003
1.3280^n	2003
1.324^n	2010 (Hertli)

3SAT time bounds

Exponential Time Hypothesis (ETH)
[Impagliazzo, Paturi & Zane 2001]
no sub exponential time
algorithm for 3SAT

- For $n = 250$ that exceeds the number of nano seconds that passed since the big bang!

CDCL SAT solvers—*SAT is easy*

- Conflict-driven Clause Learning solvers emerged around the millennium
- Orders of magnitude faster than previous algorithms
- Today solve industrial instances with millions of variables and clauses routinely
- Continuous progress in **solving, encoding, certifying, quantifying, counting, optimizing**

COMMUNICATIONS OF THE ACM | JUNE 2023 | VOL. 66 | NO. 6

https://doi.org/10.1145/3560469

The Silent (R)evolution of SAT

Today's powerful, robust SAT solvers have become primary tools for solving hard computational problems.

BY JOHANNES K. FICHTE, DANIEL LE BERRE, MARKUS HECHER, AND STEFAN SZEIDER

- 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

new computer
old algorithm

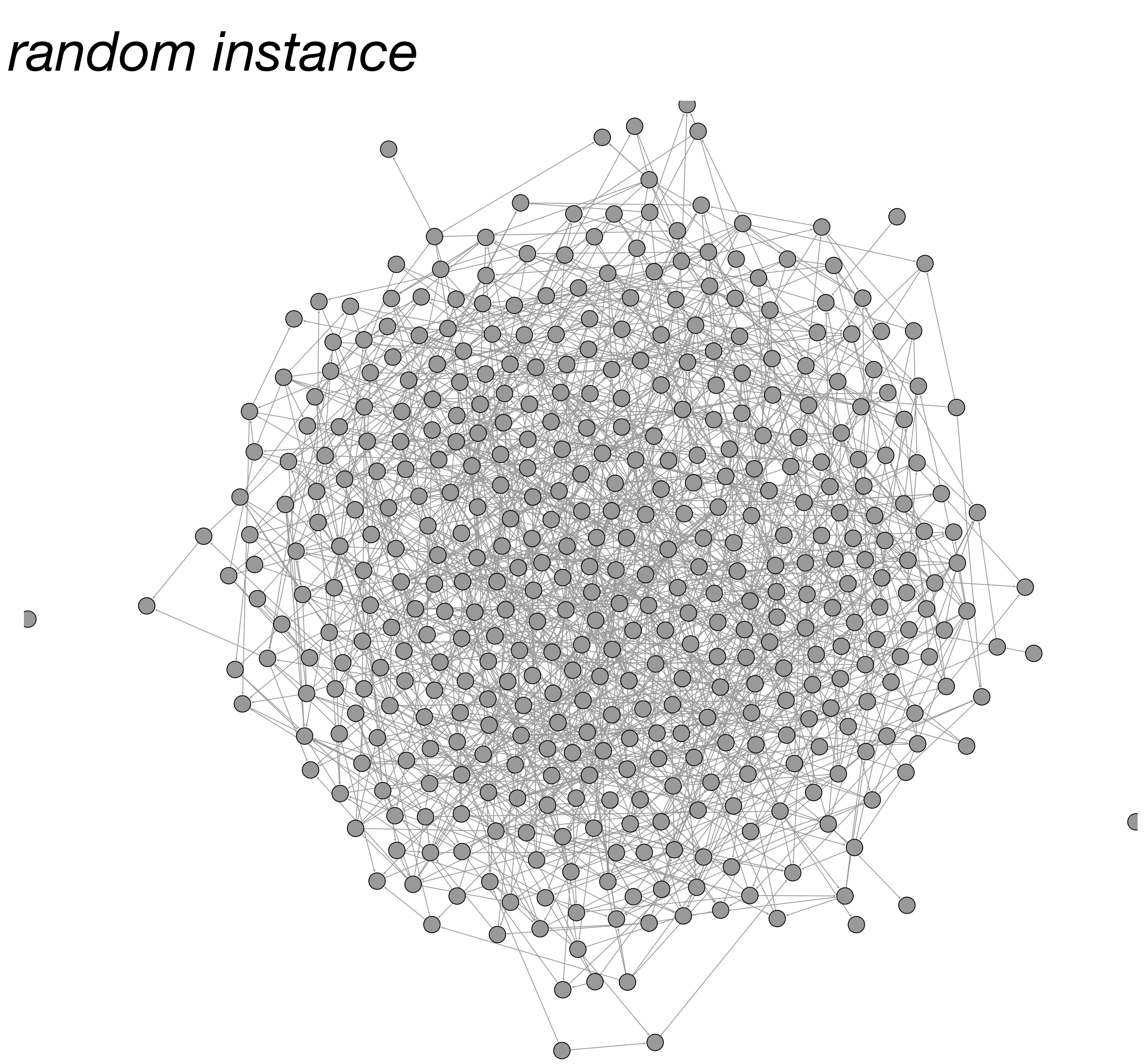
old computer
new algorithm

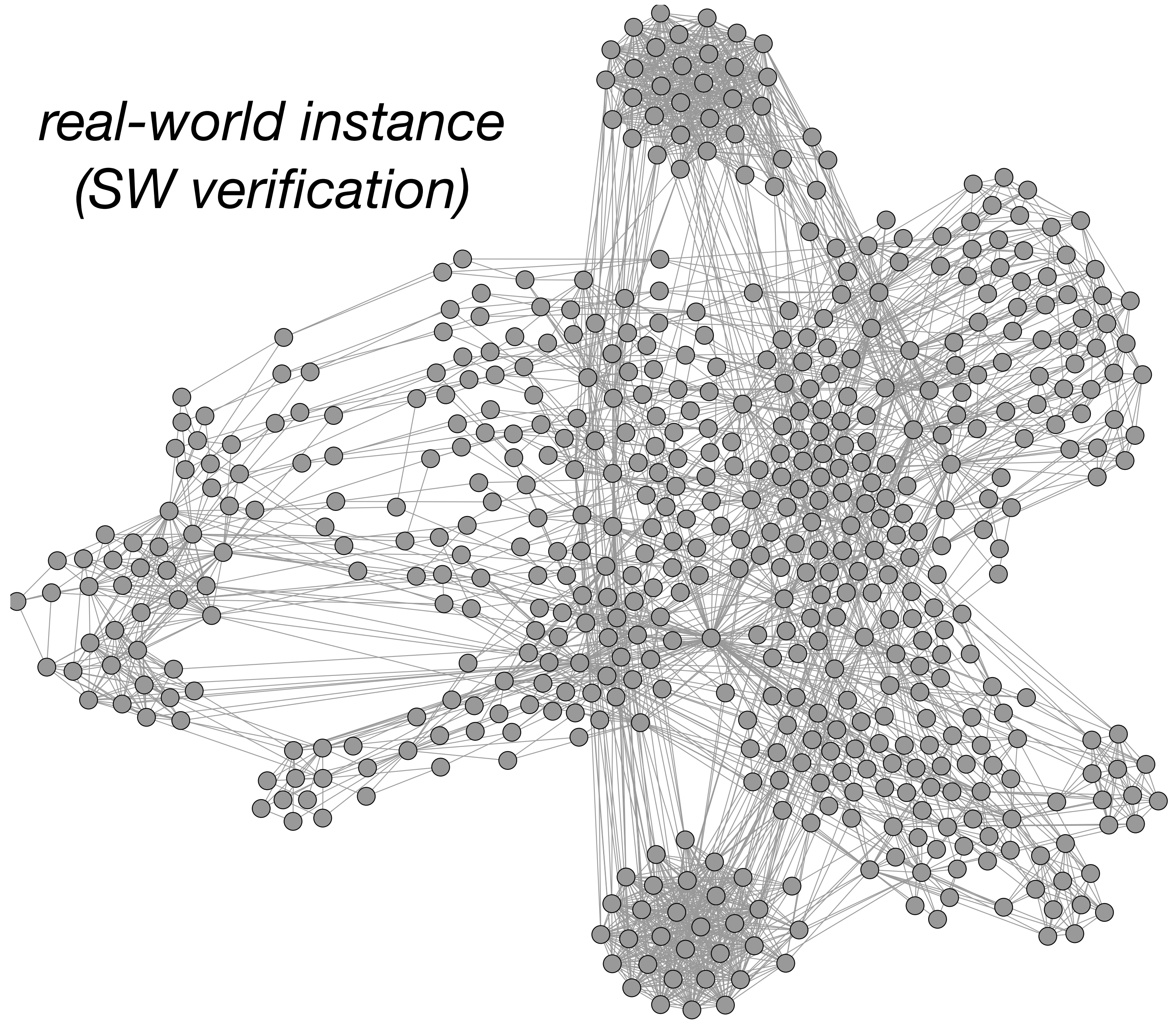
	Grasp (1996)	zChaff (2001)	siege_v3 (2003)	Glucose (2016)	CaDiCal (2019)	Maple (2019)
					Team SW	
old HW (1999)	73	48	37	106	98	77
		Team HW				
new HW (2019)	76	71	93	188	190	195

[Fichte, Hecher, Sz. CP 2020]

update arXiv:2008.02215v2

How to resolve the mystery easy vs hard?





- Structure matters!
- How to capture structure?

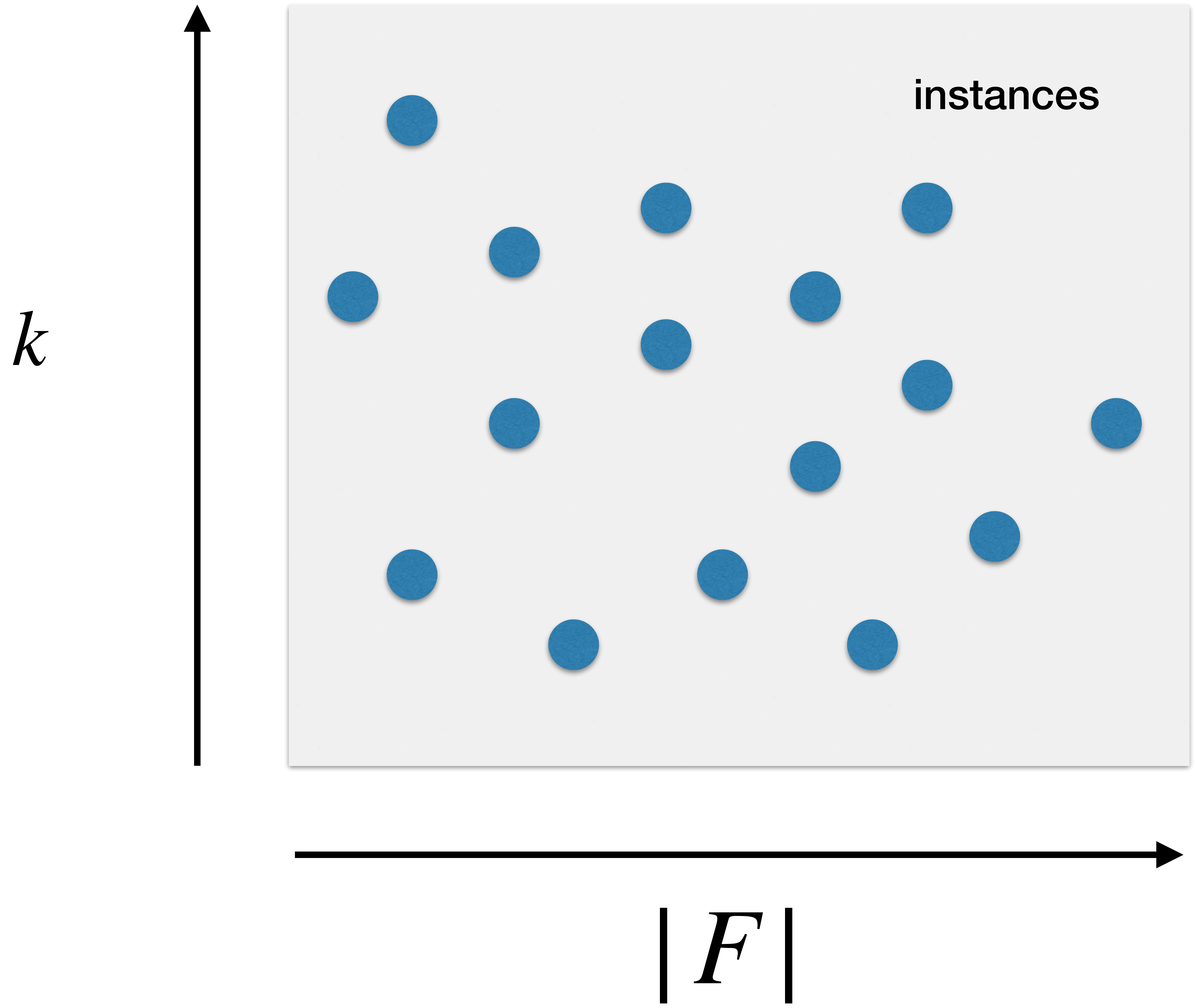
Correlational Approach

- **Try to capture structure in a way that statistically correlates with CDCL-solving time**
- community structure, modularity, centrality,..
- In general, industrial formulas have an 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]
- No performance guarantee! [Ganian, Sz, AIJ 2021]

Causational Approach

- **Try to capture structure in a way that provides worst-case performance guarantees for SAT algorithms**
- Classical results: polynomial classes like Horn, 2CNF, etc.
- Gradual dependency on how “well structured” an instance is

Framework for Rigorous Models



- runtime guarantee should depend on k and $|F|$

… but how?

First try: XP

$$|F|^{f(k)}$$

- **if k is a constant, then the runtime is polynomial**
- this doesn't scale well in k
- such runtime guarantees are called **XP**

Second Try: FPT

$$f(k) \cdot |F|^{O(1)}$$

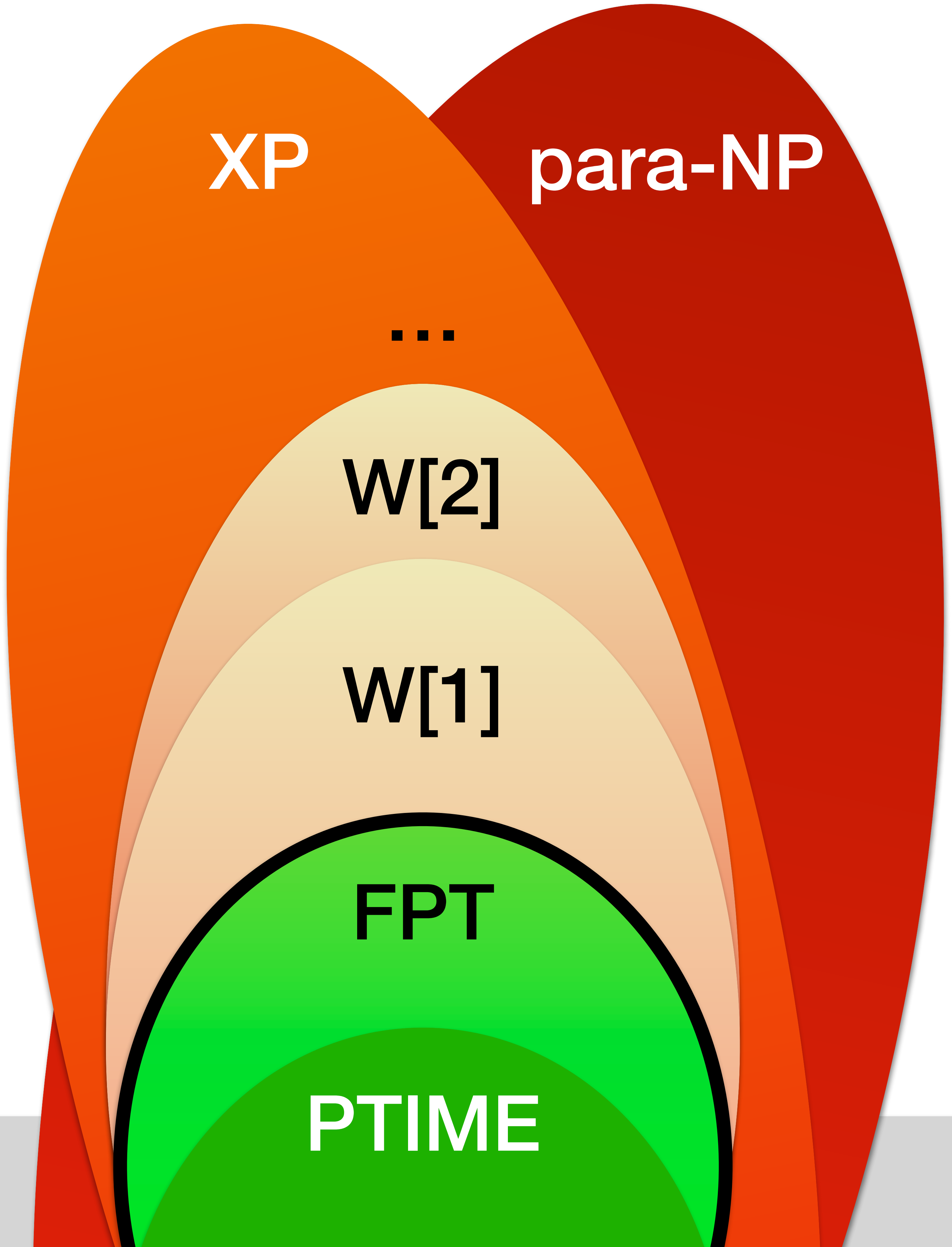
- **parameter k contributes a constant factor to the polynomial runtime, without changing the order of the polynomial**
- allows a better scaling in k
- such runtime guarantees are called **FPT** or **fixed-parameter tractable**
- well-developed area of TCS

Rich Theory

1999 2006 2006 2013 2013

Hardness Theory

- For showing that a problems is not FPT (conditionally)



FPT-SAT

Parameterized Complexity

- For the causal approach, parameterized complexity provides and ideal framework

- We can develop different parameters that capture different properties of SAT instances

- Compare parameters by their generality

FPT-SAT

“permissive” or “robust” approach

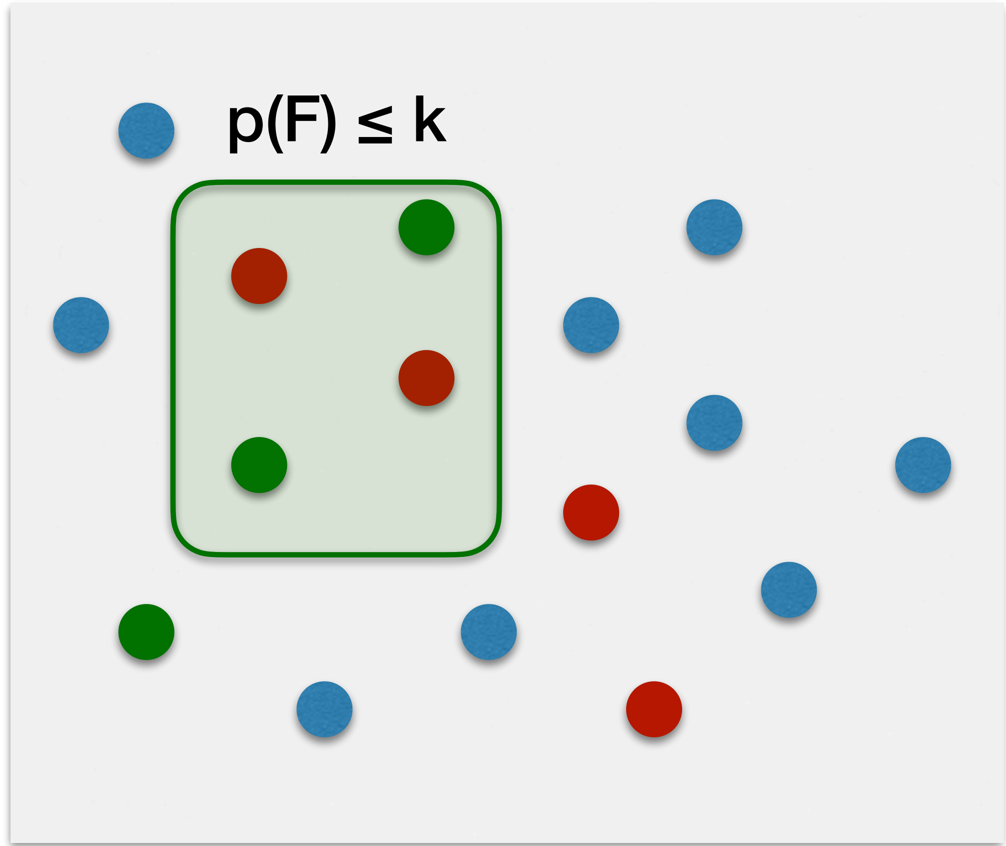
F,k

A

“SAT”

“UNSAT”

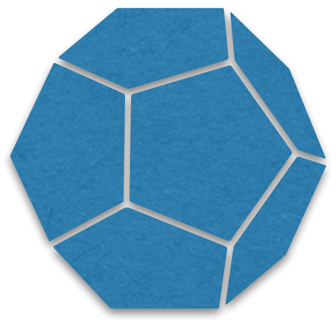
“p(F)>k”



two-phases approach

F,k

Ver



SAT

“SAT”

“UNSAT”

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

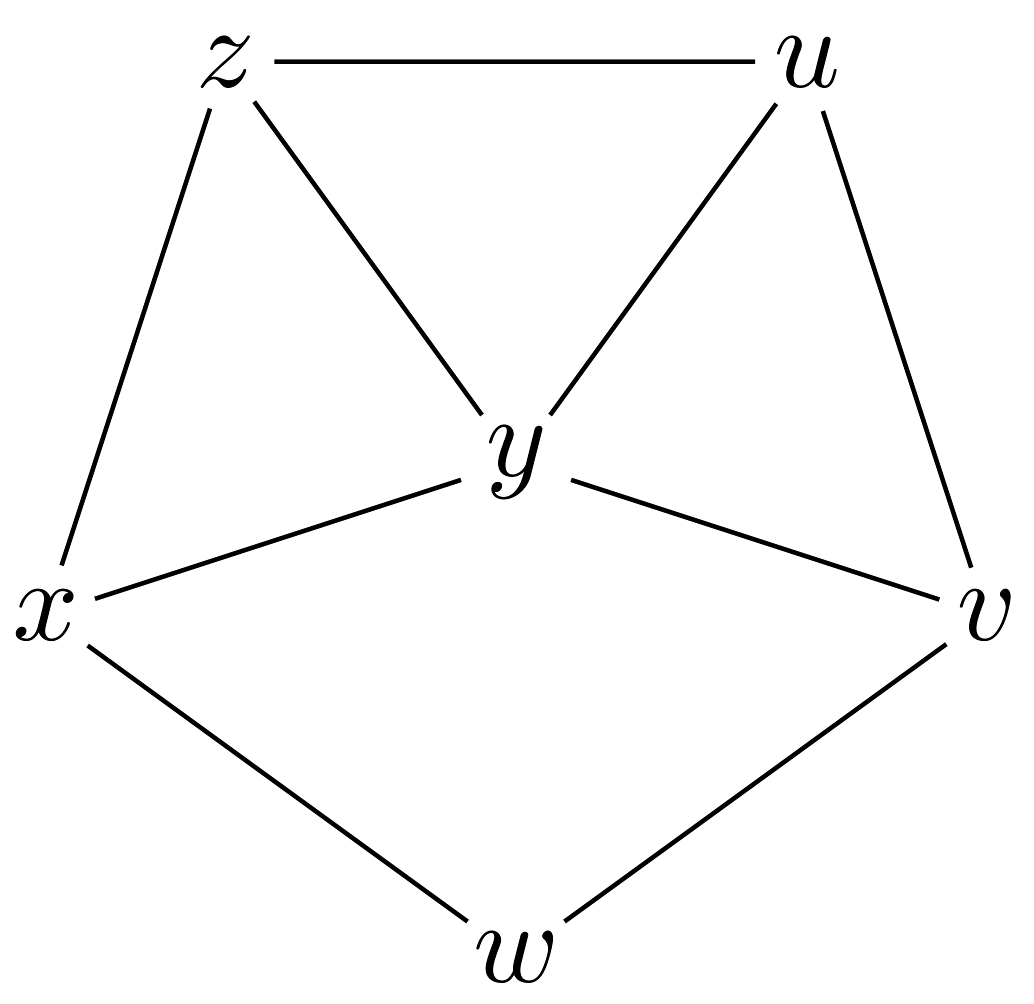
3) Hybrid Parameters

1) Graphical Structure

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

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

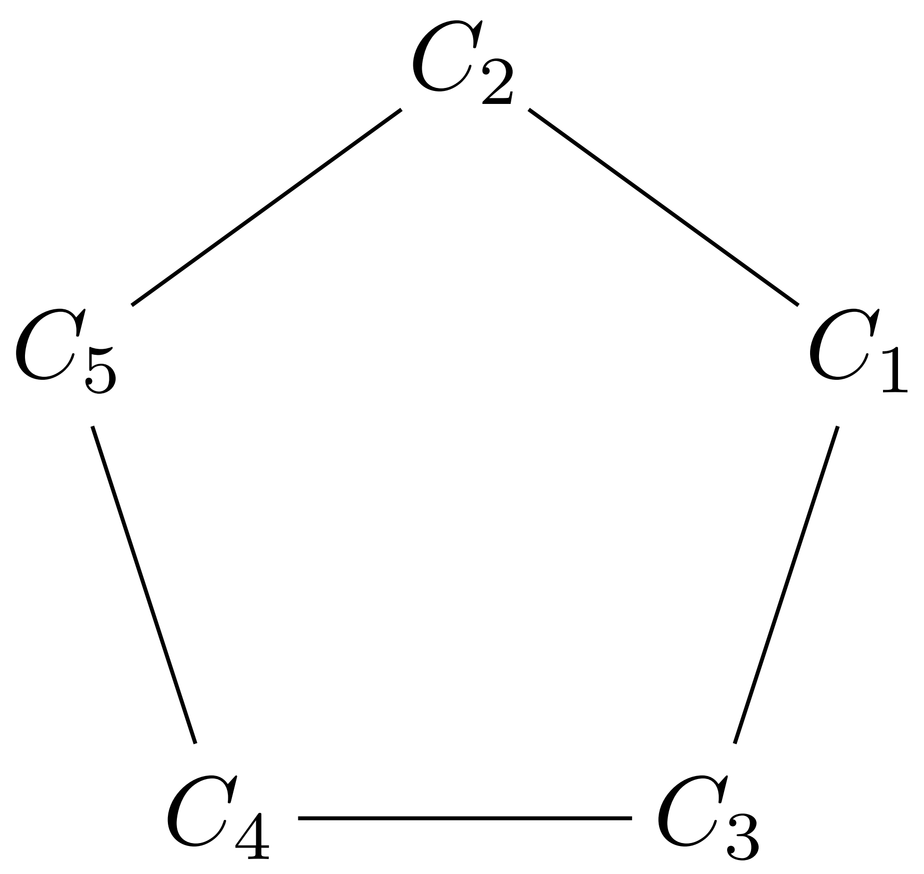
primal graph **a.k.a. VIG**



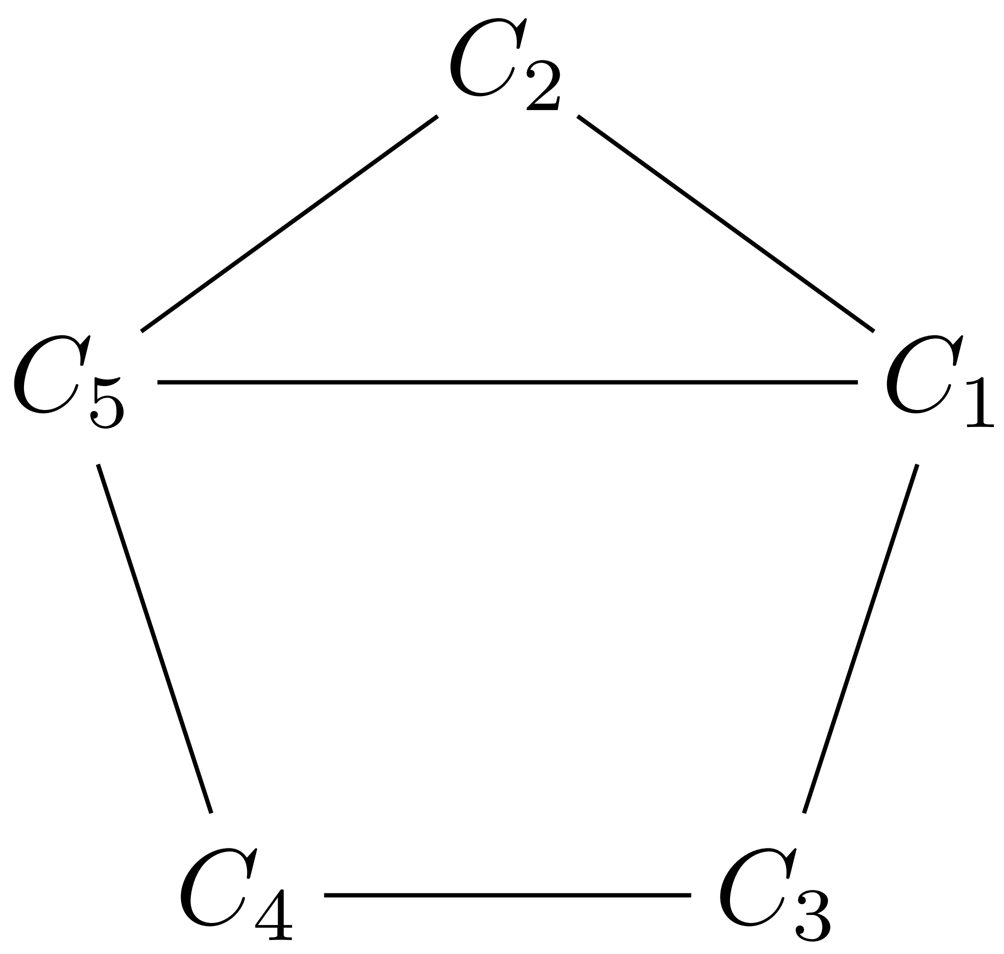
Graphs for $F = \{C_1, \ldots, C_5\}$

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

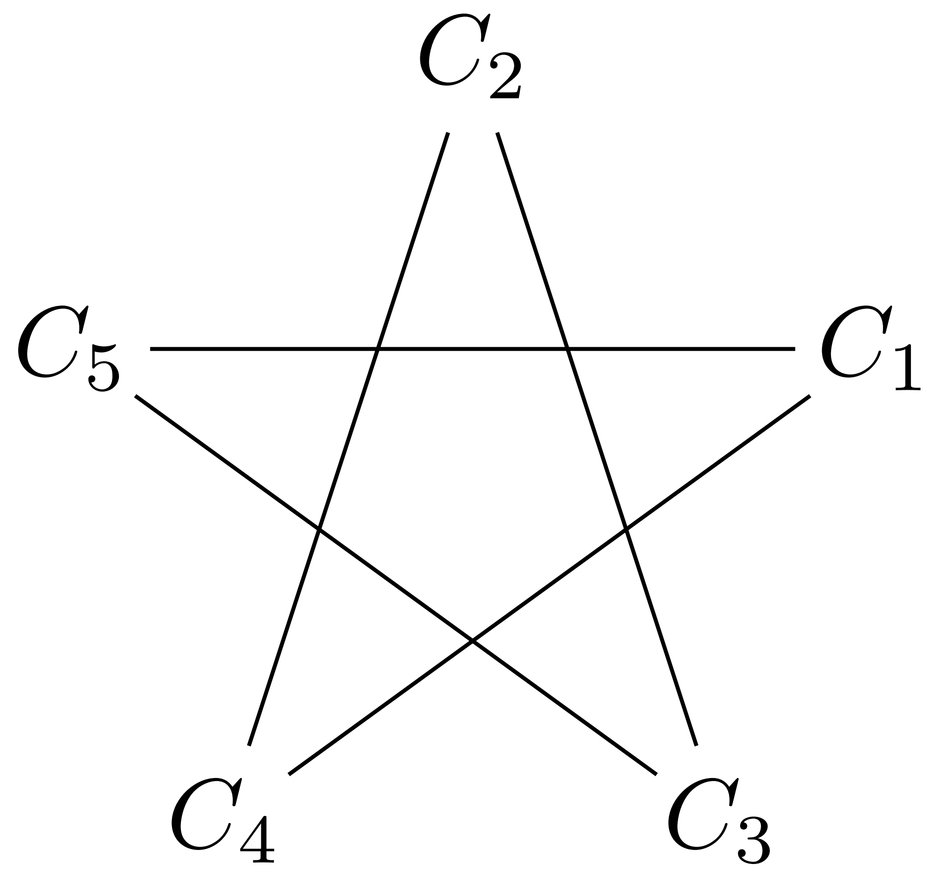
conflict graph



dual graph



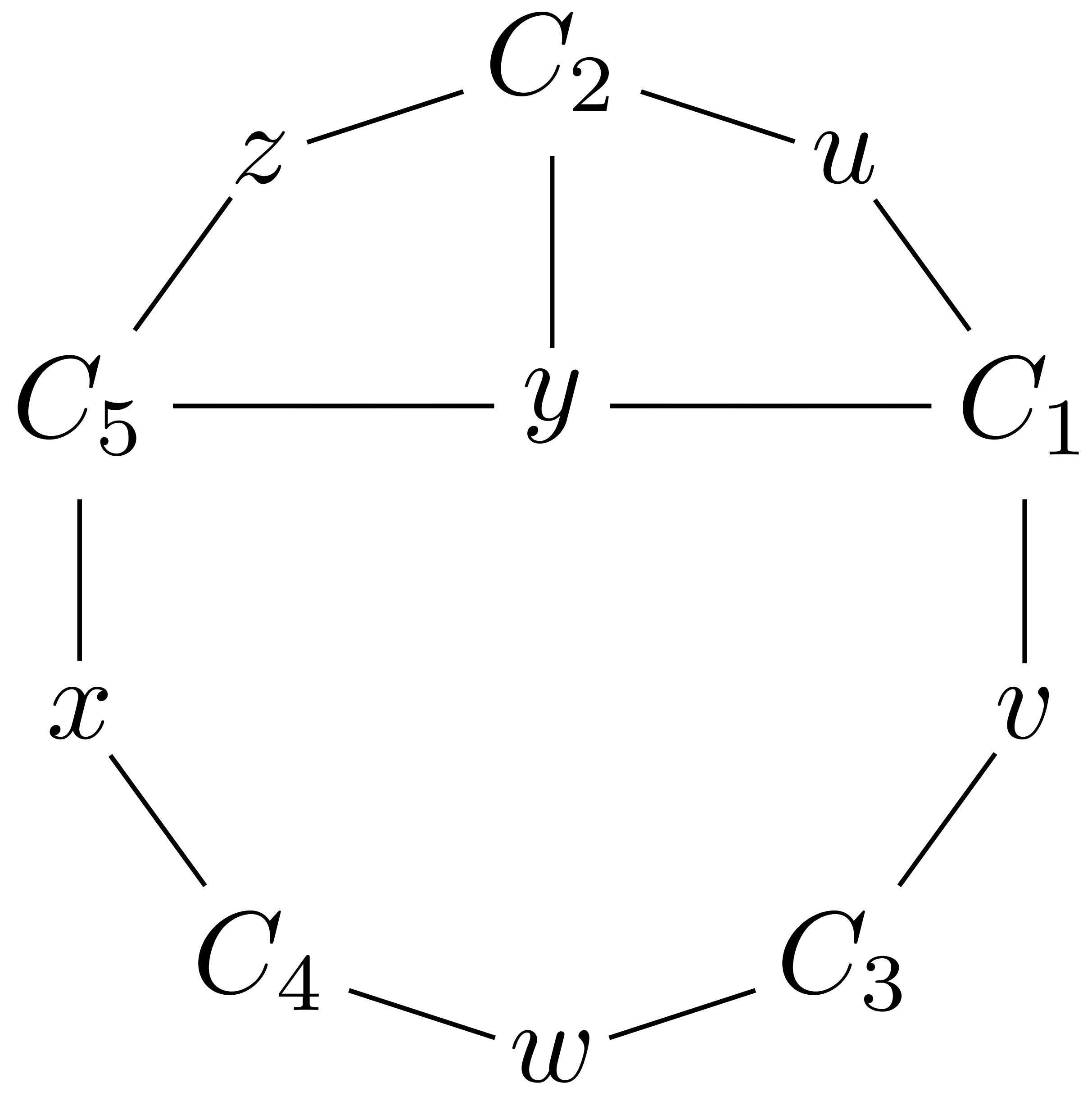
consensus graph



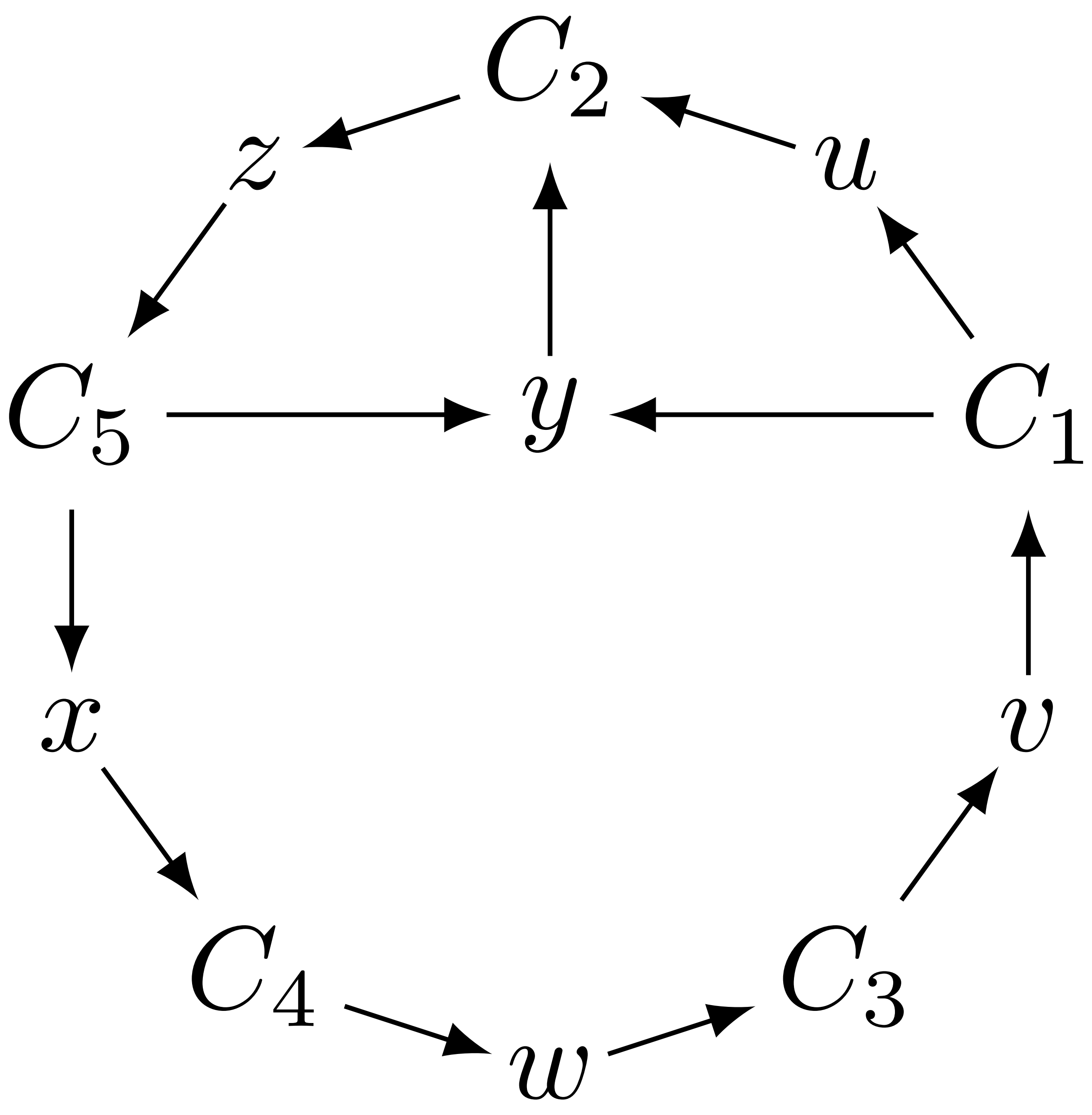
Graphs for $F = \{C_1, \ldots, C_5\}$

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

a.k.a. CVIG   incidence graph

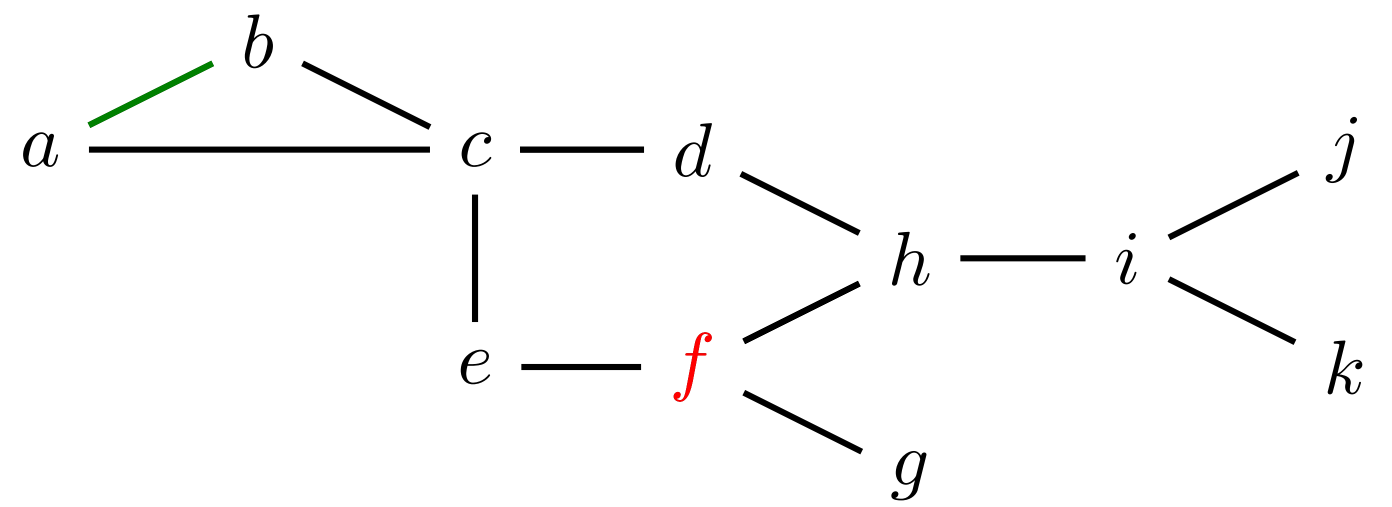


directed incidence graph
or signed incidence graph

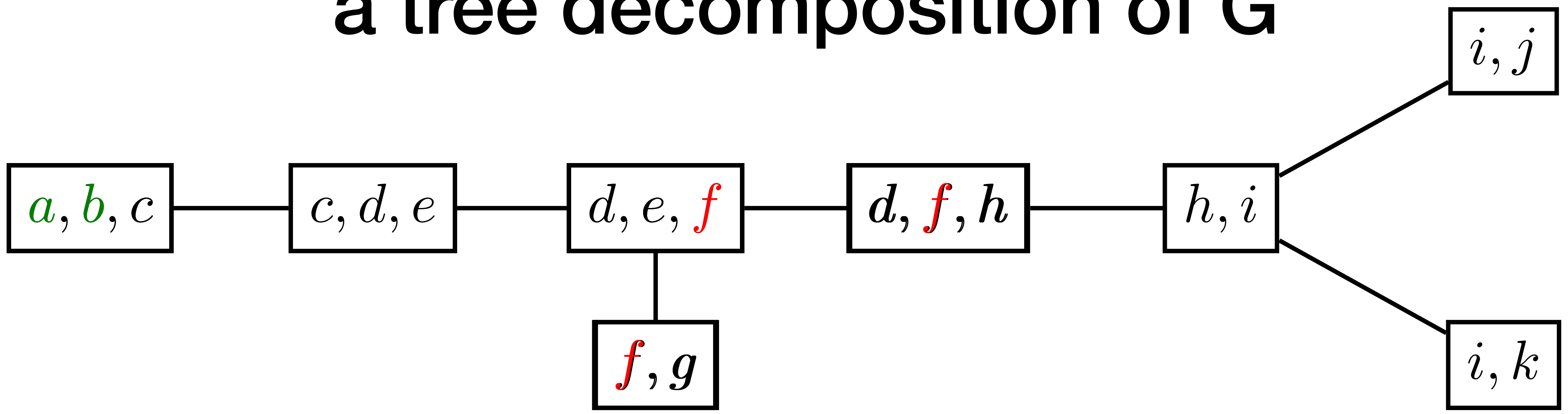


Graph Decompositions and Width Parms

a graph G



a tree decomposition of G

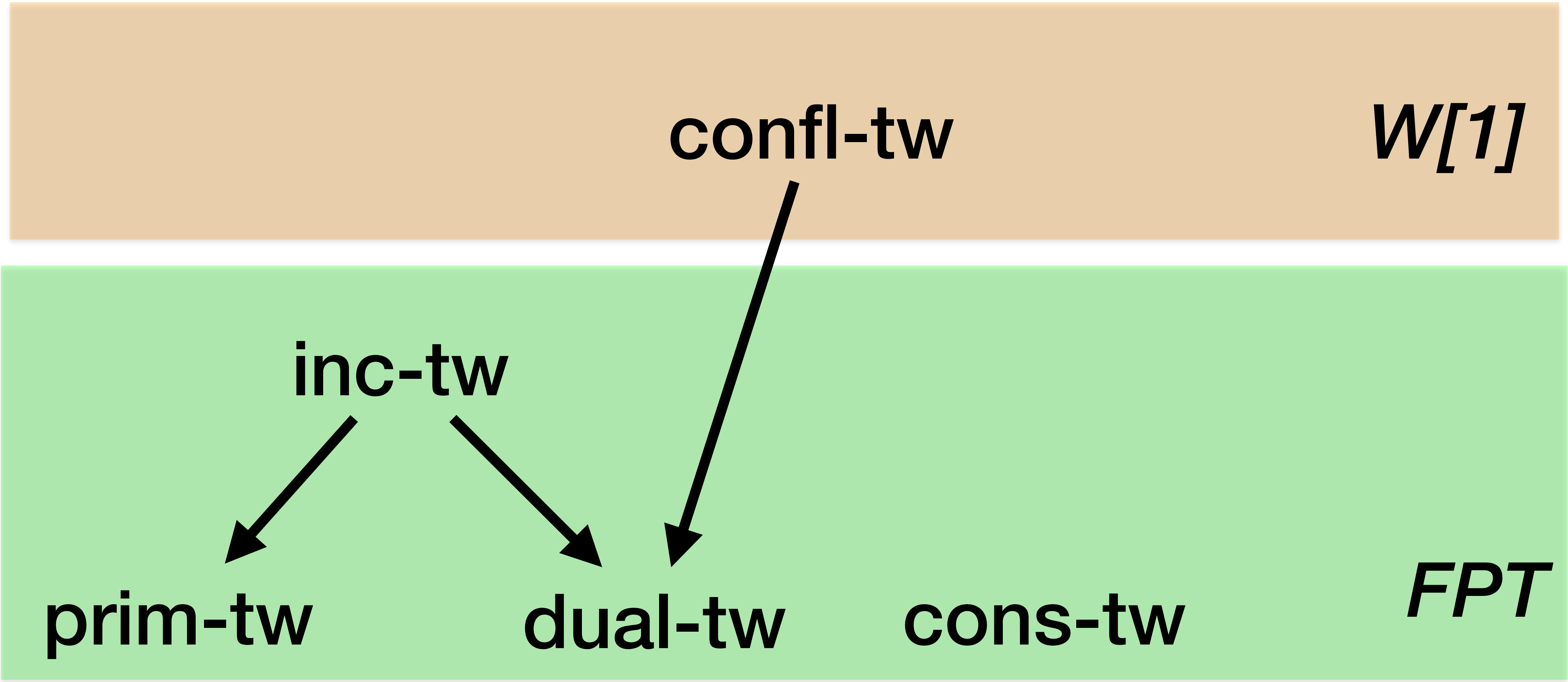


width = size of largest bag -1

- $\mathrm{tw}(G) =$ min width over all its tree decompositions
- checking $\mathrm{tw}(G) \leq k$ is FPT

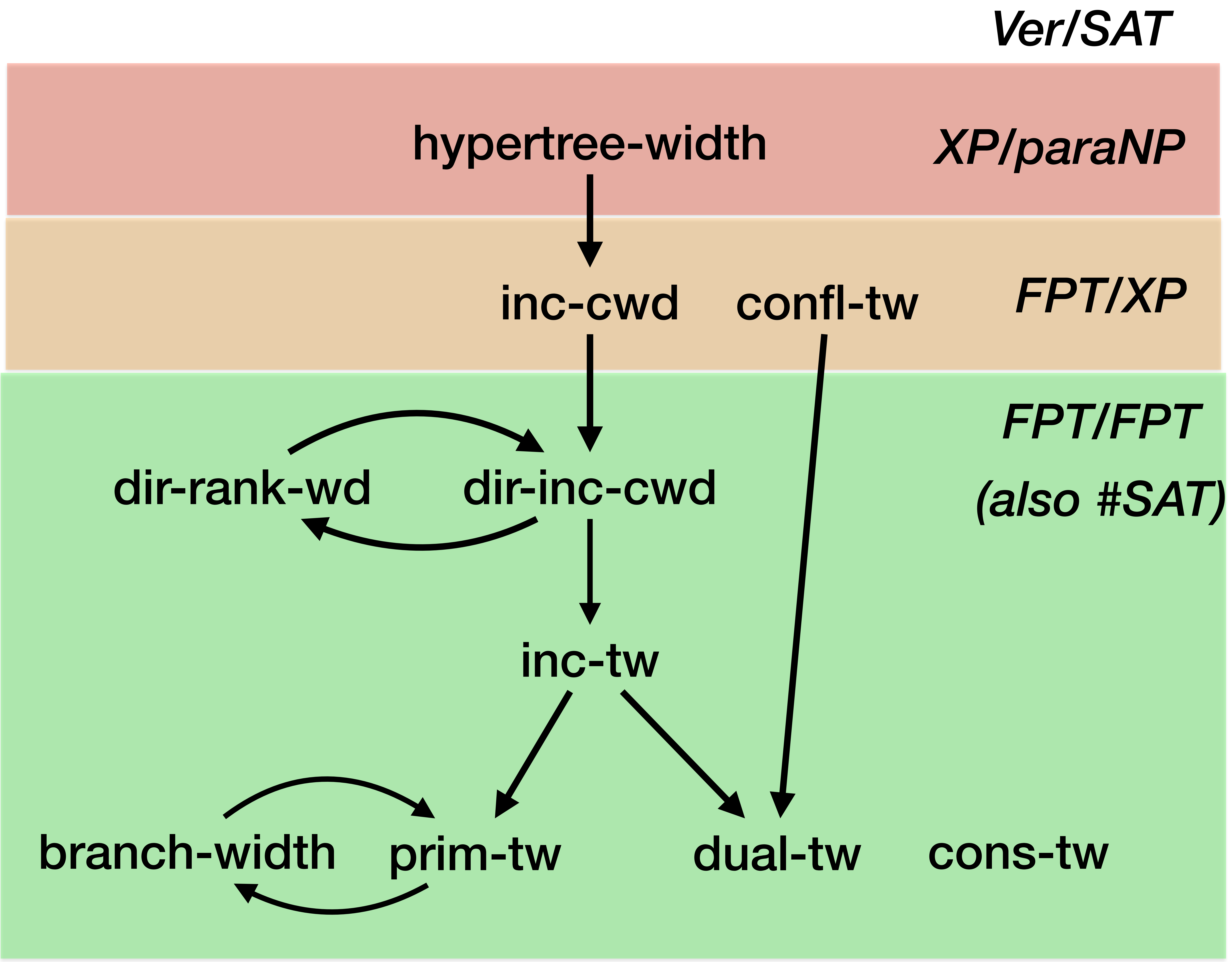
Treewidth of Formulas

- 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.

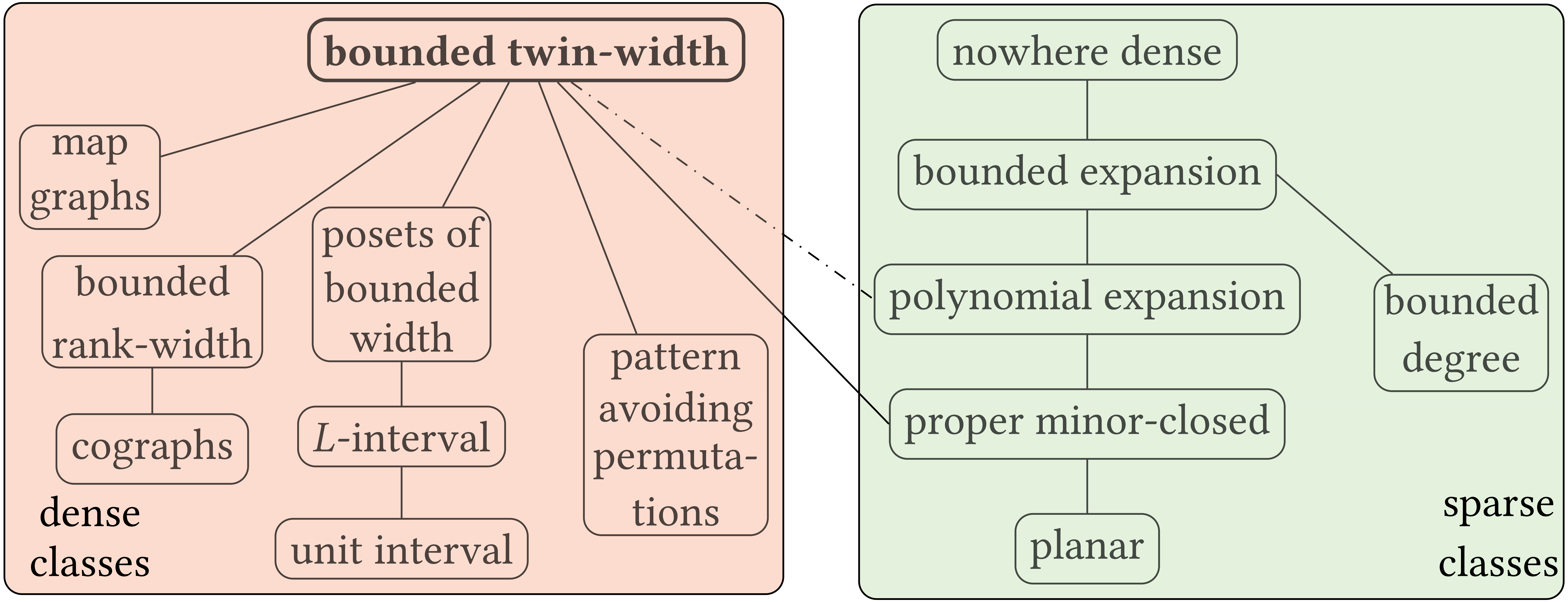


Improvement of $O^*(4^k) \Rightarrow O^*(2^k)$ for inc-tw using covering products [Slivovsky, Sz. SAT 2020]

Width Parameter Zoo



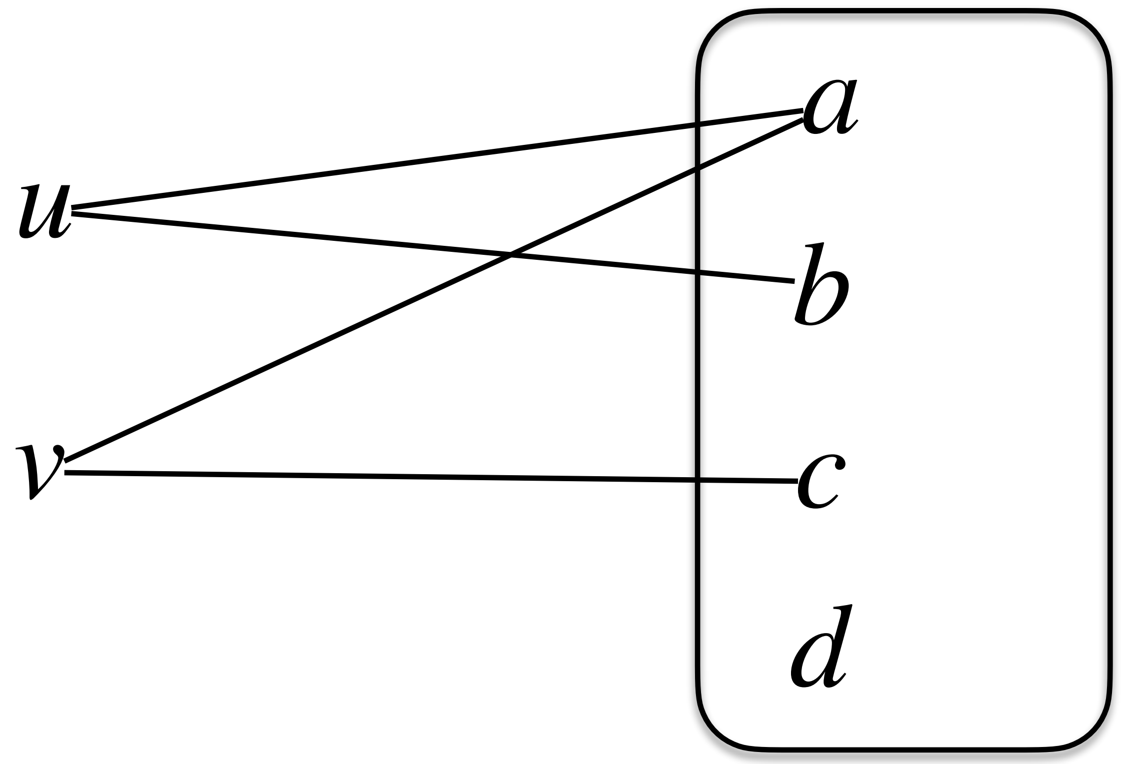
Twin-width (tww)



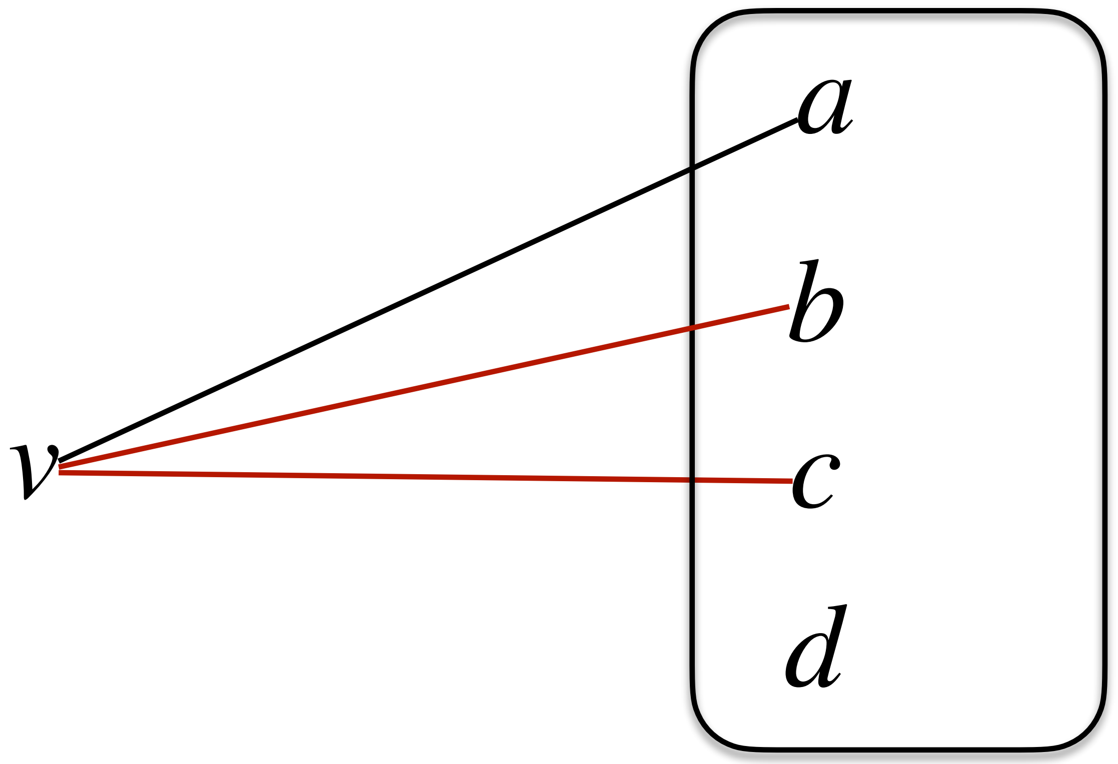
[Bonnet et al. JACM 2022]

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



$$u \rightsquigarrow v$$

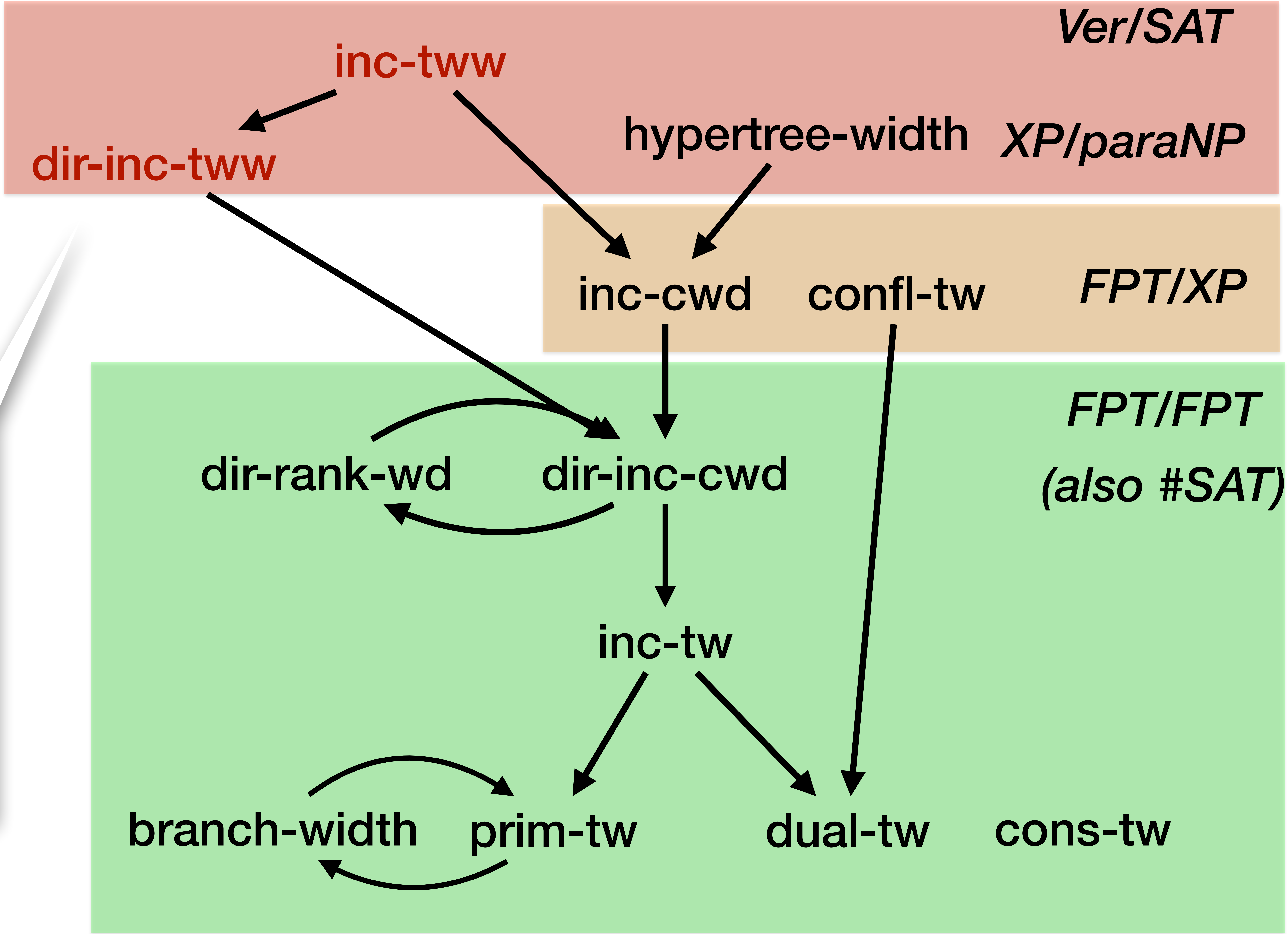


Twin-width of Graphs

- 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.

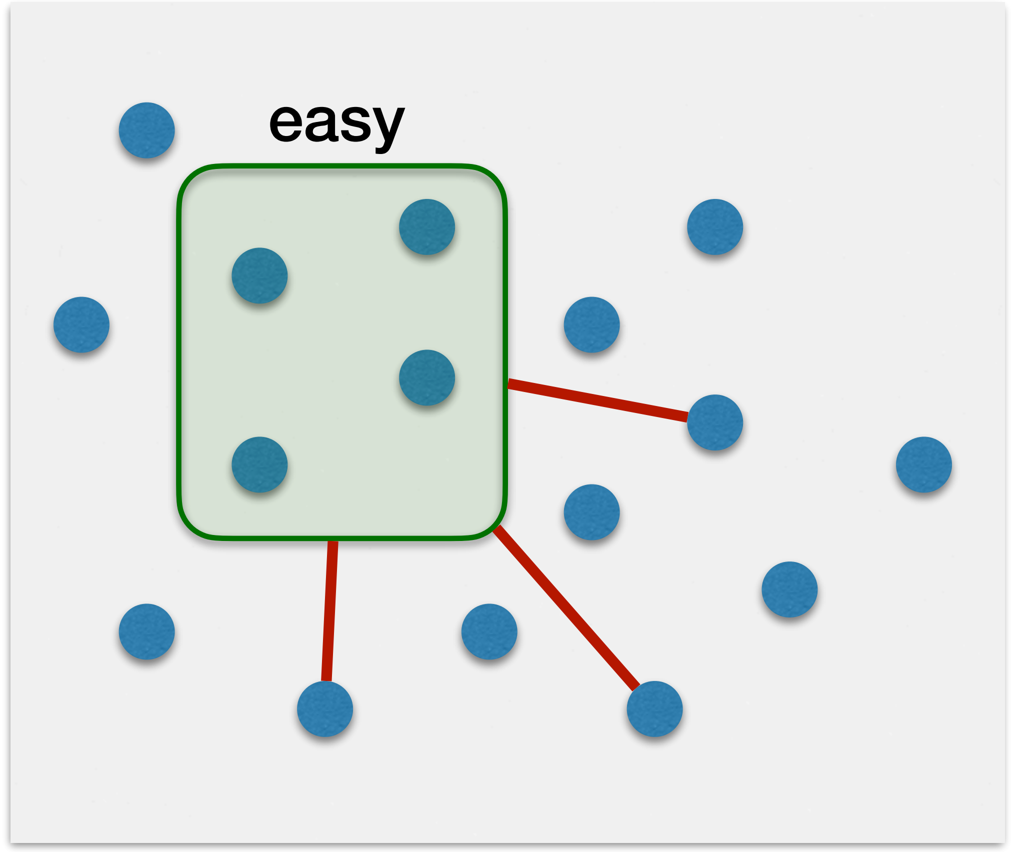
TWW in relationship to other parameters

If we add as additional parameter the number of variables set to true, then even #SAT becomes FPT [Ganian, et al. SAT 2022]



2) Syntactic Structure

Tractable Classes or Islands of Tractability



Parameterized by the **distance to a class**

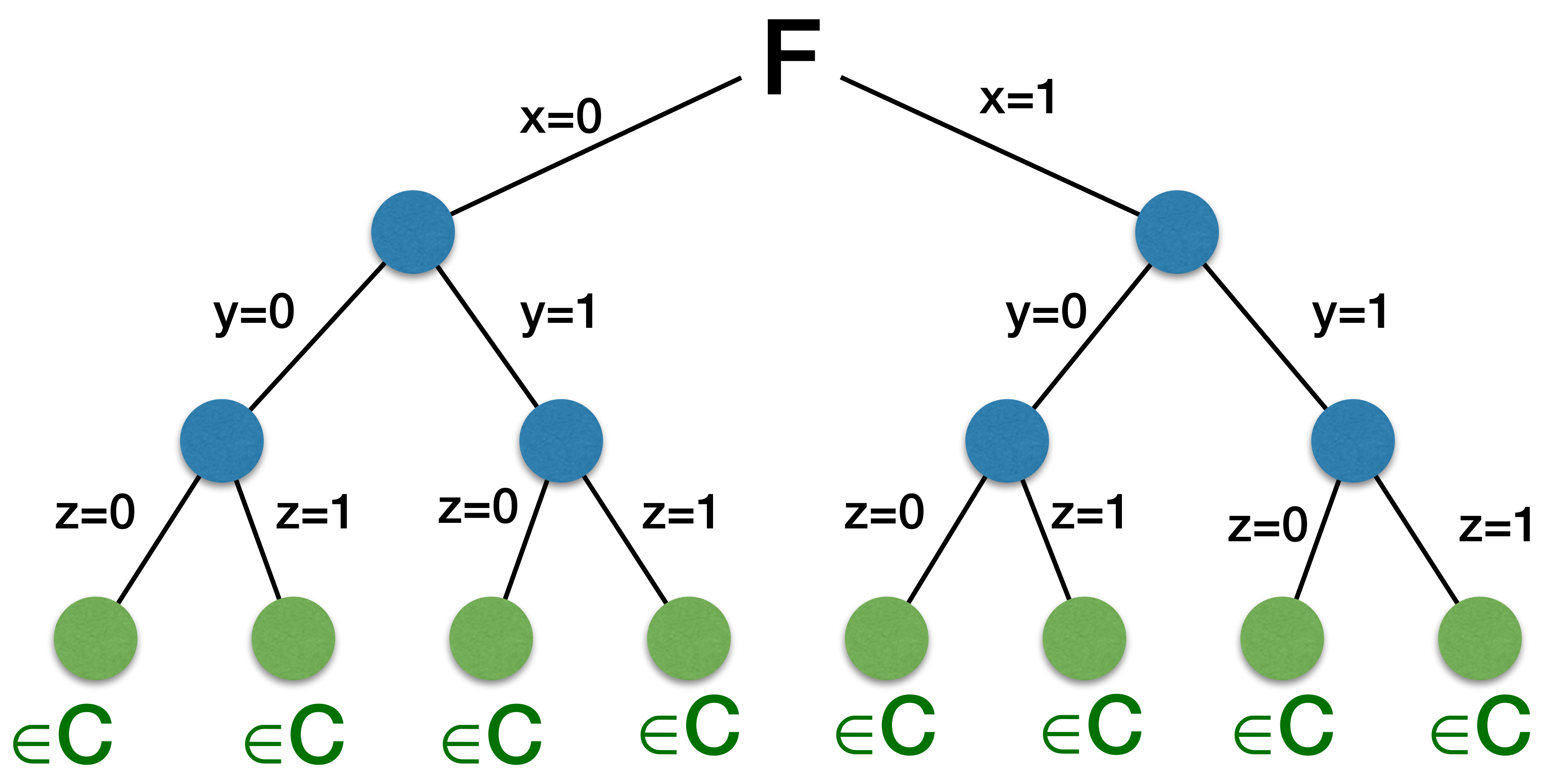
where the class is syntactical defined

strong

Distance = size of smallest backdoor set

- Fix a base class C (e.g., Horn)
- **B is a strong C-backdoor of F** if for all assignments $t : B \to \{0,1\}$ we have $F[t] \in$ 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.

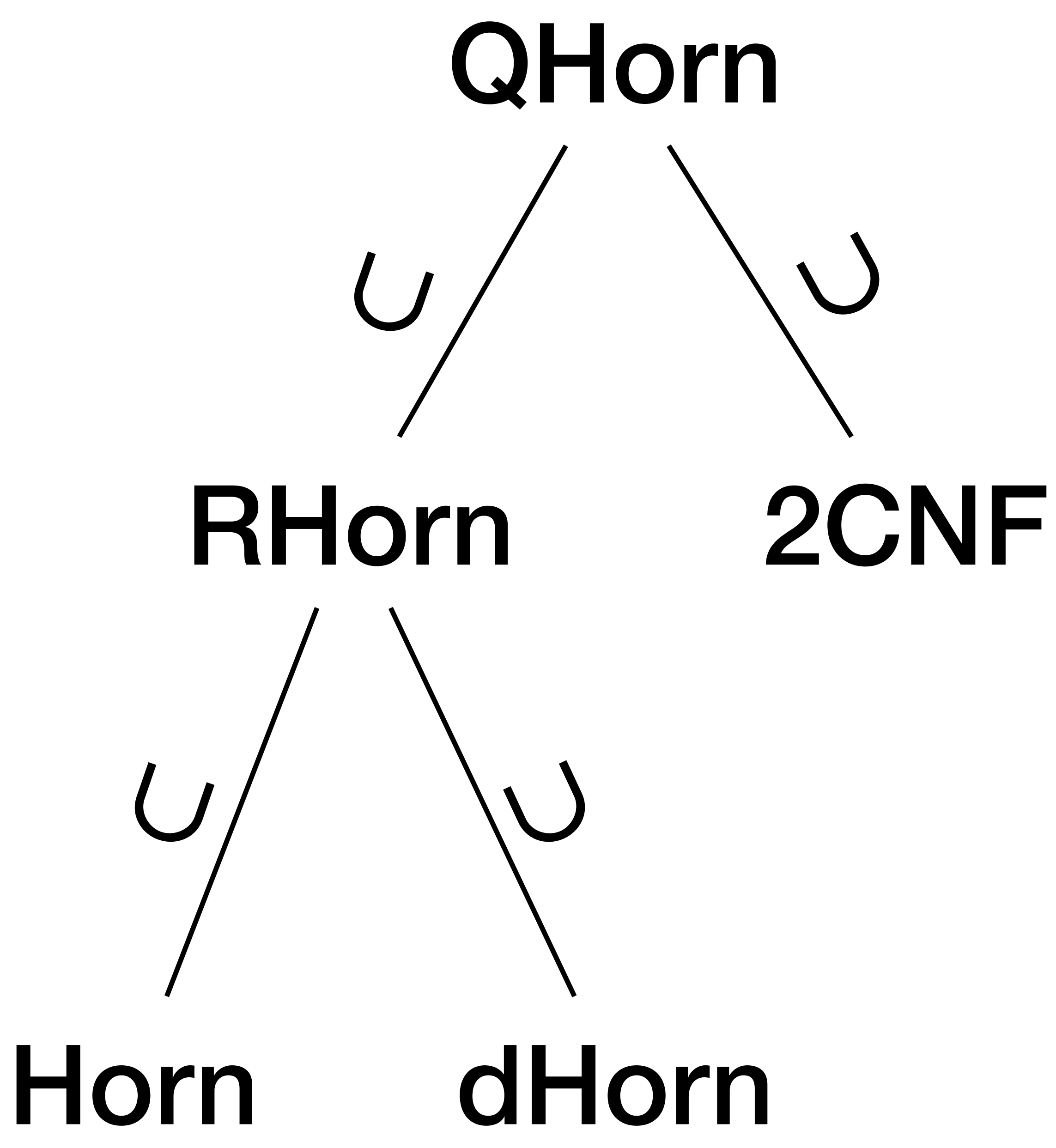
$B = \{x, y, z\}$



2^k

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.

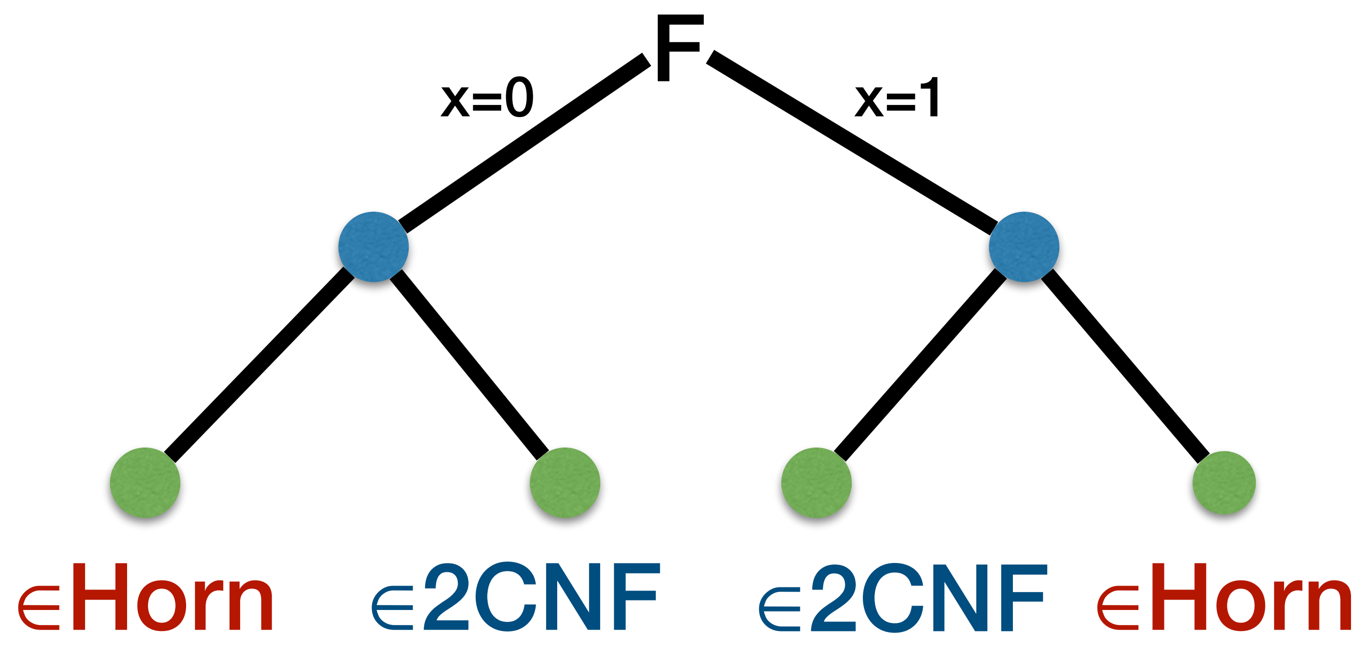


Other base classes

- **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

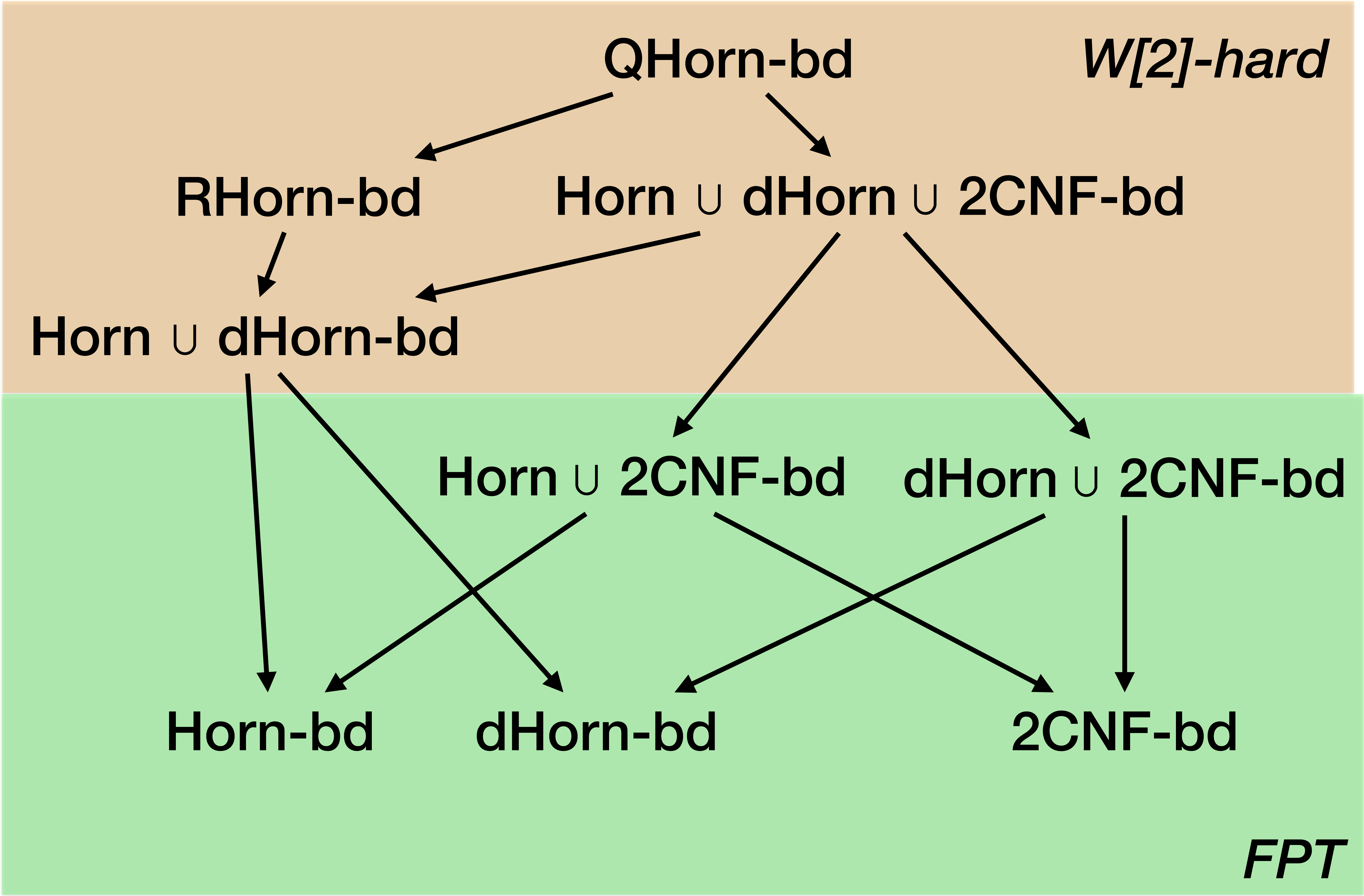
A hitting formula is unsatisfiable if

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



heterogeneous base classes

Backdoor Parameter Zoo



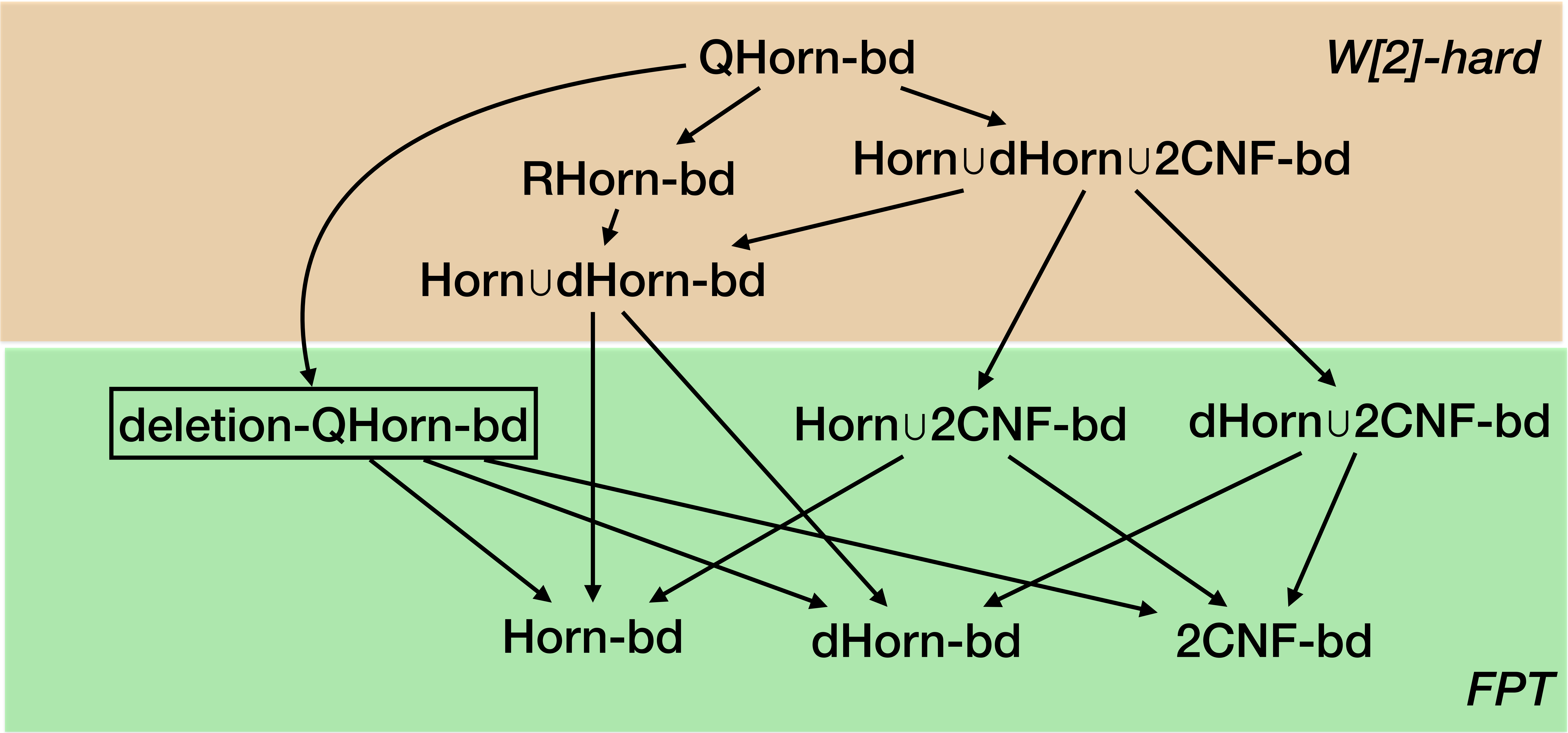
Deletion backdoor sets

- 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 $F - B$).

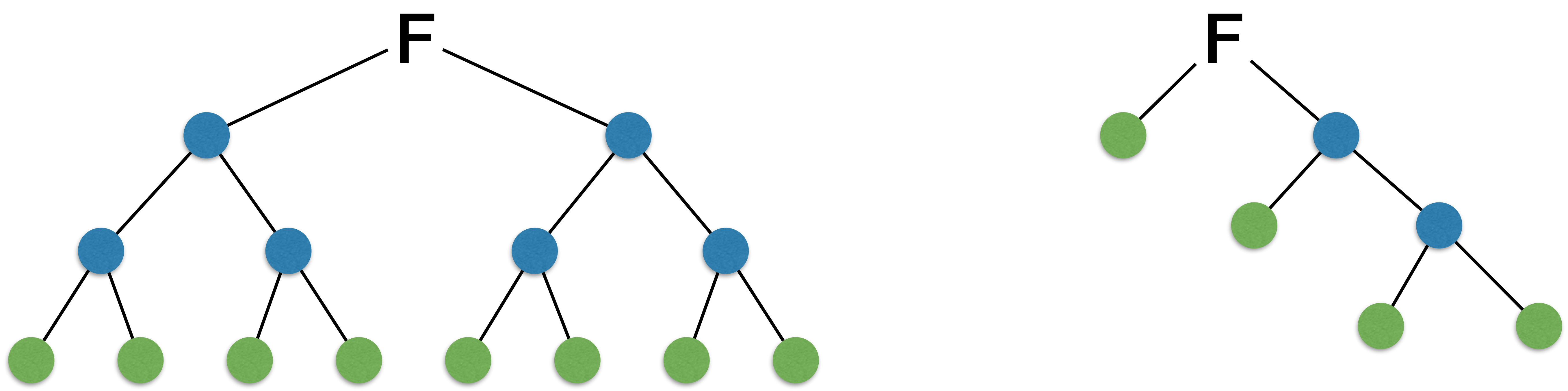
- Fact: if C is **clause-induced** ($F' \subseteq F, F \in C \Rightarrow F' \in C$) then each deletion backdoor set is also a backdoor set (but not necessarily the other way around).

Deletion Backdoor Sets



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

Avoid the 2^k assignments: Backdoor Trees



2^k

$k+1$

size of backdoor tree = number of leaves

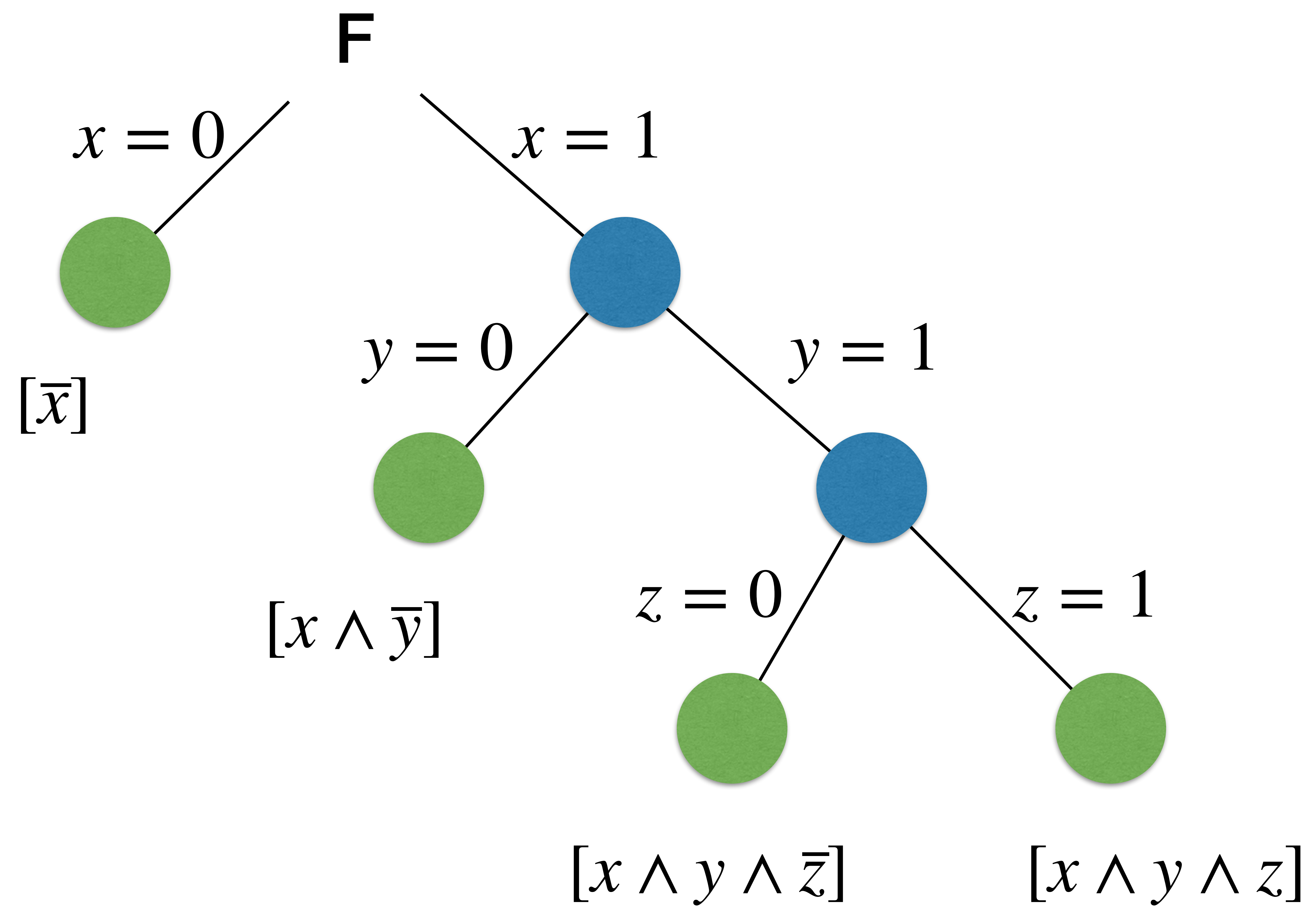
- smallest backdoor sets ≠ backdoor trees with smallest number of leaves!
- subset-minimal backdoor sets ≠ backdoor trees with smallest number of leaves

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.

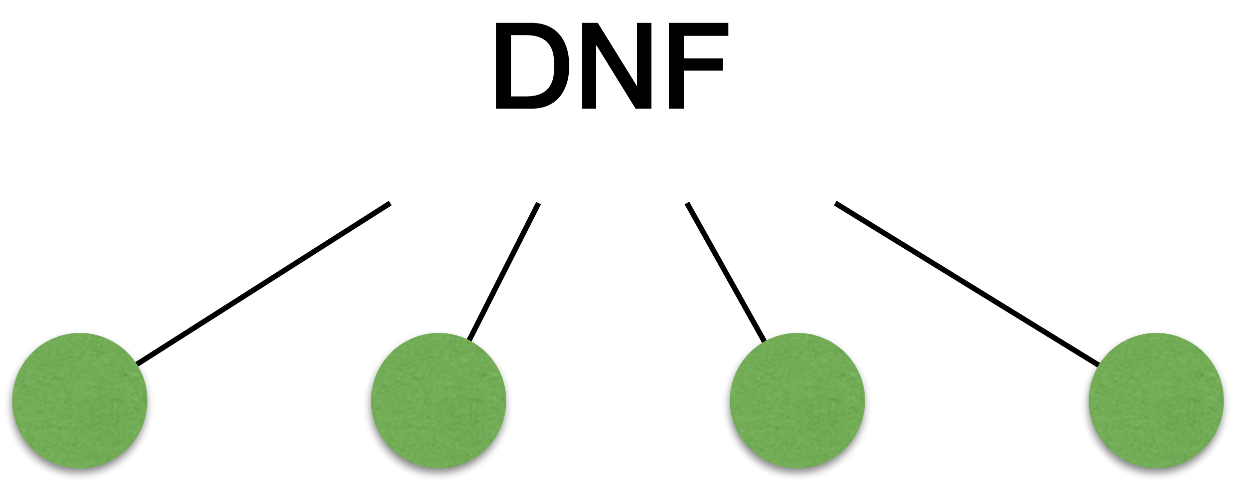


$$[\bar{x}] \vee [x \wedge \bar{y}] \vee [x \wedge y \wedge \bar{z}] \vee [x \wedge y \wedge z]$$

[Ordyniak, Sz. IJCAI 2021]

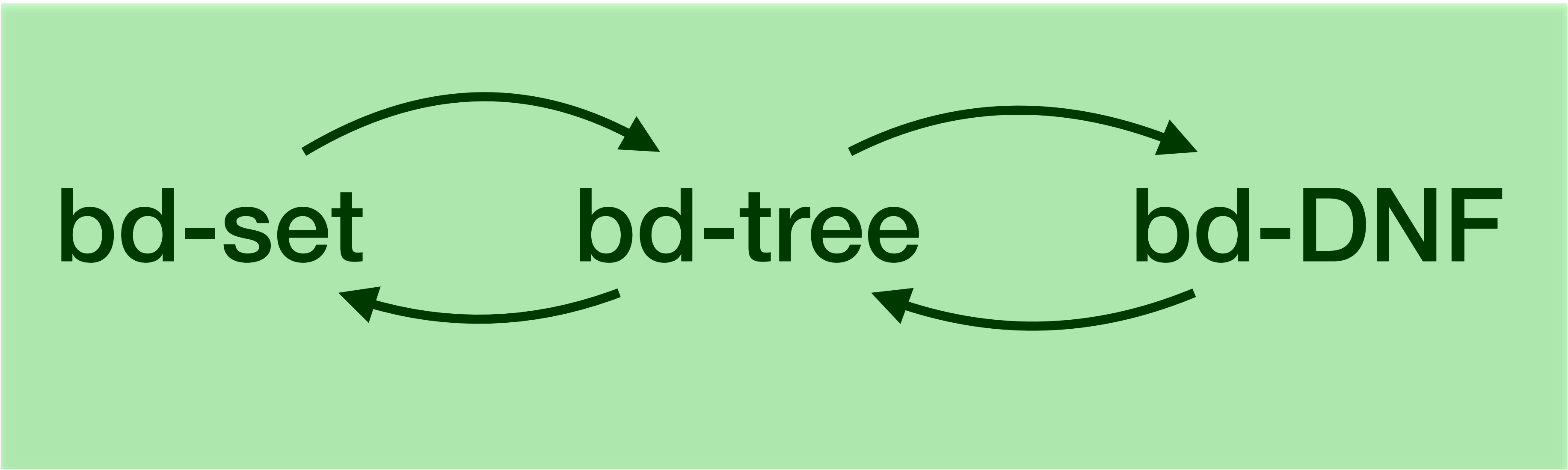
Avoid the 2^k assignments: Backdoor DNFs

- 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

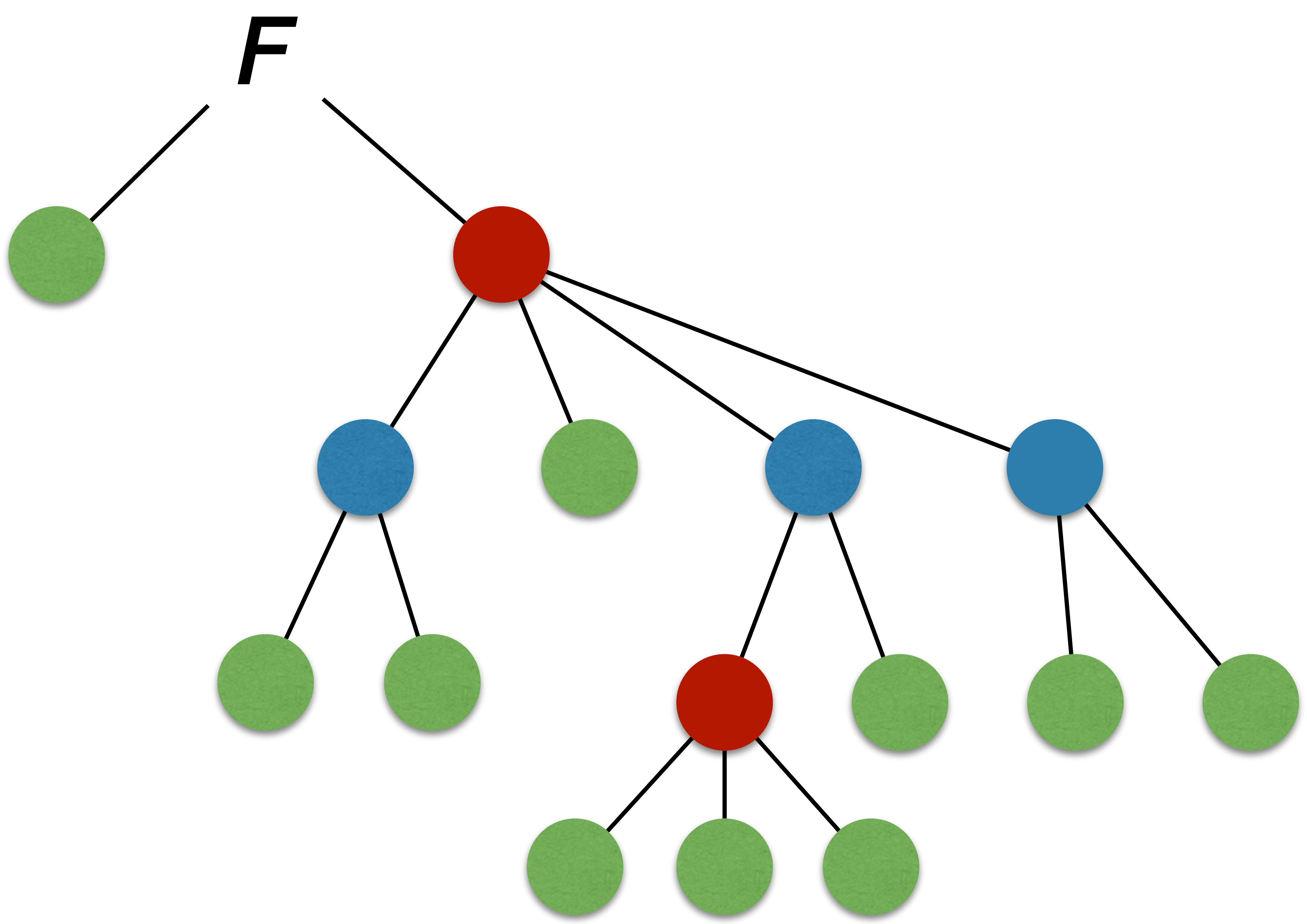
one can even mix Horn with 2CNF (or dHorn with 2CNF)



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!

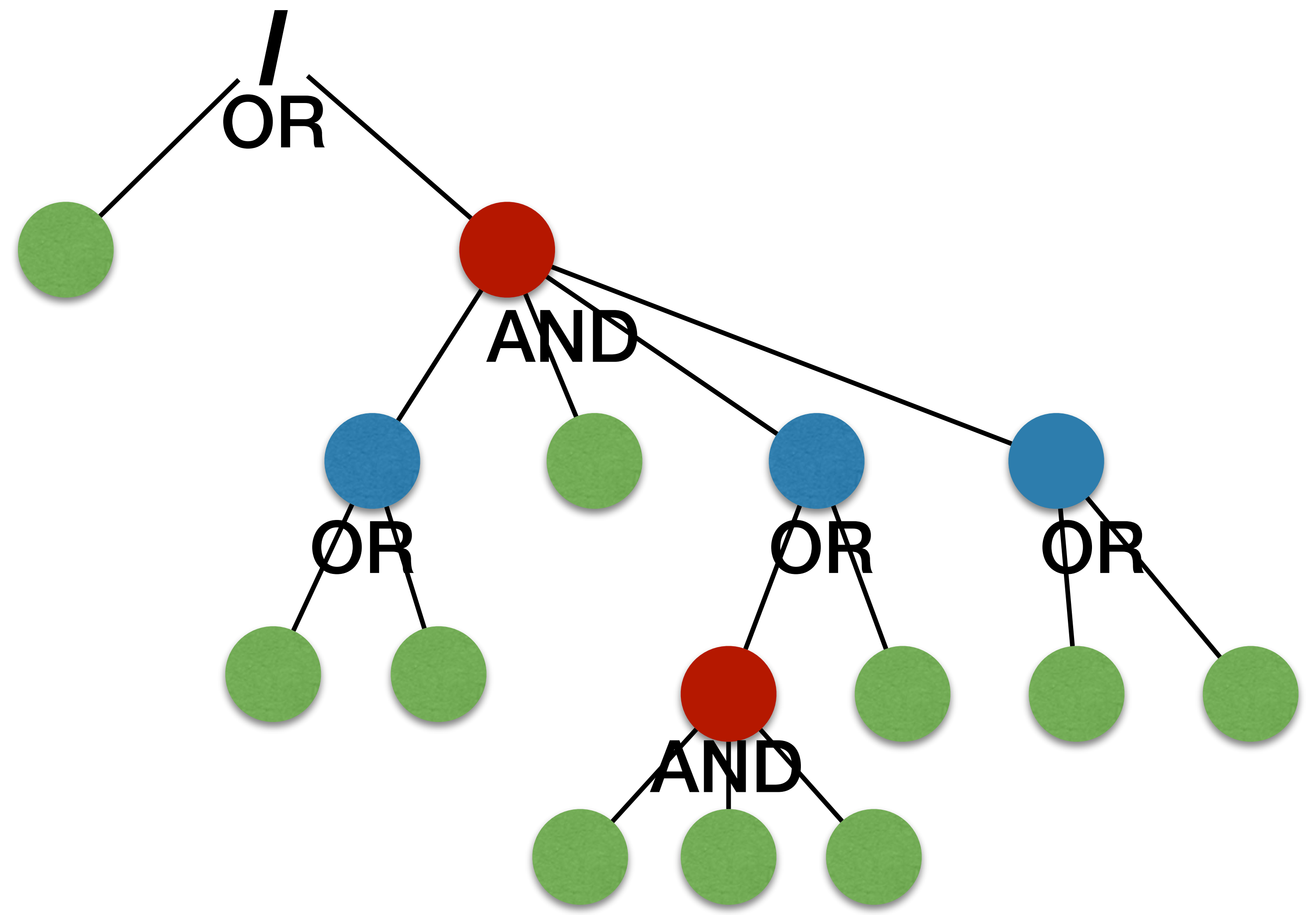
Component backdoor Trees

- Backdoor depth is significantly better parameter than backdoor size or number of backdoor tree leaves

- Definition motivated by tree-depth [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

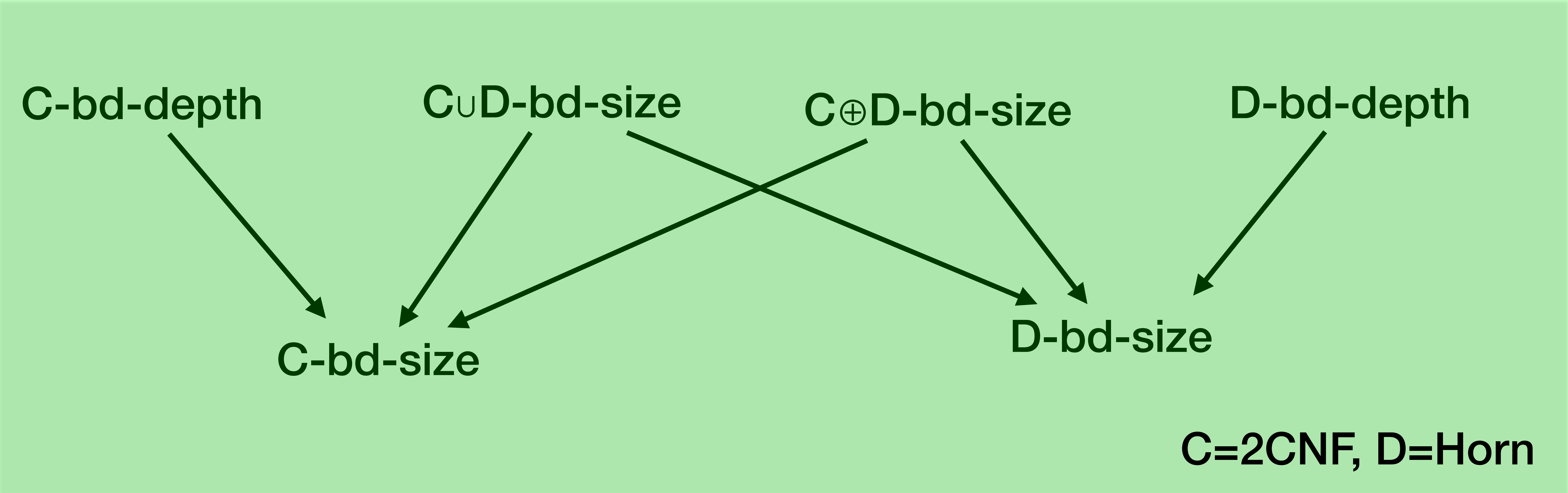
- Algorithmically challenging problem: find a component backdoor tree of small depth



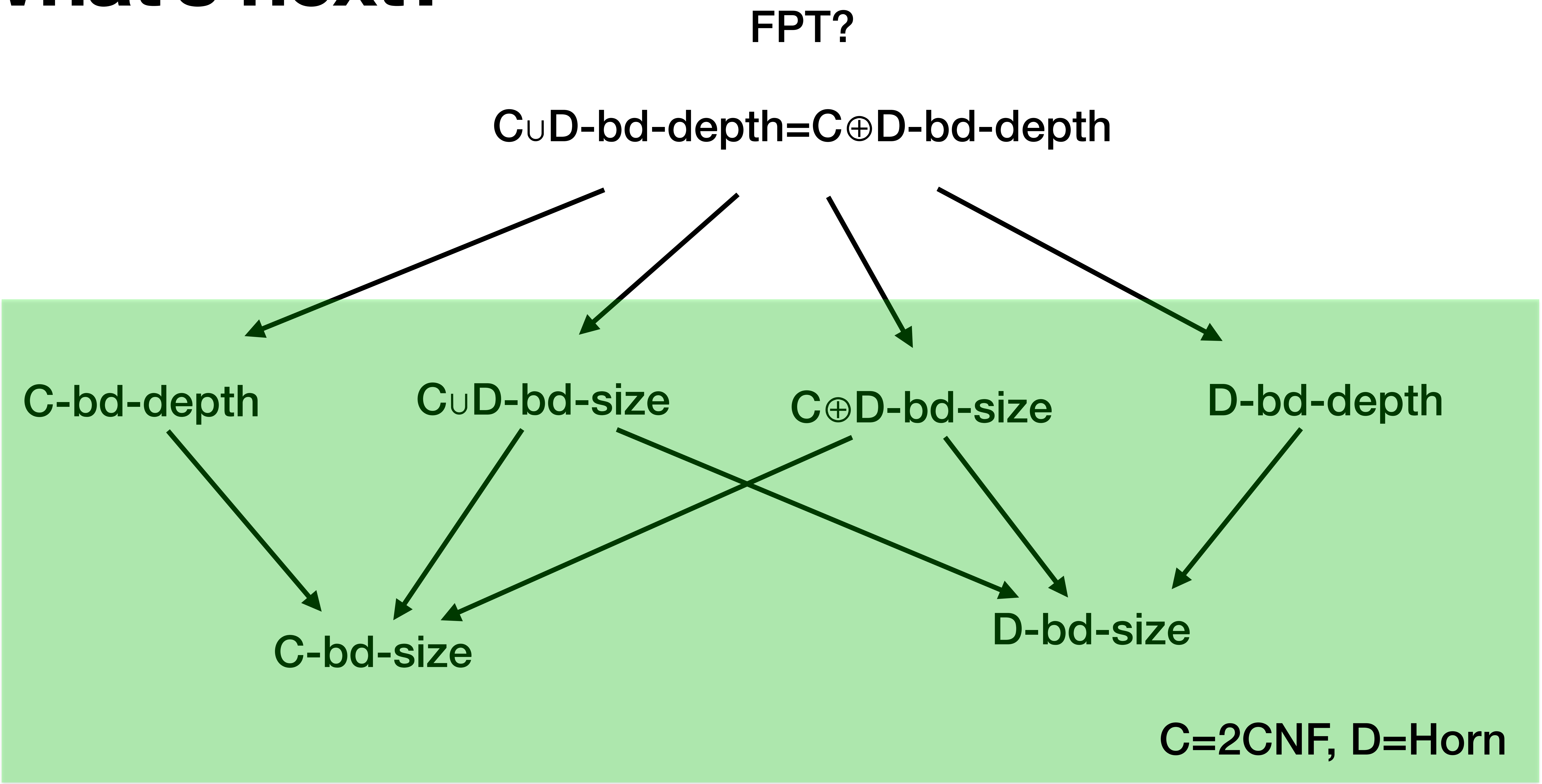
FPT-approximating backdoor depth

- **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

Comparison Summary

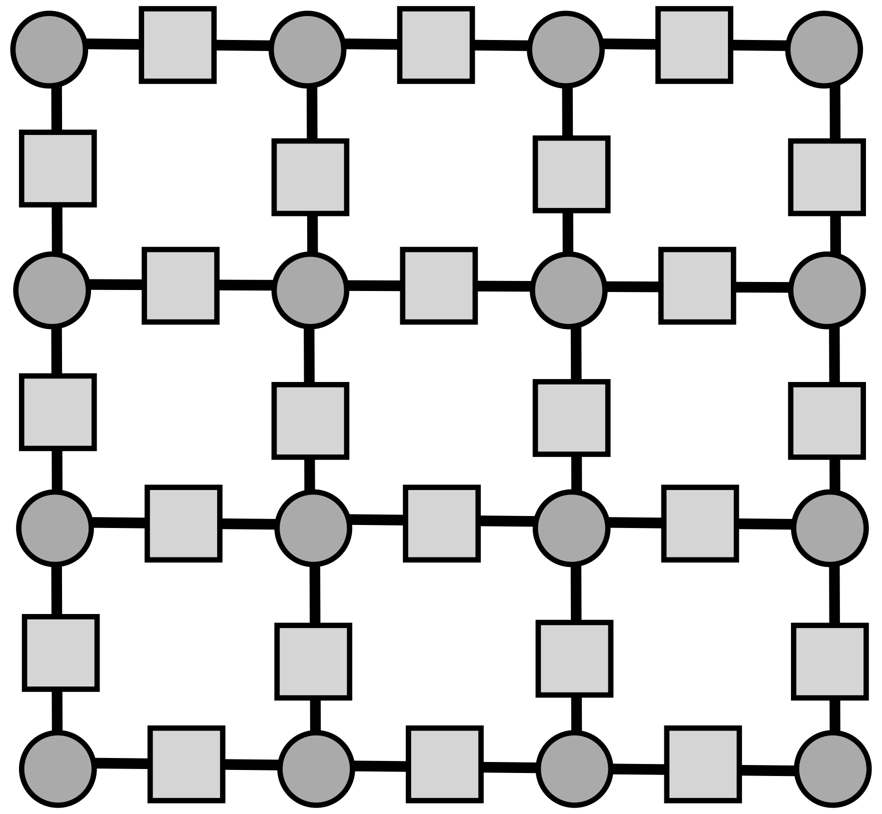


What’s next?

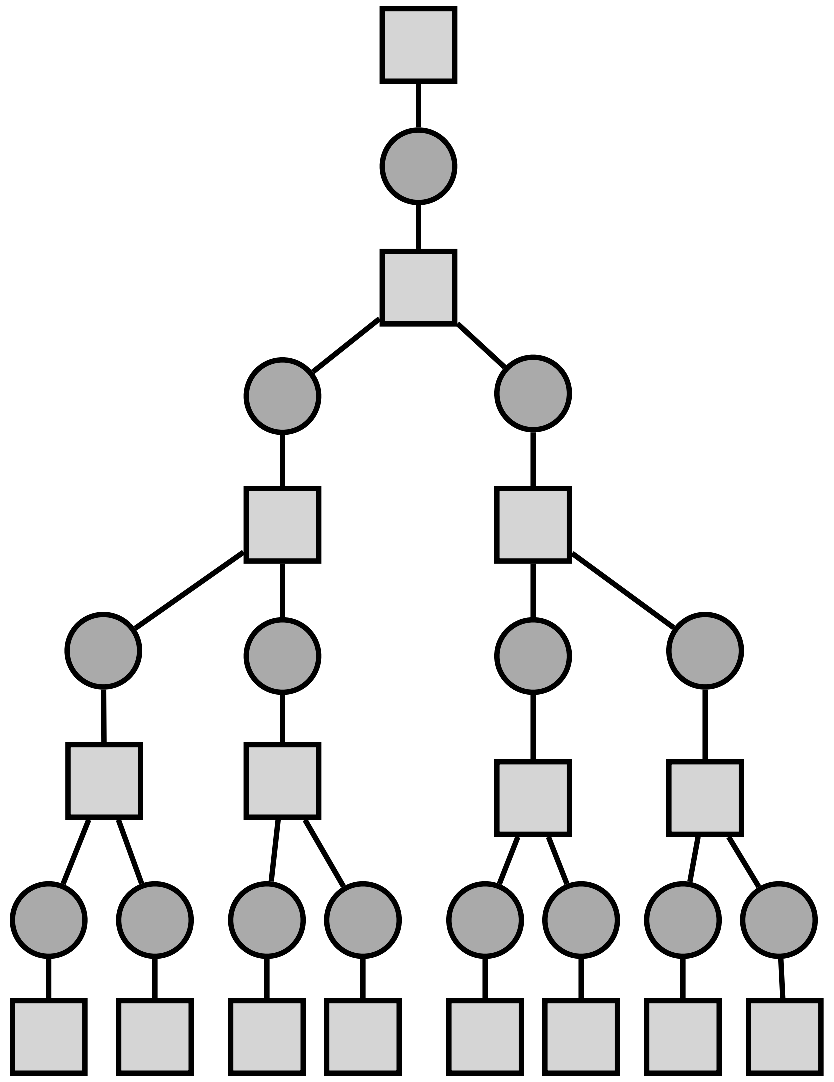

FPT?
C∪D-bd-depth=C⊕D-bd-depth
C-bd-depth
C∪D-bd-size
C⊕D-bd-size
D-bd-depth
C-bd-size
D-bd-size
C=2CNF, D=Horn

3) Hybrid parameters

Hybrid parameters



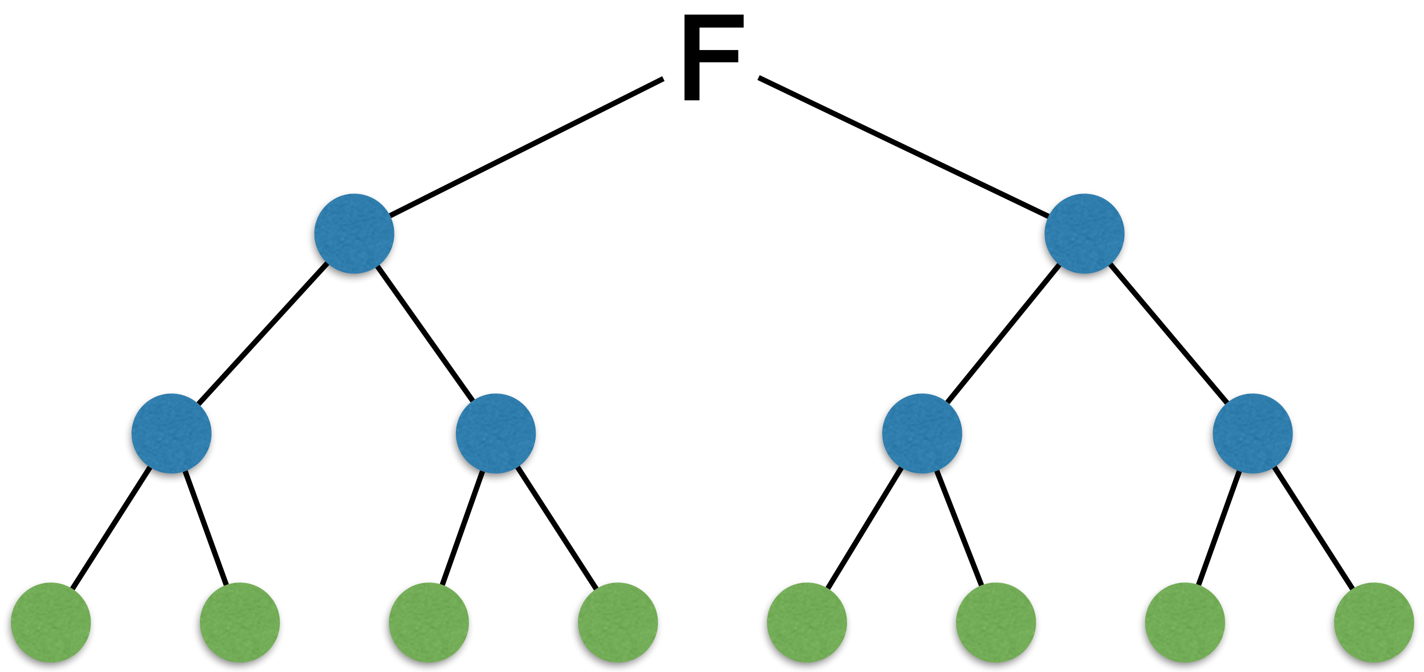
INCOMPARABLE



tw(nxn grid)=Ω(n)
Horn-backdoor=0

tw(n-tree)=1
Horn-backdoor=Ω(n)

(A) Backdoors into bounded treewidth



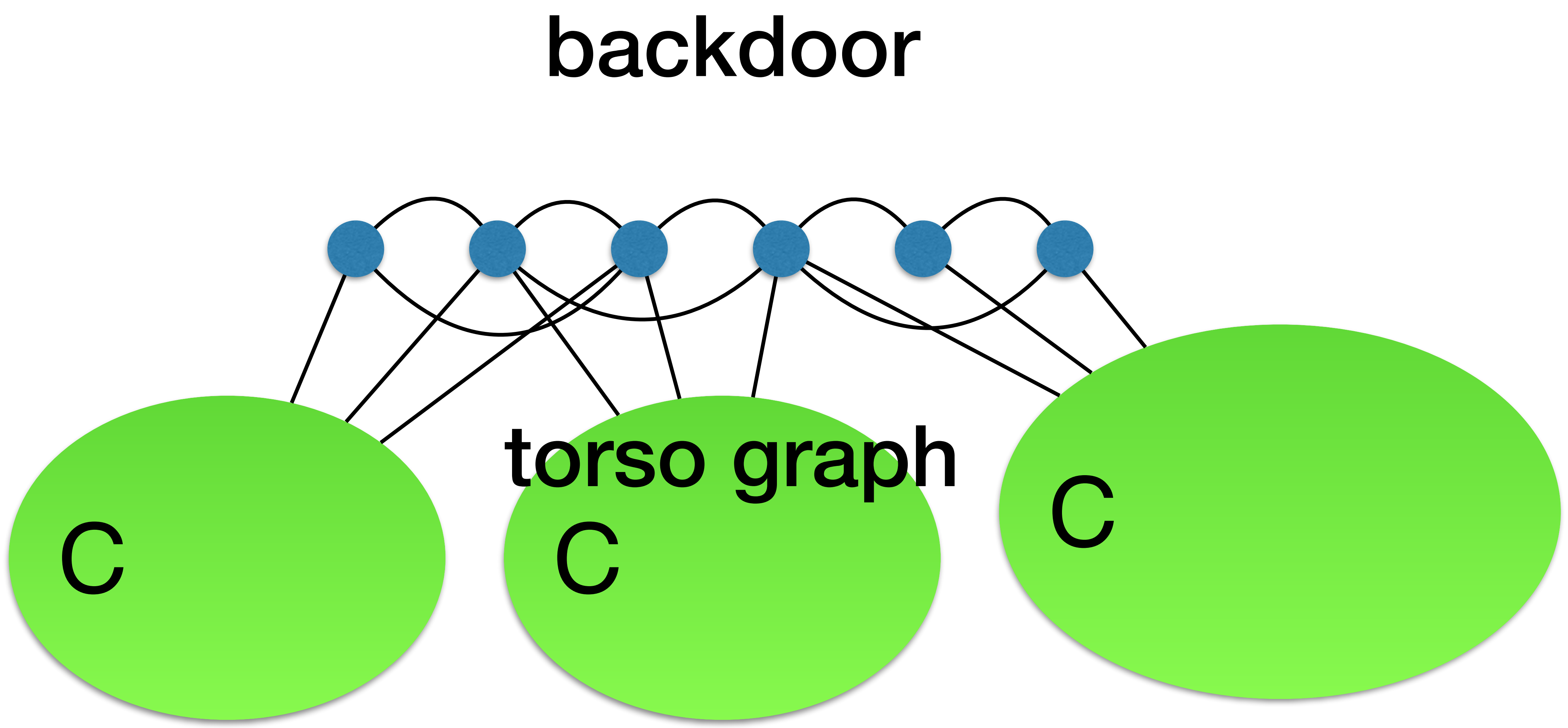
TW[t]

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.

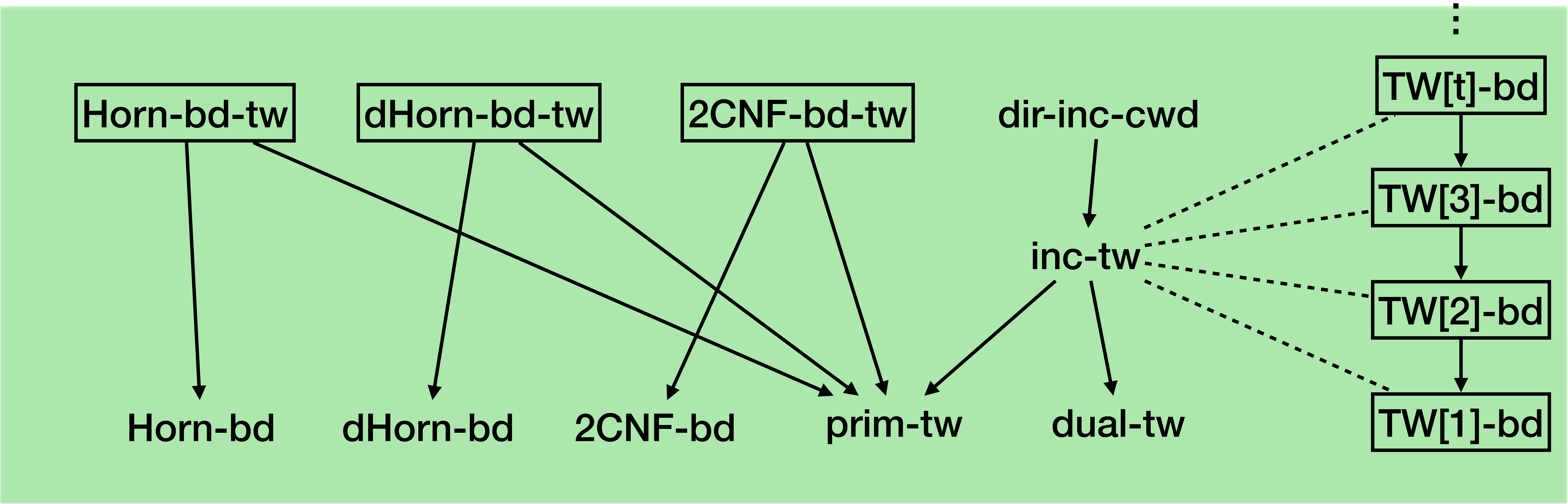
(B) backdoor treewidth



- **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}

Parameter Zoo

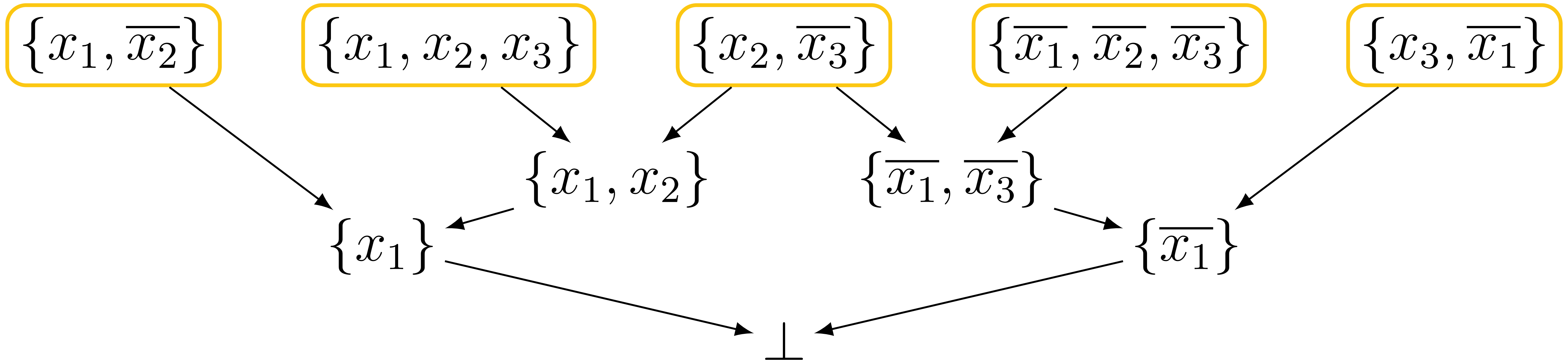


Resolution Complexity

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.

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

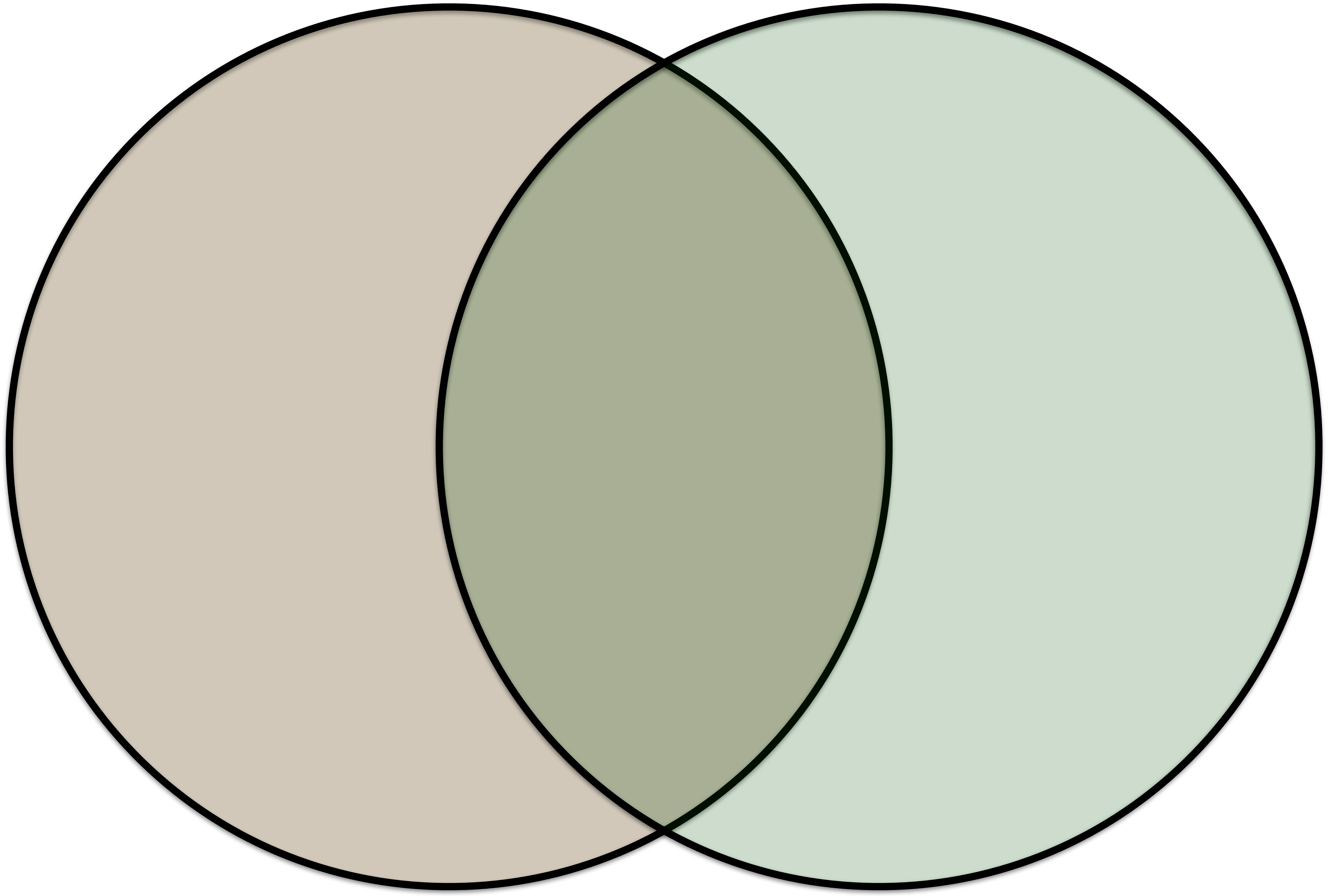


Resolution and SAT-solvers

- 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

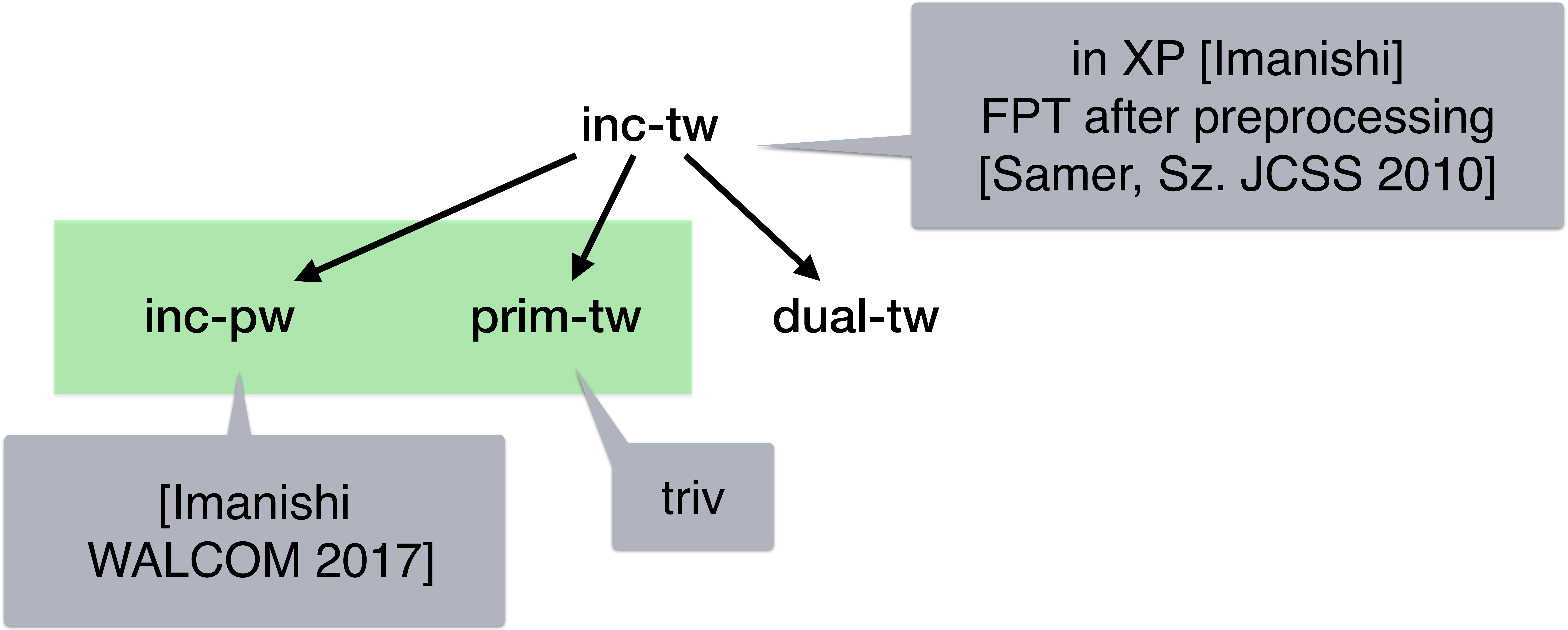
parameters for which SAT is FPT



Parameters for which FPT-size resolution proofs exist

= parameters where a SAT solver can have FPT running time

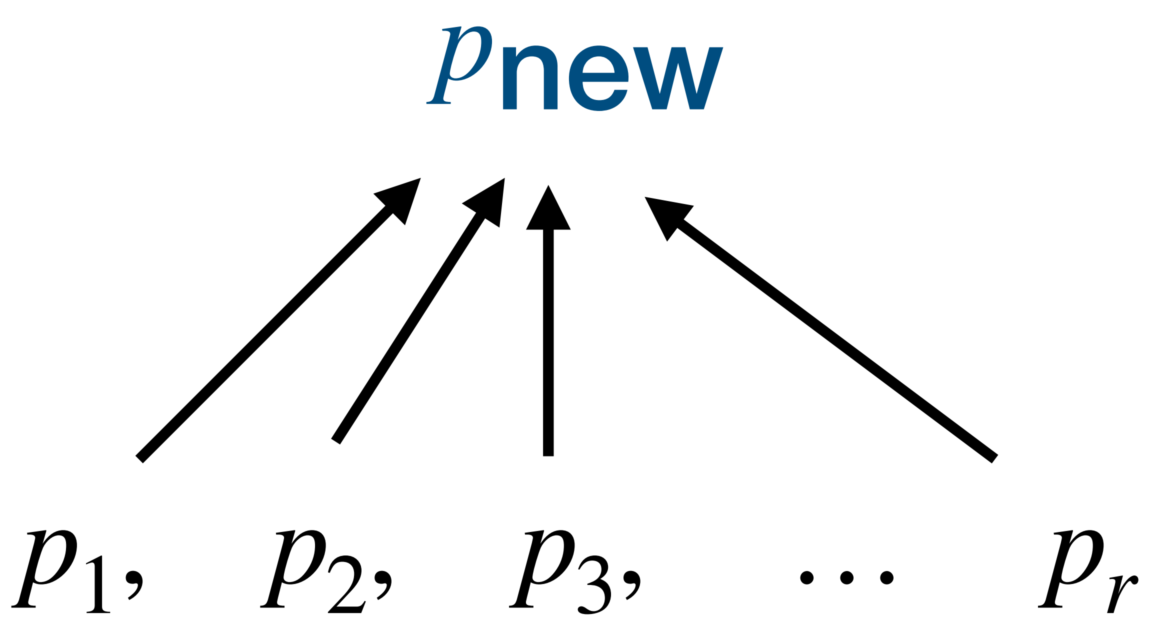
Resolution complexity by Treewidth



- Interesting open case is the resolution complexity of HIT
- [Peitl-Sz DAM 2023]

Summary

- FPT-SAT: over last 20 years, evolved into a rich research area
- Many new developments, including backdoor depth, bd-tw, twin-width,…
- Incomparable top elements of FPT-parameters - quest for unification



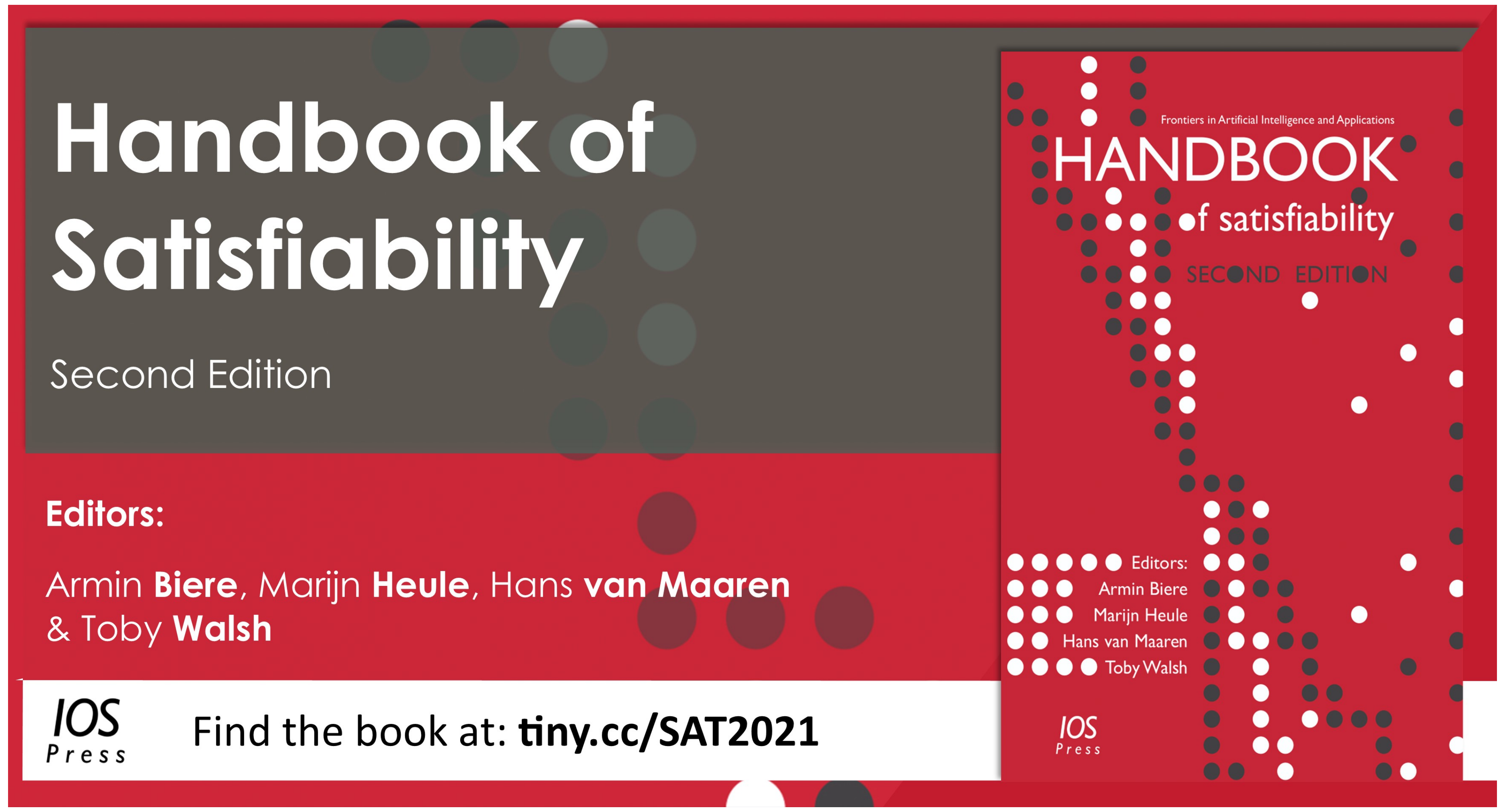
Some Challenges

- backdoor depth for heterogeneous/scattered Horn ∪ Krom
- FPT-size resolution for incidence treewidth
- resolution complexity of hitting formulas
- revisit hardness results for backdoors, avoid exact recognition (flow augmentation)
- Revisit FPT #SAT results from Knowledge Compilation Perspective

Handbook of Satisfiability, 2nd Edition, 2021

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

Extended and revised chapter 17
"Fixed-parameter Tractability"



(Some) References

- Robert Ganian, Filip Pokrývka, André Schidler, Kirill Simonov, Stefan Szeider: Weighted Model Counting with Twin-Width. SAT 2022: 15:1-15:17
- Johannes Klaus Fichte, Daniel Le Berre, Markus Hecher, Stefan Szeider: The Silent (R)evolution of SAT. Commun. ACM 66(6): 64-72 (2023)
- Tomáš Peitl, Stefan Szeider: Are Hitting Formulas Hard for Resolution? Discr. Appl. Math., volume 337, pages 173–184, 2023.
- Jan Dreier, Sebastian Ordyniak, Stefan Szeider: SAT Backdoors: Depth Beats Size. ESA 2022: 46:1-46:18
- Marko Samer, Stefan Szeider: Fixed-Parameter Tractability. Handbook of Satisfiability 2021: 693-736
- Johannes Klaus Fichte, Markus Hecher, Stefan Szeider: A Time Leap Challenge for SAT-Solving. CP 2020: 267-285; update arXiv:2008.02215v2
- Serge Gaspers, Neeldhara Misra, Sebastian Ordyniak, Stefan Szeider, Stanislav Zivný: Backdoors into heterogeneous classes of SAT and CSP. J. Comput. Syst. Sci. 85: 38-56 (2017)
- Stefan Szeider: On Fixed-Parameter Tractable Parameterizations of SAT. SAT 2003: 188-202