Parameterized Complexity of SAT

and Related Problems

Stefan Szeider

TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

ALGORITHMS AND COMPLEXITY GROUP

Theory and Practice

- For classical TCS, SAT is hard
- PPSZ [Paturi, Pudlak, Saks, Zane 05]—Algorithm for n-variable 3-SAT has running time of 1.364^n

  For $n = 200$ this gives 2 x age of universe in nanoseconds
- Exponential Time Hypothesis (ETH): 3SAT not solvable in $2^{o(n)}$
- On the other hand…. SAT solvers routinely solve industrial instances with millions of clauses and variables… theory-practice gap
- Common wisdom: real-world SAT-instances contain some kind of “hidden structure” which is implicitly utilised by solvers
- Can we utilize or capture 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]
[Xu, Hutter, Hoos, Leyton-Brown JAIR'08]

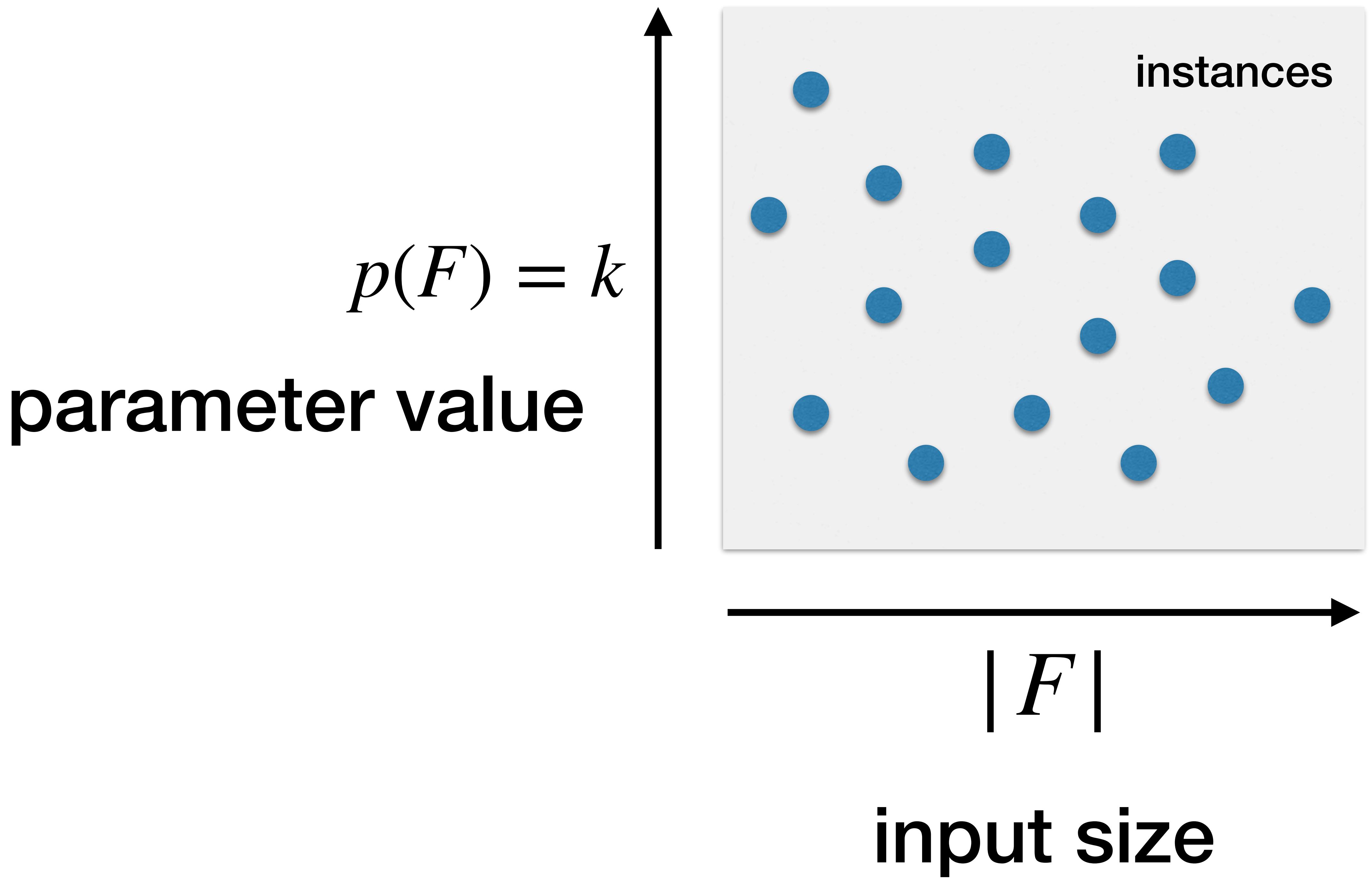
Causation

Try to capture structure in a way that provides worst-case performance guarantees for SAT algorithms

Parameterized Complexity, decomposability, backdoors, ….

Parameterized Complexity Framework

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

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

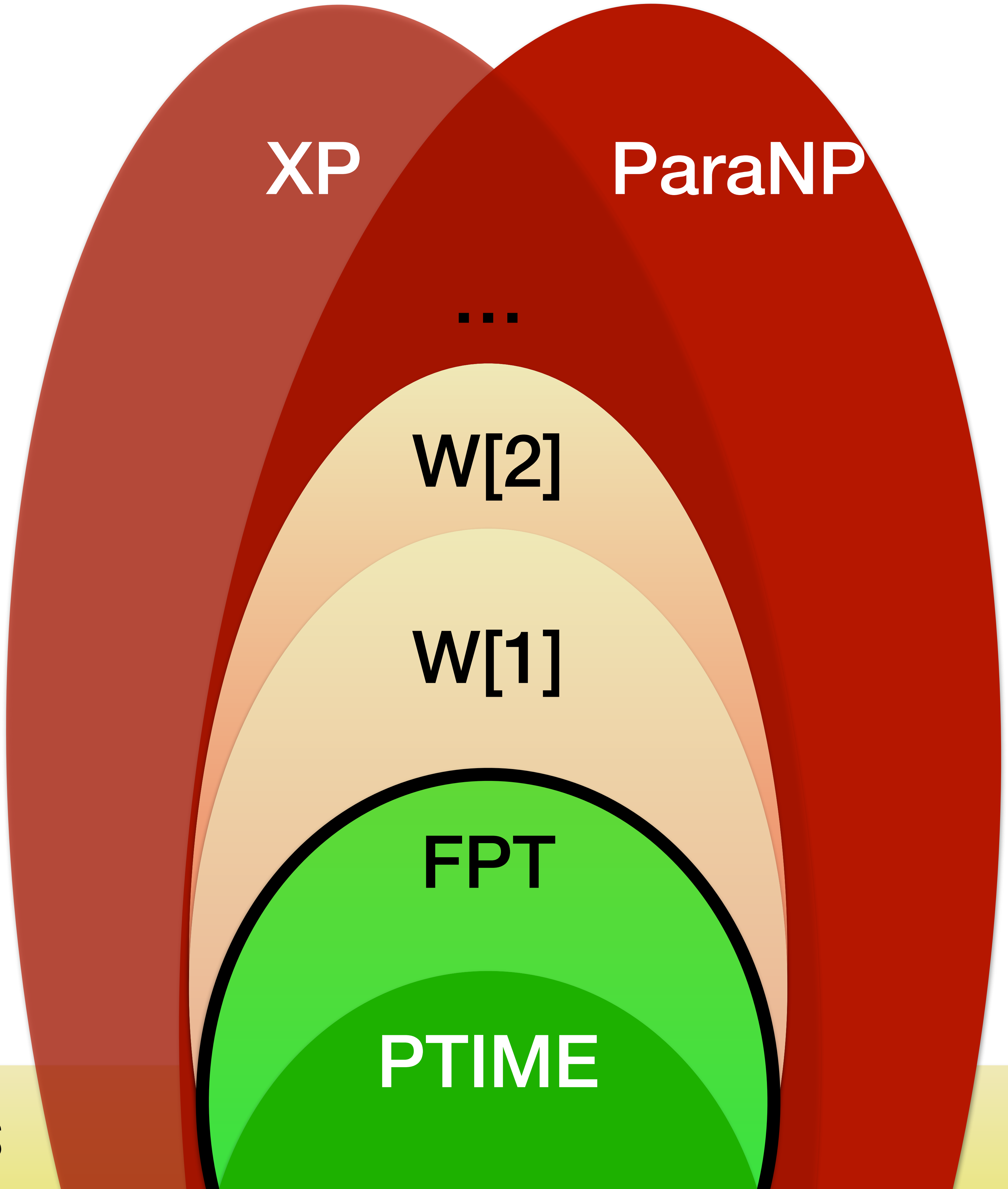
- **parameter k contributes a constant factor to the polynomial runtime, without changing the order of the polynomial**
- allows a better scaling in k
- such runtime guarantees are called **FPT** or **fixed-parameter tractable**
- well-developed area of TCS

Rich Theory

1999 2006 2006 2013 2013

Hardness Theory

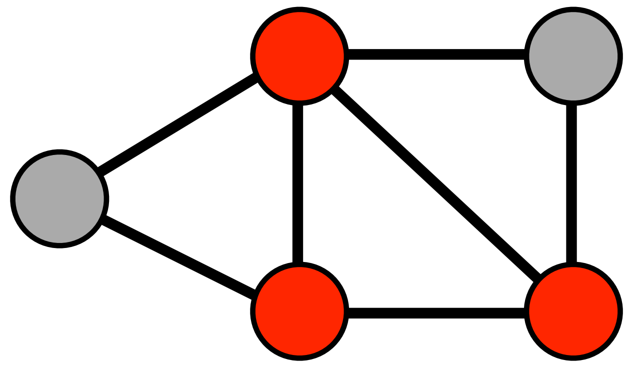
in P for constant k



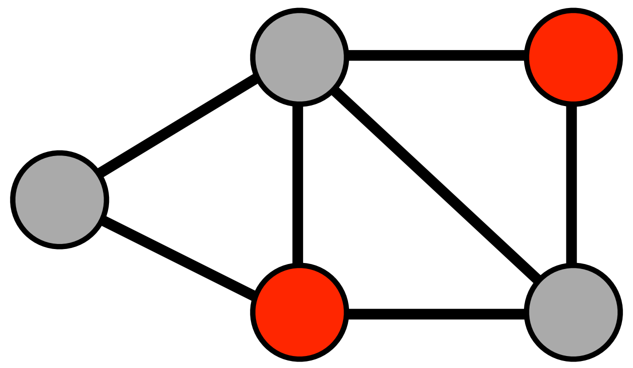
in NP but for constant k NP-complete (like k-SAT)

Standard parameter: solution size

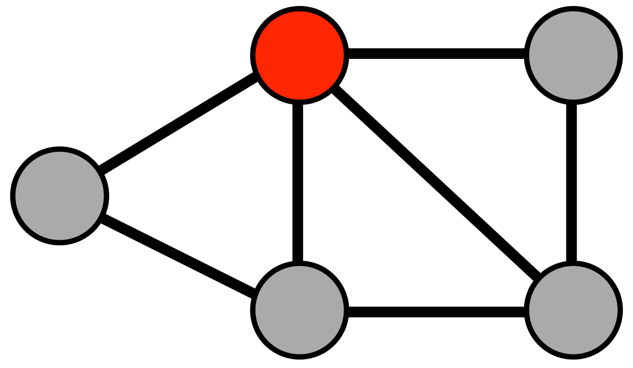
vertex cover



independent set



dominating set



Given a graph G,

1. find a **vertex cover** of size k — FPT
2. find an **independent set** of size k — W[1]-complete
3. find a **dominating set** of size k — W[2]-complete

All three problems are NP-complete
All three problems can be solved in XP-time $O(n^k)$
The problems are of *different* practical hardness
The problems are of *different* parameterized complexity

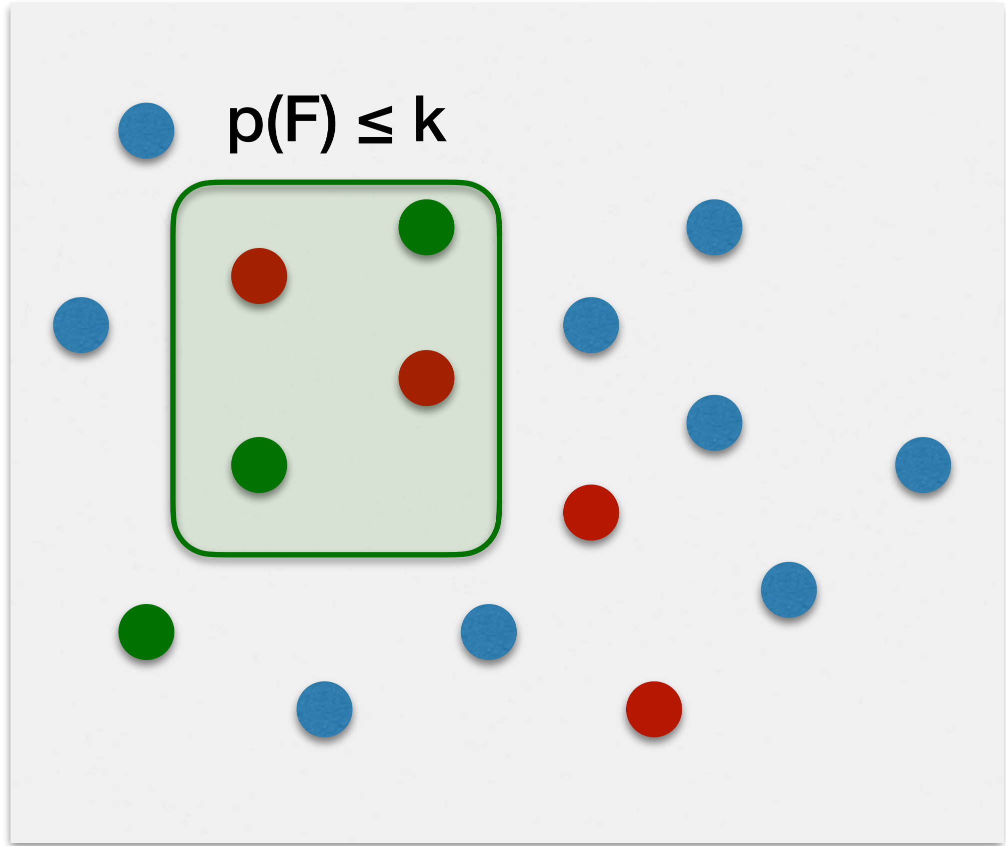
How to parameterise by “structuredness”

Take SAT as an example

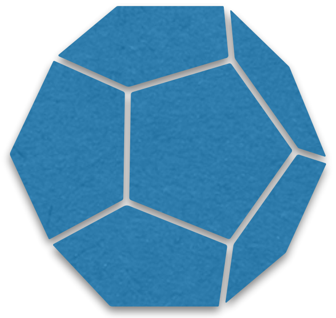
FPT-SAT

“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

[Sz. SAT'03]

1) Graphical Structure

2) Syntactical Structure

3) Hybrid Models

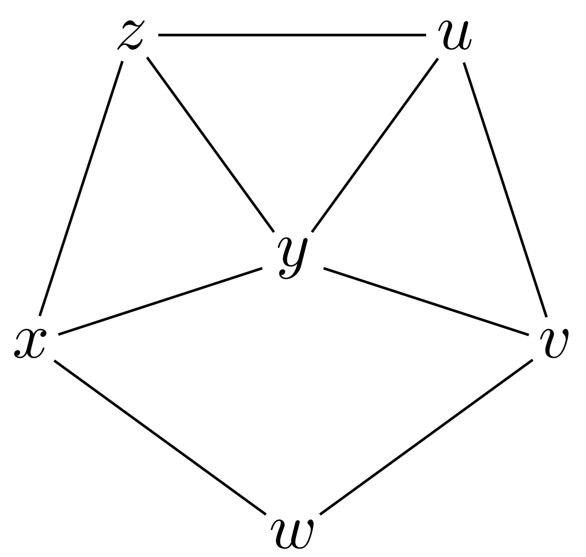
Graphical Structure

Common Graphs

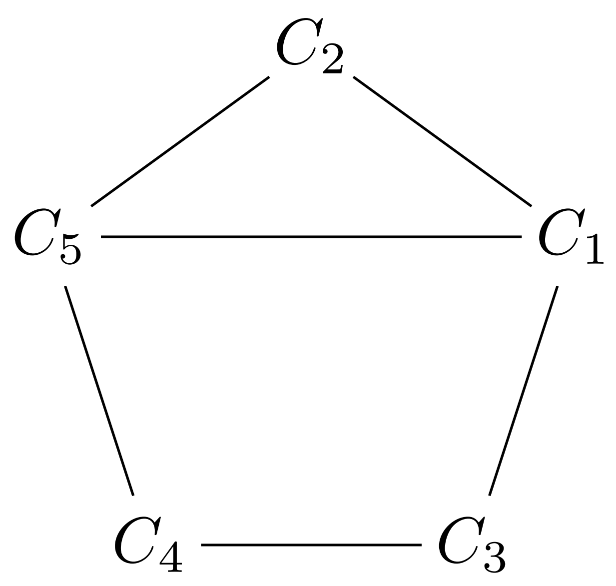
$F = \{C_1, \ldots, C_5\}$

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

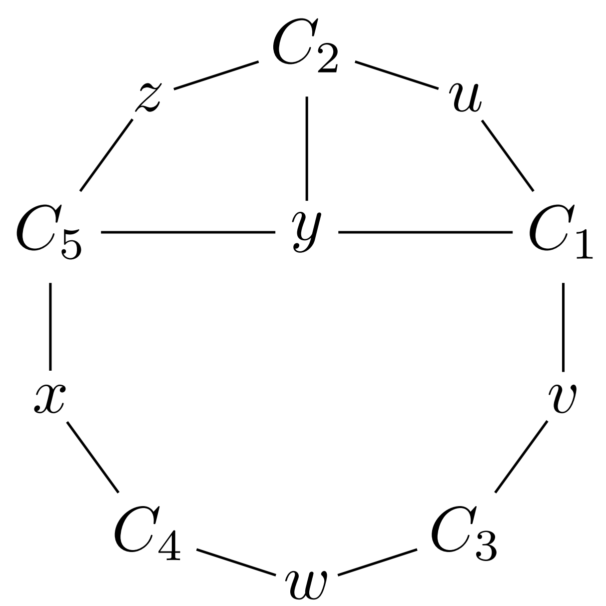
primal aka VIG



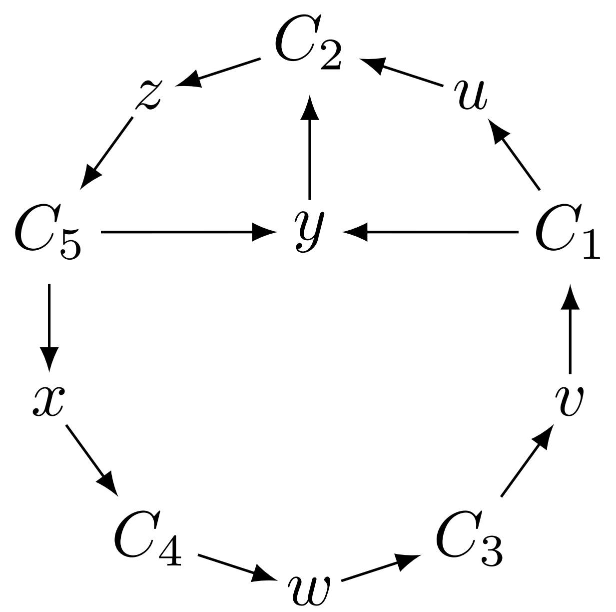
dual aka CIG



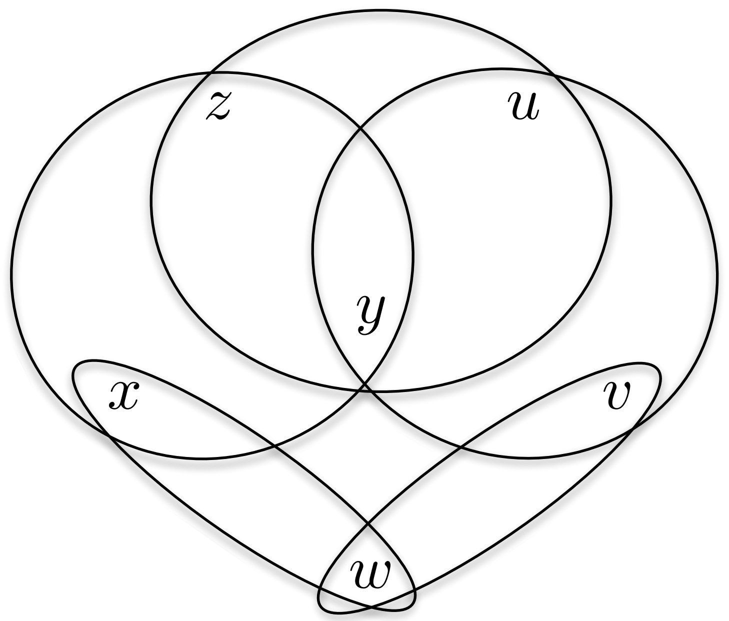
incidence aka CVIG



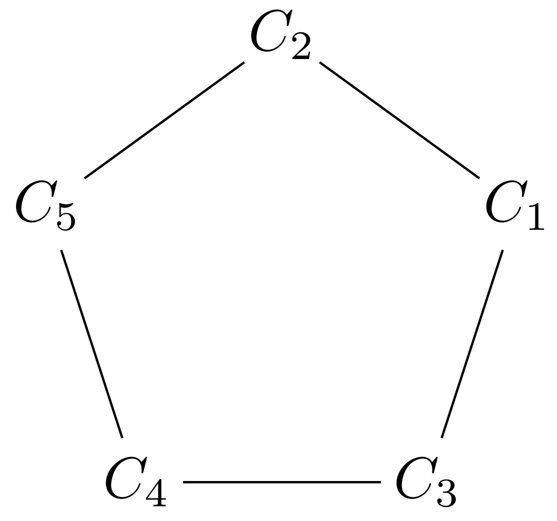
directed inc



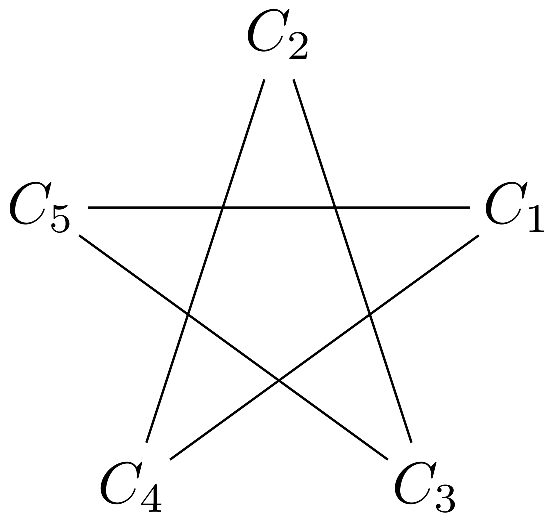
hypergraph



conflict

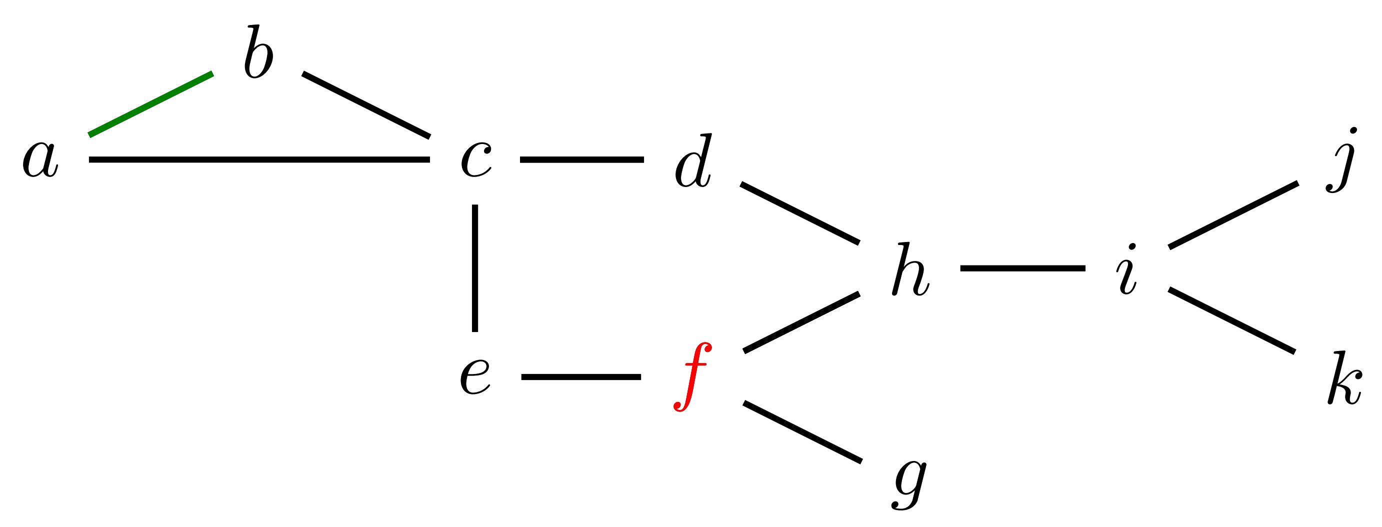


consensus

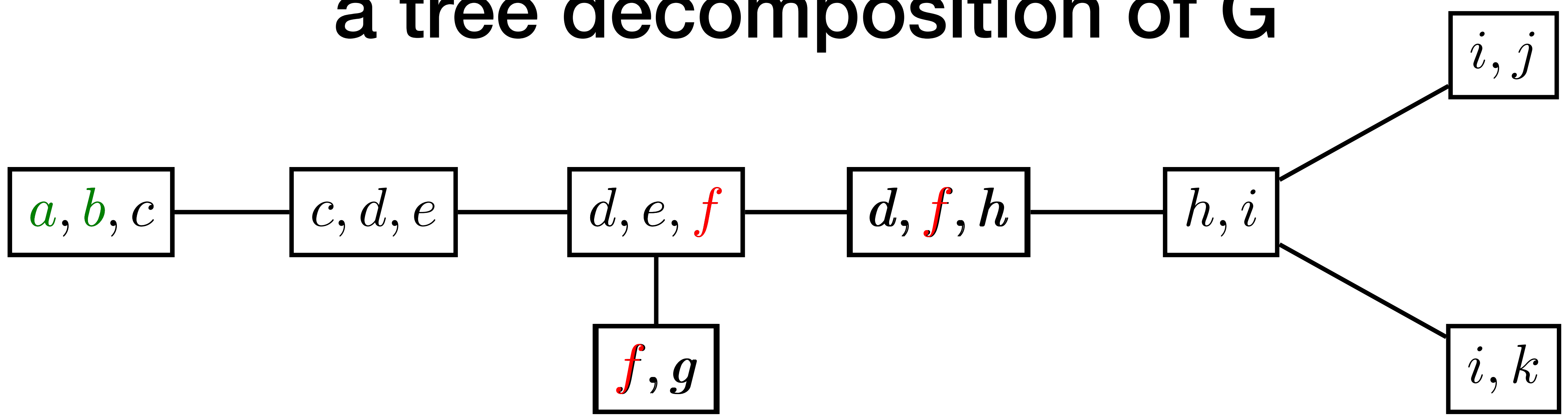


Graph Decompositions and Width Parms

a graph G



a tree decomposition of G



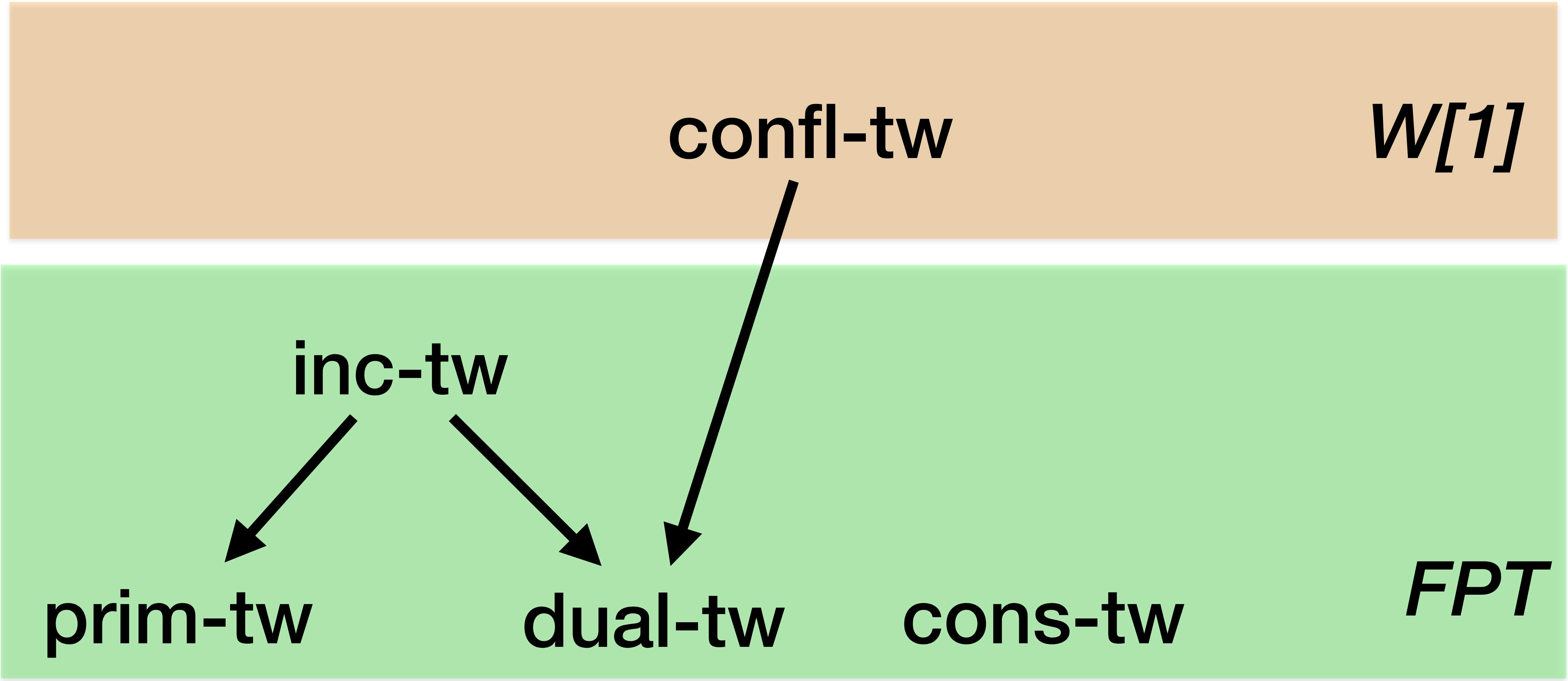
width = size of largest bag -1

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

Treewidth of Formulas

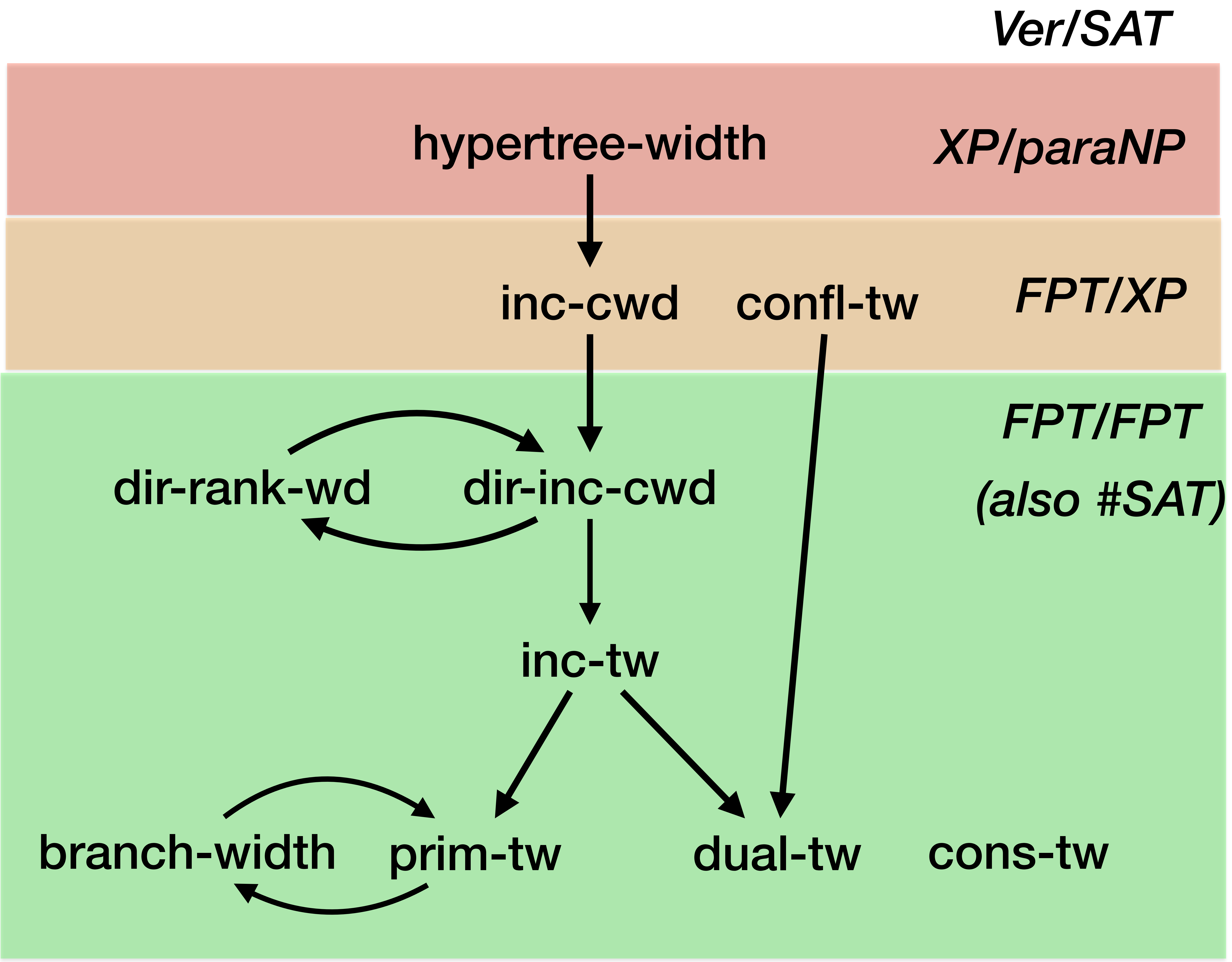
When we talk about the treewidth of a formula, we should always specify the graph we're referring to!

- 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.



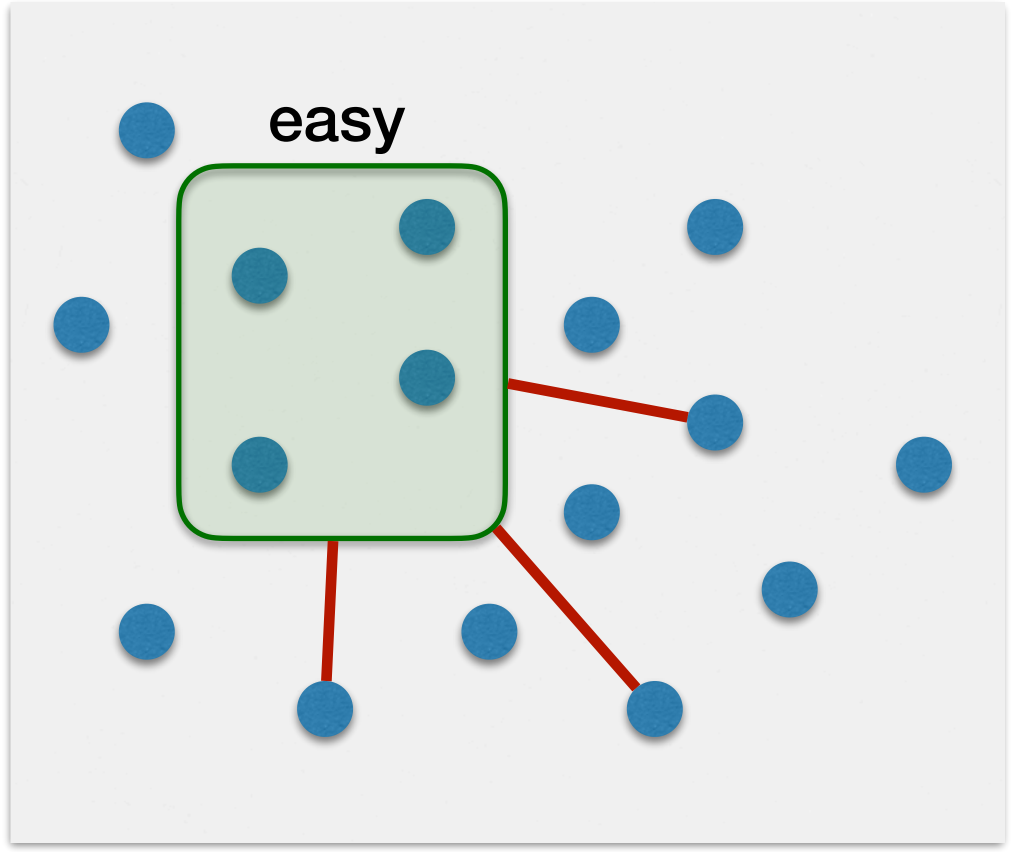
Width Parameter Zoo

width parameters:
usually, when decision is FPT, then also **counting**, **optimization** etc are FPT as well



Syntactic Structure

Tractable Classes or Islands of Tractability



Parameterize by the **distance to a class**

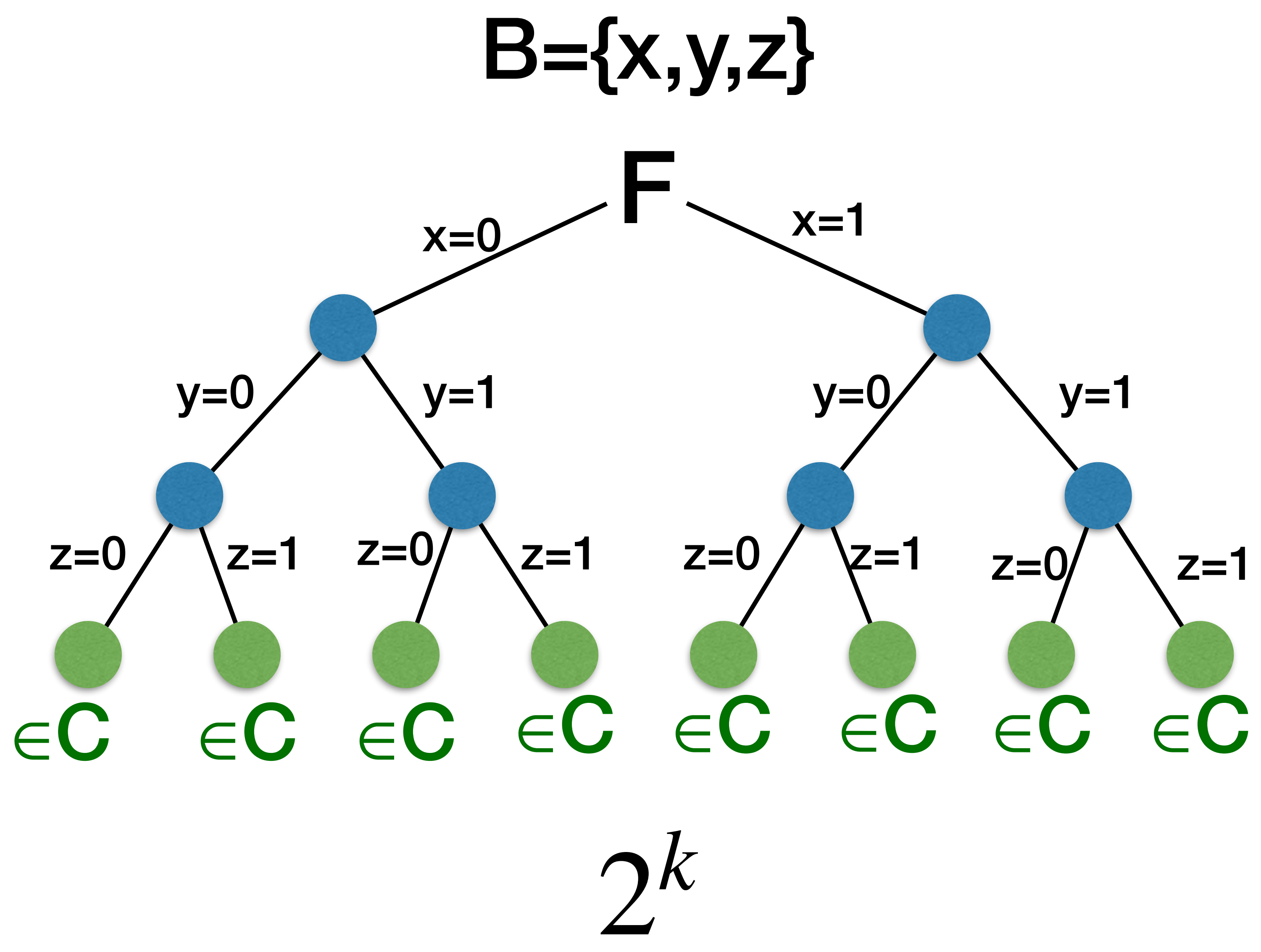
where the class is syntactical defined

(e.g., Horn or 2CNF)

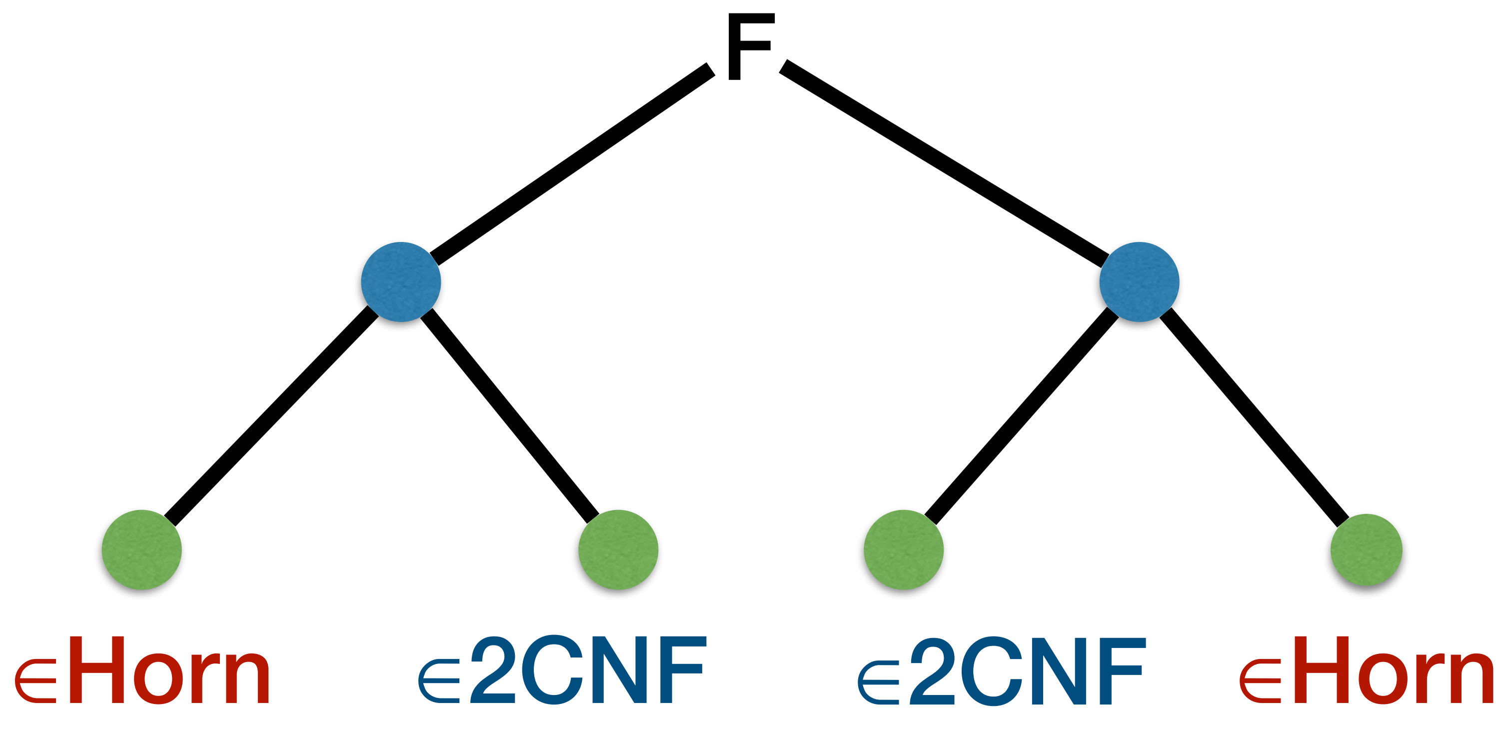
Distance = size of smallest backdoor set

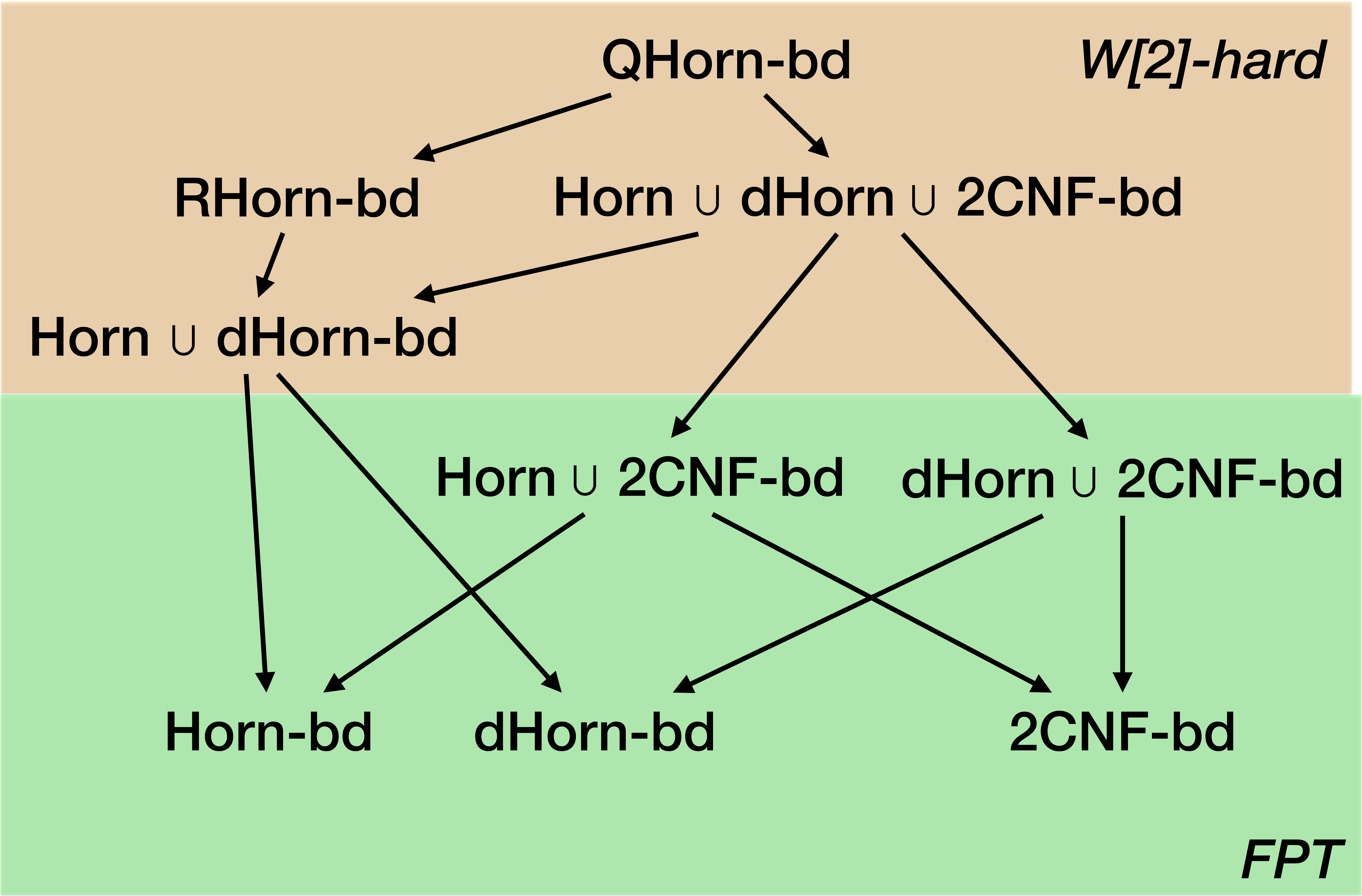
strong

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



Backdoor Parameter Zoo





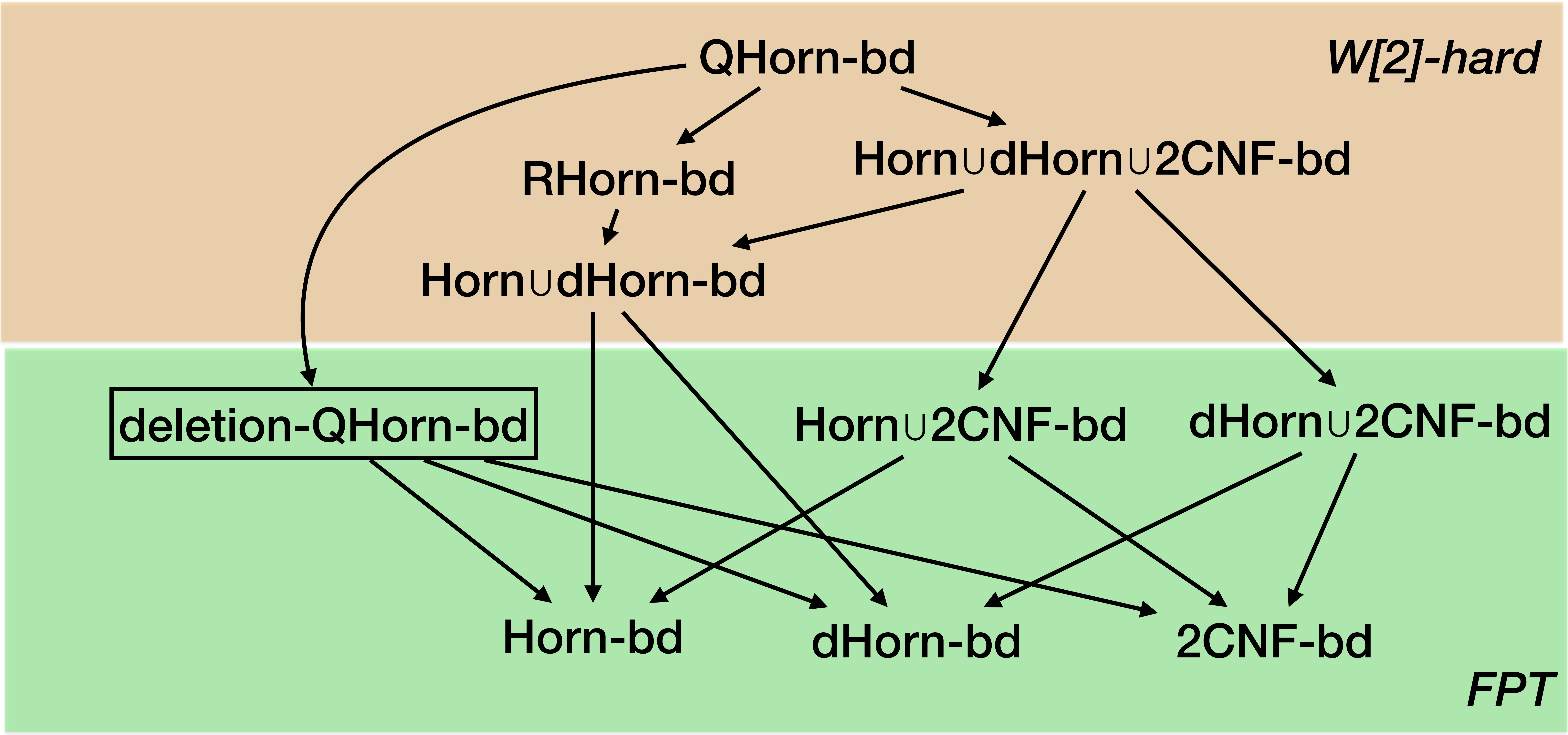
Deletion backdoor sets

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

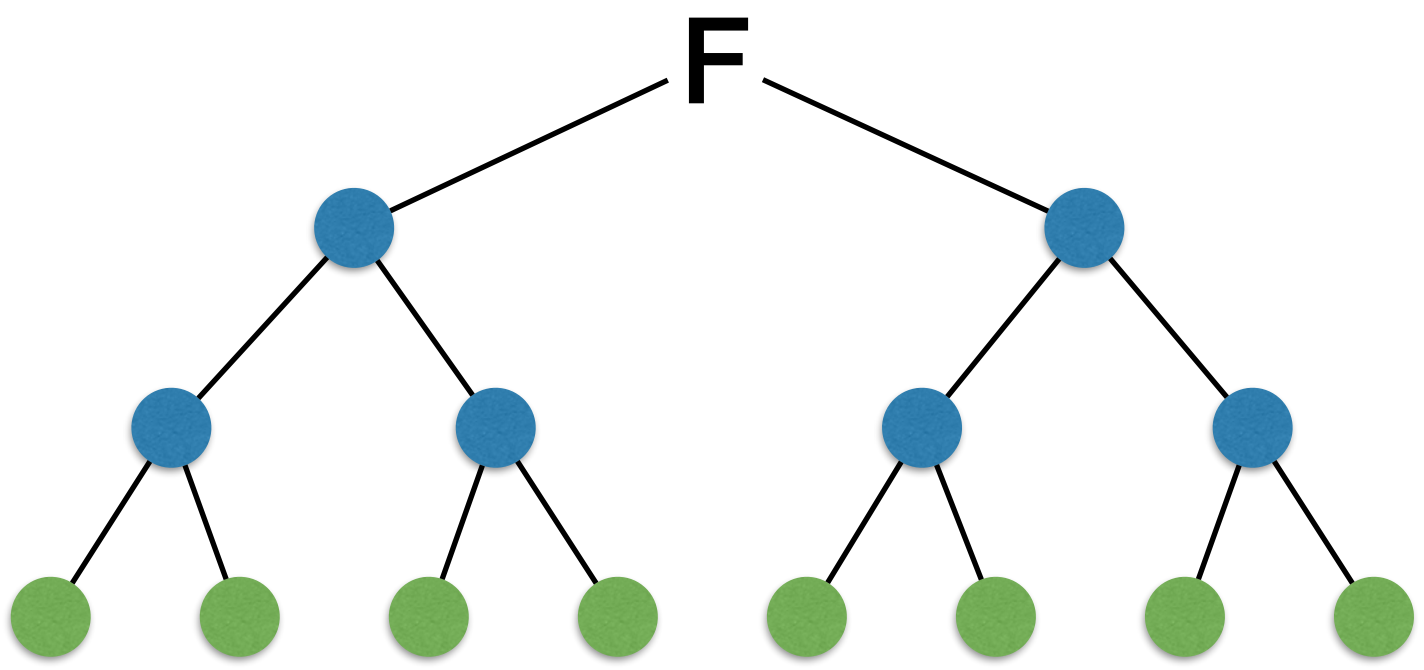
- Instead of looking at all partial assignments $t{:}B \to \{0,1\}$ we delete the backdoor variables from F (notation $F - B$)

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

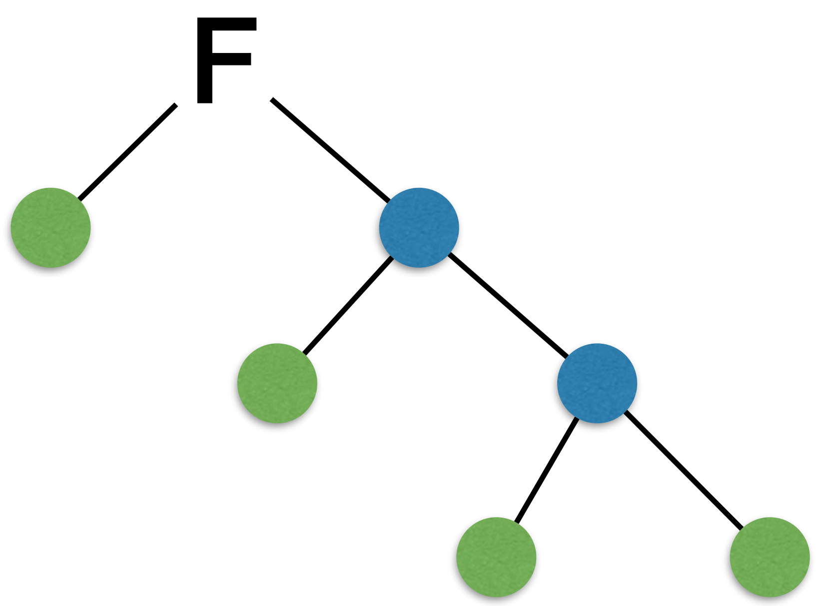
Deletion Backdoor Sets



Avoid the 2^k assignments: Backdoor Trees:



$$2^k$$



$$k+1$$

size of backdoor tree = number of leaves

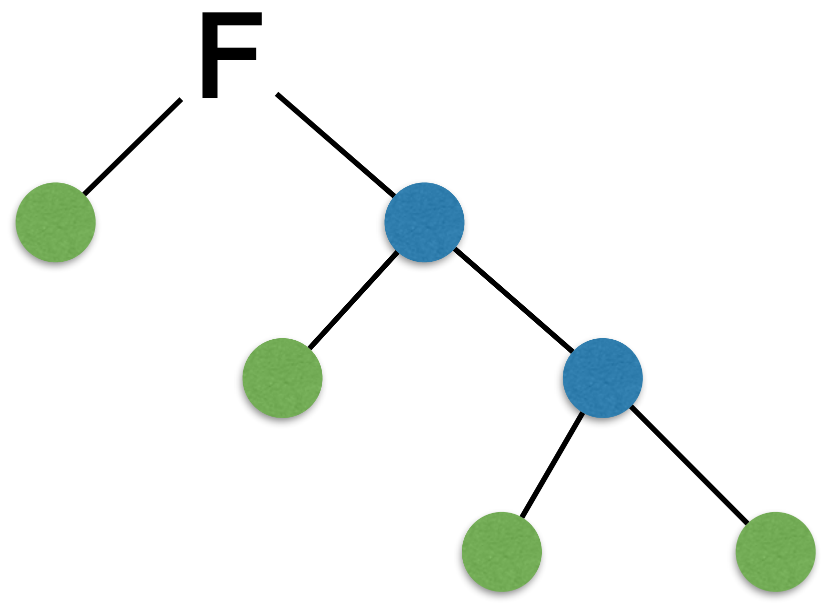
- smallest backdoor sets ≠ backdoor trees with smallest number of leaves!
- subset-minimal backdoor sets ≠ backdoor trees with smallest number of leaves

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

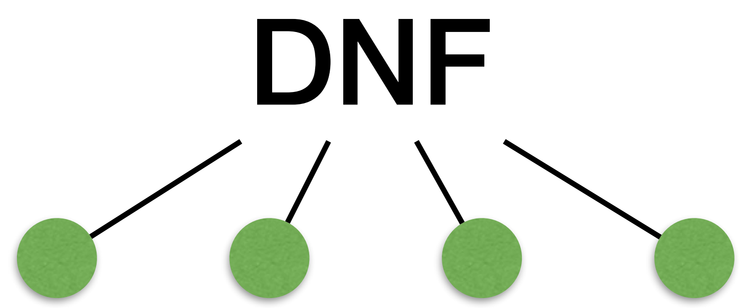
one can even mix Horn with 2CNF (or dHorn with 2CNF)

[Samer Sz. AAAI'08]

Avoid the 2^k assignments: Backdoor DNFs

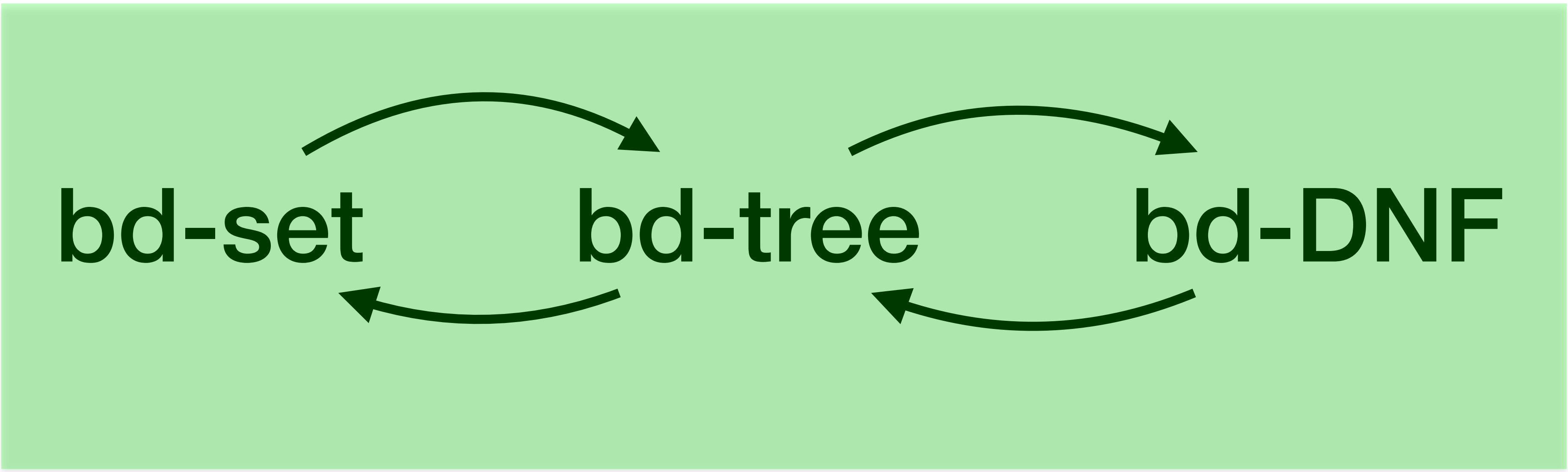


- Partial assignments at the leaves of a backdoor tree give rise to a DNF
- The DNF is a tautology
- **Backdoor DNF**: take any such tautological DNF
- Backdoor DNFs are more succinct than backdoor trees



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

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



[Ordyniak, Schidler, Sz IJCAI'21]

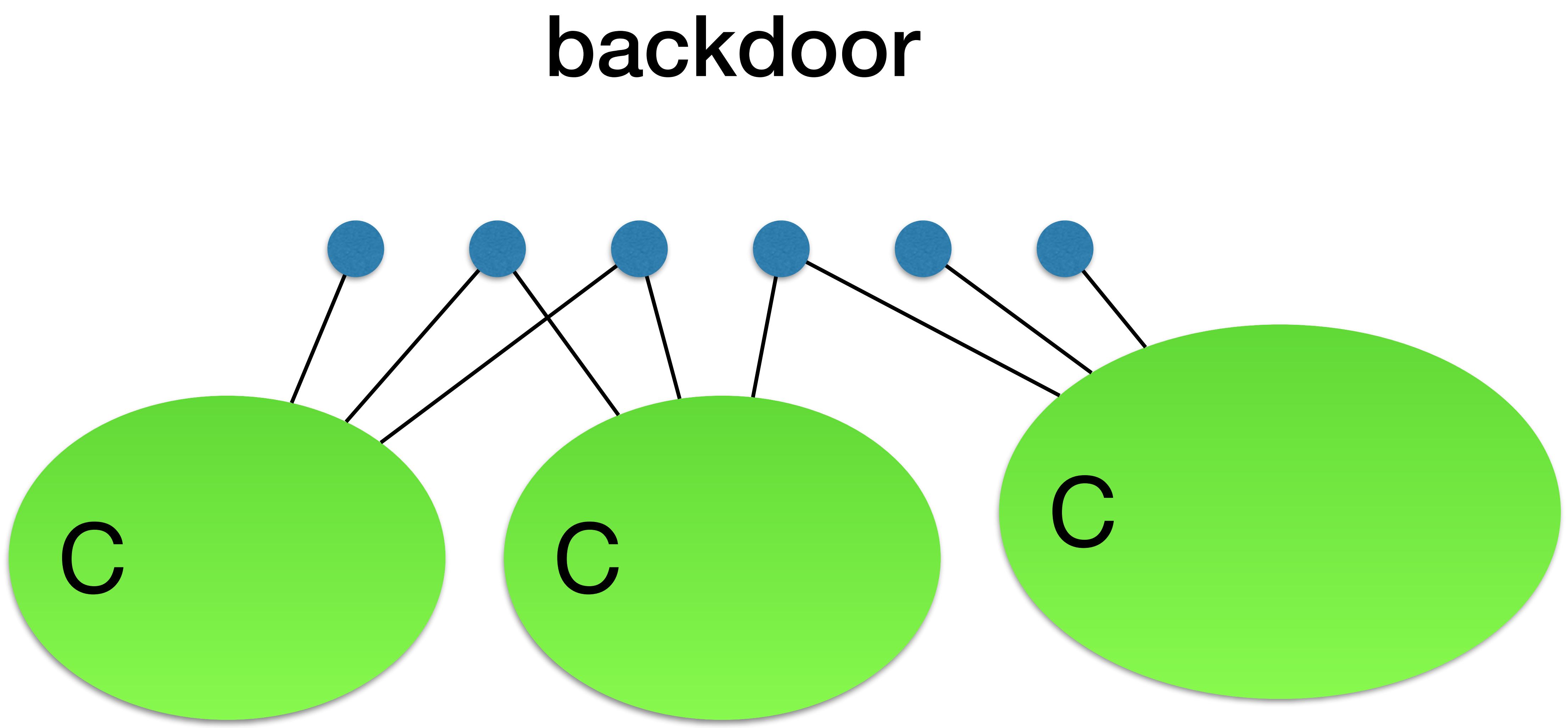
Hybrid parameters

large incidence treewidth
constant Horn-bd size

INCOMPARABLE

large Horn-bd size
constant incidence treewidth

backdoor treewidth



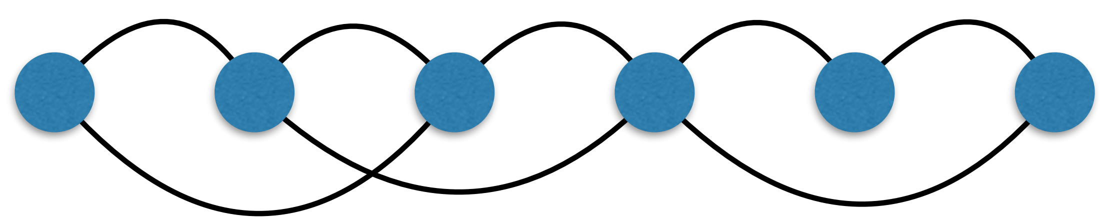
- **C-backdoor treewidth** is the minimum treewidth over the torso graphs of all the C-backdoors.
- C-backdoor treewidth
  ≤ min{ primal treewidth, C-backdoor size}

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

[Ganian, Ramanujan, Sz.
STACS'17, SAT'17]

backdoor treewidth

backdoor



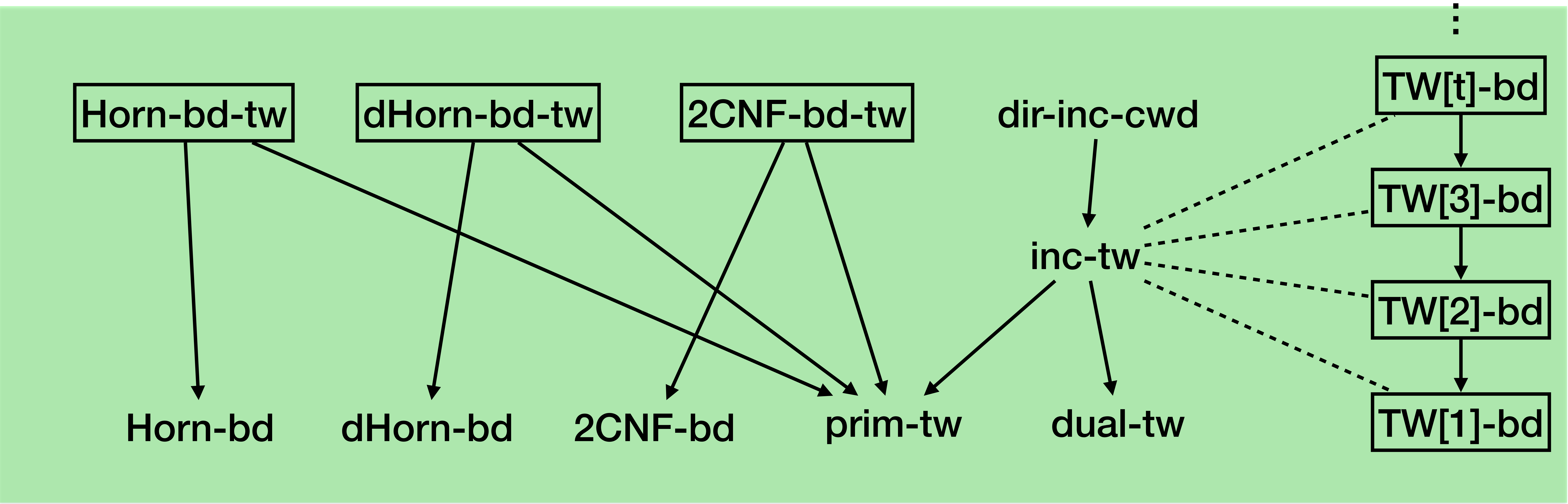
torso graph

- **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}

[Ganian, Ramanujan, Sz. STACS'17, SAT'17]

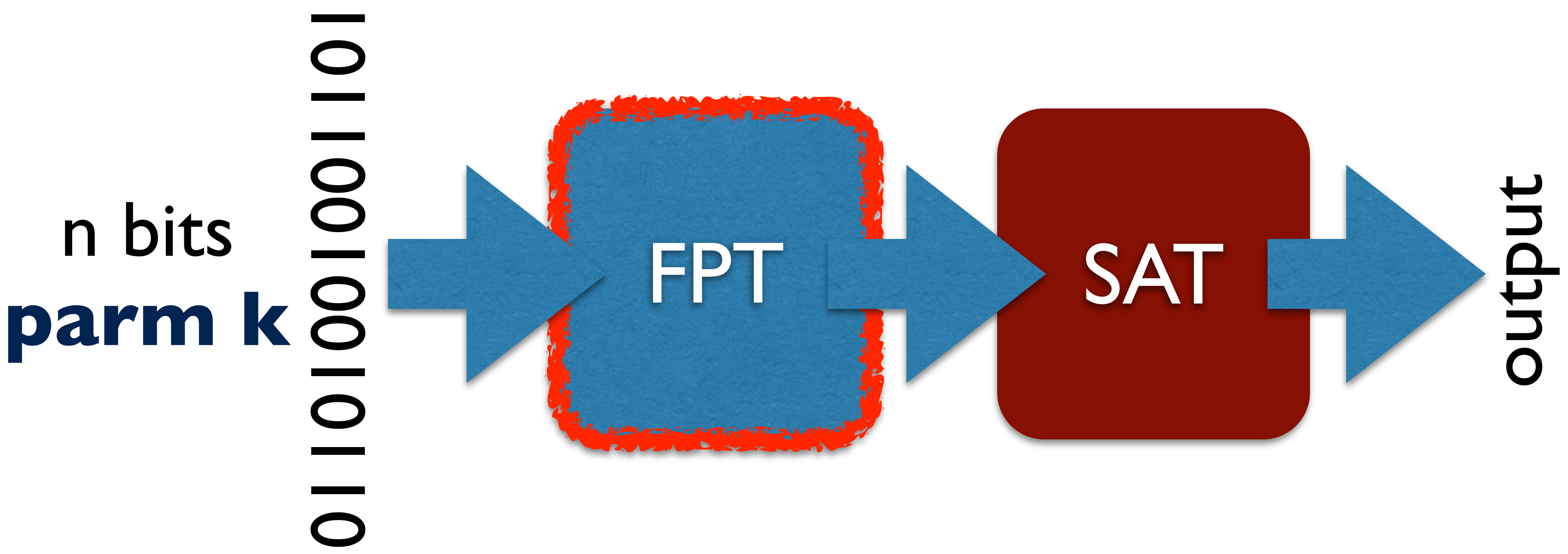
C-backdoor treewidth is FPT for C ∈ {Horn,dHorn,2CNF}

Parameter Zoo



FPT reductions to SAT

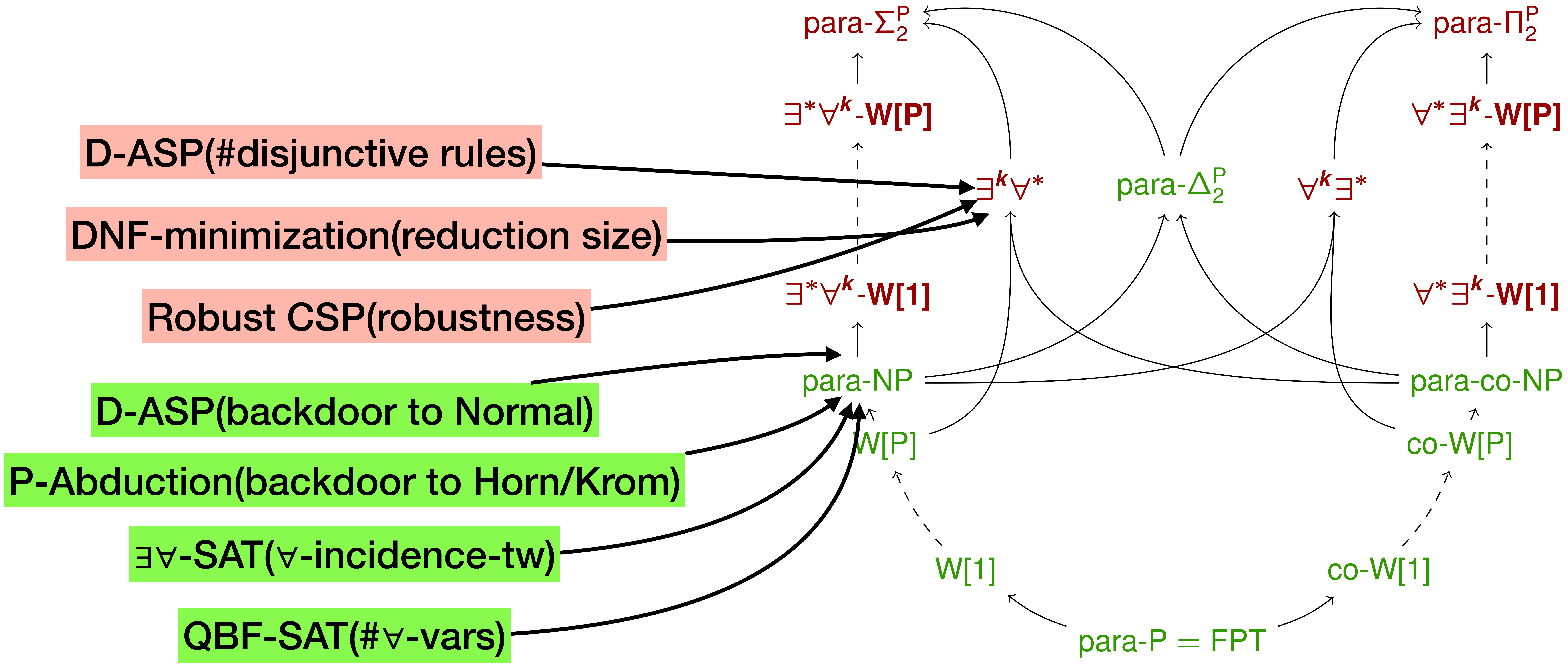
Parameterised Complexity where SAT is easy



- Combines the advantages of FPT and SAT
- parameters can be less restrictive
- breaks through barriers of classical complexity

[Pfandler, Rümmele, Sz. IJCAI'13]
[Fichte, Sz. TOCL'15]]

Results



Compendium [de Haan, Sz. Algorithms'19]

Summary

- Capturing structure in instances: correlational approach and causal approach
- Parameterized Complexity as a suitable framework for the causal approach for SAT and related reasoning problems
- Parameters: decompositions, backdoors, hybrid (backdoor treewidth)
- Dominance allows us to explore the subject systematically, relate parameters to teach other
- FPT-reductions to SAT for problems beyond NP

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”