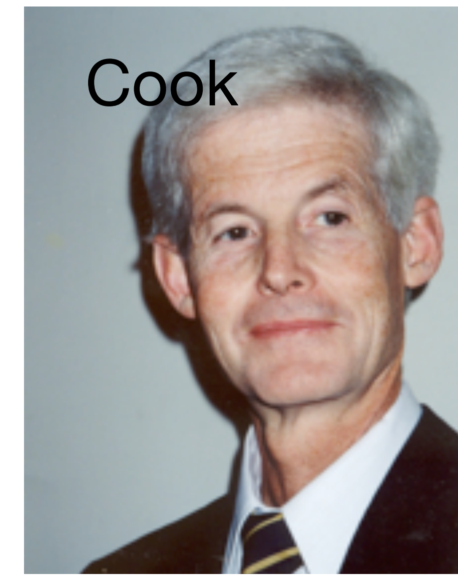
Backdoors, Satisfiability, and Problems Beyond NP

Stefan Szeider

IPEC 2014

Satisfiability (SAT)

- SAT: is a given prop. formula satisfiable?
- First problem shown to be NP-complete [Cook '71]
- Extremely well suited for representing various problems (e.g., verification, planning, …)
- Engineers started early in designing solvers, DIMACS challenge 1992
- SAT conference, annual competitions,…



DIMACS

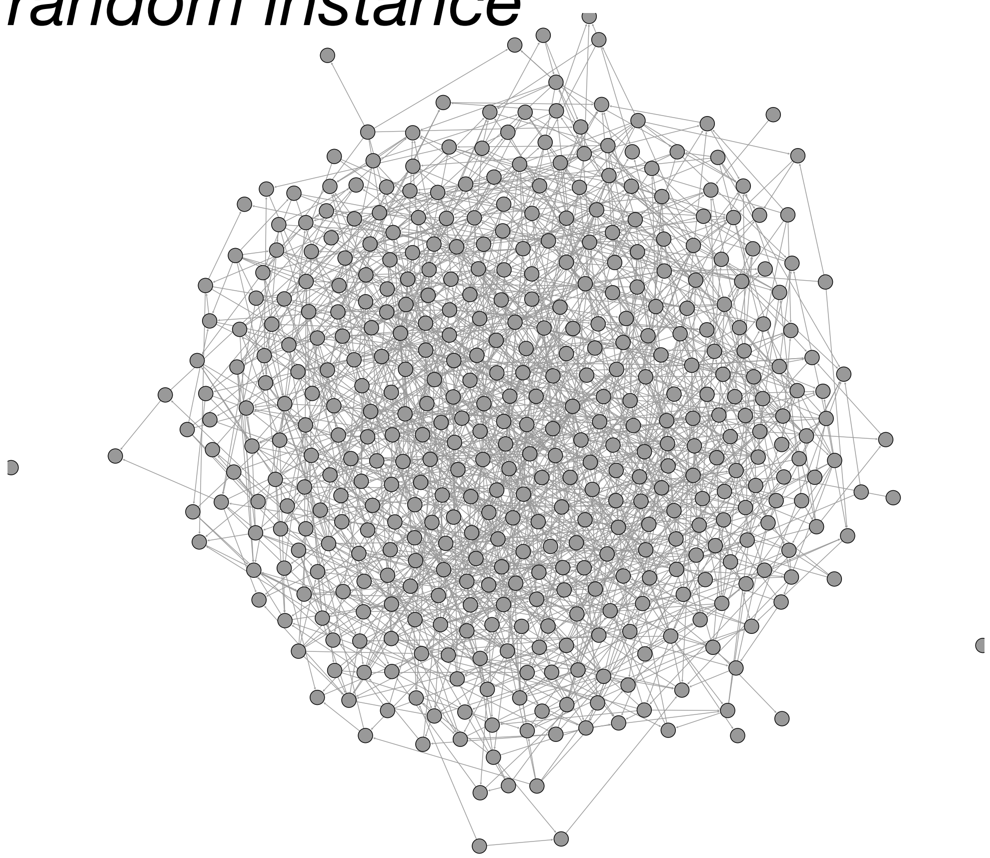
break-through ≈2000

- Heuristic techniques, e.g. **clause-learning**
- super efficient data structures like **watched literals**
- combination makes SAT solvers incredibly fast
- from about 100 vars (1990) to 1.000.000 vars
  search space from $\mathbf{10^{30}}$ to $\mathbf{10^{300.000}}$

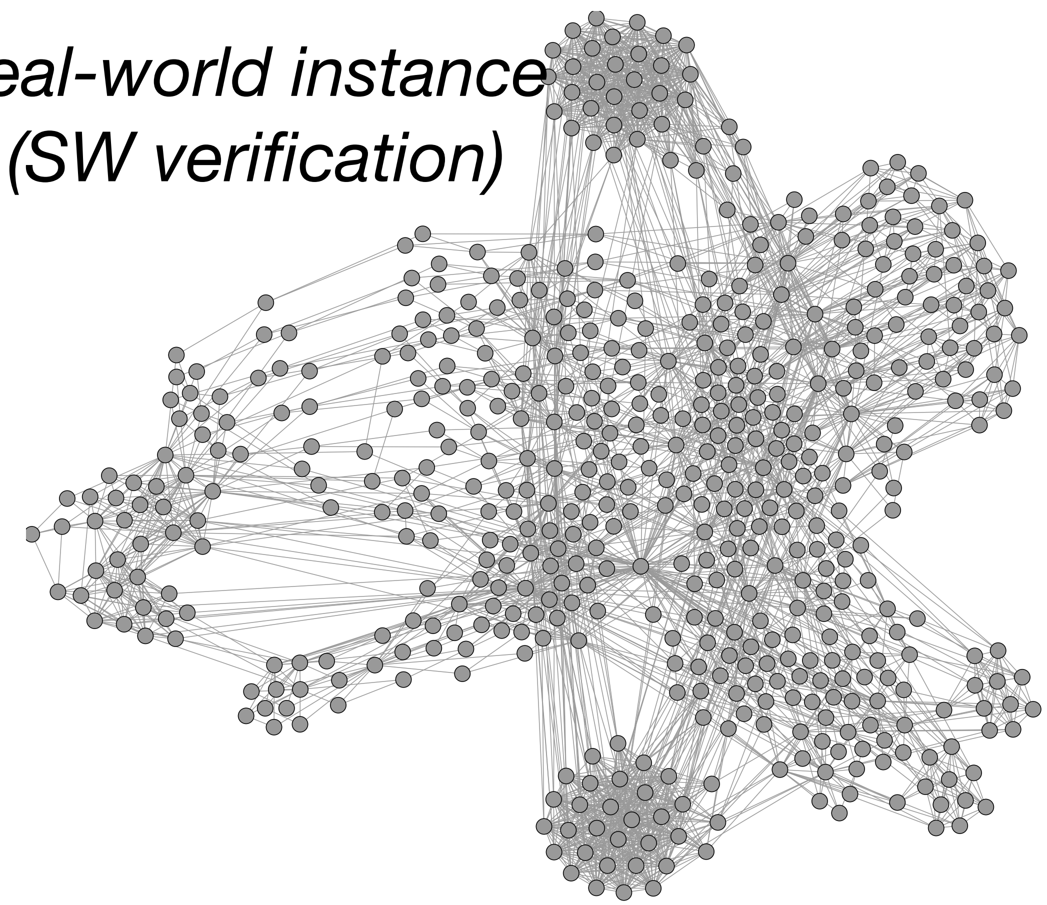
… the progress on the engineering side has been *nothing short of spectacular*
M.Vardi, C.ACM 2014

Structure Matters!

random instance



real-world instance (SW verification)



- SAT solvers are good at practical instances
- SAT solvers are not good at random instances

3SAT time bounds

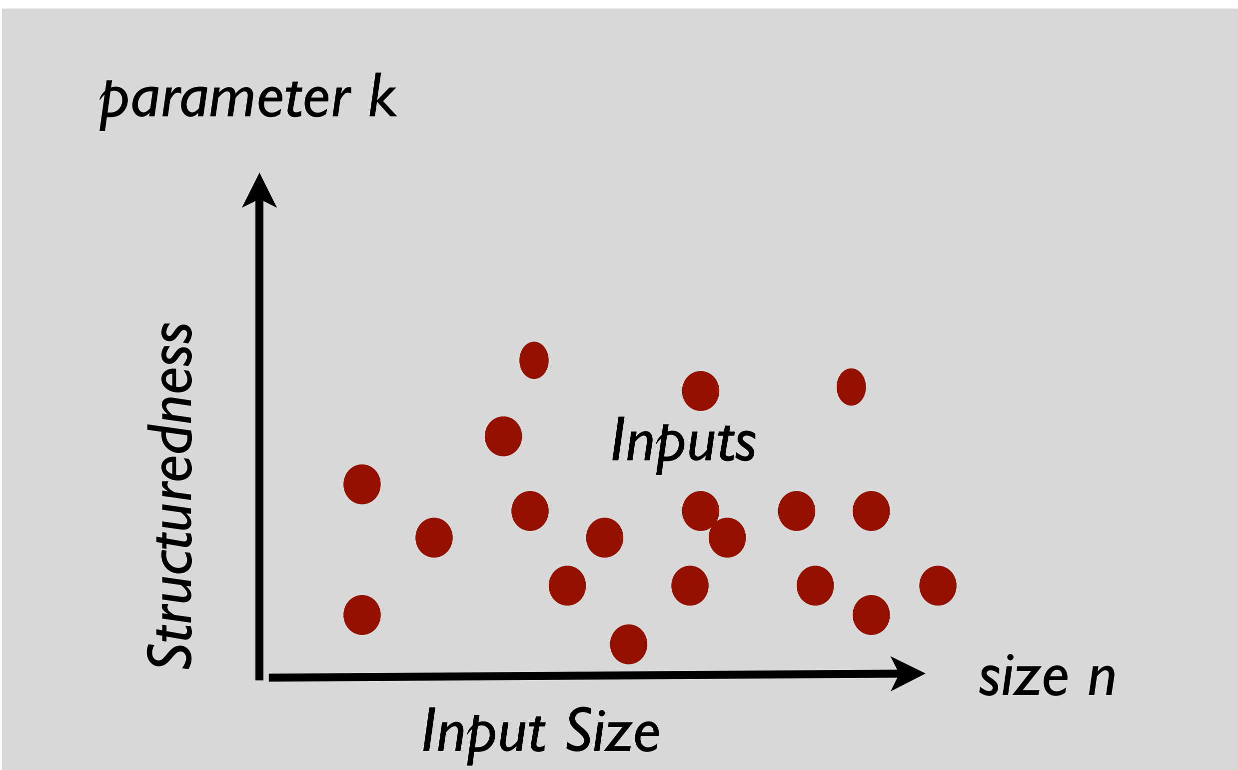
2	trivial
1.333	1999 (Schöning)
1.3302	2002
1.3290	2003
1.3280	2003
1.324	2010 (Hertli)

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

The Gap

Theory n<250 vs Practise n>1.000.000

Parameterize!



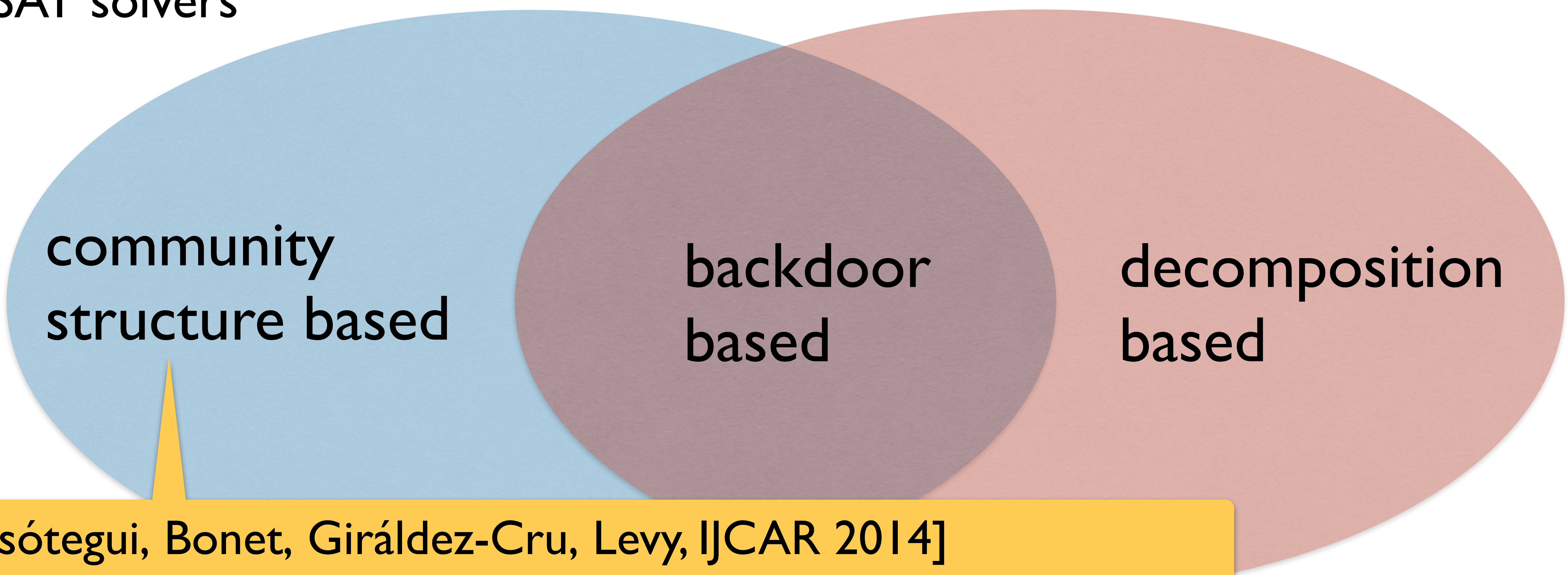
- Parameterized Complexity offers a suitable framework [Downey&Fellows 1999]
- Fixed-Parameter Tractability (FPT) time f(k) · poly(n)

Is there an FPT parameter that explains the success of SAT solvers?

The quest for the "right" parameter…

parms that describe (to some extend) the behaviour of SAT solvers

parms for which SAT is FPT



[Ansótegui, Bonet, Giráldez-Cru, Levy, IJCAR 2014]
[Newsham, Ganesh, Fischmeister, Audemard, Simon, SAT 2014]

Part I

Backdoors to P

Backdoors

[Williams, Gomes, Selman 2003]

Backdoor: small set of variables such that instantiating the variables puts the instance on an island of tractability

Similar to modulators for graphs.

Islands of Tractability (or base classes)

- A class C of CNF formulas is an **island of tractability** if
  - recognition of C is polynomial
  - SAT-decision for C is polynomial
  - C is closed under partial assignments

Examples

- Class of **Horn** formulas (all clauses contain at most 1 positive literal)
- Class of **Krom** (or **2CNF**) formulas (all clauses contain at most 2 literals)
- Class of **Renamable Horn** formulas (can be made Horn by flipping polarity of variables)
- **Acyclic** formulas (the incidence graph is a **forest**, generalization: of **bounded tw**)
- Class of instances decided by a polynomial-time **subsolver**: unit propagation, pure literal elimination, trivial decision, but no branching.

Some Notation

- CNF formula as set of clauses
  $F=\{\{\neg x,y\},\{x,\neg z,u\},\{x,\neg u\}\}$

- partial assignment:
  mapping τ from $S \subseteq var(F)$ to $\{0,1\}$,
  e.g. ,$\tau=\{x\mapsto 1,y\mapsto 0\}$

- Applying a partial assignment:
  $\mathbf{F[\tau]}=\{\{\neg x,\cancel{y}\},\cancel{\{x,\neg z,u\}},\{x,\neg u\}\}=\{\{\neg x\},\{x,\neg u\}\}$

- Deleting variables:
  $\mathbf{F-S}=\{\{\cancel{\neg x},\cancel{y}\},\{\cancel{x},\neg z,u\},\{\cancel{x},\neg u\}\}=\{\{\},\{\neg z,u\},\{\neg u\}\}\supseteq F[\tau]$

Types of Backdoors

- Let C be an island of tractability, F a CNF formula and $B \subseteq var(F)$.
- B can be a **strong** C-backdoor, a **weak** C-backdoor, or a **deletion** C-backdoor of F, according tho the following definitions.

Strong Backdoors

- B is a **strong C-backdoor** of F if for **all** truth assignments τ:B→ {0,1} we have F[τ]∈ C.
- If we know a strong backdoor of size k, then we can decide the satisfiability in time $O^*(2^k)$.

Weak Backdoors

- B is a **weak C-backdoor** of F if for **some** truth assignment τ:B→ {0,1} we have F[τ]∈ C ∩ SAT.
- If we know a weak backdoor of size k then we can find a satisfying assignment in time $O^*(2^k)$.

Deletion Backdoors

- B is a **deletion C-backdoor** of F if $F-B \in C$
- If C is **clause-induced** (i.e, $F \in C$ and $F' \subseteq F$, then $F' \in C$), then each deletion C-backdoor is a strong C-backdoor.
  (follows from $F-B \supseteq F[\tau]$)

Origins of Backdoors

- [Williams, Gomes, Selman 2003] introduced backdoors as a theoretical tool to explain:
  - heavy-tailed runtime of backtrack-based solvers
  - effectiveness of random restarts
- The solver might get lucky and find the key variables after a restart
- Industrial instances (often) have small backdoors, random instances around the threshold don't.
- [Crama, Ekin, Hammer, DAM 1997] considered a similar concepts under different names.

Algorithmic use of Backdoors

[Nishimura, Ragde, Sz 2004]

1. **Backdoor Detection:**
   Find small backdoor (say of size at most k)
2. **Backdoor Evaluation:**
   Use the backdoor to decide satisfiability (or count number of satisfying assignments)

Backdoor Detection

[Nishimura, Ragde, Sz 2004]

- **Strong C-Backdoor Detection**

  Instance: a CNF formula F, integer $k \geq 0$

  Question: does F have a strong C-backdoor of size $\leq k$?

  Parameter: k

- **Weak C-Backdoor Detection** (analogue)

- **Deletion C-Backdoor Detection** (analogue)

Horn and Krom

complexity chart:

Strong	Deletion	Weak
FPT	FPT	W[2]-h

Proof

- for Horn/Krom strong and deletion backdoors are the same!
  - Horn/Krom is clause-induced,
    hence **deletion ⇒ strong**
  - single clause obstruction cannot be eliminated by satisfying the clause, it must be eliminated by deletion,
    hence **strong ⇒ deletion**

deletion Horn-bd = VC

- Given F, we construct a graph G=(V,E)
- V=var(F)
- uv $\in$ E iff there is a clause containing u and v positively.
- Deletion Horn-backdoors of F are exactly the vertex covers of G.
- Use VC algorithm ($O^*(1.2738^k)$)
  [Chen, Kanj, Xia 2010]

deletion Krom-bd= 3HS

- Given F, we construct a 3-uniform hypergraph H=(V,E), V=var(F)
- uvw $\in$ E iff there is a clause containing these three variables positively or negatively
- Deletion Horn-backdoors of F are exactly the hitting sets of H.
- Use 3HS algorithm $O^*(2.27^k)$ [Niedermeier, Rossmanith 2003]

Renamable Horn

Strong	Deletion	Weak
W[2]-h	FPT	W[2]-h

- Deletion RHorn-Backdoor Detection can be fpt-reduced to 2SAT-Deletion
(make 2CNF formula satisfiable by deleting k clauses) [Gottlob, Sz. 2006]
- 2SAT-Deletion is FPT [Razgon, O'Sullivan 2009]

Q-Horn

f:vars→ [0,1]

$\Sigma_{x \in C}\, f(x) + \Sigma_{\neg x \in C}\, 1\text{-}f(x) \leq 1 \quad (\forall \text{ clauses } C)$

- **Q-Horn ⊇ RHorn, Horn, Krom**
  in terms of linear equations
  [Boros, Crama and Hammer 1990]

Strong	Deletion	Weak
W[2]-h	FPT-apx	W[2]-h

- Deletion Q-Horn Detection is FPT-approximable (algorithm finds backdoor of size $\leq k^2+k$) [Gaspers, Ordyniak, Ramanujan, Saurabh, Sz. 2013]

Subsolvers

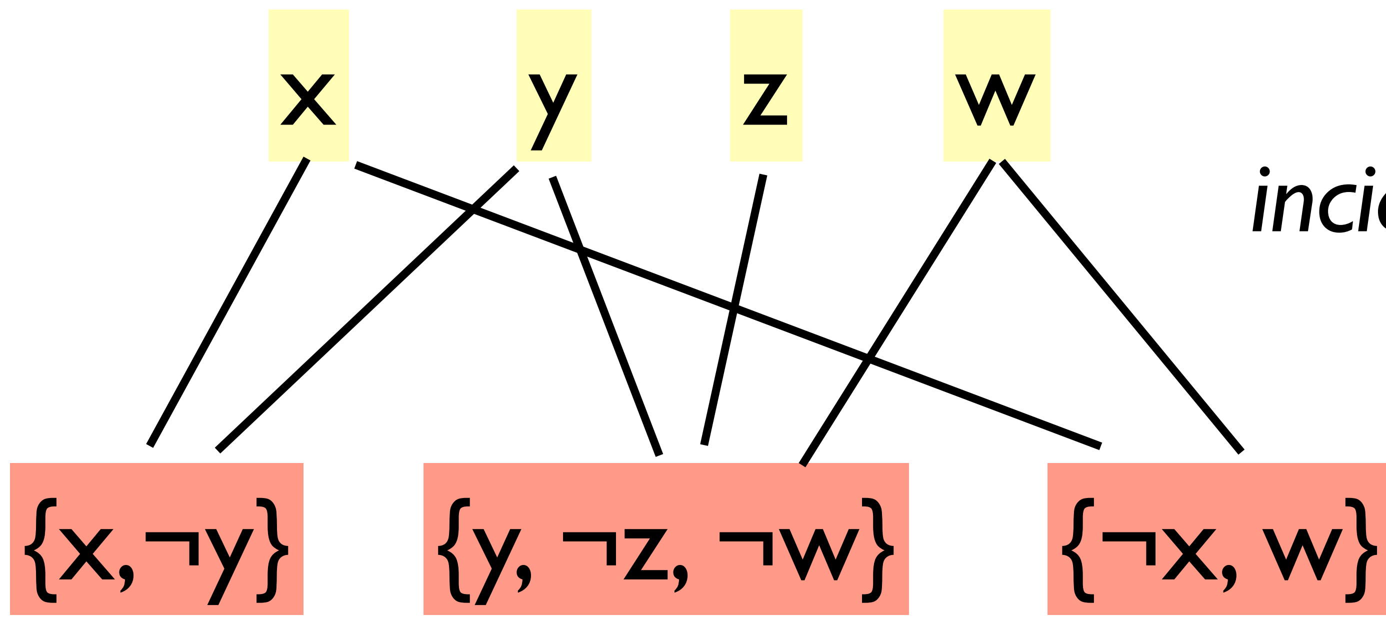
- Let C be a base class defined by subsolver UP, PL, UP+PL.
- C is not clause-induced, so deletion backdoors don't help!

Strong	Deletion	Weak
W[P]-c	n/a	W[P]-c

- Strong/Weak C-Backdoor Detection is W[P]-complete. [Sz 2005].

Acyclic Formulas

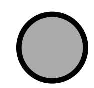
F={{x ,¬y},{y ,¬z, ¬w}, {¬x, w}}



incidence graph

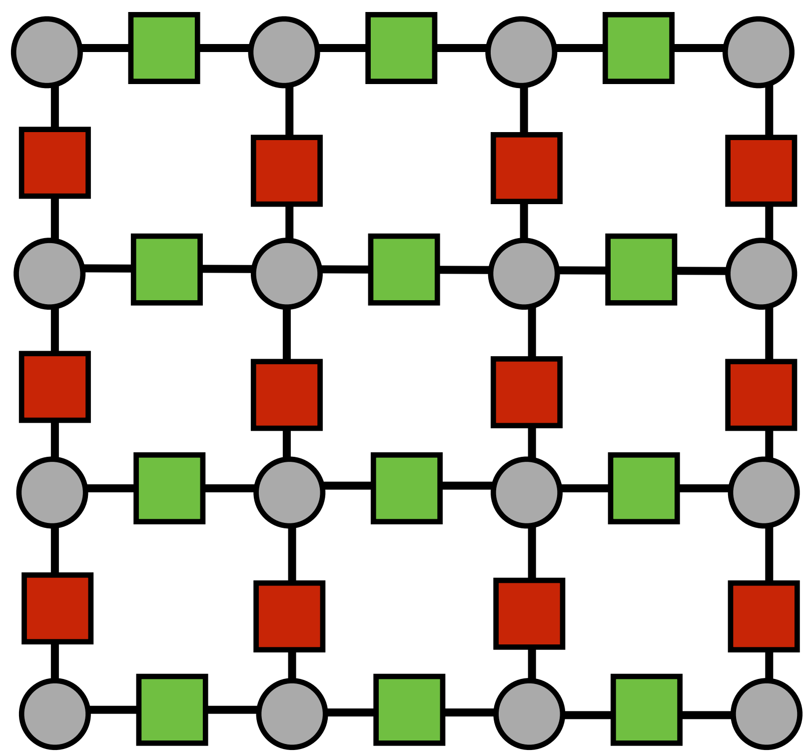
- A formula is Acyclic if it’s incidence graph is a forest.
- #SAT is solvable in linear-time for acyclic formulas

The power of partial assignments



x

appears positively in green clauses
appears negatively in red clauses



x=true

x=false

- {x} is a strong Acyclic-backdoor
- deletion Acyclic-backdoor needs to be large

Acyclic Formulas

Strong	Deletion	Weak
FPT-apx	FPT	W[2]-h

- Deletion Acyclic-Backdoor Detection can be solved by FVS
- Strong Acyclic-Backdoor Detection is fpt-approximable [Gaspers, Sz. ICALP'12]

Algorithm Outline

A. small feedback vertex set
   - small treewidth, dynamic programming

B. many disjoint cycles
   - find an **essential set S*** of size at most f(k) that intersects with every backdoor set of size k.

FPT-Approximation

1. Given (F, k) with many disjoint cycles
2. compute an essential set S*
3. for all x in S* recursively try (F[x=0],k-1) and (F[x=1],k-1)
4. If both branches produce backdoor sets X_0, X_1, then output $X_0 \cup X_1 \cup \{x\}$

Bounded Treewidth

- Let TW[t] be the class of formulas whose incidence graph has treewidth $\leq$ t.
- Strong TW[t]-Backdoor Detection is fpt-approximable [Gaspers, Sz. FOCS'13]

Strong	Deletion	Weak
FPT-apx	FPT	W[2]-h

Further Questions

- Weak backdoor detection is W[2]-hard for most base classes …. how about inputs in 3CNF? [Gaspers, Sz. 2012] [Misra, Ordyniak, Raman, Sz. SAT'13]
- How about strong backdoors where each assignment can put the formula into a different base class "heterogeneous backdoors" [Gaspers, Misra, Ordyniak,Sz, Zivny AAAI'14]

Part II

Backdoors to NP

Reductions to SAT

- Every NP-complete problem can be reduced in polytime to SAT
- In many cases this is a very practical solution, because of the power of today's SAT solvers

Above NP

MINMAX SAT

NOT-ALL-EQUAL∀∃3SAT

∀∃3SAT

MONOTONE MINIMUM WEIGHT WORD

GENERALIZED 3-CNF CONSISTENCY

IRREDUNDANT

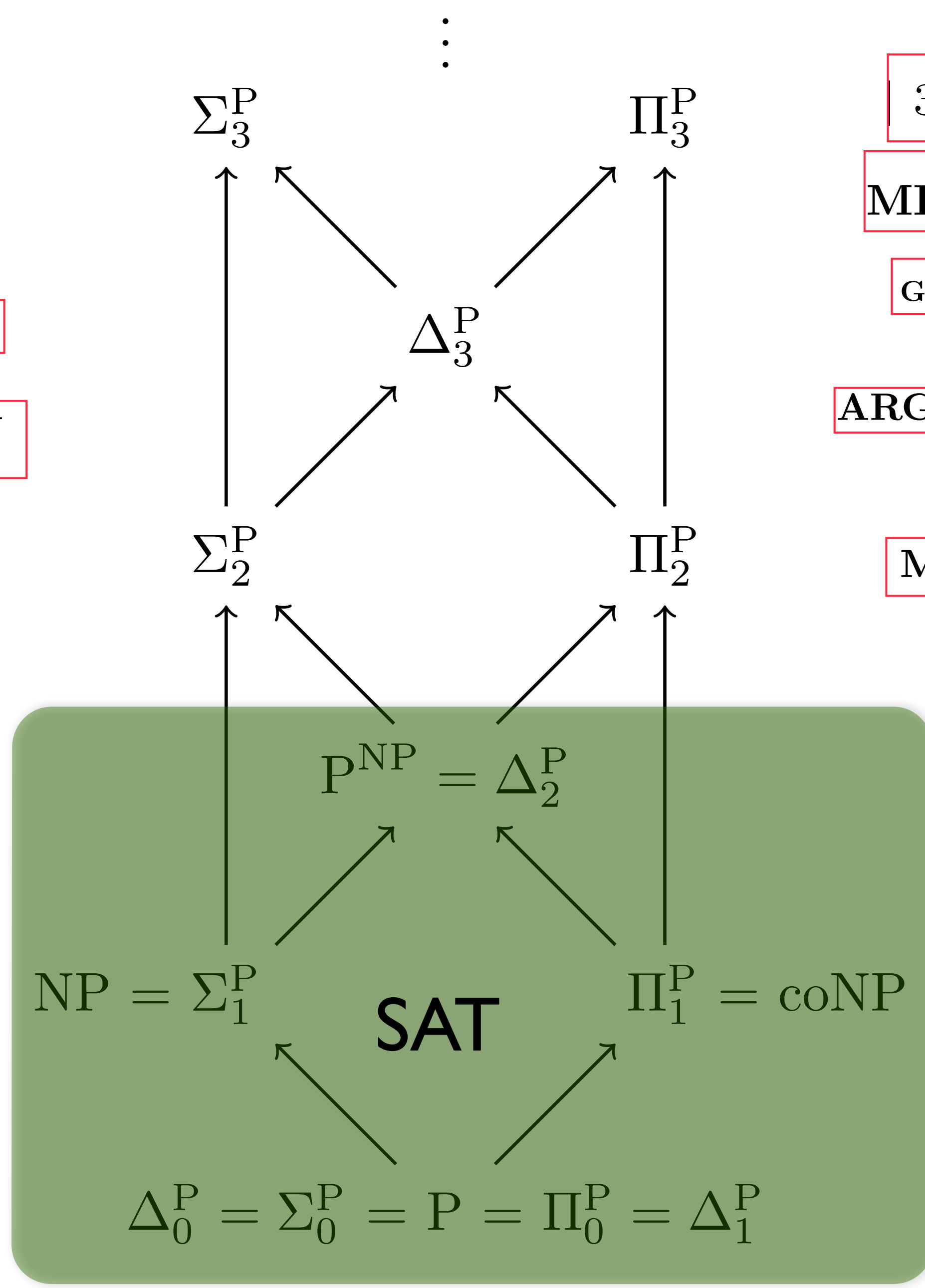
MAXIMUM TERM DELETION

SHORT CNF

MAXIMUM LITERAL DELETION

SHORTEST IMPLICANT

CIRCUIT RESTRICTION



LIST CHROMATIC NUMBER

3-COLORING EXTENSION

MINMAX CLIQUE

THUE NUMBER

GRAPH SATISFIABILITY

GENERALIZED SUBSET SUM

ARGUMENT COHERENCE

MINIMUM EQUIVALENT EXPRESSION

MINIMUM 3SAT DEFINING SET

GRAPH CONSISTENCY

GENERALIZED GRAPH COLORING

DYNAMIC HAMILTONIAN CIRCUIT

GENERALIZED RAMSEY NUMBER

LONGEST DIRECTED CIRCUIT

PEBBLING NUMBER

[Shaefer, Umans 2002,2008] Compendium

Example: Abductive Reasoning

C. S. Peirce
(1839–1914)

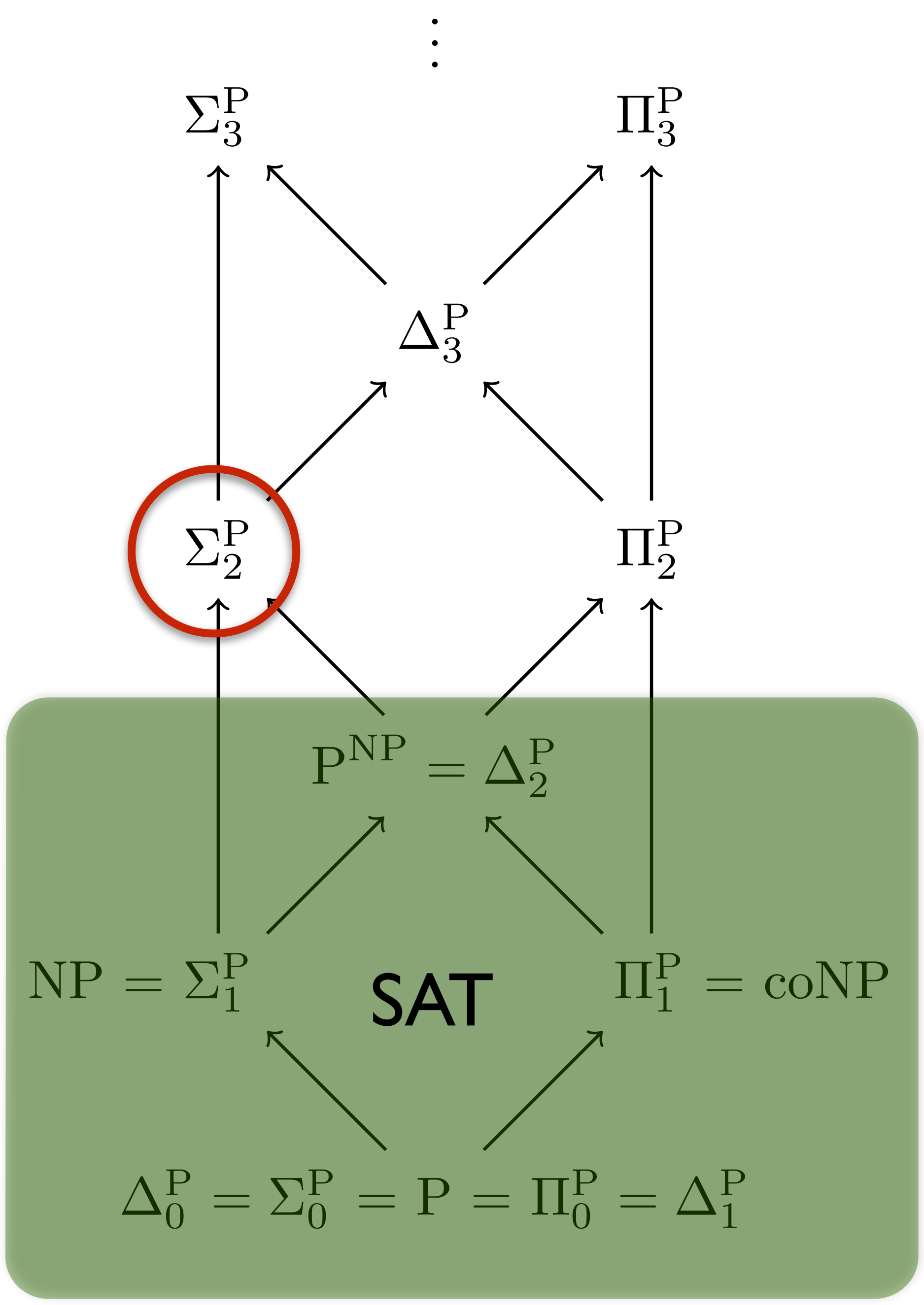
- studied by Charles Sanders Peirce
- used to generate explanations for observed symptoms and manifestations
- fundamental importance in AI, such as for logical diagnosis

The Abduction Problem

- Given:
  - a propositional formula T (the theory)
  - a set of variables M (the manifestations)
  - a set of variables H (the hypotheses)
- Task:
  - find a set $S \subseteq H$ such that that
    $S \wedge T$ is consistent and
    $S \wedge T \models M$ (all assignments that satisfy $S \wedge T$ also satisfy M)
  - "S explains M"

Abduction is Hard

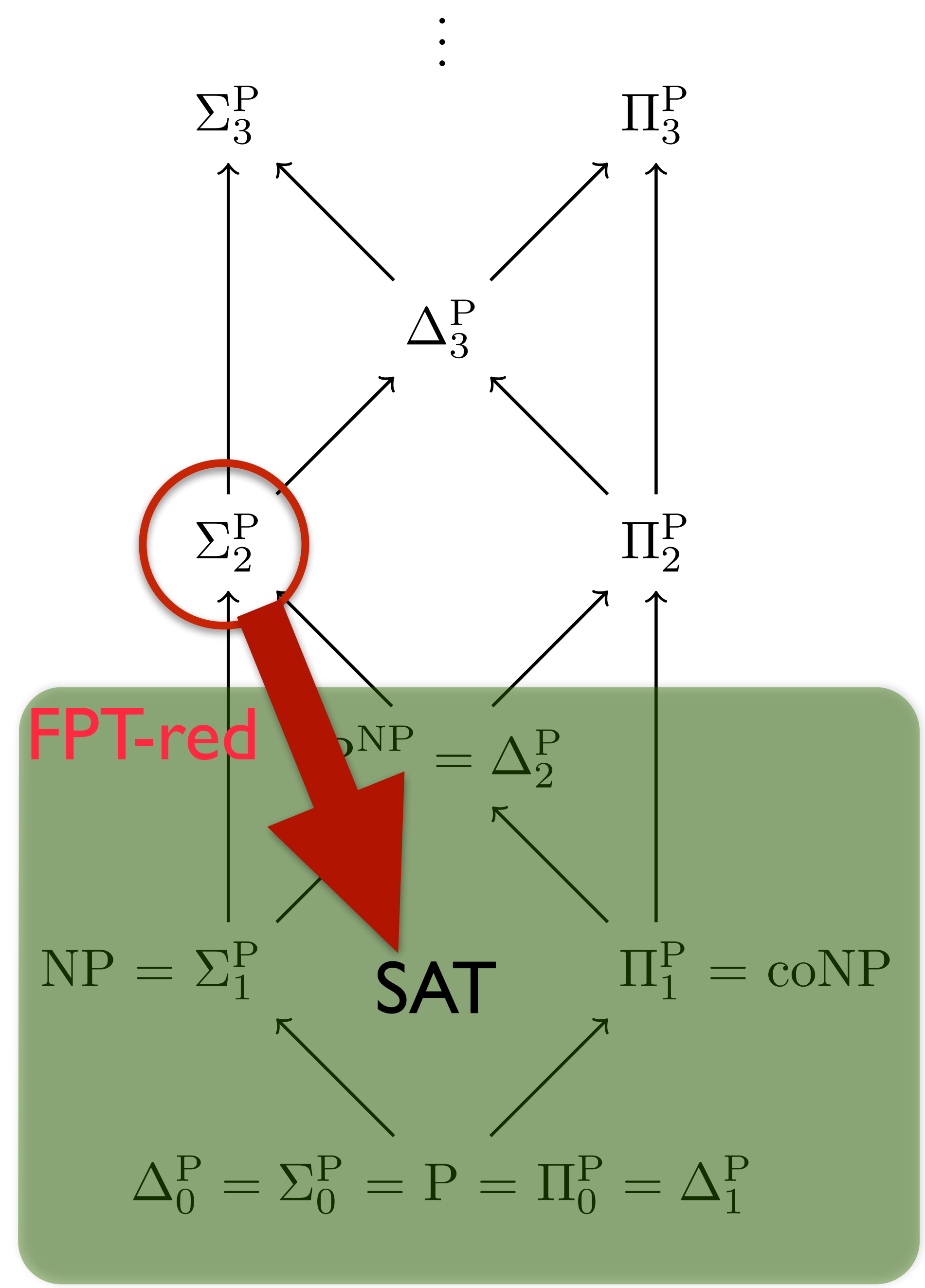
- Deciding whether an abduction instance has a solution is $\Sigma^P{}_2$ -complete [Eiter, Gottlob 1995]
- Can't be reduced in polytime to SAT
- Parameterized complexity results by [Fellows, Pfandler, Rosamond, Rümmele 2012]
  mostly hardness results,
  FPT by number of variables



Idea:
FPT-reductions to SAT

Use FPT-tractability not to **solve** the problem, but to **reduce** it to SAT!

- Parameter can be far less restrictive than for FPT-tractability
- We combine the strengths of two worlds (SAT and FPT)
- Suitable parameter?

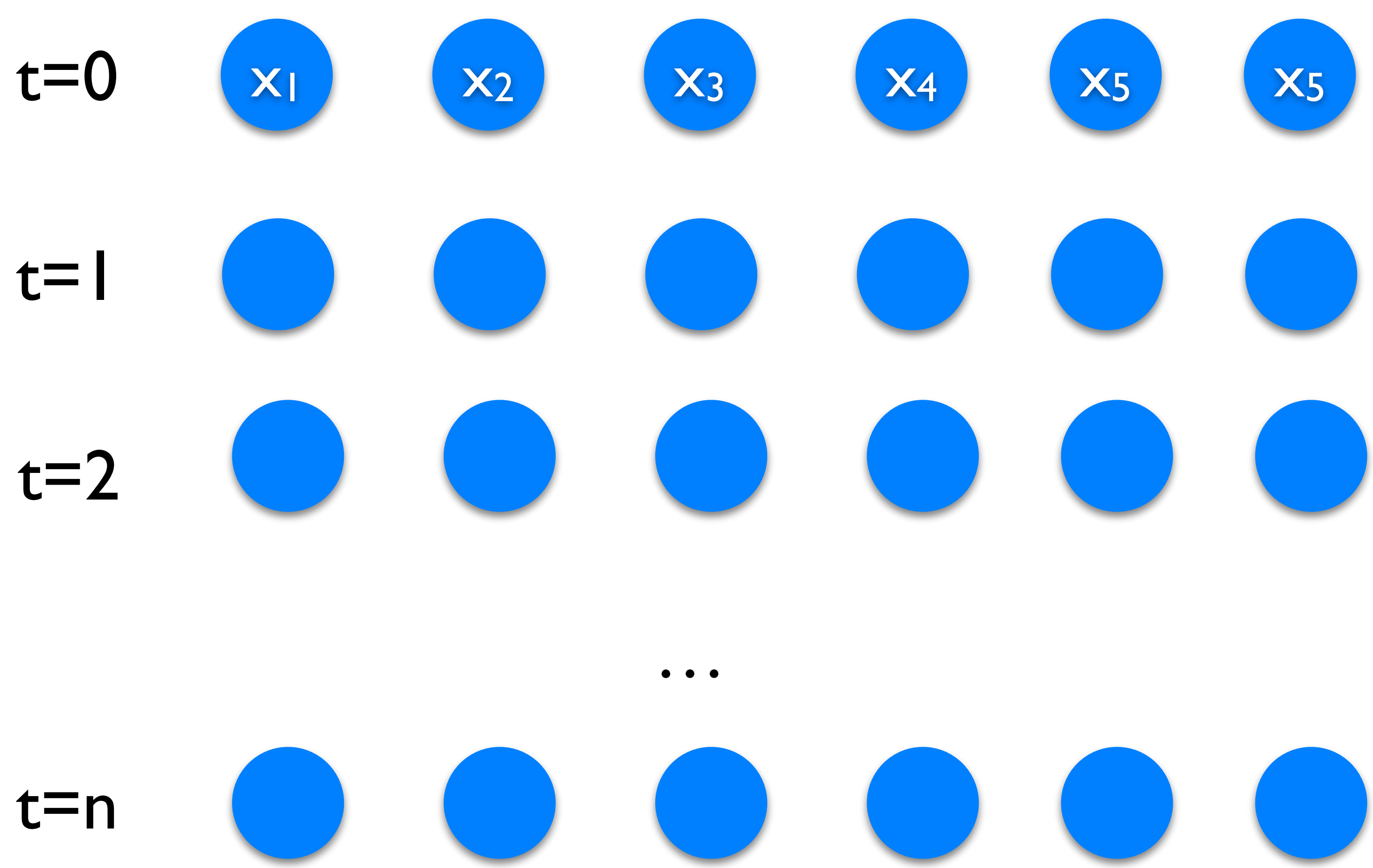


Backdoors to SAT

- If the theory is Horn, Abduction drops to NP-c [Eiter,Gottlob1995]
    - The unique minimal model of a Horn formula can be found in linear time [Dowling, Gallier 1984]
    - since it is sufficient to check $(S \wedge T) \models M$ for the **unique minimal model** of $S \wedge T$
- Use as parameter the Horn backdoor size
- Finding the backdoor is FPT (same as for SAT)
- Using the backdoor is the challenge!

Dowling-Gallier algorithm in a formula

enforced by unit clauses



unique minimal model

Take conjunction of 2^k copies of this formula, corresponding to all truths assignments to the backdoor

Results

- **Theorem:** an Abduction instance with a strong Horn backdoor of size k, can be reduced in time $O^*(2^k)$ an equivalent CNF formula.
- **Corollary:** Abduction parameterized by Horn-backdoor size is is para-NP-complete
- Hence we can break trough the barriers of classical complexity, exploiting structure in terms of parameters

Further Results

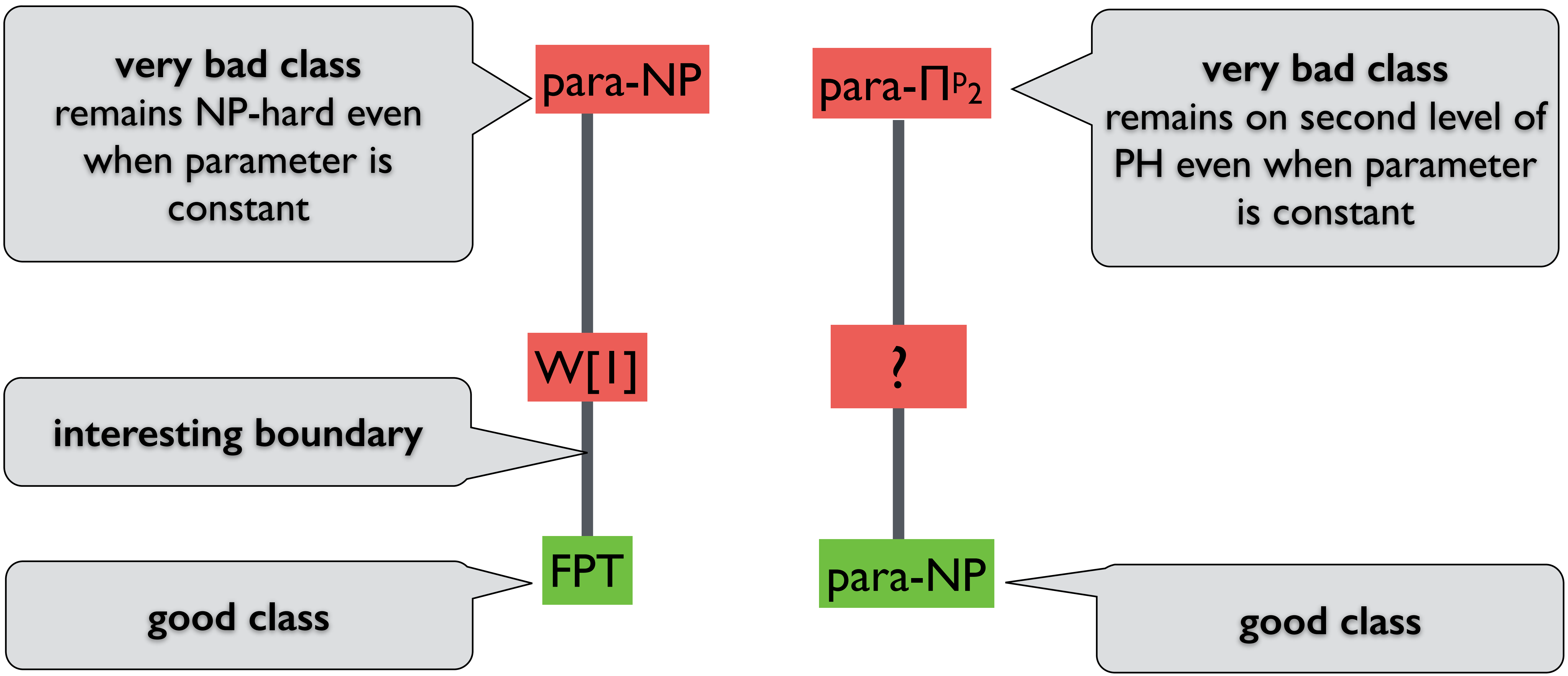
- **Abduction** par'd by backdoor to Krom is paraNP-c [Pfandler, Rümmele, Sz. IJCAI'13]
- **Disjunctive Answer Set Programming** par'd by backdoor to Normality is paraNP-c [Fichte, Sz. AAAI'13]
- **Boolean Optimization** $\exists\forall$SAT par'd by universal treewidth is paraNP-c [de Haan, Sz. SAT'14]
- in retrospect: **Bounded Model Checking** par'd by size of counterexample (industrial strength applications!) [Biere, Cimatti, Clarke, Zhu,1999] is paraNP-c

Hardness Theory

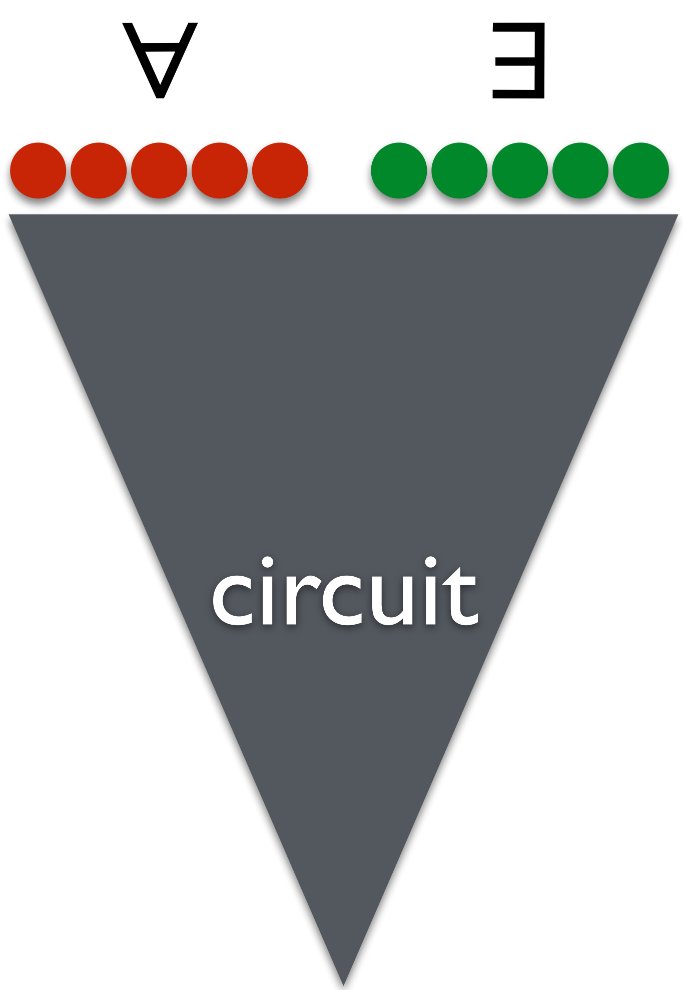
- How can we exclude that a parameterized problem is fpt-reducible to SAT?

parameterized
NP-complete problems

parameterized
Σ^P_2-complete problems



Weighted Satisfiability!

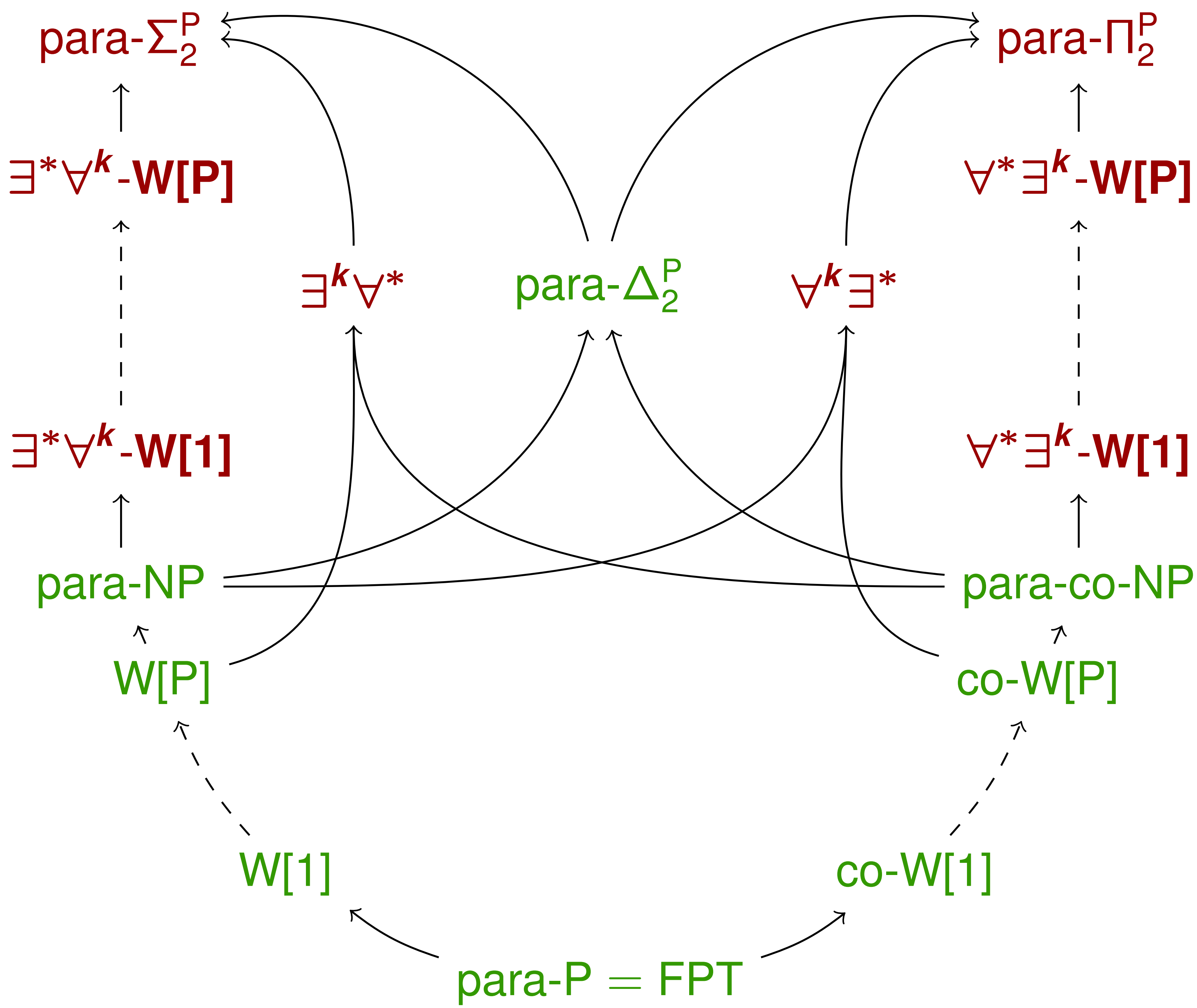


1. for all assignments to red there is an assignment of weight k to green that satisfies the circuit? $\forall^*\exists k$
2. for all assignments to red of weight k there is an assignment to green that satisfies the circuit? $\forall k\exists^*$

Is the Weft important?

- **$\forall^*\exists k$: weft is important.**
  - we end up with seemingly distinct classes $\forall^*\exists k$-W[i]
- **$\forall k\exists^*$: weft is not important.**
  - any circuit can be replaced with a 2DNF
  - a very robust class for which we have alternative characterisations in terms of alternating TMs and FO-MC

Complexity Zoo



The classes are inhabited!

- Disjunctive Answer Set Programming [Kr’14]
- Boolean Optimization [SAT’14]
- Judgement Aggregation [ComSoc’14]
- Graph Problems

Concrete Example: 3-Col-Ext

- Input: graph G with n leaves, integer m
- Question: can any 3-coloring of m of the n leaves be extended to a proper 3-coloring of G?

parameter	complexity
# uncolored leaves (m-n)	para-
# precolored leaves (m)	$\forall k \exists^*$-complete
# leaves (n)	para-NP-complete

Last Slide

- Backdoors: natural way to parameterise problems
- FPT: Backdoors to P
- para-NP: Backdoors to NP
- Interesting algorithmic and theoretical challenges

Questions?