Recursive Backdoors for SAT

Nikolas Mählmann, Sebastian Siebertz, Alexandre Vigny

23.08.2021

University of Bremen

The SAT Problem

Input: a formula ϕ of propositional logic

Output: does there exists a satisfying assignment for ϕ?

The SAT Problem

Input: a formula ϕ of propositional logic
Output: does there exists a satisfying assignment for ϕ?

Examples:

$$(x_- \vee y_+) \wedge (x_+ \vee z_+)$$

The SAT Problem

Input: a formula ϕ of propositional logic

Output: does there exists a satisfying assignment for ϕ?

Examples:

$(x_- \vee y_+) \wedge (x_+ \vee z_+)$ is SAT

The SAT Problem

Input: a formula ϕ of propositional logic

Output: does there exists a satisfying assignment for ϕ?

Examples:

$(x_- \vee y_+) \wedge (x_+ \vee z_+)$ is SAT

$(x_+ \vee y_+) \wedge (x_-) \wedge (y_-)$

The SAT Problem

Input: a formula ϕ of propositional logic

Output: does there exists a satisfying assignment for ϕ?

Examples:

$(x_- \vee y_+) \wedge (x_+ \vee z_+)$ is SAT

$(x_+ \vee y_+) \wedge (x_-) \wedge (y_-)$ is UNSAT

Tractable Base Classes

There exist tractable base classes of formulas:

- 2CNF: each clause contains at most two literals
- Horn: each clause contains at most one positive literal

However real world instances are often less homogenous!

Tractable Base Classes

There exist tractable base classes of formulas:

- 2CNF: each clause contains at most two literals
- Horn: each clause contains at most one positive literal

However real world instances are often less homogenous!

$$\phi = (x_{1_-} \vee x_{2_-} \vee x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \dots$$

Tractable Base Classes

There exist tractable base classes of formulas:

- 2CNF: each clause contains at most two literals
- Horn: each clause contains at most one positive literal

However real world instances are often less homogenous!

$$\phi = (x_{1_-} \vee x_{2_-} \vee x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \dots$$

ϕ is not in 2CNF but very *close* to 2CNF.

Backdoors for SAT

A *backdoor* B of ϕ to $\mathcal{C}$ is a set of variables that reduces ϕ to a formula from $\mathcal{C}$ **no matter which assignment is chosen**.

Backdoors for SAT

A *backdoor* B of ϕ to $\mathcal{C}$ is a set of variables that reduces ϕ to a formula from $\mathcal{C}$ **no matter which assignment is chosen**.

Example:

$$\phi = (x_{1_-} \vee x_{2_-} \vee x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \ldots$$

$\{x_1, x_2\}$ is a backdoor of ϕ to 2CNF.

Backdoors for SAT

A *backdoor* B of ϕ to $\mathcal{C}$ is a set of variables that reduces ϕ to a formula from $\mathcal{C}$ **no matter which assignment is chosen**.

Example:

$$\phi = (x_{1_-} \vee x_{2_-} \vee x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \dots$$

$\{x_1, x_2\}$ is a backdoor of ϕ to 2CNF.

$$\phi[x_{1_+}, x_{2_+}] =$$

$$\phi[x_{1_-}, x_{2_+}] =$$

$$\phi[x_{1_+}, x_{2_-}] =$$

$$\phi[x_{1_-}, x_{2_-}] =$$

Backdoors for SAT

A *backdoor* B of ϕ to $\mathcal{C}$ is a set of variables that reduces ϕ to a formula from $\mathcal{C}$ **no matter which assignment is chosen**.

Example:

$$\phi = (x_{1_-} \vee x_{2_-} \vee x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \dots$$

$\{x_1, x_2\}$ is a backdoor of ϕ to 2CNF.

$$\phi[x_{1_+}, x_{2_+}] = (x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \dots$$

$$\phi[x_{1_-}, x_{2_+}] =$$

$$\phi[x_{1_+}, x_{2_-}] =$$

$$\phi[x_{1_-}, x_{2_-}] =$$

Backdoors for SAT

A *backdoor* B of ϕ to $\mathcal{C}$ is a set of variables that reduces ϕ to a formula from $\mathcal{C}$ **no matter which assignment is chosen**.

Example:

$$\phi = (x_{1_-} \vee x_{2_-} \vee x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \ldots$$

$\{x_1, x_2\}$ is a backdoor of ϕ to 2CNF.

$$\phi[x_{1_+}, x_{2_+}] = (x_{3_+} \vee x_{4_+}) \wedge (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \ldots$$
$$\phi[x_{1_-}, x_{2_+}] = (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \ldots$$
$$\phi[x_{1_+}, x_{2_-}] = (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \ldots$$
$$\phi[x_{1_-}, x_{2_-}] = (x_{4_+} \vee x_{5_+}) \wedge (x_{5_+} \vee x_{6_+}) \wedge \ldots$$

Using Backdoors to Solve SAT

Algorithm: Given a backdoor of ϕ of size k to some tractable class $\mathcal{C}$, test every of the 2^k possible assignments.

Runtime complexity:

$$2^k \cdot \mathit{poly}(|\phi|)$$

Using Backdoors to Solve SAT

Algorithm: Given a backdoor of ϕ of size k to some tractable class $\mathcal{C}$, test every of the 2^k possible assignments.

Runtime complexity:

$$2^k \cdot poly(|\phi|)$$

Fixed Parameter Tractability: Running times of the form:

$$\mathcal{O}(f(k) \cdot |\phi|^c)$$

are efficient for small k.

Using Backdoors to Solve SAT

Algorithm: Given a backdoor of ϕ of size k to some tractable class $\mathcal{C}$, test every of the 2^k possible assignments.

Runtime complexity:

$$2^k \cdot poly(|\phi|)$$

Fixed Parameter Tractability: Running times of the form:

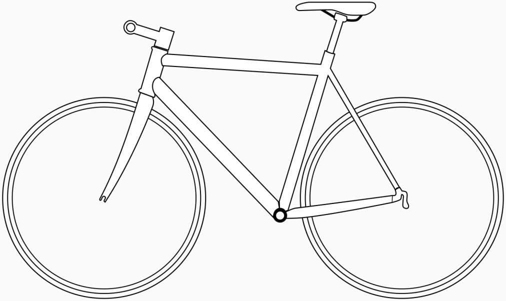
$$\mathcal{O}(f(k) \cdot |\phi|^c)$$

are efficient for small k.

There exists fpt backdoor detection algorithms to 2CNF, Horn, ...

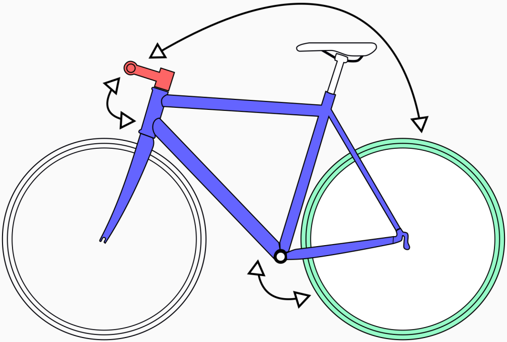
Motivation for Recursive Backdoors

`handlebars`	:	{`straight, riser, drops, wide`}
`frameset`	:	{`city, racing, mtb`}
`tire width`	:	{`21mm, 23mm, 28mm, 30mm, 35mm, 50mm`}



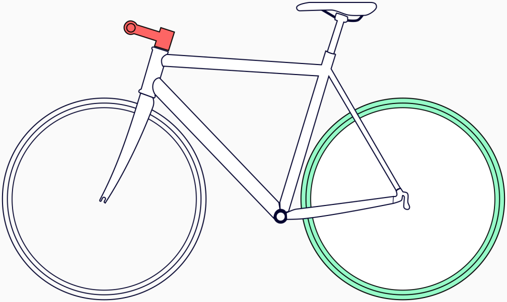
Motivation for Recursive Backdoors

handlebars	:	{straight, riser, drops, wide}
frameset	:	{city, racing, mtb}
tire width	:	{21mm, 23mm, 28mm, 30mm, 35mm, 50mm}



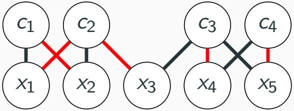
Motivation for Recursive Backdoors

```
handlebars  :  {straight, riser, drops}
frameset    :  racing
tire width  :  {21mm, 23mm, 28mm}
```



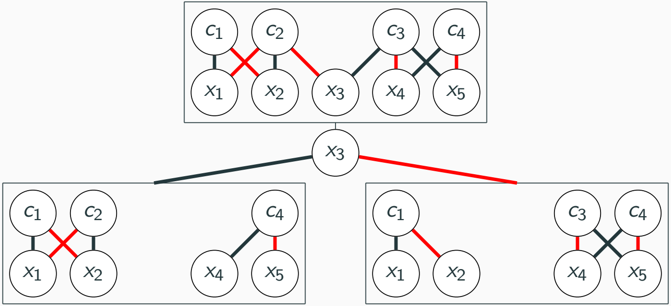
Recursive Backdoors

$$(x_{1_+} \vee x_{2_-}) \wedge (x_{1_-} \vee x_{2_+} \vee x_{3_-}) \wedge (x_{3_+} \vee x_{4_-} \vee x_{5_+}) \wedge (x_{4_+} \vee x_{5_-})$$



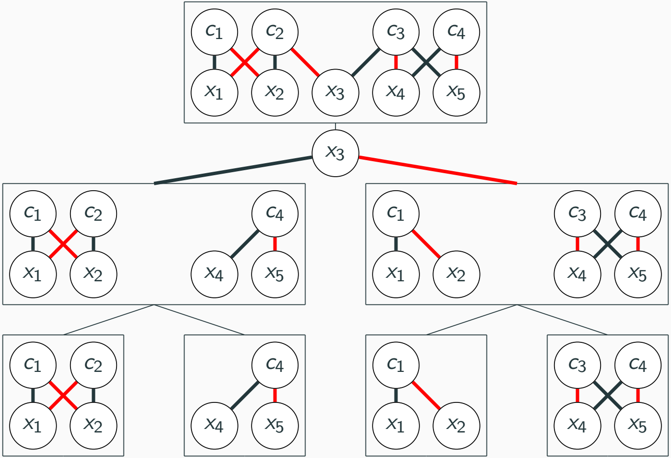
Recursive Backdoors

$$(x_{1_+} \vee x_{2_-}) \wedge (x_{1_-} \vee x_{2_+} \vee x_{3_-}) \wedge (x_{3_+} \vee x_{4_-} \vee x_{5_+}) \wedge (x_{4_+} \vee x_{5_-})$$



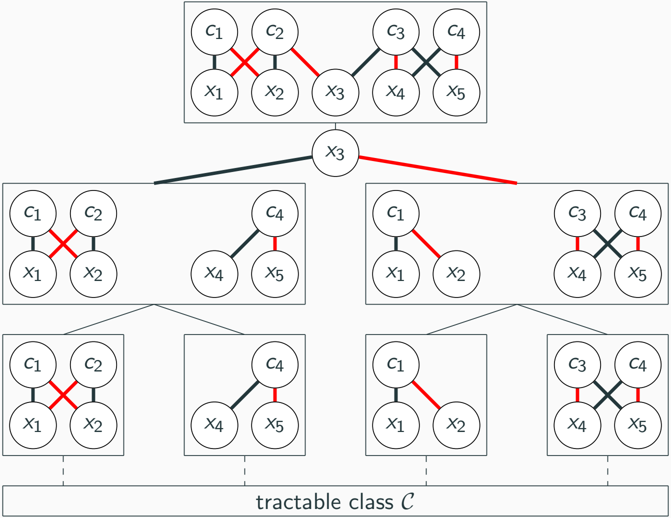
Recursive Backdoors

$$(x_{1_+} \vee x_{2_-}) \wedge (x_{1_-} \vee x_{2_+} \vee x_{3_-}) \wedge (x_{3_+} \vee x_{4_-} \vee x_{5_+}) \wedge (x_{4_+} \vee x_{5_-})$$



Recursive Backdoors

$$(x_{1_+} \vee x_{2_-}) \wedge (x_{1_-} \vee x_{2_+} \vee x_{3_-}) \wedge (x_{3_+} \vee x_{4_-} \vee x_{5_+}) \wedge (x_{4_+} \vee x_{5_-})$$



Recursive Backdoor Depth

Definition (Mählmann, Siebertz, Vigny)

$$\mathrm{rbd}_{\mathcal{C}}(G) = \begin{cases} \underline{\text{if } G \in \mathcal{C}\text{:}} \\ 0 \end{cases}$$

Recursive Backdoor Depth

Definition (Mählmann, Siebertz, Vigny)

$$\mathrm{rbd}_{\mathcal{C}}(G) = \begin{cases} \underline{\text{if } G \in \mathcal{C}\text{:}} \\ 0 \\ \\ \underline{\text{if } G \notin \mathcal{C} \text{ and } G \text{ is connected:}} \\ 1 + \min_{x \in \mathrm{var}(G)} \max_{\star \in \{+,-\}} \mathrm{rbd}_{\mathcal{C}}(G[x_\star]) \end{cases}$$

Recursive Backdoor Depth

Definition (Mählmann, Siebertz, Vigny)

$$\mathrm{rbd}_{\mathcal{C}}(G) = \begin{cases} \underline{\text{if } G \in \mathcal{C}\text{:}} \\ 0 \\ \\ \underline{\text{if } G \notin \mathcal{C} \text{ and } G \text{ is connected:}} \\ 1 + \min_{x \in \mathrm{var}(G)} \max_{\star \in \{+,-\}} \mathrm{rbd}_{\mathcal{C}}(G[x_\star]) \\ \\ \underline{\text{otherwise:}} \\ \max \{ \mathrm{rbd}_{\mathcal{C}}(H) \ : \ H \text{ connected component of } G \} \end{cases}$$

Measuring RBs

depth of a RB $\hat{=}$ maximal number of variables on a path between the root and a leaf

Measuring RBs

depth of a RB $\hat{=}$ maximal number of variables on a path between the root and a leaf

RBs with a limited depth can contain an **unbounded** number of variables!

Measuring RBs

depth of a RB $\hat{=}$ maximal number of variables on a path between the root and a leaf

RBs with a limited depth can contain an **unbounded** number of variables!

Given a RB of ϕ of depth k to a tractable class $\mathcal{C}$ we can decide satisfiability of ϕ in time:

$$2^k \cdot poly(|\phi|)$$

RB Detection

Again we need an fpt detection algorithm for RBs:

Input: (ϕ, k)

Output:

- There exists no RB of depth at most k for ϕ, or
- a RB of depth $g(k)$.

RB Detection

Again we need an fpt detection algorithm for RBs:

Input: (ϕ, k)

Output:

- There exists no RB of depth at most k for ϕ, or
- a RB of depth $g(k)$.

Base Class: $\mathcal{C}_0 \hat{=}$ the class of edgeless incidence graphs

RB Detection

Again we need an fpt detection algorithm for RBs:

Input: (ϕ, k)

Output:

- There exists no RB of depth at most k for ϕ, or
- a RB of depth $g(k)$.

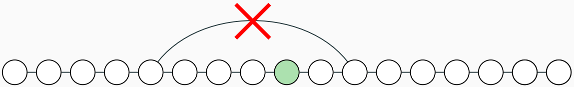
Base Class: $\mathcal{C}_0 \triangleq$ the class of edgeless incidence graphs

Theorem (Mählmann, Siebertz, Vigny)

RB detection to $\mathcal{C}_0$ is fixed parameter tractable.

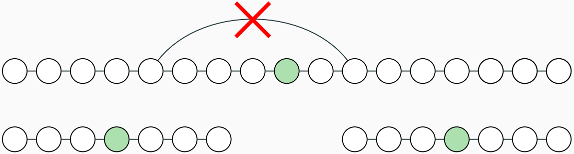
Bounded Diameter

RB to $\mathcal{C}_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$.



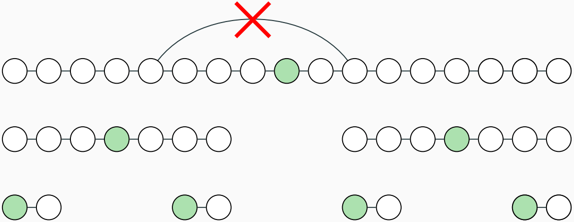
Bounded Diameter

RB to $\mathcal{C}_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$.



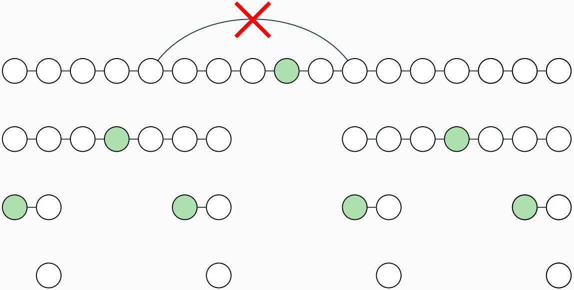
Bounded Diameter

RB to $\mathcal{C}_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$.



Bounded Diameter

RB to $\mathcal{C}_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$.



Bounded Clause Degree

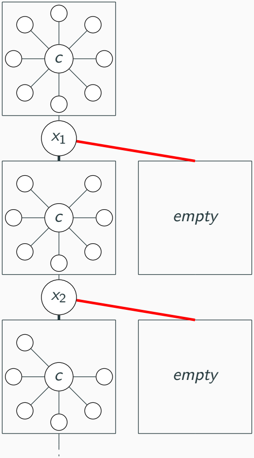
RB to $\mathcal{C}_0$ of depth $\leq k$ implies clause degree $\leq k$.

$(x_{1\_} \vee x_{2\_} \vee ... \vee x_{k\_})$

Bounded Clause Degree

RB to $\mathcal{C}_0$ of depth $\leq k$ implies clause degree $\leq k$.

$(x_{1\_} \vee x_{2\_} \vee \ldots \vee x_{k\_})$



Obstruction-Trees: $k = d$

Given: an incidence graph G with maximal clause degree $d \leq k$

Obstruction-Trees: $k = d$

Given: an incidence graph G with maximal clause degree $d \leq k$

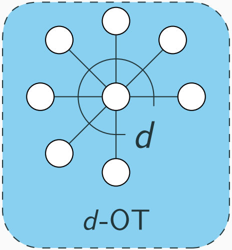
A k-obstruction-tree is a subgraph that guarantees G to have RB depth at least k.

Obstruction-Trees: $k = d$

Given: an incidence graph G with maximal clause degree $d \leq k$

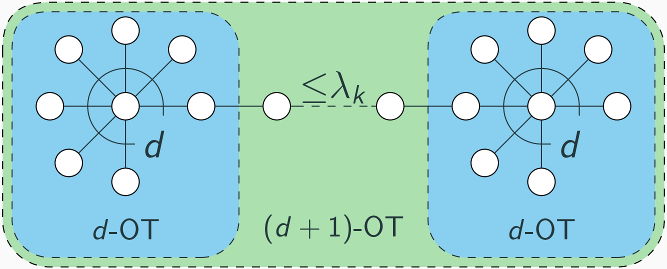
A k-obstruction-tree is a subgraph that guarantees G to have RB depth at least k.

For $k = d$:



$\rightarrow$ a d-clause in G is a d-obstruction-tree.

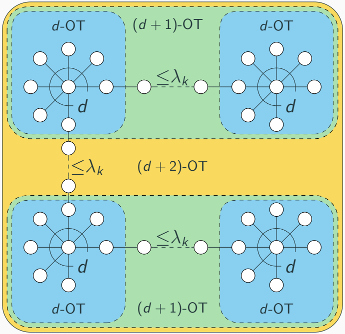
For $k = d + 1$:



$\rightarrow$ two conncected and **variable disjoint** d-clauses in G is form a $(d+1)$-obstruction-tree.

Obstruction-Trees: $k = i$

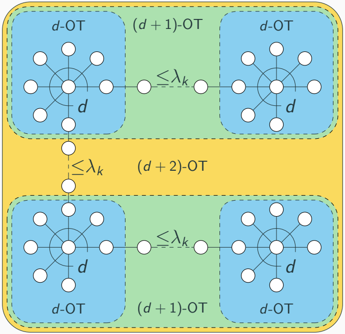
For $k = i + 1$:



$\rightarrow$ two conncected i-OTs with disjoint "neighborhoods" in G form an $(i+1)$-OT.

Obstruction-Trees: $k = i$

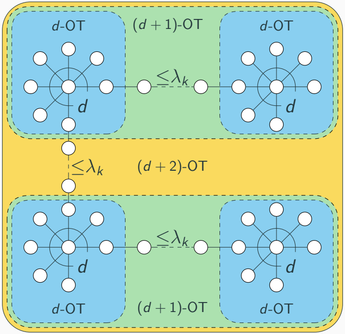
For $k = i + 1$:



$\rightarrow$ two conncected i-OTs with disjoint "neighborhoods" in G form an $(i+1)$-OT.

$\rightarrow$ the neighborhood of an obstruction-tree contains at most $f(k)$ variables

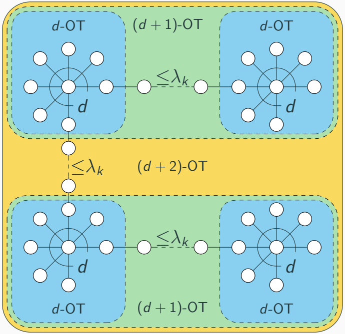
Searching for Obstruction-Trees



Given ϕ with maximal clause degree d, there exists an algorithm SEARCH_i that either:

- finds an i-obstruction-tree, or

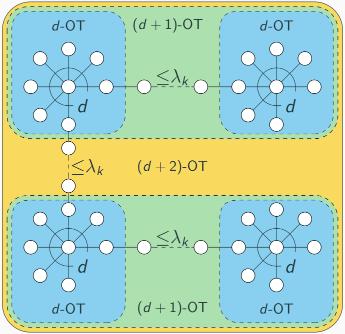
Searching for Obstruction-Trees



Given ϕ with maximal clause degree d, there exists an algorithm SEARCH$_i$ that either:

- finds an i-obstruction-tree, or
- finds an RB with bounded depth to $\mathcal{C}_{d-1}$, or

Searching for Obstruction-Trees



Given ϕ with maximal clause degree d, there exists an algorithm SEARCH$_i$ that either:

- finds an i-obstruction-tree, or
- finds an RB with bounded depth to $\mathcal{C}_{d-1}$, or
- concludes that no RB of depth $\leq k$ to $\mathcal{C}_0$ exists

Summary

What we have seen:

- Backdoors classify tractable SAT instances
- RBs generalize SAT backdoors and extend their power
- RB detection to $\mathcal{C}_0$ is fixed parameter tractable

Summary

What we have seen:

- Backdoors classify tractable SAT instances
- RBs generalize SAT backdoors and extend their power
- RB detection to $\mathcal{C}_0$ is fixed parameter tractable

What's next?

Theorem (Jan Dreier, Sebastian Ordyniak, Stefan Szeider)

RB detection to 2CNF is fixed parameter tractable.

- Further base classes are still open: Horn, Antihorn, Bounded Treewidth
- RBs to heterogenous base classes

Summary

What we have seen:

- Backdoors classify tractable SAT instances
- RBs generalize SAT backdoors and extend their power
- RB detection to $\mathcal{C}_0$ is fixed parameter tractable

What's next?

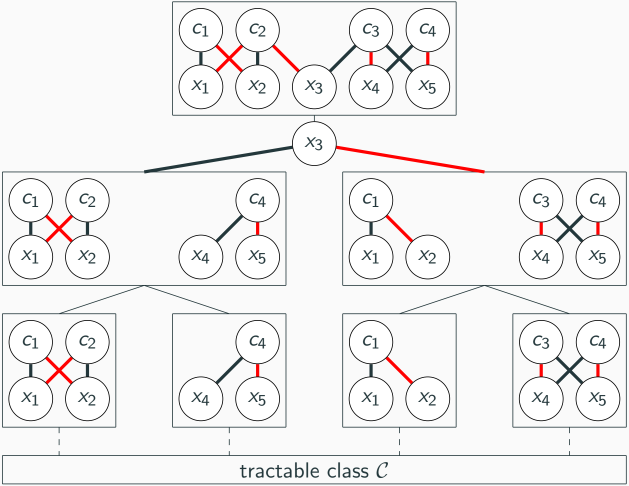
Theorem (Jan Dreier, Sebastian Ordyniak, Stefan Szeider)

RB detection to 2CNF is fixed parameter tractable.

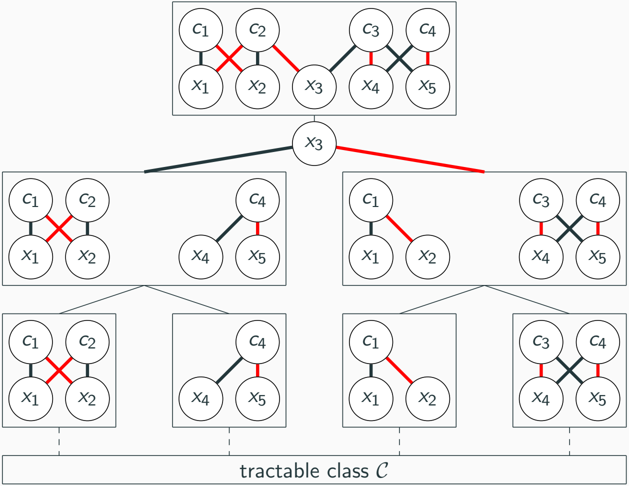
- Further base classes are still open: Horn, Antihorn, Bounded Treewidth
- RBs to heterogenous base classes

Thank you for listening!

Using RBs to Solve SAT

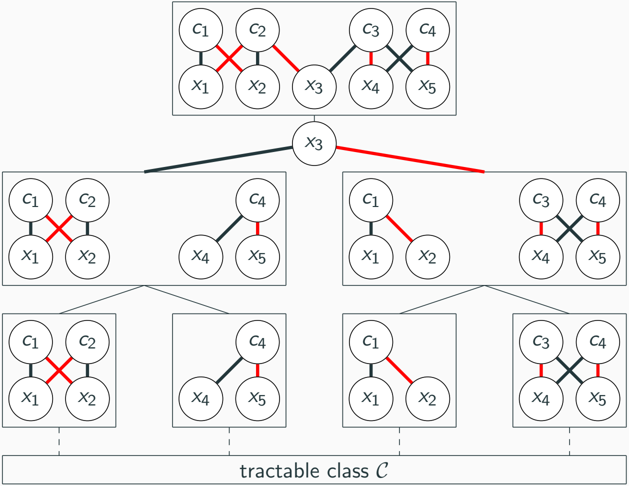


Using RBs to Solve SAT



solve leaves
in $poly(|\phi|)$

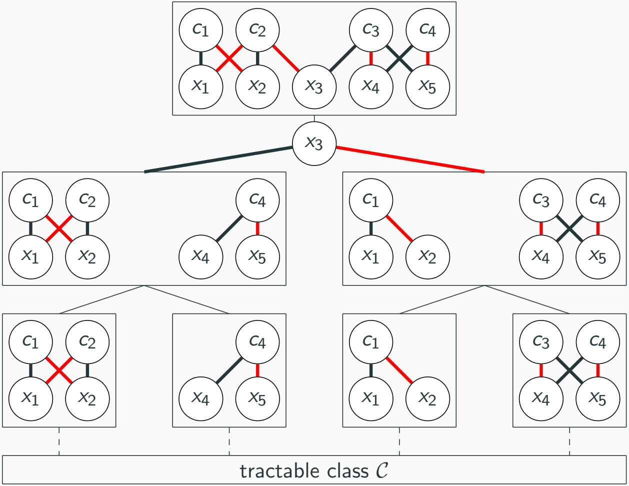
Using RBs to Solve SAT



solve both children in $2 \cdot 2^{k-1} \cdot poly(|\phi|)$

solve leaves in $poly(|\phi|)$

Using RBs to Solve SAT



solve both children in $2 \cdot 2^{k-1} \cdot poly(|\phi|)$

solve all children using superadditivity: $f(n_1 + n_2 + \ldots) \geq f(n_1) + f(n_2) + \ldots$

solve leaves in $poly(|\phi|)$