Recursive Backdoors for SAT

Nikolas Mählmann, Sebastian Siebertz, Alexandre Vigny
23.08.2021

University of Bremen
The SAT Problem

Input: a formula $\phi$ of propositional logic
Output: does there exists a satisfying assignment for $\phi$?
The SAT Problem

Input: a formula $\phi$ of propositional logic
Output: does there exists a satisfying assignment for $\phi$?

Examples:

$$(x_\neg \lor y_+) \land (x_+ \lor z_+)$$
The SAT Problem

Input: a formula $\phi$ of propositional logic
Output: does there exists a satisfying assignment for $\phi$?

Examples:

$(x_\neg \lor y_+) \land (x_+ \lor z_+)$ is SAT
The SAT Problem

Input: a formula $\phi$ of propositional logic
Output: does there exist a satisfying assignment for $\phi$?

Examples:

$$(x_{-} \lor y_{+}) \land (x_{+} \lor z_{+})$$ is SAT

$$(x_{+} \lor y_{+}) \land (x_{-}) \land (y_{-})$$
Input: a formula $\phi$ of propositional logic
Output: does there exists a satisfying assignment for $\phi$?

Examples:

$$(x_+ \lor y_+) \land (x_+ \lor z_+)$$ is SAT
$$(x_+ \lor y_+) \land (x_-) \land (y_-)$$ is UNSAT
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!
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!

\[ \phi = (x_{1-} \lor x_{2-} \lor x_{3+} \lor x_{4+}) \land (x_{4+} \lor x_{5+}) \land (x_{5+} \lor x_{6+}) \land ... \]
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!

\[ \phi = (x_1^- \lor x_2^- \lor x_3^+ \lor x_4^+) \land (x_4^+ \lor x_5^+) \land (x_5^+ \lor x_6^+) \land \ldots \]

\( \phi \) is not in 2CNF but very close to 2CNF.
A backdoor $B$ of $\phi$ to $C$ is a set of variables that reduces $\phi$ to a formula from $C$ no matter which assignment is chosen.
A backdoor $B$ of $\phi$ to $C$ is a set of variables that reduces $\phi$ to a formula from $C$ no matter which assignment is chosen.

Example:

$$\phi = (x_1 \lor x_2 \lor x_3 \lor x_4) \land (x_4 \lor x_5) \land (x_5 \lor x_6) \land \ldots$$

$\{x_1, x_2\}$ is a backdoor of $\phi$ to 2CNF.
A backdoor $B$ of $\phi$ to $C$ is a set of variables that reduces $\phi$ to a formula from $C$ no matter which assignment is chosen.

Example:

$$\phi = (x_1^- \lor x_2^- \lor x_3^+ \lor x_4^+) \land (x_4^+ \lor x_5^+) \land (x_5^+ \lor x_6^+) \land \ldots$$

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

$$\phi[x_1^+, x_2^+] =$$

$$\phi[x_1^-, x_2^+] =$$

$$\phi[x_1^+, x_2^-] =$$

$$\phi[x_1^-, x_2^-] =$$
A backdoor $B$ of $\phi$ to $C$ is a set of variables that reduces $\phi$ to a formula from $C$ no matter which assignment is chosen.

Example:

$$\phi = (x_1 \lor x_2 \lor x_3 \lor x_4) \land (x_4 \lor x_5) \land (x_5 \lor x_6) \land \ldots$$

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

$$\phi[x_1^+, x_2^+] = (x_3^+ \lor x_4^+) \land (x_4^+ \lor x_5^+) \land (x_5^+ \lor x_6^+) \land \ldots$$

$$\phi[x_1^-, x_2^+] =$$

$$\phi[x_1^+, x_2^-] =$$

$$\phi[x_1^-, x_2^-] =$$
Backdoors for SAT

A backdoor $B$ of $\phi$ to $C$ is a set of variables that reduces $\phi$ to a formula from $C$ no matter which assignment is chosen.

Example:

$$\phi = (x_{1-} \lor x_{2-} \lor x_{3+} \lor x_{4+}) \land (x_{4+} \lor x_{5+}) \land (x_{5+} \lor x_{6+}) \land \ldots$$

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

$$\phi[x_{1+}, x_{2+}] = (x_{3+} \lor x_{4+}) \land (x_{4+} \lor x_{5+}) \land (x_{5+} \lor x_{6+}) \land \ldots$$

$$\phi[x_{1-}, x_{2+}] = (x_{4+} \lor x_{5+}) \land (x_{5+} \lor x_{6+}) \land \ldots$$

$$\phi[x_{1+}, x_{2-}] = (x_{4+} \lor x_{5+}) \land (x_{5+} \lor x_{6+}) \land \ldots$$

$$\phi[x_{1-}, x_{2-}] = (x_{4+} \lor x_{5+}) \land (x_{5+} \lor x_{6+}) \land \ldots$$
Using Backdoors to Solve SAT

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

Runtime complexity:

$$2^k \cdot poly(|\phi|)$$
Using Backdoors to Solve SAT

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

Runtime complexity:

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

Fixed Parameter Tractability: Running times of the form:

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

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

Runtime complexity:

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

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 : \textcolor{red}{racing}
tire width : \{21\text{mm}, 23\text{mm}, 28\text{mm}\}
Recursive Backdoors

\((x_1^+ \lor x_2^-) \land (x_1^- \lor x_2^+ \lor x_3^-) \land (x_3^+ \lor x_4^- \lor x_5^+) \land (x_4^+ \lor x_5^-)\)
Recursive Backdoors

\[
(x_{1+} \lor x_{2-}) \land (x_{1-} \lor x_{2+} \lor x_{3-}) \land (x_{3+} \lor x_{4-} \lor x_{5+}) \land (x_{4+} \lor x_{5-})
\]
Recursive Backdoors

\[(x_{1+} \lor x_{2-}) \land (x_{1-} \lor x_{2+} \lor x_{3-}) \land (x_{3+} \lor x_{4-} \lor x_{5+}) \land (x_{4+} \lor x_{5-})\]
Recursive Backdoors

\[(x_{1+} \lor x_{2-}) \land (x_{1-} \lor x_{2+} \lor x_{3-}) \land (x_{3+} \lor x_{4-} \lor x_{5+}) \land (x_{4+} \lor x_{5-})\]
Recursive Backdoor Depth

Definition (Mählmann, Siebertz, Vigny)

\[
\text{rbd}_{\mathcal{C}}(G) = \begin{cases} 
\text{if } G \in \mathcal{C}: \\
0 \\
\text{otherwise:} \\
\max\left\{ \text{rbd}_{\mathcal{C}}(H) : H \text{ connected component of } G \right\}
\end{cases}
\]
Recursive Backdoor Depth

Definition (Mählmann, Siebertz, Vigny)

\[
rbd_C(G) = \begin{cases} 
  0 & \text{if } G \in C; \\
  \text{if } G \notin C \text{ and } G \text{ is connected:} \\
  1 + \min_{x \in \text{var}(G)} \max_{\star \in \{+,-\}} rbd_C(G[x_{\star}]) & \text{otherwise.}
\end{cases}
\]
Recursive Backdoor Depth

**Definition (Mählmann, Siebertz, Vigny)**

\[
\text{rbd}_C(G) = \begin{cases} 
\begin{align*}
& \text{if } G \in C: \\
& 0
\end{align*} \\
& \text{if } G \notin C \text{ and } G \text{ is connected:} \\
& 1 + \min_{x \in \text{var}(G)} \max_{\star \in \{+,-\}} \text{rbd}_C(G[x_{\star}])
\end{cases}
\]

otherwise:

\[
\max \{ \text{rbd}_C(H) : H \text{ connected component of } G \}
\]
Measuring RBs

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

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

RBs with a limited depth can contain an *unbounded* number of variables!
depth of a RB $\hat{=} \text{maximal number of variables on a path between the root and a leaf}$

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

Given a RB of $\phi$ of depth $k$ to a tractable class $C$ we can decide satisfiability of $\phi$ in time:

$$2^k \cdot poly(|\phi|)$$
Again we need an fpt detection algorithm for RBs:

Input: \((\phi, k)\)

Output:

- There exists no RB of depth at most \(k\) for \(\phi\), or
- a RB of depth \(g(k)\).
Again we need an fpt detection algorithm for RBs:

Input: $(\phi, k)$

Output:

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

Base Class: $\mathcal{C}_0 \triangleq$ the class of edgeless incidence graphs
Again we need an fpt detection algorithm for RBs:

Input: \((\phi, k)\)

Output:

- There exists no RB of depth at most \(k\) for \(\phi\), 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.*
RB to $C_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$. 
RB to $C_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$. 
RB to $C_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$. 
RB to $C_0$ of depth $\leq k$ implies diameter $\leq \lambda_k := 4 \cdot 2^k$. 
Bounded Clause Degree

RB to $C_0$ of depth $\leq k$ implies clause degree $\leq k$.

$$(x_1 \lor x_2 \lor \ldots \lor x_k)$$
RB to $C_0$ of depth $\leq k$ implies clause degree $\leq k$.

\[(x_1 \lor x_2 \lor ... \lor x_k)\]
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 $k = d + 1$:

→ two connected and variable disjoint $d$-clauses in $G$ form a $(d + 1)$-obstruction-tree.
Obstruction-Trees: \( k = i \)

For \( k = i + 1 \):

\[ d \leq \lambda_k \]

\[ (d + 1) \leq \lambda_k \]

\[ (d + 2) \leq \lambda_k \]

→ two connected \( i \)-OTs with disjoint “neighborhoods” in \( G \) form an \( (i + 1) \)-OT.
Obstruction-Trees: \( k = i \)

For \( k = i + 1 \): 

\[
\begin{align*}
&\rightarrow \text{two connected } i\text{-OTs with disjoint “neighborhoods” in } G \text{ form an } (i + 1)\text{-OT.} \\
&\rightarrow \text{the neighborhood of an obstruction-tree contains at most } f(k) \text{ variables}
\end{align*}
\]
Given $\phi$ with maximal clause degree $d$, there exists an algorithm $\text{SEARCH}_i$ that either:

- finds an $i$-obstruction-tree, or
Given $\phi$ with maximal clause degree $d$, there exists an algorithm $\text{SEARCH}_i$ that either:

- finds an $i$-obstruction-tree, or
- finds an RB with bounded depth to $C_{d-1}$, or
Given $\phi$ 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 $C_{d-1}$, or
- concludes that no RB of depth $\leq k$ to $C_0$ exists
Summary

What we have seen:

- Backdoors classify tractable SAT instances
- RBs generalize SAT backdoors and extend their power
- RB detection to $C_0$ 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!
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 $C_0$ 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

tractable class $C$
Using RBs to Solve SAT

tractable class $\mathcal{C}$

solve leaves in $\text{poly}(|\phi|)$
Using RBs to Solve SAT

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

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

tractable class $\mathcal{C}$
Using RBs to Solve SAT

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

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

solve leaves in $\text{poly}(|\phi|)$

tractable class $\mathcal{C}$