# Resolution Size By Graph Based Parameters

Ziyi Cai

BASICS Lab, Shanghai Jiao Tong University

March 10, 2023

# Content

# Tree Decompositions of Graphs

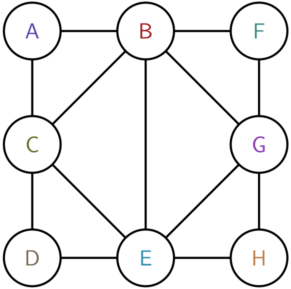
**Definition 1.1 (tree decomposition)**

Let $G$ be a graph. A *tree decomposition* of $G$ is a tuple $\left(T, (B_t)_{t \in V(T)}\right)$, where $T$ is a tree and $B_t$ the *bag* at $t$ such that the following conditions are satisfied:

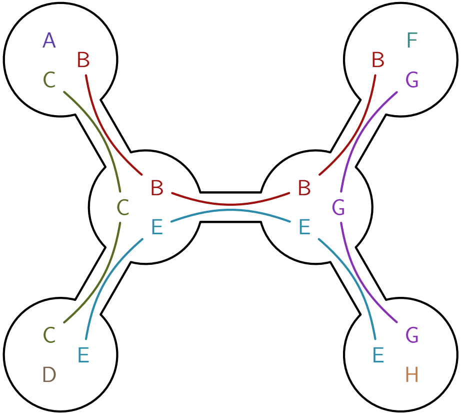
- For every $v \in V(G)$ the set
$$T_v := \{t \in V(T) \mid v \in B_t\}$$
is nonempty and connected in $T$, i.e, $T[T_v]$ is a subtree of $T$.
- For every $e \in E(G)$ there exists a $t \in V(T)$ such that $e \subseteq B_t$.

# Tree Decompositions of Graphs

Graph $G$




Tree decomposition $(T, X)$



# Treewidth

The *width* of a tree decomposition $\left(T, (B_t)_{t \in V(T)}\right)$ is

$$\operatorname{width}\left(T, (B_t)_{t \in V(T)}\right) := \max\{|B_t| - 1 \mid t \in V(T)\}.$$

The *treewidth* of $G$ is the minimum width.

# Pathwidth

## Definition 1.2 (path decomposition)

*Path decompositions* and *pathwidth* are defined similarly as tree decompositions and treewidth, except that in the definition of path decomposition, $T$ is restricted to a simple path.

## Fact 1.3

*For any graph $G$, $pw(G) \geq tw(G)$.*

## Theorem 1.4 ([KS93])

*For any graph $G$, $pw(G) = O(tw(G) \cdot \log |V(G)|)$.*

# Tree-depth

## Definition 1.5 (tree-depth)

The *tree-depth* $\mathrm{td}(G)$ of a graph $G$ is the minimum height of a rooted forest $F$ such that $G \subseteq \mathrm{clos}(F)$.

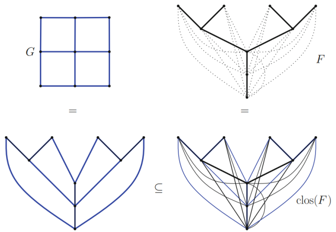
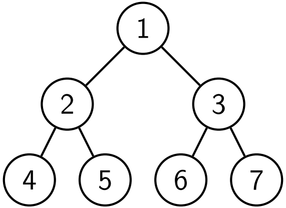



Figure: The tree-depth of the $3 \times 3$ grid is 4, adopted from [NdM12].

# Tree-depth

**Theorem 1.6**

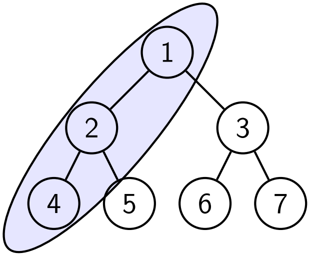
*For any graph $G$, $td(G) - 1 \geq pw(G)$.*



# Tree-depth

**Theorem 1.6**

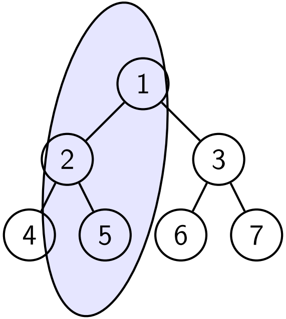
*For any graph $G$, $td(G) - 1 \geq pw(G)$.*



# Tree-depth

**Theorem 1.6**

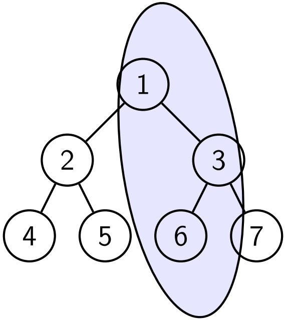
*For any graph $G$, $td(G) - 1 \geq pw(G)$.*



# Tree-depth

**Theorem 1.6**

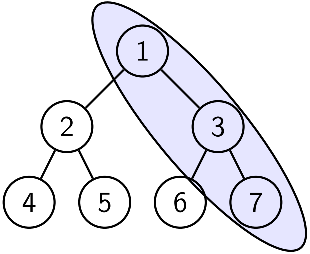
*For any graph $G$, $td(G) - 1 \geq pw(G)$.*



# Tree-depth

**Theorem 1.6**

*For any graph $G$, $td(G) - 1 \geq pw(G)$.*



# Tree-depth

**Theorem 1.6**

*For any graph $G$, $td(G) - 1 \geq pw(G)$.*

**Theorem 1.7**

*For any connected graph $G$, $td(G) = O\left(tw(G) \cdot \log |V(G)|\right)$.*

# SAT

- Variable $x$: takes Boolean value
- Literal $\ell$: variable $x$ or its negation $\overline{x}$
- Clause $C = \ell_1 \vee \cdots \vee \ell_k$: disjunction of literals
- Conjunctive normal form (CNF) formula $\varphi = C_1 \wedge \cdots \wedge C_m$: conjunction of clauses
  - $k$-CNF: conjunction of clauses with at most $k$ literals each

**Definition 1.8 (SAT)**

Given a CNF formula $\varphi$, is it satisfiable?

**Example 1.9**

$\varphi = (a \vee b \vee c) \wedge (a \vee \overline{c}) \wedge \overline{b} \wedge (\overline{a} \vee c) \wedge (b \vee \overline{c})$

# Underlying Graphs of SAT Formulas

Some methods to get a graph from a hypergraph are discussed in [HOSG07]. We use them to construct graphs from SAT formulas.

**Definition 1.10**

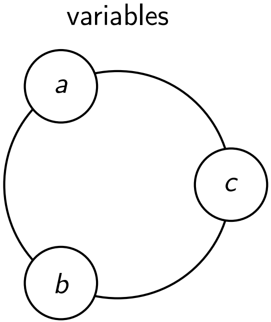
The *primal graph* $P_\varphi$ of a formula $\varphi$ is the graph whose vertices are the variables of $F$, where two vertices are connected by an edge iff the corresponding variables appear together (negated or unnegated) in some clauses.

**Definition 1.11**

The *incidence graph* $I_\varphi$ of a formula $\varphi$ is the bipartite graph between variables and clauses where two vertices are connected by an edge iff the corresponding variable appears (negated or unnegated) in the corresponding clause.

# Underlying Graphs of SAT Formulas

Primal Graph of $\varphi$




Incidence Graph of $\varphi$

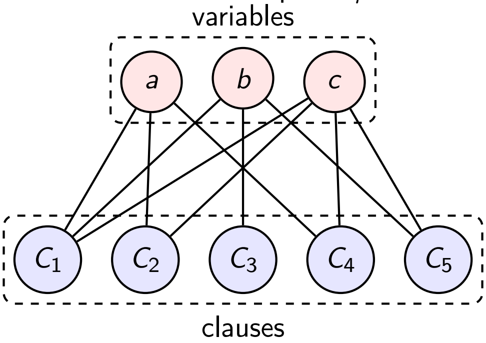



Figure: Underlying graphs of $\varphi = (a \vee b \vee c) \wedge (a \vee \overline{c}) \wedge \overline{b} \wedge (\overline{a} \vee c) \wedge (b \vee \overline{c})$.

# Primal Graph v.s. Incidence Graph

**Theorem 1.12**

*For any CNF formula $\varphi$, $tw(I_\varphi) \leq tw(P_\varphi) + 1$.*

The treewidth of an incidence graph can be significantly smaller than the one of a primal graph.

**Example 1.13**

Let $\varphi = x_1 \vee x_2 \vee \cdots \vee x_m$. Then $\text{tw}\left(P_\varphi\right) = m - 1$ while $\text{tw}\left(I_\varphi\right) = 1$.

# Resolution

**Definition 1.14 (resolution)**

*Resolution* is one of the propositional proof systems and has only one inference rule:

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$$

Resolution rules take two clauses and produce a new implied clause (*resolvent*). A *resolution refutation* of a formula $\varphi$ is a sequence of clauses $C_1, C_2, \ldots, C_k$ such that

- for each $i (1 \leq i \leq k)$, $C_i$ is a clause occurring in $\varphi$ or a resolvent of two previous clauses, and
- the last clause $C_k$ is an empty clause.

The *size* of the refutation is the number $k$ of clauses. The *width* of the refutation is the maximum number of literals in clauses.

# Resolution

**Theorem 1.15**

*Resolution is sound and complete.*

**Theorem 1.16**

*If for every unsatisfiable CNF $\varphi$ there exists a polynomially bounded resolution refutation proof, then* $\mathrm{NP} = \mathrm{coNP}$.

It is shown by [Hak85] that there are infinitely many CNF formulas $\varphi$ such that the resolution size of $\varphi$ cannot be bounded by a polynomial of the size of $\varphi$.

# Fixed Parameter Tractable

The complexity of SAT and #SAT parameterized by primal and incidence treewidth is FPT for all cases and of the form $2^{O(k)} \cdot |\varphi|^{O(1)}$, where $k$ is the (primal / incidence) treewidth.

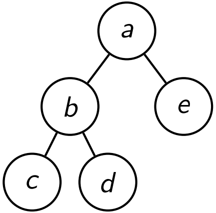
These results are obtained from numerous studies that advance in this direction, including [AR02] and [CMR01], demonstrating similar findings both explicitly and implicitly.

The topic of this talk is whether such an upper bound can be established for the resolution proof system.

# Bounded by td $(P_\varphi)$

**Theorem 2.1**

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*

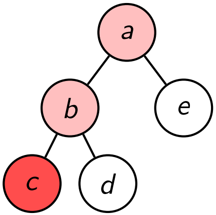



A resolution refutation for $\varphi = (a \vee c) \wedge (b \vee \overline{c}) \wedge (a \vee \overline{b} \vee d) \wedge (a \vee \overline{d}) \vee \overline{a} \vee e$.

# Bounded by td $(P_\varphi)$

### Theorem 2.1

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*



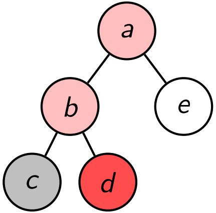

$$\frac{a \vee c \qquad b \vee \overline{c}}{a \vee b}$$

A resolution refutation for $\varphi = (a \vee c) \wedge (b \vee \overline{c}) \wedge (a \vee \overline{b} \vee d) \wedge (a \vee \overline{d}) \vee \overline{a} \vee e$.

# Bounded by td $(P_\varphi)$

**Theorem 2.1**

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*



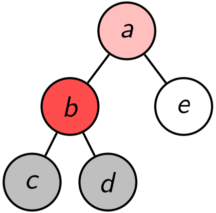

$$\frac{a \lor c \qquad b \lor \overline{c}}{a \lor b} \qquad \frac{a \lor \overline{b} \lor d \qquad a \lor \overline{d}}{a \lor \overline{b}}$$

A resolution refutation for $\varphi = (a \lor c) \land (b \lor \overline{c}) \land (a \lor \overline{b} \lor d) \land (a \lor \overline{d}) \lor \overline{a} \lor e$.

# Bounded by td $(P_\varphi)$

**Theorem 2.1**

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*



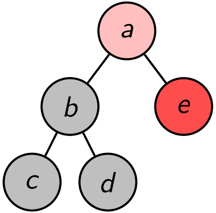

$$\dfrac{\dfrac{a \vee c \qquad b \vee \overline{c}}{a \vee b} \qquad \dfrac{a \vee \overline{b} \vee d \qquad a \vee \overline{d}}{a \vee \overline{b}}}{a}$$

A resolution refutation for $\varphi = (a \vee c) \wedge (b \vee \overline{c}) \wedge (a \vee \overline{b} \vee d) \wedge (a \vee \overline{d}) \vee \overline{a} \vee e$.

# Bounded by $\mathrm{td}(P_\varphi)$

**Theorem 2.1**

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*



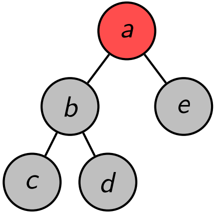

$$\dfrac{\dfrac{a \vee c \qquad b \vee \overline{c}}{a \vee b} \qquad \dfrac{a \vee \overline{b} \vee d \qquad a \vee \overline{d}}{a \vee \overline{b}}}{a}$$

A resolution refutation for $\varphi = (a \vee c) \wedge (b \vee \overline{c}) \wedge (a \vee \overline{b} \vee d) \wedge (a \vee \overline{d}) \vee \overline{a} \vee e$.

# Bounded by $\mathrm{td}(P_\varphi)$

**Theorem 2.1**

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*



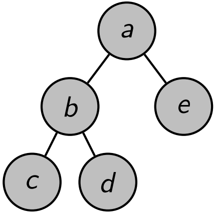

$$\dfrac{\dfrac{\dfrac{a \vee c \qquad b \vee \overline{c}}{a \vee b} \qquad \dfrac{a \vee \overline{b} \vee d \qquad a \vee \overline{d}}{a \vee \overline{b}}}{a} \qquad \overline{a}}{\perp}$$

A resolution refutation for $\varphi = (a \vee c) \wedge (b \vee \overline{c}) \wedge (a \vee \overline{b} \vee d) \wedge (a \vee \overline{d}) \vee \overline{a} \vee e$.

# Bounded by td $(P_\varphi)$

**Theorem 2.1**

*For an unsatisfiable formula $\varphi$ with primal tree-depth $td(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(td(P_\varphi))} \cdot |\varphi|$.*




$$\dfrac{\dfrac{\dfrac{a \vee c \qquad b \vee \overline{c}}{a \vee b} \qquad \dfrac{a \vee \overline{b} \vee d \qquad a \vee \overline{d}}{a \vee \overline{b}}}{a} \qquad \overline{a}}{\bot}$$

A resolution refutation for $\varphi = (a \vee c) \wedge (b \vee \overline{c}) \wedge (a \vee \overline{b} \vee d) \wedge (a \vee \overline{d}) \vee \overline{a} \vee e$.

# Bounded by $\mathrm{tw}(P_\varphi)$

Recall that $\mathrm{tw}(P_\varphi) < \mathrm{td}(P_\varphi)$.

**Theorem 2.2**

*For an unsatisfiable formula $\varphi$ with primal treewidth $tw(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(tw(P_\varphi))} \cdot |\varphi|$.*

# Bounded by $\mathrm{tw}(P_\varphi)$

## Theorem 2.2

*For an unsatisfiable formula $\varphi$ with primal treewidth $tw(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(tw(P_\varphi))} \cdot |\varphi|$.*

## Proof Sketch

A *smooth* tree decomposition is a tree decomposition satisfying that $\left|B_t \backslash B_{fa(t)}\right| = 1$ for all $t \in V(T)$ but root. For any graph $G$ there exists a smooth tree decomposition of width $\mathrm{tw}(G)$. We call $B_t \backslash B_{fa(t)}$ the forgotten vertex w.r.t. $t$.
The desired resolution refutation is obtained by resolving over forgotten vertices in a depth-first order.

# Bounded by $\mathrm{tw}(P_\varphi)$

**Theorem 2.2**

*For an unsatisfiable formula $\varphi$ with primal treewidth $tw(P_\varphi)$, there exists a resolution refutation bounded by $2^{O(tw(P_\varphi))} \cdot |\varphi|$.*

$$\{b, c, d, \dots\}$$

$$\boxed{\{a, b, c, d, \dots\}}$$

# Bounded by $\mathrm{tw}(I_\varphi)$

Recall that $\mathrm{tw}(I_\varphi) \leq \mathrm{tw}(P_\varphi) + 1$.

**Conjecture 2.2.1**

*For an unsatisfiable formula $\varphi$ with incidence treewidth $tw(I_\varphi)$, there exists a resolution refutation bounded by $2^{O(tw(I_\varphi))} \cdot |\varphi|$.*

# Bounded by $\mathrm{tw}(I_\varphi)$, Under Fixed $k$

**Theorem 2.3**

*For any $k$-CNF formula $\varphi$, $tw(P_\varphi) = O(tw(I_\varphi))$.*

**Proof Sketch**

Replace all clauses in bags with their variables.

**Corollary 2.4**

*For an unsatisfiable $k$-CNF formula $\varphi$ with incidence treewidth $tw(I_\varphi)$, there exists a resolution refutation bounded by $2^{O(tw(I_\varphi))} \cdot |\varphi|$.*

# Bounded by $\mathrm{tw}(I_\varphi)$

It's well-known that any CNF formula can be converted into an equivalent 3-CNF formula. This conversion can be done without significantly increasing the incidence treewidth.

**Theorem 2.5**

*For any CNF formula $\varphi$, there exists an equivalent 3-CNF formula $\psi$ such that $tw(I_\psi) = O(tw(I_\varphi))$.*

# Bounded by $\mathrm{tw}(I_\varphi)$

For each clause $C$, relabel its variables according to the inorder traversal.
We show a simple example which converts $C = (\cdots \vee x_1 \vee x_2 \vee x_3 \vee x_4 \vee \cdots)$ to $C_1 = (\overline{y_0} \vee x_1 \vee y_1)$, $C_2 = (\overline{y_1} \vee x_2 \vee y_2)$, $C_3 = (\overline{y_2} \vee x_3 \vee y_3)$, $C_4 = (\overline{y_3} \vee x_4 \vee y_4)$.

$\{C, \ldots\}$

$\{C, \ldots\}$ $\{C, \ldots\}$

$\{C, x_1, \ldots\}$ $\{C, x_2, \ldots\}$ $\{C, x_3, \ldots\}$ $\{C, x_4, \ldots\}$

# Bounded by $\mathrm{tw}(I_\varphi)$

For each clause $C$, relabel its variables according to the inorder traversal.
We show a simple example which converts $C = (\cdots \vee x_1 \vee x_2 \vee x_3 \vee x_4 \vee \cdots)$ to
$C_1 = (\overline{y_0} \vee x_1 \vee y_1)$, $C_2 = (\overline{y_1} \vee x_2 \vee y_2)$, $C_3 = (\overline{y_2} \vee x_3 \vee y_3)$, $C_4 = (\overline{y_3} \vee x_4 \vee y_4)$.

$$\{C, \ldots\}$$

$$\{C, \ldots\} \qquad \{C, \ldots\}$$

$$\{C_1, x_1, y_1, \ldots\}\{C_2, x_2, y_2, \ldots\}\{C_3, x_3, y_3, \ldots\}\{C_4, x_4, y_4, \ldots\}$$

# Bounded by $\operatorname{tw}(I_\varphi)$

For each clause $C$, relabel its variables according to the inorder traversal.
We show a simple example which converts $C = (\cdots \vee x_1 \vee x_2 \vee x_3 \vee x_4 \vee \cdots)$ to
$C_1 = (\overline{y_0} \vee x_1 \vee y_1)$, $C_2 = (\overline{y_1} \vee x_2 \vee y_2)$, $C_3 = (\overline{y_2} \vee x_3 \vee y_3)$, $C_4 = (\overline{y_3} \vee x_4 \vee y_4)$.

$$\{C, \dots\}$$

$$\{{\color{red}y_1}, {\color{red}C_2}, \dots\} \qquad \{{\color{red}y_3}, {\color{red}C_4}, \dots\}$$

$$\{C_1, x_1, y_1, \dots\}\{C_2, x_2, y_2, \dots\}\{C_3, x_3, y_3, \dots\}\{C_4, x_4, y_4, \dots\}$$

# Bounded by $\mathrm{tw}(I_\varphi)$

For each clause $C$, relabel its variables according to the inorder traversal.

We show a simple example which converts $C = (\cdots \vee x_1 \vee x_2 \vee x_3 \vee x_4 \vee \cdots)$ to $C_1 = (\overline{y_0} \vee x_1 \vee y_1)$, $C_2 = (\overline{y_1} \vee x_2 \vee y_2)$, $C_3 = (\overline{y_2} \vee x_3 \vee y_3)$, $C_4 = (\overline{y_3} \vee x_4 \vee y_4)$.

$\{{\color{red}y_2}, {\color{red}C_3}, \ldots\}$

$\{y_1, C_2, \ldots\}$ $\{y_3, C_4, \ldots\}$

$\{C_1, x_1, y_1, \ldots\}\{C_2, x_2, y_2, \ldots\}\{C_3, x_3, y_3, \ldots\}\{C_4, x_4, y_4, \ldots\}$

# Bounded by $\mathrm{tw}(I_\varphi)$

For each clause $C$, relabel its variables according to the inorder traversal.
We show a simple example which converts $C = (\cdots \vee x_1 \vee x_2 \vee x_3 \vee x_4 \vee \cdots)$ to $C_1 = (\overline{y_0} \vee x_1 \vee y_1)$, $C_2 = (\overline{y_1} \vee x_2 \vee y_2)$, $C_3 = (\overline{y_2} \vee x_3 \vee y_3)$, $C_4 = (\overline{y_3} \vee x_4 \vee y_4)$.

$$\{y_2, C_3, \textcolor{red}{C_1}, \textcolor{red}{y_4}, \dots\}$$

$$\{y_1, C_2, \textcolor{red}{C_1}, \textcolor{red}{y_2}, \dots\} \qquad \{y_3, C_4, \textcolor{red}{C_3}, \textcolor{red}{y_4}, \dots\}$$

$$\{C_1, x_1, y_1, \dots\}\{C_2, x_2, y_2, \dots\}\{C_3, x_3, y_3, \dots\}\{C_4, x_4, y_4, \dots\}$$

After removing an original clause $C$, sizes of bags which contained $C$ will at most increase by 4. Hence we have $\mathrm{tw}(I_\psi) \leq 4 \cdot \mathrm{tw}(I_\varphi)$.

# Bounded by $\mathrm{tw}(I_\varphi)$

**Theorem 2.5**

*For any CNF formula $\varphi$, there exists an equivalent 3-CNF formula $\psi$ such that $tw(I_\psi) = O(tw(I_\varphi))$.*

**Corollary 2.6**

*For an unsatisfiable CNF formula $\varphi$ with incidence treewidth $tw(I_\varphi)$, there exists an equivalent 3-CNF formula $\psi$ with resolution refutation bounded by $2^{O(tw(I_\varphi))} \cdot |\varphi|$.*

**Remark 2.6.1**

*Additional variables are introduced while obtaining the equivalent formula. Hence this result does not necessarily imply we have a resolution refutation with size $2^{O(tw(I_\varphi))} \cdot |\varphi|$.*

# Bounded by $\mathrm{pw}(I_\varphi)$

**Theorem 2.7 ([Ima17])**

*For an unsatisfiable CNF formula $\varphi$ with incidence pathwidth $pw(I_\varphi)$, there exists a resolution refutation bounded by $2^{O(pw(I_\varphi))} \cdot |\varphi|$.*

**Corollary 2.8**

*For an unsatisfiable CNF formula $\varphi$ with incidence treewidth $tw(I_\varphi)$, there exists a resolution refutation bounded by $|\varphi|^{O(tw(I_\varphi))}$.*

# References I


- M. Alekhnovich and A.A. Razborov, *Satisfiability, branch-width and tseitin tautologies*, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings., 2002, pp. 593–603.
- B. Courcelle, J. A. Makowsky, and U. Rotics, *On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic*, Discrete Appl. Math. **108** (2001), no. 1, 23–52.
- Armin Haken, *The intractability of resolution*, Theoretical Computer Science **39** (1985), 297–308, Third Conference on Foundations of Software Technology and Theoretical Computer Science.
- Petr Hliněný, Sang-il Oum, Detlef Seese, and Georg Gottlob, *Width Parameters Beyond Tree-width and their Applications*, The Computer Journal **51** (2007), no. 3, 326–362.
- Kensuke Imanishi, *An upper bound for resolution size: Characterization of tractable sat instances*, WALCOM: Algorithms and Computation (Cham) (Sheung-Hung Poon, Md. Saidur Rahman, and Hsu-Chun Yen, eds.), Springer International Publishing, 2017, pp. 359–369.
- Ephraim Korach and Nir Solel, *Tree-width, path-widt, and cutwidth*, Discret. Appl. Math. **43** (1993), 97–101.

# References II

Jaroslav Neetil and Patrice Ossona de Mendez, *Sparsity: Graphs, structures, and algorithms*, Springer Publishing Company, Incorporated, 2012.