Proof Complexity of Propositional Model Counting

Olaf Beyersdorff, **Tim Hoffmann**, Luc Spachmann

Friedrich Schiller University Jena

Problem setting

Let a propositional formula φ be given.

SAT problem

- Is there an assignment $\alpha : \text{vars}(\varphi) \to \{0, 1\}$ such that α satisfies φ?
- SAT is NP complete. [Cook 1971, Levin 1973]

Model counting problem (#SAT)

- How many assignments $\alpha : \text{vars}(\varphi) \to \{0, 1\}$ satisfying φ are there?
- #SAT is #P complete. [Valiant 1979]

NP vs. #P

- Model counting is at least as hard as SAT
- [Toda 1991]: PH can be solved with #P oracle in polynomial time

$\rightarrow$ under common complexity theoretic assumptions, #SAT is much harder than SAT

Applications

Plenty of real-world applications:

- probabilistic reasoning
- risk analysis
- explainable artificial intelligence

Model counting solving

Model Counting Competition at SAT since 2020
[Fichte, Hecher, Hamiti 2020]

CDCL based

- adapt common techniques from SAT solving for model counting
- solver SharpSAT [Thurley 2006]

Knowledge compilation

- represent formula using an other structure on which counting is easy, e.g. restricted circuits (usually Decision DNNFs)
- solver D4 [Lagniez, Marquis 2017]

FPT algorithms

- e.g. tree decomposition based solver DPDB [Fichte, Hecher, Thier, Woltran 2020]

Proof systems

Proof systems [Cook, Reckhow 1979]

A proof system P for model counting has the following properties:

- *Soundness*: if there is a P-proof that φ has c models, then this is correct
- *Completeness*: if φ has c models, then there is a P-proof for that
- P-proofs are *efficiently verifiable*

Why study proof systems?

- *Aim*: efficient extraction of proofs from a solver

$\rightarrow$ requires strong enough proof system that can express all steps of a solver

Proof logging

verification of proofs $\rightarrow$ verification of solver outputs

Understanding of solver performance

φ has no short P-proof $\rightarrow$ solver requires long running time to solve φ

Proof systems for #SAT

There exist two formal proof systems for model counting so far.

kcps(#SAT) [Capelli SAT'19]

- "Knowledge Compilation based Proof System"
- static proof system based on Decision DNNFs
- can be used to verify the trace of a modified D4
- some lower bounds follow from Decision DNNFs

MICE [Fichte, Hecher, Roland SAT'22]

- "Model counting Induction by Claim Extension"
- line-based proof system
- can be used to verify traces of SharpSAT, D4 and DPDB
- no proof complexity results are known

$\rightarrow$ we do proof complexity analysis for this proof system

Our contributions

Simplified proof system MICE′

- more intuitive than MICE and uses simpler rules with less conditions
- polynomially equivalent to MICE

Proof complexity of MICE′

- exponential lower bound for MICE′

$\rightarrow$ formulas are difficult for many modern solvers (without preprocessing)

The MICE′ proof system

MICE′ is a line-based proof system.

Proof lines

Every line is a claim which is a 3-tuple (F, A, c) with:

- F is a CNF,
- A is a partial assignment of variables of F called assumption,
- c is a count.

The claim states that F under assumption A has exactly c models.

Example

Claim $(\{a \vee b, \overline{b} \vee c\}, \{a = 1\}, 3)$ represents models

- $\{a = 1, b = 0, c = 0\}$,
- $\{a = 1, b = 0, c = 1\}$,
- $\{a = 1, b = 1, c = 1\}$.

MICE′ proofs

Let formula φ be given.

MICE′ proof

- A MICE′ proof π of φ is a sequence of claims

$$I_1, \ldots, I_k$$

derived only with the MICE′ rules such that

$$I_k = (\varphi, \emptyset, c)$$

for some $c \in \mathbb{N}$.

- π proves that φ has exactly c models.

The MICE′ derivation rules - overview

Axiom

$$\frac{}{(\emptyset, \emptyset, 1)}$$

Extension

$$\frac{(F_1, A_1, c_1)}{(F, A, c_1 \cdot 2^{|vars(F) \setminus (vars(F_1) \cup vars(A))|})}$$

- (E-1) $F_1 \subseteq F$
- (E-2) $A|_{vars(F_1)} = A_1$
- (E-3) A satisfies $F \setminus F_1$

Composition

$$\frac{(F, A_1, c_1) \quad \cdots \quad (F, A_n, c_n)}{(F, A, \sum_{i \in [n]} c_i)}$$

- (C-1) $vars(A_1) = vars(A_2) = \ldots = vars(A_n)$ and $A_i \neq A_j$ for $i \neq j$
- (C-2) $A \subseteq A_i$ for all $i \in [n]$
- (C-3) there exists a resolution refutation of $A \cup F \cup \{\overline{A}_i \mid i \in [n]\}$

Join

$$\frac{(F_1, A_1, c_1) \qquad (F_2, A_2, c_2)}{(F_1 \cup F_2, A_1 \cup A_2, c_1 \cdot c_2)}$$

- (J-1) A_1 and A_2 are consistent
- (J-2) $vars(F_1) \cap vars(F_2) \subseteq vars(A_i)$ for $i \in \{1, 2\}$

The Axiom rule

Axiom

$$\frac{}{(\emptyset, \emptyset, 1)}$$

$(\emptyset, \emptyset, 1)$ is usually the first claim to start a proof with

The Extension rule

Extension

$$\frac{(F_1, A_1, c_1)}{(F, A, c_1 \cdot 2^k)}$$

with $k = |\mathsf{vars}(F) \setminus (\mathsf{vars}(F_1) \cup \mathsf{vars}(A))|$

- (E-1) $F_1 \subseteq F$
- (E-2) $A|_{\mathsf{vars}(F_1)} = A_1$
- (E-3) A satisfies $F \setminus F_1$

Examples

- $$\frac{(\{a \vee b\}, \emptyset, 3)}{(\{a \vee b, a \vee c \vee d\}, \{d = 1\}, 6)}$$
- if α is a model of φ: $\dfrac{(\emptyset, \emptyset, 1)}{(\varphi, \alpha, 1)}$

The Composition rule

Composition

$$\frac{(F, A_1, c_1) \quad \cdots \quad (F, A_n, c_n)}{(F, A, \sum_{i \in [n]} c_i)}$$

- (C-1) $\mathrm{vars}(A_1) = \mathrm{vars}(A_2) = \ldots = \mathrm{vars}(A_n)$ and $A_i \neq A_j$ for $i \neq j$
- (C-2) $A \subseteq A_i$ for all $i \in [n]$
- (C-3) there exists a resolution refutation of $A \cup F \cup \{\overline{A}_i \mid i \in [n]\}$

Intuition

We combine sets of models.

- (C-1) ensures that we do not count models twice
- (C-3) ensures that there are no additional models

The Composition rule

Examples

- $$\frac{(\varphi, \{a = 0\}, c_0) \qquad (\varphi, \{a = 1\}, c_1)}{(\varphi, \emptyset, c_0 + c_1)}$$

  with resolution refutation for (C-3)

  $$\frac{a \qquad \overline{a}}{\emptyset}$$

- for unsatisfiable φ: $\dfrac{}{(\varphi, \emptyset, 0)}$

  with some resolution refutation of φ for (C-3)

  $$\frac{\begin{array}{cccc} C_1 & C_2 & \dots & C_n \end{array}}{\begin{array}{c} \vdots \\ \emptyset \end{array}}$$

The Join rule

Join

$$\frac{(F_1, A_1, c_1) \qquad (F_2, A_2, c_2)}{(F_1 \cup F_2, A_1 \cup A_2, c_1 \cdot c_2)}$$

- (J-1) A_1 and A_2 are consistent
- (J-2) $vars(F_1) \cap vars(F_2) \subseteq vars(A_i)$ for $i \in \{1, 2\}$

Intuition

- F_1 and F_2 are independent under assignments A_1, A_2
- we can pair every model of F_1 with every model of F_2

Examples

- $$\frac{(\{a \vee b \vee \overline{e}\}, \{e = 1\}, 3) \qquad (\{c \vee d \vee \overline{e}\}, \{e = 1\}, 3)}{(\{a \vee b \vee \overline{e}, c \vee d \vee \overline{e}\}, \{e = 1\}, 9)}$$

MICE
[Fichte, Hecher, Roland 2022]

Simplified MICE′
[This paper]

Exactly One Model, Axiom

$$\frac{}{(F, V, A, 1)}$$

- (O-1) $\mathrm{vars}(A) = V$
- (O-2) A satisfies F

$$\frac{}{(\emptyset, \emptyset, 1)}$$

Extension

$$\frac{(F_1, V_1, A_1, c)}{(F, V, A, c)}$$

- (E-1) $F_1 \subseteq F$, $V_1 \subseteq V$
- (E-2) $A|_{V_1} = A_1$
- (E-3) A satisfies $F \setminus F_1$
- (E-4) $V \setminus V_1 \subseteq \mathrm{vars}(A)$
- (E-5) for every $C \in F_1$: $A|_{V \setminus V_1}$ does not satisfy C

$$\frac{(F_1, A_1, c_1)}{(F, A, c_1 \cdot 2^{|\mathrm{vars}(F) \setminus (\mathrm{vars}(F_1) \cup \mathrm{vars}(A))|})}$$

- (E-1) $F_1 \subseteq F$
- (E-2) $A|_{\mathrm{vars}(F_1)} = A_1$
- (E-3) A satisfies $F \setminus F_1$

Composition

$$\frac{(F, V, A_1, c_1) \quad \cdots \quad (F, V, A_n, c_n)}{(F, V, A, \sum_{i \in [n]} c_i)}$$

- (C-1) $\mathrm{vars}(A_1) = \mathrm{vars}(A_2) = \ldots = \mathrm{vars}(A_n)$ and $A_i \not= A_j$ for $i \not= j$
- (C-2) $A \subseteq A_i$ for all $i \in [n]$
- (C-3) there exists a resolution refutation of $A \cup \{C|_V \mid C \in F\} \cup \{\overline{A}_i \mid i \in [n]\}$

$$\frac{(F, A_1, c_1) \quad \cdots \quad (F, A_n, c_n)}{(F, A, \sum_{i \in [n]} c_i)}$$

- (C-1) $\mathrm{vars}(A_1) = \mathrm{vars}(A_2) = \ldots = \mathrm{vars}(A_n)$ and $A_i \not= A_j$ for $i \not= j$
- (C-2) $A \subseteq A_i$ for all $i \in [n]$
- (C-3) there exists a resolution refutation of $A \cup F \cup \{\overline{A}_i \mid i \in [n]\}$

Join

$$\frac{(F_1, V_1, A_1, c_1) \qquad (F_2, V_2, A_2, c_2)}{(F_1 \cup F_2, V_1 \cup V_2, A_1 \cup A_2, c_1 \cdot c_2)}$$

- (J-1) A_1 and A_2 are consistent
- (J-2) $V_1 \cap V_2 \subseteq \mathrm{vars}(A_i)$ for $i \in \{1, 2\}$
- (J-3) $\mathrm{vars}(F_i) \cap ((V_1 \cup V_2) \setminus V_i) = \emptyset$ for $i \in \{1, 2\}$

$$\frac{(F_1, A_1, c_1) \qquad (F_2, A_2, c_2)}{(F_1 \cup F_2, A_1 \cup A_2, c_1 \cdot c_2)}$$

- (J-1) A_1 and A_2 are consistent
- (J-2) $\mathrm{vars}(F_1) \cap \mathrm{vars}(F_2) \subseteq \mathrm{vars}(A_i)$ for $i \in \{1, 2\}$

MICE′ is a proof system

Theorem [basically Fichte, Hecher, Roland 2022]

MICE′ is sound and complete.

Completeness

Let φ be an arbitrary formula.

- start with Axiom claim $I_A = (\emptyset, \emptyset, 1)$
- for every model α of φ, derive $I_\alpha = (\varphi, \alpha, 1)$ with Extension from I_A
- apply Composition to all claims I_α to derive $I = (\varphi, \emptyset, c)$

Theorem

MICE′ is equivalent to MICE.

MICE′ complexity measures

We use two different measures.

Proof size

- describes the number of MICE′ steps plus the number of clauses in all resolution refutations
- corresponds to the size of a complete encoding of π

Number of MICE′ steps

- ignores the sizes of the resolution refutations
- corresponds to the usage of a SAT oracle
- lower bound for MICE′ steps $\rightarrow$ lower bound for proof size

Lower bounds for the proof size

We can use MICE′ as proof system for UNSAT:

- φ is UNSAT exactly if there is a MICE′ derivation of claim $(\varphi, \emptyset, 0)$.

Proposition

MICE′ is equivalent to resolution for unsatisfiable formulas.

- Proof Idea: combine all resolution refutations of a MICE' proof

Lower bounds

- lower bounds from resolution apply also for MICE′
- e.g. Pigeonhole formulas (PHP) require MICE′ proofs of exponential size
- However, we can prove PHP with a single MICE′ step (one Composition).
- Next: Strengthen this result to #steps

Main result - lower bounds for number of MICE′ steps

XOR-PAIRS formulas

For $i, j \in [n]$ the formula XOR-PAIRS_n has variables x_i, z_{ij} and clauses encoding

$$z_{ij} = x_i \oplus x_j.$$

Theorem

Any MICE′ proof π of XOR-PAIRS_n requires $2^{\Omega(n)}$ MICE′ steps.

Proof of the lower bound

Consider an arbitrary MICE′ proof π of $\textsc{Xor-Pairs}_n$.

Join does not increase the count

If two claims (F_1, A_1, c_1) and (F_2, A_2, c_2) are joined, then $\min(c_1, c_2) = 1$.

Extension does not increase the count

If claim (F, A, c) is derived with Extension from (F_1, A_1, c_1), then $c = c_1$.

There are many models

$\textsc{Xor-Pairs}_n$ has 2^n models

Putting it together

- Axiom derives claims with count 1
- Extension does not change the count
- Join does not change the count
- Composition adds counts

Lower bound for tree-like MICE′

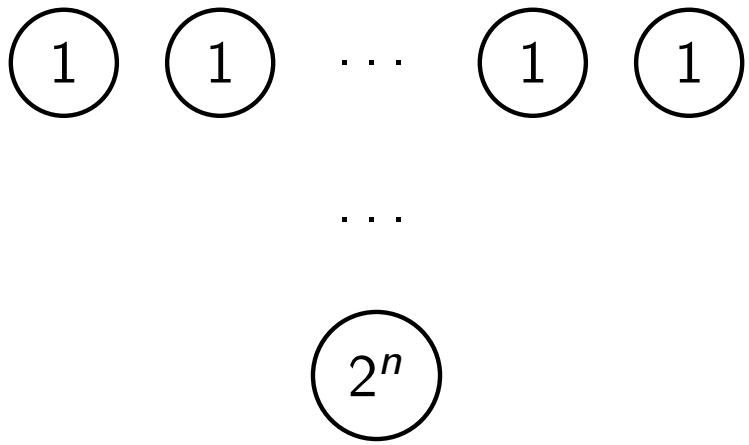

Figure: we can use every Axiom claim only once $\rightarrow$ we need 2^n Axiom claims

Lower bound for general MICE′ is more technical

Summary

Simplified proof system MICE′

- more intuitive than MICE and uses simpler rules with less conditions
- equivalent to MICE

Lower bounds

- $\textsc{Xor-Pairs}_n$ formulas need $2^{\Omega(n)}$ MICE' steps

$\rightarrow$ $\textsc{Xor-Pairs}_n$ are difficult for many modern solvers (without preprocessing), even with access to arbitrarily good SAT solvers

Open questions

Improve succinctness

Do we find stronger proof system such that

- XOR-PAIRS has short proofs?
- preprocessing from current solvers can be handled?

Relation to other proof systems

- There is an alternative approach to prove lower bounds for MICE' using Decision DNNFs techniques. [Bova, Capelli, Mengel, Slivovsky 2016]
- How do MICE′ and the knowledge compilation based proof system relate to each other? [Capelli 2019]
- Relation to the Proof System which will be presented tomorrow? [Bryant, Nawrocki, Avigad, Heule SAT'23]