UNIVERSITY OF TRENTO - Italy

On CNF Conversion for Disjoint SAT Enumeration

Gabriele Masina Giuseppe Spallitta Roberto Sebastiani

SAT Enumeration

- **Task**: Find *all the assignments* that satisfy a given Boolean formula φ

Example

$$\varphi \stackrel{\text{def}}{=} A \vee (B \wedge C)$$

Set of total assignments:

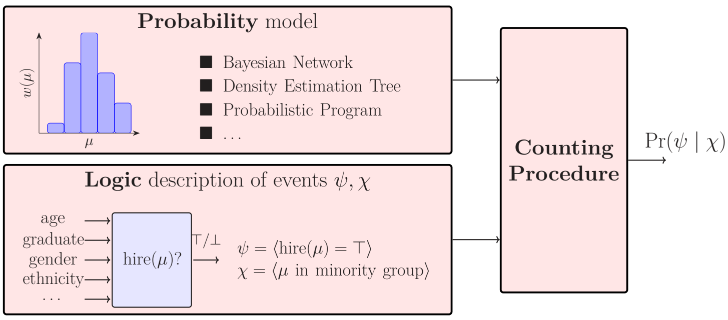
$$\mathcal{TTA}(\varphi) = \{\{A, B, C\}, \{A, B, \neg C\}, \{A, \neg B, C\}, \{A, \neg B, \neg C\}, \{\neg A, B, C\}\}$$

Set of (disjoint) partial assignments:

$$\mathcal{TA}(\varphi) = \{\{A\}, \{\neg A, B, C\}\}$$

- **Goal**: find a $\mathcal{TA}(\varphi)$ as compact as possible
  - $\Longrightarrow$ Why? Compact representation, faster enumeration
- **Key problem**: find *short* partial assignments

Motivation: SAT Enumeration for Probabilistic Inference



How do we *count*?

- Weighted Model Counting (Boolean)
  $\mathsf{WMC}(\varphi, w \mid \mathbf{A}) \stackrel{\text{def}}{=} \sum_{\mu \in \mathcal{TTA}(\varphi)} w(\mu)$
- Weighted Model Integration (SMT($\mathcal{LRA}$))
  $\mathsf{WMI}(\varphi, w | \mathbf{A}, \mathbf{x}) \stackrel{\text{def}}{=} \sum_{\mu^{\mathbf{A}} \cup \mu^{\mathcal{LRA}} \in \mathcal{TTA}(\varphi)} \int_{\mu^{\mathcal{LRA}}} w(\mathbf{x}|\mathbf{A}) d\mathbf{x}$

$$\Pr(\psi \mid \chi) = \frac{\mathsf{WM*}(\psi \wedge \chi)}{\mathsf{WM*}(\chi)}$$

Current approach & Efficiency issues

SAT solvers work with formulas in Conjunctive Normal Form (CNF)
$(l_{11} \vee l_{12} \vee \dots) \wedge (l_{21} \vee l_{22} \vee \dots) \wedge \dots$

- Convert φ to CNF using the Tseitin CNF Encoding $\mathsf{CNF}_{\mathsf{Ts}}$
- Enumerate $\mathcal{TA}(\mathsf{CNF}_{\mathsf{Ts}}(\varphi))$ projected onto the original variables only

Example

In the example above:

- Label each sub-formula with a fresh variable
  $\mathsf{CNF}_{\mathsf{Ts}}(\varphi) = (A \vee S) \wedge \mathsf{CNF}(S \leftrightarrow B \wedge C)$
- Enumerate $\mathcal{TA}(\mathsf{CNF}_{\mathsf{Ts}}(\varphi))$ projected onto $\{A, B, C\}$

$$\mathcal{TA}(\varphi) = \{\{\neg S, A, \neg B\}, \{\neg S, A, B, \neg C\}, \{S, B, C\}\}$$

Notice: Two assignments instead of one!

What causes the issues?

- Definitions as $(S_i \leftrightarrow \varphi_i)$ force to assign a truth value also to (variables in) φ_i
- Partial assignments are unnecessarily-long and $\mathcal{TA}(\varphi)$ is big
- **TLDR: Tseitin CNF is not suitable for enumeration** since **"$\leftrightarrow$" definitions do not allow finding short partial assignments**

Our solution

- Convert the formula in **Negation Normal Form** (NNF)
- Use the **Plaisted&Greenbaum CNF**
  - $\Longrightarrow$ add definitions as $(S_i \rightarrow \varphi_i)$ if φ_i occurs only positively

Example

In the example above:

- φ is already in NNF, label each sub-formula using single implications
  $\mathsf{CNF}_{\mathsf{PG}}(\mathsf{NNF}(\varphi)) = (A \vee S) \wedge \mathsf{CNF}(S \rightarrow B \wedge C)$
- Enumerate $\mathcal{TA}(\mathsf{CNF}_{\mathsf{PG}}(\mathsf{NNF}(\varphi)))$ projected onto $\{A, B, C\}$

$$\mathcal{TA}(\varphi) = \{\{\neg S, A\}, \{S, \neg A, B, C\}\}$$

Notice: Only one assignment!

Why $\mathsf{CNF}_{\mathsf{PG}}$?

- By assigning $\neg S_i$ the **definition** $(S_i \rightarrow \varphi_i)$ **can be "ignored"**
  - $\Longrightarrow$ we are not forced to assign a truth value to (variables in) φ_i anymore

Why NNF?

- If φ_i **occurs positively and negatively**, $\mathsf{CNF}_{\mathsf{PG}}$ adds $(S_i \leftrightarrow \varphi_i)$ anyway
- **NNF splits** φ_i into φ^+ and φ^-, each occurring only positively
- Then $\mathsf{CNF}_{\mathsf{PG}}$ labels them with $(S_i^+ \rightarrow \varphi^+)$ and $(S_i^- \rightarrow \varphi^-)$
- The truth value of φ_i **can be ignored** by assigning $\neg S_i^+$ and $\neg S_i^-$

Experimental results

Setting:

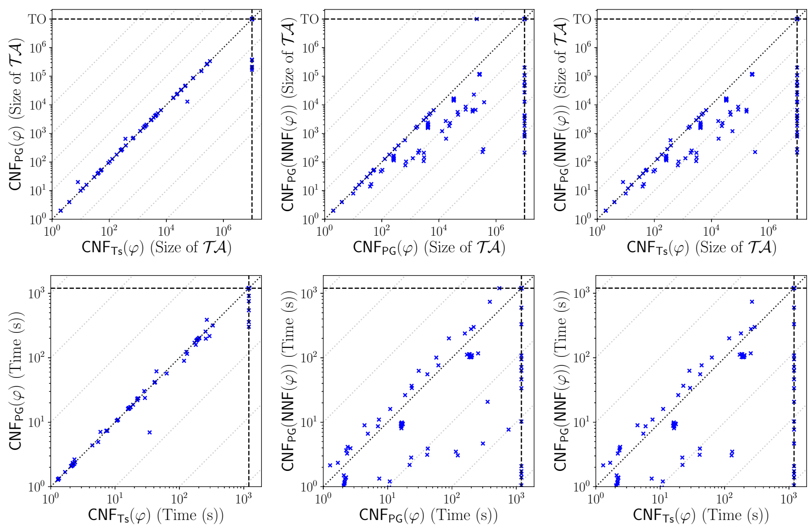
- Convert each non-CNF formula to CNF using $\mathsf{CNF}_{\mathsf{Ts}}$, $\mathsf{CNF}_{\mathsf{PG}}$, or $\mathsf{NNF} + \mathsf{CNF}_{\mathsf{PG}}$
- Enumerate the assignments projected on the original variables using MathSAT

Results & Conclusions

- $\mathsf{CNF}_{\mathsf{Ts}}$ is not good for enumeration
- $\mathsf{CNF}_{\mathsf{PG}}$ solves its problems only in part
- $\mathsf{NNF} + \mathsf{CNF}_{\mathsf{PG}}$ is the best choice
  - $\Longrightarrow$ drastically reduce size of $\mathcal{TA}(\dots)$ and enumeration time by several orders of magnitude
  - **Notice the logarithmic scale of the axes!**

Future work:

- Heuristics to better exploit the encoding
- Extend to non-disjoint SAT enumeration
- Extend to disjoint and non-disjoint SMT enumeration
- Apply it to WMI computation



Enumeration on combinatorial circuits. Timeouts (dashed lines): $\mathsf{CNF}_{\mathsf{Ts}}$ 49/250, $\mathsf{CNF}_{\mathsf{PG}}$ 44/250, $\mathsf{NNF} + \mathsf{CNF}_{\mathsf{PG}}$ 27/250