On the proof complexity of MCSAT

MCSAT vs. Res*(T) vs. CDCL(T)

Gereon Kremer, Erika Ábrahám, Vijay Ganesh

July 10th, 2019 – SC^2 Workshop 2019 – University of Bern

Satisfiability Modulo Theories

Satisfiability problem (for first-order logic)

Is an existentially quantified first-order formula φ valid?

$$\exists x.\varphi(x) \equiv true$$

Satisfiability Modulo Theories

Satisfiability problem (for first-order logic)

Is an existentially quantified first-order formula φ valid?

$$\exists x.\varphi(x) \equiv true$$

Applications:

- Software verification, test-case generation
- Termination proving
- Controller synthesis
- Scheduling and planning
- Product design automation
- And growing ...

RWTH AACHEN UNIVERSITY

Proof systems

Definition (Proof rule and proof systems)

$$\texttt{Proof rule:}\frac{A_1 \quad \cdots \quad A_n}{C_1 \quad \cdots \quad C_n} \qquad \textbf{if } \texttt{S}_1, \ldots, \texttt{S}_m$$

Proof system: set of proof rules

Example: resolution proof system

$$\texttt{Resolution:}\frac{(C \vee l) \quad (D \vee \neg l)}{(C \vee D)} \qquad \textbf{if } \textit{true}$$

Proofs

Let us prove $(a \vee b \vee \neg c) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (\neg a) \equiv \Box$

Proofs

Let us prove $(a \vee b \vee \neg c) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (\neg a) \equiv \square$

$$\dfrac{\dfrac{\dfrac{(a \vee b \vee \neg c) \quad (a \vee \neg b)}{(a \vee \neg c)} \qquad (a \vee c)}{(a)} \qquad (\neg a)}{\square}$$

Proofs

Let us prove $(a \vee b \vee \neg c) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (\neg a) \equiv \square$

$$\dfrac{\dfrac{\dfrac{(a \vee b \vee \neg c) \quad (a \vee \neg b)}{(a \vee \neg c)} \qquad (a \vee c)}{(a)} \qquad (\neg a)}{\square}$$

Definition (Proof size and proof complexity)

Proof size: number of proof rule applications.

Proofs

Let us prove $(a \vee b \vee \neg c) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (\neg a) \equiv \Box$

$$\cfrac{\cfrac{\cfrac{(a \vee b \vee \neg c) \quad (a \vee \neg b)}{(a \vee \neg c)} \qquad (a \vee c)}{(a)} \qquad (\neg a)}{\Box}$$

Definition (Proof size and proof complexity)

Proof size: number of proof rule applications.
Proof complexity: asymptotic proof size of the shortest proof.

Proofs

Let us prove $(a \vee b \vee \neg c) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (\neg a) \equiv \square$

$$\cfrac{\cfrac{\cfrac{(a \vee b \vee \neg c) \quad (a \vee \neg b)}{(a \vee \neg c)} \qquad (a \vee c)}{(a)} \qquad (\neg a)}{\square}$$

Definition (Proof size and proof complexity)

Proof size: number of proof rule applications.
Proof complexity: asymptotic proof size of the shortest proof.

Assumption: every rule costs the same.

MCSAT proof system – overview

Three groups of proof rules:

- Search: CDCL-style SAT solving.
- Conflict: CDCL-style conflict resolution.
- Theory: adds theory reasoning.

de Moura and Jovanović (2013)

Decide | Propagate | Conflict

Resolve | Backjump | Learn

T-Propagate | T-Decide | T-Conflict

RWTH AACHEN UNIVERSITY

MCSAT proof system – overview

Three groups of proof rules:

- Search: CDCL-style SAT solving.
- Conflict: CDCL-style conflict resolution.
- Theory: adds theory reasoning.

de Moura and Jovanović (2013)

Decide | Propagate | Conflict

Resolve | Backjump | Learn

T-Propagate | T-Decide | T-Conflict

Important concepts:

- Theory decisions: like Boolean decisions, but for theory variables.
- Trail: $[\![M, l_1, \neg l_2, C \rightarrow l, x \mapsto \alpha_x, \dots]\!]$
- States: $\langle M, \mathscr{C} \rangle$ and $\langle M, \mathscr{C} \rangle \Vdash C$ (initially $\langle [\![]\!], \mathscr{C} \rangle$)
- $\mathrm{value}(l)$: assigned by Boolean or theory model.

RWTH AACHEN UNIVERSITY

MCSAT proof system – search rules

$$\texttt{Decide:}\ \frac{\langle M, \mathscr{C}\rangle}{\langle [\![M, l]\!], \mathscr{C}\rangle} \qquad \textbf{if } l \textit{ is unassigned}$$

$$\texttt{Propagate:}\ \frac{\langle M, \mathscr{C}\rangle}{\langle [\![M, C \to l]\!], \mathscr{C}\rangle} \qquad \textbf{if } C \textit{ is unit and implies } l$$

$$\texttt{Conflict:}\ \frac{\langle M, \mathscr{C}\rangle}{\langle M, \mathscr{C}\rangle \Vdash C} \qquad \textbf{if } C \in \mathscr{C} \textit{ is conflicting}$$

$$\texttt{Sat:}\ \frac{\langle M, \mathscr{C}\rangle}{\texttt{SAT}} \qquad \textbf{if } M \textit{ is complete and satisfies } \mathscr{C}$$

$$\texttt{Forget:}\ \frac{\langle M, \mathscr{C}\rangle}{\langle M, \mathscr{C} \setminus \{C\}\rangle} \qquad \textbf{if } C \textit{ is a learned clause}$$

MCSAT proof system – conflict rules

$$\texttt{Resolve:}\ \frac{\langle [\![M, D \rightarrow l]\!], \mathscr{C} \rangle \Vdash C}{\langle M, \mathscr{C} \rangle \Vdash R} \qquad \textbf{if}\ \ R = \mathrm{resolve}(C, D, l)$$

$$\texttt{Consume:}\ \frac{\langle [\![M, l \text{ or } D \rightarrow l]\!], \mathscr{C} \rangle \Vdash C}{\langle M, \mathscr{C} \rangle \Vdash C} \qquad \textbf{if}\ \ \neg l \notin C$$

$$\texttt{Backjump:}\ \frac{\langle [\![M, N]\!], \mathscr{C} \rangle \Vdash C}{\langle [\![M, C \rightarrow l]\!], \mathscr{C} \rangle} \qquad \textbf{if}\ \ \begin{array}{l} C \textit{ is unit on } M \textit{ and} \\ N \textit{ starts with a decision} \end{array}$$

$$\texttt{Unsat:}\ \frac{\langle M, \mathscr{C} \rangle \Vdash \textit{false}}{\texttt{UNSAT}} \qquad \textbf{if}\ \ \textit{true}$$

$$\texttt{Learn:}\ \frac{\langle M, \mathscr{C} \rangle \Vdash C}{\langle M, \mathscr{C} \cup \{C\} \rangle \Vdash C} \qquad \textbf{if}\ \ C \textit{ is a new clause}$$

$$\texttt{Restart:}\ \frac{\langle M, \mathscr{C} \rangle \Vdash C}{\langle [\![]\!], \mathscr{C} \rangle} \qquad \textbf{if}\ \ \textit{true}$$

RWTH AACHEN UNIVERSITY

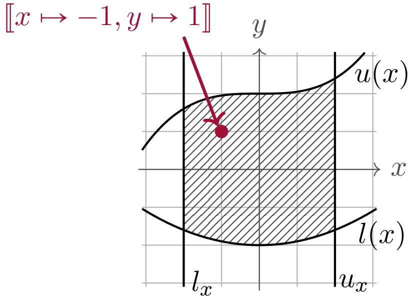
MCSAT proof system – theory rules

$\text{infeasible}(M)$: checks whether M can be extended to a full model.***
$\text{explain}(M) \mapsto C$: clause C excludes a region around M.

***: Terms and conditions may apply.

MCSAT proof system – theory rules

$\mathrm{infeasible}(M)$: checks whether M can be extended to a full model.***
$\mathrm{explain}(M) \mapsto C$: clause C excludes a region around M.



$$C = (x \leqslant -2 \vee x \geqslant 2 \vee y \leqslant l(x) \vee y \geqslant u(x))$$

***: Terms and conditions may apply.

MCSAT proof system – theory rules

$\mathrm{infeasible}(M)$: checks whether M can be extended to a full model.***
$\mathrm{explain}(M) \mapsto C$: clause C excludes a region around M.

$$\texttt{T-Propagate}: \frac{\langle M, \mathscr{C} \rangle}{\langle [\![M, E \rightarrow l]\!], \mathscr{C} \rangle}$$ **if** $\mathrm{infeasible}([\![M, \neg l]\!])$ *and* $E = \mathrm{explain}([\![M, \neg l]\!])$

$$\texttt{T-Decide}: \frac{\langle M, \mathscr{C} \rangle}{\langle [\![M, x \mapsto \alpha_x]\!], \mathscr{C} \rangle}$$ **if** x *is unassigned and* $[\![M, x \mapsto \alpha_x]\!]$ *is consistent*

$$\texttt{T-Conflict}: \frac{\langle M, \mathscr{C} \rangle}{\langle M, \mathscr{C} \rangle \Vdash E}$$ **if** $\mathrm{infeasible}(M)$ *and* $E = \mathrm{explain}(M)$

$$\texttt{T-Consume}: \frac{\langle [\![M, x \mapsto \alpha_x]\!], \mathscr{C} \rangle \Vdash C}{\langle M, \mathscr{C} \rangle \Vdash C}$$ **if** C *is infeasible on* M

$$\texttt{T-Backjump-Decide}: \frac{\langle [\![M, x \mapsto \alpha_x, N]\!], \mathscr{C} \rangle \Vdash C}{\langle [\![M, l]\!], \mathscr{C} \rangle}$$ **if** *Backtracking* $x \mapsto \alpha_x$ *"unassigns" multiple literals from* C *at once*

***: Terms and conditions may apply.

RWTH AACHEN UNIVERSITY

Res*(T) proof system

$$\texttt{Resolution:}\ \frac{(C \lor l) \quad (D \lor \neg l)}{(C \lor D)} \qquad \textbf{if } \textit{true}$$

$$\texttt{(Regular) Theory Derivation:}\ \frac{\varphi}{\varphi \land C} \qquad \textbf{if } \begin{array}{l} T \models C, \\ l \in \varphi \textit{ for all } l \in C \end{array}$$

$$\texttt{Strong Theory Derivation:}\ \frac{\varphi}{\varphi \land C} \qquad \textbf{if } T \models C$$

- `SAT` if no new clause can be generated by `Resolution` or `Regular Theory Derivation`.
- `UNSAT` if $\square$ was generated.

RWTH AACHEN UNIVERSITY

Res*(T) proof system

$$\texttt{Resolution:}\ \frac{(C \vee l) \quad (D \vee \neg l)}{(C \vee D)} \qquad \textbf{if}\ \textit{true}$$

$$\texttt{(Regular) Theory Derivation:}\ \frac{\varphi}{\varphi \wedge C} \qquad \textbf{if}\ \begin{array}{l} T \models C, \\ l \in \varphi \textit{ for all } l \in C \end{array}$$

$$\texttt{Strong Theory Derivation:}\ \frac{\varphi}{\varphi \wedge C} \qquad \textbf{if}\ T \models C$$

- `SAT` if no new clause can be generated by `Resolution` or `Regular Theory Derivation`.
- `UNSAT` if $\Box$ was generated.

We use Strong Theory Derivation!

RWTH AACHEN UNIVERSITY

Relating proof systems

Recall: Proof size and proof complexity

Proof size: number of proof rule applications.
Proof complexity: asymptotic proof size, depending on formula size.

RWTH AACHEN UNIVERSITY

Relating proof systems

Recall: Proof size and proof complexity

Proof size: number of proof rule applications.
Proof complexity: asymptotic proof size, depending on formula size.

Definition (P_1 simulates P_2)

Proof complexity of P_1 is at most polynomially larger for all inputs.

Definition (P_2 is P_1 derivable)

P_1 can simulate every rule of P_2 individually.

RWTH AACHEN UNIVERSITY

Relating proof systems

Recall: Proof size and proof complexity

Proof size: number of proof rule applications.
Proof complexity: asymptotic proof size, depending on formula size.

Definition (P_1 simulates P_2)

Proof complexity of P_1 is at most polynomially larger for all inputs.

Definition (P_2 is P_1 derivable)

P_1 can simulate every rule of P_2 individually.

Example: Res*(T) simulates CDCL(T). *Robere et al. (2018)*

RWTH AACHEN UNIVERSITY

MCSAT and Res*(T) are bisimilar

Theorem

The Res(T) proof system and the MCSAT proof system are bisimilar with respect to their proof complexity on first-order logic with any theory.*

We show: MCSAT is Res*(T) derivable and Res*(T) is MCSAT derivable.

MCSAT and Res*(T) are bisimilar

Theorem

The Res(T) proof system and the MCSAT proof system are bisimilar with respect to their proof complexity on first-order logic with any theory.*

We show: MCSAT is Res*(T) derivable and Res*(T) is MCSAT derivable.

Note that we actually show a slightly stronger statement:

MCSAT and Res*(T) are not only bisimilar but "algorithmically equivalent".

RWTH AACHEN UNIVERSITY

MCSAT simulates Res*(T)

Resolution of $(C \vee l) \wedge (D \vee \neg l)$:

- Decide literals of C and D to $false$.
- Propagate $(C \vee l)$.
- Use $(D \vee \neg l)$ for Conflict.
- Apply Resolve.
- Learn the clause and Restart.

MCSAT simulates Res*(T)

Resolution of $(C \vee l) \wedge (D \vee \neg l)$:

- Decide literals of C and D to $false$.
- Propagate $(C \vee l)$.
- Use $(D \vee \neg l)$ for Conflict.
- Apply Resolve.
- Learn the clause and Restart.

Let $\mathscr{C} = \{(a \vee l), (b \vee \neg l)\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\langle [\![]\!], \mathscr{C} \rangle}{\langle [\![\neg a, \neg b]\!], \mathscr{C} \rangle}}{\langle [\![\neg a, \neg b, (a \vee l) \rightarrow l]\!], \mathscr{C} \rangle}}{\langle [\![\neg a, \neg b, (a \vee l) \rightarrow l]\!], \mathscr{C} \rangle \Vdash (b \vee \neg l)}}{\langle [\![\neg a, \neg b]\!], \mathscr{C} \rangle \Vdash (a \vee b)}}{\langle [\![\neg a, \neg b]\!], \mathscr{C} \cup \{a \vee b\} \rangle \Vdash (a \vee b)}}{\langle [\![]\!], \mathscr{C} \cup \{a \vee b\} \rangle}$$

RWTH AACHEN UNIVERSITY

MCSAT simulates Res*(T)

Resolution of $(C \vee l) \wedge (D \vee \neg l)$:

- Decide literals of C and D to $false$.
- Propagate $(C \vee l)$.
- Use $(D \vee \neg l)$ for Conflict.
- Apply Resolve.
- Learn the clause and Restart.

Strong Theory Derivation of some clause C:

- Decide all literals of C to $false$.
- Apply T-Conflict to obtain C.
- Learn the clause and Restart.

RWTH AACHEN UNIVERSITY

MCSAT simulates Res*(T)

Resolution of $(C \vee l) \wedge (D \vee \neg l)$:

- Decide literals of C and D to $false$.
- Propagate $(C \vee l)$.
- Use $(D \vee \neg l)$ for Conflict.
- Apply Resolve.
- Learn the clause and Restart.

Strong Theory Derivation of some clause C:

- Decide all literals of C to $false$.
- Apply T-Conflict to obtain C.
- Learn the clause and Restart.

$$\dfrac{\dfrac{\dfrac{\langle [\![]\!], \mathscr{C} \rangle}{\langle [\![x < 0, x > 1]\!], \mathscr{C} \rangle}}{\langle [\![x < 0, x > 1]\!], \mathscr{C} \rangle \Vdash (x \geqslant 0 \vee x \leqslant 1)}}{\langle [\![]\!], \mathscr{C} \cup \{(x \geqslant 0 \vee x \leqslant 1)\} \rangle}$$

RWTH AACHEN UNIVERSITY

MCSAT simulates Res*(T)

`Resolution` of $(C \vee l) \wedge (D \vee \neg l)$:

- `Decide` literals of C and D to $false$.
- `Propagate` $(C \vee l)$.
- Use $(D \vee \neg l)$ for `Conflict`.
- Apply `Resolve`.
- Learn the clause and `Restart`.

`Strong Theory Derivation` of some clause C:

- `Decide` all literals of C to $false$.
- Apply `T-Conflict` to obtain C.
- Learn the clause and `Restart`.

$$\cfrac{\cfrac{\cfrac{\langle [\![]\!], \mathscr{C} \rangle}{\langle [\![x < 0, x > 1]\!], \mathscr{C} \rangle}}{\langle [\![x < 0, x > 1]\!], \mathscr{C} \rangle \Vdash (x \geqslant 0 \vee x \leqslant 1)}}{\langle [\![]\!], \mathscr{C} \cup \{(x \geqslant 0 \vee x \leqslant 1)\} \rangle}$$

Theory reasoning: `T-Conflict` checks infeasibility with $\mathrm{infeasible}$. ***

RWTH AACHEN UNIVERSITY

Res*(T) simulates MCSAT

Observation

All clauses in MCSAT "live" at: $\mathscr{C}$, M, conflict clause C.
We only need to simulate rules that create completely new clauses: `Resolve`, `T-Propagate` and `T-Conflict`.

All other rules do not manipulate clauses or only move them around.

RWTH AACHEN UNIVERSITY

Res*(T) simulates MCSAT

Observation

All clauses in MCSAT "live" at: $\mathscr{C}$, M, conflict clause C.
We only need to simulate rules that create completely new clauses:
`Resolve`, `T-Propagate` and `T-Conflict`.

All other rules do not manipulate clauses or only move them around.

- `Resolve`: essentially identical to `Resolution`.
- `T-Propagate` and `T-Conflict`: use explain to generate "a valid theory lemma", we can use `Strong Theory Derivation`.

Some observations

- All the reductions are polynomial.

RWTH AACHEN UNIVERSITY

Some observations

- All the reductions are polynomial.
- Theory decisions completely irrelevant (for the proof).

Some observations

- All the reductions are polynomial.
- Theory decisions completely irrelevant (for the proof).
- Terms and Conditions:
  We assumed $\mathrm{infeasible}$ to be complete, though it is not in practice.
  Incomplete $\mathrm{infeasible}$ needs theory exploration (may be exponential).

RWTH AACHEN UNIVERSITY

Some observations

- All the reductions are polynomial.
- Theory decisions completely irrelevant (for the proof).
- Terms and Conditions:
  We assumed $\mathrm{infeasible}$ to be complete, though it is not in practice.
  Incomplete $\mathrm{infeasible}$ needs theory exploration (may be exponential).

  What did you expect?
  MCSAT moves theory reasoning into the proof system.
  The cost is not new, but only made explicit.
  Is this a problem?

RWTH AACHEN UNIVERSITY

What about CDCL(T)?

- Literature: Res*(T) and CDCL(T) are bisimilar. *Robere et al. (2018)*
- Now: Res*(T) and MCSAT are bisimilar.
- $\Rightarrow$ MCSAT and CDCL(T) are bisimilar ...

RWTH AACHEN UNIVERSITY

What about CDCL(T)?

- Literature: Res*(T) and CDCL(T) are bisimilar. *Robere et al. (2018)*
- Now: Res*(T) and MCSAT are bisimilar.
- $\Rightarrow$ MCSAT and CDCL(T) are bisimilar ...
  with respect to proof complexity.

RWTH AACHEN UNIVERSITY

What about CDCL(T)?

- Literature: Res*(T) and CDCL(T) are bisimilar. *Robere et al. (2018)*
- Now: Res*(T) and MCSAT are bisimilar.
- $\Rightarrow$ MCSAT and CDCL(T) are bisimilar ...
  with respect to proof complexity.

- Also: Res*(T) and MCSAT are “algorithmically equivalent”. ***
- What about MCSAT and CDCL(T)?

RWTH AACHEN UNIVERSITY

What about CDCL(T)?

- Literature: Res*(T) and CDCL(T) are bisimilar. *Robere et al. (2018)*
- Now: Res*(T) and MCSAT are bisimilar.
- $\Rightarrow$ MCSAT and CDCL(T) are bisimilar …
  with respect to proof complexity.

- Also: Res*(T) and MCSAT are "algorithmically equivalent". ***
- What about MCSAT and CDCL(T)?

- We claim: MCSAT and CDCL(T) are "algorithmically equivalent". ***

RWTH AACHEN UNIVERSITY

Conclusion

With respect to proof complexity,

- Res*(T) and CDCL(T) are bisimilar, *Robere et al. (2018)*
- Res*(T) and MCSAT are bisimilar,
- thus CDCL(T) and MCSAT are bisimilar.

Conclusion

With respect to proof complexity,

- Res*(T) and CDCL(T) are bisimilar, *Robere et al. (2018)*
- Res*(T) and MCSAT are bisimilar,
- thus CDCL(T) and MCSAT are bisimilar.

We have seen that

- theory decisions are “only” a heuristic,
- the MCSAT proof system is more powerful than any implementation,
- Res*(T) and MCSAT perform roughly the same theory reasoning.

RWTH AACHEN UNIVERSITY

Conclusion

With respect to proof complexity,

- Res*(T) and CDCL(T) are bisimilar, *Robere et al. (2018)*
- Res*(T) and MCSAT are bisimilar,
- thus CDCL(T) and MCSAT are bisimilar.

We have seen that

- theory decisions are "only" a heuristic,
- the MCSAT proof system is more powerful than any implementation,
- Res*(T) and MCSAT perform roughly the same theory reasoning.

We conjecture "algorithmic equivalency" of CDCL(T) and MCSAT.

RWTH AACHEN UNIVERSITY

References

de Moura, L. and Jovanović, D. (2013). A model-constructing satisfiability calculus. In Giacobazzi, R., Berdine, J., and Mastroeni, I., editors, *Verification, Model Checking, and Abstract Interpretation*, volume 7737, pages 1–12.

Robere, R., Kolokolova, A., and Ganesh, V. (2018). The proof complexity of smt solvers. In Chockler, H. and Weissenbacher, G., editors, *Computer Aided Verification*, volume 10982, pages 275–293.