JOHANNES KEPLER
UNIVERSITY LINZ

Implicit Hitting Set Algorithms for Maximum Satisfiability Modulo Theories

Katalin Fazekas[1] Fahiem Bacchus[2] Armin Biere[1]

[1] Johannes Kepler University Linz, Austria
[2] University of Toronto, Canada

Oxford, 14. July, 2018

9th International Joint Conference on Automated Reasoning

UNIVERSITY OF TORONTO JOHANNES KEPLER UNIVERSITY LINZ logics LOGICAL METHODS IN COMPUTER SCIENCE FWF

Motivation

- Satisfiability Modulo Theories (SMT)
  - Satisfiability of ground **first-order** formula wrt. $\mathcal{T}$
  - Widely used: model checking, test case generation etc.

Motivation

- Satisfiability Modulo Theories (SMT)
  - Satisfiability of ground **first-order** formula wrt. $\mathcal{T}$
  - Widely used: model checking, test case generation etc.
- Maximum Satisfiability (MaxSAT)
  - **Optimization** in Boolean domain
  - Widely used: planning, fault localization, etc.

Motivation

- Satisfiability Modulo Theories (SMT)
  - Satisfiability of ground **first-order** formula wrt. $\mathcal{T}$
  - Widely used: model checking, test case generation etc.
- Maximum Satisfiability (MaxSAT)
  - **Optimization** in Boolean domain
  - Widely used: planning, fault localization, etc.
- Maximum Satisfiability Modulo Theories (MaxSMT)
  - Optimization over Boolean abstraction of first-order formula
  - Extension of SMT with Boolean-based optimization
  - Extension of MaxSAT with $\mathcal{T}$-reasoning
  - Many more potential application

Goals

- Efficient, sound MaxSMT solving

Goals

- Efficient, sound MaxSMT solving
- Harvest advances of SMT and MaxSAT:
  - From SMT: separate $\mathcal{T}$-reasoning and Boolean reasoning (as in DPLL($\mathcal{T}$))
  - From MaxSAT: decouple optimization from Boolean reasoning (as in IHS)

Goals

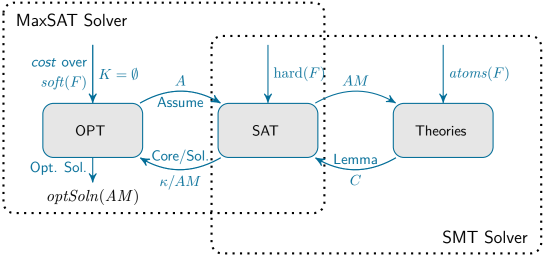
- Efficient, sound MaxSMT solving
- Harvest advances of SMT and MaxSAT:
  - From SMT: separate $\mathcal{T}$-reasoning and Boolean reasoning (as in DPLL($\mathcal{T}$))
  - From MaxSAT: decouple optimization from Boolean reasoning (as in IHS)
- Separation between optimization, propositional and theory specific reasoning
  - Exploitation of more efficient specialized solvers

Contributions

- Formal abstract framework for describing MaxSMT
  - Transition system for formal reasoning
  - Flexible: almost any scheduling leads to a solution
  - Additionally extends DPLL($\mathcal{T}$) with assumptions

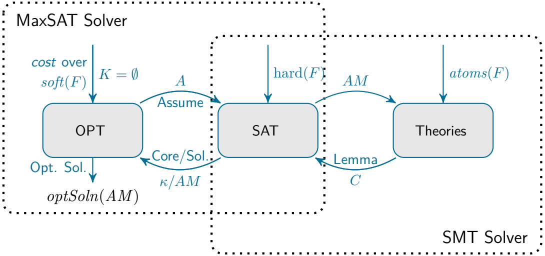
Contributions

- Formal abstract framework for describing MaxSMT
  - Transition system for formal reasoning
  - Flexible: almost any scheduling leads to a solution
  - Additionally extends DPLL($\mathcal{T}$) with assumptions
- General solver architecture



Contributions

- Formal abstract framework for describing MaxSMT
  - Transition system for formal reasoning
  - Flexible: almost any scheduling leads to a solution
  - Additionally extends DPLL($\mathcal{T}$) with assumptions
- General solver architecture



- Evaluation of some of the possible instantiations

Overview

Maximum Satisfiability

Hitting Sets

Implicit Hitting Set Algorithms for MaxSAT

Implicit Hitting Set Algorithms for MaxSMT

Instantiations & Experiments

Conclusion

Maximum Satisfiability - MaxSAT

$$\mathcal{F} = \boldsymbol{(\neg x_1 \vee \neg x_2\,; 1) \wedge (\neg x_2 \vee x_3\,; 1) \wedge (\neg x_3 \vee \neg x_4\,; 1) \wedge (x_1\,; 1) \wedge (x_2\,; 1) \wedge (x_4\,; 1)}$$

- Truth assignment for x_1, x_2, x_3, x_4 that maximizes the sum of weights of satisfied clauses

Maximum Satisfiability - MaxSAT

$\mathcal{F} = (\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2}\,; \mathbf{1}) \wedge (\neg \boldsymbol{x_2} \vee \boldsymbol{x_3}\,; \mathbf{1}) \wedge (\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4}\,; \mathbf{1}) \wedge (\boldsymbol{x_1}\,; \mathbf{1}) \wedge (\boldsymbol{x_2}\,; \mathbf{1}) \wedge (\boldsymbol{x_4}\,; \mathbf{1})$

- Truth assignment for x_1, x_2, x_3, x_4 that maximizes the sum of weights of satisfied clauses
- Weighted clauses $(\boldsymbol{C}\,; \boldsymbol{n})$: cost of falsification of C is n
  - $(\boldsymbol{C}\,; \boldsymbol{\infty})$: hard clauses must be satisfied ($\mathrm{hard}(\mathcal{F})$ vs. $\mathrm{soft}(\mathcal{F})$)

Maximum Satisfiability - MaxSAT

$$\mathcal{F} = (\neg x_1 \vee \neg x_2 \,; 1) \wedge (\neg x_2 \vee x_3 \,; 1) \wedge (\neg x_3 \vee \neg x_4 \,; 1) \wedge (x_1 \,; 1) \wedge (x_2 \,; 1) \wedge (x_4 \,; 1)$$

- Truth assignment for x_1, x_2, x_3, x_4 that maximizes the sum of weights of satisfied clauses
- Weighted clauses $(C\,; n)$: cost of falsification of C is n
  - $(C\,; \infty)$: hard clauses must be satisfied ($\mathrm{hard}(\mathcal{F})$ vs. $\mathrm{soft}(\mathcal{F})$)
- **Solution**: an assignment that satisfies all hard clauses

Maximum Satisfiability - MaxSAT

$$\mathcal{F} = (\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2} \,; \mathbf{1}) \wedge (\neg \boldsymbol{x_2} \vee \boldsymbol{x_3} \,; \mathbf{1}) \wedge (\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4} \,; \mathbf{1}) \wedge (\boldsymbol{x_1} \,; \mathbf{1}) \wedge (\boldsymbol{x_2} \,; \mathbf{1}) \wedge (\boldsymbol{x_4} \,; \mathbf{1})$$

- Truth assignment for x_1, x_2, x_3, x_4 that maximizes the sum of weights of satisfied clauses
- Weighted clauses $(\boldsymbol{C} \,; \boldsymbol{n})$: cost of falsification of C is n
  - $(\boldsymbol{C} \,; \infty)$: hard clauses must be satisfied ($\mathrm{hard}(\mathcal{F})$ vs. $\mathrm{soft}(\mathcal{F})$)
- **Solution**: an assignment that satisfies all hard clauses
- **Optimal Solution**: solution with highest sum of weights of satisfied soft clauses

Overview

Maximum Satisfiability

Hitting Sets

Implicit Hitting Set Algorithms for MaxSAT

Implicit Hitting Set Algorithms for MaxSMT

Instantiations & Experiments

Conclusion

Unsatisfiable Cores

Definition (Unsatisfiable Core)

An unsatisfiable core κ for a formula $\mathcal{F}$ is a subset of soft clauses that when combined with the hard clauses forms an unsatisfiable set of clauses:

$$\kappa \subseteq \mathrm{soft}(\mathcal{F}) \text{ s.t. } \mathrm{hard}(\mathcal{F}) \cup \kappa \text{ is UNSAT}$$

Unsatisfiable Cores

Definition (Unsatisfiable Core)

An unsatisfiable core κ for a formula $\mathcal{F}$ is a subset of soft clauses that when combined with the hard clauses forms an unsatisfiable set of clauses:

$$\kappa \subseteq \text{soft}(\mathcal{F}) \text{ s.t. } \text{hard}(\mathcal{F}) \cup \kappa \text{ is UNSAT}$$

$$\mathcal{F} = \boldsymbol{(\neg x_1 \vee \neg x_2\,;1) \wedge (\neg x_2 \vee x_3\,;1) \wedge (\neg x_3 \vee \neg x_4\,;1) \wedge (x_1\,;1) \wedge (x_2\,;1) \wedge (x_4\,;1)}$$

Unsatisfiable Cores

Definition (Unsatisfiable Core)

An unsatisfiable core κ for a formula $\mathcal{F}$ is a subset of soft clauses that when combined with the hard clauses forms an unsatisfiable set of clauses:

$$\kappa \subseteq \mathrm{soft}(\mathcal{F}) \text{ s.t. } \mathrm{hard}(\mathcal{F}) \cup \kappa \text{ is UNSAT}$$

$$\mathcal{F} = \boldsymbol{(\neg x_1 \vee \neg x_2\,;1) \wedge (\neg x_2 \vee x_3\,;1) \wedge (\neg x_3 \vee \neg x_4\,;1) \wedge (x_1\,;1) \wedge (x_2\,;1) \wedge (x_4\,;1)}$$

$$\kappa_0 = \{\boldsymbol{((\neg x_1 \vee \neg x_2)\,;1), ((\neg x_2 \vee x_3)\,;1), ((\neg x_3 \vee \neg x_4)\,;1), (x_1\,;1), (x_2\,;1), (x_4\,;1)}\}$$

Unsatisfiable Cores

Definition (Unsatisfiable Core)

An unsatisfiable core κ for a formula $\mathcal{F}$ is a subset of soft clauses that when combined with the hard clauses forms an unsatisfiable set of clauses:

$$\kappa \subseteq \mathrm{soft}(\mathcal{F}) \text{ s.t. } \mathrm{hard}(\mathcal{F}) \cup \kappa \text{ is UNSAT}$$

$$\mathcal{F} = \boldsymbol{(\neg x_1 \vee \neg x_2\,;1) \wedge (\neg x_2 \vee x_3\,;1) \wedge (\neg x_3 \vee \neg x_4\,;1) \wedge (x_1\,;1) \wedge (x_2\,;1) \wedge (x_4\,;1)}$$

$$\kappa_0 = \{\boldsymbol{((\neg x_1 \vee \neg x_2)\,;1), ((\neg x_2 \vee x_3)\,;1), ((\neg x_3 \vee \neg x_4)\,;1), (x_1\,;1), (x_2\,;1), (x_4\,;1)}\}$$

$$\kappa_1 = \{\boldsymbol{((\neg x_1 \vee \neg x_2)\,;1), (x_1\,;1), (x_2\,;1)}\}$$

Unsatisfiable Cores

Definition (Unsatisfiable Core)

An unsatisfiable core κ for a formula $\mathcal{F}$ is a subset of soft clauses that when combined with the hard clauses forms an unsatisfiable set of clauses:

$$\kappa \subseteq \mathrm{soft}(\mathcal{F}) \text{ s.t. } \mathrm{hard}(\mathcal{F}) \cup \kappa \text{ is UNSAT}$$

$$\mathcal{F} = (\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2}\, ; \mathbf{1}) \wedge (\neg \boldsymbol{x_2} \vee \boldsymbol{x_3}\, ; \mathbf{1}) \wedge (\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4}\, ; \mathbf{1}) \wedge (\boldsymbol{x_1}\, ; \mathbf{1}) \wedge (\boldsymbol{x_2}\, ; \mathbf{1}) \wedge (\boldsymbol{x_4}\, ; \mathbf{1})$$

$$\kappa_0 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\, ; \mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\, ; \mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\, ; \mathbf{1}), (\boldsymbol{x_1}\, ; \mathbf{1}), (\boldsymbol{x_2}\, ; \mathbf{1}), (\boldsymbol{x_4}\, ; \mathbf{1})\}$$

$$\kappa_1 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\, ; \mathbf{1}), (\boldsymbol{x_1}\, ; \mathbf{1}), (\boldsymbol{x_2}\, ; \mathbf{1})\}$$

$$\kappa_2 = \{((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\, ; \mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\, ; \mathbf{1}), (\boldsymbol{x_2}\, ; \mathbf{1}), (\boldsymbol{x_4}\, ; \mathbf{1})\}$$

Unsatisfiable Cores

Definition (Unsatisfiable Core)

An unsatisfiable core κ for a formula $\mathcal{F}$ is a subset of soft clauses that when combined with the hard clauses forms an unsatisfiable set of clauses:

$$\kappa \subseteq \mathrm{soft}(\mathcal{F}) \text{ s.t. } \mathrm{hard}(\mathcal{F}) \cup \kappa \text{ is UNSAT}$$

$$\mathcal{F} = \boldsymbol{(\neg x_1 \vee \neg x_2\,;1) \wedge (\neg x_2 \vee x_3\,;1) \wedge (\neg x_3 \vee \neg x_4\,;1) \wedge (x_1\,;1) \wedge (x_2\,;1) \wedge (x_4\,;1)}$$

$$\kappa_0 = \{\boldsymbol{((\neg x_1 \vee \neg x_2)\,;1)}, \boldsymbol{((\neg x_2 \vee x_3)\,;1)}, \boldsymbol{((\neg x_3 \vee \neg x_4)\,;1)}, \boldsymbol{(x_1\,;1)}, \boldsymbol{(x_2\,;1)}, \boldsymbol{(x_4\,;1)}\}$$

$$\kappa_1 = \{\boldsymbol{((\neg x_1 \vee \neg x_2)\,;1)}, \boldsymbol{(x_1\,;1)}, \boldsymbol{(x_2\,;1)}\}$$

$$\kappa_2 = \{\boldsymbol{((\neg x_2 \vee x_3)\,;1)}, \boldsymbol{((\neg x_3 \vee \neg x_4)\,;1)}, \boldsymbol{(x_2\,;1)}, \boldsymbol{(x_4\,;1)}\}$$

$\dots$

Hitting Sets

Definition (Hitting Set)

Let K be a set of cores, i.e., a set of sets of soft clauses. A hitting set η of K is a set of soft clauses that has a non-empty intersection with every set in K: $\forall \kappa \in K : \eta \cap \kappa \neq \emptyset$

Hitting Sets

Definition (Hitting Set)

Let K be a set of cores, i.e., a set of sets of soft clauses. A hitting set η of K is a set of soft clauses that has a non-empty intersection with every set in K: $\forall \kappa \in K : \eta \cap \kappa \neq \emptyset$

$\kappa_0 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$\kappa_1 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1})\}$

$\kappa_2 = \{((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

Hitting Sets

Definition (Hitting Set)

Let K be a set of cores, i.e., a set of sets of soft clauses. A hitting set η of K is a set of soft clauses that has a non-empty intersection with every set in K: $\forall\kappa \in K : \eta \cap \kappa \neq \emptyset$

$\kappa_0 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$\kappa_1 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1})\}$

$\kappa_2 = \{((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$$\mathrm{HS}(\kappa_0, \kappa_1, \kappa_2) : \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1})\} \quad \mathrm{cost} \sum : 2$$

Hitting Sets

Definition (Hitting Set)

Let K be a set of cores, i.e., a set of sets of soft clauses. A hitting set η of K is a set of soft clauses that has a non-empty intersection with every set in K: $\forall \kappa \in K : \eta \cap \kappa \neq \emptyset$

$\kappa_0 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$\kappa_1 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1})\}$

$\kappa_2 = \{((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$$\mathrm{HS}(\kappa_0, \kappa_1, \kappa_2) : \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1})\} \quad \mathrm{cost} \sum : 2$$

- Minimum Cost Hitting Set: Hitting set with cost less than or equal to the cost of any other hitting set

Hitting Sets

Definition (Hitting Set)

Let K be a set of cores, i.e., a set of sets of soft clauses. A hitting set η of K is a set of soft clauses that has a non-empty intersection with every set in K: $\forall\kappa \in K : \eta \cap \kappa \neq \emptyset$

$\kappa_0 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$\kappa_1 = \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), (\boldsymbol{x_1}\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1})\}$

$\kappa_2 = \{((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1}), ((\neg \boldsymbol{x_3} \vee \neg \boldsymbol{x_4})\,;\mathbf{1}), (\boldsymbol{x_2}\,;\mathbf{1}), (\boldsymbol{x_4}\,;\mathbf{1})\}$

$$\mathrm{HS}(\kappa_0, \kappa_1, \kappa_2) : \{((\neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2})\,;\mathbf{1}), ((\neg \boldsymbol{x_2} \vee \boldsymbol{x_3})\,;\mathbf{1})\} \quad \text{cost} \sum : 2$$

- Minimum Cost Hitting Set: Hitting set with cost less than or equal to the cost of any other hitting set

$$\mathrm{MinHS}(\kappa_0, \kappa_1, \kappa_2) : \{(\boldsymbol{x_2}\,;\mathbf{1})\} \quad \text{cost} \sum : 1$$

Overview

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

To solve a MaxSAT problem F it is enough to find a minimum cost hitting set η of **all** the unsatisfiable cores (U) of F.

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

To solve a MaxSAT problem F it is enough to find a minimum cost hitting set η of **all** the unsatisfiable cores (U) of F.

- $\forall \kappa \in U : \eta \cap \kappa \neq \emptyset \leftrightarrow \{F - \eta\}$ is SAT

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

To solve a MaxSAT problem F it is enough to find a minimum cost hitting set η of **all** the unsatisfiable cores (U) of F.

- $\forall \kappa \in U : \eta \cap \kappa \neq \emptyset \leftrightarrow \{F - \eta\}$ is SAT
- η is MinHS(U) $\leftrightarrow$ all satisfying assignments of $\{F - \eta\}$ is optimal

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

Issue: We do not know all the unsatisfiable cores in advance.

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

Issue: We do not know all the unsatisfiable cores in advance.

Approach: Calculate a hitting set η for the already known unsatisfiable cores (K).

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

Issue: We do not know all the unsatisfiable cores in advance.

Approach: Calculate a hitting set η for the already known unsatisfiable cores (K).

- If $\{F - \eta\}$ is UNSAT: new core κ can be added to K.

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

Issue: We do not know all the unsatisfiable cores in advance.

Approach: Calculate a hitting set η for the already known unsatisfiable cores (K).

- If $\{F - \eta\}$ is UNSAT: new core κ can be added to K.
- If $\{F - \eta\}$ is SAT:

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

Issue: We do not know all the unsatisfiable cores in advance.

Approach: Calculate a hitting set η for the already known unsatisfiable cores (K).

- If $\{F - \eta\}$ is UNSAT: new core κ can be added to K.
- If $\{F - \eta\}$ is SAT:
  - If η is MinHS: Any satisfying assignment is guaranteed to be optimal.

Hitting Sets & Maximum Satisfiability

J. Davies, F. Bacchus: Solving MaxSAT by Solving a Sequence of Simpler SAT Instances (CP 2011)

Issue: We do not know all the unsatisfiable cores in advance.

Approach: Calculate a hitting set η for the already known unsatisfiable cores (K).

- If $\{F - \eta\}$ is UNSAT: new core κ can be added to K.
- If $\{F - \eta\}$ is SAT:
  - If η is MinHS: Any satisfying assignment is guaranteed to be optimal.
  - If η is arbritary HS: No guarantee of optimality.

IHS Algorithms for MaxSAT

$$\mathcal{F}$$

$$\left\{ \begin{array}{ll} (\neg x_1 \lor \neg x_2) & x_1 \\ (\neg x_2 \lor x_3) & x_2 \\ (\neg x_3 \lor \neg x_4) & x_4 \end{array} \right\}$$

IHS Algorithms for MaxSAT

$$\mathcal{F}$$

$$\left\{ \begin{array}{ccc} (\neg x_1 \vee \neg x_2 \vee a_1) & \neg a_1 & x_1 \\ (\neg x_2 \vee x_3 \vee a_2) & \neg a_2 & x_2 \\ (\neg x_3 \vee \neg x_4 \vee a_3) & \neg a_3 & x_4 \end{array} \right\}$$

IHS Algorithms for MaxSAT

$$\mathcal{F} \qquad\qquad \mathsf{HS}(K)$$

$$\left\{ \begin{array}{ccc} \boldsymbol{(\neg x_1 \vee \neg x_2 \vee a_1)} & \boldsymbol{\neg a_1} & \boldsymbol{x_1} \\ \boldsymbol{(\neg x_2 \vee x_3 \vee a_2)} & \boldsymbol{\neg a_2} & \boldsymbol{x_2} \\ \boldsymbol{(\neg x_3 \vee \neg x_4 \vee a_3)} & \boldsymbol{\neg a_3} & \boldsymbol{x_4} \end{array} \right\} \qquad \left\{ \quad \right\}$$

cost over $soft(\mathcal{F})$

$K = \emptyset$

OPT

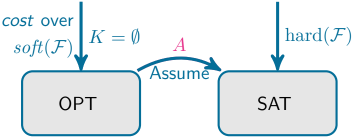
SAT

IHS Algorithms for MaxSAT

$$\mathcal{F} \qquad\qquad\qquad\qquad\qquad\qquad \mathrm{HS}(K)$$

$$\left\{ \begin{array}{ccc} (\neg x_1 \vee \neg x_2 \vee a_1) & \neg a_1 & x_1 \\ (\neg x_2 \vee x_3 \vee a_2) & \neg a_2 & x_2 \\ (\neg x_3 \vee \neg x_4 \vee a_3) & \neg a_3 & x_4 \end{array} \right\} \quad \left\{ \quad \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

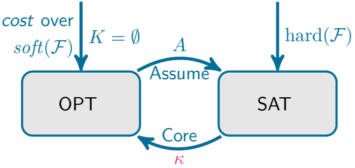


IHS Algorithms for MaxSAT

$$\mathcal{F} \qquad\qquad\qquad\qquad \text{HS}(K)$$

$$\left\{ \begin{array}{lll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & x_1 \\ (\neg x_2 \lor x_3 \lor a_2) & \neg a_2 & x_2 \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \left\{ \qquad \right\}$$

$\text{SAT}(\text{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

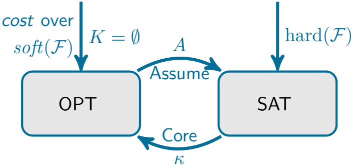


IHS Algorithms for MaxSAT

$$\mathcal{F} \qquad\qquad\qquad\qquad \text{HS}(K)$$

$$\left\{\begin{array}{lll} (\neg x_1 \vee \neg x_2 \vee a_1) & \neg a_1 & \\ (\neg x_2 \vee x_3 \vee a_2) & \neg a_2 & x_2 \\ (\neg x_3 \vee \neg x_4 \vee a_3) & \neg a_3 & x_4 \end{array}\right\} \qquad \left\{\begin{array}{c} x_1 \\ \\ \\ \end{array}\right\}$$

$\text{SAT}(\text{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

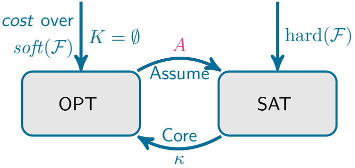


IHS Algorithms for MaxSAT

$$\mathcal{F} = \left\{ \begin{array}{lll} (\neg x_1 \vee \neg x_2 \vee a_1) & \neg a_1 & \\ (\neg x_2 \vee x_3 \vee a_2) & \neg a_2 & x_2 \\ (\neg x_3 \vee \neg x_4 \vee a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \mathsf{HS}(K) = \left\{ \begin{array}{c} x_1 \\ \\ \\ \end{array} \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_2, x_4\}$?

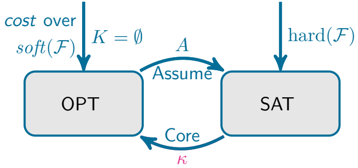


IHS Algorithms for MaxSAT

$$\mathcal{F} = \left\{ \begin{array}{lll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & \\ (\neg x_2 \lor x_3 \lor a_2) & \neg a_2 & x_2 \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \mathsf{HS}(K) = \left\{ \begin{array}{c} x_1 \\ \\ \\ \end{array} \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, \quad x_2, x_4\}$?

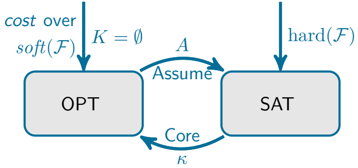


IHS Algorithms for MaxSAT

$$\overset{\mathcal{F}}{\left\{ \begin{array}{lll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & \\ (\neg x_2 \lor x_3 \lor a_2) & & x_2 \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\}} \qquad \overset{\mathsf{HS}(K)}{\left\{ \begin{array}{c} x_1 \\ \neg a_2 \\ \end{array} \right\}}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, \quad x_2, x_4\}$?



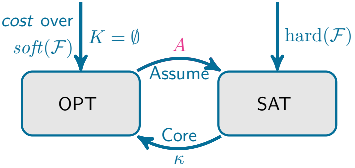
IHS Algorithms for MaxSAT

$$\mathcal{F} = \left\{ \begin{array}{lll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & \\ (\neg x_2 \lor x_3 \lor a_2) & & x_2 \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \mathsf{HS}(K) = \left\{ \begin{array}{c} x_1 \\ \neg a_2 \\ \\ \end{array} \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_3, x_2, x_4\}$?



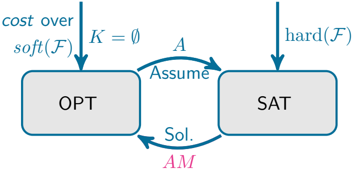
IHS Algorithms for MaxSAT

$$\mathcal{F} = \left\{ \begin{array}{ll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 \\ (\neg x_2 \lor x_3 \lor a_2) & \qquad x_2 \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 \quad x_4 \end{array} \right\} \qquad \mathsf{HS}(K) = \left\{ \begin{array}{c} x_1 \\ \neg a_2 \\ \end{array} \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_3, x_2, x_4\}$?



IHS Algorithms for MaxSAT

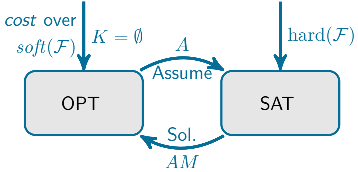
$$\mathcal{F} \qquad\qquad\qquad\qquad\qquad \text{MinHS}(K)$$

$$\left\{ \begin{array}{lll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & x_1 \\ (\neg x_2 \lor x_3 \lor a_2) & \neg a_2 & \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \left\{ \; x_2 \; \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_3, x_2, x_4\}$?



IHS Algorithms for MaxSAT

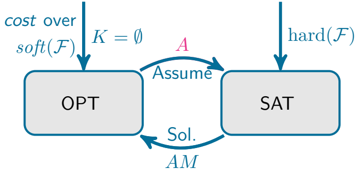
$$\mathcal{F} \quad \left\{ \begin{array}{lll} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & x_1 \\ (\neg x_2 \lor x_3 \lor a_2) & \neg a_2 & \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \text{MinHS}(K) \quad \left\{ \; x_2 \; \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_4\}$?



IHS Algorithms for MaxSAT

$$\mathcal{F} \qquad\qquad\qquad\qquad\qquad\qquad \text{MinHS}(K)$$

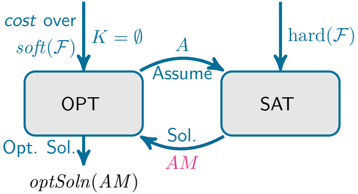
$$\left\{ \begin{array}{ccc} (\neg x_1 \lor \neg x_2 \lor a_1) & \neg a_1 & x_1 \\ (\neg x_2 \lor x_3 \lor a_2) & \neg a_2 & \\ (\neg x_3 \lor \neg x_4 \lor a_3) & \neg a_3 & x_4 \end{array} \right\} \qquad \left\{ \begin{array}{c} \\ x_2 \\ \\ \end{array} \right\}$$

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_3, x_2, x_4\}$?

$\mathrm{SAT}(\mathrm{hard}(\mathcal{F}))$ assuming $\{\neg a_1, \neg a_2, \neg a_3, x_1, x_4\}$?



Overview

IHS Algorithms for MaxSMT

$$\left\{ \begin{array}{ccc} (t_1 \neq t_2 \lor t_1 \neq t_1 \lor a_1) & \neg a_1 & t_1 = t_2 \\ (t_1 \neq t_1 \lor x_3 \lor a_2) & \neg a_2 & t_1 = t_1 \\ (\neg x_3 \lor f(t_1) \neq f(t_2) \lor a_3) & \neg a_3 & f(t_1) = f(t_2) \end{array} \right\}$$

IHS Algorithms for MaxSMT

$$\left\{ \begin{array}{ccc} (t_1 \neq t_2 \lor t_1 \neq t_1 \lor a_1) & \neg a_1 & t_1 = t_2 \\ (t_1 \neq t_1 \lor x_3 \lor a_2) & \neg a_2 & t_1 = t_1 \\ (\neg x_3 \lor f(t_1) \neq f(t_2) \lor a_3) & \neg a_3 & f(t_1) = f(t_2) \end{array} \right\}$$

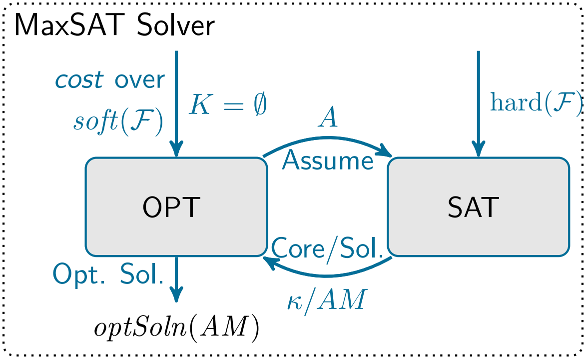
- Solution in MaxSMT: satisfies all hard clauses and all **theory axioms**

IHS Algorithms for MaxSMT

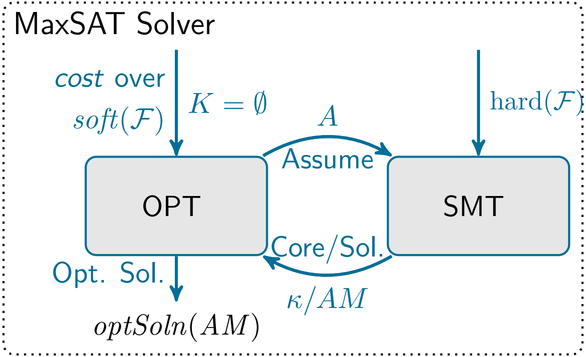
$$\left\{ \begin{array}{ccc} (t_1 \neq t_2 \lor t_1 \neq t_1 \lor a_1) & \neg a_1 & t_1 = t_2 \\ (t_1 \neq t_1 \lor x_3 \lor a_2) & \neg a_2 & t_1 = t_1 \\ (\neg x_3 \lor f(t_1) \neq f(t_2) \lor a_3) & \neg a_3 & f(t_1) = f(t_2) \end{array} \right\}$$

- Solution in MaxSMT: satisfies all hard clauses and all **theory axioms**
- When should we start to consider these axioms?

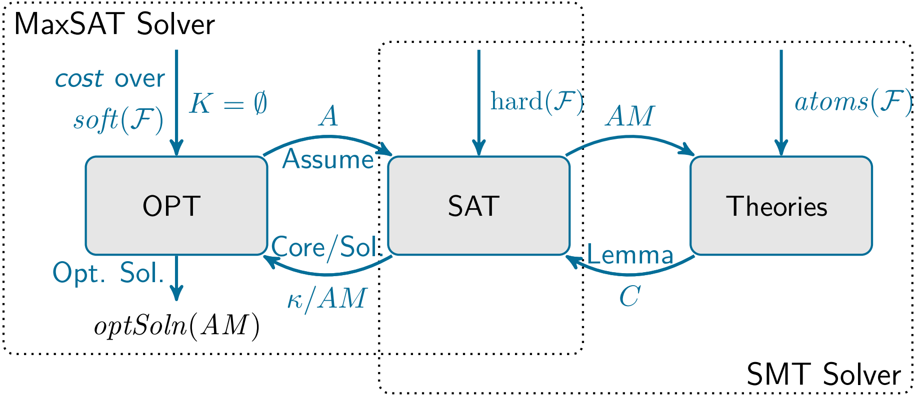
IHS Algorithms for MaxSMT



IHS Algorithms for MaxSMT



IHS Algorithms for MaxSMT



A-MaxSMT Calculus

SAT/SMT-Transition

$(LB, UB, \mu) \mid K \mid \langle * \rangle \Longrightarrow$
$\quad (LB, UB, \mu) \mid K \mid \langle *' \rangle$

if $\left\{ \begin{array}{l} *' \text{ is reachable from } * \text{ by} \\ \text{a single } \textbf{A-Sat}/\textbf{A-Smt} \text{ transition step} \end{array} \right.$

Core

$(LB, UB, \mu) \mid K \mid \langle \mathit{conflict}(F, C) \rangle \Longrightarrow$
$\quad (LB, UB, \mu) \mid K, \kappa \mid \langle \mathit{conflict}(F, C) \rangle$

if $\left\{ \begin{array}{l} \kappa = \{(\neg \ell) \mid \ell \in C\} \text{ and } \kappa \notin K \\ (\kappa \text{ is set of soft clauses}) \end{array} \right.$

HS

$(LB, UB, \mu) \mid K \mid \langle * \rangle \Longrightarrow$
$\quad (LB, UB, \mu) \mid K \mid \langle A' \mid \emptyset \mid F \rangle$

if $\left\{ \begin{array}{l} \eta = HS(K) \\ A' = \{\ell \mid (\ell) \in (\mathit{soft}(F) - \eta)\} \end{array} \right.$

MinHS

$(LB, UB, \mu) \mid K \mid \langle * \rangle \Longrightarrow$
$\quad (LB', UB, \mu) \mid K \mid \langle A' \mid \emptyset \mid F \rangle$

if $\left\{ \begin{array}{l} \eta = \mathit{minHS}(K) \\ A' = \{\ell \mid (\ell) \in (\mathit{soft}(F) - \eta)\} \\ LB' = \max(LB, \mathit{cost}(\eta)) \end{array} \right.$

ImprovedSolution

$(LB, UB, \mu) \mid K \mid \langle T\text{-}SAT(AM,F) \rangle \Longrightarrow$
$\quad (LB, \mathit{cost}(AM), AM) \mid K \mid \langle T\text{-}SAT(AM,F) \rangle$

if $\mathit{cost}(AM) < UB$

OptimalSolution

$(LB, UB, \mu) \mid K \mid \langle * \rangle \Longrightarrow \mathit{optSoln}(\mu)$

if $LB \geq UB$

A-SMT Calculus

UnitProp

$A \mid M \mid F \Longrightarrow A \mid M\,\ell \mid F$ **if** $\begin{cases} \text{There is a clause } (C \vee \ell) \in F \text{ s.t.} \\ AM \models \neg C \text{ and } atom(\ell) \notin atoms(AM) \end{cases}$

Decide

$A \mid M \mid F \Longrightarrow A \mid M\,\ell^d \mid F$ **if** $atom(\ell) \in \Big(atoms(F) \setminus atoms(AM)\Big)$

T-**Backjump**

$A \mid M\ell^d N \mid F \Longrightarrow A \mid M\,\ell' \mid F$ **if** $\begin{cases} \text{There is a clause } C \in F \text{ s.t. } AM\ell^d N \models \neg C \\ \text{and a clause } C' \vee \ell' \text{ s.t. } F \models_T C' \vee \ell', \\ \quad AM \models \neg C' \text{ and } atom(\ell') \in atoms(\ell^d N) \end{cases}$

T-**Learn**

$A \mid M \mid F \Longrightarrow A \mid M \mid F, C$ **if** $\begin{cases} F \models_T C \text{ and } C \notin F \\ atoms(C) \subseteq \Big(atoms(F) \cup atoms(AM)\Big) \end{cases}$

T-**Forget**

$A \mid M \mid F, C \Longrightarrow A \mid M \mid F$ **if** $F \models_T C$

T-**Model**

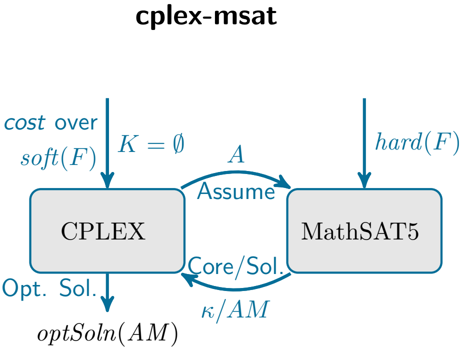
$A \mid M \mid F \Longrightarrow T\text{-}SAT(AM, F)$ **if** $AM \models_T F$

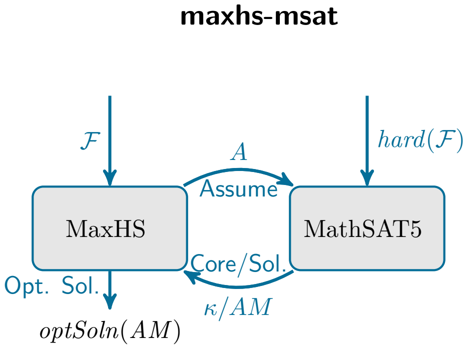
UnSat

$A \mid M \mid F \Longrightarrow conflict(F, C)$ **if** $\begin{cases} \text{There is a clause } D \in F \text{ s.t. } AM \models \neg D \\ M \text{ contains no decision literals} \\ \text{and } C \text{ is a clause s.t. } F \models C \text{ and } A \models \neg C \end{cases}$

Overview

Some Instantiations





Experiments

On benchmarks from A. Cimatti, A. Griggio, B. Joost Schaafsma, R. Sebastiani:

A Modular Approach to MaxSAT Modulo Theories (SAT 2013)

Solver	LIA(212)		LRA(186)		Total
	U	R	U	R	
cplex-msat	82	**90**	85	85	342
maxhs-msat	85	87	85	85	342
optimathsat-maxres	**87**	**90**	85	86	**348**
optimathsat-omt	75	72	85	85	317
z3-maxres	73	79	86	85	323
z3-wmax	69	77	**88**	**88**	322

Experiments - Scaling

On benchmarks generated from a QF-LIA SMT-LIB benchmark family

- 10%-100% random unit soft clauses
- 312 problems in %-groups

Solver	10%	25%	50%	100%	Total
cplex-msat	289	271	**203**	**4**	**767**
optimathsat-maxres	291	258	123	0	672
optimathsat-omt	240	130	0	0	370
z3-maxres	280	224	103	0	607
z3-wmax	**304**	**288**	4	0	596

Experiments - Lexicographic problems

On benchmarks from R. Sebastiani, P. Trentin:

On Optimization Modulo Theories, MaxSMT and Sorting Networks (TACAS 2017)

Solver	CTW	Time[s]	WTC	Time[s]
maxhs-msat	3699	**2401 s**	2399	**1367 s**
optimathsat-maxres	3410	13851 s	1850	10209 s
optimathsat-omt	3481	9710 s	2068	10483 s
z3-maxres	3699	4555 s	2399	2231 s
z3-wmax	3651	5566 s	2295	9513 s

Overview

Maximum Satisfiability

Hitting Sets

Implicit Hitting Set Algorithms for MaxSAT

Implicit Hitting Set Algorithms for MaxSMT

Instantiations & Experiments

Conclusion

Conclusion

- Different solvers for different problems
- Flexible formal framework to describe & reason
- Separation of Optimization, SAT solving, $\mathcal{T}$-reasoning

Implicit Hitting Set Algorithms for Maximum Satisfiability Modulo Theories

Katalin Fazekas[1] Fahiem Bacchus[2] Armin Biere[1]

[1] Johannes Kepler University Linz, Austria
[2] University of Toronto, Canada

Oxford, 14. July, 2018

9th International Joint Conference on Automated Reasoning