A Modular Approach to MaxSAT Modulo Theories

Alessandro Cimatti[1], Alberto Griggio[1],
Bastiaan Joost Schaafsma[1,2], Roberto Sebastiani[2]

[1]FBK-Irst, Trento, Italy;
[2]DISI, University of Trento, Italy

— SAT 2013, Helsinki, Finland, July 8-12, 2013 —

Outline

1. Context
2. A Modular Approach to MaxSMT
3. Experimental Evaluation
4. Conclusions & Future Work

Context: Optimization Modulo Theories (OMT)

Need for Satisfiability Modulo Theories (SMT)

SMT solvers widely used as backend engines in formal verification, and many other applications

Need for Optimization Modulo Theories (OMT)

Many SMT-encodable problems require optimal solutions wrt. some cost function:

- optimization of physical layout of circuit designs
- formal verification of parametric systems
- scheduling and temporal reasoning
- displacement of tools
- synthesis of Bayesian networks
- radio link frequency assignment
- ...

Context: Optimization Modulo Theories (OMT)

Need for Satisfiability Modulo Theories (SMT)

SMT solvers widely used as backend engines in formal verification, and many other applications

Need for Optimization Modulo Theories (OMT)

Many SMT-encodable problems require optimal solutions wrt. some cost function:

- optimization of physical layout of circuit designs
- formal verification of parametric systems
- scheduling and temporal reasoning
- displacement of tools
- synthesis of Bayesian networks
- radio link frequency assignment
- ...

Optimization Modulo Theories (OMT): Previous Work

- A general framework [7]
- OMT with cost functions in the Boolean domain
  - SMT + Pseudo-Boolean cost functions [7, 3]
  - MaxSMT [7]

  N.B.: can be encoded into each other.
- OMT with cost functions in the theory domain
  - OMT with linear real costs [8]

  N.B.: Can encode OMT with Boolean costs, not vice versa

[7]: [Nieuwenhuis & Oliveras; SAT-06]
[3]: [Cimatti et al. ; TACAS-10]
[8]: [Sebastiani & Tomasi; IJCAR-12]

Optimization Modulo Theories (OMT): Previous Work

- A general framework [7]
- OMT with cost functions in the Boolean domain
  - SMT + Pseudo-Boolean cost functions [7, 3]
  - MaxSMT [7]

  N.B.: can be encoded into each other.
- OMT with cost functions in the theory domain
  - OMT with linear real costs [8]

  N.B.: Can encode OMT with Boolean costs, not vice versa

[7]: [Nieuwenhuis & Oliveras; SAT-06]
[3]: [Cimatti et al. ; TACAS-10]
[8]: [Sebastiani & Tomasi; IJCAR-12]

MaxSAT & MaxSMT: The Problems

notation

- $\langle ... \rangle^{\mathcal{B}}$: objects in the Boolean space
- $\langle ... \rangle^{\mathcal{T}}$: objects in the Theory space
- $\langle ... \rangle^{\mathcal{B}} \stackrel{\text{def}}{=} \mathcal{T}2\mathcal{B}(\langle ... \rangle^{\mathcal{T}})$, $\langle ... \rangle^{\mathcal{T}} \stackrel{\text{def}}{=} \mathcal{B}2\mathcal{T}(\langle ... \rangle^{\mathcal{B}})$
  - $\mathcal{T}2\mathcal{B}$, $\mathcal{B}2\mathcal{T}$: Boolean abstraction & Theory refinement

[Partial Weighted] MaxSMT

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$: sets of hard and (weighted) soft clauses;

- weight $w_i > 0$ of soft clause C_i: penalty if unsatisfied

Output: a maximum-weight set alert$\psi_s^{\mathcal{T}}$ of soft clauses s.t. $\psi_s^{\mathcal{T}} \subseteq \varphi_s^{\mathcal{T}}$ and $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is satisfiable

MaxSAT & MaxSMT: The Problems

notation

- $\langle ... \rangle^{\mathcal{B}}$: objects in the Boolean space
- $\langle ... \rangle^{\mathcal{T}}$: objects in the Theory space
- $\langle ... \rangle^{\mathcal{B}} \stackrel{\text{def}}{=} \mathcal{T}2\mathcal{B}(\langle ... \rangle^{\mathcal{T}})$, $\langle ... \rangle^{\mathcal{T}} \stackrel{\text{def}}{=} \mathcal{B}2\mathcal{T}(\langle ... \rangle^{\mathcal{B}})$
  - $\mathcal{T}2\mathcal{B}$, $\mathcal{B}2\mathcal{T}$: Boolean abstraction & Theory refinement

[Partial Weighted] MaxSAT

Input: $\varphi_h^{\mathcal{B}}$,$\varphi_s^{\mathcal{B}}$: sets of hard and (weighted) soft clauses;

- weight $w_i > 0$ of soft clause C_i: penalty if unsatisfied

Output: a maximum-weight set $\psi_s^{\mathcal{B}}$ of soft clauses s.t. $\psi_s^{\mathcal{B}} \subseteq \varphi_s^{\mathcal{B}}$ and $\varphi_h^{\mathcal{B}} \cup \psi_s^{\mathcal{B}}$ is satisfiable

MaxSAT & MaxSMT: The Problems

notation

- $\langle ... \rangle^{\mathcal{B}}$: objects in the Boolean space
- $\langle ... \rangle^{\mathcal{T}}$: objects in the Theory space
- $\langle ... \rangle^{\mathcal{B}} \stackrel{\text{def}}{=} \mathcal{T}2\mathcal{B}(\langle ... \rangle^{\mathcal{T}})$, $\langle ... \rangle^{\mathcal{T}} \stackrel{\text{def}}{=} \mathcal{B}2\mathcal{T}(\langle ... \rangle^{\mathcal{B}})$
  - $\mathcal{T}2\mathcal{B}$, $\mathcal{B}2\mathcal{T}$: Boolean abstraction & Theory refinement

[Partial Weighted] MaxSMT

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$: sets of hard and (weighted) soft $\mathcal{T}$-clauses;

- weight $w_i > 0$ of soft clause C_i: penalty if unsatisfied

Output: a maximum-weight set alert$\psi_s^{\mathcal{T}}$ of soft $\mathcal{T}$-clauses s.t. $\psi_s^{\mathcal{T}} \subseteq \varphi_s^{\mathcal{T}}$ and $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is $\mathcal{T}$-satisfiable

MaxSMT: State of the Art

- Vaste bibliography on MaxSAT, plenty of tools available
- Very few work on MaxSMT:
  - a few related [7, 1, 3, 8] papers
  - few specific (Yices, Z3) or related [3, 8] implementations available

Note: involves both MaxSAT and SMT solving techniques:

- need expertise on both MaxSAT and SMT solving
- need access to both MaxSAT and SMT solvers' code

$\Longrightarrow$ sometimes hard to get (e.g., for MaxSAT experts/developers)

[7]: [Nieuwenhuis & Oliveras; SAT-2006]
[1]: [Ansotegui et al.; SARA-11]
[3]: [Cimatti et al. ; TACAS-10]
[8]: [Sebastiani & Tomasi; IJCAR-12]

MaxSMT: State of the Art

- Vaste bibliography on MaxSAT, plenty of tools available
- Very few work on MaxSMT:
  - a few related [7, 1, 3, 8] papers
  - few specific (Yices, Z3) or related [3, 8] implementations available

Note: involves both MaxSAT and SMT solving techniques:

- need expertise on both MaxSAT and SMT solving
- need access to both MaxSAT and SMT solvers' code

$\Longrightarrow$ sometimes hard to get (e.g., for MaxSAT experts/developers)

[7]: [Nieuwenhuis & Oliveras; SAT-2006]
[1]: [Ansotegui et al.; SARA-11]
[3]: [Cimatti et al. ; TACAS-10]
[8]: [Sebastiani & Tomasi; IJCAR-12]

Outline

1 Context

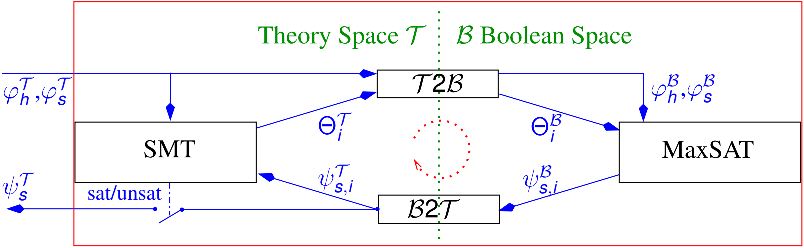
2 A Modular Approach to MaxSMT

3 Experimental Evaluation

4 Conclusions & Future Work

Lemma-Lifting (LL): a Novel Approach to MaxSMT

- Based on a cyclic interaction of a lazy SMT and a MaxSAT solver:
  - SMT produces and feeds to MaxSAT a chain of $\mathcal{T}$-lemma sets $\Theta_0^{\mathcal{T}}, \Theta_1^{\mathcal{T}}, \Theta_2^{\mathcal{T}}, ..., \Theta_N^{\mathcal{T}}$
  - MaxSAT produces and feeds to SMT a chain of soft-clause subsets $\psi_{s,0}^{\mathcal{B}}, \psi_{s,1}^{\mathcal{B}}, ..., \psi_{s,N}^{\mathcal{B}}$



Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$ // sets of hard and (weighted) soft $\mathcal{T}$-clauses
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\,(\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$; // their Boolean abstraction
$\Theta^{\mathcal{T}} \leftarrow \emptyset$; // current set of $\mathcal{T}$-lemmas
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$; // current approximation of the result
while *(* SMT.Solve $(\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}})$ = UNSAT *)* **do**
    $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\,(\Theta^{\mathcal{T}})$;
    $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT $(\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}})$; $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\,(\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Cyclic interaction of an SMT and a MaxSAT solver:

- SMT used as generator of $\mathcal{T}$-lemma sets $\Theta_0^{\mathcal{T}} \subseteq \Theta_1^{\mathcal{T}} \subseteq \Theta_2^{\mathcal{T}} \subseteq \ldots$
  $\Longrightarrow$ provides the information to rule-out $\mathcal{T}$-inconsistent solutions
- MaxSAT used to extract maximum-weight clause sets $\psi_{s,0}^{\mathcal{B}}, \psi_{s,1}^{\mathcal{B}}, \ldots$
  - $\psi_{s,i}^{\mathcal{B}}$ current approximation of the solution, $w(\psi_{s,i+1}^{\mathcal{B}}) \leq w(\psi_{s,i}^{\mathcal{B}})$

Repeated until $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is found $\mathcal{T}$-satisfiable.

Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\ (\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$;
$\Theta^{\mathcal{T}} \leftarrow \emptyset$;
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$;
while *(* SMT.Solve $(\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}})$ = UNSAT *)* **do**
    $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\ (\Theta^{\mathcal{T}})$;
    $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT $(\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}})$; $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\ (\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Cyclic interaction of an SMT and a MaxSAT solver:

- SMT used as generator of $\mathcal{T}$-lemma sets $\Theta_0^{\mathcal{T}} \subseteq \Theta_1^{\mathcal{T}} \subset \Theta_2^{\mathcal{T}} \subset \ldots$
  $\Longrightarrow$ provides the information to rule-out $\mathcal{T}$-inconsistent solutions
- MaxSAT used to extract maximum-weight clause sets $\psi_{s,0}^{\mathcal{B}}, \psi_{s,1}^{\mathcal{B}}, \ldots$
  - $\psi_{s,i}^{\mathcal{B}}$ current approximation of the solution, $w(\psi_{s,i+1}^{\mathcal{B}}) \leq w(\psi_{s,i}^{\mathcal{B}})$

Repeated until $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is found $\mathcal{T}$-satisfiable.

Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\ (\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$;
$\Theta^{\mathcal{T}} \leftarrow \emptyset$;
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$;
while *(* SMT.Solve $(\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}})$ = UNSAT *)* **do**
  $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\ (\Theta^{\mathcal{T}})$;
  $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT $(\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}})$; $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\ (\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Cyclic interaction of an SMT and a MaxSAT solver:

- SMT used as generator of $\mathcal{T}$-lemma sets $\Theta_0^{\mathcal{T}} \subseteq \Theta_1^{\mathcal{T}} \subset \Theta_2^{\mathcal{T}} \subset ...$
  $\Longrightarrow$ provides the information to rule-out $\mathcal{T}$-inconsistent solutions
- MaxSAT used to extract maximum-weight clause sets $\psi_{s,0}^{\mathcal{B}}, \psi_{s,1}^{\mathcal{B}}, \ldots$
  - $\psi_{s,i}^{\mathcal{B}}$ current approximation of the solution, $w(\psi_{s,i+1}^{\mathcal{B}}) \leq w(\psi_{s,i}^{\mathcal{B}})$

Repeated until $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is found $\mathcal{T}$-satisfiable.

Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\ (\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$;
$\Theta^{\mathcal{T}} \leftarrow \emptyset$;
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$;
while *(* SMT.Solve ($\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}}$) = UNSAT*)* **do**
  $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\ (\Theta^{\mathcal{T}})$;
  $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT ($\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$); $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\ (\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Cyclic interaction of an SMT and a MaxSAT solver:

- SMT used as generator of $\mathcal{T}$-lemma sets $\Theta_0^{\mathcal{T}} \subseteq \Theta_1^{\mathcal{T}} \subset \Theta_2^{\mathcal{T}} \subset \ldots$
  $\Longrightarrow$ provides the information to rule-out $\mathcal{T}$-inconsistent solutions
- MaxSAT used to extract maximum-weight clause sets $\psi_{s,0}^{\mathcal{B}}, \psi_{s,1}^{\mathcal{B}}, \ldots$
  - $\psi_{s,i}^{\mathcal{B}}$ current approximation of the solution, w($\psi_{s,i+1}^{\mathcal{B}}$) $\leq$ w($\psi_{s,i}^{\mathcal{B}}$)

Repeated until $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is found $\mathcal{T}$-satisfiable.

Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\,(\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$;
$\Theta^{\mathcal{T}} \leftarrow \emptyset$;
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$;
while (SMT.Solve ($\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}}$) = UNSAT) **do**
  $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\,(\Theta^{\mathcal{T}})$;
  $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT ($\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$); $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\,(\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Cyclic interaction of an SMT and a MaxSAT solver:

- SMT used as generator of $\mathcal{T}$-lemma sets $\Theta_0^{\mathcal{T}} \subseteq \Theta_1^{\mathcal{T}} \subset \Theta_2^{\mathcal{T}} \subset \ldots$
  $\Longrightarrow$ provides the information to rule-out $\mathcal{T}$-inconsistent solutions
- MaxSAT used to extract maximum-weight clause sets $\psi_{s,0}^{\mathcal{B}}, \psi_{s,1}^{\mathcal{B}}, \ldots$
  - $\psi_{s,i}^{\mathcal{B}}$ current approximation of the solution, $w(\psi_{s,i+1}^{\mathcal{B}}) \leq w(\psi_{s,i}^{\mathcal{B}})$

Repeated until $\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}}$ is found $\mathcal{T}$-satisfiable.

Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\ (\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$;
$\Theta^{\mathcal{T}} \leftarrow \emptyset$;
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$;
while *(* SMT.Solve ($\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}}$) = UNSAT *)* **do**
  $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\ (\Theta^{\mathcal{T}})$;
  $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT ($\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$); $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\ (\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Intuition:

- SMT progressively generates enough $\mathcal{T}$-lemmas in $\Theta^{\mathcal{T}}$ s.t the theory refinement $\psi_s^{\mathcal{T}}$ of $\psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \mathit{MaxSAT}(\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}})$ is $\mathcal{T}$-satisfiable.
- MaxSAT produces progressively-finer approximations of the solution, $\psi_s^{\mathcal{B}}$, whose refinement $\psi_s^{\mathcal{T}}$ drives SMT to produce the next $\mathcal{T}$-lemmas

Lemma-Lifting Approach: MaxSMT($\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$)

Input: $\varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}}$
$\langle \varphi_h^{\mathcal{B}}, \varphi_s^{\mathcal{B}} \rangle \leftarrow \mathcal{T}2\mathcal{B}\ (\langle \varphi_h^{\mathcal{T}}, \varphi_s^{\mathcal{T}} \rangle)$;
$\Theta^{\mathcal{T}} \leftarrow \emptyset$;
$\psi_s^{\mathcal{T}} \leftarrow \varphi_s^{\mathcal{T}}$;
while *(* SMT.Solve ($\varphi_h^{\mathcal{T}} \cup \psi_s^{\mathcal{T}} \cup \Theta^{\mathcal{T}}$) = UNSAT *)* **do**
  $\Theta^{\mathcal{T}} \leftarrow \Theta^{\mathcal{T}} \cup$ SMT.GetTLemmas(); $\Theta^{\mathcal{B}} \leftarrow \mathcal{T}2\mathcal{B}\ (\Theta^{\mathcal{T}})$;
  $\psi_s^{\mathcal{B}} \leftarrow$ MaxSAT ($\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$); $\psi_s^{\mathcal{T}} \leftarrow \mathcal{B}2\mathcal{T}\ (\psi_s^{\mathcal{B}})$;
return $\psi_s^{\mathcal{T}}$;

Intuition:

- SMT progressively generates enough $\mathcal{T}$-lemmas in $\Theta^{\mathcal{T}}$ s.t the theory refinement $\psi_s^{\mathcal{T}}$ of $\psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \mathit{MaxSAT}(\varphi_h^{\mathcal{B}} \cup \Theta^{\mathcal{B}}, \varphi_s^{\mathcal{B}})$ is $\mathcal{T}$-satisfiable.
- MaxSAT produces progressively-finer approximations of the solution, $\psi_s^{\mathcal{B}}$, whose refinement $\psi_s^{\mathcal{T}}$ drives SMT to produce the next $\mathcal{T}$-lemmas

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\varphi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\} \qquad \varphi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0: & ((x \le 0)) & [4] \\ C_1: & ((x \le 1)) & [3] \\ C_2: & ((x \ge 2)) & [2] \\ C_3: & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1: & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2: & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3: & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4: & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5: & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6: & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0: & ((x \leq 0)) & [4] \\ C_1: & ((x \leq 1)) & [3] \\ C_2: & ((x \geq 2)) & [2] \\ C_3: & ((x \geq 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1: & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2: & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3: & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4: & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5: & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6: & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0: & ((x \le 0)) & [4] \\ C_1: & ((x \le 1)) & [3] \\ C_2: & ((x \ge 2)) & [2] \\ C_3: & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1: & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2: & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3: & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4: & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5: & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6: & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	w($\psi_{s,i}^{\mathcal{T}}$)	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\quad, C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2, \quad\}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\quad, \quad, C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2, \ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ , \ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \le 0)) & [4] \\ C_1 : & ((x \le 1)) & [3] \\ C_2 : & ((x \ge 2)) & [2] \\ C_3 : & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2 : & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3 : & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4 : & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5 : & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6 : & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\quad, C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2, \quad\}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\quad, \quad, C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \le 0)) & [4] \\ C_1 : & ((x \le 1)) & [3] \\ C_2 : & ((x \ge 2)) & [2] \\ C_3 : & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2 : & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3 : & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4 : & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5 : & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6 : & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0: & ((x \le 0)) & [4] \\ C_1: & ((x \le 1)) & [3] \\ C_2: & ((x \ge 2)) & [2] \\ C_3: & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1: & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2: & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3: & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4: & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5: & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6: & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example I

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0: & ((x \le 0)) & [4] \\ C_1: & ((x \le 1)) & [3] \\ C_2: & ((x \ge 2)) & [2] \\ C_3: & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1: & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2: & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3: & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4: & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5: & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6: & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

An "unlucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_4\}$	$\{\ \ , C_1, C_2, C_3\}$	11	UNSAT
2	$\{\theta_4, \theta_6\}$	$\{C_0, C_1, C_2,\ \ \}$	9	UNSAT
3	$\{\theta_4, \theta_6, \theta_3\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example II

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\varphi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : ((x \le 0)) & [4] \\ C_1 : ((x \le 1)) & [3] \\ C_2 : ((x \ge 2)) & [2] \\ C_3 : ((x \ge 3)) & [6] \end{array} \right\} \qquad \varphi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{l} \theta_1 : (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2 : (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3 : (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4 : (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5 : (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6 : (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

A "lucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_1, \theta_2, \theta_5\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example II

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{llll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

A "lucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_1, \theta_2, \theta_5\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example II

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

A "lucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_1, \theta_2, \theta_5\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example II

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\}$$

$$\varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\}$$

$$\Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

A "lucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_1, \theta_2, \theta_5\}$	$\{\ ,\ , C_2, C_3\}$	8	SAT

A Toy Example II

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \leq 0)) & [4] \\ C_1 : & ((x \leq 1)) & [3] \\ C_2 : & ((x \geq 2)) & [2] \\ C_3 : & ((x \geq 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \leq 0) \vee (x \leq 1)) \\ \theta_2 : & (\neg(x \geq 3) \vee (x \geq 2)) \\ \theta_3 : & (\neg(x \leq 0) \vee \neg(x \geq 2)) \\ \theta_4 : & (\neg(x \leq 0) \vee \neg(x \geq 3)) \\ \theta_5 : & (\neg(x \leq 1) \vee \neg(x \geq 2)) \\ \theta_6 : & (\neg(x \leq 1) \vee \neg(x \geq 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

A "lucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_1, \theta_2, \theta_5\}$	$\{\ \ ,\ \ , C_2, C_3\}$	8	SAT

A Toy Example II

$$\varphi_h^{\mathcal{T}} \stackrel{\text{def}}{=} \emptyset \qquad \varphi_h^{\mathcal{B}} \stackrel{\text{def}}{=} \emptyset$$

$$\psi_s^{\mathcal{T}} \stackrel{\text{def}}{=} \left\{ \begin{array}{lll} C_0 : & ((x \le 0)) & [4] \\ C_1 : & ((x \le 1)) & [3] \\ C_2 : & ((x \ge 2)) & [2] \\ C_3 : & ((x \ge 3)) & [6] \end{array} \right\} \qquad \psi_s^{\mathcal{B}} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} (A_0) & [4] \\ (A_1) & [3] \\ (A_2) & [2] \\ (A_3) & [6] \end{array} \right\}$$

$$\Theta_*^{\mathcal{T}} = \left\{ \begin{array}{ll} \theta_1 : & (\neg(x \le 0) \vee (x \le 1)) \\ \theta_2 : & (\neg(x \ge 3) \vee (x \ge 2)) \\ \theta_3 : & (\neg(x \le 0) \vee \neg(x \ge 2)) \\ \theta_4 : & (\neg(x \le 0) \vee \neg(x \ge 3)) \\ \theta_5 : & (\neg(x \le 1) \vee \neg(x \ge 2)) \\ \theta_6 : & (\neg(x \le 1) \vee \neg(x \ge 3)) \end{array} \right\} \qquad \Theta_*^{\mathcal{B}} = \left\{ \begin{array}{l} (\neg A_0 \vee A_1) \\ (\neg A_3 \vee A_2) \\ (\neg A_0 \vee \neg A_2) \\ (\neg A_0 \vee \neg A_3) \\ (\neg A_1 \vee \neg A_2) \\ (\neg A_1 \vee \neg A_3) \end{array} \right\}$$

A "lucky" possible execution of the algorithm is:

i	$\Theta_i^{\mathcal{T}}$	$\psi_{s,i}^{\mathcal{T}}$	$w(\psi_{s,i}^{\mathcal{T}})$	$SMT(\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}})$
0	$\{\}$	$\{C_0, C_1, C_2, C_3\}$	15	UNSAT
1	$\{\theta_1, \theta_2, \theta_5\}$	$\{\ ,\ , C_2, C_3\}$	8	SAT

Some efficiency issues

- MaxSAT repeatedly invoked on incremental formulas $\varphi_h^{\mathcal{B}} \cup \Theta_i^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$
  $\Longrightarrow$ incrementality of the MaxSAT tool desirable
- SMT.Solve repeatedly invoked on "similar" formulas $\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}}$
  $\Longrightarrow$ SMT under assumptions: (a proxi variable for) each soft clause $C_j^{\mathcal{T}}$ is assumed iff $C_j^{\mathcal{T}} \in \psi_s^{\mathcal{T}}$
  $\Longrightarrow$ learned clauses can be reused from call to call
- The more $\mathcal{T}$-lemmas generated, the less cycles performed.
  $\Longrightarrow$ techniques to enlarge the $\mathcal{T}$-lemma pool generated by SMT (e.g., static learning, eager learning of $\mathcal{T}$-propagation clauses, ...)

Some efficiency issues

- MaxSAT repeatedly invoked on incremental formulas $\varphi_h^{\mathcal{B}} \cup \Theta_i^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$
  $\Longrightarrow$ incrementality of the MaxSAT tool desirable
- SMT.Solve repeatedly invoked on "similar" formulas $\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}}$
  $\Longrightarrow$ SMT under assumptions: (a proxi variable for) each soft clause $C_j^{\mathcal{T}}$ is assumed iff $C_j^{\mathcal{T}} \in \psi_s^{\mathcal{T}}$
  $\Longrightarrow$ learned clauses can be reused from call to call
- The more $\mathcal{T}$-lemmas generated, the less cycles performed.
  $\Longrightarrow$ techniques to enlarge the $\mathcal{T}$-lemma pool generated by SMT (e.g., static learning, eager learning of $\mathcal{T}$-propagation clauses, ...)

Some efficiency issues

- MaxSAT repeatedly invoked on incremental formulas $\varphi_h^{\mathcal{B}} \cup \Theta_i^{\mathcal{B}}, \varphi_s^{\mathcal{B}}$
  $\Longrightarrow$ incrementality of the MaxSAT tool desirable
- SMT.Solve repeatedly invoked on "similar" formulas $\varphi_h^{\mathcal{T}} \cup \psi_{s,i}^{\mathcal{T}} \cup \Theta_i^{\mathcal{T}}$
  $\Longrightarrow$ SMT under assumptions: (a proxi variable for) each soft clause $C_j^{\mathcal{T}}$ is assumed iff $C_j^{\mathcal{T}} \in \psi_s^{\mathcal{T}}$
  $\Longrightarrow$ learned clauses can be reused from call to call
- The more $\mathcal{T}$-lemmas generated, the less cycles performed.
  $\Longrightarrow$ techniques to enlarge the $\mathcal{T}$-lemma pool generated by SMT (e.g., static learning, eager learning of $\mathcal{T}$-propagation clauses, ...)

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required
    $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf
    $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required
    $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf
    $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required
    $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf
    $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required
    $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf
    $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required
    $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf
    $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Features

- Very simple
- Independent from the theory/theories addressed
- Independent from the SMT technique (if provides the $\mathcal{T}$-lemmas)
- Independent from the MaxSAT technique
- The MaxSAT and/or SMT solver can be used as blackboxes
  - no expertise or code access to SMT or MaxSAT solvers required
    $\Longrightarrow$ basic version simple to implement
  - choice of tools from the shelf
    $\Longrightarrow$ benefits for free from progress in the field
- ... or not
  - interleaving SMT and MaxSAT steps
  - sharing more information between SMT and MaxSAT steps
  - interleaving with other operations
  - ...

... is it also efficient?

Outline

1. Context
2. A Modular Approach to MaxSMT
3. Experimental Evaluation
4. Conclusions & Future Work

An empirical evaluation I

Benchmarks

- general lack of MaxSMT benchmark problems!

$\Longrightarrow$ crafted two groups: weighted & un-weighted, 199 formulas each
  - SMT-LIB $\mathcal{LA}(\mathbb{Q})$ and $\mathcal{LA}(\mathbb{Z})$ formulas
  - random hard/soft partitioning
  - random weights

LL Implementations

- LL implemented on top of our MATHSAT5 SMT solver [4]
- Four dinstinct MaxSAT implementations
  - LL_{WPM}, the publicly-available WPM [2] implementation (PICOSAT);
  - $LL_{YICES\text{-}WPM}$, non-public implementation of WPM (YICES);
  - LL_{OWPM}, our own (incremental) WPM implementation (MINISAT);
  - $LL_{NI\text{-}OWPM}$, as before, (non-incremental) version.

[4]: [Cimatti et al. ; TACAS-13]
[2]: [Ansotegui et al.; AIJ-13]

An empirical evaluation II

Competitors

- MathSMT implementations
  - YICES , the MaxSMT extension
  - Z3 , the MaxSMT extension (Fu&Malik core-guided algorithm [6])
  - MATHSAT5-MAX our implementation of the algorithm of [6],
- Related tools:
  - MATHSAT4+$\mathcal{C}$(L), the SMT+PB tool from [3], linear-search mode;
  - MATHSAT5+$\mathcal{C}$(L), the porting of the above procedure into MATHSAT5, linear-search mode;
  - MATHSAT5+$\mathcal{C}$(B), as before, binary-search mode;
  - OPTIMATHSAT, the OMT($\mathcal{LA}(\mathbb{Q}) \cup \mathcal{T}$) tool of [8], with adaptive binary/linear search

[6]: [Fu & Malik; SAT-06]
[3]: [Cimatti et al.; TACAS-10]
[8]: [Sebastiani & Tomasi; IJCAR-12]

Results (un-weighted MaxSMT)

Solver	$\mathcal{LA}(\mathbb{Z})$		$\mathcal{LA}(\mathbb{Q})$		Total	
	#S.	time (sec)	#S.	time (sec)	#S.	time (sec)
MATHSAT5-MAX	95	6575.60	88	2274.69	183	8850.29
LL_{OWPM}	92	5942.20	88	1785.48	180	7727.68
YICES	92	14478.43	87	5537.47	179	20015.9
$\text{LL}_{\text{NI-OWPM}}$	89	4439.98	88	1780.97	177	6220.95
$\text{LL}_{\text{YICES-WPM}}$	89	4937.91	87	1855.45	176	6793.36
LL_{WPM}	88	7154.19	88	2071.27	176	9225.46
MATHSAT5+$\mathcal{C}$(L)	84	7112.43	87	2175.34	171	9287.77
MATHSAT4+$\mathcal{C}$(L)	83	5220.14	85	1944.48	168	7164.62
Z3	89	4066.92	76	2427.59	165	6494.51
MATHSAT5+$\mathcal{C}$(B)	78	5030.85	87	2545.69	165	7576.54
OPTIMATHSAT	—	—	89	1360.05	—	—
TOTAL #:	106		93		199	

Results (weighted MaxSMT)

Solver	$\mathcal{LA}(\mathbb{Z})$		$\mathcal{LA}(\mathbb{Q})$		Total	
	#S.	time (sec)	#S.	time (s)	#S.	time (s)
LL_{WPM}	90	5194.73	87	3033.66	177	8228.39
$LL_{NI-OWPM}$	86	1672.41	88	2062.35	174	3734.76
MATHSAT5+$\mathcal{C}$(L)	89	5501.38	84	2359.61	173	7860.99
LL_{OWPM}	85	1304.13	87	1836.53	172	3140.66
MATHSAT4+$\mathcal{C}$(L)	87	3105.01	85	2541.83	172	5646.84
$LL_{YICES-WPM}$	82	1423.53	87	2350.02	169	3773.55
YICES	83	12305.88	80	9804.16	163	22110.04
MATHSAT5+$\mathcal{C}$(B)	79	9482.61	83	2627.35	162	12109.96
OPTIMATHSAT	—	—	88	1947.06	—	—
TOTAL #:	106		93		199	

Outline

1. Context
2. A Modular Approach to MaxSMT
3. Experimental Evaluation
4. Conclusions & Future Work

Conclusions

A novel "Modular" approach for MaxSMT

- combine an SMT and MaxSAT solver
  - easy to implement
  - independent on theory, SMT technique and MaxSAT technique
  - allows to use s.o.a. tools from the shelf
- performance comparable with s.o.a.

Ongoing & Future Research Directions

Many possible improvements

- specialized SMT and MaxSAT solvers
- interleaving SMT and MaxSAT steps
- sharing more information between SMT and MaxSAT steps
- enrich with SMT unsat-core extraction [5], PB-cost constraints [3]
- ...

More extensive empirical evaluation

- collecting real-word problems
- Do you have MaxSMT/SMT+cost problems? Please send us!!

Ongoing & Future Research Directions

Many possible improvements

- specialized SMT and MaxSAT solvers
- interleaving SMT and MaxSAT steps
- sharing more information between SMT and MaxSAT steps
- enrich with SMT unsat-core extraction [5], PB-cost constraints [3]
- ...

More extensive empirical evaluation

- collecting real-word problems
- Do you have MaxSMT/SMT+cost problems? Please send us!!


"That's all Folks!"

Questions?

References I

[1] C. Ansótegui, M. Bofill, M. Palahí, J. Suy, and M. Villaret.
Satisfiability Modulo Theories: An Efficient Approach for the Resource-Constrained Project Scheduling Problem.
In *SARA*, 2011.

[2] C. Ansótegui, M. L. Bonet, and J. Levy.
Sat-based maxsat algorithms.
Artif. Intell., 196:77–105, 2013.

[3] A. Cimatti, A. Franzén, A. Griggio, R. Sebastiani, and C. Stenico.
Satisfiability modulo the theory of costs: Foundations and applications.
In *TACAS*, volume 6015 of *LNCS*, pages 99–113. Springer, 2010.

[4] A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani.
The MathSAT 5 SMT Solver.
In *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'13.*, volume 7795 of *LNCS*, pages 95–109. Springer, 2013.

[5] A. Cimatti, A. Griggio, and R. Sebastiani.
Computing Small Unsatisfiable Cores in SAT Modulo Theories.
Journal of Artificial Intelligence Research, JAIR, 40:701–728, April 2011.

[6] Z. Fu and S. Malik.
On solving the partial max-sat problem.
In *In International Conference on Theory and Applications of Satisfiability Testing (SAT), LNCS 4121*, pages 252–265. Springer, 2006.

[7] R. Nieuwenhuis and A. Oliveras.
On SAT Modulo Theories and Optimization Problems.
In *Proc. Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121 of *LNCS*. Springer, 2006.

References II

[8] R. Sebastiani and S. Tomasi.
Optimization in SMT with LA(Q) Cost Functions.
In *IJCAR*, volume 7364 of *LNAI*, pages 484–498. Springer, July 2012.