universität innsbruck

SAT and SMT Solving

Sarah Winkler

Computational Logic Group
Department of Computer Science
University of Innsbruck

lecture 12
SS 2019

Outline

Definitions

- theory consists of
  - signature Σ: set of function and predicate symbols
  - axioms T: set of sentences in first-order logic in which only function and predicate symbols of Σ appear
- theory is stably infinite if every satisfiable quantifier-free formula has model with infinite carrier set
- theory T is convex if $F \vDash_T \bigvee_{i=1}^{n} u_i = v_i$ implies $F \vDash_T u_i = v_i$ for some $1 \leqslant i \leqslant n$ $\forall$ quantifier-free conjunction F and variables u_i, v_i

Definition

theory combination $T_1 \oplus T_2$ of two theories

- T_1 over signature Σ_1
- T_2 over signature Σ_2

has signature $\Sigma_1 \cup \Sigma_2$ and axioms $T_1 \cup T_2$

Assumptions

Nelson-Oppen Method: Nondeterministic Version

Input quantifier-free conjunction φ in theory combination $T_1 \oplus T_2$
Output satisfiable or unsatisfiable

1. purification

$$\varphi \approx \varphi_1 \wedge \varphi_2 \quad \text{for } \Sigma_1\text{-formula } \varphi_1 \text{ and } \Sigma_2\text{-formula } \varphi_2$$

2. guess and check
   - V is set of shared variables in φ_1 and φ_2
   - guess equivalence relation E on V
   - arrangement $\alpha(V, E)$ is formula

$$\bigwedge_{x\,E\,y} x = y \quad \wedge \bigwedge_{\neg(x\,E\,y)} x \neq y$$

   - if $\varphi_1 \wedge \alpha(V, E)$ is T_1-satisfiable and $\varphi_2 \wedge \alpha(V, E)$ is T_2-satisfiable then return satisfiable else return unsatisfiable

Nelson-Oppen Method: Deterministic Version

Input: quantifier-free conjunction φ in combination $T_1 \oplus T_2$ of convex theories T_1 and T_2

Output: satisfiable or unsatisfiable

1. purification $\varphi \approx \varphi_1 \wedge \varphi_2$ for Σ_1-formula φ_1 and Σ_2-formula φ_2
2. V: set of shared variables in φ_1 and φ_2
   E: already discovered equalities between variables in V
3. test satisfiability of $\varphi_1 \wedge E$ (and add implied equations)
   - if $\varphi_1 \wedge E$ is T_1-unsatisfiable then return unsatisfiable
   - else add new implied equalities to E
4. test satisfiability of $\varphi_2 \wedge E$ (and add implied equations)
   - if $\varphi_2 \wedge E$ is T_2-unsatisfiable then return unsatisfiable
   - else add new implied equalities to E
5. if E has been extended in steps 3 or 4 then go to step 2
   else return satisfiable

Outline

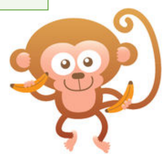
- Summary of Last Week
- Quantifiers for SMT
  - Skolemization
- Instantiation Techniques

Applications of Quantifiers in SMT

Example (Homework 5)

Imagine a village of monkeys where each monkey owns at least two bananas. As the monkeys are well-organised, each tree contains exactly three monkeys. Monkeys are also very friendly, so every monkey has a partner.

quantifiers!

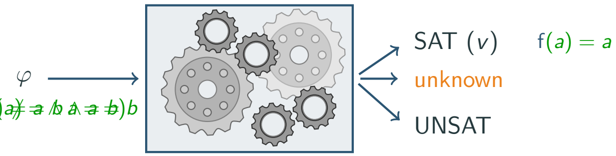


More important applications

- automated theorem proving
  $\forall x\, y\, z.\ \mathrm{inv}(x) \cdot x = 0 \wedge 0 \cdot x = x \wedge x \cdot (y \cdot z) = (x \cdot y) \cdot z$
- software verification
  $\forall x.\ \mathrm{pre}(x) \longrightarrow \mathrm{post}(x)$
- function synthesis
  $\forall input.\ \exists output.\ \mathrm{F}(input, output)$
- planning
  $\exists plan.\ \forall time.\ \mathrm{spec}(plan, time)$

SMT Solving with Quantifiers

SMT solver



Decision Properties

first-order logic is undecidable!

- SMT solvers can decide propositional logic + LIA/LRA/EUF/BV/...
- many SMT solvers also have support for quantifiers, but have in general no decision procedure for theories + quantifiers

Skolemization

name witness for existential quantifier

Getting rid of $\exists$ quantifiers

- replace $\exists x.\ \mathsf{P}(x)$ by $\mathsf{P}(\mathsf{a})$
- replace $\forall y\ \exists x.\ \mathsf{P}(x)$ by $\forall y\ \mathsf{P}(\mathsf{f}(y))$
- replace $\forall z\ \forall y\ \exists x.\ \mathsf{R}(x)$ by $\forall z\ \forall y\ \mathsf{R}(\mathsf{f}(y,z))$

Thoralf Skolem

Definitions

- φ is in prenex form if $\varphi = Q_1 x_1 \dots Q_n x_n\ \psi$ for ψ quantifier-free and $Q_i \in \{\forall, \exists\}$
- φ is in Skolem form if in prenex form without existential quantifier

Skolemization

1. bring formula into prenex form
2. replace $\forall x_1, \dots, x_k \exists y\ \psi[y]$ by $\forall x_1, \dots, x_k\ \psi[\mathsf{f}(x_1, \dots, x_k)]$ for fresh f

   until no existential quantifiers left

can consider formulas of shape $\forall x_1, \dots, x_n\ \varphi[x_1, \dots, x_n]$

Theorem

if φ' is skolemization of φ then φ and φ' are equisatisfiable

Definition

set of function symbols and constants

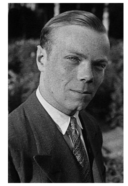
Herbrand instance of Skolem formula $\forall x_1, \dots, x_n\ \varphi[x_1, \dots, x_n]$ is $\varphi[t_1, \dots, t_n]$ where t_i is term over signature of φ

Remark

Herbrand instances are ground formulas, i.e., without (quantified) variables

Theorem (Herbrand)

Skolem formula φ is unsatisfiable $\Longleftrightarrow$
there exists finite unsatisfiable set of Herbrand instances of φ



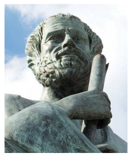
Jacques Herbrand

Caveats

- at least one constant required per sort
- holds for pure first order logic, not necessarily in presence of theories

Example: Is this syllogism correct?

All humans are mortal.	$\forall x.\ H(x) \longrightarrow M(x)$
All Greeks are humans.	$\forall x.\ G(x) \longrightarrow H(x)$
So all Greeks are mortal.	$\forall x.\ G(x) \longrightarrow M(x)$



Aristotle

- translate to first-order logic
- check validity of

cannot be answered by SMT solver

$$((\forall x.\ H(x) \longrightarrow M(x)) \wedge (\forall x.\ G(x) \longrightarrow H(x))) \longrightarrow (\forall x.\ G(x) \longrightarrow M(x))$$

- check unsatisfiability of

$$\forall x.\ H(x) \longrightarrow M(x), \quad \forall x.\ G(x) \longrightarrow H(x), \quad \exists x.\ G(x) \wedge \neg M(x)$$

- skolemize

$$\forall x.\ H(x) \longrightarrow M(x),$$

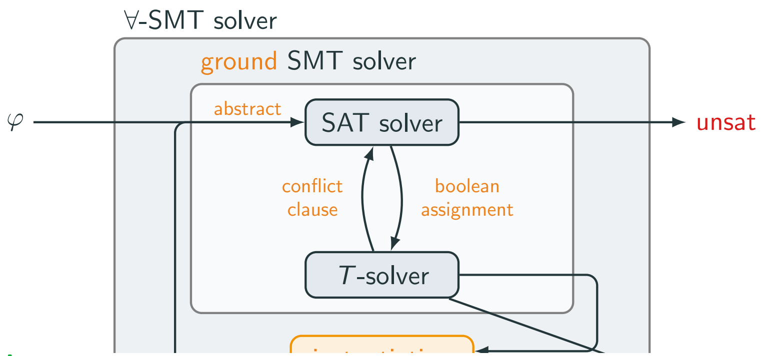
when adding right Herbrand instances
unsatisfiability can be detected by SMT solver

- already unsatisfiable when replacing quantified formulas by Herbrand instances

$$H(a) \longrightarrow M(a), \quad G(a) \longrightarrow H(a), \quad G(a) \wedge \neg M(a)$$

Outline

- Summary of Last Week
- Quantifiers for SMT
- Instantiation Techniques
  - E-Matching
  - Enumerative Instantiation



Example

- $a = b \wedge \mathrm{g}(a) = a \wedge (\mathrm{f}(a) \neq \mathrm{f}(b) \vee b \neq \mathrm{g}(\mathrm{g}(a)))$
- abstract to $p_{a=b} \wedge p_{\mathrm{g}(a)=a} \wedge (p_{\mathrm{f}(a)\neq \mathrm{f}(b)} \vee p_{b\neq \mathrm{g}(\mathrm{g}(a))})$
- SAT solver: $p_{a=b}, p_{\mathrm{g}(a)=a}, p_{\mathrm{f}(a)\neq \mathrm{f}(b)}$ T-solver: $\neg p_{a=b} \vee \neg p_{\mathrm{f}(a)\neq \mathrm{f}(b)}$
- SAT solver: $p_{a=b}, p_{\mathrm{g}(a)=a}, p_{b\neq \mathrm{g}(\mathrm{g}(a))}$ T-solver: $\neg p_{a=b} \vee \neg p_{\mathrm{g}(a)=a} \vee \neg p_{b\neq \mathrm{g}(\mathrm{g}(a))}$
- SAT solver: unsat

Instantiation

Definition (Instance)

$$(\forall \overline{x}\ \varphi(\overline{x})) \longrightarrow \varphi\sigma$$

is instance where $\overline{x}\sigma$ does not contain variables $\overline{x}$

Example

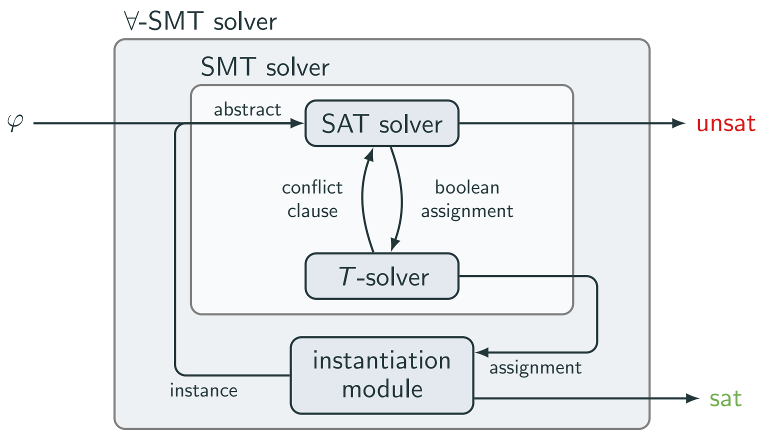
$\forall x.\ H(x) \longrightarrow M(x)$ has instance $(\forall x.\ H(x) \longrightarrow M(x)) \longrightarrow (H(a) \longrightarrow M(a))$

Remarks

- as first-order logic formula, every instance is tautology
- in SAT solver, $\forall \overline{x}\ \varphi(\overline{x})$ gets abstracted to propositional variable $p_{\forall \overline{x}\ \varphi(\overline{x})}$, which has meaning only for instantiation module
- $\varphi\sigma$ gets abstracted to propositional formula: involved variables have meaning for theory solver
- idea: $\varphi\sigma$ gets "activated" if propositional variable $p_{\forall \overline{x}\ \varphi(\overline{x})}$ is assigned true

Example

Instantiation Framework



- split φ into
  - literals φ_Q with quantifiers
  - literals φ_E without quantifiers
- instantiation module generates instances of φ_Q to extend φ_E

E-Matching

Example

$\varphi_E : \neg P(a),\ \neg P(b),\ \neg R(b)$

$\varphi_Q : \forall x.\ P(x) \vee R(x)$

trigger

- assume literal $P(x)$ is instantiation pattern
- find substitutions σ such that $P(x)\sigma$ occurs in φ_E matching
- obtain $\{x \mapsto a\}$, $\{x \mapsto b\}$
- add $P(a) \vee R(a)$ and $P(b) \vee R(b)$ to φ_E

Instantiation via E-matching

for each $\forall \overline{x}.\psi$

- select set of instantiation patterns $\{t_1, \dots, t_n\}$
- for each t_i let S_i be set of substitutions σ such that $t_i\sigma$ occurs in φ_E
- add $\{\psi\sigma \mid \sigma \in S_i\}$ to φ_E

Example

$\forall x \forall y.\ \mathsf{sibling}(x, y) \longleftrightarrow \mathsf{mother}(x) = \mathsf{mother}(y) \land \mathsf{father}(x) = \mathsf{father}(y)$

$\mathsf{sibling}(\mathsf{adam}, \mathsf{bea})$

$\mathsf{sibling}(\mathsf{bea}, \mathsf{chris})$

$\neg\mathsf{sibling}(\mathsf{adam}, \mathsf{chris})$

- unsatisfiable
- suitable instantiation patterns?
  $\mathsf{sibling}(x, y)$ sufficient

Remarks

- works as decision procedure for some theories (e.g., lists and arrays) but can easily omit necessary instances in other cases
- mostly efficient
- requires instantiation patterns (manually or heuristically determined)
- instantiation patterns can be specified in SMT-LIB

Outline

- Summary of Last Week
- Quantifiers for SMT
- Instantiation Techniques
  - E-Matching
  - Enumerative Instantiation

Enumerative Instantiation

Why not use Herbrand's theorem directly?

Theorem (Herbrand)

Skolem formula φ is unsatisfiable $\Longleftrightarrow$
there exists finite unsatisfiable set of Herbrand instances of φ

Early days of theorem proving

- first theorem provers enumerated Herbrand instances, looked for refutation
- infeasible in practice
- approach was forgotten

Enumerative instantiation

- instantiation module based on stronger version of Herbrand's theorem
- efficient implementation techniques

Theorem (Stronger Herbrand)

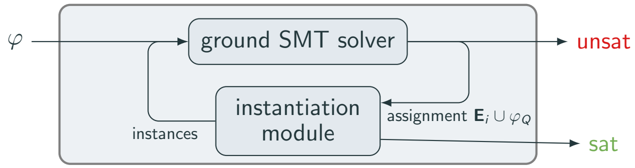
$\varphi_E \land \varphi_Q$ is unsatisfiable if and only if there exist infinite series

- *$\mathbf{E}_i$ of finite literals sets*
- *$\mathbf{Q}_i$ of finite sets of φ_Q instances*

such that

- $\mathbf{Q}_i \subseteq \{\psi\sigma \mid \forall \overline{x}.\ \psi$ *occurs in* φ_Q *and* $\mathrm{dom}(\sigma) = \overline{x}$ *and* $\mathrm{ran}(\sigma) \subseteq \mathcal{T}(\mathbf{E}_i)\}$
- $\mathbf{E}_0 = \varphi_E$ *and* $\mathbf{E}_{i+1} = \mathbf{E}_i \cup \mathbf{Q}_i$
- *some* $\mathbf{E}_n$ *is unsatisfiable*

Direct application in $\forall$-SMT solver



- ground solver enumerates assignments $\mathbf{E}_i \cup \varphi_Q$
- instantiation returns $\forall \overline{x}\ \psi(\overline{x}) \longrightarrow Q$ for all $Q \in \mathbf{Q}_i$ generated from $\forall \overline{x}\ \psi(\overline{x})$

Lemma

Instantiation via enumeration

Fix ordering $>$ on tuples of terms without quantified variables.
Given assignment $\mathbf{E}_i$ from T-solver

- for each $\forall \overline{x}.\psi$ in φ_Q
  - search minimal $\overline{x}\sigma$ with respect to $\succeq$ such that $\overline{x}\sigma \in \mathcal{T}(\mathbf{E}_i)$ and $\mathbf{E}_i \not\models \psi\sigma$
  - if exists, add $\{\psi\sigma\}$ to $\mathbf{Q}_i$

If $\mathbf{Q}_i = \varnothing$ then sat, otherwise return $\mathbf{Q}_i$

Example

$$\varphi_E: P(a) \vee a = b,\ \neg P(b),\ \neg P(g(b))$$
$$\varphi_Q: \forall x.\ P(x) \vee P(f(x)),\ \forall x.\ g(x) = f(x)$$

- suppose order $a < b < f(a) < f(b) < \dots$
- ground solver: model $P(a),\ \neg P(b),\ \neg P(g(b)$ (and φ_Q)
- instantiation: $\mathbf{Q}_1$ consists of $P(b) \vee P(f(b))$ and $f(a) = g(a)$
- ground solver: model $P(a),\ \neg P(b),\ \neg P(g(b),\ f(a) = g(a),\ P(f(b))$ (and φ_Q)
- instantiation: $\mathbf{Q}_2$ consists of $P(f(a)) \vee P(f(f(a)))$ and $f(b) = g(b)$
- ground solver: unsat

Bibliography

David Detlefs, Greg Nelson, and James B. Saxe.
Simplify: A Theorem Prover for Program Checking.
J. ACM, 52(3):365-473, 2005.

Andrew Reynolds, Haniel Barbosa and Pascal Fontaine.
Revisiting Enumerative Instantiation.
Proc. TACAS, pp 112–131, 2018.

Slide material partially taken from Pascal Fontaine's talk at SMT Summer School 2018.