Combination of Theories

IA085: Satisfiability and Automated Reasoning

Martin Jonáš

FI MUNI, Spring 2024

Last time

- theory solvers for UF and difference logic
- sketches of ideas of other theory solvers

Assumptions

- all formulas are quantifier-free (necessary)
- all formulas are conjunctions of literals (not necessary)

Combination of theories

Motivation

Practical applications combine several theories

$$x = y + 2 \land \Big(f(x-1) \neq f(y+1) \;\lor\; \texttt{read}(a, x) = \texttt{read}(a, y) \Big)$$

- formula over T_{LIA}, T_{UF}, and T_{A}

Motivation

Practical applications combine several theories

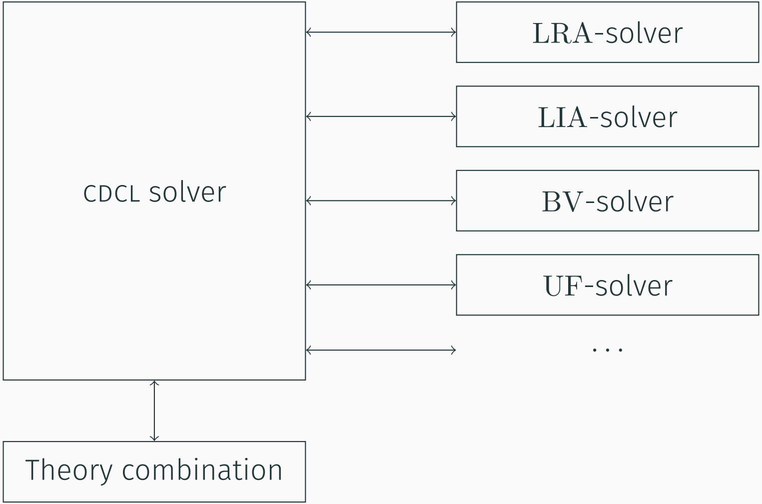
$$x = y + 2 \land \Big(f(x-1) \neq f(y+1) \;\lor\; \texttt{read}(a,x) = \texttt{read}(a,y)\Big)$$

- formula over T_{LIA}, T_{UF}, and T_{A}
- it is impractical to create a new T solver for each combination of theories

Goal

- construct a solver for the combined theory modularly from existing T-solvers for the individual theories

Goal


CDCL solver
LRA-solver
LIA-solver
BV-solver
UF-solver
...
Theory combination

Combined theory

Setup

- T_1 over signature Σ_1
- T_2 over signature Σ_2
- are signature disjoint, i.e., $\Sigma_1 \cap \Sigma_2 = \{=\}$

Want to define combined theory $T_1 \oplus T_2$ over $\Sigma = \Sigma_1 \cup \Sigma_2$.

Combined theory: axiom-based view

If

- T_1 is given by axioms A_1 and
- T_2 is given by axioms A_2

then

- $T_1 \oplus T_2$ is given by axioms $A_1 \cup A_2$

Combined theory: model-based view

Let

- $\mathcal{A} = (A, (\_)^{\mathcal{A}})$ be Σ_A-structure
- $\Sigma_B \subseteq \Sigma_A$

Σ_B **reduct** of $\mathcal{A}$

- Σ_B structure $\mathcal{B} = (A, (\_)^{\mathcal{B}})$ where
- $f^{\mathcal{B}} = f^{\mathcal{A}}$ for all $f \in \Sigma_B^f$
- $P^{\mathcal{B}} = P^{\mathcal{A}}$ for all $P \in \Sigma_B^p$
- also denoted $\mathcal{A}|_{\Sigma_B}$

Example
$\{+, \leq\}$-reduct of $(\mathbb{Z}, +, \times, \leq)$ is $(\mathbb{Z}, +, \leq)$

Combined theory: model-based view

Σ-structures $\mathcal{A}$ and $\mathcal{B}$ are elementarily equivalent if they satisfy exactly the same Σ-formulas.

Example

- all isomorphic Σ-structures are elementarily equivalent
- $(\mathbb{Q}, \leq)$ and $(\mathbb{R}, \leq)$ are elementarily equivalent
- $(\mathbb{Z}, \leq)$ and $(\mathbb{Q}, \leq)$ are not elementarily equivalent

Combined theory: model-based view

$$
\begin{aligned}
T_1 \oplus T_2 \quad = \quad \{A \mid\ & A \text{ is a } \Sigma\text{-structure}, \\
& \Sigma_1\text{-reduct of } A \text{ is elementarily equivalent to some } A_1 \in T_1, \text{ and} \\
& \Sigma_2\text{-reduct of } A \text{ is elementarily equivalent to some } A_2 \in T_2\}
\end{aligned}
$$

Nelson-Oppen method (1979)

Convention

Without loss of generality, we will consider only combination of two theories T_1 and T_2.

Necessary terminology

- i-term = its topmost function symbol is in Σ_i
- i-atom = its predicate function symbol is in Σ_i or is of form $s = t$ with i-terms s and t
- i-literal = i-atom or its negation
- i-pure subformula or subterm = all its function and predicate symbols are from Σ_i
- pure subformula or subterm = i-pure for some i
- alien subterm = maximal i-subterm of an atom that is not i-atom

Terminology: examples

$$f(x+1) = f(42) \ \wedge \ f(g(x)) = x + y \ \wedge \ y \geq x + 10$$

What are

- LIA-terms, UF-terms?
- LIA-literals, UF-literals?
- pure LIA subterms/subformulas, pure UF subterms/subformulas?
- alien subterms?

Purification

- given φ, compute φ_1 and φ_2 such that
  - φ_1 is 1-pure,
  - φ_2 is 2-pure, and
  - $\varphi_1 \wedge \varphi_2$ is equisatisfiable with φ

Purification

- given φ, compute φ_1 and φ_2 such that
  - φ_1 is 1-pure,
  - φ_2 is 2-pure, and
  - $\varphi_1 \wedge \varphi_2$ is equisatisfiable with φ

Algorithm

1. while φ contains any alien subterm t, create a new variable x_t, replace all occurrences of t by x_t and add a new equality $x_t = t$
2. now all literals are pure
3. set φ_i to all i-pure literals

Example

$$(f(x+1) = f(42)) \ \wedge \ (f(g(x)) = x + y) \ \wedge \ (y \geq x + 10)$$

Example

$$(f(x+1) = f(42)) \;\wedge\; (f(g(x)) = x + y) \;\wedge\; (y \geq x + 10)$$

$$(f(z) = f(v)) \;\wedge\; (w = x + y) \;\wedge\; (y \geq x + 10) \;\wedge\; (z = x + 1) \;\wedge\; (v = 42) \;\wedge\; (w = f(g(x)))$$

Example

$$(f(x+1) = f(42)) \ \wedge \ (f(g(x)) = x + y) \ \wedge \ (y \geq x + 10)$$

$$(f(z) = f(v)) \ \wedge \ (w = x + y) \ \wedge \ (y \geq x + 10) \ \wedge \ (z = x + 1) \ \wedge \ (v = 42) \ \wedge \ (w = f(g(x)))$$

$$\begin{aligned} \varphi_1 &= f(z) = f(x) \ \wedge \ w = f(g(x)) \\ \varphi_2 &= v = x + y \ \wedge \ y \geq x + 10 \ \wedge \ w = x + 1 \ \wedge \ v = 42 \end{aligned}$$

Setup

From now on

- $\varphi = \varphi_1 \wedge \varphi_2$
- φ_1 is 1-pure
- φ_2 is 2-pure

Non-theorem 1

Theorem (Attempt 1)
The following are equivalent

1. φ *is* $(T_1 \oplus T_2)$*-satisfiable*
2. φ_1 *is* T_1*-satisfiable and* φ_2 *is* T_2*-satisfiable.*

Does not hold

Non-theorem 1

Theorem (Attempt 1)
The following are equivalent

1. φ *is* $(T_1 \oplus T_2)$*-satisfiable*
2. φ_1 *is* T_1*-satisfiable and* φ_2 *is* T_2*-satisfiable.*

Does not hold

- $\varphi_1 = (z = x + y) \land (v = y + x)$ is satisfiable in LRA
- $\varphi_2 = f(z) \neq f(v)$ is satisfiable in UF
- $\varphi = \varphi_1 \land \varphi_2$ is not satisfiable in $\mathrm{LRA} \oplus \mathrm{UF}$

Interface equalities

Observation

- the T_1-model and the T_2-model have to agree on interface equalities = equalities between variables that are shared by φ_1 and φ_2

Arrangement

- an equivalence R over a finite set of terms S
- induces a formula

$$ar_R(S) = \bigwedge_{(s,t)\in R} (s = t) \wedge \bigwedge_{(s,t)\notin R} (s \neq t)$$

- we will consider arrangements over shared variables $C = \mathit{Vars}(\varphi_1) \cap \mathit{Vars}(\varphi_2)$.

Non-theorem 2

Theorem (Attempt 2)
The following are equivalent

1. φ *is* $(T_1 \oplus T_2)$*-satisfiable,*
2. *there is an arrangement* R *of variables* C *such that* $\varphi_1 \wedge ar_R(C)$ *is* T_1*-satisfiable and* $\varphi_2 \wedge ar_R(C)$ *is* T_2*-satisfiable.*

Does not hold

- $x = 42 \wedge y = 42 \wedge x = y$ is satisfiable in LIA
- $x = y$ is satisfiable in BV_8 (bit-vectors of width 8)
- $x = 42 \wedge y = 42 \wedge x = y$ is not satisfiable in $\mathrm{LIA} \oplus \mathrm{BV}_8$ (why?)
- $\mathrm{LIA} \oplus \mathrm{BV}_8$ is empty

Stably infinite theories

Stably infinite theory T

- if φ has a T-model, then it has a T-model with infinite universe

Example

- LIA, LRA, NIA, NRA are stably infinite
- UF is stably infinite
- bit-vectors of fixed width are not stably infinite
- strings of bounded length are not stably infinite

Main theorem

Theorem (Nelson-Oppen, 1980)
If T_1 and T_2 are stably infinite, then the following are equivalent

1. *φ is $(T_1 \oplus T_2)$-satisfiable,*
2. *there is an arrangement R of variables C such that $\varphi_1 \wedge ar_R(C)$ is T_1-satisfiable and $\varphi_2 \wedge ar_R(C)$ is T_2-satisfiable.*

Main theorem

Theorem (Nelson-Oppen, 1980)
If T_1 and T_2 are stably infinite, then the following are equivalent

1. *φ is $(T_1 \oplus T_2)$-satisfiable,*
2. *there is an arrangement R of variables C such that $\varphi_1 \wedge ar_R(C)$ is T_1-satisfiable and $\varphi_2 \wedge ar_R(C)$ is T_2-satisfiable.*

Proof (sketch).

- $1 \Rightarrow 2$: straighforward.
- $2 \Rightarrow 1$: get two models $\mathcal{A}_1$ and $\mathcal{A}_2$; use stable infiniteness and upward Löwenheim-Skolem theorem to get elementarily equivalent models $\mathcal{A}_1'$ and $\mathcal{A}_2'$ that have the same cardinality; combine the structures $\square$

Non-deterministic Nelson-Oppen algorithm

Algorithm

1. Purify the input formula into $\varphi_1 \wedge \varphi_2$
2. Non-deterministically guess an arrangement R of shared variables C.
3. Check T_1-satisfiability of $\varphi_1 \wedge ar_R(C)$ and T_2 satisfiability of $\varphi_2 \wedge ar_R(C)$.
4. If both are satisfiable, return satisfiable.
5. Otherwise return unsatisfiable.

Example

$$\begin{aligned}\varphi_{\mathrm{LIA}} &= (v_1 \geq 0) \ \wedge \ (v_1 \leq 1) \ \wedge \ (v_2 = 0) \ \wedge \ (v_3 = 1) \\ \varphi_{\mathrm{UF}} &= (f(v_1) \neq f(v_2))\end{aligned}$$

Non-deterministic Nelson-Oppen algorithm

Problems

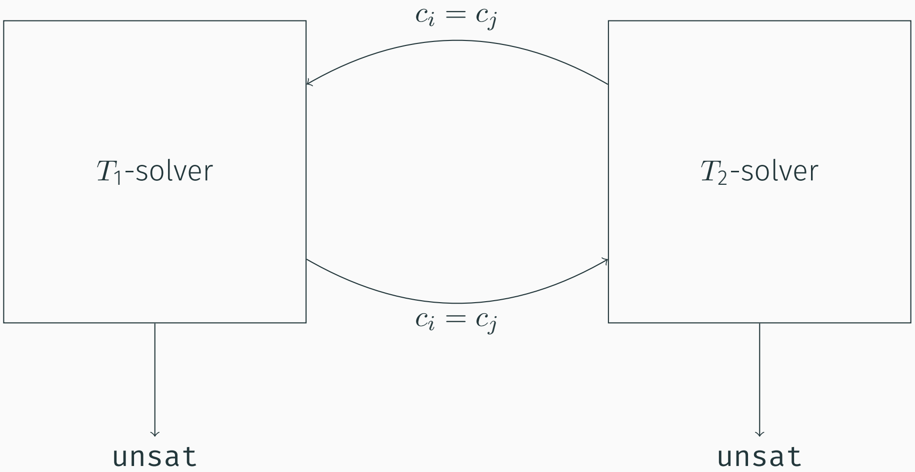
- non-deterministic algorithms are not practical
- if made deterministic, the number of arrangements is exponential with respect to $|C|$ (Bell number)

Deterministic Nelson-Oppen algorithm

Idea

- do not guess the arrangement, let the T-solvers build it together
- T_1-solver propagates all implied interface equalities (equalities between of shared variables) to T_2-solver
- T_2-solver propagates all implied interface equalities to T_1-solver
- if both solvers T_i decide that the formulas φ_i are satisfiable and do not imply any new equalities, the formula φ is satisfiable

Deterministic Nelson-Oppen algorithm



Deterministic Nelson-Oppen algorithm

Problem

- Does not work in general

Deterministic Nelson-Oppen algorithm

Problem

- Does not work in general

$$
\begin{aligned}
\varphi_{\mathrm{LIA}} &= (v_1 \geq 0) \ \wedge \ (v_1 \leq 1) \ \wedge \ (v_2 = 0) \ \wedge \ (v_3 = 1) \\
\varphi_{\mathrm{UF}} &= (f(v_1) \neq f(v_2)) \ \wedge \ (f(v_1) \neq f(v_3))
\end{aligned}
$$

Convexity

Convex theory

- if $\varphi \models_T \psi \vee \rho$
- then $\varphi \models_T \psi$ or $\varphi \models_T \rho$

Example

- UF, LRA are convex
- LIA is not convex:
  - $x \geq 1 \wedge x \leq 2 \models_{\mathrm{LIA}} x = 1 \vee x = 2$;
  - $x \geq 1 \wedge x \leq 2 \not\models_{\mathrm{LIA}} x = 1$
  - $x \geq 1 \wedge x \leq 2 \not\models_{\mathrm{LIA}} x = 2$

Algorithm

1. Purify the input formula into $\varphi_1 \wedge \varphi_2$
2. For both $i \in \{1, 2\}$
   2.1 Check satisfiability of φ_i by T_i
   2.2 Detect all equalities of variables C implied by φ_i
   2.3 Propagate them to the T_j solver $(i \neq j)$
3. If any of the T_i-solvers returned unsat, return unsat.
4. If no more equalities are propagated, return sat.
5. Go to 2.

Sound and complete for stably infinite and convex theories.

Problems

- deterministic Nelson-Oppen algorithm works only for convex theories
- complete propagation of equalities can be expensive and complicate the T-solver

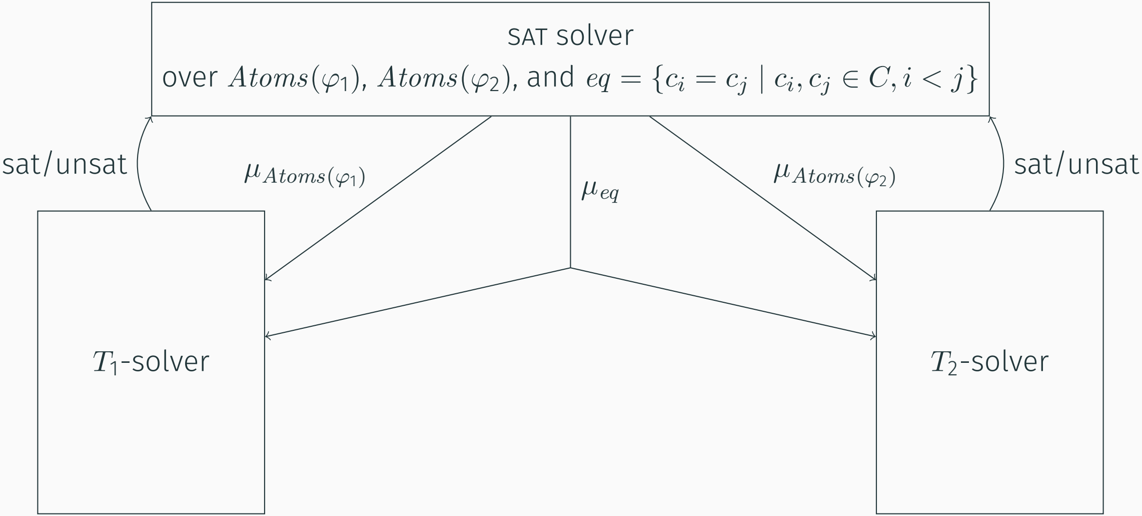
Delayed theory combination (2005)

Delayed theory combination

Idea

- combine CDCL(T) and non-determinisic Nelson-Oppen algorithm
- use a SAT solver to “guess” the interface equalities and send them to the T_i-solvers
- T_i-unsatisfiability causes backtracking in the SAT solver $\rightarrow$ try another arrangement

Delayed theory Combination



Delayed theory combination

Benefits

- fits well into CDCL(T) paradigm
- the T_i solvers can additionally perform theory propagation (but do not have to)

Example

$$
\begin{aligned}
\varphi_{\mathrm{LIA}} &= (v_1 \geq 0) \ \wedge \ (v_1 \leq 1) \ \wedge \ (v_2 = 0) \ \wedge \ (v_3 = 1) \\
\varphi_{\mathrm{UF}} &= (f(v_1) \neq f(v_2))
\end{aligned}
$$

Model-based theory combination (2007)

Problem

- delayed theory combination adds $\mathcal{O}(|C|^2)$ atoms to the SAT solver
- can slowdown the search

Solution

- be lazy

Model-based theory combination

Algorithm

1. Start DTC with empty set of interface equalities.
2. If unsatisfiable, return unsatisfiable.
3. If satisfiable, check whether the obtained models A_1 and A_2 agree on all interface equalities.
    3.1 if they do, return satisfiable
    3.2 if they do not and all the interface equalities have been added, return unsatisfiable
    3.3 otherwise add the equalities in which they differ to the DTC solver and repeat

Summary of requirements

Non-deterministic Nelson-Oppen

- stably infinite theories

Deterministic Nelson-Oppen

- stably infinite theories
- T_i solvers that can deduce all implied equalities
- convex theories (or deduce also all disjunctions of equalities)

Delayed Theory Combination (and Model-based TC)

- stably infinite theories

Where are we?

Contents

Propositional satisfiability (SAT)

- $(A \lor \neg B) \land (\neg A \lor C)$
- is it satisfiable?

Satisfiability modulo theories (SMT)

- $x = 1 \ \land \ x = y + y \ \land \ y > 0$
- is it satisfiable over reals?
- is it satisfiable over integers?

- $\leftarrow$ YOU ARE STANDING HERE

Automated theorem proving (ATP)

- axioms: $\forall x\,(x + x = 0)$, $\forall x \forall y\,(x + y = y + x)$
- do they imply $\forall x \forall y\,((x + y) + (y + x) = 0)$?

We already know

- definition of first-order logic
- definition of first-order theories and satisfiability modulo theories
- theories useful in practice
- CDCL(T) algorithm for solving SMT
- several algorithms for T-solvers (UF, difference logic) and ideas of others
- combination of theories

Next time

- satisfiability of quantified formulas
- first-order resolution (again)
- first-order superposition (again)