Decision Procedures for Verification

Combinations of Decision Procedures (1)

14.01.2019

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Until now:

Decision Procedures

- Uninterpreted functions

  congruence closure

- Numerical domains

  difference logic

  $LI(\mathbb{R})$ and $LI(\mathbb{Q})$

    Method of Fourier-Motzkin

    Method of Weisspfenning-Loos

3.5. Combinations of theories

The combined validity problem

For $i = 1, 2$ • let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
• let $\mathcal{L}_i$ be a class of (closed) Σ-formulae

Let $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ be a combination of $\mathcal{T}_1$ and $\mathcal{T}_2$
Let $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ be a combination of $\mathcal{L}_1$ and $\mathcal{L}_2$

Problem: Given ϕ in $\mathcal{L}_1 \bigoplus \mathcal{L}_2$, is it the case that $\mathcal{T}_1 \bigoplus \mathcal{T}_2 \models \phi$?

Problems

The combined decidability problem I

For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- let $\mathcal{L}_i$ be a class of (closed) Σ-formulae
- assume the $\mathcal{T}_i$-validity problem for $\mathcal{L}_i$ is decidable

Let $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ be a combination of $\mathcal{T}_1$ and $\mathcal{T}_2$
Let $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ be a combination of $\mathcal{L}_1$ and $\mathcal{L}_2$

Question: Is the $\mathcal{T}_1 \bigoplus \mathcal{T}_2$-validity problem for $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ decidable?

Problems

The combined decidability problem II

For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- let $\mathcal{L}_i$ be a class of (closed) Σ-formulae
- P_i decision procedure for $\mathcal{T}_i$-validity for $\mathcal{L}_i$

Let $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ be a combination of $\mathcal{T}_1$ and $\mathcal{T}_2$
Let $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ be a combination of $\mathcal{L}_1$ and $\mathcal{L}_2$

Question: Can we combine P_1 and P_2 modularly into a decision procedure for the $\mathcal{T}_1 \bigoplus \mathcal{T}_2$-validity problem for $\mathcal{L}_1 \bigoplus \mathcal{L}_2$?

Main issue: How are $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ and $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ defined?

Combinations of theories and models

Forgetting symbols

Let $\Sigma = (\Omega, \Pi)$ and $\Sigma' = (\Omega', \Pi')$ s.t. $\Sigma \subseteq \Sigma'$, i.e., $\Omega \subseteq \Omega'$ and $\Pi \subseteq \Pi'$

For $\mathcal{A} \in \Sigma'$-alg, we denote by $\mathcal{A}_{|\Sigma}$ the Σ-structure for which:

$$\begin{array}{lll} U_{\mathcal{A}_{|\Sigma}} = U_{\mathcal{A}}, & f_{\mathcal{A}_{|\Sigma}} = f_{\mathcal{A}} & \text{for } f \in \Omega; \\ & P_{\mathcal{A}_{|\Sigma}} = P_{\mathcal{A}} & \text{for } P \in \Pi \end{array}$$

(ignore functions and predicates associated with symbols in $\Sigma' \backslash \Sigma$)

$\mathcal{A}_{|\Sigma}$ is called the restriction (or the reduct) of $\mathcal{A}$ to Σ.

Example: $\Sigma' = (\{+/2, */2, 1/0\}, \{\leq /2, \text{even}/1, \text{odd}/1\})$

$\Sigma = (\{+/2, 1/0\}, \{\leq /2\}) \subseteq \Sigma'$

$\mathcal{N} = (\mathbb{N}, +, *, 1, \leq, \text{even}, \text{odd})$ $\qquad$ $\mathcal{N}_{|\Sigma} = (\mathbb{N}, +, 1, \leq)$

One possibility of combining theories

Syntactic view: $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\mathrm{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2\}$

where $\Sigma_1 \cup \Sigma_2 = (\Omega_1, \Pi_1) \cup (\Omega_2, \Pi_2) = (\Omega_1 \cup \Omega_2, \Pi_1 \cup \Pi_2)$

One possibility of combining theories

Syntactic view: $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\mathrm{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2\}$

Semantic view: Let $\mathcal{M}_i = \mathrm{Mod}(\mathcal{T}_i), i = 1, 2$

$\mathcal{M}_1 + \mathcal{M}_2 = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A}_{|\Sigma_i} \in \mathcal{M}_i \text{ for } i = 1, 2\}$

One possibility of combining theories

Syntactic view: $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\mathsf{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2\}$

Semantic view: Let $\mathcal{M}_i = \mathsf{Mod}(\mathcal{T}_i), i = 1, 2$

$\mathcal{M}_1 + \mathcal{M}_2 = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A}_{|\Sigma_i} \in \mathcal{M}_i \text{ for } i = 1, 2\}$

$$
\begin{array}{lll}
\mathcal{A} \in \mathsf{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) & \text{iff} & \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2 \\
 & \text{iff} & \mathcal{A}_{|\Sigma_i} \models G, \text{ for all } G \text{ in } \mathcal{T}_i, i = 1, 2 \\
 & \text{iff} & \mathcal{A}_{|\Sigma_i} \in \mathcal{M}_i, i = 1, 2 \\
 & \text{iff} & \mathcal{A} \in \mathcal{M}_1 + \mathcal{M}_2
\end{array}
$$

One possibility of combining theories

Syntactic view: $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\mathrm{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2\}$

Semantic view: Let $\mathcal{M}_i = \mathrm{Mod}(\mathcal{T}_i)$, $i = 1, 2$

$\mathcal{M}_1 + \mathcal{M}_2 = \{\mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A}_{|\Sigma_i} \in \mathcal{M}_i \text{ for } i = 1, 2\}$

Remark: $\mathcal{A} \in \mathrm{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2)$ iff $(\mathcal{A}_{|\Sigma_1} \in \mathrm{Mod}(\mathcal{T}_1)$ and $\mathcal{A}_{|\Sigma_2} \in \mathrm{Mod}(\mathcal{T}_2))$

Consequence: $\mathrm{Th}(\mathrm{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2)) = \mathrm{Th}(\mathcal{M}_1 + \mathcal{M}_2)$

Example

1. **Presburger arithmetic $+$ UIF**

   $\mathrm{Th}(\mathbb{Z}_+) \cup UIF \qquad \Sigma = (\Omega, \Pi)$

   Models: $(A, 0, s, +, \{f_A\}_{f \in \Omega}, \leq, \{P_A\}_{P \in \Pi})$

   where $(A, 0, s, +, \leq) \in \mathrm{Mod}(\mathrm{Th}(\mathbb{Z}_+))$.

2. **The theory of reals $+$ the theory of a monotone function f**

   $\mathrm{Th}(\mathbb{R}) \cup \mathrm{Mon}(f) \qquad \mathrm{Mon}(f) : \forall x, y(x \leq y \rightarrow f(x) \leq f(y))$

   Models: $(A, +, *, f_A, \{\leq\})$, where

   where $(A, +, *, \leq) \in \mathrm{Mod}(\mathrm{Th}(\mathbb{R}))$.

   $$(A, f_A, \leq) \models \mathrm{Mon}(f), \text{ i.e. } f_A : A \rightarrow A \text{ monotone.}$$

   Note: The signatures of the two theories share the $\leq$ predicate symbol

Combinations of theories

Definition. A theory is consistent if it has at least one model.

Question: Is the union of two consistent theories always consistent?

Answer: No. (Not even when the two theories have disjoint signatures)

Example: $\Sigma_1 = (\Omega_1, \emptyset), \quad \Sigma_2 = (\{c/0, d/0\}, \emptyset), \quad c, d \notin \Omega_1$

$$\mathcal{T}_1 = \{\exists x, y, z(x \not\approx y \ \wedge \ x \not\approx z \ \wedge \ y \not\approx z)\}$$

$$\mathcal{T}_2 = \{\forall x(x \approx c \ \vee \ x \approx d)\}$$

$$\mathcal{A} \in \mathrm{Mod}(\mathcal{T}_1) \quad \text{iff} \quad |U_{\mathcal{A}}| \geq 3.$$

$$\mathcal{B} \in \mathrm{Mod}(\mathcal{T}_2) \quad \text{iff} \quad |U_{\mathcal{B}}| \leq 2.$$

Combinations of theories

The combined decidability problem

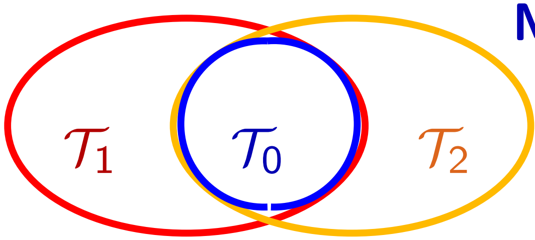
For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- assume the $\mathcal{T}_i$ ground satisfiability problem is decidable

Let $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ be a combination of $\mathcal{T}_1$ and $\mathcal{T}_2$

Question:

Is the $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ ground satisfiability problem decidable?

Goal: Modularity



Modular Reasoning

$\mathcal{T}_0$: Σ_0-theory.

$\mathcal{T}_i$: Σ_i-theory; $\quad \mathcal{T}_0 \subseteq \mathcal{T}_i \quad \Sigma_0 \subseteq \Sigma_i$.

Example:

lists$(\mathbb{R}) \cup$ **arrays**$(\mathbb{R})$

Can use provers for $\mathcal{T}_1, \mathcal{T}_2$ as blackboxes to prove theorems in $\mathcal{T}_1 \cup \mathcal{T}_2$?

Which information needs to be exchanged between the provers?

Combinations of theories

For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- s.t. the ground satisfiability problem for $\mathcal{T}_i$ is decidable

Question: Is the ground satisfiability problem for $\mathcal{T}_1 \cup \mathcal{T}_2$ decidable?

Combinations of theories

For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- s.t. the ground satisfiability problem for $\mathcal{T}_i$ is decidable

Question: Is the ground satisfiability problem for $\mathcal{T}_1 \cup \mathcal{T}_2$ decidable?

In general: No (restrictions needed for affirmative answer)

Example. Word problem for $\mathcal{T}$: Decide if $\mathcal{T} \models \forall x(s \approx t)$

$\mathcal{A}$: theory of associativity

$\mathcal{G}$ finite set of ground equations
(presentation for semigroup
with undecidable word problem)

$\uparrow$

($\exists$ finitely-presented semigroup with
undecidable word problem [Matijasevic'67])

Word problem: decidable for $\mathcal{A}, \mathcal{G}$; undecidable for $\mathcal{A} \cup \mathcal{G}$

Combinations of theories

For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- s.t. the ground satisfiability problem for $\mathcal{T}_i$ is decidable

Question: Is the ground satisfiability problem for $\mathcal{T}_1 \cup \mathcal{T}_2$ decidable?

In general: No (restrictions needed for affirmative answer)

Example. Word problem for $\mathcal{T}$: Decide if $\mathcal{T} \models \forall x(s \approx t)$

Simpler instances: combinations of theories over disjoint signatures, theories sharing constructors, compatibility with shared theory ...

Combinations of theories

For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- s.t. the ground satisfiability problem for $\mathcal{T}_i$ is decidable

Question: Is the ground satisfiability problem for $\mathcal{T}_1 \cup \mathcal{T}_2$ decidable?

In general: No (restrictions needed for affirmative answer)

Theorem [Bonacina, Ghilardi et.al, IJCAR 2006]
There are theories $\mathcal{T}_1, \mathcal{T}_2$ with disjoint signatures and decidable ground satisfiability problem such that ground satisfiability in $\mathcal{T}_1 \cup \mathcal{T}_2$ is undecidable.

Idea: Construct $\mathcal{T}_1$ such that ground satisfiability is decidable, but it is undecidable whether a constraint Γ_1 is satisfiable in an infinite model of $\mathcal{T}_1$. (Construction uses Turing Machines). Let $\mathcal{T}_2$ having only infinite models.

Combination of theories over disjoint signatures

The Nelson/Oppen procedure

Given: $\mathcal{T}_1, \mathcal{T}_2$ first-order theories with signatures Σ_1, Σ_2

Assume that $\Sigma_1 \cap \Sigma_2 = \emptyset$ (share only $\approx$)

P_i decision procedures for satisfiability of ground formulae w.r.t. $\mathcal{T}_i$

ϕ quantifier-free formula over $\Sigma_1 \cup \Sigma_2$

Task: Check whether ϕ is satisfiable w.r.t. $\mathcal{T}_1 \cup \mathcal{T}_2$

Note: Restrict to conjunctive quantifier-free formulae

$\phi \mapsto DNF(\phi)$

$DNF(\phi)$ satisfiable in $\mathcal{T}$ iff one of the disjuncts satisfiable in $\mathcal{T}$

Example

[Nelson & Oppen, 1979]

Theories

$\mathcal{R}$	theory of rationals	$\Sigma_{\mathcal{R}} = \{\leq, +, -, 0, 1\}$	$\approx$
$\mathcal{L}$	theory of lists	$\Sigma_{\mathcal{L}} = \{\mathsf{car}, \mathsf{cdr}, \mathsf{cons}\}$	$\approx$
$\mathcal{E}$	theory of equality (UIF)	Σ: free function and predicate symbols	$\approx$

Example

[Nelson & Oppen, 1979]

Theories

$\mathcal{R}$	theory of rationals	$\Sigma_{\mathcal{R}} = \{\leq, +, -, 0, 1\}$	$\approx$
$\mathcal{L}$	theory of lists	$\Sigma_{\mathcal{L}} = \{\mathsf{car}, \mathsf{cdr}, \mathsf{cons}\}$	$\approx$
$\mathcal{E}$	theory of equality (UIF)	Σ: free function and predicate symbols	$\approx$

Problems:

1. $\mathcal{R}\cup\mathcal{L}\cup\mathcal{E} \models \forall x, y(x\leq y \wedge y\leq x+\mathsf{car}(\mathsf{cons}(0, x)) \wedge P(h(x)-h(y)) \rightarrow P(0))$

2. Is the following conjunction:

$$c \leq d \ \wedge \ d \leq c + \mathsf{car}(\mathsf{cons}(0, c)) \ \wedge \ P(h(c) - h(d)) \ \wedge \ \neg P(0)$$

satisfiable in $\mathcal{R} \cup \mathcal{L} \cup \mathcal{E}$?

An Example

	$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
Σ	$\{\leq, +, -, 0, 1\}$	$\{\text{car}, \text{cdr}, \text{cons}\}$	$F \cup P$
Axioms	$x + 0 \approx x$	$\text{car}(\text{cons}(x, y)) \approx x$	
	$x - x \approx 0$	$\text{cdr}(\text{cons}(x, y)) \approx y$	
(univ.	$+$ is A, C	$\text{at}(x) \vee \text{cons}(\text{car}(x), \text{cdr}(x)) \approx x$	
quantif.)	$\leq$ is R, T, A	$\neg \text{at}(\text{cons}(x, y))$	
	$x \leq y \vee y \leq x$		
	$x \leq y \rightarrow x + z \leq y + z$		

Is the following conjunction:

$$c \leq d \;\wedge\; d \leq c + \text{car}(\text{cons}(0, c)) \;\wedge\; P(h(c) - h(d)) \;\wedge\; \neg P(0)$$

satisfiable in $\mathcal{R} \cup \mathcal{L} \cup \mathcal{E}$?

Step 1: Purification

Given: ϕ conjunctive quantifier-free formula over $\Sigma_1 \cup \Sigma_2$

Task: Find ϕ_1, ϕ_2 s.t. ϕ_i is a pure Σ_i-formula and $\phi_1 \wedge \phi_2$ equivalent with ϕ

$$
\begin{array}{lcl}
f(s_1, \ldots, s_n) \approx g(t_1, \ldots, t_m) & \mapsto & u{\approx}f(s_1, \ldots, s_n) \wedge u{\approx}g(t_1, \ldots, t_m) \\
f(s_1, \ldots, s_n) \not\approx g(t_1, \ldots, t_m) & \mapsto & u{\approx}f(s_1, \ldots, s_n) \wedge v{\approx}g(t_1, \ldots, t_m) \wedge u \not\approx v \\
(\neg)P(\ldots, s_i, \ldots) & \mapsto & (\neg)P(\ldots, u, \ldots) \wedge u{\approx}s_i \\
(\neg)P(\ldots, s_i[t], \ldots) & \mapsto & (\neg)P(\ldots, s_i[t \mapsto u], \ldots) \wedge u{\approx}t \\
\quad \text{where } t \approx f(t_1, \ldots, t_n) & &
\end{array}
$$

Termination: Obvious

Correctness: $\phi_1 \wedge \phi_2$ and ϕ equisatisfiable.

Step 1: Purification

$$c \le d \;\wedge\; d \le c + \mathrm{car}(\mathrm{cons}(0, c)) \;\wedge\; P(h(c) - h(d)) \;\wedge\; \neg P(0)$$

Step 1: Purification

$$c \leq d \ \wedge \ d \leq c + \underbrace{\mathsf{car}(\mathsf{cons}(0, c))}_{c_1} \ \wedge \ P(h(c) - h(d)) \ \wedge \ \neg P(0)$$

Step 1: Purification

$$c \leq d \;\wedge\; d \leq c + \underbrace{\mathrm{car}(\mathrm{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{h(c) - h(d)}_{c_2}) \;\wedge\; \neg P(0)$$

Step 1: Purification

$$c \leq d \;\wedge\; d \leq c + \underbrace{\mathrm{car}(\mathrm{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \;\wedge\; \neg P(\underbrace{\;0\;}_{c_5})$$

Step 1: Purification

$$c \leq d \;\wedge\; d \leq c + \underbrace{\mathrm{car}(\mathrm{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \;\wedge\; \neg P(\underbrace{0}_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \leq d$	$c_1 \approx \mathrm{car}(\mathrm{cons}(c_5, c))$	$P(c_2)$
$d \leq c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$

Step 1: Purification

$$c \leq d \;\wedge\; d \leq c + \underbrace{\mathrm{car}(\mathrm{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \;\wedge\; \neg P(\underbrace{\;0\;}_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \leq d$	$c_1 \approx \mathrm{car}(\mathrm{cons}(c_5, c))$	$P(c_2)$
$d \leq c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$
satisfiable	satisfiable	satisfiable

Step 2: Propagation

$$c \leq d \ \wedge \ d \leq c + \underbrace{\mathsf{car}(\mathsf{cons}(0, c))}_{c_1} \ \wedge \ P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \ \wedge \ \neg P(\underbrace{\ 0\ }_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \leq d$	$c_1 \approx \mathsf{car}(\mathsf{cons}(c_5, c))$	$P(c_2)$
$d \leq c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$

deduce and propagate equalities between constants entailed by components

Step 2: Propagation

$$c \leq d \;\wedge\; d \leq c + \underbrace{\mathrm{car}(\mathrm{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \;\wedge\; \neg P(\underbrace{0}_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \leq d$	$c_1 \approx \mathrm{car}(\mathrm{cons}(c_5, c))$	$P(c_2)$
$d \leq c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$

$$c_1 \approx c_5$$

Step 2: Propagation

$$c \leq d \ \wedge \ d \leq c + \underbrace{\mathsf{car}(\mathsf{cons}(0, c))}_{c_1} \ \wedge \ P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \ \wedge \ \neg P(\underbrace{0}_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \leq d$	$c_1 \approx \mathsf{car}(\mathsf{cons}(c_5, c))$	$P(c_2)$
$d \leq c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$
$c_1 \approx c_5$	$c_1 \approx c_5$	
$c \approx d$		

Step 2: Propagation

$$c \leq d \;\wedge\; d \leq c + \underbrace{\mathrm{car}(\mathrm{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \;\wedge\; \neg P(\underbrace{0}_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \leq d$	$c_1 \approx \mathrm{car}(\mathrm{cons}(c_5, c))$	$P(c_2)$
$d \leq c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$
$c_1 \approx c_5$	$c_1 \approx c_5$	$c \approx d$
$c \approx d$		$c_3 \approx c_4$

Step 2: Propagation

$$c \le d \;\wedge\; d \le c + \underbrace{\mathsf{car}(\mathsf{cons}(0, c))}_{c_1} \;\wedge\; P(\underbrace{\underbrace{h(c)}_{c_3} - \underbrace{h(d)}_{c_4}}_{c_2}) \;\wedge\; \neg P(\underbrace{0}_{c_5})$$

$\mathcal{R}$	$\mathcal{L}$	$\mathcal{E}$
$c \le d$	$c_1 \approx \mathsf{car}(\mathsf{cons}(c_5, c))$	$P(c_2)$
$d \le c + c_1$		$\neg P(c_5)$
$c_2 \approx c_3 - c_4$		$c_3 \approx h(c)$
$c_5 \approx 0$		$c_4 \approx h(d)$
$c_1 \approx c_5$	$c_1 \approx c_5$	$c \approx d$
$c \approx d$		$c_3 \approx c_4$
$c_2 \approx c_5$		$\bot$

The Nelson-Oppen algorithm

ϕ conjunction of literals

Step 1. Purification $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \phi \mapsto (\mathcal{T}_1 \cup \phi_1) \cup (\mathcal{T}_2 \cup \phi_2)$:

where ϕ_i is a pure Σ_i-formula and $\phi_1 \wedge \phi_2$ is equisatisfiable with ϕ.

Step 2. Propagation.

The decision procedure for ground satisfiability for $\mathcal{T}_1$ and $\mathcal{T}_2$ fairly exchange information concerning entailed unsatisfiability of constraints in the shared signature
i.e. clauses over the shared variables.

until an inconsistency is detected or a saturation state is reached.

The Nelson-Oppen algorithm

ϕ conjunction of literals

Step 1. Purification $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \phi \mapsto (\mathcal{T}_1 \cup \phi_1) \cup (\mathcal{T}_2 \cup \phi_2)$:

where ϕ_i is a pure Σ_i-formula and $\phi_1 \wedge \phi_2$ is equisatisfiable with ϕ.

not problematic; requires linear time

Step 2. Propagation.

The decision procedure for ground satisfiability for $\mathcal{T}_1$ and $\mathcal{T}_2$ fairly exchange information concerning entailed unsatisfiability of constraints in the shared signature

i.e. clauses over the shared variables.

until an inconsistency is detected or a saturation state is reached.

not problematic; termination guaranteed

Sound: if inconsistency detected input unsatisfiable

Complete: under additional assumptions

Implementation

ϕ conjunction of literals

Step 1. Purification: $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \phi \mapsto (\mathcal{T}_1 \cup \phi_1) \cup (\mathcal{T}_2 \cup \phi_2)$,
where ϕ_i is a pure Σ_i-formula and $\phi_1 \wedge \phi_2$ is equisatisfiable with ϕ.

Step 2. Propagation: The decision procedure for ground satisfiability for $\mathcal{T}_1$ and $\mathcal{T}_2$ fairly exchange information concerning entailed unsatisfiability of constraints in the shared signature
i.e. clauses over the shared variables.

until an inconsistency is detected or a saturation state is reached.

How to implement Propagation?

Guessing: guess a maximal set of literals containing the shared variables; check it for $\mathcal{T}_i \cup \phi_i$ consistency.

Backtracking: identify disjunction of equalities between shared variables entailed by $\mathcal{T}_i \cup \phi_i$; make case split by adding some of these equalities to ϕ_1, ϕ_2. Repeat as long as possible.

Implementation of propagation

Guessing variant

Guess a maximal set of literals containing the shared variables V (arrangement: $\alpha(V, E) = (\bigwedge_{(u,v)\in E} u \approx v \wedge \bigwedge_{(u,v)\notin E} u \not\approx v)$, where E equivalence relation); check it for $\mathcal{T}_i \cup \phi_i$ consistency.

On the blackboard: Example 10.5 and 10.7 pages 272, 273
Example 10.6 and 10.9 pages 272, 275

from the book "The Calculus of Computation" by A. Bradley and Z. Manna

Advantage: Whenever constraints are represented as Boolean combinations of atoms, one may combine heuristics of SMT solvers with specific features of the theories to be combined to produce the right arrangement efficiently.

Implementation of propagation

Backtracking variant

Identify disjunction of equalities between shared variables entailed by $\mathcal{T}_i \cup \phi_i$; make case split by adding some of these equalities to ϕ_1, ϕ_2.

Repeat as long as possible.

On the blackboard: Example 10.14, page 280-281, and Example 10.13, page 279, from the book "The Calculus of Computation" by A. Bradley and Z. Manna

Advantages:

- it works on the non-disjoint case as well
- can be made deterministic for combinations of convex theories

$\mathcal{T}$ convex iff whenever $\mathcal{T} \models \bigwedge_{i=1}^{n} A_i \rightarrow \bigvee_{j=1}^{m} B_j$
there exists j s.t. $\mathcal{T} \models \bigwedge_{i=1}^{n} A_i \rightarrow B_j$

The Nelson-Oppen algorithm

Termination: only finitely many shared variables to be identified

The Nelson-Oppen algorithm

Termination: only finitely many shared variables to be identified

Soundness: If procedure answers "unsatisfiable" then ϕ is unsatisfiable

Proof: Assume that ϕ is satisfiable. Then $\phi_1 \wedge \phi_2$ satisfiable.

- The procedure cannot answer "unsatisfiable" in Step 2.
- Let $(\mathcal{M}, \beta) \models \phi_1 \wedge \phi_2$. Assume that $(\mathcal{M}, \beta) \models \bigwedge_{(c_i,c_j)\in E} c_i \approx c_j \wedge \bigwedge_{(c_i,c_j)\notin E} c_i \not\approx c_j$

Then
$$(\mathcal{M}_{|\Sigma_1}, \beta) \models \phi_1 \wedge \bigwedge_{(c_i,c_j)\in E} c_i \approx c_j$$
$$(\mathcal{M}_{|\Sigma_2}, \beta) \models \phi_2 \wedge \bigwedge_{(c_i,c_j)\in E} c_i \approx c_j$$

Guessing: $\bigwedge_{(c_i,c_j)\in E} c_i \approx c_j \wedge \bigwedge_{(c_i,c_j)\notin E} c_i \not\approx c_j$ "satisfiable arrangement".

Backtracking: Procedure answers satisfiable on the corresponding branch.

The Nelson-Oppen algorithm

Termination:	only finitely many shared variables to be identified
Soundness:	If procedure answers "unsatisfiable" then ϕ is unsatisfiable
Completeness:	Under additional hypotheses

Completeness

Example:

E_1	E_2
$f(g(x), g(y)) \approx x$	$k(x) \approx k(x)$
$f(g(x), h(y)) \approx y$	
non-trivial	non-trivial

$g(c) \approx h(c) \wedge k(c) \not\approx c$

$g(c) \approx h(c)$	$k(c) \not\approx c$
satisfiable in E_1	satisfiable in E_2

no equations between shared variables; Nelson-Oppen answers "satisfiable"

Completeness

Example:

E_1	E_2
$f(g(x), g(y)) \approx x$	$k(x) \approx k(x)$
$f(g(x), h(y)) \approx y$	
non-trivial	non-trivial

$g(c) \approx h(c) \wedge k(c) \not\approx c$

$g(c) \approx h(c)$	$k(c) \not\approx c$
satisfiable in E_1	satisfiable in E_2

no equations between shared variables; Nelson-Oppen answers "satisfiable"

> A model of E_1 satisfies $g(c) \approx h(c)$ iff $\exists e \in A$ s.t. $g(e) = h(e)$.
> Then, for all $a \in A$: $a = f_A(g(a), g(e)) = f_A(g(a), h(e)) = e$

$g(c) \approx h(c) \wedge k(c) \not\approx c$ unsatisfiable

Completeness

Another example

$\mathcal{T}_1$ theory admitting models of cardinality at most 2

$\mathcal{T}_2$ theory admitting models of any cardinality

$$f_1 \in \Sigma_1, f_2 \in \Sigma_2 \quad \text{such that} \quad \mathcal{T}_i \not\models \forall x, y \quad f_i(x) = f_i(y).$$

$$\begin{array}{lcl} \phi & = & f_1(c_1) \not\approx f_1(c_2) \quad \wedge \quad f_2(c_1) \not\approx f_2(c_3) \quad \wedge \quad f_2(c_2) \not\approx f_2(c_3) \\ \phi_1 & = & f_1(c_1) \not\approx f_1(c_2) \qquad \phi_2 \ = \ f_2(c_1) \not\approx f_2(c_3) \ \wedge \ f_2(c_2) \not\approx f_2(c_3) \end{array}$$

The Nelson-Oppen procedure returns "satisfiable"

$$\mathcal{T}_1 \cup \mathcal{T}_2 \models \forall x, y, z(f_1(x) \not\approx f_1(y) \ \wedge \ f_2(x) \not\approx f_2(z) \ \wedge \ f_2(y) \not\approx f_2(z) \\ \rightarrow (x \not\approx y \wedge x \not\approx z \wedge y \not\approx z))$$

$f_1(c_1) \not\approx f_1(c_2) \quad \wedge \quad f_2(c_1) \not\approx f_2(c_3) \quad \wedge \quad f_2(c_2) \not\approx f_2(c_3)$ unsatisfiable

Completeness

Cause of incompleteness

There exist formulae satisfiable in finite models of bounded cardinality

Solution: Consider stably infinite theories.

$\mathcal{T}$ is stably infinite iff for every quantifier-free formula ϕ

ϕ satisfiable in $\mathcal{T}$ iff ϕ satisfiable in an infinite model of $\mathcal{T}$.

Note: This restriction is not mentioned in [Nelson Oppen 1979];

introduced by Oppen in 1980.

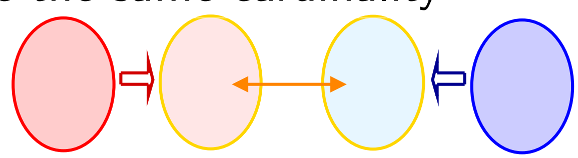
Completeness

Guessing version: C set of constants shared by ϕ_1, ϕ_2

R equiv. relation assoc. with partition of $C \mapsto ar(C, R) = \bigwedge_{R(c,d)} c \approx d \wedge \bigwedge_{\neg R(c,d)} c \not\approx d$

Lemma. Assume that there exists a partition of C s.t. $\phi_i \wedge ar(C, R)$ is $\mathcal{T}_i$-satisfiable. Then $\phi_1 \wedge \phi_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$-satisfiable.

Idea of proof: Let $\mathcal{A}_i \in \text{Mod}(\mathcal{T}_i)$ s.t. $\mathcal{A}_i \models \phi_i \wedge ar(C, R)$. Then $c_{A_1} = d_{A_1}$ iff $c_{A_2} = d_{A_2}$.
Let $i : \{c_{A_1} \mid c \in C\} \to \{c_{A_2} \mid c \in C\}$, $i(c_{A_1}) = c_{A_2}$ well-defined; bijection.
Stable infinity: can assume w.l.o.g. that $\mathcal{A}_1, \mathcal{A}_2$ have the same cardinality
Let $h : \mathcal{A}_1 \to \mathcal{A}_2$ bijection s.t. $h(c_{A_1}) = c_{A_2}$



Use h to transfer the Σ_1-structure on $\mathcal{A}_2$.

Theorem. If $\mathcal{T}_1, \mathcal{T}_2$ are both stably infinite and the shared signature is empty then the Nelson-Oppen procedure is sound, complete and terminating. Thus, it transfers decidability of ground satisfiability from $\mathcal{T}_1, \mathcal{T}_2$ to $\mathcal{T}_1 \cup \mathcal{T}_2$.

Complexity

Main sources of complexity:

(i) transformation of the formula in DNF

(ii) propagation

(a) decide whether there is a disjunction of equalities between variables

(b) investigate different branches corresponding to disjunctions

Complexity

Main sources of complexity:

(i) transformation of the formula in DNF

(ii) propagation

$\mathcal{T}$ is convex	iff	for every quantifier-free formula ϕ,
		$\phi \models \bigvee_i x_i \approx y_i$ implies $\phi \models x_j \approx y_j$ for some j.

$\mapsto$ No branching

Complexity

Main sources of complexity:

(i) transformation of the formula in DNF

(ii) propagation

$\mathcal{T}$ is convex	iff	for every quantifier-free formula ϕ,
		$\phi \models \bigvee_i x_i \approx y_i$ implies $\phi \models x_j \approx y_j$ for some j.

$\mapsto$ No branching

Theorem.	Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be convex and stably infinite; $\Sigma_1 \cap \Sigma_2 = \emptyset$
	If satisfiability of conjunctions of literals in $\mathcal{T}_i$ is in PTIME
	Then satisfiability of conjunctions of literals in $\mathcal{T}_1 \cup \mathcal{T}_2$ is in PTIME

Complexity

In general: non-deterministic procedure

Theorem.	Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be convex and stably infinite; $\Sigma_1 \cap \Sigma_2 = \emptyset$
	If satisfiability of conjunctions of literals in $\mathcal{T}_i$ is in NP
	Then satisfiability of conjunctions of literals in $\mathcal{T}_1 \cup \mathcal{T}_2$ is in NP

Extensions of the Nelson-Oppen procedure

- relax the stable infiniteness requirement

- relax the requirement that the theories have disjoint signatures

Extensions of the Nelson-Oppen procedure

- relax the stable infiniteness requirement

> [Tinelli,Zarba'03] One theory "shiny" (for each satisf. constraint we can compute a finite k s.t. the theory has models of every cardinality $\lambda \geq k$)

- relax the requirement that the theories have disjoint signatures

> [Tinelli,Ringeissen'03] Theories sharing absolutely free constructors
>
> [Ghilardi'04] Model theoretical conditions.

Main idea:

Find situations in which $\mathcal{T}_i$ models of $\phi_i, i = 1, 2$ can be "amalgamated" to a $\mathcal{T}_1 \cup \mathcal{T}_2$ model of $\phi_1 \wedge \phi_2$.