Decision Procedures for Verification

Decision Procedures (4)

Combinations of decision procedures

20.01.2015

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Until now:

Decidable subclasses of FOL

The Bernays-Schönfinkel class
(definition; decidability;tractable fragment: Horn clauses)
The Ackermann class
The monadic class

Decision problems/restrictions

Uninterpreted function symbols

Decision procedures for numeric domains

Difference logic
Linear arithmetic over $\mathbb{R}$, $\mathbb{Q}$
The Fourier-Motzkin method
The Loos/Weispfenning method.

Loos-Weispfenning Quantifier Elimination

A more efficient way to eliminate quantifiers in linear rational arithmetic was developed by R. Loos and V. Weispfenning (1993).

The method is also known as “test point method” or “virtual substitution method”.

For simplicity, we consider only one particular ODAG, namely $\mathbb{Q}$ (as we have seen above, the results are the same for all ODAGs).

Loos-Weispfenning Quantifier Elimination

Let $F(x, \overline{y})$ be a positive boolean combination of linear (in-)equations of the form $\quad x \sim_i s_i(\overline{y}) \quad$ and $\quad 0 \sim_j s_j(\overline{y})$ with $\sim_i, \sim_j \in \{\approx, \not\approx, <, \leq, >, \geq\}$,

(i.e. a formula built from linear (in-) equations, $\vee$ and $\wedge$, but without $\neg$).

Goal: Find a finite set T of "test points" so that

$$\exists x F(x, \overline{y}) \models\mid \bigvee_{t \in T} F(x, \overline{y})[t/x].$$

In other words:

We want to replace the infinite disjunction $\exists x$ by a finite disjunction.

Loos-Weispfenning Quantifier Elimination

If we keep the values of the variables $\overline{y}$ fixed, we can regard F as a function

$$F : \mathbb{Q} \to \{0, 1\} \quad \text{defined by } x \mapsto F(x, \overline{y})$$

Remarks:

(1) The value of each of the atoms $s_i(\overline{y}) \sim_i x$ changes only at $s_i(\overline{y})$,
(2) The value of F can only change if the value of one of its atoms changes.

(3) F is a piecewise constant function; more precisely:
the set of all x with $F(x, \overline{y}) = 1$ is a finite union of intervals.

(The union may be empty, the individual intervals may be finite or infinite and open or closed.)

Let $\delta(\overline{y}) = \min\{|s_i(\overline{y}) - s_j(\overline{y})| \mid s_i(\overline{y}) \neq s_j(\overline{y})\}$.

Each of the intervals has either length 0 (i.e., it consists of one point), or its length is at least $\delta(\overline{y})$.

Loos-Weispfenning Quantifier Elimination

If the set of all x for which $F(x, \overline{y})$ is 1 is non-empty, then

(i) $F(x, \overline{y}) = 1$ for all $x \leq r(\overline{y})$ for some $r(\overline{y}) \in \mathbb{Q}$

(ii) or there is some point where the value of $F(x, \overline{y})$ switches from 0 to 1 when we traverse the real axis from $-\infty$ to $+\infty$.

We use this observation to construct a set of test points.

Loos-Weispfenning Quantifier Elimination

We start with a "sufficiently small" test point $r(\overline{y})$ to take care of case (i).

For case (ii), we observe that $F(x, \overline{y})$ can only switch from 0 to 1 if one of the atoms switches from 0 to 1. (We consider only positive boolean combinations of atoms and $\wedge$ and $\vee$ are monotonic w.r.t. truth values.)

- $x \leq s_i(\overline{y})$ and $x < s_i(\overline{y})$ do not switch from 0 to 1 when x grows.
- $x \geq s_i(\overline{y})$ and $x \approx s_i(\overline{y})$ switch from 0 to 1 at $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y})$ is a test point.
- $x > s_i(\overline{y})$ and $x \not\approx s_i(\overline{y})$ switch from 0 to 1 "right after" $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y}) + \epsilon$ (for some $0 < \epsilon < \delta(\overline{y})$) is a test point.

Loos-Weispfenning Quantifier Elimination

We start with a "sufficiently small" test point $r(\overline{y})$ to take care of case (i).

For case (ii), we observe that $F(x, \overline{y})$ can only switch from 0 to 1 if one of the atoms switches from 0 to 1. (We consider only positive boolean combinations of atoms and $\wedge$ and $\vee$ are monotonic w.r.t. truth values.)

- $x \leq s_i(\overline{y})$ and $x < s_i(\overline{y})$ do not switch from 0 to 1 when x grows.
- $x \geq s_i(\overline{y})$ and $x \approx s_i(\overline{y})$ switch from 0 to 1 at $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y})$ is a test point.
- $x > s_i(\overline{y})$ and $x \not\approx s_i(\overline{y})$ switch from 0 to 1 "right after" $s_i(\overline{y})$
  $\Rightarrow$ $s_i(\overline{y}) + \epsilon$ (for some $0 < \epsilon < \delta(\overline{y})$) is a test point.

If $r(\overline{y})$ is sufficiently small and $0 < \epsilon < \delta(\overline{y})$, then

$$T := \{r(\overline{y})\} \cup \{s_i(\overline{y}) \mid \sim_i \in \{\geq, \approx\}\} \cup \{s_i(\overline{y}) + \epsilon \mid \sim_i \in \{>, \not\approx\}\}.$$

is a set of test points.

Loos-Weispfenning Quantifier Elimination

Problems:

(1) We don't know how small $r(\overline{y})$ has to be for case (i).

(2) We don't know $\delta(\overline{y})$ for case (ii).

Idea: We consider the limits for $r \to -\infty$ and for $\epsilon \to 0$ (but positive), that is, we redefine

$$T := \{-\infty\} \cup \{s_i(\overline{y}) \mid \sim_i \in \{\geq, \approx\}\} \cup \{s_i(\overline{y}) + \epsilon \mid \sim_i \in \{>, \not\approx\}\}.$$

New problem:

How can we eliminate the infinitesimals $-\infty$ and ϵ when we substitute elements of T for x?

Loos-Weispfenning Quantifier Elimination

Virtual substitution:

$(x < s(\overline{y}))[-\infty/x] := \lim_{r\to-\infty}(r < s(\overline{y})) = \top$

$(x \leq s(\overline{y}))[-\infty/x] := \lim_{r\to-\infty}(r \leq s(\overline{y})) = \top$

$(x > s(\overline{y}))[-\infty/x] := \lim_{r\to-\infty}(r > s(\overline{y})) = \bot$

$(x \geq s(\overline{y}))[-\infty/x] := \lim_{r\to-\infty}(r \geq s(\overline{y})) = \bot$

$(x \approx s(\overline{y}))[-\infty/x] := \lim_{r\to-\infty}(r \approx s(\overline{y})) = \bot$

$(x \not\approx s(\overline{y}))[-\infty/x] := \lim_{r\to-\infty}(r \not\approx s(\overline{y})) = \top$

Loos-Weispfenning Quantifier Elimination

Virtual substitution:

$$(x < s(\bar{y}))[u+\epsilon/x] := \lim_{\substack{\epsilon\to 0\\ \epsilon>0}}(u+\epsilon < s(\bar{y})) = (u < s(\bar{y}))$$

$$(x \leq s(\bar{y}))[u+\epsilon/x] := \lim_{\substack{\epsilon\to 0\\ \epsilon>0}}(u+\epsilon \leq s(\bar{y})) = (u < s(\bar{y}))$$

$$(x > s(\bar{y}))[u+\epsilon/x] := \lim_{\substack{\epsilon\to 0\\ \epsilon>0}}(u+\epsilon > s(\bar{y})) = (u \geq s(\bar{y}))$$

$$(x \geq s(\bar{y}))[u+\epsilon/x] := \lim_{\substack{\epsilon\to 0\\ \epsilon>0}}(u+\epsilon \geq s(\bar{y})) = (u \geq s(\bar{y}))$$

$$(x \approx s(\bar{y}))[u+\epsilon/x] := \lim_{\substack{\epsilon\to 0\\ \epsilon>0}}(u+\epsilon \approx s(\bar{y})) = \bot$$

$$(x \not\approx s(\bar{y}))[u+\epsilon/x] := \lim_{\substack{\epsilon\to 0\\ \epsilon>0}}(u+\epsilon \not\approx s(\bar{y})) = \top$$

We have traversed the real axis from $-\infty$ to $+\infty$.

Loos-Weispfenning Quantifier Elimination

Virtual substitution:

Alternatively, we can traverse it from $+\infty$ to $-\infty$.

In this case, the test points are

$$T' := \{+\infty\} \cup \{s_i(\overline{y}) \mid \sim_i \in \{\leq, \approx\}\} \cup \{s_i(\overline{y}) - \epsilon \mid \sim_i \in \{<, \not\approx\}\}.$$

Infinitesimals are eliminated in a similar way as before.

In practice: Compute both T and T' and take the smaller set.

For a universally quantified formula $\forall x F$, we replace it by $\neg\exists x \neg F$, push inner negation downwards, and then continue as before.

Note that there is no CNF/DNF transformation required.

Loos-Weispfenning quantifier elimination works on arbitrary positive formulas.

Loos-Weispfenning: Complexity

- **One LW-step for $\exists$ or $\forall$:**

  As the number of test points is at most equal to the number of atoms, the formula size grows quadratically; therefore $O(n^2)$ runtime.

- **Multiple quantifiers of the same kind:**

$$
\begin{array}{ll}
 & \exists x_2 \exists x_1 . F(x_1, x_2, \overline{y}) \\
\mapsto & \exists x_2 . \bigvee_{t_1 \in T_1} F(x_1, x_2, \overline{y})[t_1/x_1] \\
\mapsto & \bigvee_{t_1 \in T_1} (\exists x_2 . F(x_1, x_2, \overline{y})[t_1/x_1]) \\
\mapsto & \bigvee_{t_1 \in T_1} \bigvee_{t_2 \in T_2} F(x_1, x_2, \overline{y})[t_1/x_1][t_2/x_2]
\end{array}
$$

- **m quantifiers $\exists \ldots \exists$ or $\forall \ldots \forall$:**
  formula size is multiplied by n in each step $\Rightarrow O(n^{m+1})$ runtime.

- **m quantifiers $\exists\forall\exists\forall \ldots \forall$:** doubly exponential runtime.

Note: The formula resulting from a LW-step is usually highly redundant. An efficient implementation must make use of simplification techniques.

Until now

Decidable fragments of first-order logic

Decision procedures for single theories

- UIF
- Numeric domains

  Here:

  Difference logic

  Linear arithmetic over $\mathbb{R}$, $\mathbb{Q}$

Next: Reasoning in combinations of theories

Combinations of decision procedures

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:

$$U_{\mathcal{A}_{|\Sigma}} = U_{\mathcal{A}}, \quad f_{\mathcal{A}_{|\Sigma}} = f_{\mathcal{A}} \quad \text{for } f \in \Omega;$$
$$P_{\mathcal{A}_{|\Sigma}} = P_{\mathcal{A}} \quad \text{for } P \in \Pi$$

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

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

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

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

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

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

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

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

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

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

Example

1. **Presburger arithmetic + UIF**

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

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

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

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

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

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

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

   $(A, f_A, \leq) \models Mon(f)$, i.e. $f_A : A \rightarrow A$ 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

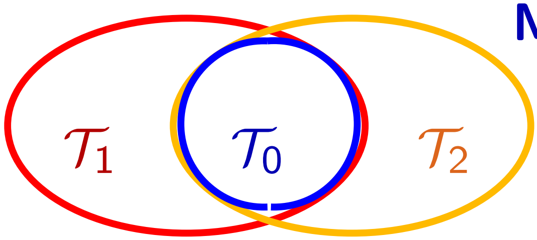
For $i = 1, 2$
- let $\mathcal{T}_i$ be a first-order theory in signature Σ_i
- assume the $\mathcal{T}_i$ ground satisfiability problem is decidable

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; $\mathcal{T}_0 \subseteq \mathcal{T}_i$ $\Sigma_0 \subseteq \Sigma_i$.

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

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

Step 1: Purification

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

Step 1: Purification

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

Step 1: Purification

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

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

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

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

Step 2: Propagation

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

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

deduce and propagate equalities between constants entailed by components

Step 2: Propagation

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

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

$$c_1 \approx c_5$$

Step 2: Propagation

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

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

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

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

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

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.