Decision Procedures for Verification

Combinations of Decision Procedures (2)

2.02.2017

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Last time

Combinations of Decision Procedures

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	$\Sigma_{\mathcal{L}} = \{\text{car}, \text{cdr}, \text{cons}\}$	$\approx$
$\mathcal{E}$	theory of equality (UIF)	Σ: free function and predicate symbols	$\approx$

Problems:

1. $\mathcal{R} \cup \mathcal{L} \cup \mathcal{E} \models \forall x, y (x \leq y \wedge y \leq x + \text{car}(\text{cons}(0, x)) \wedge P(h(x) - h(y)) \rightarrow P(0))$
2. 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

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

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

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

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

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

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

Step 2: Propagation

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

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

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

$$\underbrace{c \le d \;\wedge\; d \le c + \text{car}(\text{cons}(0, c))}_{}\;$$

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

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

Step 2: Propagation

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

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

Step 2: Propagation

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

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

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.

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

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

$$\phi \;=\; f_1(c_1)\not\approx f_1(c_2) \quad \wedge \quad f_2(c_1)\not\approx f_2(c_3) \quad \wedge \quad f_2(c_2)\not\approx f_2(c_3)$$

$$\phi_1 \;=\; f_1(c_1)\not\approx f_1(c_2) \qquad \phi_2 \;=\; f_2(c_1)\not\approx f_2(c_3) \;\wedge\; f_2(c_2)\not\approx f_2(c_3)$$

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.

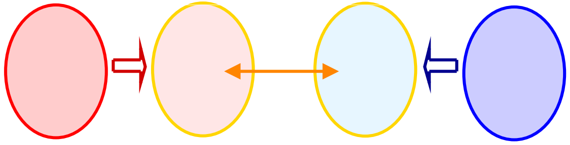
Completeness

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