Decision Procedures in Verification

Decision Procedures (1)

20.12.2021

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de
Until now:

**Syntax** (one-sorted signatures vs. many-sorted signatures)

**Semantics**

**Theories (Syntactic vs. Semantics view)**

**Herbrand models** $\mapsto$ The Bernays-Schönfinkel class

**Algorithmic Problems**

- Decidability/Undecidability
  - Methods: Ordered Resolution with Selection
    - $\mapsto$ Craig Interpolation
    - $\mapsto$ redundancy

**Decidable classes:**
The Bernays-Schönfinkel class, the Ackermann class, the monadic class
3.2 Deduction problems

Satisfiability w.r.t. a theory
Satisfiability w.r.t. a theory

Example

Let $\Sigma = (\{e/0,*/2,i/1\},\emptyset)$

Let $\mathcal{F}$ consist of all (universally quantified) group axioms:

$$\forall x, y, z \quad x \ast (y \ast z) \approx (x \ast y) \ast z$$

$$\forall x \quad x \ast i(x) \approx e \quad \land \quad i(x) \ast x \approx e$$

$$\forall x \quad x \ast e \approx x \quad \land \quad e \ast x \approx x$$

Question: Is $\forall x, y(x \ast y = y \ast x)$ entailed by $\mathcal{F}$?
Satisfiability w.r.t. a theory

Example

Let $\Sigma = (\{e/0, \ast/2, i/1\}, \emptyset)$

Let $\mathcal{F}$ consist of all (universally quantified) group axioms:

$$\forall x, y, z \quad x \ast (y \ast z) \approx (x \ast y) \ast z$$
$$\forall x \quad x \ast i(x) \approx e \land i(x) \ast x \approx e$$
$$\forall x \quad x \ast e \approx x \land e \ast x \approx x$$

**Question:** Is $\forall x, y (x \ast y = y \ast x)$ entailed by $\mathcal{F}$?

**Alternative question:**

Is $\forall x, y (x \ast y = y \ast x)$ true in the class of all groups?
Logical theories

**Syntactic view**

first-order theory: given by a set $\mathcal{F}$ of (closed) first-order $\Sigma$-formulae.

the models of $\mathcal{F}$: $\text{Mod}(\mathcal{F}) = \{ \mathcal{A} \in \Sigma\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \in \mathcal{F} \}$

**Semantic view**

given a class $\mathcal{M}$ of $\Sigma$-algebras

the first-order theory of $\mathcal{M}$: $\text{Th}(\mathcal{M}) = \{ G \in F_{\Sigma}(X) \text{ closed} \mid \mathcal{M} \models G \}$
Decidable theories

Let $\Sigma = (\Omega, \Pi)$ be a signature.

$\mathcal{M}$: class of $\Sigma$-algebras. $\mathcal{T} = \text{Th}(\mathcal{M})$ is decidable

iff

there is an algorithm which, for every closed first-order formula $\phi$, can
decide (after a finite number of steps) whether $\phi$ is in $\mathcal{T}$ or not.

$\mathcal{F}$: class of (closed) first-order formulae.

The theory $\mathcal{T} = \text{Th}(\text{Mod}(\mathcal{F}))$ is decidable

iff

there is an algorithm which, for every closed first-order formula $\phi$, can
decide (in finite time) whether $\mathcal{F} \models \phi$ or not.
Examples

Undecidable theories

- $\text{Th}((\mathbb{Z}, \{0, 1, +, *\}, \{\leq\}))$
- Peano arithmetic
- $\text{Th}(\Sigma\text{-alg})$
Peano arithmetic

Peano axioms: \[\forall x \neg(x + 1 \approx 0)\] (zero)
\[\forall x \forall y (x + 1 \approx y + 1 \rightarrow x \approx y)\] (successor)
\[F[0] \land (\forall x (F[x] \rightarrow F[x + 1]) \rightarrow \forall x F[x])\] (induction)
\[\forall x (x + 0 \approx x)\] (plus zero)
\[\forall x, y (x + (y + 1) \approx (x + y) + 1)\] (plus successor)
\[\forall x, y (x * 0 \approx 0)\] (times 0)
\[\forall x, y (x * (y + 1) \approx x * y + x)\] (times successor)

3 \* y + 5 > 2 \* y expressed as \(\exists z (z \neq 0 \land 3 \* y + 5 \approx 2 \* y + z)\)

Intended interpretation: \((\mathbb{N}, \{0, 1, +, \times\}, \{\approx, \leq\})\)

(Does not capture true arithmetic by Goedel's incompleteness theorem)
Examples

Undecidable theories

- \( \text{Th}(\mathbb{Z}, \{0, 1, +, \times\}, \{\leq\}) \)
- Peano arithmetic
- \( \text{Th}(\Sigma\text{-alg}) \)

Idea of undecidability proof: Suppose there is an algorithm P that, given a formula in one of the theories above decides whether that formula is valid. We use P to give a decision algorithm for the language

\[ \{(G(M), w) | G(M) \text{ is the Gödelisation of a TM } M \text{ that accepts the string } w \} \]

As the latter problem is undecidable, this will show that P cannot exist.
Examples

Undecidable theories

- $\text{Th}(\langle \mathbb{Z}, \{0, 1, +, \ast\}, \{\leq\}\rangle)$
- Peano arithmetic
- $\text{Th}(\Sigma\text{-alg})$

Idea of undecidability proof: (ctd)

1. For $\text{Th}(\langle \mathbb{Z}, \{0, 1, +, \ast\}, \{\leq\}\rangle)$ and Peano arithmetic:
   multiplication can be used for modeling Gödelisation

2. For $\text{Th}(\Sigma\text{-alg})$:
   Given $M$ and $w$, we create a FOL signature and a set of formulae over this signature encoding the way $M$ functions, and a formula which is valid iff $M$ accepts $w$. 
Examples

In order to obtain decidability results:

- Restrict the signature
- Enrich axioms
- Look at certain fragments
Examples

In order to obtain decidability results:

- Restrict the signature
- Enrich axioms
- Look at certain fragments

Decidable theories

- Presburger arithmetic decidable in 3EXPTIME [Presburger’29]
  Signature: (\{0, 1, +\}, \{≈, ≤\}) (no *)
  Axioms \{ (zero), (successor), (induction), (plus zero), (plus successor) \}

- \( \text{Th}(\mathbb{Z}_+) \) \( \mathbb{Z}_+ = (\mathbb{Z}, 0, s, +, ≤) \) the standard interpretation of integers.
Examples

In order to obtain decidability results:

- Restrict the signature
- Enrich axioms
- Look at certain fragments

Decidable theories

- The theory of real numbers (with addition and multiplication) is decidable in 2EXPTIME \([\text{Tarski'30}]\)
Examples

In order to obtain decidability results:

- Restrict the signature
- Enrich axioms
- Look at certain fragments
Problems

\( \mathcal{T} \): first-order theory in signature \( \Sigma \); \( \mathcal{L} \) class of (closed) \( \Sigma \)-formulae

Given \( \phi \) in \( \mathcal{L} \), is it the case that \( \mathcal{T} \models \phi \)?

Common restrictions on \( \mathcal{L} \)

<table>
<thead>
<tr>
<th>( \mathcal{L} )</th>
<th>Pred = ( \emptyset )</th>
<th>( { \phi \in \mathcal{L} \mid \mathcal{T} \models \phi } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall x A(x) \mid A ) atomic</td>
<td>word problem</td>
<td></td>
</tr>
</tbody>
</table>
| \( \forall x (A_1 \land \ldots \land A_n \rightarrow B) \mid A_i, B \) atomic | uniform word problem | \( \text{Th}_{\forall \text{Horn}} \)
| \( \forall x C(x) \mid C(x) \) clause | clausal validity problem | \( \text{Th}_{\forall, \text{cl}} \)
| \( \forall x \phi(x) \mid \phi(x) \) unquantified | universal validity problem | \( \text{Th}_{\forall} \)
| \( \exists x A_1 \land \ldots \land A_n \mid A_i \) atomic | unification problem | \( \text{Th}_{\exists} \)
| \( \forall x \exists x A_1 \land \ldots \land A_n \mid A_i \) atomic | unification with constants | \( \text{Th}_{\forall \exists} \)
$\mathcal{T}$-validity vs. $\mathcal{T}$-satisfiability

**$\mathcal{T}$-validity:** Let $\mathcal{T}$ be a first-order theory in signature $\Sigma$
Let $\mathcal{L}$ be a class of (closed) $\Sigma$-formulae
Given $\phi$ in $\mathcal{L}$, is it the case that $\mathcal{T} \models \phi$?

**Remark:** $\mathcal{T} \models \phi$ iff $\mathcal{T} \cup \neg \phi$ unsatisfiable

Every $\mathcal{T}$-validity problem has a dual $\mathcal{T}$-satisfiability problem:

**$\mathcal{T}$-satisfiability:** Let $\mathcal{T}$ be a first-order theory in signature $\Sigma$
Let $\mathcal{L}$ be a class of (closed) $\Sigma$-formulae
\[ \neg \mathcal{L} = \{ \neg \phi \mid \phi \in \mathcal{L} \} \]
Given $\psi$ in $\neg \mathcal{L}$, is it the case that $\mathcal{T} \cup \psi$ is satisfiable?
### $\mathcal{T}$-validity vs. $\mathcal{T}$-satisfiability

#### Common restrictions on $\mathcal{L} / \neg\mathcal{L}$

<table>
<thead>
<tr>
<th>$\mathcal{L}$</th>
<th>$\neg\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${\forall x A(x) \mid A \text{ atomic}}$</td>
<td>${\exists x \neg A(x) \mid A \text{ atomic}}$</td>
</tr>
<tr>
<td>${\forall x (A_1 \land \ldots \land A_n \rightarrow B) \mid A_i, B \text{ atomic}}$</td>
<td>${\exists x (A_1 \land \ldots \land A_n \land \neg B) \mid A_i, B \text{ atomic}}$</td>
</tr>
<tr>
<td>$\forall x \bigvee L_i \mid L_i \text{ literals}$</td>
<td>$\exists x \bigwedge L_i^\prime \mid L_i^\prime \text{ literals}$</td>
</tr>
<tr>
<td>$\forall x \phi(x) \mid \phi(x) \text{ unquantified}$</td>
<td>$\exists x \phi'(x) \mid \phi'(x) \text{ unquantified}$</td>
</tr>
</tbody>
</table>

**validity problem for universal formulae**  
**ground satisfiability problem**
**$\mathcal{T}$-validity vs. $\mathcal{T}$-satisfiability**

**Common restrictions on $\mathcal{L} / \neg \mathcal{L}$**

<table>
<thead>
<tr>
<th>$\mathcal{L}$</th>
<th>$\neg \mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall x A(x) \mid A$ atomic</td>
<td>$\exists x \neg A(x) \mid A$ atomic</td>
</tr>
<tr>
<td>$\forall x (A_1 \land \ldots \land A_n \rightarrow B) \mid A_i, B$ atomic</td>
<td>$\exists x (A_1 \land \ldots \land A_n \land \neg B) \mid A_i, B$ atomic</td>
</tr>
<tr>
<td>$\forall x \lor L_i \mid L_i$ literals</td>
<td>$\exists x \land L'_i \mid L'_i$ literals</td>
</tr>
<tr>
<td>$\forall x \phi(x) \mid \phi(x)$ unquantified</td>
<td>$\exists x \phi'(x) \mid \phi'(x)$ unquantified</td>
</tr>
</tbody>
</table>

Validity problem for universal formulae  
ground satisfiability problem

In what follows we will focus on the problem of checking the satisfiability of conjunctions of ground literals.
\( \mathcal{T} \)-validity vs. \( \mathcal{T} \)-satisfiability

\[ \mathcal{T} \models \forall x A(x) \quad \text{iff} \quad \mathcal{T} \cup \exists x \neg A(x) \text{ unsatisfiable} \]

\[ \mathcal{T} \models \forall x (A_1 \land \cdots \land A_n \rightarrow B) \quad \text{iff} \quad \mathcal{T} \cup \exists x (A_1 \land \cdots \land A_n \land \neg B) \text{ unsatisfiable} \]

\[ \mathcal{T} \models \forall x (\bigvee_{i=1}^{n} A_i \lor \bigvee_{j=1}^{m} \neg B_j) \quad \text{iff} \quad \mathcal{T} \cup \exists x (\neg A_1 \land \cdots \land \neg A_n \land B_1 \land \cdots \land B_m) \text{ unsatisfiable} \]

\( \mathcal{T} \)-satisfiability vs. Constraint Solving

The field of Constraint Solving also deals with satisfiability problems.

But be careful:

- in Constraint Solving one is interested if a formula is satisfiable in a given, fixed model of \( \mathcal{T} \).

- in \( \mathcal{T} \)-satisfiability one is interested if a formula is satisfiable in any model of \( \mathcal{T} \) at all.
3.3. Theory of Uninterpreted Function Symbols

Why?

- Reasoning about equalities is important in automated reasoning
- Applications to program verification
  (approximation: abstract from additional properties)
Application: Compiler Validation

Example: prove equivalence of source and target program

1: y := 1
2: if z = x*x*x
3: then y := x*x + y
4: endif
5: y := R1+1

To prove: (indexes refer to values at line numbers)

\[
y_1 \approx 1 \land [(z_0 \approx x_0 \ast x_0 \ast x_0 \land y_3 \approx x_0 \ast x_0 + y_1) \lor (z_0 \not\approx x_0 \ast x_0 \ast x_0 \land y_3 \approx y_1)] \land \\
y_1' \approx 1 \land R1_2 \approx x_0' \ast x_0' \land R2_3 \approx R1_2 \ast x_0' \land \\
\land [(z_0' \approx R2_3 \land y_5' \approx R1_2 + 1) \lor (z_0' \not\approx R2_3 \land y_5' \approx y_1')] \land \\
x_0 \approx x_0' \land y_0 \approx y_0' \land z_0 \approx z_0' \implies x_0 \approx x_0' \land y_3 \approx y_5' \land z_0 \approx z_0'
\]
Possibilities for checking it

(1) **Abstraction.**
Consider * to be a “free” function symbol (forget its properties). Test it property can be proved in this approximation. If so, then we know that implication holds also under the normal interpretation of *.

(2) **Reasoning about formulae in fragments of arithmetic.**
Uninterpreted function symbols

Let $\Sigma = (\Omega, \Pi)$ be arbitrary

Let $\mathcal{M} = \Sigma\text{-alg}$ be the class of all $\Sigma$-structures

The theory of uninterpreted function symbols is $\text{Th}(\Sigma\text{-alg})$ the family of all first-order formulae which are true in all $\Sigma$-algebras.

in general undecidable

Decidable fragment:

e.g. the class $\text{Th}_\forall(\Sigma\text{-alg})$ of all universal formulae which are true in all $\Sigma$-algebras.
Uninterpreted function symbols

Assume $\Pi = \emptyset$ (and $\approx$ is the only predicate)

In this case we denote the theory of uninterpreted function symbols by $UIF(\Sigma)$ (or UIF when the signature is clear from the context).

This theory is sometimes called the theory of free functions and denoted Free($\Sigma$)
Theorem 3.3.1

The following are equivalent:

1. testing validity of universal formulae w.r.t. UIF is decidable
2. testing validity of (universally quantified) clauses w.r.t. UIF is decidable

Proof: Follows from the fact that any universal formula is equivalent to a conjunction of (universally quantified) clauses.