Decision Procedures in Verification

Decision Procedures (1)

10.12.2012

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Until now:

First-Order Logic

Syntax, semantics

Algorithmic Problems; Decidability, Undecidability

Methods for checking satisfiability: resolution

Herbrand Interpretations

Assume Ω contains at least one constant symbol.

A Herbrand interpretation (over Σ) is a Σ-algebra $\mathcal{A}$ such that:

- $U_{\mathcal{A}} = \mathrm{T}_\Sigma$ (= the set of ground terms over Σ)
- $f_{\mathcal{A}} : (s_1, \ldots, s_n) \mapsto f(s_1, \ldots, s_n),\ f/n \in \Omega$

A Herbrand interpretation I is called a Herbrand model of F if $I \models F$.

Theorem 2.13
Let N be a set of Σ-clauses.

$$\begin{aligned} N \text{ satisfiable} \quad &\Leftrightarrow \quad N \text{ has a Herbrand model (over } \Sigma) \\ &\Leftrightarrow \quad G_\Sigma(N) \text{ has a Herbrand model (over } \Sigma) \end{aligned}$$

where $G_\Sigma(N) = \{C\sigma \text{ ground clause} \mid C \in N,\ \sigma : X \to \mathrm{T}_\Sigma\}$ is the set of ground instances of N.

The Bernays-Schönfinkel Class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Bernays-Schönfinkel class consists only of sentences of the form

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m F(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Idea: CNF translation:

$$\begin{aligned}
&\exists \overline{x}_1 \forall \overline{y}_1 F_1 \wedge \ldots \exists \overline{x}_n \forall \overline{y}_n F_n \\
&\qquad \Rightarrow_P \exists \overline{x}_1 \ldots \exists \overline{x}_n \forall \overline{y}_1 \ldots \forall \overline{y}_n F(\overline{x}_1, \ldots, \overline{x}_n, \overline{y}_1, \ldots, \overline{y}_n) \\
&\qquad \Rightarrow_S \forall \overline{y}_1 \ldots \forall \overline{y}_m F(\overline{c}_1, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n) \\
&\qquad \Rightarrow_K \forall \overline{y}_1 \ldots \forall \overline{y}_m \bigwedge \bigvee L_i((\overline{c}_1, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n)
\end{aligned}$$

$\overline{c}_1, \ldots, \overline{c}_n$ are tuples of Skolem constants

The Bernays-Schönfinkel Class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Bernays-Schönfinkel class consists only of sentences of the form

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m F(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Idea: CNF translation:

$$\begin{aligned} &\exists \overline{x}_1 \forall \overline{y}_1 F_1 \wedge \ldots \exists \overline{x}_n \forall \overline{y}_n F_n \\ &\qquad\qquad \Rightarrow^*_K \forall \overline{y}_1 \ldots \forall \overline{y}_m \bigwedge \bigvee L_i((\overline{c}_1, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n) \end{aligned}$$

$\overline{c}_1, \ldots, \overline{c}_n$ are tuples of Skolem constants

The Herbrand Universe is finite $\mapsto$ decidability

Tractable fragments of FOL

In the exercise we saw that satisfiability of any finite set of ground Horn clauses can be checked in PTIME (linear time)

Variable-free Horn clauses

Data structures

Atoms $P_1, \ldots, P_n \mapsto \{1, \ldots, n\}$

neg-occ-list(A): list of all clauses in which A occurs negatively

pos-occ-list(A): list of all clauses in which A occurs positively

Clause:	P_1	P_2	$\ldots$	P_n	counter
	neg	neg		pos	↑
		↑			number of literals

first-active-literal: first literal not marked as deleted.

atom status:

atom status:	pos	(deduced as positive unit clause)
	neg	(deduced as negative unit clause)
	nounit	(otherwise)

Variable-free Horn clauses

Input: Set N of Horn formulae

Step 1. Collect unit clauses; check if complementary pairs exist

forall $C \in N$ **do**

if is-unit(C) **then begin** const. time

L := first-active-literal(C) const. time

if state(atom(L)) = nounit **then** state(atom(L)) = sign(L) const. time

push(atom(L), stack)

else if state(atom(L)) $\neq$ sign(L) **then return false**

Variable-free Horn clauses

2. Process the unit clauses in the stack

while stack $\neq \emptyset$ **do**

  begin A := top(stack); pop(stack)

    if state(A) = pos **then** delete-literal-list := neg-oc-list(A) O(# neg-oc-list)

            else delete-literal-list := pos-oc-list(A) O(# pos-oc-list)

    endif

    for all C in delete-literal-list **do**

        if state(A) = pos **then** delete-literal(A,C) const. time + nfal - ofal

        if state(A) = neg **then** delete-literal($\neg$ A,C) const. time + nfal - ofal

        if unit(C) **then** L1 := first-active-literal(C) const. time

            if state(atom(L1)) = nounit **then** state(atom(L1)) = sign(L1),
            L1 $\longrightarrow$ stack

            elseif state(atom(L1)) $\neq$ sign(L1) **then** return false

        endif

  end

Tractable fragments of FOL

We showed that satisfiability of any finite set of ground Horn clauses can be checked in PTIME (linear time)

- Similar fragment of the Bernays-Schönfinkel class?

Motivation: Deductive Databases

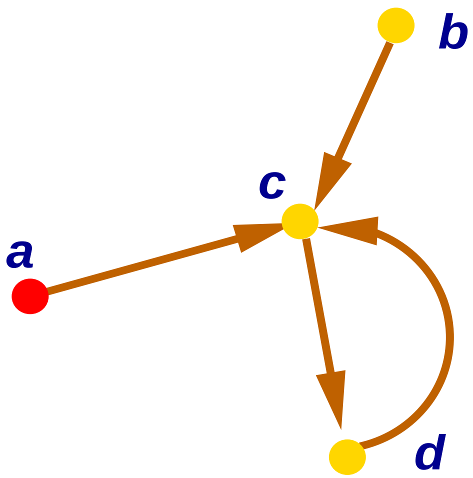
Deductive database

Inference rules:
Facts:
Query:

Motivation: Deductive Databases

Deductive database Example: reachability in graphs

Inference rules:	$\dfrac{S(x)}{R(x)} \qquad \dfrac{R(x) \quad E(x,y)}{R(y)}$
Facts:	$S(a), E(a,c), E(c,d), E(d,c), E(b,c)$
Query:	$R(d)$



$S(a), E(a,c), E(c,d), E(d,c), E(b,c)$

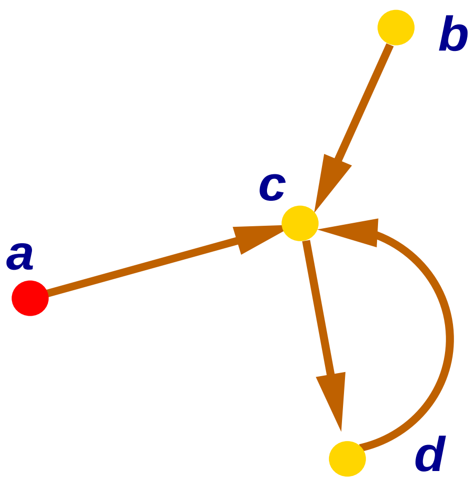
Note: S, E stored relations (Extensional DB)

R defined relation (Intensional DB)

Motivation: Deductive Databases

Deductive database Example: reachability in graphs

Inference rules:	$\dfrac{S(x)}{R(x)} \qquad \dfrac{R(x) \quad E(x,y)}{R(y)}$
Facts:	$S(a), E(a,c), E(c,d), E(d,c), E(b,c)$
Query:	$R(d)$



$S(a), E(a,c), E(a,d), E(c,d), E(b,c),$
$R(a)$

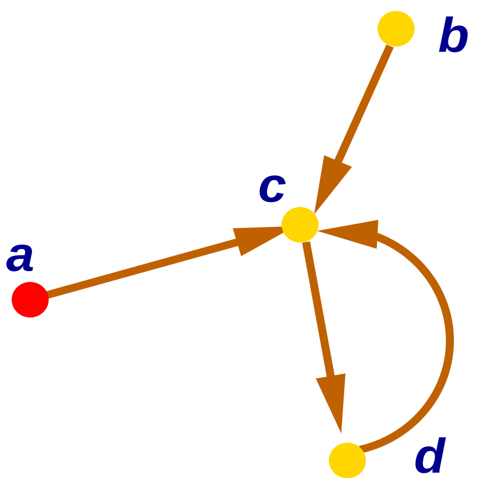
Note: S, E stored relations (Extensional DB)

R defined relation (Intensional DB)

Motivation: Deductive Databases

Deductive database Example: reachability in graphs

Inference rules:	$\dfrac{S(x)}{R(x)} \qquad \dfrac{R(x) \quad E(x,y)}{R(y)}$
Facts:	$S(a), E(a,c), E(c,d), E(d,c), E(b,c)$
Query:	$R(d)$



$S(a), E(a,c), E(a,d), E(c,d), E(b,c),$
$R(a), R(c)$

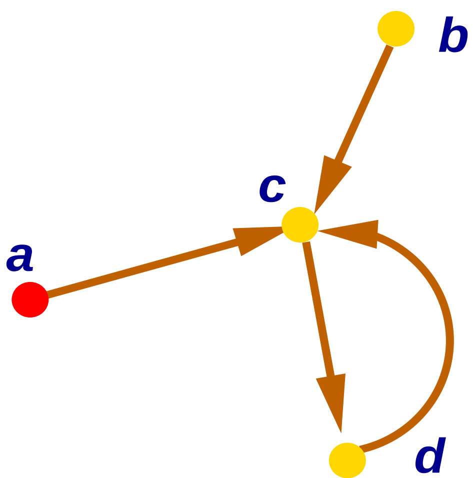
Note: S, E stored relations (Extensional DB)

R defined relation (Intensional DB)

Motivation: Deductive Databases

Deductive database Example: reachability in graphs

Inference rules:	$\dfrac{S(x)}{R(x)} \qquad \dfrac{R(x) \quad E(x,y)}{R(y)}$
Facts:	$S(a), E(a,c), E(c,d), E(d,c), E(b,c)$
Query:	$R(d)$



$S(a), E(a,c), E(a,d), E(c,d), E(b,c),$
$R(a), R(c), R(d)$

Note: S, E stored relations (Extensional DB)

R defined relation (Intensional DB)

Motivation: Deductive Databases

Deductive database $\mapsto$ **Datalog** (Horn clauses, no function symbols)

Inference rules:	$\underbrace{S(x) \rightarrow R(x) \quad R(x) \wedge E(x,y) \rightarrow R(y)}_{\text{set } \mathcal{K} \text{ of Horn clauses}}$
Facts:	$\underbrace{S(a), E(a,c), E(c,d), E(d,c), E(b,c)}_{\text{set } \mathcal{F} \text{ of ground atoms}}$
Query:	$\underbrace{R(d)}_{\text{ground atom } G}$

$$\mathcal{F} \models_{\mathcal{K}} G \quad \text{iff} \quad \mathcal{K} \cup \mathcal{F} \models G \quad \text{iff} \quad \mathcal{K} \cup \mathcal{F} \cup \neg G \models \perp$$

Note: S, E stored relations (Extensional DB)

R defined relation (Intensional DB)

Motivation: Deductive Databases

Deductive database $\mapsto$ **Datalog** (Horn clauses, no function symbols)

Inference rules:	$\underbrace{S(x) \to R(x) \qquad R(x) \wedge E(x,y) \to R(y)}_{\text{set } \mathcal{K} \text{ of Horn clauses}}$
Facts:	$\underbrace{S(a), E(a,c), E(c,d), E(d,c), E(b,c)}_{\text{set } \mathcal{F} \text{ of ground atoms}}$
Query:	$\underbrace{R(d)}_{\text{ground atom } G}$

Ex:

$$\dfrac{\dfrac{\dfrac{S(a) \quad S(x) \to R(x)}{R(a)} \qquad E(a,c) \quad R(x) \wedge E(x,y) \to R(y)}{R(c)} \qquad E(c,d) \quad R(x) \wedge E(x,y) \to R(y)}{R(d)}$$

Ground entailment for function-free Horn clauses

Assumption:

The signature does not contain function symbols of arity ≥ 1.

Given:

- Set H of (function-free) Horn clauses
- Ground Horn clause $G = \bigwedge A_i \rightarrow A$.

The following are equivalent:

(1) $H \models \bigwedge A_i \rightarrow A$
(2) $H \wedge \bigwedge A_i \models A$
(3) $H \wedge \bigwedge A_i \wedge \neg A \models \perp$

Decidable in PTIME in the size of G for a fixed H.

Generalization: Superficial Horn clauses

Assumption:
The signature may contain function symbols of arity ≥ 1.

Definition: A Horn clause is called superficial if it is of the form

$$A_1 \wedge A_2 \cdots \wedge A_n \rightarrow A$$

and every term which occurs in the atom A occurs also in one of the atoms $A_1, A_2, \ldots, A_n$.

Generalization: Superficial Horn clauses

Theorem. Let H be a set of superficial Horn clauses and let C be a ground Horn clause. Then the following are equivalent:

(1) $H \models C$

(2) $H[C] \models C$

where $H[C]$ is the family of all instances of H in which all terms are ground terms occurrring in C or in H.

For every ground clause C, $H \models C$ can be checked in PTIME
(if we assume H is fixed)

Proof: Use ordered resolution with selection.

Generalization: Local theories

[McAllester,Givan'92], [Basin,Ganzinger'96,01], [Ganzinger'01]

Assumption: the signature is allowed to contain function symbols

Definition. H set of Horn clauses is called local iff for every ground clause C the following are equivalent:

(1) $H \models C$

(2) $H[C] \models C$,
where $H[C]$ is the family of all instances of H in which the variables are replaced by ground subterms occurring in H or C.

Theorem. For a fixed local theory H, testing ground entailment w.r.t. H is in PTIME.

Will be discussed in more detail later

Applications

Use ordered resolution with selection to give a decision procedure for the Ackermann class.

The Ackermann class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Ackermann class consists of all sentences of the form

$$\exists x_1 \ldots \exists x_n \forall x \exists y_1 \ldots \exists y_m F(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

Idea: CNF translation:

$$\begin{aligned}
&\exists x_1 \ldots \exists x_n \forall x \exists y_1 \ldots \exists y_m F(x_1, \ldots, x_n, x, y_1, \ldots, y_m) \\
&\qquad \Rightarrow_S \forall x F(\overline{c}_1, \ldots, \overline{c}_n, x, f_1(x), \ldots, f_m(x)) \\
&\qquad \Rightarrow_K \forall x \bigwedge \bigvee L_i(c_1, \ldots, c_n, x, f_1(x), \ldots, f_m(x))
\end{aligned}$$

$c_1, \ldots, c_n$ are Skolem constants

$f_1, \ldots, f_m$ are unary Skolem functions

The Ackermann class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Ackermann class consists of all sentences of the form

$$\exists x_1 \dots \exists x_n \forall x \exists y_1 \dots \exists y_m F(x_1, \dots, x_n, y_1, \dots, y_m)$$

Idea: CNF translation:

$$\exists x_1 \dots \exists x_n \forall x \exists y_1 \dots \exists y_m F(x_1, \dots, x_n, x, y_1, \dots, y_m)$$
$$\Rightarrow^* \forall x \bigwedge \bigvee L_i(c_1, \dots, c_n, x, f_1(x), \dots, f_m(x))$$

The clauses are in the following classes:

$G = G(c_1, \dots, c_n)$ ground clauses without function symbols
$V = V(x, c_1, \dots, c_n)$ clauses with one variable and without function symbols
$G_f = G(c_1, \dots, c_n, f_1, \dots, f_n)$ ground clauses with function symbols
$V_f = V(x, c_1, \dots, c_n, f_1(x), \dots, f_n(x))$ clauses with a variable & function symbols

The Ackermann class

$G = G(c_1, \dots, c_n)$ ground clauses without function symbols
$V = V(x, c_1, \dots, c_n)$ clauses with one variable and without function symbols
$G_f = G(c_1, \dots, c_n, f_1, \dots, f_n)$ ground clauses with function symbols
$V_f = V(x, c_1, \dots, c_n, f_1(x), \dots, f_n(x))$ clauses with a variable & function symbols

Term ordering

$f(t) \succ t$; terms containing function symbols larger than those who do not.

$B \succ A$ iff exists argument u of B such that every argument t of A: $u \succ t$

Ordered resolution: $G \cup V \cup G_f \cup V_f$ is closed under ordered resolution.

$G, G \mapsto G$; $G, V \mapsto G$; $G, G_f \mapsto$ nothing; $G, V_f \mapsto$ nothing

$V, V \mapsto V \cup G$; $V, G_f \mapsto G \cup G_f$; $V, V_f \mapsto G \cup V \cup G_f \cup V_f$

$G_f, G_f \mapsto G_f$; $G_f, V_f \mapsto G_f \cup G$; $V_f, V_f \mapsto G \cup V \cup V_f \cup G_f$

Observation 1: $G \cup V \cup G_f \cup V_f$ finite set of clauses (up to remaming of variables).

The Ackermann class

$G = G(c_1, \ldots, c_n)$ ground clauses without function symbols
$V = V(x, c_1, \ldots, c_n)$ clauses with one variable and without function symbols
$G_f = G(c_1, \ldots, c_n, f_i)$ ground clauses with function symbols
$V_f = V(x, c_1, \ldots, c_n, f_1(x), \ldots, f_n(x))$ clauses with a variable & function symbols

Term ordering

$f(t) \succ t$; terms containing function symbols larger than those who do not.

$B \succ A$ iff exists argument u of B such that every argument t of A: $u \succ t$

Ordered resolution: $G \cup V \cup G_f \cup V_f$ is closed under ordered resolution.

$G, G \mapsto G$; $G, V \mapsto G$; $G, G_f \mapsto$ nothing; $G, V_f \mapsto$ nothing

$V, V \mapsto V \cup G$; $V, G_f \mapsto G \cup G_f$; $V, V_f \mapsto G \cup V \cup G_f \cup V_f$

$G_f, G_f \mapsto G_f$; $G_f, V_f \mapsto G_f \cup G$; $V_f, V_f \mapsto G \cup V \cup V_f \cup G_f$

Observation 2: No clauses with nested function symbols can be generated.

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:

$$
\begin{array}{ll}
\forall x, y, z & x * (y * z) \approx (x * y) * z \\
\forall x & x * i(x) \approx e \quad \wedge \quad i(x) * x \approx e \\
\forall x & x * e \approx x \quad \wedge \quad e * x \approx x
\end{array}
$$

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

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:

$$
\begin{array}{ll}
\forall x, y, z & x * (y * z) \approx (x * y) * z \\
\forall x & x * i(x) \approx e \quad \wedge \quad i(x) * x \approx e \\
\forall x & x * e \approx x \quad \wedge \quad e * x \approx x
\end{array}
$$

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

Alternative question:

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

Logical theories

Syntactic view

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

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

Semantic view

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

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

Decidable theories

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

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

iff

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

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

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

iff

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

Examples

Undecidable theories

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

Peano arithmetic

Peano axioms:

$$\forall x \, \neg(x + 1 \approx 0) \qquad \text{(zero)}$$

$$\forall x \forall y \, (x + 1 \approx y + 1 \rightarrow x \approx y) \qquad \text{(successor)}$$

$$F[0] \wedge (\forall x \, (F[x] \rightarrow F[x + 1]) \rightarrow \forall x F[x]) \qquad \text{(induction)}$$

$$\forall x \, (x + 0 \approx x) \qquad \text{(plus zero)}$$

$$\forall x, y \, (x + (y + 1) \approx (x + y) + 1) \qquad \text{(plus successor)}$$

$$\forall x, y \, (x * 0 \approx 0) \qquad \text{(times 0)}$$

$$\forall x, y \, (x * (y + 1) \approx x * y + x) \qquad \text{(times successor)}$$

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

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

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

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: $(\{0, 1, +\}, \{\approx, \leq\})$ (no $*$)

  Axioms { (zero), (successor), (induction), (plus zero), (plus successor) }

- Th($\mathbb{Z}_+$) $\mathbb{Z}_+ = (\mathbb{Z}, 0, s, +, \leq)$ 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 [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 Σ; $\mathcal{L}$ class of (closed) Σ-formulae

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

Common restrictions on $\mathcal{L}$

	$\mathsf{Pred} = \emptyset$	$\{\phi \in \mathcal{L} \mid \mathcal{T} \models \phi\}$
$\mathcal{L}=\{\forall x A(x) \mid A \text{ atomic}\}$	word problem	
$\mathcal{L}=\{\forall x (A_1 \wedge \ldots \wedge A_n \rightarrow B) \mid A_i, B \text{ atomic}\}$	uniform word problem	$\mathsf{Th}_{\forall\mathsf{Horn}}$
$\mathcal{L}=\{\forall x C(x) \mid C(x) \text{ clause}\}$	clausal validity problem	$\mathsf{Th}_{\forall,\mathsf{cl}}$
$\mathcal{L}=\{\forall x \phi(x) \mid \phi(x) \text{ unquantified}\}$	universal validity problem	$\mathsf{Th}_{\forall}$
$\mathcal{L}=\{\exists x A_1 \wedge \ldots \wedge A_n \mid A_i \text{ atomic}\}$	unification problem	$\mathsf{Th}_{\exists}$
$\mathcal{L}=\{\forall x \exists x A_1 \wedge \ldots \wedge A_n \mid A_i \text{ atomic}\}$	unification with constants	$\mathsf{Th}_{\forall\exists}$

$\mathcal{T}$-validity vs. $\mathcal{T}$-satisfiability

$\mathcal{T}$-validity: Let $\mathcal{T}$ be a first-order theory in signature Σ
Let $\mathcal{L}$ be a class of (closed) Σ-formulae

Given ϕ 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 Σ
Let $\mathcal{L}$ be a class of (closed) Σ-formulae

$$\neg\mathcal{L} = \{\neg\phi \mid \phi \in \mathcal{L}\}$$

Given ψ 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 $\mathcal{L}$ / $\neg\mathcal{L}$

$\mathcal{L}$	$\neg\mathcal{L}$
$\{\forall x A(x) \mid A \text{ atomic}\}$	$\{\exists x \neg A(x) \mid A \text{ atomic}\}$
$\{\forall x(A_1 \wedge \ldots \wedge A_n \rightarrow B) \mid A_i, B \text{ atomic}\}$	$\{\exists x(A_1 \wedge \ldots \wedge A_n \wedge \neg B) \mid A_i, B \text{ atomic}\}$
$\{\forall x \bigvee L_i \mid L_i \text{ literals}\}$	$\{\exists x \bigwedge L'_i \mid L'_i \text{ literals}\}$
$\{\forall x \phi(x) \mid \phi(x) \text{ unquantified}\}$	$\{\exists x \phi'(x) \mid \phi'(x) \text{ unquantified}\}$

validity problem for universal formulae   ground satisfiability problem

$\mathcal{T}$-validity vs. $\mathcal{T}$-satisfiability

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

$\mathcal{L}$	$\neg\mathcal{L}$
$\{\forall x A(x) \mid A \text{ atomic}\}$	$\{\exists x \neg A(x) \mid A \text{ atomic}\}$
$\{\forall x (A_1 \wedge \ldots \wedge A_n \rightarrow B) \mid A_i, B \text{ atomic}\}$	$\{\exists x (A_1 \wedge \ldots \wedge A_n \wedge \neg B) \mid A_i, B \text{ atomic}\}$
$\{\forall x \bigvee L_i \mid L_i \text{ literals}\}$	$\{\exists x \bigwedge L'_i \mid L'_i \text{ literals}\}$
$\{\forall x \phi(x) \mid \phi(x) \text{ unquantified}\}$	$\{\exists x \phi'(x) \mid \phi'(x) \text{ unquantified}\}$

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

$\mathcal{T} \models \forall x A(x)$	iff	$\mathcal{T} \cup \exists x \neg A(x)$ unsatisfiable
$\mathcal{T} \models \forall x (A_1 \wedge \cdots \wedge A_n \rightarrow B)$	iff	$\mathcal{T} \cup \exists x (A_1 \wedge \cdots \wedge A_n \wedge \neg B)$ unsatisfiable
$\mathcal{T} \models \forall x (\bigvee_{i=1}^{n} A_i \vee \bigvee_{j=1}^{m} \neg B_j)$	iff	$\mathcal{T} \cup \exists x (\neg A_1 \wedge \cdots \wedge \neg A_n \wedge B_1 \wedge \cdots \wedge B_m)$ 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
```

```
1: y := 1
2: R1 := x*x
3: R2 := R1*x
4: jmpNE(z,R2,6)
5: y := R1+1
```

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

$$y_1 \approx 1 \wedge [(z_0 \approx x_0 * x_0 * x_0 \wedge y_3 \approx x_0 * x_0 + y_1) \vee (z_0 \not\approx x_0 * x_0 * x_0 \wedge y_3 \approx y_1)] \wedge$$

$$y_1' \approx 1 \wedge R1_2 \approx x_0' * x_0' \wedge R2_3 \approx R1_2 * x_0' \wedge$$

$$\wedge [(z_0' \approx R2_3 \wedge y_5' \approx R1_2 + 1) \vee (z_0' \neq R2_3 \wedge y_5' \approx y_1')] \wedge$$

$$x_0 \approx x_0' \wedge y_0 \approx y_0' \wedge z_0 \approx z_0' \quad \Longrightarrow \quad x_0 \approx x_0' \wedge y_3 \approx y_5' \wedge z_0 \approx z_0'$$

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

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

Uninterpreted function symbols

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.