Automated Deduction for Equational Logic
(SS 2003)

Uwe Waldmann `<uwe@mpi-sb.mpg.de>`

Thomas Hillenbrand `<hillen@mpi-sb.mpg.de>`

Motivation

Equality is the most important relation in mathematics and functional programming.

In principle, problems in first-order logic with equality can be handled by, e.g., resolution theorem provers.

Equality is theoretically difficult:
First-order functional programming is Turing-complete.

But: resolution theorem provers cannot even solve problems that are intuitively easy.

Consequence: to handle equality efficiently, knowledge must be integrated into the theorem prover.

Contents

Syntax and semantics of first-order logic.

Reduction systems and term rewriting
(termination, confluence, critical pairs, etc.).

Termination orderings (e.g., path orderings, polynomial
orderings, Knuth-Bendix ordering).

(Theory) unification.

Superposition calculus.

Theory reasoning (e.g., calculi with built-in asssociative and
commutative operators, transitive relations, Abelian groups).

Implementation issues.

Literature

Slides: `http://www.mpi-sb.mpg.de/~uwe/lehre/eqlogic/`
(Monday morning before the lecture)
Warning: not complete!

Franz Baader and Tobias Nipkow: Term Rewriting and All That, Cambridge Univ. Press, 1998.

Handbook and journal articles
(to be announced).

1 Recapitulation: First-Order Logic

Signatures

Signature: $\Sigma = (\Omega, \Pi)$, where

Ω is a set of function symbols f with arity n
(special case $n = 0$: constant symbol),

Π is a set of predicate symbols p with arity m.

Variables: X is a (usually infinite) set of variable symbols.

Terms

Terms over Σ and X are formed according to this syntax:

$$
\begin{array}{rcll}
s, t, u, v & ::= & x, & \text{where } x \in X \qquad \text{(variable)} \\
& | & f(s_1, \ldots, s_n), & \text{where } f/n \in \Omega \\
& & & \qquad\qquad \text{(functional term)}
\end{array}
$$

The set of all terms over Σ and X is denoted by $\mathrm{T}_\Sigma(X)$.

$\mathrm{T}_\Sigma = \mathrm{T}_\Sigma(\emptyset)$: set of ground terms.

Terms

Variables of a term s:

$$\mathrm{Var}(x) = \{x\},$$

$$\mathrm{Var}(f(s_1, \ldots, s_n)) = \bigcup_{i=1}^{n} \mathrm{Var}(s_i).$$

Terms

Positions of a term s:

$\text{Pos}(x) = \{\varepsilon\}$,

$\text{Pos}(f(s_1, \ldots, s_n)) = \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{\, i\,p \mid p \in \text{Pos}(s_i)\,\}$.

Size of a term s:

$|s| =$ cardinality of $\text{Pos}(s)$.

Prefix order for $p, q \in \text{Pos}(s)$:

p above q: $p \leq q$ if $p\,p' = q$ for some p',

p strictly above q: $p < q$ if $p \leq q$ and not $q \leq p$,

p and q parallel: $p \parallel q$ if neither $p \leq q$ nor $q \leq p$.

Terms

Subterm of s at a position $p \in \mathrm{Pos}(s)$:

$$s/\varepsilon = s,$$

$$f(s_1, \dots, s_n)/i\,p = s_i/p.$$

Replacement of the subterm at position $p \in \mathrm{Pos}(s)$ by t:

$$s[t]_\varepsilon = t,$$

$$f(s_1, \dots, s_n)[t]_{ip} = f(s_1, \dots, s_i[t]_p, \dots, s_n).$$

Formulas

Atoms (atomic formulas) over Σ and X are formed according to this syntax:

$$
\begin{array}{lcll}
A, B & ::= & p(s_1, \dots, s_m), & \text{where } p/m \in \Pi \\
& & & \text{(non-equational atom)} \\
& \mid & s \approx t, & \text{(equation)}
\end{array}
$$

It is sufficient to consider the case that $\Pi = \emptyset$, i. e., only equations, no other atoms.

Formulas

Literals over Σ and X are formed according to this syntax:

$$
\begin{array}{rcll}
L & ::= & A, & \text{(positive literal)} \\
 & \mid & \neg A, & \text{(negative literal)}
\end{array}
$$

Abbreviation: $s \not\approx t$ instead of $\neg\, s \approx t$.

Formulas

First-order formulas over Σ are formed according to this syntax:

$$
\begin{array}{rcll}
F, G, H & ::= & \bot & \text{(false)} \\
& \mid & \top & \text{(true)} \\
& \mid & A, & \text{(atomic formula)} \\
& \mid & \neg F & \text{(negation)} \\
& \mid & (F \wedge G) & \text{(conjunction)} \\
& \mid & (F \vee G) & \text{(disjunction)} \\
& \mid & (F \rightarrow G) & \text{(implication)} \\
& \mid & (F \leftrightarrow G) & \text{(equivalence)} \\
& \mid & \forall x\, F & \text{(universal quantification)} \\
& \mid & \exists x\, F & \text{(existential quantification)}
\end{array}
$$

Formulas

Bound and free variables:

In $Qx\,F$ with $Q \in \{\exists, \forall\}$, we call F the scope of the quantifier Qx.

An occurrence of a variable x is called bound, if it is inside the scope of a quantifier Qx. Any other occurrence of a variable is called free.

Formulas without free variables are also called closed formulas.

Formulas without variables are called ground.

Formulas

Clauses over Σ and X are formed according to this syntax:

$$
\begin{array}{rcll}
C, D & ::= & \bot & \text{(empty clause)} \\
& \mid & L_1 \vee \cdots \vee L_k, \quad \text{where } k \geq 1 & \\
& & & \text{(non-empty clause)}
\end{array}
$$

Convention: All variables in a clause are implicitly universally quantified.

Usually in this lecture (w.o.l.o.g.): Clauses instead of general formulas.

Substitutions

A substitution is a mapping $\sigma : X \to \mathrm{T}_\Sigma(X)$ such that the domain of σ, that is, the set $\mathrm{Dom}(\sigma) = \{\, x \in X \mid \sigma(x) \neq x \,\}$ is finite.

Notation: $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$.

$\mathrm{Ran}(\sigma) = \{\, \sigma(x) \mid x \in \mathrm{Dom}(\sigma) \,\}$
$\mathrm{Codom}(\sigma) = \mathrm{Var}(\mathrm{Ran}(\sigma))$.

Usually: postfix notation $x\sigma = \sigma(x)$.

Substitutions

Substitutions are extended homomorphically to terms and formulas:

$$f(s_1, \ldots, s_n)\sigma = f(s_1\sigma, \ldots, s_n\sigma)$$

$$\begin{aligned}
\bot\sigma &= \bot \\
\top\sigma &= \top \\
p(s_1, \ldots, s_n)\sigma &= p(s_1\sigma, \ldots, s_n\sigma) \\
(u \approx v)\sigma &= (u\sigma \approx v\sigma) \\
\neg F\sigma &= \neg(F\sigma) \\
(F \,\rho\, G)\sigma &= (F\sigma \,\rho\, G\sigma), \quad \text{for each binary connective } \rho \\
(Qx\, F)\sigma &= Qz\,(F\,\sigma[x \mapsto z]), \quad \text{with } z \text{ a fresh variable}
\end{aligned}$$

where $x\sigma[x \mapsto t] = t$ and $y\sigma[x \mapsto t] = y\sigma$ for $y \neq x$.

Substitutions

If $t = s\sigma$ for some substitution, then t is called an instance of s.

(Analogously for atoms, literals, ...)

Semantics

A Σ-algebra (or Σ-interpretation) is a triple

$$\mathcal{A} = (U_{\mathcal{A}},\ (f_{\mathcal{A}} : U^n \to U)_{f/n\in\Omega},\ (p_{\mathcal{A}} \subseteq U^m)_{p/m\in\Pi})$$

where $U_{\mathcal{A}} \neq \emptyset$ is a set, called the universe of $\mathcal{A}$.

If $\Pi = \emptyset$, we will omit the third component.

We will usually use the symbol $\mathcal{A}$ to denote both the algebra and its universe.

Semantics

Special case: term algebras:

$U_{\mathcal{A}} = \mathsf{T}_\Sigma(Y)$ for some (possibly empty) set Y of variables,

$f_{\mathcal{A}} : (t_1, \dots, t_n) \mapsto f(t_1, \dots, t_n)$.

Semantics

An assignment is a mapping $\alpha : X \to \mathcal{A}$.

An assignment α can be homomorphically extended to a function $\mathcal{A}(\alpha) : \mathsf{T}_\Sigma(X) \to \mathcal{A}$:

$$\begin{aligned}
\mathcal{A}(\alpha)(x) &= \alpha(x), \;\text{ for } x \in X \\
\mathcal{A}(\alpha)(f(s_1, \ldots, s_n)) &= f_{\mathcal{A}}(\mathcal{A}(\alpha)(s_1), \ldots, \mathcal{A}(\alpha)(s_n)), \\
&\qquad \text{for } f/n \in \Omega.
\end{aligned}$$

Semantics

The set of truth values is $\{0, 1\}$. The truth value $\mathcal{A}(\alpha)(F)$ of a formula F in $\mathcal{A}$ with respect to α is defined inductively:

$$
\begin{aligned}
\mathcal{A}(\alpha)(\bot) &= 0 \\
\mathcal{A}(\alpha)(\top) &= 1 \\
\mathcal{A}(\alpha)(p(s_1, \ldots, s_n)) &= 1 \text{ iff } (\mathcal{A}(\alpha)(s_1), \ldots, \mathcal{A}(\alpha)(s_n)) \in p_{\mathcal{A}} \\
\mathcal{A}(\alpha)(s \approx t) &= 1 \text{ iff } \mathcal{A}(\alpha)(s) = \mathcal{A}(\alpha)(t) \\
\mathcal{A}(\alpha)(\neg F) &= 1 \text{ iff } \mathcal{A}(\alpha)(F) = 0 \\
\mathcal{A}(\alpha)(F \rho G) &= \mathrm{B}_\rho(\mathcal{A}(\alpha)(F), \mathcal{A}(\alpha)(G)), \\
&\quad \text{where } \mathrm{B}_\rho \text{ is the Boolean function associated with } \rho \\
\mathcal{A}(\alpha)(\forall x F) &= 1 \text{ iff } \mathcal{A}(\alpha[x \mapsto a])(F) = 1 \text{ for all } a \in U_{\mathcal{A}} \\
\mathcal{A}(\alpha)(\exists x F) &= 1 \text{ iff } \mathcal{A}(\alpha[x \mapsto a])(F) = 1 \text{ for some } a \in U_{\mathcal{A}}
\end{aligned}
$$

where $\alpha[x \mapsto a](x) = a$ and $\alpha[x \mapsto a](y) = \alpha(y)$ for $y \neq x$.

Validity and Satisfiability

F is valid in a Σ-algebra $\mathcal{A}$ under assignment α:

$\mathcal{A}, \alpha \models F$ iff $\mathcal{A}(\alpha)(F) = 1$.

F is valid in a Σ-algebra $\mathcal{A}$ (or: $\mathcal{A}$ is a model of F):

$\mathcal{A} \models F$ iff $\mathcal{A}, \alpha \models F$ for all $\alpha \in X \to \mathcal{A}$.

F is (universally) valid (or: F is a tautology):

$\models F$ iff $\mathcal{A} \models F$ for all Σ-algebras $\mathcal{A}$.

F is satisfiable iff there exist $\mathcal{A}$ and α such that $\mathcal{A}, \alpha \models F$.
Otherwise F is called unsatisfiable.

Entailment and Equivalence

We will use the notion of entailment only for closed formulas.

Let F and G be closed formulas.

F entails G:

$F \models G$ iff for all Σ-algebras $\mathcal{A}$, $\mathcal{A} \models F$ implies $\mathcal{A} \models G$.

A set of closed formulas N entails G:

$N \models G$ iff for all Σ-algebras $\mathcal{A}$, if $\mathcal{A} \models F$ for all $F \in N$, then $\mathcal{A} \models G$.

F and G are equivalent iff for all Σ-algebras $\mathcal{A}$, $\mathcal{A} \models F \Leftrightarrow \mathcal{A} \models G$. (analogously for sets of closed formulas).

Semantics

Proposition:
F is valid if and only if $\neg F$ is unsatisfiable.

Proposition:
$N \models F$ if and only if $N \cup \{\neg F\}$ is unsatisfiable.