

(iii) the indicated occurrences of the $\neg A_i$ are exactly the ones selected by S , or else nothing is selected in the right premise and $n = 1$ and $\neg A_1\sigma$ is maximal in $C\sigma$.

Similarly to resolution, hyperresolution has to be complemented by a factoring inference.

As we have seen, hyperresolution can be simulated by iterated binary resolution.

However this yields intermediate clauses which HR might not derive, and many of them might not be extendable into a full HR inference.

3.14 Summary: Resolution Theorem Proving

- Resolution is a machine calculus.
- Subtle interleaving of enumerating ground instances and proving inconsistency through the use of unification.
- Parameters: atom ordering \succ and selection function S . On the non-ground level, ordering constraints can (only) be solved approximatively.
- Completeness proof by constructing candidate interpretations from productive clauses $C \vee A$, $A \succ C$; inferences with those reduce counterexamples.
- *Local* restrictions of inferences via \succ and S
 \Rightarrow fewer proof variants.
- *Global* restrictions of the search space via elimination of redundancy
 \Rightarrow computing with “smaller” clause sets;
 \Rightarrow termination on many decidable fragments.
- However: not good enough for dealing with orderings, equality and more specific algebraic theories (lattices, abelian groups, rings, fields)
 \Rightarrow further specialization of inference systems required.

3.15 Other Inference Systems

Instantiation-based methods for FOL:

- (Semantic) Tableau;
- Resolution-based instance generation;
- Disconnection calculus.

Further (mainly propositional) proof systems:

- Hilbert calculus;
- Sequent calculus;
- Natural deduction.

Instantiation-Based Methods for FOL

Idea:

Overlaps of complementary literals produce instantiations (as in resolution);

However, contrary to resolution, clauses are not recombined.

Instead: treat remaining variables as constant and use efficient propositional proof methods, such as DPLL.

There are both saturation-based variants, such as partial instantiation [Hooker et al.] or resolution-based instance generation (Inst-Gen) [Ganzinger and Korovin], and tableau-style variants, such as the disconnection calculus [Billon; Letz and Stenz].

Hilbert Calculus

Hilbert calculus:

Direct proof method (proves a theorem from axioms, rather than refuting its negation)

Axiom schemes, e. g.,

$$(F \rightarrow (G \rightarrow F)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$$

plus Modus ponens:

$$\frac{F \quad F \rightarrow G}{G}$$

Unsuitable for both humans and machines.

Natural Deduction

Natural deduction (Prawitz):

Models the concept of proofs from assumptions as humans do it (cf. Fitting or Huth/Ryan).

Sequent Calculus

Sequent calculus (Gentzen):

Assumptions internalized into the data structure of sequents

$$F_1, \dots, F_m \rightarrow G_1, \dots, G_k$$

meaning

$$F_1 \wedge \dots \wedge F_m \rightarrow G_1 \vee \dots \vee G_k$$

A kind of mixture between natural deduction and semantic tableaux.

Perfect symmetry between the handling of assumptions and their consequences.

Can be used both backwards and forwards.

4 First-Order Logic with Equality

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.

4.1 Handling Equality Naively

Proposition 4.1 *Let F be a closed first-order formula with equality. Let $\sim \notin \Pi$ be a new predicate symbol. The set $Eq(\Sigma)$ contains the formulas*

$$\begin{aligned} & \forall x (x \sim x) \\ & \forall x, y (x \sim y \rightarrow y \sim x) \\ & \forall x, y, z (x \sim y \wedge y \sim z \rightarrow x \sim z) \\ & \forall \vec{x}, \vec{y} (x_1 \sim y_1 \wedge \dots \wedge x_n \sim y_n \rightarrow f(x_1, \dots, x_n) \sim f(y_1, \dots, y_n)) \\ & \forall \vec{x}, \vec{y} (x_1 \sim y_1 \wedge \dots \wedge x_m \sim y_m \wedge p(x_1, \dots, x_m) \rightarrow p(y_1, \dots, y_m)) \end{aligned}$$

for every $f \in \Omega$ and $p \in \Pi$. Let \tilde{F} be the formula that one obtains from F if every occurrence of \approx is replaced by \sim . Then F is satisfiable if and only if $Eq(\Sigma) \cup \{\tilde{F}\}$ is satisfiable.

Proof. Let $\Sigma = (\Omega, \Pi)$, let $\Sigma_1 = (\Omega, \Pi \cup \{\sim\})$.

For the “only if” part assume that F is satisfiable and let \mathcal{A} be a Σ -model of F . Then we define a Σ_1 -algebra \mathcal{B} in such a way that \mathcal{B} and \mathcal{A} have the same universe, $f_{\mathcal{B}} = f_{\mathcal{A}}$ for every $f \in \Omega$, $p_{\mathcal{B}} = p_{\mathcal{A}}$ for every $p \in \Pi$, and $\sim_{\mathcal{B}}$ is the identity relation on the universe. It is easy to check that \mathcal{B} is a model of both \tilde{F} and of $Eq(\Sigma)$.

The proof of the “if” part consists of two steps.

Assume that the Σ_1 -algebra $\mathcal{B} = (U_{\mathcal{B}}, (f_{\mathcal{B}} : U^n \rightarrow U)_{f \in \Omega}, (p_{\mathcal{B}} \subseteq U_{\mathcal{B}}^m)_{p \in \Pi \cup \{\sim\}})$ is a model of $Eq(\Sigma) \cup \{\tilde{F}\}$. In the first step, we can show that the interpretation $\sim_{\mathcal{B}}$ of \sim in \mathcal{B} is a congruence relation. We will prove this for the symmetry property, the other properties of congruence relations, that is, reflexivity, transitivity, and congruence with respect to functions and predicates are shown analogously. Let $a, a' \in U_{\mathcal{B}}$ such that $a \sim_{\mathcal{B}} a'$. We have to show that $a' \sim_{\mathcal{B}} a$. Since \mathcal{B} is a model of $Eq(\Sigma)$, $\mathcal{B}(\beta)(\forall x, y (x \sim y \rightarrow y \sim x)) = 1$ for every β , hence $\mathcal{B}(\beta[x \mapsto b_1, y \mapsto b_2])(x \sim y \rightarrow y \sim x) = 1$ for every β and every

$b_1, b_2 \in U_{\mathcal{B}}$. Set $b_1 = a$ and $b_2 = a'$, then $1 = \mathcal{B}(\beta[x \mapsto a, y \mapsto a'])(x \sim y \rightarrow y \sim x) = (a \sim_{\mathcal{B}} a' \rightarrow a' \sim_{\mathcal{B}} a)$, and since $a \sim_{\mathcal{B}} a'$ holds by assumption, $a' \sim_{\mathcal{B}} a$ must also hold.

In the second step, we will now construct a Σ -algebra \mathcal{A} from \mathcal{B} and the congruence relation $\sim_{\mathcal{B}}$. Let $[a]$ be the congruence class of an element $a \in U_{\mathcal{B}}$ with respect to $\sim_{\mathcal{B}}$. The universe $U_{\mathcal{A}}$ of \mathcal{A} is the set $\{[a] \mid a \in U_{\mathcal{B}}\}$ of congruence classes of the universe of \mathcal{B} . For a function symbol $f \in \Omega$, we define $f_{\mathcal{A}}([a_1], \dots, [a_n]) = [f_{\mathcal{B}}(a_1, \dots, a_n)]$, and for a predicate symbol $p \in \Pi$, we define $([a_1], \dots, [a_n]) \in p_{\mathcal{A}}$ if and only if $(a_1, \dots, a_n) \in p_{\mathcal{B}}$. Observe that this is well-defined: If we take different representatives of the same congruence class, we get the same result by congruence of $\sim_{\mathcal{B}}$. Now for every Σ -term t and every \mathcal{B} -assignment β , $[\mathcal{B}(\beta)(t)] = \mathcal{A}(\gamma)(t)$, where γ is the \mathcal{A} -assignment that maps every variable x to $[\beta(x)]$, and analogously for every Σ -formula G , $\mathcal{B}(\beta)(\tilde{G}) = \mathcal{A}(\gamma)(G)$. Both properties can easily be shown by structural induction. Consequently, \mathcal{A} is a model of F . \square

By giving the equality axioms explicitly, first-order problems with equality can in principle be solved by a standard resolution or tableaux prover.

But this is unfortunately not efficient (mainly due to the transitivity and congruence axioms).

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):
 - Term rewrite systems.
 - Expressing semantic consequence syntactically.
 - Entailment for equations.
- Equational clauses:
 - Entailment for clauses with equality.

4.2 Abstract Reduction Systems

Abstract reduction system: (A, \rightarrow) , where

A is a set,

$\rightarrow \subseteq A \times A$ is a binary relation on A .

\rightarrow^0	$= \{ (a, a) \mid a \in A \}$	<i>identity</i>
\rightarrow^{i+1}	$= \rightarrow^i \circ \rightarrow$	<i>i + 1-fold composition</i>
\rightarrow^+	$= \bigcup_{i>0} \rightarrow^i$	<i>transitive closure</i>
\rightarrow^*	$= \bigcup_{i \geq 0} \rightarrow^i = \rightarrow^+ \cup \rightarrow^0$	<i>reflexive transitive closure</i>
$\rightarrow^=$	$= \rightarrow \cup \rightarrow^0$	<i>reflexive closure</i>
\rightarrow^{-1}	$= \leftarrow = \{ (b, c) \mid c \rightarrow b \}$	<i>inverse</i>
\leftrightarrow	$= \rightarrow \cup \leftarrow$	<i>symmetric closure</i>
\leftrightarrow^+	$= (\leftrightarrow)^+$	<i>transitive symmetric closure</i>
\leftrightarrow^*	$= (\leftrightarrow)^*$	<i>refl. trans. symmetric closure</i>

$b \in A$ is *reducible*, if there is a c such that $b \rightarrow c$.

b is *in normal form (irreducible)*, if it is not reducible.

c is a *normal form of b* , if $b \rightarrow^* c$ and c is in normal form.

Notation: $c = b \downarrow$ (if the normal form of b is unique).

b and c are *joinable*, if there is a a such that $b \rightarrow^* a \leftarrow^* c$.

Notation: $b \downarrow c$.

A relation \rightarrow is called

Church-Rosser, if $b \leftrightarrow^* c$ implies $b \downarrow c$.

confluent, if $b \leftarrow^* a \rightarrow^* c$ implies $b \downarrow c$.

locally confluent, if $b \leftarrow a \rightarrow c$ implies $b \downarrow c$.

terminating, if there is no infinite descending chain $b_0 \rightarrow b_1 \rightarrow b_2 \rightarrow \dots$.

normalizing, if every $b \in A$ has a normal form.

convergent, if it is confluent and terminating.

Lemma 4.2 *If \rightarrow is terminating, then it is normalizing.*

Note: The reverse implication does not hold.

Theorem 4.3 *The following properties are equivalent:*

(i) \rightarrow has the Church-Rosser property.

(ii) \rightarrow is confluent.

Proof. (i) \Rightarrow (ii): trivial.

(ii) \Rightarrow (i): by induction on the number of peaks in the derivation $b \leftrightarrow^* c$.

□