A Saturation-Based Automated Theorem Prover for RISCAL

Viktoria Langenreither

28.03.2023

Goals of this Thesis and Presentation

- extension of RISCTP/RISCAL by a saturation-based automated theorem prover for first-order logic with equality
- the theoretical basis for such a prover and the support for special theories (integer and arrays)
- implementation of the prover
- experiments and tests with the prover

- overview and theoretic understanding of the important proving techniques needed for the prover

Preliminaries

- classical first-order predicate logic with equality
- all standard logical connectives ($\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$) and quantifiers ($\forall, \exists$)
- propositional constants $\top$ (always true) and $\bot$ (always false)
- in addition the binary predicate symbol $=$

Theorem (Herbrand's theorem)

A quantifier-free formula p is first-order satisfiable if and only if the set of all its ground instances is (propositionally) satisfiable.

Preliminaries

Definition

Let $\mathcal{I}$ be an inference system and $\mathcal{S}$ a set of formulas in $\mathcal{I}$. If $\mathcal{S}$ is unsatisfiable, then the empty clause $\square$ is derivable from $\mathcal{S}$ in $\mathcal{I}$. $\mathcal{I}$ is then called refutation complete (sometimes also refutationally complete) .

Definition

A selection function is a mapping, which selects in every clause C a non-empty subset of literals (or equations).

Preliminaries

Definition

An ordering $\succ$ on terms is denoted as a simplification ordering on terms if these statements are fulfilled:

- $\succ$ *is well-founded, i.e., there is no infinite sequence of terms $s, t, \ldots$ such that $s \succ t \succ \ldots$*
- $\succ$ *is monotonic, i.e., if $s \succ t$, then $r[s] \succ r[t]$ for all terms r, s, t*
- $\succ$ *is stable under substitutions, i.e., if $s \succ t$, then also $s\sigma \succ t\sigma$ for all terms s, t and all substitutions σ*
- $\succ$ *has the subterm property, i.e., if s is a subterm of t and $s \neq t$, then $t \succ s$*

Resolution

One version of resolution combines the two inference rules "(binary) resolution" and "factoring" to a single rule:

$$\frac{C \vee A \vee \ldots \vee A \qquad \neg A \vee D}{C \vee D}$$

Remark

Remember first-order resolution uses unification. But a MGU might be too general. This means we have to additionally consider unifying some subset of the literals in the same clause (Lifting Lemma).

Refutation Completeness

Theorem

Let S be a set of first-order clauses. If S is unsatisfiable, then the empty clause $\square$ is derivable by resolution.

Proof.

Let S be a set of first-order clauses. Then by Herbrand's theorem and compactness, there is a finite set of ground instances of clauses in S that is unsatisfiable. By the refutation completeness of the propositional resolution the empty clause is derivable by resolution. Let C' be an instance of C. Using induction on the structure, it is possible to apply the lifting lemma to show that for each subproof of a clause C' there exists a corresponding proof using first-order resolution of a clause C. To finally conclude the empty clause, the empty clause has to be derivable by first-order resolution, because the empty clause cannot be an instance of a non-empty clause. $\square$

Resolution

- selection functions and simplification orderings as control mechanisms
- restrict resolution on the maximal atoms in the side premise only
- sufficient to only resolve on negative literals in a non-deterministic way
- by simultaneously resolving on more than one atom we achieve a larger inference step
- still refutation complete

$$\frac{C_1 \vee A_1 \vee \ldots \vee A_1 \qquad \ldots \qquad C_n \vee A_n \vee \ldots \vee A_n \qquad \neg A_1 \vee \ldots \vee \neg A_n \vee D}{C_1 \vee \ldots \vee C_n \vee D}$$

Paramodulation

Let's consider the paramodulation inference rule for variable free formulas:

$$\frac{\Gamma_1 \rightarrow \Delta_1, s = t \qquad \Gamma_2 \rightarrow \Delta_2, u[s] = v}{\Gamma_1, \Gamma_2 \rightarrow \Delta_1, \Delta_2, u[t] = v}$$

Remark

To restrict the number of inferences computed during the proof search term orderings are an important tool. Assume $\succ$ *is an ordering which is total on variable-free terms and formulas. The basic idea of ordered paramodulation is to only replace big terms by smaller ones according to* $\succ$*; same idea as in ordered rewriting.*

Paramodulation

Definition

A paramodulation inference is called ordered (with respect to $\succ$) if the following conditions are fulfilled:

1. $s \succ t$
2. $s = t$ *is strictly maximal with respect to* $\Gamma_1 \cup \Delta_1$
3. $u[s] = v$ *is strictly maximal with respect to* $\Gamma_2 \cup \Delta_2$

Superposition

Definition

An ordered paramodulation inference is called a superposition inference if the additional condition is satisfied:

4. $u[s] \succ v$

If s does not occur in Γ_1 the superposition inference is called strict. A weak superposition inference denoted a paramodulation inference for which the conditions (1), (3) and (4) hold (but (2) is not necessary).

Superposition

We define the following inference rules with respect to the reduction ordering $\succ$:

$$\frac{\Gamma, u = v \rightarrow \Delta}{\Gamma\sigma \rightarrow \Delta\sigma} \text{ (Equality resolution)}$$

where σ is a MGU of u and v and $u\sigma = v\sigma$ is a maximal equation in $\Gamma\sigma, u\sigma = v\sigma \rightarrow \Delta\sigma$.

$$\frac{\Gamma \rightarrow \Delta, A, B}{\Gamma\sigma \rightarrow \Delta\sigma, A\sigma} \text{ (Ordered factoring)}$$

where σ is a MGU of A and B and $A\sigma$ is a maximal equation in $\Gamma\sigma \rightarrow \Delta\sigma, A\sigma, B\sigma$.

Superposition

$$\frac{\Gamma_1 \rightarrow \Delta_1, s = t \qquad \Gamma_2, u[s'] = v \rightarrow \Delta_2}{\Gamma_1\sigma, \Gamma_2\sigma, u[t]\sigma = v\sigma \rightarrow \Delta_1\sigma, \Delta_2\sigma} \text{(Superpos. left)}$$

where σ is a MGU of s and s'; $\Gamma_1\sigma \rightarrow \Delta_1\sigma, s\sigma = t\sigma$ is a reductive clause for $s\sigma = t\sigma$; $v\sigma \prec u\sigma$ and $u\sigma = v\sigma$ is a maximal equation in $u\sigma = v\sigma, \Gamma_2\sigma \rightarrow \Delta_2\sigma$; and s' is not a variable.

$$\frac{\Gamma_1 \rightarrow \Delta_1, s = t \qquad \Gamma_2 \rightarrow \Delta_2, u[s'] = v}{\Gamma_1\sigma, \Gamma_2\sigma \rightarrow \Delta_1\sigma, \Delta_2\sigma, u[t]\sigma = v\sigma} \text{(Superpos. right)}$$

where σ is a MGU of s and s'; $\Gamma_1\sigma \rightarrow \Delta_1\sigma, s\sigma = t\sigma$ is a reductive clause for $s\sigma = t\sigma$; $\Gamma_2\sigma \rightarrow u\sigma = v\sigma, \Delta_2$ is a reductive clause for $u\sigma = v\sigma$; and s' is not a variable.

Superposition

To achieve refutation completeness the inference system gets combined with additional rules. We could for example add this paramodulation inference rule called Merging Paramodulation:

$$\frac{\Gamma_1 \rightarrow \Delta_1, s = t \qquad \Gamma_2 \rightarrow \Delta_2, u = v[s'], u' = v'}{\Gamma_1\sigma, \Gamma_2 \rightarrow \Delta_1\sigma, \Delta_2\sigma, u\sigma = v[t]\sigma, u\sigma = v'\sigma}$$

where $\sigma = \tau\rho$ is the composition of a MGU τ of s and s' and a MGU ρ of $u\tau$ and $u'\tau$; $\Gamma_1\sigma \rightarrow \Delta_1\sigma, s\sigma = t\sigma$ is a reductive clause for $s\sigma = t\sigma$; $\Gamma_2\sigma \rightarrow \Delta_2\sigma, u\sigma = v\sigma, u'\sigma = v'\sigma$ is a reductive clause for $u\sigma = v\sigma$; $u\tau \succ v\tau$ and $v'\sigma \prec v\sigma$; and s' is not a variable.

Refutation Completeness

- model generation method based on strict superposition i.e. the paramodulation involves only maximal terms of maximal equations of clauses
- introduce the concept on ground Horn clauses with equality only
- inference system $\mathcal{I}$: Superposition right, superposition left, equality resolution

Proof Idea

- Let S be a set of ground Horn clauses closed under the inference system $\mathcal{I}$
- If $\square \notin S$, then S is satisfiable
- An equality Herbrand interpretation will be constructed, then we can show that this interpretation is a model of S
- We construct the interpretation by a congruence R^* created by a set of ground rewrite rules R
- Every of these rules has been generated by some clause of S
- This generation process is formally defined by induction on the ordering $\succ_C$
- Every clause C in S either generates a rule or not.
- This depends on the set R_C of rules generated by clauses D in S where $C \succ_C D$

Further Work

What we have done so far:

- State of the art
- Throughout theoretical representation of the concepts needed for the prover

What we are doing now:

- Collecting strategies to make those concepts reasonably efficient
- Designing the prover
- Start with the implementation of the prover

References

- John Harrison. Handbook of Practical Logic and Automated Reasoning. Cambridge, UK: Cambridge University Press, 2009. doi: 10.1017/CBO9780511576430.
- Melvin Fitting. First-order logic and automated theorem proving. Texts in Computer Science. New York, NY, USA: Springer, 1997. doi: 10.1007/978-1-4612-2360-3.
- Laura Kovacs and Andrei Voronkov. "First-Order Theorem Proving and VAMPIRE". In: Computer Aided Verification. Springer, Berlin, Heidelberg, 2013, pp. 1–35. doi: 10.1007/978-3-642-39799-8_1
- Leo Bachmair and Harald Ganzinger. Rewrite-Based Equational Theorem Proving With Selection and Simplification. Department of Computer Science , SUNY at Stony Brook, NY, USA and Max-Planck-Institut für Informatik, Im Stadtwald, Saarbrücken, Germany, Sept. 1991. doi: 10.1093/logcom/4.3.217. url: https://pure.mpg.de/rest/items/item_1834970_5/component/file_1857487/content