Redundancy for Geometric Resolution

Hans de Nivelle

University of Wrocław, Poland

Oslo, 13.06.2008

Introduction

Geometric Resolution is a sound and complete calculus for first-order logic with equality.

We called the calculus geometric resolution, because it operates on a normal form, which is derived from geometric formulas. (this is a first-order fragment introduced by Thoralf Skolem)

- The calculus is sound and complete for first-order logic.
- It has been implemented. The system is called **geo**. It can be downloaded from my homepage.

Motivation

- Try out something new.
- Avoid use of Herbrand's theorem, because (unrestricted) interpretations can be much more compact than Herbrand interpretations.
- Find general theorem proving strategies with good termination behaviour, and which give more information in the case of termination.
- Find theorem proving strategies that can deal better with partial functions, and incompletely defined functions.

Geometric Formulas

Definition: We assume an infinite set of variables $\mathcal{V}$.

A variable atom is an atom of one of the following two forms:

1. $p(v_1, \ldots, v_n)$ with $n \geq 0$ and $v_1, \ldots, v_n \in \mathcal{V}$.
2. $v_1 \not\approx v_2$ with $v_1, v_2 \in \mathcal{V}$.

Observe that:

- There are no positive equalities.
- There are no constants and no function symbols.

Definition: A geometric formula has form

$$\forall \overline{x}\ A_1(\overline{x}) \wedge \cdots \wedge A_p(\overline{x}) \wedge x_1 \not\approx x'_1 \wedge \cdots \wedge x_q \not\approx x'_q \rightarrow Z(\overline{x}),$$

in which $x_1, x'_1, \ldots, x_q, x'_q \in \overline{x} \subseteq \mathcal{V}$.

$Z(\overline{x})$ can have one of the following three forms:

1. The false constant $\bot$.
2. A disjunction of atoms $B_1(\overline{x}) \vee \cdots \vee B_r(\overline{x})$, with $r > 0$.
3. An existential formula of form $\exists y\ B(\overline{x}, y)$.

Types 1 and 2 overlap (if one would allow $r = 0$) but we prefer to distinguish the types. Geometric formulas of Type 1 are called lemmas. Formulas of Type 2 are called disjunctive. Formulas of Type 3 are called existential.

Example 1

We might be interested in finding out whether
$a \approx b,\ b \approx c\ \vdash\ a \approx c.$

We try to find a model for

$$a \approx b, \quad b \approx c, \quad a \not\approx c.$$

Resulting geometric formulas are:

$A(X) \land B(Y) \land X \not\approx Y \to \bot,$
$B(X) \land C(Y) \land X \not\approx Y \to \bot,$
$A(X) \land C(X) \to \bot,$

$\to \exists x\ A(x),$
$\to \exists x\ B(x),$
$\to \exists x\ C(x).$

Example 2

What about $s(a) \approx a \vdash s(s(a)) \approx a$?

Try to find model for

$$s(a) \approx a, \quad s(s(a)) \not\approx a.$$

$A(X) \wedge S(X, Y) \wedge X \not\approx Y \rightarrow \bot,$
$A(X) \wedge S(X, Y) \wedge S(Y, X) \rightarrow \bot,$

$\exists x \; A(x),$
$\forall x \exists y \; S(x, y).$

Example 3

$a \approx s(a), \quad p(a,a) \vee p(s(a),s(a)) \vdash p(a,a).$

Negation of goal:

$$a \approx s(a), \quad p(a,a) \vee p(s(a),s(a)), \quad \neg p(a,a).$$

$A(X) \wedge S(X,Y) \wedge X \not\approx Y \rightarrow \bot,$
$A(X) \wedge S(X,Y) \rightarrow p(X,X) \vee p(Y,Y),$
$A(X) \wedge p(X,X) \rightarrow \bot,$

$\exists x \; A(x),$
$\forall x \exists y \; S(x,y).$

In general, the following holds:

Theorem:

Every set of first-order formulas can be translated into a set of geometric formulas, which is equisatisfiable.

The result (and the computation) can be linear in the size of the input.

The procedure is somewhat similar to clausification.

Outline of Model Search Algorithm

Definition: An interpretation is a finite set of atoms, with arguments from a given, infinite set $\mathcal{E}$.

Equality is interpreted as object equality, therefore there are no disequality atoms in interpretations.

Examples of interpretations are

$$A(e_0), \quad S(e_0, e_1), \quad S(e_1, e_2), \quad B(e_2).$$

$$A(e_0), \quad B(e_1), \quad P(e_0, e_1, e_2), \quad Q(e_2, e_2, e_1).$$

- Try to extend an interpretation I into a model. Normally, search starts with $M = \emptyset$.
- If I is not a model, there must exist a rule $\forall \overline{x}\ \Phi(\overline{x}) \rightarrow Z(\overline{x})$ and a ground substitution Θ, s.t. $I \models \Phi(\overline{x}\Theta)$, and $I \not\models Z(\overline{x}\Theta)$.
- If $Z(\overline{x}) = \bot$, then give up (go back to last choice)
- If $Z(\overline{x}) = B_1(\overline{x}) \vee \cdots \vee B_q(\overline{x})$, then backtrack over the extensions $I \cup \{B_j(\overline{x}\Theta)\}$.
- If $Z(\overline{x}) = \exists y\ B(\overline{x}, y)$, then let E be the set of constants in I. Backtrack over all $I \cup \{B(\overline{x}\Theta, e)\}$ with $e \in E$, and also try $I \cup \{B(\overline{x}\Theta, \hat{e})\}$ with a new $\hat{e} \notin E$.

Lemma Learning

Whenever I cannot be extended into a model, ensure that there exists a lemma of form $\forall \overline{z}\ \Psi(\overline{z}) \rightarrow \bot$, for which there exists a ground substitution Θ such that $I \models \Psi(\overline{z}\Theta)$.

It is not terribly important how the closing lemma is obtained. It should meet the following conditions:

- It must be a logical consequence of the initial set of formulas.
- There exists a ground substitution Θ, s.t. $M \models \Psi(\overline{z}\Theta)$.
- $\Psi(\overline{z})$ contains only variables. (i.e. no constants)
- And of course, we want the lemma to be as general, and as small as possible.

First calculus for Lemmas Construction (implemented in Geo)

The calculus consists of three rules:

- Lemma factoring (combined with merging)
- Disjunction resolution.
- Existential resolution.

Lemma Factoring: Let $\lambda =$

$$\forall \overline{x}\ A_1(\overline{x}) \wedge A_2(\overline{x}) \wedge \cdots \wedge A_p(\overline{x}) \wedge x_1 \not\approx x'_1 \wedge \cdots \wedge x_q \not\approx x'_q \rightarrow \bot,$$

be a lemma. Let Σ be a substitution of form $\{y := y'\}$. Then the following lemma is a factor of λ:

$$\forall \overline{x}\Sigma\ \ A_1(\overline{x}\Sigma) \wedge \cdots \wedge A_p(\overline{x}\Sigma) \wedge\ x_1\Sigma \not\approx x'_1\Sigma\ \wedge \cdots \wedge\ x_q\Sigma \not\approx x'_q\Sigma\ \rightarrow \bot.$$

Disjunction Resolution:

Let $\rho =$

$$\forall \overline{x}\ \Phi(\overline{x}) \rightarrow B_1(\overline{x}) \vee \cdots \vee B_q(\overline{x})$$

be a disjunctive formula.

Let $\lambda =$

$$\forall \overline{y}\ D_1(\overline{y}) \wedge \cdots \wedge D_r(\overline{y}) \wedge y_1 \not\approx y_1' \wedge \cdots \wedge y_s \not\approx y_s' \rightarrow \bot,$$

be a lemma, s.t. $B_1(\overline{x})$ and $D_1(\overline{y})$ are unifiable. Then the following formula is a disjunction resolvent of ρ and λ:

$$\forall\ \ \overline{x}\Sigma\ \ \overline{y}\Sigma\ \ \Phi(\overline{x})\Sigma \wedge$$

$$D_2(\overline{y})\Sigma \wedge \cdots \wedge D_r(\overline{y})\Sigma\ \wedge\ y_1\Sigma \not\approx y_1'\Sigma\ \wedge \cdots \wedge\ y_s\Sigma \not\approx y_s'\Sigma\ \rightarrow$$

$$B_2(\overline{x})\Sigma \vee \cdots \vee B_q(\overline{x})\Sigma.$$

Existential Resolution:

Let $\rho =$

$$\forall\overline{x}\ \Phi(\overline{x}) \rightarrow \exists y\ B(\overline{x}, y)$$

be an existential formula.

Let $\lambda =$

$$\forall\overline{z}\ v\quad \Psi(\overline{z}) \wedge B(\overline{z}, v) \wedge v \not\approx z_1 \wedge \cdots \wedge v \not\approx z_s \rightarrow \bot,$$

be a lemma, s.t. $B(\overline{x}, y)$ and $B(\overline{z}, v)$ are unifiable and $v \notin \overline{z}$. Then the following formula is an existential resolvent of ρ and λ:

$$\forall\ \overline{x}\Sigma\quad \overline{z}\Sigma\quad \Phi(\overline{x})\Sigma \wedge \Psi(\overline{z})\Sigma \rightarrow B(\overline{z}, z_1)\Sigma \vee \cdots \vee B(\overline{z}, z_s)\Sigma.$$

Theorem Using instantiation, disjunction resolution and existential resolution, it is always possible to construct a lemma.

A Reconstruction of Existential Resolution

Existential rule is complicated, and it has a an unpleasant proof theory. (For example if you you want to show that two applications of existential resolution can be permuted)

Existential Resolution:

Let $\rho =$

$$\forall \overline{x}\ \Phi(\overline{x}) \rightarrow \exists y\ B(\overline{x}, y)$$

be an existential formula.

Let $\lambda =$

$$\forall \overline{z}\ v \quad \Psi(\overline{z}) \wedge B(\overline{z}, v) \rightarrow \bot,$$

be a lemma, s.t. $B(\overline{x}, y)$ and $B(\overline{z}, v)$ are unifiable and $v \notin \overline{z}$. Let Σ be the mgu. Then the following lemma is an existential resolvent of ρ and λ:

$$\forall\ \overline{x}\Sigma \quad \overline{z}\Sigma \quad \Phi(\overline{x}\Sigma) \wedge \Psi(\overline{z}\Sigma) \rightarrow \bot.$$

Equality Resolution

Let $\rho =$

$$\forall \overline{x}\ \Phi(\overline{x}) \wedge x_1 \not\approx x_2 \rightarrow \bot.$$

Let λ be a lemma of form

$$\forall (\overline{x}\Sigma)\ \Psi(\overline{x}\Sigma) \rightarrow \bot,$$

where Σ is the substitution $x_1 := x_2$. Then the following lemma is an equality resolvent of ρ and λ:

$$\forall \overline{x}\ \Phi(\overline{x}) \wedge \Psi(\overline{x}) \rightarrow \bot.$$

Theorem: A complete application of existential resolution (the complicated version) with disjunction resolution can be replaced by an application of simple existential resolution and some steps of equality resolution.

Proof: In case $s = 0$, immediately apply simple existential resolution. Otherwise, use equality resolution in order to resolve

$$\forall \overline{z} v \ \Psi(\overline{z}) \wedge B(\overline{z}, v) \wedge v \not\approx z_1 \wedge v \not\approx z_2 \wedge \cdots \wedge v \not\approx z_s \rightarrow \bot,$$

with

$$\forall \overline{z} \ \Psi(\overline{z}) \wedge B(\overline{z}, z_1) \wedge z_1 \not\approx z_2 \wedge \cdots \wedge z_1 \not\approx z_s \rightarrow \bot$$

into

$$\forall \overline{z} v \ \Psi(\overline{z}) \wedge B(\overline{z}, v) \wedge v \not\approx z_2 \wedge \cdots \wedge v \not\approx z_s \rightarrow \bot.$$

Apply induction on result.

Redundancy

In superposition/resolution, there exist many kinds of elimination/simplification rules. (subsumption, tautology elimination, demodulation)

There are two ways of using redundancy: Simplification and deletion.

In superposition, both non-deletion and non-simplicity are inherited.

In geometric resolution, only non-simplicity is inherited. Forward deletion cannot happen.

Simplification: Functional Reduction

Suppose that predicate F has only one positive occurrence, which has form $\forall x \ \Phi(\overline{x}) \rightarrow \exists y \ F(\overline{x}, y)$.

Then F is functional in every candidate model M.

In lemmas $\forall \overline{y} \ \Psi(\overline{y}) \rightarrow \bot$, simplify all '$F$-forks'.

If $\Psi(\overline{y})$ contains $F(y_1, \ldots, y_n, y'), \ \ F(y_1, \ldots, y_n, y'')$, then substitute $y' := y''$.

$\Rightarrow$ Big performance improvement.

Equality Splitting

Consider two lemmas of form $\forall \overline{x}_1 \ \Phi_1(\overline{x}_1) \wedge x_1 \not\approx x_2 \rightarrow \bot$, and $\forall \overline{x}_2 \ \Psi_2(\overline{x}_2) \rightarrow \bot$, s.t. $\Phi_1(\overline{x}_1) \subset \Phi_2(\overline{x}_2)$.

$\forall \overline{x}_2 \ \Phi_2(\overline{x}_2) \rightarrow \bot$ is applied only when $x_1 = x_2$.

Therefore, one might as well instantiate $x_1 := x_2$ in $\forall \overline{x}_2 \ \Phi_2(\overline{x}_2) \rightarrow \bot$.

$\Rightarrow$ makes geometric resolution comparable to resolution in terms of size of proofs, attempted models.

Functional Reduction Revisited

Functional reduction can be viewed as equality split using functional axioms:

$\forall \overline{x}\ x_1 x_2\ F(\overline{x}, x_1) \wedge F(\overline{x}, x_2) \wedge x_1 \not\approx x_2 \rightarrow \bot.$

Add these axioms, and functional reduction comes for free.

They are not logical consequences of the input, but they are safe, because they are never applied. (They can be viewed as inductive consequences of the input)

Algorithm for Lemma Simplification

Let λ be lemma. Start with set $\Lambda = \{\lambda\}$.

As long as there is a $\lambda' \in \Lambda$ of form $\forall\overline{x}\ \Phi(\overline{x}) \rightarrow \bot$, and a lemma $\forall\overline{x}\ \Psi(\overline{x}) \wedge x_1 \not\approx x_1' \wedge \cdots \wedge x_s \not\approx x_s' \rightarrow \bot$, s.t. $\Psi(\overline{x}) \subset \Phi(\overline{x})$, replace λ by

$$(\forall\overline{x}\ \Phi(\overline{x}) \rightarrow \bot)[x_1 := x_1'],$$

$$(\forall\overline{x}\ \Phi(\overline{x}) \wedge x_1 \not\approx x_1' \rightarrow \bot)[x_2 := x_2'],$$

$$(\forall\overline{x}\ \Phi(\overline{x}) \wedge x_1 \not\approx x_1' \wedge x_2 \not\approx x_2' \rightarrow \bot)[x_3 := x_3'],$$

$$\cdots$$

$$(\forall\overline{x}\ \Phi(\overline{x}) \wedge x_1 \not\approx x_1' \wedge x_2 \not\approx x_2' \wedge \cdots \wedge x_{s-1} \not\approx x_{s-1}' \rightarrow \bot)[x_s := x_s'].$$

Example

$$\forall xy \; A(x) \land B(y) \land x \not\approx y \to \bot,$$

simplifies

$$\forall xyzt \; A(x) \land S(x,z) \land B(y) \land S(y,t) \land z \not\approx t \to \bot$$

into

$$\forall xzt \; A(x) \land S(x,z) \land B(x) \land S(x,t) \land z \not\approx t \to \bot.$$

In case S is functional, this lemma is subsumed by the functional axiom.

Completeness of Equality Splitting

Completeness is very easy to prove. It follows from the fact that the model search algorithm does not care much about the lemma returned.

But completeness proofs are not informative.

What we really need are simulation proofs:

Define $G_1 \prec G_2$ as follows: If M has a closing lemma in G_1, then it has a closing lemma in G_2.

Let $G_1 =$ unrefined set of rules.

Let $G_2 =$ set of rules in which lemmas are eagerly simplified.

Show that model search algorithm preserves $\prec$. (or possibly reserves $\prec$)

Simulation by Proof Permutation

- The calculus (consisting of lemma factoring, disjunctive resolution, existential resolution) is complete.

  Equality splitting is backward application of the equality resolution rule.

  So the now the same calculus is used in two ways.

- Proofs in a certain normal form ($\approx\exists$-normal form) correspond to runs of the model search algorithm.

- Proofs that are not in $\approx\exists$-normal form can be permuted back into $\approx\exists$-normal form.

- In case the violating part is obtained by equality splitting, there is no increase in size. (and the proofs are isomorphic)

This proves the simulation property from the last slide.

Long Term Research Agenda

- Understand the proof theory and behaviour of geometric resolution.
- Understand its 'psychology'.
- Find techniques for eliminating/simplifying lemmas (redundancy)
- Obtain an efficient implementation. Find efficient indexing data structures. Find good heuristics.
- Obtain decision procedures for the standard fragments of FOL.
- Become a regular CASC-winner. Make the users happy.

Conclusions

- We developed a variant of geometric resolution that has a nice proof theory.
- Using this, we can show that every proof in this calculus can be permuted into proof that could have been obtained by the model search algorithm.
- We need to understand better how the proof trees correspond to runs of the algorithm. For the algorithm, the proof as DAG matters.
- More sophisticated forms of redundancy probably can be obtained when the selection function is restricted.
- More empirical data is needed.