Transforming and Analyzing Proofs in the CERES-system

S. Hetzl A. Leitsch D. Weller B. Woltzenlogel Paleo

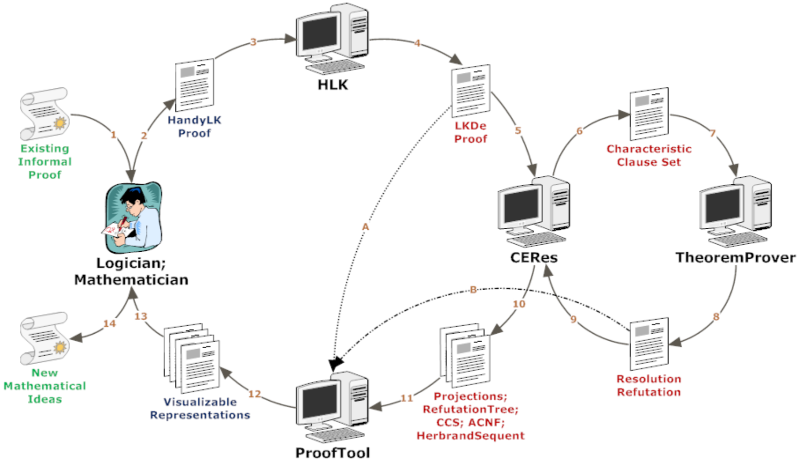
KEAPPA, 22 November 2008

Outline

Purpose

- Proof transformations
- In particular: cut-elimination by resolution
- Goal: obtain new (analytic) proofs from known ones

Overview


HLK
HandyLK Proof
LKDe Proof
Existing Informal Proof
Logician; Mathematician
Characteristic Clause Set
CERes
TheoremProver
New Mathematical Ideas
Visualizable Representations
Projections; RefutationTree; CCS; ACNF; HerbrandSequent
Resolution Refutation
ProofTool
1
2
3
4
5
6
7
8
9
10
11
12
13
14
A
B

LK

- Proof calculus: sequent calculus **LK**

Example

Rules for $\wedge$:

$$\frac{\Gamma \vdash \Delta, A \quad \Pi \vdash \Lambda, B}{\Gamma, \Pi \vdash \Delta, \Lambda, A \wedge B} \wedge : r$$

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge : l1 \qquad \frac{A, \Gamma \vdash \Delta}{B \wedge A, \Gamma \vdash \Delta} \wedge : l2$$

LKDe

- Additional rules for easier proof formalization

LKDe

- Additional rules for easier proof formalization
- Definition introduction

$$\frac{A(t_1, \ldots, t_k), \Gamma \vdash \Delta}{P(t_1, \ldots, t_k), \Gamma \vdash \Delta} \operatorname{def}_P : l$$

LKDe

- Additional rules for easier proof formalization
- Definition introduction
- Equality handling

$$\frac{\Gamma_1 \vdash \Delta_1, s = t \quad A[s], \Gamma_2 \vdash \Delta_2}{A[t], \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} =: l1$$

Writing **LKDe** proofs

- Specialized language: *HandyLK*
- Why not Isabelle, Coq, etc.?
  - Higher-order logic vs. first-order method
  - Proof assistants focus on existance of proof, not proof object itself

The *HandyLK* language

- Between natural language and sequent calculus
  - closer to sequent calculus
- Supports many-sorted first-order language

HandyLK example - predicate definitions

- `define predicate I by all n ex k f( n + k ) = x;`
- $\forall x \, (I(x) \leftrightarrow \forall n \exists k \; f(n + k) = x)$

HandyLK example - predicate definitions

- `define predicate I by all n ex k f( n + k ) = x;`
- $\forall x\ (I(x) \leftrightarrow \forall n \exists k\ f(n+k) = x)$
- `with undef I`
  `:- all n ex k f( n + k ) = 0;`
- $$\frac{\Gamma \vdash \Delta, \forall n \exists k f(n+k) = 0}{\Gamma \vdash \Delta, I(0)}\ \mathrm{def}_I : r$$

HandyLK features

- Prove propositional tautologies automatically
- Define proofs recursively
- Define proofs with parameters that can be instantiated

Storing proofs — XML

- Proof transformations do not work directly on *HandyLK* proofs
- Compiled by `HLK` to **LKDe** in XML
- `proofdatabase.dtd` allows storage of proofs as DAGs
- Formulas, terms stored as trees

The CERES method

- Clause set $\mathrm{CL}(\pi)$ is extracted from **LKDe**-proof π
- $\mathrm{CL}(\pi)$ is refuted by a resolution theorem prover
- Resolution refutation is converted to an **LK** refutation γ
- γ is composed with material from π: **LKDe**-proof ψ
- ψ contains at most atomic cuts

System demonstration

Background: Tape with infinitely many cells where each cell is labelled 0 or 1.

Theorem

There are two distinct cells that are labelled the same.

Lemma

Either infinitely many cells are labelled 0*, or infinitely many cells are labelled* 1*.*

System demonstration

Simplified Herbrand Sequent

$$f(p_1) = 0 \vee f(p_1) = 1, f(p_2) = 0 \vee f(p_2) = 1, f(p_3) = 0 \vee f(p_3) = 1,$$
$$f(p_4) = 0 \vee f(p_4) = 1, f(p_5) = 0 \vee f(p_5) = 1, f(p_6) = 0 \vee f(p_6) = 1,$$
$$f(p_7) = 0 \vee f(p_7) = 1$$
$$\vdash$$
$$p_1 \neq p_2 \wedge f(p_1) = f(p_2), p_3 \neq p_1 \wedge f(p_3) = f(p_1),$$
$$p_3 \neq p_2 \wedge f(p_3) = f(p_2), p_1 \neq p_4 \wedge f(p_1) = f(p_4),$$
$$p_5 \neq p_6 \wedge f(p_5) = f(p_6), p_7 \neq p_5 \wedge f(p_7) = f(p_5),$$
$$p_7 \neq p_6 \wedge f(p_7) = f(p_6), p_4 \neq p_7 \wedge f(p_4) = f(p_7).$$

where the p_i are distinct positions on the tape.

Even More Simplified Herbrand Sequent

$$f(p_1) = 0 \lor f(p_1) = 1, f(p_2) = 0 \lor f(p_2) = 1, f(p_3) = 0 \lor f(p_3) = 1,$$
$$\vdash$$
$$p_1 \neq p_2 \land f(p_1) = f(p_2),$$
$$p_3 \neq p_1 \land f(p_3) = f(p_1),$$
$$p_3 \neq p_2 \land f(p_3) = f(p_2).$$

where the p_i are distinct positions on the tape.

Future Work

- Extend CERES method to fragments of higher-order logic
- Enhance HLK by term-rewriting features to handle equational aspects of proofs
- Long term: Use existing proof assistants
- Simplify Herbrand sequent automatically