CERES: a program for cut-elimination

Daniel Weller

KGRC, Vienna, 5 June 2009

Proof analysis

- Mathematical proofs usually contain more information (bounds, algorithms, ...) than theorems
- Some information may be implicit in the use of lemmas
- The informal use of lemmas corresponds to the use of *cuts* in sequent calculus proofs
- Goal: Make implicit information explicit by cut-elimination

Sequent calculus **LK**

- Rules like

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\ \wedge : l_1 \qquad \frac{\Gamma \vdash \Delta, A[x \leftarrow t]}{\Gamma \vdash \Delta, (\exists x)A,}\ \exists : r$$

$$\frac{\Gamma \vdash \Delta, A \quad A, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\ cut$$

Sequent calculus **LK**

- Rules like

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge : l_1 \qquad \frac{\Gamma \vdash \Delta, A[x \leftarrow t]}{\Gamma \vdash \Delta, (\exists x)A,} \exists : r$$

$$\frac{\Gamma \vdash \Delta, A \quad A, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda} \; cut$$

- *Subformula property* of cut-free proofs: If A occurs in a proof, then A occurs (modulo substitution) in the end-sequent

Resolution calculus

- Clausal calculus (i.e. only atomic sequents)
- Based on *most general unification* and *atomic* cut
- Only three rules:

$$\frac{\Gamma \vdash \Delta, A \quad A', \Pi \vdash \Lambda}{(\Gamma, \Pi \vdash \Delta, \Lambda)\sigma}\ res \qquad \frac{A, A', \Gamma \vdash \Delta}{(A, \Gamma \vdash \Delta)\sigma}\ fact : l$$

  σ is an mgu of $\{A, A'\}$.

- Popular for automated theorem proving — many implementations exist.

Methodology

- CERES: cut-elimination method for classical first-order logic (M. Baaz, A. Leitsch 2000)
- based on set of clauses $\mathrm{CL}(\pi)$ (the *characteristic clause set*) extracted from **LK**-proof π
- resolution refutation of $\mathrm{CL}(\pi)$ serves as skeleton of proof with at most atomic cuts (ACNF)

CERES

- Idea: partition **LK**-proof π into *implicit* and *explicit* parts
- Implicit part: characteristic clause set $\mathrm{CL}(\pi)$
- Explicit part: proof projections $\mathcal{P}(\pi)$

CERES

- Input: Skolemized proof π of S
- Compute $\mathrm{CL}(\pi)$
- Compute resolution refutation γ of $\mathrm{CL}(\pi)$
- Apply global substitution to γ to obtain **LK** refutation γ'
- Compute $\mathcal{P}(\pi)$
- Combine instances of proofs in $\mathcal{P}(\pi)$ with γ' to obtain proof of S with at most atomic cuts

CERES - $\mathrm{CL}(\pi)$

- Let ρ be a rule in π, then define $\mathrm{CL}_\rho(\pi)$:
- If ρ is an axiom where $\Gamma_1 \vdash \Delta_1$ are the cut ancestors, define $\mathrm{CL}_\rho(\pi) = \{\Gamma_1 \vdash \Delta_1\}$.

CERES - $\mathrm{CL}(\pi)$

- Let ρ be a rule in π, then define $\mathrm{CL}_\rho(\pi)$:
- If ρ is a unary rule with immediate predecessor ρ', then $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho'}(\pi)$.

CERES - $\mathrm{CL}(\pi)$

- Let ρ be a rule in π, then define $\mathrm{CL}_\rho(\pi)$:
- If ρ is a binary rule with immediate predecessors ρ_1, ρ_2, then
  - If the active formulas of ρ are cut ancestors, define $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho_1}(\pi) \cup \mathrm{CL}_{\rho_2}(\pi)$, and
  - if the active formulas of ρ are end-sequent ancestors, then define $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho_1}(\pi) \times \mathrm{CL}_{\rho_2}(\pi)$ where $S_1 \times S_2 = \{\Gamma, \Pi \vdash \Delta, \Lambda \mid \Gamma \vdash \Delta \in S_1, \Pi \vdash \Lambda \in S_2\}$.

CERES - $\mathrm{CL}(\pi)$

- Let ρ be a rule in π, then define $\mathrm{CL}_\rho(\pi)$:
- If ρ is a binary rule with immediate predecessors ρ_1, ρ_2, then
  - If the active formulas of ρ are cut ancestors, define $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho_1}(\pi) \cup \mathrm{CL}_{\rho_2}(\pi)$, and
  - if the active formulas of ρ are end-sequent ancestors, then define $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho_1}(\pi) \times \mathrm{CL}_{\rho_2}(\pi)$ where $S_1 \times S_2 = \{\Gamma, \Pi \vdash \Delta, \Lambda \mid \Gamma \vdash \Delta \in S_1, \Pi \vdash \Lambda \in S_2\}$.
- Define $\mathrm{CL}(\pi) = \mathrm{CL}_{\rho_0}(\pi)$ where ρ_0 is the last rule of π.

CERES - $\mathrm{CL}(\pi)$

- Let ρ be a rule in π, then define $\mathrm{CL}_\rho(\pi)$:
- If ρ is a binary rule with immediate predecessors ρ_1, ρ_2, then
  - If the active formulas of ρ are cut ancestors, define $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho_1}(\pi) \cup \mathrm{CL}_{\rho_2}(\pi)$, and
  - if the active formulas of ρ are end-sequent ancestors, then define $\mathrm{CL}_\rho(\pi) = \mathrm{CL}_{\rho_1}(\pi) \times \mathrm{CL}_{\rho_2}(\pi)$ where $S_1 \times S_2 = \{\Gamma, \Pi \vdash \Delta, \Lambda \mid \Gamma \vdash \Delta \in S_1, \Pi \vdash \Lambda \in S_2\}$.
- Define $\mathrm{CL}(\pi) = \mathrm{CL}_{\rho_0}(\pi)$ where ρ_0 is the last rule of π.
- One can show that $\mathrm{CL}(\pi)$ is always unsatisfiable.

CERES example

$$\varphi_1 = \dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \quad Q(u) \vdash Q(u)}{P(u) \to Q(u) \vdash (\exists y)(P(u) \to Q(y))}\ \exists : r, \to : r, \to : l}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(u) \to Q(y))}\ \forall : l}{(\forall x)(P(x) \to Q(x)) \vdash (\forall x)(\exists y)(P(x) \to Q(y))}\ \forall : r$$

$$\varphi_2 = \dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \quad Q(v) \vdash Q(v)}{P(a) \to Q(v) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : r, \to : r, \to : l}{(\exists y)(P(a) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : l}{(\forall x)(\exists y)(P(x) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \forall : l$$

$$\varphi = \dfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$$\varphi_1 = \dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \quad Q(u) \vdash Q(u)}{P(u) \to Q(u) \vdash (\exists y)(P(u) \to Q(y))}\ \exists : r, \to : r, \to : l}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(u) \to Q(y))}\ \forall : l}{(\forall x)(P(x) \to Q(x)) \vdash (\forall x)(\exists y)(P(x) \to Q(y))}\ \forall : r$$

$$\varphi_2 = \dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \quad Q(v) \vdash Q(v)}{P(a) \to Q(v) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : r, \to : r, \to : l}{(\exists y)(P(a) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : l}{(\forall x)(\exists y)(P(x) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \forall : l$$

$$\varphi = \dfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$$\varphi_1 = \dfrac{\dfrac{\dfrac{P(u)\vdash P(u) \qquad Q(u)\vdash Q(u)}{P(u)\to Q(u)\vdash(\exists y)(P(u)\to Q(y))}\ \exists:r,\to:r,\to:l}{(\forall x)(P(x)\to Q(x))\vdash(\exists y)(P(u)\to Q(y))}\ \forall:l}{(\forall x)(P(x)\to Q(x))\vdash(\forall x)(\exists y)(P(x)\to Q(y))}\ \forall:r$$

$$\varphi_2 = \dfrac{\dfrac{\dfrac{\color{red}{P(a)}\color{black}{\vdash P(a) \qquad Q(v)\vdash} \color{red}{Q(v)}}{P(a)\to Q(v)\vdash\color{red}{(\exists y)(P(a)\to Q(y))}}\ \exists:r,\to:r,\to:l}{(\exists y)(P(a)\to Q(y))\vdash\color{red}{(\exists y)(P(a)\to Q(y))}}\ \exists:l}{(\forall x)(\exists y)(P(x)\to Q(y))\vdash\color{red}{(\exists y)(P(a)\to Q(y))}}\ \forall:l$$

$$\varphi = \dfrac{\varphi_1 \qquad\qquad \varphi_2}{\vdash}\ cut$$

CERES example

$$\varphi_1 = \cfrac{\cfrac{\cfrac{P(u) \vdash P(u) \qquad Q(u) \vdash Q(u)}{P(u) \rightarrow Q(u) \vdash (\exists y)(P(u) \rightarrow Q(y))}\ \exists : r, \rightarrow: r, \rightarrow: l}{(\forall x)(P(x) \rightarrow Q(x)) \vdash (\exists y)(P(u) \rightarrow Q(y))}\ \forall : l}{(\forall x)(P(x) \rightarrow Q(x)) \vdash (\forall x)(\exists y)(P(x) \rightarrow Q(y))}\ \forall : r$$

$$\varphi_2 = \cfrac{\cfrac{\cfrac{\vdash P(a) \qquad Q(v) \vdash}{P(a) \rightarrow Q(v) \vdash}\ \rightarrow: l}{(\exists y)(P(a) \rightarrow Q(y)) \vdash}\ \exists : l}{(\forall x)(\exists y)(P(x) \rightarrow Q(y)) \vdash}\ \forall : l$$

$$\varphi = \cfrac{\varphi_1 \qquad\qquad \varphi_2}{\vdash}\ cut$$

CERES example

$$\varphi_1 = \cfrac{\cfrac{\cfrac{P(u) \vdash Q(u)}{\vdash (\exists y)(P(u) \to Q(y))}\ \exists : r, \to : r}{\vdash (\exists y)(P(u) \to Q(y))}}{\vdash (\forall x)(\exists y)(P(x) \to Q(y))}\ \forall : r$$

$$\varphi_2 = \cfrac{\cfrac{\cfrac{\vdash P(a) \quad Q(v) \vdash}{P(a) \to Q(v) \vdash}\ \to : l}{(\exists y)(P(a) \to Q(y)) \vdash}\ \exists : l}{(\forall x)(\exists y)(P(x) \to Q(y)) \vdash}\ \forall : l$$

$$\varphi = \cfrac{\varphi_1 \qquad \varphi_2}{\vdash}\ cut$$

CERES example

$$\mathrm{CL}(\varphi) = \{P(u) \vdash Q(u); \;\; \vdash P(a); \;\; Q(v) \vdash\}$$

refutation:

$$\dfrac{\dfrac{\vdash P(a) \quad P(u) \vdash Q(u)}{\vdash Q(a)}\;R \qquad Q(v) \vdash}{\vdash}\;R$$

$\sigma = [u \leftarrow a, v \leftarrow a]$
ground refutation:

$$\dfrac{\dfrac{\vdash P(a) \quad P(a) \vdash Q(a)}{\vdash Q(a)}\;R \qquad Q(a) \vdash}{\vdash}\;R$$

CERES - $\mathcal{P}(\pi)$

- Let ρ be a rule in π, then define $\mathcal{P}_\rho(\pi)$:
- If ρ is an axiom S, define $\mathcal{P}_\rho(\pi) = \{S\}$

CERES - $\mathcal{P}(\pi)$

- Let ρ be a rule in π, then define $\mathcal{P}_\rho(\pi)$:
- If ρ is a unary rule with immediate predecessor ρ', distinguish:
  - The active formulas of ρ are ancestors of cut formulas. Then define $\mathcal{P}_\rho(\pi) = \mathcal{P}_{\rho'}(\pi)$
  - The active formulas of ρ are ancestors of the end-sequent. Then define $\mathcal{P}_\rho(\pi) := \{\rho(\psi) \mid \psi \in \mathcal{P}_{\rho'}(\pi)\}$ where $\rho(\psi)$ is the proof that is obtained from ψ by applying ρ to its end-sequent.

    Need assumption: π skolemized!

CERES - $\mathcal{P}(\pi)$

- Let ρ be a rule in π, then define $\mathcal{P}_\rho(\pi)$:
- If ρ is a binary rule with immediate predecessors ρ_1 and ρ_2, distinguish:
  - If the active formulas of ρ are cut ancestors, let $\Gamma_i \vdash \Delta_i$ be the end-sequent ancestors in the conclusion sequent of ρ_i and define

    $$\mathcal{P}_\rho(\pi) = \mathcal{P}_{\rho_1}(\pi)^{\Gamma_2 \vdash \Delta_2} \cup \mathcal{P}_{\rho_2}(\pi)^{\Gamma_1 \vdash \Delta_1}$$

    where $P^{\Gamma\vdash\Delta} = \{\psi^{\Gamma\vdash\Delta} \mid \psi \in P\}$ and $\psi^{\Gamma\vdash\Delta}$ is ψ followed by weakenings adding $\Gamma \vdash \Delta$.
  - If the active formulas of ρ are end-sequent ancestors, then

    $$\mathcal{P}_\rho(\pi) := \mathcal{P}_{\rho_1}(\pi) \times \mathcal{P}_{\rho_2}(\pi).$$

    where $P \times Q = \{\rho(\psi, \chi) \mid \psi \in P, \chi \in Q\}$ and $\rho(\psi, \chi)$ is the proof that is obtained from the proofs ψ and χ by applying the binary rule ρ.

CERES - $\mathcal{P}(\pi)$

- Let ρ be a rule in π, then define $\mathcal{P}_\rho(\pi)$:
- Define $\mathcal{P}(\pi) = \mathcal{P}_{\rho_0}(\pi)$ where ρ_0 is the last rule of π.

CERES - $\mathcal{P}(\pi)$

- Let ρ be a rule in π, then define $\mathcal{P}_{\rho}(\pi)$:
- Define $\mathcal{P}(\pi) = \mathcal{P}_{\rho_0}(\pi)$ where ρ_0 is the last rule of π.
- Let the end-sequent of π be $\Gamma \vdash \Delta$. One can show that for every $\Pi \vdash \Lambda \in \mathrm{CL}(\pi)$ there is a $\psi \in \mathcal{P}(\pi)$ with end-sequent $\Gamma, \Pi \vdash \Delta, \Lambda$.

CERES example

$$\varphi_1 = \dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \qquad Q(u) \vdash Q(u)}{P(u) \to Q(u) \vdash (\exists y)(P(u) \to Q(y))}\ \exists : r, \to : r, \to : l}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(u) \to Q(y))}\ \forall : l}{(\forall x)(P(x) \to Q(x)) \vdash (\forall x)(\exists y)(P(x) \to Q(y))}\ \forall : r$$

$$\varphi_2 = \dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(v) \vdash Q(v)}{P(a) \to Q(v) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : r, \to : r, \to : l}{(\exists y)(P(a) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : l}{(\forall x)(\exists y)(P(x) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \forall : l$$

$$\varphi = \dfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$$\dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \quad Q(u) \vdash Q(u)}{P(u) \to Q(u), P(u) \vdash Q(u)}\ \to: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}\ \forall : l}{\varphi_1 = (\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}$$

$$\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \quad Q(v) \vdash Q(v)}{P(a) \to Q(v) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : r, \to: r, \to: l}{(\exists y)(P(a) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \exists : l}{\varphi_2 = (\forall x)(\exists y)(P(x) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \forall : l$$

$$\varphi = \dfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$$\varphi_1 = \dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \quad Q(u) \vdash Q(u)}{P(u) \to Q(u), P(u) \vdash Q(u)} \;\to: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)} \;\forall: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}$$

$$\varphi_2 = \dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \quad Q(v) \vdash Q(v)}{P(a) \to Q(v) \vdash (\exists y)(P(a) \to Q(y))} \;\exists: r, \to: r, \to: l}{(\exists y)(P(a) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))} \;\exists: l}{(\forall x)(\exists y)(P(x) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))} \;\forall: l$$

$$\varphi = \dfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))} \; cut$$

CERES example

$$\varphi_1 = \cfrac{\cfrac{\cfrac{P(u) \vdash P(u) \qquad Q(u) \vdash Q(u)}{P(u) \to Q(u), P(u) \vdash Q(u)}\ \to : l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}\ \forall : l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}$$

$$\varphi_2 = \cfrac{\cfrac{\cfrac{P(a) \vdash P(a)}{\vdash P(a), (\exists y)(P(a) \to Q(y))}\ \exists : r, \to : r, w : r}{\vdash P(a), (\exists y)(P(a) \to Q(y))}}{\vdash P(a), (\exists y)(P(a) \to Q(y))}$$

$$\varphi = \cfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$$\varphi_1 = \cfrac{\cfrac{\cfrac{P(u) \vdash P(u) \quad Q(u) \vdash Q(u)}{P(u) \to Q(u), P(u) \vdash Q(u)}\ \to: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}\ \forall: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}$$

$$\varphi_2 = \cfrac{\cfrac{\cfrac{P(a) \vdash P(a) \quad Q(v) \vdash Q(v)}{P(a) \to Q(v) \vdash (\exists y)(P(a) \to Q(y))}\ \exists: r, \to: r, \to: l}{(\exists y)(P(a) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \exists: l}{(\forall x)(\exists y)(P(x) \to Q(y)) \vdash (\exists y)(P(a) \to Q(y))}\ \forall: l$$

$$\varphi = \cfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$$\varphi_1 = \cfrac{\cfrac{\cfrac{P(u) \vdash P(u) \quad Q(u) \vdash Q(u)}{P(u) \to Q(u), P(u) \vdash Q(u)}\ \to: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}\ \forall: l}{(\forall x)(P(x) \to Q(x)), P(u) \vdash Q(u)}$$

$$\varphi_2 = \cfrac{\cfrac{\cfrac{Q(v) \vdash Q(v)}{Q(v) \vdash (\exists y)(P(a) \to Q(y))}\ \exists: r, \to: r, w: l}{Q(v) \vdash (\exists y)(P(a) \to Q(y))}}{Q(v) \vdash (\exists y)(P(a) \to Q(y))}$$

$$\varphi = \cfrac{\varphi_1 \qquad \varphi_2}{(\forall x)(P(x) \to Q(x)) \vdash (\exists y)(P(a) \to Q(y))}\ cut$$

CERES example

$\varphi(\gamma) =$

$$\cfrac{\cfrac{\begin{matrix}\varphi(\vdash P(a)) \\ B \vdash C, P(a)\end{matrix} \qquad \begin{matrix}\varphi(P(a) \vdash Q(a)) \\ P(a), B \vdash C, Q(a)\end{matrix}}{B, B \vdash C, C, Q(a)}\ cut \qquad \begin{matrix}\varphi(\vdash Q(a)) \\ Q(a), B \vdash C\end{matrix}}{\cfrac{B, B, B \vdash C, C, C}{B \vdash C}\ \text{contractions}}\ cut$$

where $B = (\forall x)(P(x) \to Q(x)),\ C = (\exists y)(P(a) \to Q(y))$.

CERES vs. Gentzen

Theorem (Baaz, Leitsch 2000)

There exists a sequence of **LK**-*proofs* $(\psi_n)_{n \in \mathbb{N}}$ *such that*

1. *The Gentzen method produces proof trees with non-elementarily many nodes on* ψ_n.
2. *CERES constructs a cut-free proof out of* ψ_n *in exponentially many steps.*

CERES vs. Gentzen

Theorem (Baaz, Leitsch 2000)

There exists a sequence of **LK**-*proofs* $(\psi_n)_{n\in\mathbb{N}}$ *such that*

1. *The Gentzen method produces proof trees with non-elementarily many nodes on* ψ_n.
2. *CERES constructs a cut-free proof out of* ψ_n *in exponentially many steps.*

Theorem (Baaz, Leitsch 2006)

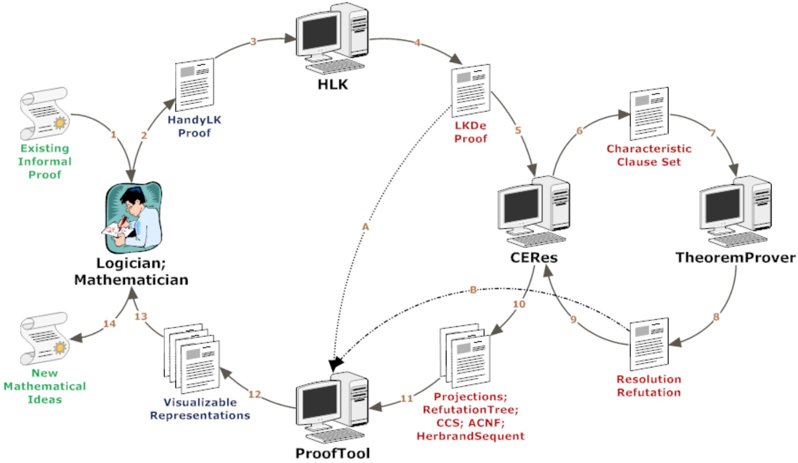
Let φ *be an* **LK**-*proof and* ψ *be an ACNF of* φ *under Gentzen's or Tait's method. Then there exists an ACNF* χ *of* φ *under CERES such that*

$$l(\chi) \leq l(\varphi) * l(\psi) * 2^{2*l(\psi)} + 2.$$

CERES system

- The CERES method has been implemented to allow (semi-)automatic proof analysis by cut-elimination.
- Auxiliary tools have been developed to
  - aid in the formalization of proofs and
  - let users visualize proofs, sequents, formulas, . . .

System overview



The calculus

- CERES uses a version of **LK**, called **LKDe**, which has additional rules for easier proof formalization:

The calculus

- CERES uses a version of **LK**, called **LKDe**, which has additional rules for easier proof formalization:
- Definition introduction

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

The calculus

- CERES uses a version of **LK**, called **LKDe**, which has additional rules for easier proof formalization:
- Definition introduction
- Equality handling

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

The tools

- Proofs are written in the language *HandyLK* and compiled to **LKDe** using the `hlk` compiler
  - Proves propositional tautologies automatically

The tools

- Proofs are written in the language *HandyLK* and compiled to **LKDe** using the `hlk` compiler
  - Proves propositional tautologies automatically
  - Proofs can be defined recursively

The tools

- Proofs are written in the language *HandyLK* and compiled to **LKDe** using the `hlk` compiler
  - Proves propositional tautologies automatically
  - Proofs can be defined recursively
  - Proof schemes can be defined and instantiated

The tools

- Proofs are written in the language *HandyLK* and compiled to **LKDe** using the `hlk` compiler
  - Proves propositional tautologies automatically
  - Proofs can be defined recursively
  - Proof schemes can be defined and instantiated
  - Many-sorted first-order languages are supported

The tools

- Input and output of the system can be visualized using the `ProofTool`
  - Supports some very light proof editing
  - Proofs, sequents, ... can be exported to LaTeX

The tools

- Input and output of the system can be visualized using the `ProofTool`
  - Supports some very light proof editing
  - Proofs, sequents, . . . can be exported to LaTeX
- In fully automatic mode, CERES supports the resolution provers Otter and Prover9

The tools

- Input and output of the system can be visualized using the ProofTool
    - Supports some very light proof editing
    - Proofs, sequents, . . . can be exported to LaTeX
- In fully automatic mode, CERES supports the resolution provers Otter and Prover9
- Resolution refutations can also be constructed (semi-)automatically using our prover ATP

Example proof

- A version of the pigeon hole principle: The “tape proof” due to C. Urban.
- On a tape with infinitely many cells, each labelled either 0 or 1, there are two distinct cells with the same label.
- Uses a classical lemma: Either infinitely many cells are labelled 0, or infinitely many cells are labelled 1.

System demonstration

Successful applications

Example (Hetzl, Leitsch, Weller, Woltzenlogel Paleo 2008)

There are different equivalent formulations of the notion of lattice:

1. $\langle S, \cap, \cup \rangle$ such that $\cup$ and $\cap$ are commutative, associative, idempotent and $(\forall x)(\forall y) x \cap y = x \iff x \cup y = y$.
2. $\langle S, \cap, \cup \rangle$ such that $\cup$ and $\cap$ are commutative, associative, idempotent and $(\forall x)(\forall y)(x \cap y) \cup x = x \text{ and } (\forall x)(\forall y)(x \cup y) \cap x = x$.
3. A partially ordered set $\langle S, \leq \rangle$ such that $\cap$ is the greatest lower bound and $\cup$ is the least upper bound.

One proves (1) → (2) by proving (1) → (3) and (3) → (2).
Using CERES, a proof of (1) → (2) is obtained where the notion of partially ordered set does not appear.

Successful applications

Example (Baaz, Hetzl, Leitsch, Richter, Spohr 2008)

H. Fürstenberg gave a proof of the infinity of primes by topological means, where the topology is induced by arithmetic progressions over the integers. One of the analytic arguments obtainable from Fürstenberg's proof by CERES is *Euclid's original proof*!

Herbrand sequents

- Sequent calculus proofs are uncomfortable to read.
- By Herbrand's theorem, the essence of first-order proofs are the substitutions used.
- The *Herbrand sequent* summarizes these substitutions.
- The extraction of Herbrand sequents has been implemented in the CERES system.

(Simplified) Herbrand sequent of example proof

$$\begin{array}{l}
f(p_1) = 0 \vee f(p_1) = 1, f(p_2) = 0 \vee f(p_2) = 1, f(p_3) = 0 \vee f(p_3) = 1, \\
f(p_4) = 0 \vee f(p_4) = 1, f(p_5) = 0 \vee f(p_5) = 1, f(p_6) = 0 \vee f(p_6) = 1, \\
f(p_7) = 0 \vee f(p_7) = 1 \\
\vdash \\
p_1 \neq p_2 \wedge f(p_1) = f(p_2), p_3 \neq p_1 \wedge f(p_3) = f(p_1), \\
p_3 \neq p_2 \wedge f(p_3) = f(p_2), p_1 \neq p_4 \wedge f(p_1) = f(p_4), \\
p_5 \neq p_6 \wedge f(p_5) = f(p_6), p_7 \neq p_5 \wedge f(p_7) = f(p_5), \\
p_7 \neq p_6 \wedge f(p_7) = f(p_6), p_4 \neq p_7 \wedge f(p_4) = f(p_7).
\end{array}$$

where the p_i are distinct positions on the tape.
Algorithm can be read off!

Higher order logic

- Proofs can be formalized more naturally and succinctly in higher-order logic.
- Arithmetic is finitely axiomatizable in second-order logic.
- CERES has been theoretically extended to the (weak) class of proofs in second-order logic using only quantifier-free comprehension.
- Extension to full higher-order logic: current work.

Future work

- Extend the scope of the system: Use proofs formalized in Mizar, Coq, Isabelle, . . .
- Use the CERES method to characterize classes of proofs where fast cut-elimination is possible
- Characterize reductive cut-elimination methods (Gentzen, Tait, . . .) as resolution refinements

Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr.
CERES: An Analysis of Fürstenberg's Proof of the Infinity of Primes.
Theoretical Computer Science, 403:160–175, August 2008.

Matthias Baaz and Alexander Leitsch.
Cut-elimination and Redundancy-elimination by Resolution.
Journal of Symbolic Computation, 29(2):149–176, 2000.

Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo.
Herbrand sequent extraction.
In *Intelligent Computer Mathematics*, volume 5144 of *Lecture Notes in Computer Science*, pages 462–477. Springer Berlin, 2008.

Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo.
A clausal approach to proof analysis in second-order logic.
In *Logical Foundations of Computer Science*, volume 5407 of *Lecture Notes in Computer Science*, pages 214–229. Springer Berlin, 2009.