Hierarchical Reflection

TPHOLs, 15 September 2004

Luís Cruz-Filipe

Radboud University of Nijmegen, Netherlands

Center for Logic and Computation, Portugal

(joint work with Freek Wiedijk)

Motivation

Motivation

FTA-project at (then) University of Nijmegen

Motivation

FTA-project at (then) University of Nijmegen

Tactic to prove equalities in fields

Motivation

FTA-project at (then) University of Nijmegen

Tactic to prove equalities in fields

- intuitively “admissible” in simpler structures

Motivation

FTA-project at (then) University of Nijmegen

Tactic to prove equalities in fields

- intuitively “admissible” in simpler structures
- uses partial reflection. . .

Motivation

FTA-project at (then) University of Nijmegen

Tactic to prove equalities in fields

- intuitively “admissible” in simpler structures
- uses partial reflection…

Goal: hierarchy of tactics parallel to hierarchy of structures

Contents

Contents

Contents

Contents

Contents

Contents

Reflection

Reflection

Given: a predicate P on a domain $\mathcal{D}$

Reflection

Given: a predicate P on a domain $\mathcal{D}$

Decision procedure f for (subset $\mathcal{E}$ of) $\mathcal{D}$

Reflection

Given: a predicate P on a domain $\mathcal{D}$

Decision procedure f for (subset $\mathcal{E}$ of) $\mathcal{D}$

such that $f(e) = 1 \iff P(\llbracket e \rrbracket)$

Reflection

Given: a predicate P on a domain $\mathcal{D}$

Decision procedure f for (subset $\mathcal{E}$ of) $\mathcal{D}$

such that $f(e) = 1 \iff P(\llbracket e \rrbracket)$

$$\begin{tikzcd}
\mathcal{E} \arrow[r, "\llbracket\cdot\rrbracket"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & \mathcal{D} \arrow[d, "P"] \\
\mathsf{Prop} \arrow[r, "\mathrm{id}"] & \mathsf{Prop}
\end{tikzcd}$$

Partial Reflection

Partial Reflection

Generalization where $[\![\cdot]\!] : \mathcal{E} \to \mathcal{D}$ is replaced by $]\!] \subseteq \mathcal{E} \times \mathcal{D}$

Partial Reflection

Generalization where $[\![\cdot]\!] : \mathcal{E} \to \mathcal{D}$ is replaced by $]\!] \subseteq \mathcal{E} \times \mathcal{D}$

- not (necessarily) total

Partial Reflection

Generalization where $[\![\cdot]\!] : \mathcal{E} \to \mathcal{D}$ is replaced by $]\!] \subseteq \mathcal{E} \times \mathcal{D}$

- not (necessarily) total
- not (necessarily) functional

Partial Reflection

Generalization where $[\![\cdot]\!] : \mathcal{E} \to \mathcal{D}$ is replaced by $]\!] \subseteq \mathcal{E} \times \mathcal{D}$

- not (necessarily) total
- not (necessarily) functional

used to get past induction-recursion required by

$$[\![e/f]\!] = [\![e]\!]/[\![f]\!]$$

Partial Reflection

Generalization where $[\![\cdot]\!] : \mathcal{E} \to \mathcal{D}$ is replaced by $]\!] \subseteq \mathcal{E} \times \mathcal{D}$

- not (necessarily) total
- not (necessarily) functional

used to get past induction-recursion required by

$$[\![e/f]\!] = [\![e]\!]/[\![f]\!]$$

(we can now write $e \;]\!]\; x \to f \;]\!]\; y \to e/f \;]\!]\; x/y$)

Our tactic: Rational

Our tactic: Rational

In our situation:

Our tactic: Rational

In our situation:

- $\mathcal{D}$ is a field

Our tactic: Rational

In our situation:

- $\mathcal{D}$ is a field
- $P(x, y) := (x = y)$

Our tactic: Rational

In our situation:

- $\mathcal{D}$ is a field
- $P(x, y) := (x = y)$
- f computes $\mathcal{N}(x - y)$ and checks whether it outputs 0

How the normalization works

How the normalization works

For any expression e, $\mathcal{N}(e) = p_1/p_2$

How the normalization works

For any expression e, $\mathcal{N}(e) = p_1/p_2$

p_1 and p_2 are polynomials on some variables fully expanded and sorted

How the normalization works

For any expression e, $\mathcal{N}(e) = p_1/p_2$

p_1 and p_2 are polynomials on some variables fully expanded and sorted

take $p_2 = 1$: we get a tactic for rings!

How the normalization works

For any expression e, $\mathcal{N}(e) = p_1/p_2$

p_1 and p_2 are polynomials on some variables fully expanded and sorted

take $p_2 = 1$: we get a tactic for rings!

(actually even in abelian groups with some extra work. . .)

The big picture

The big picture

$$
\begin{tikzcd}
 & F : \text{Field} \arrow[d] \\
\mathcal{E} \arrow[ur, "\rrbracket^{F}"] \arrow[r, "\rrbracket^{R}"] \arrow[dr, "\rrbracket^{G}"'] & R : \text{Ring} \arrow[d] \\
 & G : \text{Group}
\end{tikzcd}
$$

The big picture

$$
\begin{tikzcd}
 & F : \text{Field} \arrow[d] \\
\mathcal{E} \arrow[ur, "\rrbracket^F"] \arrow[r, "\rrbracket^R"] \arrow[dr, "\rrbracket^G"'] & R : \text{Ring} \arrow[d] \\
 & G : \text{Group}
\end{tikzcd}
$$

$e \times f \ \rrbracket^R x \times y$ but not $e \times f \ \rrbracket^G x \times y$

The big picture

$$\begin{tikzcd} & F : \text{Field} \arrow[d] \\ \mathcal{E} \arrow[ur, "\rrbracket^F"] \arrow[r, "\rrbracket^R"] \arrow[dr, "\rrbracket^G"'] & R : \text{Ring} \arrow[d] \\ & G : \text{Group} \end{tikzcd}$$

$e \times f \ \rrbracket^R x \times y$ but not $e \times f \ \rrbracket^G x \times y$

$e/f \ \rrbracket^F x/y$ but not $e/f \ \rrbracket^G x/y$ or $e/f \ \rrbracket^R x/y$

Good news, bad news

Good news, bad news

The good news…

Good news, bad news

The good news. . .

- it works!

Good news, bad news

The good news...

- it works!
- reuse of around 60% of the code (the type $\mathcal{E}$ and the normalization function $\mathcal{N}$)

Good news, bad news

The good news...

- it works!
- reuse of around 60% of the code (the type $\mathcal{E}$ and the normalization function $\mathcal{N}$)

...and the bad news

Good news, bad news

The good news. . .

- it works!
- reuse of around 60% of the code (the type $\mathcal{E}$ and the normalization function $\mathcal{N}$)

. . . and the bad news

- further unification requires extra axiom

In a perfect world (I)

$$
\begin{tikzcd}
 & F : \text{Field} \arrow[d] \\
\mathcal{E} \arrow[ur, "\rrbracket^{F}"] \arrow[r, "\rrbracket^{R}"] \arrow[dr, "\rrbracket^{G}"] & R : \text{Ring} \arrow[d] \\
 & G : \text{Group}
\end{tikzcd}
\quad \mapsto \quad
\begin{tikzcd}
 & F : \text{Field} \arrow[d] \\
\mathcal{E} \arrow[ur, "\rrbracket"] \arrow[r, "\rrbracket"] \arrow[dr, "\rrbracket"] & R : \text{Ring} \arrow[d] \\
 & G : \text{Group}
\end{tikzcd}
$$

In a perfect world (II)

In a perfect world (II)

Instead of defining $]\!]^G$, $]\!]^R$ and $]\!]^F$ by e.g.

In a perfect world (II)

Instead of defining $\rrbracket^G$, $\rrbracket^R$ and $\rrbracket^F$ by e.g.

$$\begin{aligned} e \rrbracket^G x \;\wedge\; f \rrbracket^G y &\;\Rightarrow\; e + f \rrbracket^G x + y \\ e \rrbracket^F x \;\wedge\; f \rrbracket^F y \;\wedge y \neq 0 &\;\Rightarrow\; e/f \rrbracket^F x/y \end{aligned}$$

In a perfect world (II)

Instead of defining $]\!]^G$, $]\!]^R$ and $]\!]^F$ by e.g.

$$\begin{array}{rcl} e \;]\!]^G \, x \;\wedge\; f \;]\!]^G \, y & \Rightarrow & e+f \;]\!]^G \, x+y \\ e \;]\!]^F \, x \;\wedge\; f \;]\!]^F \, y \;\wedge y \neq 0 & \Rightarrow & e/f \;]\!]^F \, x/y \end{array}$$

define $]\!]^- : \Pi_{A:\mathrm{Setoid}} \mathcal{E} \to A$ s.t.

In a perfect world (II)

Instead of defining $]\!]^G$, $]\!]^R$ and $]\!]^F$ by e.g.

$$
\begin{aligned}
e \;]\!]^G\, x \;\wedge\; f \;]\!]^G\, y \;&\Rightarrow\; e+f \;]\!]^G\, x+y \\
e \;]\!]^F\, x \;\wedge\; f \;]\!]^F\, y \;\wedge y \neq 0 \;&\Rightarrow\; e/f \;]\!]^F\, x/y
\end{aligned}
$$

define $]\!]^- : \Pi_{A:\mathrm{Setoid}} \mathcal{E} \to A$ s.t.

$$
\begin{aligned}
A \text{ is group } \wedge\; e \;]\!]^A\, x \;\wedge\; f \;]\!]^A\, y \;&\Rightarrow\; e+f \;]\!]^A\, x+y \\
A \text{ is field } \wedge\; e \;]\!]^A\, x \;\wedge\; f \;]\!]^A\, y \;\wedge y \neq 0 \;&\Rightarrow\; e/f \;]\!]^A\, x/y
\end{aligned}
$$

using subtyping of algebraic structures.

The K-axiom

The K-axiom

We cannot prove

The K-axiom

We cannot prove

$$e \rrbracket^A a \;\wedge\; e \rrbracket^A b \;\Rightarrow\; a =_A b$$

The K-axiom

We cannot prove

$$e \rrbracket^A a \ \wedge\ e \rrbracket^A b \ \Rightarrow\ a =_A b$$

without the K-axiom

The K-axiom

We cannot prove

$$e \rrbracket^A a \ \wedge \ e \rrbracket^A b \ \Rightarrow \ a =_A b$$

without the K-axiom

$$\langle x, y[x] \rangle = \langle x', y'[x'] \rangle \ \Rightarrow \ x = x' \ \wedge \ y = y'$$

The K-axiom

We cannot prove

$$e \rrbracket^A a \ \wedge \ e \rrbracket^A b \ \Rightarrow \ a =_A b$$

without the K-axiom

$$\langle x, y[x] \rangle = \langle x', y'[x'] \rangle \ \Rightarrow \ x = x' \ \wedge \ y = y'$$

The K-axiom, although consistent with, is not provable within Coq.

Conclusions

Conclusions

- Powerful tactics for equational reasoning

Conclusions

- Powerful tactics for equational reasoning
- Reuse of code for fields, rings and groups

Conclusions

- Powerful tactics for equational reasoning
- Reuse of code for fields, rings and groups
- Improvement possible using K-axiom

Conclusions

- Powerful tactics for equational reasoning
- Reuse of code for fields, rings and groups
- Improvement possible using K-axiom
- (and more on the paper)