

# Hierarchical Reflection

Luís Cruz-Filipe<sup>1,2</sup> and Freek Wiedijk<sup>1</sup>

Logic and Computation Seminar

January 16, 2004

<sup>1</sup>University of Nijmegen, The Netherlands

<sup>2</sup>Centro de Lógica e Computação, Portugal

*From 1.9.2004 the University of Nijmegen will be called Radboud University of Nijmegen*

# Hierarchical Reflection

1. Motivation
2. (Partial) Reflection
3. Normalization Function
4. Uninterpreted Function Symbols
5. Hierarchical Reflection
6. Tighter Integration
7. Conclusions

## Equational Reasoning via (Partial) Reflection

Syntactic expressions:  $E ::= \mathbb{Z} \mid \mathbb{V} \mid E + E \mid E \cdot E \mid E/E$

Normalization function:  $\mathcal{N} : E \rightarrow E$

Interpretation *relation*:  $\llbracket_{\rho} \subseteq E \times A$

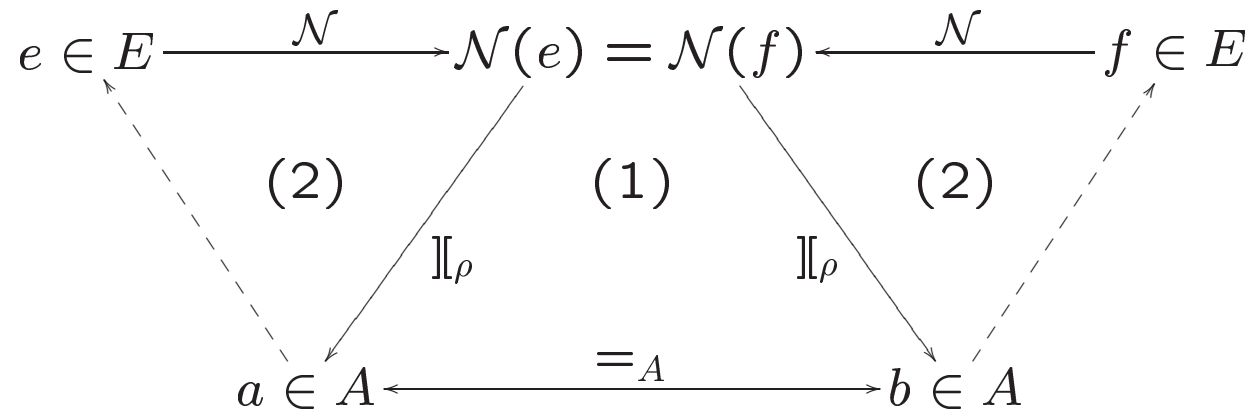
1. well defined:  $e \llbracket_{\rho} a \wedge e \llbracket_{\rho} b \Rightarrow a =_A b$

2.  $\mathcal{N}$  is correct:  $e \llbracket_{\rho} a \Rightarrow \mathcal{N}(e) \llbracket_{\rho} a$

## Equational Reasoning via Partial Reflection (tactic)

$$1. e \llbracket_{\rho} a \wedge e \llbracket_{\rho} b \Rightarrow a =_A b$$

$$2. e \llbracket_{\rho} a \Rightarrow \mathcal{N}(e) \llbracket_{\rho} a$$



$$1'. e \llbracket_{\rho} a \wedge e = 0/e' \Rightarrow a =_A 0$$

## Normalization Function

$$\begin{aligned} F & ::= P/P \\ P & ::= M + P \mid \mathbb{Z} \\ M & ::= \mathbb{V} \cdot M \mid \mathbb{Z} \end{aligned}$$

$M$  are “lists of variables” ( $\cdot$  is “cons”, integers are “nil”)

$P$  are “lists of monomials” ( $+$  is “cons”, integers are “nil”)

Normal forms are formal “quotients of sorted lists” without duplication:

$$\mathcal{N} \left( \frac{1}{x-y} + \frac{1}{x+y} \right) = \frac{x \cdot 2 + 0}{x \cdot x \cdot 1 + y \cdot y \cdot (-1) + 0}$$

## Normalization Function (definition)

Recursively defined functions

$$\begin{aligned} - \cdot_{MZ} - & : M \times \mathbb{Z} \rightarrow M \\ - \cdot_{MV} - & : M \times \mathbb{V} \rightarrow M \\ - \cdot_{MM} - & : M \times M \rightarrow M \\ - \dagger_{MM} - & : M \times M \rightarrow M \\ - \dagger_{PM} - & : P \times M \rightarrow P \\ - \dagger_{PP} - & : P \times P \rightarrow P \\ - \cdot_{PM} - & : P \times M \rightarrow P \\ - \cdot_{PP} - & : P \times P \rightarrow P \\ - \dagger_{FF} - & : F \times F \rightarrow F \\ - \cdot_{FF} - & : F \times F \rightarrow F \\ - /_{FF} - & : F \times F \rightarrow F \end{aligned}$$

## Normalization Function (examples)

$$e \cdot_{MM} f := \begin{cases} (e_2 \cdot_{MM} f) \cdot_{MV} e_1 & \text{if } e = e_1 \cdot e_2 \\ f \cdot_{MZ} i & \text{if } e = i \in \mathbb{Z} \end{cases}$$

$$e \dagger_{PM} f := \begin{cases} j \dagger_{MM} i & \text{if } e = j \in \mathbb{Z}, f = i \in \mathbb{Z} \\ f \dagger i & \text{if } e = i \in \mathbb{Z} \\ e_1 \dagger (e_2 \dagger_{PM} i) & \text{if } e = e_1 \dagger e_2, f = i \in \mathbb{Z} \\ e_2 \dagger_{PM} (e_1 \dagger_{MM} f) & \text{if } e = e_1 \dagger e_2, e_1 =_M f \\ e_1 \dagger (e_2 \dagger_{PM} f) & \text{if } e = e_1 \dagger e_2, e_1 <_{\text{lex}} f \\ f \dagger e & \text{if } e = e_1 \dagger e_2, e_1 >_{\text{lex}} f \end{cases}$$

$$\mathcal{N}(e/f) := N(e) /_{FF} N(f)$$

$$\mathcal{N}(v) := \frac{v \cdot 1 \dagger 0}{1}$$

## Uninterpreted Function Symbols

Goal:  $f(a + b) = f(b + a)$

$$f(a + b) \rightsquigarrow x, f(b + a) \rightsquigarrow y, \mathcal{N}(x - y) = \frac{x \cdot 1 + y \cdot (-1) + 0}{1}$$

Solution: extend  $E$  with  $\mathbb{V}_1 : E \rightarrow E$

$$E ::= \mathbb{Z} \mid \mathbb{V}_0 \mid \mathbb{V}_1(E) \mid E + E \mid E \cdot E \mid E/E$$

Normal forms:

$$F ::= P/P$$

$$P ::= M + P \mid \mathbb{Z}$$

$$M ::= \mathbb{V}_0 \cdot M \mid \mathbb{V}_1(F) \cdot M \mid \mathbb{Z}$$

ordered...



## Uninterpreted Function Symbols (order)

Ordering on  $E$  (assumes  $<_{\mathbb{V}_0}$  on  $\mathbb{V}_0$  and  $<_{\mathbb{V}_1}$  on  $\mathbb{V}_1$ ):

$$x <_E i <_E e \dagger f <_E e \cdot f <_E e/f <_E v(e)$$

Expressions with the same operator are sorted lexicographically.

Example (with  $x <_{\mathbb{V}_0} y$  and  $u <_{\mathbb{V}_1} v$ ):

$$x <_E y <_E 34 <_E x/4 <_E u(x + 3) <_E u(2 \cdot y) <_E v(x + 3)$$

Same normalization function with added rule

$$\mathcal{N}(v(e)) := \frac{v(\mathcal{N}(e)) \cdot 1 \dagger 0}{1}$$

## Uninterpreted Function Symbols (valuations)

Two valuations  $\rho_0 : \mathbb{V}_0 \rightarrow A$  and  $\rho_1 : \mathbb{V}_1 \rightarrow (A \rightarrow A)$

Once again, one can prove

$$\begin{aligned} e \Vdash_{\rho_0, \rho_1} a \wedge e \Vdash_{\rho_0, \rho_1} b &\Rightarrow a =_A b \\ e \Vdash_{\rho_0, \rho_1} a &\Rightarrow \mathcal{N}(e) \Vdash_{\rho_0, \rho_1} a \end{aligned}$$

Goal:  $f(a + b) = f(b + a)$

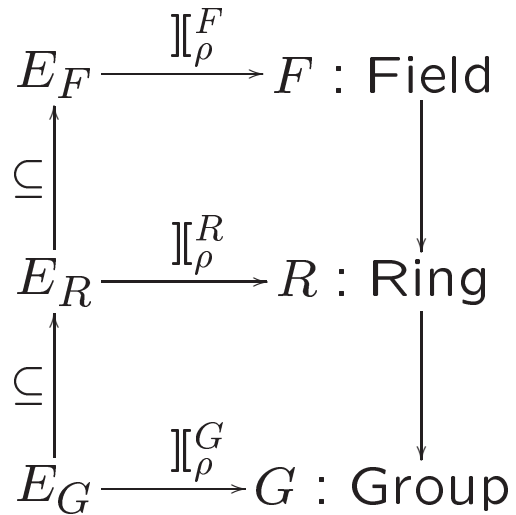
$$f \rightsquigarrow v, a \rightsquigarrow x, b \rightsquigarrow y$$

$$\mathcal{N}(v(x + y)) = \mathcal{N}(v(y + x)) = \frac{v\left(\frac{x \cdot 1 + y \cdot 1 + 0}{1}\right) \cdot 1 + 0}{1}$$

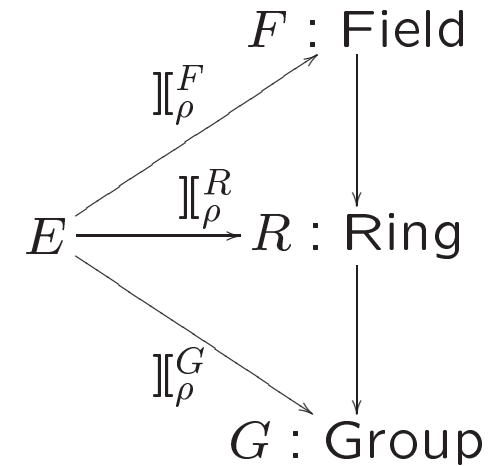
Binary functions, partial functions similarly treated.

## Hierarchical Reflection

Similar procedures for other structures?



but better is



making use of the *partiality* of the interpretation

## Hierarchical Reflection (interpretation relations)

But...

If  $\rho(x) = a$ , then  $a + a$  is represented by  $x + x$ , but

$$\mathcal{N}(x + x) = \frac{x \cdot 2 + 0}{1} \llbracket_{\rho}^G a + a$$

does not hold.

We need to interpret  $e/1$  and  $e \cdot i$  when we can interpret  $e$

## Hierarchical Reflection (interpretation relations)

	$\llbracket_{\rho}^G$	$\llbracket_{\rho}^R$	$\llbracket_{\rho}^F$
$v \in \mathbb{V}$	yes	yes	yes
$i \in \mathbb{Z}$	if $i = 0$	yes	yes
$e + f$	yes	yes	yes
$e \cdot f$	if $f \in \mathbb{Z}$	yes	yes
$e/f$	if $f = 1$	if $f = 1$	if $f \neq 0$

In the last three cases the additional requirement that  $e$  (and eventually  $f$ ) be interpreted is implicit.

## Hierarchical Reflection (correctness)

To prove

$$e \llbracket_{\rho}^G a \Rightarrow \mathcal{N}(e) \llbracket_{\rho}^G a$$

one needs to use the knowledge that the auxiliary functions will only be applied to the “right” arguments.

For example, correctness of  $\cdot_{MM}$  w.r.t.  $\llbracket_{\rho}^F$  states that

$$e \llbracket_{\rho}^F a \wedge f \llbracket_{\rho}^F b \Rightarrow e \cdot_{MM} f \llbracket_{\rho}^F a \cdot b$$

but  $a \cdot b$  has no meaning in a group!

## Hierarchical Reflection (correctness)

However,

$$e \llbracket_{\rho}^F a \wedge f \llbracket_{\rho}^F b \Rightarrow e \cdot_{MM} f \llbracket_{\rho}^F a \cdot b$$

is equivalent to

$$e \cdot f \llbracket_{\rho}^F a \cdot b \Rightarrow e \cdot_{MM} f \llbracket_{\rho}^F a \cdot b$$

and the same property w.r.t.  $\llbracket_{\rho}^G$  can be written down as

$$e \cdot f \llbracket_{\rho}^G c \vee f \cdot e \llbracket_{\rho}^G c \Rightarrow e \cdot_{MM} f \llbracket_{\rho}^G c$$

(the disjunction is needed because  $\cdot_{MM}$  can swap the order of its arguments)

## Hierarchical Reflection (optimization for rings and groups)

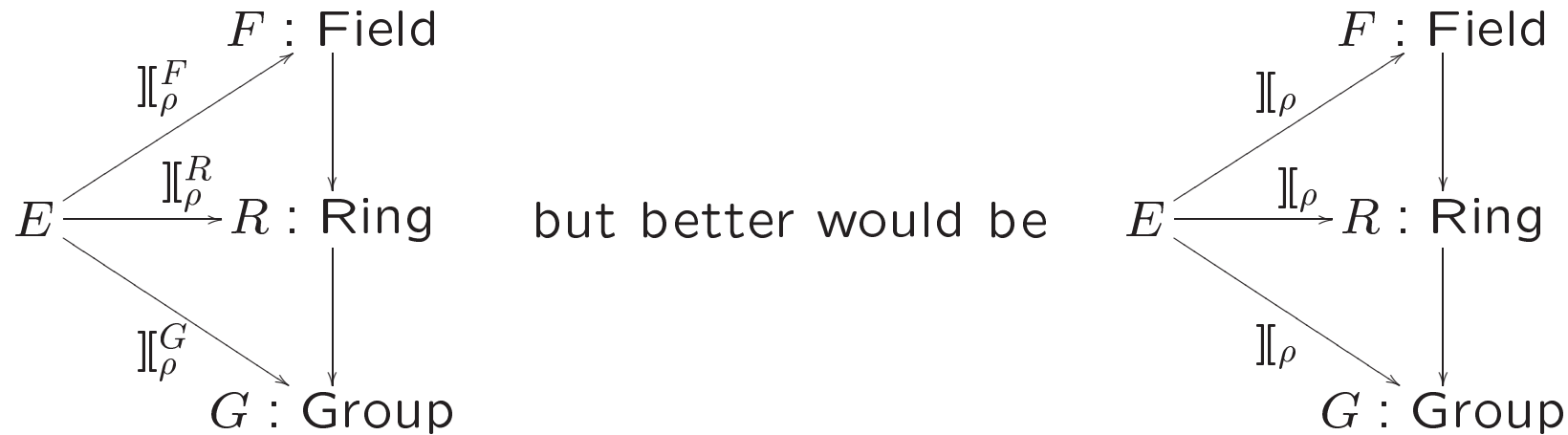
To avoid divisions by 1, one can forget about the type  $F$  altogether and define  $\mathcal{N}_R$  directly using  $\cdot_{MM}$  and the like; the base case now looks like

$$\mathcal{N}(v) := v \cdot 1 + 0$$

Also, in groups and rings normal forms *are* unique, so the last subtraction can also be avoided.



## Tighter Integration?



The first requires *all* functions  $\dagger_{MM}$ ,  $\cdot_{MM}$ , etc. to be proved correct w.r.t.  $\llbracket_{\rho}^G$ ,  $\llbracket_{\rho}^R$  and  $\llbracket_{\rho}^F$ .

Most of these proofs are (almost) the same, yet they cannot be reused!

## Tighter Integration?

Instead of defining  $\llbracket_{\rho}^G$ ,  $\llbracket_{\rho}^R$  and  $\llbracket_{\rho}^F$  by e.g.

$$\begin{aligned}e \llbracket_{\rho}^G x \wedge f \llbracket_{\rho}^G y &\Rightarrow e + f \llbracket_{\rho}^G x + y \\e \llbracket_{\rho}^R x \wedge f \llbracket_{\rho}^R y &\Rightarrow e \cdot f \llbracket_{\rho}^R x \cdot y \\e \llbracket_{\rho}^F x \wedge f \llbracket_{\rho}^F y \wedge y \neq 0 &\Rightarrow e/f \llbracket_{\rho}^F x/y\end{aligned}$$

define  $\llbracket_{\rho}^- : \prod_{A:\text{Setoid}} E \rightarrow A$  s.t.

$$\begin{aligned}A \text{ is group} \wedge e \llbracket_{\rho}^A x \wedge f \llbracket_{\rho}^A y &\Rightarrow e + f \llbracket_{\rho}^A x + y \\A \text{ is ring} \wedge e \llbracket_{\rho}^A x \wedge f \llbracket_{\rho}^A y &\Rightarrow e \cdot f \llbracket_{\rho}^A x \cdot y \\A \text{ is field} \wedge e \llbracket_{\rho}^A x \wedge f \llbracket_{\rho}^A y \wedge y \neq 0 &\Rightarrow e/f \llbracket_{\rho}^A x/y\end{aligned}$$

using subtyping of algebraic structures.

## Tighter Integration (the bad news)

Does not work!

Proving

$$e \Vdash_{\rho}^A a \wedge e \Vdash_{\rho}^A b \Rightarrow a =_A b$$

requires a strong induction principle — 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

- Powerful tactics for equational reasoning
- Can now deal with functions e.g. absolute value on  $\mathbb{R}$
- Reuse of code for fields, rings and groups
- Improvement possible using  $K$ -axiom