Restoring Natural Language as a Computerised Mathematics Input Method

Robert Lamar

joint work with
Fairouz Kamareddine   Manuel Maarek   J. B. Wells

ULTRA group, Heriot-Watt University
http://www.macs.hw.ac.uk/ultra/

30 June 2007
Mathematical Knowledge Management conference
RISC, Hagenburg, Austria

A Bit of Mathematics

CML There is an element $-a$ in R such that $a + (-a) = 0$ for all a in R.

A Bit of Mathematics

In a variety of languages

CML There is an element $-a$ in R such that $a + (-a) = 0$ for all a in R.

Mizar `ex b being Element of R st a + b = 0.`

Isar $a \in R \Longrightarrow -a \in R \wedge a + -a = 0$.

Omega `(inverse-exist R op (struct-unit R op)).`

A Bit of Mathematics

In a variety of languages

CML	There is an element $-a$ in R such that $a + (-a) = 0$ for all a in R.
Mizar	`ex b being Element of R st a + b = 0.`
Isar	$a \in R \Longrightarrow$ `-a` $\in R \wedge$ `a + -a = 0`.
Omega	`(inverse-exist R op (struct-unit R op)).`

Goal: Smoothing and strengthening transitions.

MathLang grammatical categories

term	Common mathematical objects.	"R", "0", "$a + b$"
set	Sets of mathematical objects.	"$\mathbb{R}$"
noun	Categories to classify **terms**	"ring"
adjective	Modifiers for **nouns**	"Abelian"
statement	Assertions of truth	"$a = b$"
declaration	Type signature designations	"Addition is denoted $a + b$"
definition	New symbol introductions	"A ring is. . ."
step	A group of mathematical assertions.	"We have. . . . . . and also. . ."
context	Assertions preliminary to a **step**	"Given a ring R, . . ."

Example

There is an element $-a$ in R such that

$a + (-a) = 0$ for all a in R.

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

a + (-a) = 0 for all a in R .

Example

There is an element -a in R such that

$a + (-a) = 0$ for all a in R.

Example

There is an element -a in R such that a + (-a) = 0 for all a in R.

Example

There is an element -a in R such that
a + (-a) = 0 for all a in R.

Example

There is an element -a in R such that

$a + (-a) = 0$ for all a in R.

Anatomy of a box

<interp> contents

Color	Grammatical category
Contents	Original mathematics
<interp>	Logical interpretation

Anatomy of a box

<interp> contents

Color	Grammatical category
Contents	Original mathematics
<interp>	Logical interpretation

Examples:

<equal> a is equal to b

<equal> $a = b$

<ident> $a = b$

<A> A <reals> $\mathbb{R}$

<inter> <apples> A $\cap$ <oranges> O

Another Example

$$a = b = c$$

Another Example

$$\boxed{a} = \boxed{b} = \boxed{c}$$

Another Example

$$a = b = c$$

Another Example

$$a = b = c$$

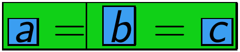
What next?

Another Example

$$a = b = c$$

?

Another Example



?

What does this mean?

How do we cope?

$$a = b = c$$

What does this mean?

How do we cope?

$$a = b = c$$

- Compound statement
- Short for "$a = b$ and $b = c$"
- Must be translated before computerisation

Syntax sugaring:

- Common in many computer languages
- Used for pretty-printing
- Eases human use of languages
- Always: nice for computers $\longrightarrow$ nice for humans

Syntax sugaring:

- Common in many computer languages
- Used for pretty-printing
- Eases human use of languages
- Always: nice for computers $\longrightarrow$ nice for humans

Syntax souring:

- A new transformation: Syntax souring
- Syntax souring solves the problem of $a = b = c$.
- Other direction: nice for humans $\longrightarrow$ nice for computers

The ULTRA Solution

Another look at the problem

$$a = b = c$$

- The relation "$=$" is binary: it takes *two* arguments
- The term "b":
  - Appears only once
  - Is actually provided as argument twice
  - Is "shared"
- **Goal:** tell $=$ to be nice and share

The ULTRA Solution

$$a = \qquad\qquad b \quad = c$$

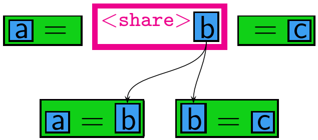
The ULTRA Solution

$$\boxed{a} = \qquad \boxed{b} \quad = \boxed{c}$$

The ULTRA Solution

The ULTRA Solution

The ULTRA Solution



Kinds of Souring

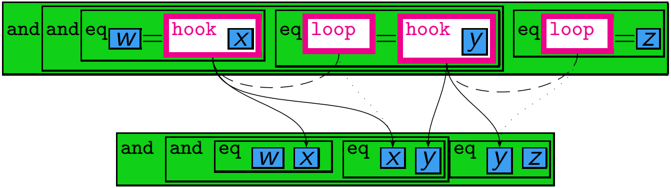
share	Natural splitting of single argument	}	Duplication
chain	More flexible forwarding of entities		
fold	Recursion upon lists	}	List operations
map	Iteration over lists		
position	Reordering of arguments	}	Reordering

Kinds of Souring: Duplication

share • chain • fold • map • position

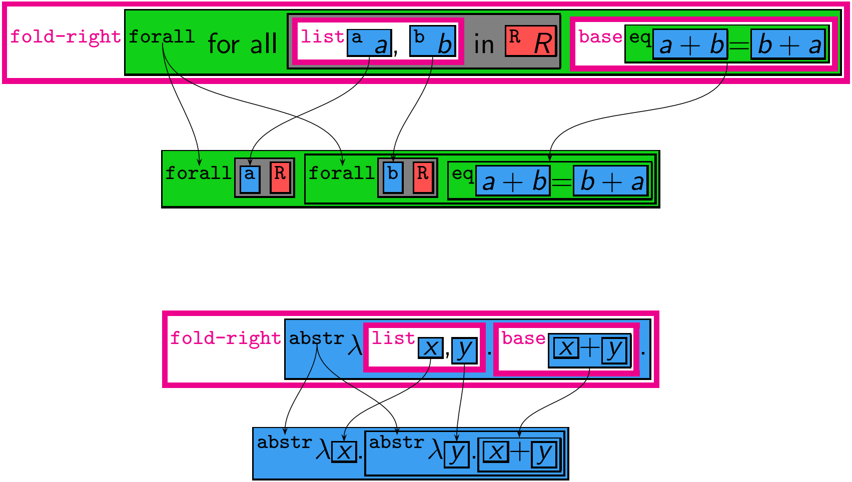
Kinds of Souring: Duplication

share • chain • fold • map • position



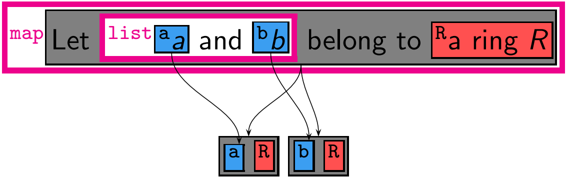
Kinds of Souring: Lists

share • chain • **fold** • map • position



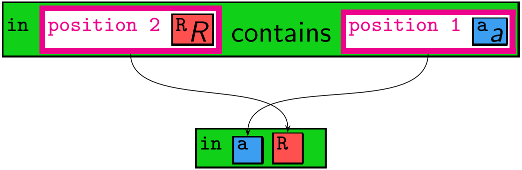
Kinds of Souring: Lists

share • chain • fold • map • position



Kinds of Souring: Reordering

share • chain • fold • map • position



Conclusion

- Five kinds of souring:

  `share` • `chain` • `fold` • `map` • `position`
- Common goal: elucidating the intent of language

- Future Work:
  - Look for other souring needs
  - Automate the annotation process
  - Identify appropriate granularity for annotation
  - Arrive at recommendations/conventions for annotation
  - Cope with ellipsis.

$$\overbrace{x + \ldots + x}^{n \text{ times}} \qquad 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \qquad \frac{1}{1 + \frac{1}{1+\cdots}}$$

Text and Symbol

Box Annotation

Souring annotation

Souring Examples

Conclusion