universität innsbruck

Computational Logic

Isabelle Import for Mizar

Cezary Kaliszyk   Karol Pąk

CICM'18, Hagenberg

Proof Interoperability

Proof Analysis

- Comparing, Presentation, Search...

Proof Auditing

- HOL Zero

Re-use and Combining

- Particularly useful if shallow

Mizar

Proof Assistant

- Many features quite different from the usual
- Developed by mathematicians for mathematicians
- Initially as a type-setting system

Mizar

Proof Assistant

- Many features quite different from the usual
- Developed by mathematicians for mathematicians
- Initially as a type-setting system

Math type-setting system (1971)

- Extended to check proofs (in 1973)
- Consistent library of formalized Math (1980s)

Natural deduction

- Stays as long as possible in first-order logic

Mizar

Proof Assistant

- Many features quite different from the usual
- Developed by mathematicians for mathematicians
- Initially as a type-setting system

Math type-setting system (1971)

- Extended to check proofs (in 1973)
- Consistent library of formalized Math (1980s)

Natural deduction

- Stays as long as possible in first-order logic

Foundations

- Set Theory (with universes, rarely used)
- Dependent soft type system and type inference mechanism

even natural number bijective Function of A,B

Other Mizar features

Rich input language and LaTeX generation

- Contextual parsing: more than 100 meanings of "+"
- Journal of Formalized Mathematics

Focus on mathematics

- A lot not covered elsewhere (lattices)
- Much less computer related proofs (random access Turing machines)

The system has evolved

- unfortunately many features have not changed since the 1980s...

Can we express it all in a modern logical framework?

Isabelle from our point of view

The good

- Easy to define a new object logic and its basic components
- Isar inspired by Mizar, and so similar to it
- Some powerful automation
- Small(ish) kernel, easy to extend by ML

The bad

- A lot of features optimized for HOL (foundations, notations, auto..)
- Isabelle/FOL is rather poor
- Notation language is limited
- Speed issues

The ugly

- Need lots of ML code: background knowledge, types, definitions, ...
- Isar not as good as Mizar's proof language

Encoding the Mizar foundations in Isabelle

We can start with Isabelle/FOL

- Features beyond first-order can be encoded in the logical framework
- Added some hacks to allow switching to Isabelle/HOL

Define the meta-types

- Isabelle types of Mizar sets and types
- Set equality and set membership introduced
- Type definition and membership axiomatized

Soft type system with dependent types and intersection types

- even natural number
- bijective Function of A,B

Tarski-Grothendieck Set Theory

reserve *x,y,z,u,a* **for** *object*
reserve *M,N,X,Y,Z* **for** *set*

— Set axiom
theorem *tarski_0_1*:
 $\forall x.\ x$ *be set* **using** *SET_def* **by** *simp*

— Extensionality axiom
axiomatization where *tarski_0_2*:
 $\forall X.\ \forall Y.\ (\forall x.\ x\ in\ X \longleftrightarrow x\ in\ Y)$
 $\longrightarrow X = Y$

— Axiom of pair
axiomatization where *tarski_0_3*:
 $\forall x.\ \forall y.\ \exists Z.\ \forall a.$
 $a\ in\ Z \longleftrightarrow a = x \lor a = y$

— Axiom of union
axiomatization where *tarski_0_4*:
 $\forall X.\ \exists Z.\ \forall x.$
 $x\ in\ Z \longleftrightarrow (\exists Y.\ x\ in\ Y \land Y\ in\ X)$

— Axiom of regularity
axiomatization where *tarski_0_5*:
 $\forall x.\ \forall X.\ x\ in\ X \longrightarrow (\exists Y.\ Y\ in\ X \land$
 $\neg(\exists z.\ z\ in\ X \land z\ in\ Y))$

Tarski-Grothendieck Set Theory

reserve *x,y,z,u,a* **for** *object*
reserve *M,N,X,Y,Z* **for** *set*

— Set axiom
theorem *tarski_0_1*:
$\forall x.\ x$ *be set* **using** *SET_def* **by** *simp*

— Extensionality axiom
axiomatization where *tarski_0_2*:
$\forall X.\ \forall Y.\ (\forall x.\ x \textit{ in } X \longleftrightarrow x \textit{ in } Y)$
$\longrightarrow X = Y$

— Axiom of pair
axiomatization where *tarski_0_3*:
$\forall x.\ \forall y.\ \exists Z.\ \forall a.$
$a \textit{ in } Z \longleftrightarrow a = x \vee a = y$

— Axiom of union
axiomatization where *tarski_0_4*:
$\forall X.\ \exists Z.\ \forall x.$
$x \textit{ in } Z \longleftrightarrow (\exists Y.\ x \textit{ in } Y \wedge Y \textit{ in } X)$

— Axiom of regularity
axiomatization where *tarski_0_5*:
$\forall x.\ \forall X.\ x \textit{ in } X \longrightarrow (\exists Y.\ Y \textit{ in } X \wedge$
$\neg(\exists z.\ z \textit{ in } X \wedge z \textit{ in } Y))$

differences: quantification, types, parentheses, schemes

Support for Mizar Definitions

Conditional Definitions

Definitions by “means”

Type definitions

Structures

Simple definition package

- Core definitions
- User obligations
- Derived properties

Definitions

mdef *tarski_def_1* ({_}) **where**
mlet *y be object*
func $\{y\} \rightarrow$ *set means* $\lambda it.$
$\forall x.\ x$ *in it* $\longleftrightarrow x = y$

mdef *tarski_def_4* (*union* _) **where**
mlet *X be set*
func union $X \rightarrow$ *set means* $\lambda it.$
$\forall x.\ x$ *in it* $\longleftrightarrow (\exists Y.\ x$ *in* $Y \wedge Y$ *in* $X)$

mdef *xboole_0_def_2* ({}) **where**
func $\{\} \rightarrow$ *set equals*
the empty|set

Tuples: Consider the ring structure: $\langle R, +, \mathbf{0}, \cdot, \mathbf{1} \rangle$

Tuples: Consider the ring structure: $\langle R, +, \mathbf{0}, \cdot, \mathbf{1} \rangle$

Modeled as partial functions:

mdefinition *doubleLoopStr_d*(*doubleLoopStr*) **where**
struct doubleLoopStr (#
  carrier → (λ*S*. *set*);
  addF → (λ*S*. *BinOp-of the carrier of S*);
  ZeroF → (λ*S*. *Element-of the carrier of S*);
  multF → (λ*S*. *BinOp-of the carrier of S*);
  OneF → (λ*S*. *Element-of the carrier of S*)
#) : *struct_well_defined...*

Tuples: Consider the ring structure: $\langle R, +, \mathbf{0}, \cdot, \mathbf{1} \rangle$

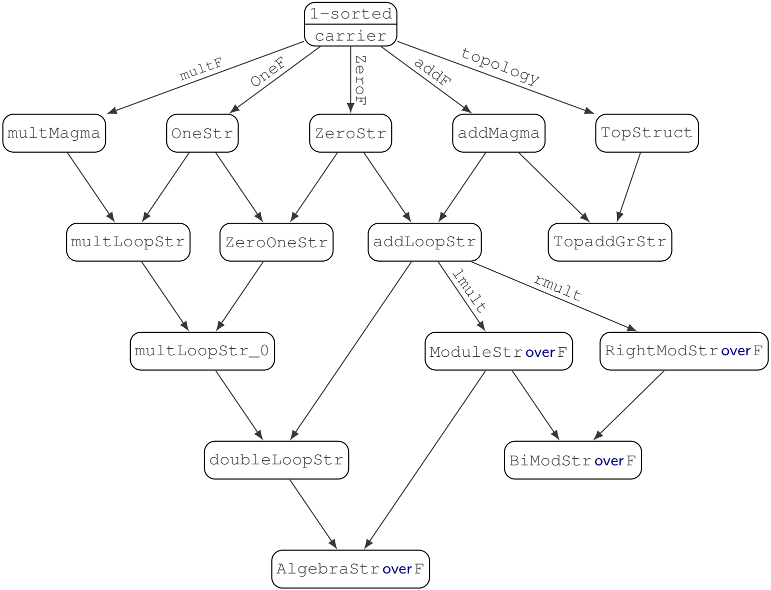
Modeled as partial functions:

mdefinition *doubleLoopStr_d*(*doubleLoopStr*) **where**
struct doubleLoopStr (#
  carrier $\rightarrow$ (λS. *set*);
  addF $\rightarrow$ (λS. *BinOp-of the carrier of S*);
  ZeroF $\rightarrow$ (λS. *Element-of the carrier of S*);
  multF $\rightarrow$ (λS. *BinOp-of the carrier of S*);
  OneF $\rightarrow$ (λS. *Element-of the carrier of S*)
#) : *struct_well_defined...*

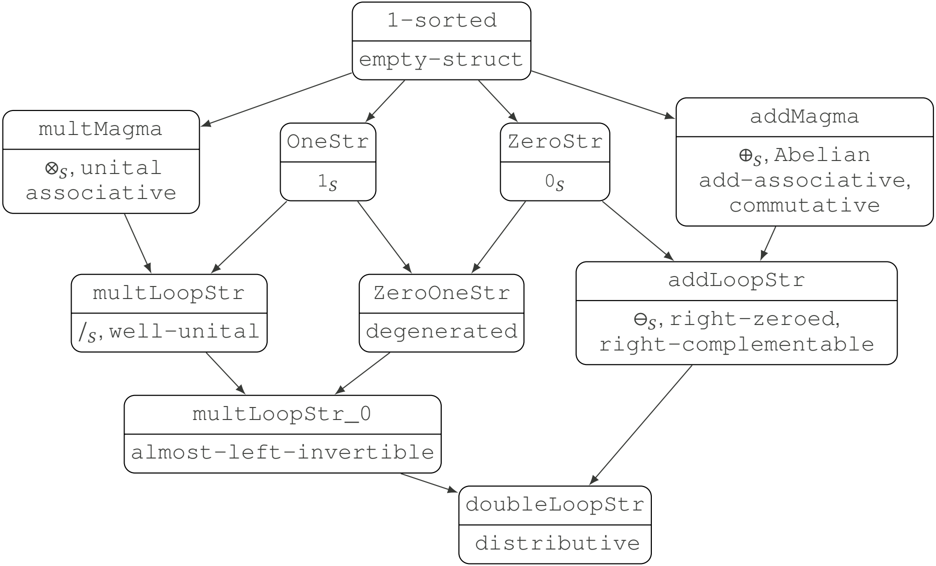
Actual Ring

abbreviation
Ring $\equiv$ *Abelian* | *add-associative* | *right-zeroed* |
right-complementable | *associative* |
well-unital | *distributive* |
non empty-struct | *doubleLoopStr*

Lattice of basic algebraic structures in Mizar



Lattice of basic algebraic structures in Mizar



Example: Algebra

```
reserve G for Group
reserve h,g for Element-of-struct G

mtheorem group_1_th_16:
  (h ⊗_G g)^-1_G = g^-1_G ⊗_G h^-1_G
proof-
  have (g^-1_G ⊗_G h^-1_G) ⊗_G (h ⊗_G g)
              = (g^-1_G ⊗_G h^-1_G) ⊗_G h ⊗_G g
    using group_1_def_3E[of _ _ h] by mauto
  also have ... = g^-1_G ⊗_G (h^-1_G ⊗_G h) ⊗_G g
    using group_1_def_3E by mty auto
  also have ... = g^-1_G ⊗_G 1._G ⊗_G g
    using group_1_def_5 by mauto
  also have ... = (g^-1_G)  ⊗_G g
    using group_1_def_4 by mauto
  also have ... = 1._G
    using group_1_def_5 by mauto
  finally show ?thesis
    using group_1_th_11[of _ h ⊗_G g,
    THEN conjunct1] by mauto
```

Examples (2/2)

Ordinals

theorem *ordinal_2_sch_19*:
  assumes [*ty*]: *a is Nat*
    and *A1*: $P(\{\})$
    and *A2*: $\forall\, n : Nat.\ P(n) \longrightarrow P(succ\ n)$
  shows $P(a)$

Examples (2/2)

Ordinals

theorem *ordinal_2_sch_19*:
assumes [*ty*]: *a is Nat*
and *A1*: $P(\{\})$
and *A2*: $\forall\, n : Nat.\ P(n) \longrightarrow P(succ\ n)$
shows $P(a)$

Turing Machines

theorem *extpro_1*:
assumes [*ty*]: *N be with_zero* | *set*
shows $halt_{\textit{Trivial-AMI N}}$ *is halting Trivial-AMI N, N*

Mizar's knowledge hard to access. Syntax in WSX:

```
<Proposition>
 <Label idnr= 0  spelling=   line= 27  col= 5 />
 <Universal–Quantifier–Formula line= 27  col= 5 >
  <Explicitly–Qualified–Segment line= 27  col= 5 >
   <Variables>
    <Variable idnr= 2  spelling= x  line= 27  col= 7 />
   </Variables>
   <Standard–Type nr= 2  spelling= object  line= 27  col= 20 />
  </Explicitly–Qualified–Segment>
  <Qualifying–Formula line= 27  col= 35 >
   <Simple–Term idnr= 2  spelling= x  line= 27  col= 28 />
   <Standard–Type nr= 1  spelling= set  line= 27  col= 35 />
  </Qualifying–Formula>
 </Universal–Quantifier–Formula>
</Proposition>
```

Semantics spread across files from different stages

tarski.xml

```
<Proposition line= 27  col= 35 >
 <For pid= 0  vid= 2 >
  <Typ kind= M  nr= 1  pid= 1 ><Cluster/><Cluster/></Typ>
  <Is>
   <Var nr= 1 />
   <Typ kind= M  nr= 2  pid= 2 ><Cluster/><Cluster/></Typ>
  </Is>
 </For>
</Proposition>
```

Semantics spread across files from different stages

tarski.xml

```
<Proposition line= 27 col= 35 >
 <For pid= 0 vid= 2 >
  <Typ kind= M nr= 1 pid= 1 ><Cluster/><Cluster/></Typ>
  <Is>
   <Var nr= 1 />
   <Typ kind= M nr= 2 pid= 2 ><Cluster/><Cluster/></Typ>
  </Is>
 </For>
</Proposition>
```

tarski.idx

```
<Symbol kind= I nr= 2 name= x />
```

tarski.eno

```
<Pattern kind= M nr= 1 aid= HIDDEN formatnr= 2 constrkind= M
   constrnr= 1 relnr= 1 >
```

tarski.frm

```
<Format kind= M nr= 2 symbolnr= 2 argnr= 0 />
```

tarski.dcx

```
<Symbol kind= M nr= 2 name= object />
```

Combined Syntactic-Semantic Representation

All syntactic nodes correctly identified with their semantic content

All background knowledge listed (thesis, ...)

Proof structure closer to natural deduction

Separation of meta-logic from set theory

Semi-Automated Translation

Export combined syntactic-semantic Mizar

Isabelle can import first 100 MML articles

All definitions, theorems, user typing rules

- So far the proofs are assumed in the import
- Intermediate steps already in the Mis files

Usable environment for (further) proof development

- Type inference

Usable Environment: NEWTON

mdef newton_def_1 (_ - [90,0]91) **where**
mlet x is Complex,
n is natural|Number
func x^n → number equals Π (n ↦ x)

Usable Environment: NEWTON

mdef newton_def_1 (_ ^ [90,0]91) **where**
mlet x is Complex,
n is natural|Number
func x^n → number equals Π (n ↦ x)

Basic properties of the power operator

mtheorem newton_th_4:
$z^{0} = 1$

mtheorem newton_th_5:
$z^{1} = z$

mtheorem newton_th_6:
$z^{s +_{\mathbb{N}} 1} = z^{s} *_{\mathbb{C}} z$

mtheorem newton_th_7:
$(x *_{\mathbb{C}} y)^{s} = x^{s} *_{\mathbb{C}} y^{s}$

mtheorem newton_th_8:
$x^{s +_{\mathbb{C}} t} = x^{s} *_{\mathbb{C}} x^{t}$

mtheorem newton_th_9:
$(x^{s})^{t} = x^{(s *_{\mathbb{C}} t)}$

Isabelle/Mizar features interesting for formalization

Familiar mathematical foundations

Convenient proof style

Curated the library

In a modern logical framework

But: A lot of convenience and features of Mizar missing