Generalizing theorems of the **mizar** Mathematical Library by type promotion and property omission

Jesse Alama
Center for Artificial Intelligence,
New University of Lisbon, Portugal

Logical Frameworks and Metalanguages: Theory and Practice/Modules and Libraries for Proof Assistants 2011

2011-08-26

Motivation: making implicit information of computer-assisted proofs explicit

- An experiment involving a large library of formal mathematics to bring out unarticulated inferences.
- Looking for cases where information that is implicitly supplied by an ITP is actually not needed.
- Applications:
  - Discovering necessary and sufficient conditions for theorems; reverse mathematics 'in the small'.
  - Library refactorings.
  - AI: Inventing generalizations of theorems/definitions.

mizar

- ITP based on first-order logic, natural deduction (declarative proofs) and classical set theory
- Large library:
  - 1132 articles
  - More than 2 million lines of text
  - Several tens of thousands of theorems and definitions
  - Devoted primarily to pure mathematics

Constructor properties in **mizar**

‘Constructor’ is a **mizar** term that covers the concepts **function/constant**, **relation**, **structure**, ...

Relations	**Functions**	**Structures**
reflexivity	projectivity	abstractness
symmetry	involutiveness	
asymmetry	idempotence	
connectedness	commutativity	
irreflexivity		

Properties in first-order logic

- Reflexivity: $R(x,x)$
- Symmetry: $R(x,y) \to R(y,x)$
- Asymmetry $R(x,y) \to \neg R(y,x)$
- Connectedness: $R(x,y) \vee R(y,x)$
- Irreflexivity: $\neg R(x,x)$
- Projectivity: $f(f(x)) = f(x)$
- Involutiveness: $f(f(x)) = x$
- Idempotence: $f(x,x) = x$
- Commutativity: $f(x,y) = f(y,x)$
- Abstractness: "A structure is determined by its selectors/slots/fields"

Eliciting dependence upon constructor properties

- Decompose the 1100+ articles of the **mizar** Mathematical Library, yielding about 100,000 self-standing ‘microarticles’.
- Verification of an item is a two-step process:
  1. Construct the environment in which the verification proper will be carried out. (Run BibTeX, creating a `.bbl` file that will be used later.)
  2. Carry out the verification proper of the article with respect to the constructed environment. (Run LaTeX with the `.bbl` file.)
- Step (1) imports all properties of all imported constructors. (`\nocite{*}`)
- Between step (1) and (2), omit a constructor property. (Delete an entry from a `.bbl` file ‘behind the back’ of LaTeX, then see whether one’s document compiles cleanly.)

Most needed properties

Property	Reliant items	Non-reliant items
reflexivity	54113	52736 (39859 excluding equality)
symmetry	29744	40817 (10854 excluding equality)
asymmetry	256	35285
connectedness	5020	12876
irreflexivity	91	974
projectivity	153	15201
involutiveness	533	9707
idempotence	535	17627
commutativity	14055	472621
abstractness	1935	3764

Statistics

\#1: $\leq$ on extended real numbers.

- 4856 items implicitly rely on $\leq$'s **connectedness**
- 4606 items implicitly rely on $\leq$'s **reflexivity**

\#2: $\subseteq$ on sets

- 2037 items implicitly rely on $\subseteq$'s **reflexivity**

\#3: $+$ on natural numbers (redefinition/restriction of $+$ on complex numbers)

- 1710 items implicitly rely on $+$'s **commutativity**

Maximal unnecessariness

#1: set membership $\in$

- 34935 items do not need $\in$'s **asymmetry** (a weak form of the axiom of foundation)

#2: $\leq$ on extended real numbers

- 10018 items rely on $\leq$ but need neither its reflexivity nor its connectivity
- 2101 items rely on $\leq$'s reflexivity, but do not need its connectedness
- 2351 items rely on $\leq$'s connectedness, but do not need its reflexivity

#3: the subset relation $\subseteq$

- 16581 items do not need $\subseteq$'s **reflexivity**

Exploiting properties

Among those items of the **mizar** Mathematical Library that do exploit constructor properties:

- Average number of exploited constructor properties: **2**.
- Median: **2**.
- Standard deviation: **1.2**
- Maximum number of exploited properties: 14.

Exploiting properties: example

- The **mizar** article `BROUWER`, devoted to the proof of Brouwer's fixed-point theorem for plane disks, contains a theorem that, among all theorems of the **mizar** Mathematical Library, implicitly exploits the greatest number of constructor properties:

Brouwer example

```
for a being real number for r being non negative real number
for n being non empty Element of NAT
for s, t, o being Point of (TOP-REAL n)
for S, T, O being Element of REAL n st
  S = s & T = t & O = o
 holds
  (s is Point of Tdisk(o,r) & t is Point of Tdisk(o,r) &
   s <> t &
   a = ((-|(t-s,s-o)| + sqrt(|(t-s,s-o)|^2 -
        (Sum sqr (T-S)) * (Sum sqr (S-O)-r^2)))
        / Sum sqr (T-S))
  implies
  HC(s,t,o,r) = (1-a)*s+a*t
```

Brouwer example (continued)

- This theorem implicitly uses 14 constructor properties:
  - Asymmetry of $\in$
  - Reflexivity and connectedness of $\leq$ on extended real numbers
  - Commutativity and idempotence of $\cap$
  - Projectivity of the absolute value operation on complex numbers
  - Commutativity of the norm operation on finite sequences of real numbers
  - Commutativity of $+$ on abelian magmas
  - ...

Not exploiting properties

Among those 67587 items in the **mizar** Mathematical Library that do **not** exploit a constructor property:

- Median number of unexploited constructor properties: **3**
- Average number of unexploited constructor properties: **3.5**.
- Standard deviation: **2.3**
- Maximum number of unexploited constructor properties: **20** (three items)

Not exploiting properties: Tietze extension theorem

```
for T being non empty TopSpace
 st (for A being non empty closed Subset of T
     for f being continuous Function
       of (T | A), (Closed-Interval-TSpace ((- 1),1))
     ex g being continuous Function
       of T, (Closed-Interval-TSpace ((- 1),1))
       st g | A = f)
 holds
 T is normal
```

Not exploiting properties: Tietze extension theorem (continued)

- This theorem is verified in an environment in which 20 constructor properties can be dispensed with:
  - Commutativity of $+$ on real numbers
  - Involutiveness of unary subtraction on real numbers
  - Projectivity of the norm operation on complex numbers
  - Projectivity of the absolute value operation on complex numbers
  - Commutativity of multiplication on natural numbers
  - ...

Using the information that a property of a constructor is (un)necessary

- When we learn that a property P of a constructor C is not needed for a theorem ϕ, we have, prima facie, a kind of generalization of ϕ.
- But what kind of generalization?
  - Suppose we find that a theorem does not require commutativity of $+$ on real numbers.
  - We may say, axiomatically, that we can prove ϕ from a set Γ of axioms that does not include commutativity of $+$.
  - We may or may not be interested in the range of models of Γ.
- It is one thing to find out that commutativity of $+$ on real numbers isn't needed for a particular theorem; it's another to find out that commutativity of group addition is not needed for a theorem about groups.

Type promotion

- Terms in **mizar** are typed. Terms can have multiple types simultaneously (e.g., 2 is a real number, a complex number, a positive natural number, …)
- Since **mizar**'s approach to mathematics is the foundationally conventional 'everything is a set', the lattice of types has a top element, `set`. Every term has `set` among its types, in any context.
- Relations take arguments of given types. Functions take arguments of specified types, and their value has a specified type.

Promote types throughout the **mizar** Mathematical Library

- Experiment: given a function f whose value has type τ, 'promote' f by declaring that the type of its value is the topmost type `set`.
- Cases where the verifiability of a theorem ϕ involving f unaffected by such a 'type promotion' suggest, as in the case of safely omitting constructor properties, that a generalization of ϕ is available.

Limitations of the current information

- This information that an item I in the **mizar** Mathematical Library requires a constructor property P is complete:
  - Verification of I in the absence of P fails.
- By contrast, the information that an item I does **not** require a constructor property P is incomplete:
  - Verification of I in the absence of the given constructor property succeeds.
  - But the items on which I depends may themselves require P.
  - In this sense I can be said to require P, even if I does not **immediately** require P.

Future work

- Visualization/exploration of constructor property (in)dependencies.
- When we find that the **mizar** verifier accepts an inference in the absence of a given constructor property, it may very well be that the deduction it finds **differs** from the deduction it finds in the presence of the constructor. Such cases, if they exist, would be instructive.
- An advisor that suggests generalizations of theorems, using fine-grained information about what implicit information is truly needed for the theorem, would be a valuable aid.

Even more future work

- Turning formal proofs into ATP problems would provide another approach of the (un)necessity of constructor properties for particular theorems (as well as provide a way of testing our data).
- Rather than keeping theorems/proofs intact but manipulating the environment in which they're verified, one could **systematically rewrite them**—possibly with the assistance of the aforementioned advisor—to make explicit the dependence a theorem on the properties of the functions and predicates appearing in it.

Literature

- A. Arana, "On formally measuring and eliminating extraneous notions in proofs", **Philosophia Mathematica 17** (2009), 189-207
- **The mizar homepage**
- **Grabowski, A. and Kornilowicz, A. and Naumowicz, A., "mizar in a Nutshell", in** *Journal of Formalized Reasoning* **3(2), pp. 153–245, 2010.**
- J. Urban, "MPTP—Motivation, implementation, first experiments", **Journal of Automated Reasoning**, **33**(3-4) (2004), pp. 319–339
- **The mizar-items site**