Mizar

Piotr Rudnicki

Department of Computing Science
University of Alberta
piotr@cs.ualberta.ca
The name

MIZAR, ζ Ursae Majoris, distance 78.2 ± 1.1 light years, the first binary star imaged telescopically, Riccioli (1650). Mizar A – the first star discovered to be spectroscopically binary, Pickering (1889). MIZAR B is at least binary.
The MIZAR project

Goal: A data base of computer verified mathematics

Language: Close to mathematical vernacular yet allowing mechanical checking of correctness

Leader: Andrzej Trybulec, University of Białystok, Poland.

Authors: Software: Currently 8 developers
          MIZAR texts: 200+

Since: 1973

Stable: since 1989 a data base has been maintained

Motto: Proving is a pleasure

Thus: No stress on automated theorem proving (ATP)
Why? – Pure mathematics

Gaussian integers: \( a + bi : a, b \in \mathbb{Z} \)

Generalized: \( a + b\sqrt{-d} \) where \( d \in \mathbb{Z}^+ \)

\[
a, b \in \frac{1}{2}\mathbb{Z} \quad \text{when } d \mod 4 = 3
\]
\[
a, b \in \mathbb{Z} \quad \text{otherwise}.
\]

For which \( d \) do we have unique factorization? Not for 5:

\[
6 = 2 \times 3 \quad \text{but also} \quad 6 = \left(1 + \sqrt{-5}\right) \times \left(1 - \sqrt{-5}\right)
\]

1855: At Gauss’s death: \( d = 1, 2, 3, 7, 11, 19, 43, 67, 163 \)

1934: Heilbronn and Linfoot: there could be at most one more.

1952: Heegner: no more.

Nobody believed him—he was not a mathematician.

1967: Stark and Baker: no tenth \( d \).

Then they confirmed that Heegner was correct.
Why? – Specification and verification

Ricky W. Butler (NASA)
Airplane Seat Reservation System

seat_assignment: TYPE [# seat : [row, position], pass : passenger #]
flight_assignments: TYPE = set[seat_assignment]
flt_db: TYPE = [flight -> flight_assignments]
Next_seat: [flt_db, flight, preference -> [row, position]]

AXIOM (FORALL a: a in db(flt) -> seat(a) /= Next_seat(db, flt, pref)

Next_seat(db, flt, pref) returns a seat even when a flight is full. Which seat? Contradiction.

I now avoid axioms like the plague. It is surprisingly easy to get them wrong! – RWB

In MIZAR, one must construct things, no additional axioms. The technology of mathematics is robust. Let us follow it.
Why? – Proof correctness


**Theorem** There does not exist \( r \) in \( \mathbb{Q} \) such that \( r^2 = 2 \).

**Assume:**
1. \( r \in \mathbb{Q} \)
2. \( r^2 = 2 \)

\(<1>\) 1. Choose \( m, n \) in \( \mathbb{Z} \) such that \( \cdots \)
\(<2>\) 4. \( \gcd(m, n) = 1 \)

**Proof:** By the definition of \( \gcd \), it suffices to:

**Assume:**
1. \( s \) divides \( m \)
2. \( s \) divides \( n \)

**Prove:** \( s = 1 \) \( \cdots \)

LL manages to prove it without even saying where \( s \) is from!

Anecdotal evidence suggests that as many as a third of all papers published in mathematical journals contain mistakes—not just minor errors, but incorrect theorems and proofs. ibidem, p. 311.
Some relatives

**HOL** interactive theorem proving in a higher-order logic, widely used for hardware verification

**Coq** calculus of constructions, enables extraction of programs from proofs

**PVS** support for formal specification and verification based on higher order logic, applications in industry

**Isabelle** generic theorem proving environment, attempts at applications in protocol design and cryptography

**ACL2** logic is a subset of applicative Common Lisp, tailored for modeling computing machines

- 100s more

The above offer more automation than MIZAR, are geared toward some specific applications, do not build a comprehensive data base of mathematics and use a language far removed from mathematical practice.
MIZAR: points of interest

- The MIZAR language
- MML – MIZAR Mathematical Library
  - axioms of the Tarski-Grothendieck set theory
  - library articles: user interface and internals
- MIZAR article and its processing
- MIZAR processor
- MIZAR on the web
- How to become a MIZAR author?

(Mathematical Knowledge) Management

emerging field dealing with math presence on the web.

Not: Mathematical (Knowledge Management)
Theorem: an example

Prove that for all natural $n$, $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$

**MIZAR:** for $n$ being Nat holds Sum idseq $n = n\times(n+1)/2$

Local environment: imports from MML 4.181.1147

environ

vocabulary RLVECT_1, FINSEQ_2, FINSEQ_1, ARYTM_3, RELAT_1, RVSUM_1, XBOOLE_0, SQUARE_1, NAT_1, CARD_1, NUMBERS, CARD_3, ORDINAL4, NEWTON, VALUED_0;
notations NUMBERS, XBOOLE_0, REAL_1, NAT_1, FINSEQ_1, FINSEQ_2, SQUARE_1, ORDINAL1, RVSUM_1;
constructors REAL_1, RVSUM_1, SQUARE_1, BINOP_2;
registrations NUMBERS, RELSET_1, VALUED_0, MEMBERED, FINSEQ_2, NEWTON, RVSUM_1;
requirements NUMERALS, BOOLE, SUBSET, ARITHM;
definitions FINSEQ_1;
theorems FINSEQ_2, RVSUM_1, RELAT_1, TOPREAL7, SQUARE_1, VALUED_1;
schemes NAT_1;

begin

and now we can continue our proof
defpred P[Nat] means Sum idseq $1 = $1*($1+1)/2;

Basis: P[0] by RVSUM_1:72;
IndStep:
for n being Nat st P[n] holds P[n+1]
proof let n be Nat such that
  IndHyp: Sum idseq n = n*(n+1)/2;
  thus Sum idseq(n+1)
    = Sum((idseq n)^<*n+1*>) by FINSEQ_2:51
    .= Sum(idseq n) + (n+1) by RVSUM_1:74
    .= (n+1)*(n+1+1)/2 by IndHyp;
end;

for n being Nat holds P[n] from NAT_1:sch 2(Basis, IndStep);
then for n being Nat holds Sum idseq n = n*(n+1)/2;
Theorem: another example

defpred S[Nat] means \( \text{Sum } \text{sqr idseq } 1 = 1 \times (1+1) \times (2 \times 1+1)/6; \)

Basis: \( S[0] \) proof
   \( \text{dom } \text{sqr idseq } 0 = \text{dom idseq } 0 \) by VALUED_1:11 \( = \{\} \) by RELAT_1:38;
   hence \( \text{Sum } \text{sqr idseq } 0 = 0 \) by RELAT_1:41, RVSUM_1:72
      \( = 0 \times (0+1) \times (2 \times 0+1)/6; \)

IndStep: for \( n \) being Nat st \( S[n] \) holds \( S[n+1] \) proof
   let \( n \) be Nat such that \( \text{IndHyp: } S[n] \);
   Aux: \( \text{idseq } n \) is FinSequence of REAL by RVSUM_1:145;
   thus \( \text{Sum } \text{sqr idseq } (n+1) \)
      \( = \text{Sum } \text{sqr } ((\text{idseq } n)^{<n+1>}) \) by FINSEQ_2:51
      \( = \text{Sum } ((\text{sqr idseq } n)^{\text{sqr } <n+1>}) \) by Aux, RVSUM_1:144
      \( = \text{Sum } ((\text{sqr idseq } n)^{<*(n+1)^2>}) \) by RVSUM_1:55
      \( = \text{Sum } \text{sqr idseq } n + (n+1)^2 \) by RVSUM_1:74
      \( = n \times (n+1) \times (2 \times n+1)/6 + (n+1) \times (n+1) \) by IndHyp, SQUARE_1:def 1
      \( = (n+1) \times (n+1+1) \times (2 \times (n+1)+1)/6; \)

for \( n \) being Nat holds \( S[n] \) from NAT_1:sch 2(Basis, IndStep);
then for \( n \) being Nat holds \( \text{Sum } \text{sqr idseq } n = n \times (n+1) \times (2 \times n+1)/6; \)
The Mizar language

- The language mimics traditional mathematics.
- Based on classical, typed, first order logic with equality. The natural deduction system of Jaśkowski (Fitch).
- Definitions of constructors:

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Construction</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicate</td>
<td>Atomic formula</td>
<td>x is_a_fixpoint_of f</td>
</tr>
<tr>
<td>Functor</td>
<td>Term</td>
<td>lfp (X, f)</td>
</tr>
<tr>
<td>Mode</td>
<td>Type</td>
<td>Relation of X, Y</td>
</tr>
<tr>
<td>Attribute</td>
<td>Adjective</td>
<td>n is even</td>
</tr>
<tr>
<td>Structure</td>
<td>Type</td>
<td>struct DB-Rel (# fields #)</td>
</tr>
</tbody>
</table>

- Propositional schemes with free second order variables.
MML: MIZAR Mathematical Library: foundations

Unit HIDDEN: primitive notions set, in, =
Unit TARSKI: Tarski-Grothendieck set theory axioms

- Axiom of extensionality  
  equality of sets
- Axiom of singleton and pair  
  existence
- Axiom of union  
  existence
- Axiom of regularity  
  no infinite descending $\epsilon$ chains
- Axioms of replacement  
  functional image of a set
- Tarski’s axiom of strongly inaccessible cardinals

TG = ZF - { some existence axioms } - { AC } + { large cardinals }

Operational built-ins:

BOOLE, SUBSET, ARITHM, REAL, NUMERALS
MML – MIZAR Mathematical Library

MML is a collection of articles.
What is a MIZAR article?

Analogy to *What is a published paper?*

What is in MML?
- March 2003: 765 articles
- September 2008: 1033 articles
- May 2012: 1147 articles

Basic mathematical toolkit: relations, functions, ...

How many of these?

On top of the toolkit (some examples)

- Set theory
- Meta-logic
- Algebra
- Analysis
- Topology
- Number theory
- Graph theory

Reflection lemma
Gödel completeness theorem
FTA, Wedderburn theorem
l’Hôpital theorem
Jordan curve theorem
Bertrand’s postulate, CRT
Chordal graphs recognition
Some efforts focused in narrower areas

- Continuous lattices
- Algebra of polynomials
- Real and complex analysis
- Modeling computations
- Graph algorithms

The blow-up factor for the number of words (tokens) is $\approx 10$ when translating mathematical monographs into MIZAR
### MML: Some Numbers

<table>
<thead>
<tr>
<th></th>
<th>3.46.767</th>
<th>4.100.1011</th>
<th>4.187.1147</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apr '03</td>
<td>Apr '08</td>
<td>May '12</td>
</tr>
<tr>
<td>Theorems</td>
<td>33178</td>
<td>46506</td>
<td>51762</td>
</tr>
<tr>
<td>Definitions</td>
<td>6557</td>
<td>8804</td>
<td>10158</td>
</tr>
<tr>
<td>Schemes</td>
<td>684</td>
<td>756</td>
<td>787</td>
</tr>
<tr>
<td>Constructors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functor</td>
<td>5043</td>
<td>6823</td>
<td>7768</td>
</tr>
<tr>
<td>Mode</td>
<td>406</td>
<td>438</td>
<td>447</td>
</tr>
<tr>
<td>Predicate</td>
<td>684</td>
<td>878</td>
<td>1013</td>
</tr>
<tr>
<td>Attribute</td>
<td>1498</td>
<td>2043</td>
<td>2345</td>
</tr>
<tr>
<td>Structures</td>
<td>88</td>
<td>116</td>
<td>132</td>
</tr>
<tr>
<td>Registrations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existential</td>
<td>1416</td>
<td>1861</td>
<td>2219</td>
</tr>
<tr>
<td>Functorial</td>
<td>2796</td>
<td>4568</td>
<td>6598</td>
</tr>
<tr>
<td>Conditional</td>
<td>1025</td>
<td>1496</td>
<td>2044</td>
</tr>
</tbody>
</table>
Functor: example of a definition

From XBOOLE_0

definition let X,Y be set;
  func X \/ Y -> set means x in it iff x in X or x in Y;

existence proof
  take union \{X,Y\}; let x;
  thus x in union \{X,Y\} implies x in X or x in Y
  proof ... end;
  assume x in X or x in Y; ... 
  hence x in union \{X,Y\} by ... 
end;

uniqueness proof let A1, A2 be set such that
  A6: x in A1 iff x in X or x in Y and
  A7: x in A2 iff x in X or x in Y;
  ... 
  hence A1 = A2 by TARSKI:2;
end;

commutativity;
idempotence;
end;
Predicate, attribute, cluster: examples

From ASYMPT_0

definition let f be Real_Sequence;
  attr f is eventually-nonnegative means :: ASYMPT_0:def 4
    ex N st for n st n >= N holds f.n >= 0;
end;

registration
cluster eventually-nonnegative eventually-nonzero positive
  eventually-positive eventually-nondecreasing Real_Sequence;
existence proof
  reconsider f = NAT-->1 as Function of NAT,REAL by FUNCOP_1:57;
  take f;
  thus f is eventually-nonnegative proof ... end;
  thus f is eventually-nonzero proof ... end;
  ... 
end;
end;

definition
  let f be eventually-nonnegative Real_Sequence, b be Nat;
  pred f is_smooth_wrt b means :: ASYMPT_0:def 19
    f is eventually-nondecreasing & f taken_every b in Big_Oh(f);
end;
Clusters: examples

From ASYMPT_0

registration
cluster eventually-nonnegative eventually-nonzero
  -> eventually-positive Real_Sequence;
coherence proof let f be Real_Sequence; assume
A3: f is eventually-nonnegative & f is eventually-nonzero;
then consider N such that
A4: for n st n >= N holds f.n >= 0 by Def4;
...
A8: n >= N & n >= M by A6,XXREAL_0:2;
f.n <> 0 by A5,A6,A7,XXREAL_0:2;
  hence thesis by A4,A8;
end;
end;

registration
let f, g be eventually-nonnegative Real_Sequence;
cluster f+g -> eventually-nonnegative;
coherence proof
  ... let n; ...
  hence (f + g).n >= 0 by SEQ_1:11;
end;
end;
Mode: example

From FINSEQ_1

definition let D be set;
  mode FinSequence of D -> FinSequence means
    rng it c= D;
  existence proof
    ...
  end;
end;

registration
  let D be set;
  cluster FinSequence-like PartFunc of NAT,D;
  existence
  proof {} is PartFunc of NAT,D by RELSET_1:25;
    hence thesis;
  end;
end;

definition let D be set;
  redefine mode FinSequence of D
    -> FinSequence-like PartFunc of NAT,D;
  coherence proof ... end;
end;
Hierarchy of notions: example

\[ \text{FinSequence of } D \rightarrow \text{FinSequence} \]
\[ \text{FinSequence is FinSequence-like Function} \]
\[ \text{FinSequence-like attribute to Relation} \]
\[ \text{Relation is Relation-like set} \]
\[ \text{Relation-like attribute to set} \]
\[ \text{Function is Function-like Relation-like set} \]
\[ \text{Function-like attribute to set} \]

\[ \text{FinSequence of } D \rightarrow \text{FinSequence-like PartFunc of } \text{NAT,D} \]

\[ \text{NAT} \rightarrow \text{Subset of } \text{REAL} \]
\[ \text{REAL} \rightarrow \text{set} \quad :: \text{a construction of reals is behind it} \]
\[ \text{Subset of } X \text{ is Element of bool } X \]
\[ \text{Element of } X \rightarrow \text{set} \]
\[ \text{bool } X \rightarrow \text{set} \]

\[ \text{PartFunc of } X,Y \text{ is Function-like Relation of } X,Y \]
\[ \text{Relation of } X,Y \rightarrow \text{Subset of } [:X,Y:] \]
\[ [:X,Y:] \rightarrow \text{set} \]
Concerns of (not only casual) authors

- Learning the system
- Knowing the library
- Searching the library
- Reading formal proofs
- Presentation of formal proofs
- Gaps in the library
- Algebraic manipulations
- Introducing new notations
- Theory vs examples
Article processing: batch

Mizar DB

Accommodator

Local environment

Verifier

Error−printer

environ

Directives

begin

Text Proper

Error file

Error−printer
Including a new article into MML

Finished article

environ

Directives

begin

Text Proper

Exporter

DB Files
defs thms schms

Accommodator

Local environment

Mizar DB

Text Proper Exporter
Presentation and distribution

- Source articles
- Library files (internal)
- \textsc{Mizar} \textbf{abstract} of an article — a text file of definitions and theorems but without proofs
- Abstracts and entire article available hyper-linked in html/xml for web browsing thanks to Josef Urban
- Abstracts automatically \textsc{TeX}ed and published and on paper \textit{Formalized Mathematics}
- Work on \textsc{Mizar} Encyclopedia: monographic articles
Applications

- Building a data base of formalized mathematics
- Exporting from Mizar to other systems
- Education: logic and mathematics
- Specification and verification
Josef Urban applies AI to play with MML

- Can an automated prover prove theorems from MML which have been proved by hand?
  - All premises are exported
  - Most provers choke on several thousand premises
- Machine learning to find the relevant premises for a theorem.
  - Assists humans when proving Well, not really. Too many false positives.
  - Assists theorem provers on simpler cases with spectacular results.
- Escape to ATP from MIZAR
- ATPs as a search engine for facts in MML

Jesse Alama translates ATP found proofs to MIZAR.
Lessons from MIZAR

- Any similar project should focus on building a data base.
- Any similar system will evolve: language, checker, ...
- The evolution will be driven by the growing data base.
- How to involve mathematicians?
- How to find authors?
- Who will pay for all this?
- What has happened to QED? A project to build a computer system that effectively represents all important mathematical knowledge and techniques.