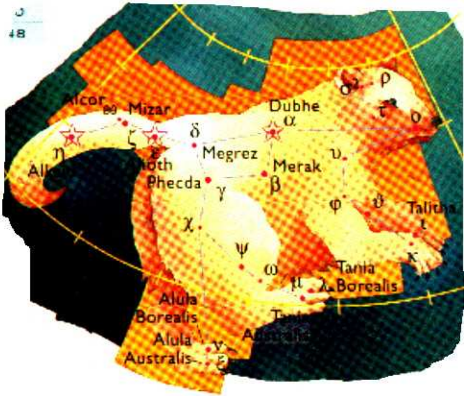
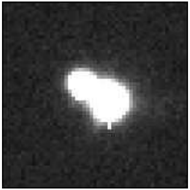
Mizar

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: $a + bi : a, b \in \mathbf{Z}$

Generalized: $a + b\sqrt{-d}$ where $d \in \mathbf{Z}^+$

$$\begin{array}{ll} a, b \in \frac{1}{2}\mathbf{Z} & \text{when } d \bmod 4 = 3 \\ a, b \in \mathbf{Z} & \text{otherwise.} \end{array}$$

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

$$6 = 2 \times 3 \quad \text{but also} \quad 6 = (1 + \sqrt{-5}) \times (1 - \sqrt{-5})$$

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)
Tutorial on Formal Methods, PVS (1992–1996)
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

Leslie Lamport, How to write a proof, *Global Analysis in Modern Mathematics*, PUBLISH OR PERISH, INC., 1993.

Theorem There does not exist r in $\mathbf{Q}$ such that $r^2 = 2$.

ASSUME: 1. $r \in \mathbf{Q}$

2. $r^2 = 2 \cdots$

$\langle 1 \rangle 1.$ Choose m, n in $\mathbf{Z}$ such that $\cdots$

$\langle 2 \rangle 4.$ $\gcd(m, n) = 1$

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, $\sum_{i=0}^{i=n} i = \frac{n(n+1)}{2}$

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

Local environment: imports from MML 4.181.1147

```
environ
 vocabularies 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

Theorem: an example, cntd

```
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 Sum sqr idseq $1 = $1*($1+1)*(2*$1+1)/6;

Basis: S[0] proof
  dom sqr idseq 0 = dom idseq 0 by VALUED_1:11 .= {} by RELAT_1:3
 hence Sum sqr idseq 0 = 0 by RELAT_1:41, RVSUM_1:72
   .= 0*(0+1)*(2*0+1)/6;
end;

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

for n being Nat holds S[n] from NAT_1:sch 2(Basis, IndStep);
then for n being Nat holds Sum sqr idseq n = n*(n+1)*(2*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:

Constructor	Construction	Example
Predicate	Atomic formula	`x is_a_fixpoint_of f`
Functor	Term	`lfp (X, f)`
Mode	Type	`Relation of X, Y`
Attribute	Adjective	`n is even`
Structure	Type	`struct DB-Rel (#` *fields* `#)`

- 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 ϵ 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 — Reflection lemma
- Meta-logic — Gödel completness theorem
- Algebra — FTA, Wedderburn theorem
- Analysis — l'Hôpital theorem
- Topology — Jordan curve theorem
- Number theory — Bertrand's postulate, CRT
- Graph theory — Chordal graphs recognition

MML – MIZAR Mathematical Library

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 ≈ 10 when translating mathematical monographs into MIZAR

MML: Some Numbers

	3.46.767 Apr ’03	4.100.1011 Apr ’08	4.187.1147 May ’12
Theorems	33178	46506	51762
Definitions	6557	8804	10158
Schemes	684	756	787
Constructors			
Functor	5043	6823	7768
Mode	406	438	447
Predicate	684	878	1013
Attribute	1498	2043	2345
Structures	88	116	132
Registrations			
Existential	1416	1861	2219
Functorial	2796	4568	6598
Conditional	1025	1496	2044

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;
```

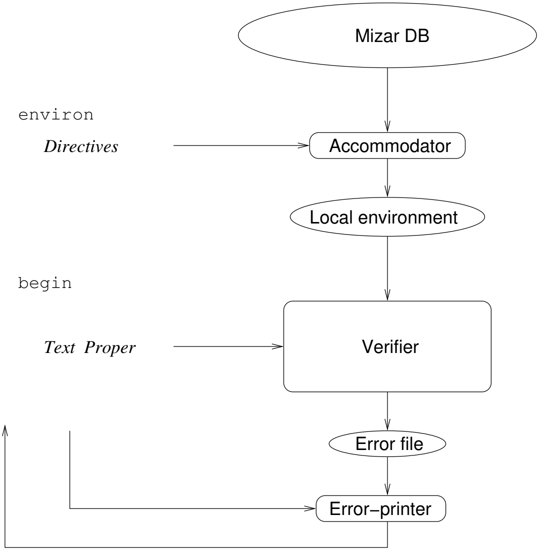
Hierarchy of notions: example

```
FinSequence of D -> FinSequence
  FinSequence is FinSequence-like Function
    FinSequence-like attribute to Relation
      Relation is Relation-like set
        Relation-like attribute to set
    Function is Function-like Relation-like set
      Function-like attribute to set

FinSequence of D -> FinSequence-like PartFunc of NAT,D

  NAT -> Subset of REAL
    REAL -> set        :: a construction of reals is behind it
    Subset of X is Element of bool X
      Element of X -> set
      bool X -> set

  PartFunc of X,Y is Function-like Relation of X,Y
    Relation of X,Y -> Subset of [:X,Y:]
      [:X,Y:] -> set
```

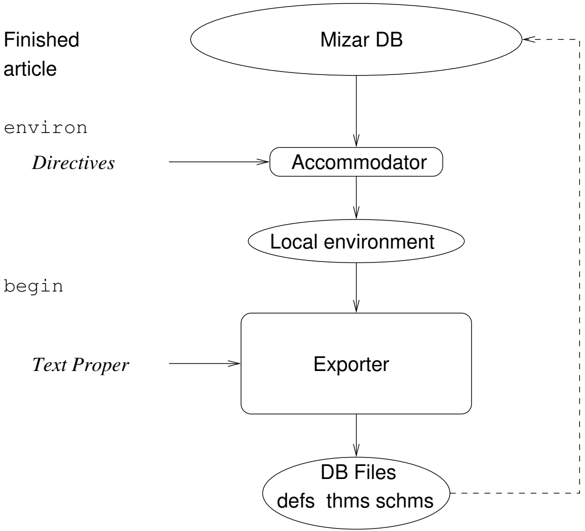
Concerns of (not only casual) authors

- 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



Including a new article into MML



Presentation and distribution

- Source articles
- Library files (internal)
- MIZAR **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 TEXed and published and on paper *Formalized Mathematics*
- Work on 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.*