Using evolved MML

Grzegorz Bancerek
bancerek@mizar.org

Białystok Technical University

II Conference
on Technologies of Knowledge Exploration and Representation
TERW 2007
5-9 September 2007, Hołny Mejera

1. Mizar

- a language used for such a formalization of mathematics that is close to the vernacular used in mathematical publications

1. Mizar

- a language used for such a formalization of mathematics that is close to the vernacular used in mathematical publications

- a verifier for checking correctness of Mizar texts according to Jaśkowski's natural deduction

1. MIZAR

- a language used for such a formalization of mathematics that is close to the vernacular used in mathematical publications
- a verifier for checking correctness of MIZAR texts according to Jaśkowski's natural deduction
- MIZAR Mathematical Library (MML)—a centrally maintained library of formalized mathematics based on Tarski–Grothendieck set theory (MML is commonly considered the biggest library of computer proof-checked mathematics)

2. Mizar Mathematical Library

now (year ago)

- **985** (942) articles (**79048** (71744) kB) by **198** (180) authors from **13** (12) countries

2. Mizar Mathematical Library

now (year ago)

- **985** (942) articles (**79048** (71744) kB) by **198** (180) authors from **13** (12) countries
- **45002** (42694) theorems, **8524** (7957) definitions, **753** (728) schemes, **7511** (6942) registrations, **6320** (5879) symbols

2. Mizar Mathematical Library

now (year ago)

- **985** (942) articles (**79048** (71744) kB) by **198** (180) authors from **13** (12) countries
- **45002** (42694) theorems, **8524** (7957) definitions, **753** (728) schemes, **7511** (6942) registrations, **6320** (5879) symbols
- development of MML is managed by the Library Committee and the Development Committee of **the Association of Mizar Users** (the owner of MML)

3. Library Committee – submission

task: the acceptance of new articles into MML and the distribution of MML

3. Library Committee – submission

task: the acceptance of new articles into MML and the distribution of MML

past: the *non-triviality* criterion for acceptance

3. Library Committee – submission

task: the acceptance of new articles into MML and the distribution of MML

past: the *non-triviality* criterion for acceptance

The criterion was successively completed by software for recognition of unused elements, inaccessible fragments, irrelevant premises, etc.

3. Library Committee – submission

task: the acceptance of new articles into MML and the distribution of MML

past: the *non-triviality* criterion for acceptance

The criterion was successively completed by software for recognition of unused elements, inaccessible fragments, irrelevant premises, etc.

recently: a new MIZAR article must pass successfully the so-called revision software tests and additionally is must get 3 positive human reviews

3. Library Committee – submission

task: the acceptance of new articles into MML and the distribution of MML

past: the *non-triviality* criterion for acceptance

The criterion was successively completed by software for recognition of unused elements, inaccessible fragments, irrelevant premises, etc.

recently: a new MIZAR article must pass successfully the so-called revision software tests and additionally is must get 3 positive human reviews

the result: new articles are free from formalization mistakes which can be recognized by reviewers

4. Development Committee – revisions

created as a part of the Library Committee

4. Development Committee – revisions

created as a part of the Library Committee

task: keeping the MML in the state allowing for development

4. Development Committee – revisions

created as a part of the Library Committee

task: keeping the MML in the state allowing for development

tool - revision: reorganization of the material stored in MML

4. Development Committee – revisions

created as a part of the Library Committee

task: keeping the MML in the state allowing for development

tool - revision: reorganization of the material stored in MML

essential goals of revisions: the quality and integrity of MML

5. Revision of MML

conservative – the content of MML is not reduced

5. Revision of MML

conservative – the content of MML is not reduced

unsubstantial

substantial

5. Revision of MML

conservative – the content of MML is not reduced

unsubstantial – rearrangement of labels, reformulating of formulae, and removing of inaccessible fragments, irrelevant references (premises), irrelevant proof steps, trivial proofs, environment directives, etc. done with revision software

substantial

5. Revision of MML

conservative – the content of MML is not reduced

unsubstantial – rearrangement of labels, reformulating of formulae, and removing of inaccessible fragments, irrelevant references (premises), irrelevant proof steps, trivial proofs, environment directives, etc. done with revision software

substantial – change database files (exportable elements)

5. Revision of MML

conservative – the content of MML is not reduced

unsubstantial – rearrangement of labels, reformulating of formulae, and removing of inaccessible fragments, irrelevant references (premises), irrelevant proof steps, trivial proofs, environment directives, etc. done with revision software

substantial – change database files (exportable elements)

some formalizations and reasoning could be done shorter and/or easier

6. Example 1

```
definition
  let f be Function;
  let x be set such that  PermisiveAssumption: x in dom f;
  func f.x means [x, it] in f;
end;
```

6. Example 1

```
definition
  let f be Function;
  let x be set such that  PermisiveAssumption: x in dom f;
  func f.x means [x, it] in f;
end;

definition
  let f be Function;
  let x be set;
  func f.x means [x, it] in f  if  x in dom f
           otherwise it = {};
end;
```

6.1. Example 1, cont.

```
theorem :: FUNCT_1:22 (old)
  x in dom f & f.x in dom g implies (g*f).x = g.(f.x)
```

```
theorem :: FUNCT_1:22 (new)
  x in dom f implies (g*f).x = g.(f.x)
```

7. Substantial revisions

- elimination of redundant definitions

7. Substantial revisions

- elimination of redundant definitions
- correction and reformulation of definientia

7. Substantial revisions

- elimination of redundant definitions
- correction and reformulation of definientia
- generalization of arguments

7. Substantial revisions

- elimination of redundant definitions
- correction and reformulation of definientia
- generalization of arguments
- order change of definitions, theorems, schemes, or/and articles

7. Substantial revisions

- elimination of redundant definitions
- correction and reformulation of definientia
- generalization of arguments
- order change of definitions, theorems, schemes, or/and articles
- elimination of obvious consequences

8. Obviousness of canceled theorems

- automatic removing of obvious theorems (cancellation of theorems)

8. Obviousness of canceled theorems

- automatic removing of obvious theorems (cancellation of theorems)
- automatic obviousness is not so obvious to humans

8. Obviousness of canceled theorems

- automatic removing of obvious theorems (cancellation of theorems)
- automatic obviousness is not so obvious to humans
- complication of Mizar obviousness:

  synonyms and antonyms, special treatment of some constructors (associativity, connectidness, idempotence, involutiveness, projectivity, . . .), adjective addition by registrations, term unfolding according to implicit definitions, identification of terms

9. Example 2

The theorem

```
x < y & y < z implies x < z
```

was canceled.

9. Example 2

The theorem

```
x < y & y < z implies x < z
```

was canceled. It is identical modulo antonyms to the formula

```
z <= x & not y <= x implies x <= y
```

9. Example 2

The theorem

```
 x < y & y < z implies x < z
```

was canceled. It is identical modulo antonyms to the formula

```
 z <= x & not y <= x implies x <= y
```

which must be a consequence of

```
theorem  :: XXREAL_0:2
  x <= y & y <= x implies x <= z
```

9. Example 2

The theorem

```
x < y & y < z implies x < z
```

was canceled. It is identical modulo antonyms to the formula

```
z <= x & not y <= x implies x <= y
```

which must be a consequence of

```
theorem  :: XXREAL_0:2
  x <= y & y <= x implies x <= z
```

because we know from `connectedness` that `x <= y or y <= x`

10. Example 3

```
::  CARD_1:def 5
M = Card X iff X, M are_equipotent

:: CARD_1:20 (canceled)
Card M = M
```

where `X` is a set and `M` is cardinal number. The predicate `are_equipotent` is reflexive

```
  M, M are_equipotent
```

and therefore references to `CARD_1:20` may be replaced by references to `CARD_1:def 5`.

11. Example 4

The same argumentation was used to remove the theorem

```
A \/ A = A
```

because there was already in MML the theorem

```
A c= B implies A \/ B = B
```

where `c=` stands for *reflexive* inclusion and \/ stands for the union.

12. **Using evolved** MML

- some theorems convenient for particular users may disappear

12. Using evolved MML

- some theorems convenient for particular users may disappear
- distributed documentation is not satisfying

12. Using evolved MML

- some theorems convenient for particular users may disappear
- distributed documentation is not satisfying
- meaning of canceled theorems is not available

12. **Using evolved** MML

- some theorems convenient for particular users may disappear
- distributed documentation is not satisfying
- meaning of canceled theorems is not available
- prohibited introduction of “trivial corollaries” (e.g., explaining the idea of formalization)

12. **Using evolved** MML

- some theorems convenient for particular users may disappear
- distributed documentation is not satisfying
- meaning of canceled theorems is not available
- prohibited introduction of “trivial corollaries” (e.g., explaining the idea of formalization)
- MML Query – the solution?

13. MML Query - browsing

```
theorem :: XXREAL_0:1
for b1, b2 being ext-real set
      st b1 <= b2 & b2 <= b1
   holds b1 = b2;

Other forms:

theorem
for b1, b2 being ext-real set
      st b2 <= b1 & b1 <> b2
   holds b2 < b1;
```

14. MML **Query - documentation**

Automatic comparison of two versions:

- removed/added articles
- removed/added theorems
- removed/added definitions
- ....
- further querying hints

15. MML **Query - version query**

Theorem `FUNCT_2:74` was canceled. It appeared in version 4.50.934.

The version query:

```
exactly * (version 4.50.934 FUNCT_2:74 ref)
```

finds theorem `FUNCT_2:23` which after some revisions became identical to `FUNCT_2:74`

15. MML **Query - version query**

Theorem `FUNCT_2:74` was canceled. It appeared in version 4.50.934.

The version query:

```
exactly * (version 4.50.934 FUNCT_2:74 ref)
```

finds theorem `FUNCT_2:23` which after some revisions became identical to `FUNCT_2:74`

(!) A version query does not work when order of definitions is changed.

16. MML Query - non-expert query

```
for a being cardinal number holds a = Card a

(symbol cardinal [notation | [[[direct | constructor]
       | negative occur] or [[opposite | constructor]
       | positive occur]]]) and
(symbol number [...]) and
(symbol = [...]) and
(symbol Card [...])
ordered by number of ref
```

the result: 18 elements with `CARD_1:def 5` as the first

17. Non-expert queries

`Mizar` $(s_1\ s_2\ \ldots\ s_n)$

17. Non-expert queries

$$\texttt{Mizar}\ (s_1\ s_2\ \ldots\ s_n)$$

$$\texttt{rough max} - \texttt{count}(q_1, q_2, \ldots, q_n)$$
$$q_i = \texttt{symbol}\ s_i\ [\texttt{notation}|\texttt{constructor}|\texttt{occur}]$$

17.1. Non-expert queries

$\texttt{Mizar}\ (s_1\ \ldots\ s_n\ \texttt{implies}\ s'_1\ \ldots\ s'_m)$

17.1. Non-expert queries

$$\texttt{Mizar}\ (s_1\ \ldots\ s_n\ \texttt{implies}\ s'_1\ \ldots\ s'_m)$$

$$\texttt{rough max} - \texttt{count}(q_1, \ldots, q_n, q'_1, \ldots, q'_m)$$

$$q_i = \texttt{symbol}\ s_i\ [\texttt{notation}|\texttt{constructor}|\texttt{negative occur}]$$

$$q'_j = \texttt{symbol}\ s_j\ [\texttt{notation}|\texttt{constructor}|\texttt{positive occur}]$$

18. Further development

- more convenient browsing

18. Further development

- more convenient browsing
- queries considering “properties”

18. Further development

- more convenient browsing
- queries considering “properties”
- local lemmas obtained with data mining techniques