Using Mizar in Computer Aided Instruction of Mathematics

Andrzej Trybulec
Institute of Mathematics
Warsaw University
15-267 Białystok, ul. Akademicka 2
Poland

Peter Rudnicki
University of Alberta

1 Introduction

Before we can explain applications of Mizar in teaching mathematics we have to explain what Mizar is. And here we are facing a challenging task of presenting an evolving project for which the evolution is the essence of its development and growth.

What do we need to use computers in doing the actual mathematics? Cumulative processes characterize all sciences but mathematics employs an extreme version of them. They are believed to be based on a small collection of axioms from which the magnificent rest is systematically derived. Any contribution to the derivation requires enormous skills that are nowhere codified and usually acquired by osmosis.

Contributions to mathematics take form of articles that continue the work reported in other articles and then collected in monographs. How to find the facts sought for by a mathematician preparing their contribution? Such facts can be buried in other articles that may seem irrelevant to the task at hand. Can this process of accommodating the established knowledge be assisted by computers? Let us note that traditional bibliographic systems based on key words search offer only a limited help.

Once an article is finished in the opinion of its author the question arises of its validity and importance. The latter may be subject of taste so let us not raise it here. The former is the problem of proof checking which includes but is not limited to checking correctness of logical inferences.

With the moment a mathematician believed their contribution to be valid and important, certain social in nature process begins. This is the process of extracting pieces of the contribution and including them into the established knowledge base.

The heart of Mizar is a library of articles written in a mathematical vernacular designed by A. Trybulec. Two articles form the foundation of the library: the axioms of the Tarski-Grothendieck set theory and the axioms of strong arithmetic of real numbers. All other articles undergo verification by the Mizar software to be correct consequences of those axioms.

The Mizar processor assist the author of a new article in preparing available terminology and facts, verifies the claims of the article and extracts facts and definitions for inclusion into the library.

Is a working mathematician willing to use Mizar to report their discoveries? We are still some distance away from this being a common practice. The formidable task of building a rich Mizar library is currently the main effort of the project. As of now the library includes more than 300 articles contributed by more than 60 authors, with about 10 of them active on a long term basis.

Most of the library articles formalizes the fundamentals of introductory mathematics. Some of the articles contribute more advanced results:

- Stone theorem on representation of Boolean algebras,
- another Stone theorem on paracompactness of metrizable spaces,
- Hahn-Banach theorem,
- Banach-Tarski fixed point theorem,
- reflection lemma for ZF,
- Koenig's lemma,
- axiom of choice which in Tarski-Grothendieck set theory is just a theorem.

The development of the library is not tightly controlled, the contributors are free to pursue their interests. This policy resulted in some new mathematical results first reported in Mizar articles before announcing them to the community by more traditional means.

Some of the articles contribute results of applied mathematics, mostly by Japanese authors. They include mathematical models of random access machines, reasoning about programs for the machines, finite topologies as applied in pattern recognition, the theory of Petri nets.

One may consider the Mizar achievements as modest when one learns that the project started in 1973. The techniques and technology for maintaining the library date back to 1988 with the first article accepted to the library on January 6, 1989. This was the article on Boolean properties of sets.

Let us give some technical details of the Mizar system. Please note that we are not addressing here the creative aspects of mathematical work. Let us focus on routine efforts facing a mathematician reviewing a paper.

```
A mathematical paper -------|
|                           Preliminaries
|                           |
|                           v
|    Books and journals --> Accommodation --> Environment of the paper
|                                             |
|                                             |
|----------------------->   Verification  <----
                            |
                            v
                            Remarks
```

Similar activities of many mathematicians form a part of the social process of doing mathematics. What remains of a paper once there are no negative remarks? The facts stated in the paper are slowly extracted into monographs to be referred by many in the future. Time shows that only small piece of the entire paper survives.

The reality is probably much more complicated but in Mizar we try to mimic the above picture. To automate the accommodation process, we have to build a data base containing knowledge from the books and journals. For archival reasons we store entire papers but it is hard to work with this collection. The accommodations for new articles are build on top of data base which was automatically extracted from submitted articles. In comparison to the tools for accommodating articles and extracting reusable facts from them, the process of verifying a mathematical paper is relatively well defined. There were many attempts to automate the process of checking correctness of mathematical reasoning, and our is one among many.

The Mizar software is centered around three programs named

- Accommodator (80 kB of code),
- Extractor (162 kB),
- Verifier (237 kB),

implemented in Turbo Pascal 6.0 under DOS. In addition there is a bunch of auxiliary software facilitating the navigation in the data base.

The mechanics of preparing a Mizar article is as follows:

- The source text is prepared using arbitrary ASCII editor and typically includes from 500 to 5000 lines.
- The text is run through the Accommodator. The directives from the Environment-Declaration guide the production of the local working environment, specific for the article. The environment is produced from the available data base and stored in a number of auxiliary files.
- Now the verifier is ready to start the checking. Typical running time varies from seconds to minutes. The output gives remarks on unaccepted fragments of the source text.

These three steps are repeated in a loop until no errors are flagged and the text author is satisfied with their achievement. The accommodator is called only after the environment directives have changed.

The finished article is now submitted to the Library Committee of Mizar Society for inclusion into Main Mizar Library. The submitted article is subjected to a review and if needed the author must revise their paper. Accepted paper is now incorporated into the data base distributed to all the Mizar users. (Remark. There is a possibility of building private data bases.)

2 Anatomy of a Mizar article

A Mizar article consists of two parts:

- Environment-Declaration defining imports from the Mizar library
- Text-Proper containing the definitions and theorems that the article contributes to the library.

The Text-Proper is written in a standardized mathematical vernacular designed by A. Trybulec. Many a mathematician agree that they can follow a Mizar text with ordinary effort, even if they would not like to write a paper in such a demanding formalism. Any detailed discussion of the Mizar language is far beyond the scope of this presentation. To give you an impression of how a Mizar article looks, here is an example:

```
   environ
vocabulary RELATION, FUNC_REL;
signature TARSKI, RELAT_1;
definitions RELAT_1;
theorems RELAT_1;
begin

   theorem for R, S, T being Relation holds R · (S · T) = (R · S) · T
proof
let R, S, T be Relation;
let x, y be Any;
```

```
hereby assume [x,y] ∈ R · (S · T); then
consider z1 being Any such that
A: [x,z1] ∈ R & [z1,y] ∈ S · T by RELAT_1:43;
consider z2 being Any such that
B: [z1,z2] ∈ S & [z2,y] ∈ T by A, RELAT_1:43;
C: [x,z2] ∈ R · S by A, B, RELAT_1:43;
hence [x,y] ∈ (R · S) · T by B, RELAT_1:43;
end;
assume [x,y] ∈ (R · S) · T; then
consider z1 being Any such that
A: [x,z1] ∈ R · S & [z1,y] ∈ T by RELAT_1:43;
consider z2 being Any such that
B: [x,z2] ∈ R & [z2,z1] ∈ S by A, RELAT_1:43;
C: [z2,y] ∈ S · T by A, B, RELAT_1:43;
hence [x,y] ∈ R · (S · T) by B, RELAT_1:43;
end;
```

We owe the reader some explanation. The directive vocabulary indicates that we need symbols from the mentioned vocabularies that are kept in the data base.

Each of the used library directives informs the accommodator which part of the listed articles is needed for our text.

- signature announces that we are using notation introduced elsewhere,
- definitions announce that structure of some proofs will reflect definitions introduced elsewhere (in our case it is the redefinition of equality for two relations),
- theorems indicate to the accommodator that we make references to theorems proved elsewhere.

An interested reader may have a look at a different proof of the same fact.

```
  theorem for R, S, T being Relation holds R · (S · T) = (R · S) · T
proof
let R, S, T be Relation;
now let x, y be Any;
hereby assume [x,y] ∈ R · (S · T); then
consider z1 being Any such that
A: [x,z1] ∈ R & [z1,y] ∈ S · T by RELAT_1:43;
consider z2 being Any such that
B: [z1,z2] ∈ S & [z2,y] ∈ T by A, RELAT_1:43;
take z2;
thus [x,z2] ∈ R · S & [z2,y] ∈ T by A, B, RELAT_1:43;
end;
given z1 being Any such that
A: [x,z1] ∈ R · S & [z1,y] ∈ T;
consider z2 being Any such that
B: [x,z2] ∈ R & [z2,z1] ∈ S by A, RELAT_1:43;
C: [z2,y] ∈ S · T by A, B, RELAT_1:43;
hence [x,y] ∈ R · (S · T) by B, RELAT_1:43;
end;
hence thesis by RELAT_1:def 8;
end;
```

Getting acquainted with Mizar is a rather challenging task especially due to the lack of documentation. The new Mizar users get the needed skills while spending a couple of weeks with someone from the Mizar team and writing an article.

3 Experience with Mizar MSE

In 1982, a small subset of the system was defined to facilitate the process of presenting Mizar to interested parties. Mizar MSE, as it was called, found application in introductory logic teaching at many sites. Articles in Mizar MSE are self contained and the environment part instead of pointers to the data base, contains the needed axioms and theorems without their justification. The language of Mizar MSE covers natural deduction in Jaskowski style. The syntax of atomic formulae is very restricted and terms may take only a form of a variable. The system employs many sorted logic with equality as the only interpreted predicate.

Despite its apparent simplicity, Mizar MSE proved adequate for practicing basic reasoning techniques of the full Mizar and for many users smoothed the process of becoming Mizar experts.

In Mizar MSE based logic courses the stress was on mastering student skills in structuring proofs. At the University of Alberta, Edmonton, Canada such a course has been run for last 6 years at the Department of Computing Science with the enrollment of 300 students each year. Two parts form the course: one is on rigorous reasoning about small programs written in Dijkstra style and the second is devoted to the practice of reading and writing proofs in Mizar MSE. Some part of using MSE deals with manipulating uninterpreted formulae but most of the assignments which consist of doing some 50 proofs cover logic puzzles in the style of Smullyan, and then Boolean properties of sets and binary relations. The checker of Mizar MSE works on the same principles as the verifier of the full Mizar, albeit it is much simpler. Despite that, one of the perceived obstacles in using MSE is the strength of the proof checker. Many inferences obvious to checker are beyond the grasp of beginners. It is planned to implement a version of the checker which will require the specification of rules of inference used at a step. The benefits of using Mizar MSE in this course can be summarized as follows:

- for many students it is the first time they see a proof in all the detail,
- students start believing that even larger proofs can be understood if one realizes their structure,
- there are substantial savings on the assignment marking effort as the most tedious work is done by the machine,
- the teaching assistants are more frequently consulted on the idea of how to do a proof and not on the details of syntax,
- students build their self confidence as they know whether their assignment is correct before submitting it,
- the instructors of some senior courses (analysis of algorithms, artificial intelligence) appreciate students habits of first seeing the structure of the proof before going into details,
- as an unexpected side effect of the course, we heard students' claims that practice with rigorous reasoning helps them in structuring their writings in English.

Mizar MSE software is in public domain and nobody tracks where and how it is used. The universities where Mizar MSE was or is used for instruction include: Gent Rijks Universiteit, Belgium, Sofia University, Bulgaria, Science University of Tokyo, Japan, Universidad Politecnica de Madrid, Spain, Nantes University, France, Washington University in St. Louis, USA, University of Connecticut in Storrs, USA, Lodz University and Warsaw University in Poland.

4 Master theses in Mizar

During 6 last years full Mizar has been routinely used for master theses in mathematics (ca. 20). Master students have played an essential role in building Main Mizar Library. A typical topic of such a thesis was to formalize in Mizar a fragment of handbook material. For example: introduction to connected spaces or introduction to compact spaces. Such work required the students to cooperate in preparing shared terminology.

There is another category of master theses that result in Mizar articles not included into the data base. These works had experimental character as students worked in areas foundations of which have not been formalized yet. In such case a student was allowed to prepare auxiliary articles stating the necessary prerequisites without justification.

5 Mizar based topical courses in mathematics

Experience shows that it is hardly feasible to use the full Mizar data base for a topical course in mathematics. The base contains articles from various areas and efficient usage of the base requires a lot of experience which cannot be collected during a one semester course. This is mainly caused by very poor tools for navigating the data base, practically restricted to textual searches. The need for better tools is clear but they are not available as of now.

There is a possibility in Mizar to prepare a local, private data base that is tailored for a narrowed, topical need. As an example consider a scenario for a Mizar based course on lattice theory. The Mizar library contains a number of articles on the subject that which include some advanced results. Starting with the existing articles the instructor can prepare a monographic article that will constitute the local data base for students taking the course. Such an article is not intended for inclusion into the main data base.

6 Other applications of Mizar

There are other cases when creation of a local data base is useful. It is a common practice among the Mizar authors, but let us have a look at the following application. When proving properties of programs it is convenient to start with a given description of the program meaning. It is certainly possible to develop the description by some tedious means which are hardly of interest when we want to expose the students to techniques of reasoning about a single program. The description of the program can put into a separate article and the student access is then restricted to referring it. We have conducted such experiments and it may be worthy mentioning that for programs of 10 instructions the description was of the order of 50 lines while the proof of program correctness was about 200 lines of Mizar text.