Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 63–68

ISBN 978-83-60810-51-4

Towards Automatically Categorizing Mathematical Knowledge

Adam Grabowski
Institute of Informatics
University of Białystok
Akademicka 2, 15-267 Białystok, Poland
Email: adam@math.uwb.edu.pl

Christoph Schwarzweller
Department of Computer Science
University of Gdańsk
Wita Stwosza 57, 80-952 Gdańsk, Poland
Email: schwarzw@inf.ug.edu.pl

***Abstract*—Clearly, there is no definitive standard for categorizing information contained in mathematical papers. Even if AMS Mathematics Subject Classification was important for mathematicians for years, nowadays we can observe growing popularity of other schemes, e.g. arXiv categories. On the other hand, in the era of digital information storing one can expect from the process of classification to be more or less automatic. Furthermore, generic categorization can be done inside the search engine.**

At different level, the distinction between such classical tagging items as lemma, proposition, theorem etc. had the aim of showing importance of proven facts. Here the automatization is much harder, or, to be more precise, the results obtained can be far from the original tagging given by the author. In the paper we point out some problems and thoughts concerned with the categorization of mathematical knowledge, illustrating some of them by examples taken from the Mizar Mathematical Library, large machine-checked repository of mathematical facts.

I. Introduction

THE design, construction, and maintenance of mathematical knowledge repositories is at heart of mathematical knowledge management. Computer-supported processing of mathematics such as theorem proving, knowledge retrieval, distribution over the Internet, or development of lecture material is highly driven by the way mathematical knowledge is represented and maintained. And last but not least, the acceptance of mathematical knowledge management systems by mathematicians themselves also depends in essence on how the knowledge in such systems is represented, developed and used.

Over the last years a lot of efforts have been spent in building large repositories containing more and more advanced computer-verified mathematical knowledge, such as for example the Mizar Mathematical Library (MML) [8] or the Coq Library [3]. It is a challenging task to keep such big repositories manageable in the sense that users can search – and find – mathematical knowledge supporting their own developments. This task has been addressed by structuring repositories and developing efficient search tools. The Coq library, for example, has been divided into a basic library, a standard library and a part containing users' contributions. For MML there exists a promising search tool – MML Query [2]. Also there have been efforts to build encyclopedias within the library, that is to collect knowledge on common topic in adequate places.

Both organization of the library and search tools, however, ignore two kinds of information mathematicians usually add to their newly discovered mathematical knowledge items. Firstly, mathematical knowledge is categorized by the topic an item – usually a whole article – is about. In most cases the AMS classification is used, but we are not aware of a repository supplying an AMS-style categorization for browsing and searching. Such information is usually tried to give by appropriate file naming. Secondly, inside a paper mathematicians categorize items according to their importance: besides definitions we find for instance lemmas, (main) theorems, or facts. In repositories each item actually is a theorem. This stems from using theorem provers to verify the included knowledge. One finds some exceptions, where more information is given as a comment, e.g. in MML theorem `POLYNOM5:74` is preceded by the comment *Fundamental Theorem of Algebra* identifying it in fact as a main theorem in the sense above. This, to our knowledge, however, has no further impact on searching in MML.

We believe that adding such information to the knowledge in repositories would greatly improve the possibility of both reading and searching, and even may enhance mathematicians to – at least – have a little look what's going on there. It also would enable a better structuring of the libraries allowing (possible) new users to get an overview of the repository: information about both mathematical areas addressed by a repository and important knowledge from an area that has already been formalized within the repository can then easily be generated. In this paper we point out some problems and thoughts concerned with the categorization of mathematical knowledge, illustrating some of them by examples taken from the Mizar Mathematical Library.

The paper is organized as follows: in the second section we focus on the (informal) classification of mathematical statements, giving some correspondence of these with knowledge repositories in Section III. Fourth section contains the description of the experiments with the Mizar formalization of mathematical text while in the next section we inspect the repository as a whole. At the end we point out potential benefits and conclude the paper. We did not expect especially deep results due to informal nature of the topic and of the fuzziness of selected categories of propositions, but at least we can explain the incoherence in obtained experimental data.

II. Importance of Mathematical Knowledge Items

In the following we discuss how mathematicians label their results with different phrases in order to underline their importance in the course of an article or a textbook. We start with definitions from Oxford Dictionary [16] and Encyclopedia Britannica [6], and analyze how this fits with mathematical intuition, at the informal level.

Mathematicians use quite a number of phrases to label knowledge items, most of them of course statements that come with a proof. The most obvious phrases here are *theorem* and *proposition*, explained in Encyclopedia Britannica by

> *theorem, in mathematics and logic, a proposition or statement that is demonstrated*

and in Oxford Dictionary by

> *theorem, a general proposition not self-evident but proved by a chain of reasoning; a truth established by means of accepted truths*

and

> *proposition, a formal statement of a theorem or problem, typically including the demonstration.*

So, a *theorem* and a *proposition* are essentially the same, only that *proposition* underlines that the topic is a formal one. More interesting that the Oxford Dictionary emphasizes the fact that a theorem is not self-evident, a view that obviously is shared by mathematicians. For items that are self-evident we find in Oxford Dictionary

> *fact, a thing that is known or proved to be true*

This, of course, can be a bit irritating from a mathematician's viewpoint, as one can conclude that every *theorem* is a *fact* and vice versa that every *fact* is a *theorem*. The usual mathematician's use of *fact* is that a *fact* either is so obvious that there is no need for a proof or that the statement (already) is common knowledge.

Quite similar to *facts* are *corollaries*. They describe knowledge that is rather obvious, however, with respect to some other usually more involved piece of knowledge. The Oxford Dictionary defines them as

> *corollary, a proposition that follows from (and is often appended to) one already proved, a direct or natural consequence or result*

Mathematicians use *corollary* to label obvious or easy consequences of a just demonstrated theorem. Sometimes, in fact, this causes the main result of a paper being labeled as a *corollary* just because it follows (easily) from another more general theorem maybe easier to prove than the original intended result.

The most promiscuous label is the *lemma* which is explained in Oxford Dictionary by

> *lemma, a subsidiary or intermediate theorem in an argument or proof*

Note that here a *lemma* is just a special kind of theorem, so it heavily depends on the author whether he considers this result as intermediate or not. This can have amusing consequences. For example, the following statement

> *Suppose a partially ordered set P has the property that every chain, i.e. totally ordered subset, has an upper bound in P. Then the set P contains at least one maximal element.*

is known as Zorn's lemma or as Kuratowski-Zorn lemma. From its content though, it should be rather labeled as a *theorem* – or, as it can be considered as common knowledge as a *fact*?

A complete different kind of labeling are *examples*. Not restricted to mathematical use we find the following definition in Oxford Dictionary

> *example, a thing characteristic of its kind or illustrating a general rule*

Mathematicians usually use examples for illustrating definitions or properties of proven results. Examples often, however, include statements that are to be proved, such as *the integers are a ring* or *every field is a ring*. In the sense from above these two would be *facts*, so this is not really a problem. But in general mathematicians are much more generous, especially in text books or lectures: One of the authors remembers an algebra lecture in which arithmetics in $\mathcal{Z}[\sqrt{-5}]$ was illustrated by an example. Right after the example one learns, that it follows $2 = 2 + 0 \cdot \sqrt{-5}$ is irreducible, but not prime, hence

> *There are irreducible elements that are not prime.*

So hidden in this *example* actually is (the proof of) a *theorem* or a *lemma*, which in the lecture – and the accompanying text book – was not more than a remark.

III. Importance of Mathematical Knowledge Items in Repositories

In this section we discuss how the phrases from the previous section can be adapted for use in mathematical repositories. The goal is to supply repositories with more information than that each proved mathematical item is a theorem. First of all there is of course the possibility to just introduce lemma, fact etc. as synonyms for theorem and leave the labeling to the user. It may be interesting to see such a strategy would result in. We, however, would prefer to automatically generate such information from the repository itself, that is from the proofs accompanying mathematical items. So, the question is: how to distinguish between lemmas and theorems and facts in the sense of the preceding section, if we have at hand the proofs of probably many different authors only? In the following we try to elucidate this question by comparing intuitive properties and intentions of the different kinds of labels.

Let us start with the difference between theorems and lemmas. A lemma by its mathematical intention describes a rather technical statement necessary in the course of proving another statement. One can therefore argue that the proof of a lemma therefore tends to be longer than the one of theorem. In a repository the length of a proof strongly depends on the author though. It looks more promising to focus on the references, that is on the use of other statements within the proof. Being more technically the proof of a lemma includes much more references than the one of a theorem. Another criterion can be that a statement of a lemma includes more detailed preconditions.

More practically we note that knowledge in repositories usually is grouped by packages or articles. Then a statement not being referenced inside can be (automatically) considered as a theorem. In addition, work described in [17] might help. Having a large proof in a repository one can label its statement as a theorem. Lemma extraction then generates a number of statements to be labeled as lemmas.

Facts and corollaries are quite similar. A fact easily follows from a definition, so that its proof is short, that the proof contains a given number of references only, and the references used in its proof are restricted: they include the definition only plus maybe some other references to basic statements about basic topics, such as the natural number or Boolean properties. For corollaries holds almost the same, the only change is that the role of the definition here is played by a theorem.

There is another important point on lemmas. Though not having a detailed definition of a lemma, the intuitive use is to cut off parts of a proof. One can distinguish two reasons for that: Firstly, it is just to improve readability of the whole proof. Rather technical parts of the proof are divided from the main line of argument. Secondly, an intermediate result of a proof can be considered so important that it is cut off and stated as a lemma. Of course this distinction is blurred and the view on a lemma can change over time and depending on the author/reader. In repositories the first case naturally is much more present: a lot of technical details have to be explicitely shown. In MML we find, for instance

```
theorem :: FUNCT_2:3
  for f being Function st
    dom f = X & for x st x in X holds f.x in Y
  holds f is Function of X,Y;
```

This property is often used in proofs and is therefore stated on its own. From the above follows that it can be labeled as a fact. This however would ignore that it is uninteresting from a mathematician's point of view. Therefore we propose to introduce an additional kind of lemma in mathematical repositories: a *technical lemma*. A *lemma* states an interesting piece of mathematical knowledge though not being important enough to be a theorem. By contrast a *technical lemma* (or a *technical fact*) just states a property often necessary in other proofs. Note that this allows for two kinds of reading a repository: first, ignoring technical knowledge concentrating on mathematically interesting knowledge and second, reading technical knowledge to find statements that can help in one's own proof.

We close with a note on *examples*: identifying examples is extremely difficult. Not only due to our discussion from the last section, but one also finds other examples characteristic of mathematical repositories. In the article FINSEQ_1, for instance, we find the following definition for segments of natural numbers

```
definition
  let n be Nat;
  func Seg n -> set equals :: FINSEQ_1:def 1
    { k where k is Element of NAT :
      1 <= k & k <= n };
end;
```

and right after that between other theorems the following.

```
theorem :: FINSEQ_1:2
  Seg 1 = { 1 } & Seg 2 = { 1, 2 };
```

Should this be a *theorem* (*fact*) or rather an *example*? It obviously illustrates the definition of Seg. On the other hand, its proof allows to identify it as a *fact* as the proof is rather short – about 30 lines of Mizar code – and uses basic theorems about sets and natural numbers only.

IV. EXPERIMENTS WITH MIZAR

As a testbed for our experiments we have chosen repository of mathematical texts written in the Mizar language. It is authored by over 200 authors with nearly 1200 articles – so we expected from the results not to be meaningless.

A. MML Items Numbering Scheme

Formalization, apart from machine-verification, offers new possibilities to analyze proofs automatically, even at pure syntactic level. We focus on references in Mizar proofs, which can be roughly divided into library references (from other files), references for exportable items from the current article and auxiliary labels. More precisely, there are

1) library references, e.g. MMLID:<num1>;
2) library definitional references, e.g. MMLID:def <num1>;
3) a kind of article self-references – Th<num>: – exportable theorem;
4) the same as above, but for definitions – Def<num>: – exportable definition (the only definition which is not exportable is a private predicate or private functor definition which is not labelled as a rule);
5) internal lemmas – Lm<num>: – Mizar *lemma*, i.e. theorem which is not exportable to the database (not flagged by theorem keyword);
6) auxiliary labels, e.g. A<num>: – the <num> resets every time main first-level proof block starts;

where <num1> and <num> are numerals.

From the above list first two are obligatory for the author, the rest is unified after the inclusion of the article in the Mizar Mathematical Library (the author can use his own numbering scheme, so potentially also the distinction for lemmas, theorems etc. is possible). We will see the concrete example flagged according to the abovementioned rules in the succeeding subsection.

B. Single Concrete Mizar Article – an Example

As an example, we can take arbitrary article [13] with the tags marking lemmas, propositions etc. and compare with the Mizar source. We have chosen this specific file just because it faithfully reflects the real established mathematical textbook – *A Compendium of Continuous Lattices* by Gierz, Hofmann et al (CCL for short – [7]). Hence, all facts are categorized by the authors of the monograph. Furthermore, we were curious how fifteen years of the evolution of the language and numerous revisions of the repository affected this article.

Pragmas are relatively recent concept implemented in the Mizar language – `::$N` is transparent for the verifier since it begins with two colons – a comment sign, but it allows the authors to identify the most important items in the article (apart from the usual possibility of inserting comments).

```
::$N Baire Category Theorem for Continuous
::   Lattices
theorem Th39: :: Theorem 3.43.7
  for L being lower-bounded continuous LATTICE
  for D being non empty countable dense
          Subset of L,
      u being Element of L st u <> Bottom L
   ex p being irreducible Element of L st
   p <> Top L & not p in uparrow ({u} "/\" D)
proof
  let L be lower-bounded continuous LATTICE,
    D be non empty countable dense Subset of L,
    u be Element of L such that
A1: u <> Bottom L;
A2: for d, y being Element of L st
      not y <= Bottom L & d in D holds
    not y "/\" d <= Bottom L
  proof
    let d, y be Element of L such that
A3: not y <= Bottom L;
    assume d in D;
    then d is dense by Def5; then
A4: y "/\" d <> Bottom L by A3,Def4;
    Bottom L <= y "/\" d by YELLOW_0:44;
    hence thesis by A4,ORDERS_2:2;
  end;
  Bottom L <= u by YELLOW_0:44;
  then not u <= Bottom L by A1,ORDERS_2:2;
  then consider p being irreducible
    Element of L such that
    Bottom L <= p and
A5: not p in uparrow ({u} "/\" D) by A2,Th36;
  take p;
  thus p <> Top L by A5,Th9;
  thus thesis by A5;
end;
```

For simplicity of further considerations, we ignore the structure of the proof (as we do not take into account implicit mechanisms of the Mizar verifier anyway), hence it can be represented just as

```
WAYBEL12:39 = (WAYBEL12:def 5, WAYBEL12:def 4,
  YELLOW_0:44, ORDERS_2:2, YELLOW_0:44,
  ORDERS_2:2, WAYBEL12:36, WAYBEL12:9)
```

Observe `Th36` and `Th9` were resolved into corresponding library references (the same with `Def4` and `Def5`). Also multiple uses of the same theorem is visible. Resolving all items from [13] in this way we collected all tagged items in Table I.

These results are not very convincing, with one exception: all three Mizar theorems (out of 44 in the file covering 7 numbered propositions formalized from CCL) with the biggest number of outside references were taken as significant by the authors of CCL.

TABLE I
TAGGED ITEMS AS TAKEN FROM [13].

MML Item	MML exportable item	refs in proof
WAYBEL12:33	Proposition 3.43.1	40
WAYBEL12:34	Corollary 3.43.2	11
WAYBEL12:35	Proposition 3.43.3	9
WAYBEL12:36	Corollary 3.43.4	4
WAYBEL12:39	Theorem 3.43.7 – main theorem	6
WAYBEL12:43	Theorem 3.43.8	22
WAYBEL12:44	Corollary 3.43.10	19

TABLE II
MML PROOFS WITH THE BIGGEST NUMBER OF OUTSIDE REFERENCES.

No.	MML exportable item	Outside references
1	`JORDAN13:def 1`	218
2	`JORDAN9:def 1`	203
3	`JORDAN15:46`	193
4	`JORDAN15:47`	193
5	`JORDAN19:22`	193
6	`JORDAN19:23`	193
7	`JORDAN15:44`	192
8	`JORDAN15:45`	192
9	`JORDAN19:20`	192
10	`JORDAN19:21`	192

Some other interestingness measures can be e.g.:

- The length of the proof - measured either by the number of proof steps or just in terms of words used.
- The number of times a fact is used in a library (as shown later on in Table III).
- The complexity of premises (longest assumptions usually imply more technical fact while corollaries have the premises much simplified).

V. Statistical Data – Mizar Mathematical Library

Of course, even if the file from the previous section can represent a real paper written by mathematician, we wanted to have a wider look for the mathematical knowledge repository, so similar experiment as in case of the single article we did on the whole MML (version 4.181.1147). Applying similar techniques as in the preceding section we collected theorems in MML which have biggest number of outside references in Table II, keeping in mind that these are only exportable theorems, just forgetting about all statements called informally *technical lemmas* in Mizar jargon.

The results appeared a bit surprising for us. As we mention exportable “theorem” items, why definitions are ranked so high (in fact accidentally we counted all exportable items, not only theorems)? We checked that the article with the MML identifier `JORDAN13` [18] contains only this single definition, and nearly 4500 lines and 1200 numbered statements in there are only to justify the existence and the uniqueness of the object called *span* for simple closed curve in the Euclidean plane. Furthermore, most of facts from `JORDAN15` and `JORDAN19` are just mirror cases, hence the identical number of references.

The solution is to extract some technical lemmas (and here methods similar to those described e.g. in [20] might help) but inspecting the proof we have seen that the complication of the construction of the object makes it at least not sufficiently justified. What can be done easily in this case is that two lemmas can be formulated outside of the definition block and complicated proofs can be replaced by straightforward references. But in such a case the policy of the Library Committee of the Association of Mizar Users is not to export such lemmas to the public database.

MML item `INT_5:49` is number 19 with 151 theorems referenced in the proof – the first not-Jordan fact if we take into account the number of library references (the law of quadratic reciprocity).

Observe that we count here only explicit references, and many additional implicit arguments, such as even just rules of reasoning or definitional expansions can potentially break the expected results. Nearly 80% of MML items immediately depend on at least one of these implicit type mechanisms (according to [1]), but as long as the Mizar system lacks the proof object generation which can be called by ordinary user, with the flexible level of verbosity, these results are not binding. For example, by proper application of mechanism of registration of clusters [8], the proofs can be significantly shortened.

Potentially, the weight of the theorem should be seen from a wider perspective – it depends also on the context, so formalizing significant and complicated result one might merge theorems taken from various theories, e.g. category theory etc. In such context, e.g. arithmetical theorems just vanish and even Fundamental Theorem of Arithmetic could obtain no explicit reference in such informal proof.

Just as a curiosity, we checked the three fundamental theorems proven within MML and one of the probably best known as the formalization challenge – and the number of library references in proof were as follows:

- Fundamental Theorem of Integral Calculus – 13;
- Fundamental Theorem of Arithmetic – 42;
- Fundamental Theorem of Algebra – 81;
- Jordan Curve Theorem – 1(!).

A. The Statistics of MML

The Mizar Mathematical Library contains 51,762 Mizar theorems (which should be named *propositions* or *facts* rather than theorems).

Nearly one third of them can be treated either useless or terminal objects at the current state of the library because:

- 16,354 were not used in the MML;
- 14,940 were used exactly once.

Here we can count additionally 5,725 Mizar lemmas (marked `Lm`) and not exportable – which are used at least once.

According to definitions from Section 2 some 14% of propositions is more or less trivial:

- 4,686 *corollaries* (propositions justified by a single library reference);

TABLE III
IMPORTANT LEMMAS IN MML AND THE COMPLEXITY OF THEIR PROOFS

	Name of the fact	MML Identifier	Refs	Used
1	Alexander's Lemma	`WAYBEL_7:31`	32	2
2	Contraction Lemma	`ZF_COLLA:12`	12	0
3	Dickson's Lemma	`DICKSON:freg 15`	8	0
4	Dynkin Lemma	`DYNKIN:24`	5	1
5	Fatou's Lemma	`MESFUN10:7`	29	0
6	Gauss Lemma	`INT_5:41`	117	2
7	Koenig Lemma	`TREES_2:30`	34	0
8	Lebesgue's Lemma	`UNIFORM1:6`	39	1
9	Sperner's Lemma	`SIMPLEX1:47`	114	1
10	Urysohn Lemma	`URYSOHN3:20`	54	3
11	Yoneda Lemma	`YONEDA_1:freg 3`	51	0
12	Zassenhaus Lemma	`GROUP_9:93`	4	1
13	Zorn Lemma	`ORDERS_1:65`	17	9

- 2,371 trivial propositions or *facts* (no library references or no proof at all).

Only 9,439 propositions have ten on more references, but hopefully it allows the human reader of the proofs to track the idea of the proof, splitting large and complex proofs for small verifiable steps. Here, the readability can be an objective; total 333,172 references in proofs of 51,762 theorems gives an average 6.44 references in a proof. Only 2,669 theorems' proofs (which makes about 5% of all MML propositions) contain more than 20 references, so we can establish 100 refs as a good threshold for a Mizar theorem to be a *technical lemma*.

B. Important Lemmas

In Table III we collected some well-known mathematical lemmas. Although as a rule, *lemmas* should be rather of intermediate character, statistical data on MML shows that many of them were proven just for their importance, not reusability. This table unleashes however the importance of the Zorn Lemma – it was referenced 9 times in MML. The other important lemmas formalized in Mizar are rarely mentioned. Notable exceptions are zeroes in case of Dickson's and Yoneda lemmas, because they do not need be referenced due to the mechanism of registration of clusters.

In case of items numbered 7, 9, and 10 the primary results were misleading – in fact the lemmas were straightforward corollaries, hence we gave the complexity of proof of the original technical lemma.

Our feeling was that repositories have much more lemmas than theorems. Experiments have shown that in case of Mizar these numbers are comparable. We performed additional experiments both on mathematical books (with the ratio of lemmas : theorems : corollaries as 260 : 94 : 23) and on computer repositories (e.g. in Isabelle's AFP we counted ratio 27523 : 1061 : 513). The MML is closer to mathematical textbook with the exception of the (bigger) number of corollaries.

VI. POTENTIAL GAIN: FORMALIZED MATHEMATICS JOURNAL

Formalized Mathematics (ISSN 1426-2630) is a journal publishing papers automatically generated from Mizar abstracts. As the first issue is dated back in 1990, and the Mizar language itself evolves much faster than the translating software, its syntax is relatively modest.

As of the time of writing this paper, to ensure better human-readability of the journal one of the following constructions were used before the propositions (we observed the keyword "theorem" is not used in FM):

- "One can prove the following proposition..."
- "The following propositions are true..."
- "Next we state (three, several, ...) propositions..."
- "We now state the proposition..."

with sentences of the form

- "One can check that..."
- "Note that..."
- "One can verify that there exists..."
- "Observe that there exists..."

reserved for registrations which can have even more complex proofs than the aforementioned propositions.

Also

- "Let us observe..." or
- "... can be characterized by the condition..."

kept for Mizar redefinitions can be better used in some other context.

At least corollaries (or trivial theorems, if really needed) identified in the previous section can be introduced by phrases of the form

- "It is easily seen" or
- "As a direct corollary of ... we get..."

We hope that such suggestions will be implemented in future issues of the *Formalized Mathematics* journal.

VII. Conclusion

At first glance, automatic tagging of propositions seems to be useless for machine math-assistants – it is meaningless whether the prover uses lemma or theorem as a hint. But it can be of bigger importance if we take into account that the described statistics offer a kind of interestingness measure for knowledge discovery. Here potentially theorems with more complex proofs can give more possibilities for reuse.

In our opinion the point is that though mathematicians don't use tags consistently, one can adapt (and extend) these notations to improve organization of mathematical knowledge in repositories. Furthermore, our experiments give evidence that to a certain extent these notations can be attached automatically. Such general classification rules and the improvement of Mizar library are the main contribution of our work. We can also enhance the presentation of the formal text to be more attractive to ordinary mathematician.

If we take into account the possibility of labeling propositions by the author, the Mizar syntax is rather fixed – here other systems are a bit more flexible: Isabelle [11], offers interchangeability of `corollary` and `lemma` keywords depending on author's choice. The same with Coq [3], where `Theorem` and `Lemma` are present.

The importance of theorems can be measured in many ways – but if e.g. we resolve all dependencies contained in a single proof, the notions of lemma and corollary can loose their intended meaning. But as the compression of the text made the syntactical differences between various formal languages unimportant for the value of the de Bruijn factor [21], the simple measure we have chosen for our considerations also seems to be acceptable. Furthermore, revisions of handwritten proofs with the help of automatic theorem-provers as proposed by Urban [19] or the linking between both informal (as Wikipedia) or other machine-checked formal repositories (e.g. Archive of Formal Proofs) to eliminate accidental tagging can offer a kind of error-correction. On the other hand, the distinction between various kinds of machine-verified facts might catch the human's eye much better, which looks like one of the potential aims for computer math-assistants.

References

[1] J. Alama, L. Mamane, and J. Urban, *Dependencies in Formal Mathematics*, available from arxiv.org.
[2] G. Bancerek, *Information Retrieval and Rendering with MML Query*, in: J.M. Borwein and W.M. Farmer (eds.), Proc. of MKM 2006, Lecture Notes in Artificial Intelligence 4108, pp. 266–279, 2006.
[3] *The Coq Proof Assistant*, available at http://coq.inria.fr.
[4] J. H. Davenport, *MKM from Book to Computer: A Case Study*, in: A. Asperti, B. Buchberger, and J. Davenport (eds.), Proc. of MKM 2003, Lecture Notes in Computer Science 2594, pp. 17–29, 2003.
[5] N.G. de Bruijn, *The Mathematical Vernacular, a language for mathematics with typed sets*, in: P. Dybjer et al. (eds.), Proceedings of the Workshop on Programming Languages, Marstrand, Sweden, 1987.
[6] Encyclopedia Britannica, available at http://wwww.britannica.com.
[7] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott, *Continuous Lattices and Domains,* Cambridge University Press, 2003.
[8] A. Grabowski, A. Korniłowicz, and A. Naumowicz, *Mizar in a Nutshell*, Journal of Formalized Reasoning, 3(2), 2010, pp. 153–245.
[9] A. Grabowski and C. Schwarzweller, *Translating Mathematical Vernacular into Knowledge Repositories*, in: M. Kohlhase (ed.), Proceedings of the 4th International Conference on Mathematical Knowledge Management, Lecture Notes in Artificial Intelligence 3863, pp. 49–64, 2005.
[10] J. Harrison, *The HOL Light Theorem Prover*, available at http://www.cl.cam.ac.uk/~jrh13/hol-light.
[11] The Isabelle Home Page, available at http://www.cl.cam.ac.uk/research/hvg/Isabelle.
[12] F. Kamareddine and R. Nederpelt, *A Refinement of de Bruijn's Formal Language of Mathematics*, in: Journal of Logic, Language and Information, 13(3), pp. 287–340, 2004.
[13] A. Korniłowicz, *On the Baire Category Theorem*, Formalized Mathematics, 6(2), 1997, pp. 321–327.
[14] The Mizar Home Page, http://mizar.org.
[15] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, 1999.
[16] Oxford Dictionaries, available at http://oxforddictionaries.com.
[17] K. Pąk, *The Algorithms for Improving and Reorganizing Natural Deduction Proofs*, Studies in Logic, Grammar and Rhetoric, 22(35), 2010, pp. 95–112.
[18] A. Trybulec, *Introducing Spans*,Formalized Mathematics, 10(2), 2002, pp. 97–98.
[19] J. Urban and G. Sutcliffe, *Automated Reasoning and Presentation Support for Formalizing Mathematics in Mizar*, in: S. Autexier et al (eds.), Proceedings of AISC/Calculemus/MKM 2010, Lecture Notes in Artificial Intelligence 6167, pp. 132–146, 2010.
[20] J. Vyskocil, D. Stanowsky, and J. Urban, *Automated Proof Compression by Invention of New Definitions,* LPAR 2010, Lecture Notes in Computer Science 6355, 2010, pp. 447–462.
[21] F. Wiedijk, *The De Bruijn Factor,* available from www.cs.ru.nl/~freek/factor/.