Linking to Compound Conditions in Mizar

Adam Naumowicz

Institute of Informatics, University of Białystok
Konstantego Ciołkowskiego 1M, 15-245 Białystok, Poland
adamn@math.uwb.edu.pl

Abstract

The processing and analysis of the Mizar library has been performed using the infrastructure of the University of Bialystok High Performance Computing Center.

In the current Mizar language, direct linking to statements formulated as compound conditions is prohibited. This particular language feature has often been considered by the users as an unnecessary disruption of the natural course of proof steps. In this paper, we present an analysis of the linking structure in the current contents of the Mizar Mathematical Library which provides statistical data needed to implement a usability-based solution to this problem and propose corresponding changes in the library.

1 Introduction

One of the main design principles of the Mizar [Ban15] language has been to enable writing formal mathematical documents checked by a computer for syntactical, semantical and logical correctness, while at the same time the language should as much as possible resemble standard mathematical papers. The language's richness of features adopted from the natural language contributed to its popularity, which in turn allowed to collect a vast body of formalized mathematical data available as the Mizar Mathematical Library (MML) over the four decades of Mizar's active use [Gra15]. However, there is still a lot of space for improvement, adding new features and eliminating some of the language's idiosyncrasies which hinder its comfortable use. A detailed analysis highlighting a number of difficulties posed by the complexity of the language was presented in a paper by Cairns and Gow [Cai04]. Among such problematic features which users often complain about is that direct linking to choice statements is prohibited in current Mizar. It is percived as an unnecessary disruption of the natural course of proof steps. In fact, this problem have re-appeared several times in the Mizar Forum mailing list discussions[1]. Apparently, this is a more general issue which concerns linking to many sorts of statements formulated as compound conditions.

2 Compound Conditions in Mizar

A brief look at the Mizar syntax[2] shows that there are quite a few constructs of the language that make use of compound conditions. A complete list of such constructs includes: `Loci-Declaration`, `Case`, `Suppose`, `Generalization`, `Collective-Assumption`, `Existential-Assumption`, and finally the `ChoiceStatement`. Their grammar specifications either explicitely mention the `that` and `and` keywords, or use the `Conditions` nonterminal symbol:

```
Conditions = "that" Proposition { "and" Proposition } .
```

[1] See e.g. the following discussion threads in the Mizar Forum mailing list's archives: `http://mizar.uwb.edu.pl/forum/archive/0104/msg00002.html`, and `http://mizar.uwb.edu.pl/forum/archive/0203/msg00001.html`.

[2] The Mizar language is described by its grammar available in the system's distribution (`syntax.txt` and `syntax.xml` files) as well as on-line on the project's website: `http://mizar.org/language/mizar-grammar.xml`

Using a Weakly-Strict Mizar (WSM) [Nau16, Byl12] parser[3], we have collected some statistical data which shows the usage of these constructs in the MML version (5.33.1254) distributed together with the current official Mizar version (8.1.04)[4]. The table below shows the number of occurences of these language constructs.

Table 1: Occurences of compound conditions in current MML.

Language: construct	Assumption	Loci-declaration/ Generalization	Case	Suppose	Existential assumption	Choice statement
With conditions:	17311	13811	59	587	2761	58909
W/o conditions:	91914	136733	4782	32136	n/a	n/a

As we can see, some of the constructs are significantly less frequent than others. Notably, `case` and `suppose` are used only in special kinds of proofs (reasoning *per cases*) and the use of compound statements in this context is usually connected with applying definitional expansion [Kor15]. Let us also note that loci declarations and generalizations both use the same `let` keyword and so syntactically they are similar (what makes them different is the context, since loci declarations are only available within definitions). In most contexts the `case` and `suppose` constructs can be used interchangeably and most users tend to prefer the more common `suppose`. The syntax of choice statements (`consider`) and existential assumptions (`given`) requires a `that` phrase, so they never occur without the conditions part, even if they are followed by a single proposition. In all other cases, a user might decide to use only the simple forms of the constructs, e.g. a sequence of single assumptions, and never use compound conditions. And so there are articles in the Mizar library whose authors avoided using compound conditions completely. However, the compound conditions are in general used quite often in the MML. The number of conditions used in an article ranges from 0 to 412 (total 94392). If we assume (very roughly) that each condition is represented by one line of formal text and we compare it to the total number of 2473054 lines of the WSM representation of all the articles, this gives about 4% of all the text content.

It is also worth mentioning that the existential assumption is actually a syntactic sugar construct which replaces an assumption followed by a choice statement - this language feature is rarely used by less experienced Mizar users who are interested primarily in the correctness of their proofs and not in their brevity or "good style".

Now let us look at the simplest example which demonstrates the underlying linking problem. Here we have a common choice statement with one chosen constant and two labelled compound conditions potentially related to this constant, and we want to immediately refer to this statement in the next proof step:

`consider x such that A1:` α `and A2:` β `by` *references*$_1$`;`
`then` γ `by` *references*$_2$`;`

In the original Mizar parser the linking (here using the `then` keyword) is not allowed, because it is ambiguous what it should link to. In consequence, the parser marks the second line with the "`164: Nothing to link`" error message. In fact, the comment might as well say "`Too many options to link`", because in this case there are four possible interpretations of the linked statement, namely:

1. `ex x st` α `&` β
2. α `&` β
3. α
4. β

Although there are cases when option 1. would be useful, usually reasoning in Mizar requires an existentially quantified statements to be eliminated in order to advance the proof. As for option 2., it would be quite useful for a small number of compounds. The average number of compounds in the current MML is 1.92, which makes this option potentially useful for accumulating available facts rather than being very selective in inference statements. However, there are cases of compound statements in the library composed of as many as 33 propositions[5] and in this case linking to a conjunction of all these propositions would produce too complicated inferences to be practically justified. Option 3. seems least useful and would be most misleading for the users, especially if there were more than two compound propositions involved. This leaves us with option 4., and apparently this

[3] The parser can be downloaded from a dedicated Git repository: `https://github.com/MizarProject/wsm-tools`.
[4] `http://mizar.org/`
[5] E.g. the `GATE_3` article developing the theory of digital circuits.

is the semantics adopted by two Mizar-inspired systems: Mizar Light for HOL Light developed by F. Wiedijk [Wie01] as well as Makarius Wenzel's implementation of the Isabelle/Isar language [Wen02]. Therefore, we have implemented the fourth option in an experimental Mizar version. Mizar verifiers pre-compiled for main supported software platforms are available for download and further experimentation at the author's website[6]. There you can also find an example of a simple set-theoretical Mizar article adapted to this new linking method (file `xboole_0.miz`) to demonstrate its usefulness.

3 Conclusions

First of all, the proposed enhancement of the Mizar parsing module might be beneficial for some Mizar users. But apparently, the inability of linking to compound statements is also one of the main obstacles that complicate automatic improving the legibility of natural deduction proofs in Mizar [Pak14] and make lemma selection based on the close reference principle equivalent to an NP-hard graph problem [Pak15]. So the enhancement might help advance the research on proof legibility. And finally, Mizar has been widely known for its influence on other proof assistant systems and their proof languages - this work is an example that the influence might as well work in the other direction.

3.0.1 Acknowledgement

The processing and analysis of the Mizar library has been performed using the infrastructure of the University of Bialystok High Performance Computing Center.

References

[Ban15] G. Bancerek, C. Byliński, A. Grabowski, A. Korniłowicz, R. Matuszewski, A. Naumowicz, K. Pąk, J. Urban, Mizar: State-of-the-art and beyond, in: M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, V. Sorge (Eds.), Intelligent Computer Mathematics – International Conference, CICM 2015, Washington, DC, USA, July 13–17, 2015, Proceedings, Vol. 9150 of Lecture Notes in Computer Science, Springer, 2015, pp. 261–279. doi:10.1007/978-3-319-20615-8_17.

[Gra15] A. Grabowski, A. Korniłowicz, A. Naumowicz, Four decades of Mizar, Journal of Automated Reasoning 55 (3) (2015) 191–198. doi:10.1007/s10817-015-9345-1.

[Cai04] P. A. Cairns, J. Gow, Using and parsing the Mizar language, Electr. Notes Theor. Comput. Sci. 93 (2004) 60–69. doi:10.1016/j.entcs.2003.12.028.

[Nau16] A. Naumowicz, R. Piliszek, Accessing the Mizar library with a Weakly Strict Mizar parser, in: Intelligent Computer Mathematics - 9th International Conference, CICM 2016, Bialystok, Poland, July 25-29, 2016, Proceedings, 2016, pp. 77–82. doi:10.1007/978-3-319-42547-4_6.

[Byl12] C. Bylinski, J. Alama, New developments in parsing Mizar, in: J. Jeuring, J. A. Campbell, J. Carette, G. D. Reis, P. Sojka, M. Wenzel, V. Sorge (Eds.), Intelligent Computer Mathematics - 11th International Conference, AISC 2012, 19th Symposium, Calculemus 2012, 5th International Workshop, DML 2012, 11th International Conference, MKM 2012, Systems and Projects, Held as Part of CICM 2012, Bremen, Germany, July 8-13, 2012. Proceedings, Vol. 7362 of Lecture Notes in Computer Science, Springer, 2012, pp. 427–431. doi:10.1007/978-3-642-31374-5_30.

[Kor15] A. Korniłowicz, Definitional expansions in Mizar, Journal of Automated Reasoning 55 (3) (2015) 257–268. doi:10.1007/s10817-015-9331-7.

[Wie01] F. Wiedijk, Mizar light for HOL light, in: Theorem Proving in Higher Order Logics, 14th International Conference, TPHOLs 2001, Edinburgh, Scotland, UK, September 3-6, 2001, Proceedings, 2001, pp. 378–394. doi:10.1007/3-540-44755-5_26.

[Wen02] M. Wenzel, F. Wiedijk, A comparison of Mizar and Isar, J. Autom. Reasoning 29 (3-4) (2002) 389–411. doi:10.1023/A:1021935419355.

[6] `http://mizar.uwb.edu.pl/~softadm/linking`

[Pak14] K. Pąk, Improving legibility of natural deduction proofs is not trivial, Logical Methods in Computer Science 10 (3) (2014) 1–30. doi:10.2168/LMCS-10(3:23)2014.

[Pak15] K. Pąk, Improving legibility of formal proofs based on the close reference principle is NP-hard, Journal of Automated Reasoning 55 (3) (2015) 295–306. doi:10.1007/s10817-015-9337-1.