Implementing More Explicit Definitional Expansions in Mizar – Appendix

Adam Grabowski
Institute of Computer Science, University of Białystok, Białystok, Poland

Artur Korniłowicz
Institute of Computer Science, University of Białystok, Białystok, Poland

Abstract

The Mizar language and its corresponding proof-checker offers the tactic of definitional expansions in proof skeletons. This apparatus is rather fragile in the case of intensive overloading of notions (which is widely observed e.g. in the field of algebra, but it is also present in the more fundamental set-theory contexts). We propose the extension of this mechanism: the change should offer users the more precise control over expansions via choosing the right definitional variant for the proof under consideration, still letting the authors to retain the more conservative approach. As a rule, the change will affect new Mizar texts, but obviously, it allows also for solving some context conflicts caused by the original approach in the Mizar repository. The usefulness of our approach is shown by a number of experiments carried out within MML, which is also affected by the change.

2012 ACM Subject Classification Theory of computation → Interactive proof systems

Keywords and phrases Mizar, definitions, proof assistants, mechanization of proof

Supplementary Material Test files can be found at `http://mizar.uwb.edu.pl/project/unfolding/`

1 Errors recovery and reporting

In this section we present possible errors related to the newly implemented construction `unfolding` which can occur when it is used in an inappropriate way. Before we list them, we present a correct form of a proof of the statement saying that the composition of a transitive relation `R` is included in the relation `R`. The proof will use a version of the inclusion specific to relations (compare Listing 1 and Listing 2 from the paper) stored in the Mizar Mathematical Library under the label `RELAT_1:def 3` [1].

```
environ

 vocabularies TARSKI, RELAT_1, RELAT_2;
 notations TARSKI, RELAT_1, RELAT_2;
 constructors TARSKI, RELAT_1, RELAT_2;
 theorems TARSKI, RELAT_1, RELAT_2;
 definitions RELAT_1, TARSKI;

begin

for R being Relation st R is transitive holds R*R c= R
  proof
    let R be Relation;
    assume
A1: R is transitive;
    unfolding RELAT_1:def 3;
    let x,y be object;
    assume [x,y] in R*R;
    then ex z being object st [x,z] in R & [z,y] in R by RELAT_1:def 8;
    hence [x,y] in R by A1,RELAT_2:31;
  end;
```

Possible errors can be divided into two groups: syntactic and semantic. Syntactic errors are listed in Table 1.

Table 1 Possible syntactic errors

Incorrect line	Error no.	Description
`unfolding;`	308	References are missing
`unfolding RELAT_1:3;`	288	A reference to a definitional theorem required
`unfolding RELAT_1:def 3;`	144, 203	Definitional theorem is not imported in the environment directive
`unfolding RELAT_1:def 3000;`	199	Inaccessible definitional theorem – probably too big number

When proofs are syntactically correct, and if a definition which references of `unfolding` refers to is accessible, but it is not related to the proven statement, then the error `67` is reported:

```
for R being Relation st R is transitive holds R*R c= R
proof
  let R be Relation;
  assume
A1: R is transitive;
  unfolding RELAT_1:def 2;
::>                     *67
end;
::> 67: Expanded definition does not match thesis
```

As mentioned in the paper, only atomic formulae can be expanded, so when `unfolding` is used when the current thesis is a complex formula, then the error `61` is reported. For example:

```
for R being Relation st R is transitive holds R*R c= R
proof
  let R be Relation;
::   assume
:: A1: R is transitive;
  unfolding RELAT_1:def 3;
::>                     *61
end;
::> 61: Thesis is not expandable
```

And finally, if structures of the proofs do not agree with expanded notions, then several different errors can be reported. For example:

```
for R being Relation st R is transitive holds R*R c= R
  proof
    let R be Relation;
    assume
A1: R is transitive;
    unfolding RELAT_1:def 3;
    let x,y be Relation;
::>   *56
  end;
::> 56: Disagreement of types
```

The above error occurred, because the type of `x` and `y` differs from the type of the according variable in the definition `RELAT_1:def 3`, see the paper for the listing.

2 Test set

To test the system fragility, as another experiment we shuffled filenames in the `definitions` environment directive and put them in a random order. The experiment was repeated: after each shuffling, all articles were verified and the numbers are in Table 2: the number of errors varies between 516 and 672 in ca. 120 articles. Based on these results, we can observe that the order in which files are listed in the `definitions` environment directive definitely plays an important role. Previously, authors had to quite carefully construct the list of imports of definitions; this new extension allows to indicate which definition should be expanded at given stages of proofs without paying special attention to the order in which definitions were imported.

Table 2 Errors in directive `definitions`

	articles	errors
sorted	30	65
random test 1	112	672
random test 2	116	494
random test 3	120	516

References

1 Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990. URL: `http://fm.mizar.org/1990-1/pdf1-1/relat_1.pdf`.