Sethood Property in Mizar

Artur Korniłowicz

Institute of Informatics
University of Bialystok, Poland
`arturk@math.uwb.edu.pl`

Abstract

The sethood property is introduced into the Mizar system to declare that objects of a given type or possessing given properties represented as adjectives belong to some universes. It is applied to control whether such objects can be used in definitions of sets by so called *Fraenkel operators*. In this paper we propose an extension of the Mizar system including its language and functionality with a possibility of application of those universes not only within Fraenkel operators, but also by other Mizar modules to reason about types of objects with the sethood property declared.

1 Introduction

In theories based on axioms of Zermelo-Fraenkel set theory [Jec02] a way to define new sets is using the axiom schema of restricted comprehension, known also as the axiom schema of separation or axiom schema of specification. The axiom allows to define subsets X of a given set A where all elements of X satisfy given property P. In the Mizar proof assistant [BBG$^+$15, GKN15] it is formulated as:

```
scheme :: XBOOLE_0:sch 1
  Separation { A() -> set, P[object] } :
    ex X being set st for x being object holds x in X iff x in A() & P[x];
```

Another Mizar construct to define sets is so called *Fraenkel operator* of the form:

$$\{\tau(c_1,\ldots,c_k,v_1,\ldots,v_n) \text{ where } v_1 \text{ is } \Theta_1,\ldots,v_n \text{ is } \Theta_n : \Phi(c_1,\ldots,c_k,v_1,\ldots,v_n)\}$$

where $\tau(c_1,\ldots,c_k,v_1,\ldots,v_n)$[1] is a term involving constants $c_1,\ldots,c_k$ and variables $v_1,\ldots,v_n$, each v_i (for $1 \leq i \leq n$) is a variable of the type Θ_i, and $\Phi(c_1,\ldots,c_k,v_1,\ldots,v_n)$ is a formula. To prevent inconsistency of the theory each variable v_i, in the sense of its type, must satisfy special requirements to be elements of some *pre-existing* sets (universes) `A_i`. In older versions of the Mizar system it was required that types of each v_i should be simply `Element of A_i`, where `A_i` is an arbitrary, not necessarily non-empty, set, or a type Θ_i which widens to the `Element of A_i`, for some sets `A_i` (Θ_i is a subtype of `Element of A_i`). In the Mizar version `7.12.01` released in 2011, in which *sethoodness* property was implemented, this requirement was weakened, and now each type Θ_i must have the property **sethood** registered. For non-expandable types such a registration can be done using the following syntax[1]:

In: C. Kaliszyk, E. Brady, J. Davenport, W.M. Farmer, A. Kohlhase, M. Kohlhase, D. Müller, K. Pąk, and C. Sacerdoti Coen (eds.): Joint Proceedings of the FMM and LML Workshops, Doctoral Program and Work in Progress at the Conference on Intelligent Computer Mathematics 2019 co-located with the 12th Conference on Intelligent Computer Mathematics (CICM 2019), Prague, Czech Republic, July 8–12, 2019, published at http://ceur-ws.org

[1] Three dots used in this context are not Mizar symbols.

```
definition
 let $x_1$ be $\theta_1$, $x_2$ be $\theta_2$, ..., $x_n$ be $\theta_n$;
 mode $\mu$ of $x_1, x_2, \ldots, x_n$ -> $\theta$ means :ident:
  $\Phi(x_1, x_2, \ldots, x_n,$ it$)$;
 existence;
 sethood
 proof
  thus ex $X$ being set st for $a$ being $\theta$ st $\Phi(x_1, x_2, \ldots, x_n, a)$ holds $a$ in $X$;
 end;
end;
```

and for expandable types with the following syntax:

```
registration
 let $x_1$ be $\theta_1$, $x_2$ be $\theta_2$, ..., $x_n$ be $\theta_n$;
 sethood of $\alpha_1$ ... $\alpha_m$ $\theta$
 proof
  thus ex $X$ being set st for $a$ being $\alpha_1$ ... $\alpha_m$ $\theta$ holds $a$ in $X$;
 end;
end;
```

This property registers that each term of a type with the property registered is an element of some set. But the sets are hidden within the proofs of the correctness of the property (sets X in the above proofs) and cannot be used in other formulas and reasonings outside the proofs.

In this paper we propose an extension of the Mizar language to make declared universes visible outside the registrations and to use them for various purposes during verification process. We present the syntax of the extension, possible usage and profits of new functionality, its influence on the Mizar Mathematical Library [BBG$^+$18] and briefly discuss possible negative consequences of the proposed solution.

2 Sethood Property

2.1 Syntax

To achieve our goal that is making universes of types visible and accessible in the library, not only locally in particular proofs, we propose a simple extension of the grammar of the Mizar language with a new word **wrt** followed by a proposed universe. In the cases of non-expandable modes the syntax is:

```
definition
 let $x_1$ be $\theta_1$, $x_2$ be $\theta_2$, ..., $x_n$ be $\theta_n$;
 mode $\mu$ of $x_1, x_2, \ldots, x_n$ -> $\theta$ means :ident:
  $\Phi(x_1, x_2, \ldots, x_n,$ it$)$;
 existence;
 sethood wrt $X(x_1, x_2, \ldots, x_n)$
 proof
  thus for $a$ being $\theta$ st $\Phi(x_1, x_2, \ldots, x_n, a)$ holds $a$ in $X(x_1, x_2, \ldots, x_n)$;
 end;
end;
```

and in the case of expandable modes it is:

```
registration
 let $x_1$ be $\theta_1$, $x_2$ be $\theta_2$, ..., $x_n$ be $\theta_n$;
 sethood of $\alpha_1$ ... $\alpha_m$ $\theta$ wrt $X(x_1, x_2, \ldots, x_n)$
 proof
  thus for $a$ being $\alpha_1$ ... $\alpha_m$ $\theta$ holds $a$ in $X(x_1, x_2, \ldots, x_n)$;
 end;
end;
```

Correctness conditions generated by the system for proving the soundness of properties are presented in the above snaps as well. They ensure that all terms of types for which the sethood property is declared belong to the declared universes.

Examples of practical usages of those declarations taken from the Mizar Mathematical Library for both non-expandable and expandable types are [INKK17]:

```
registration
 sethood of Complex wrt COMPLEX
 proof
  thus for c being Complex holds c in COMPLEX;
 end;
end;
```

```
definition
 let V,A be set;
 mode NonatomicND of V,A -> Function means
 ex S being FinSequence st S IsNDRankSeq V,A & it in Union S;
 existence;
 sethood wrt Union FNDSC(V,A)
 proof
   thus for f being Function st ex S being FinSequence st S IsNDRankSeq V,A & f in Union S
        holds f in Union FNDSC(V,A);
 end;
end;
```

It is worth to mention that **wrt** clause is not obligatory, authors can decided whether to make the universe visible or not. If **wrt** is not used for some property, no changes in the processing of the property will be observed.

2.2 Properties

Two main modules of the Mizar verifier are: ANALYZER responsible, among others, for type checking, identification of objects, and resolving disambiguations, and CHECKER responsible for verification of the logical correctness of proof steps.

At the stage of processing Fraenkel operators, ANALYZER checks whether types of all variables bounded within the operators (using clause **where**) have the sethood property registered (if not, appropriate errors are reported). These functionality remains in our new approach of processing sethoodness. Sethoodness remains also inherited by sub-types and is propagated among all adjectives which expand to the adjective with the property declared explicitly, as well.

To see one of the most important profits of the new sethood property, what we claim that the better reusability of existing in the Mizar Mathematical Library theorems is, let us start from an example of using natural numbers in the Mizar system. Basically, there are two ways of using natural numbers in formulas: as variables of the type `Nat`, which is an abbreviation of the type `natural number`, or as variables of the type `Element of NAT`, where `NAT` is defined as the set of all natural numbers. By default in the Mizar system there is no automatic communication[2] between these two types, but using the *conditional registration* mechanism Mizar allows to declare that each `Element of NAT` is a `natural number` as follows:

```
registration
 cluster -> natural for Element of NAT;
 coherence;
end;
```

Having such a registration each `Element of NAT` is also identified as a `Nat`. On the other hand to allow the Mizar verifier to know that each `natural number` is an `Element of NAT` one has to explicitly refer to the appropriate theorem (or definition). Also theorems about `Element of NAT` are not directly accessible for terms of the type `Nat`.

New implementation of the sethood property establishes also the automatic communication from the type `Nat` to the type `Element of NAT`. For example, the following registration:

```
registration
 sethood of Nat wrt NAT;
end;
```

[2] By automatic communication we mean that the Mizar ANALYZER and CHECKER can infer some facts without any explicit references to any definitions, theorems and schemas, but only “playing” with Mizar constructions like *redefinitions*, *registrations*, *term reductions* [Kor13], *term identifications* and *properties* [NB04] that are automatically processed by dedicated Mizar procedures.

makes Checker to understand all `Nat`s also as `Element`s `of NAT`. Then, all theorems about an `Element of NAT` will be applicable for terms of the type `Nat`, what will possibly result with less number of theorems stored in the Mizar Mathematical Library. Of course, a question: *Why theorems involving variables of the type* Element of NAT *instead of* Nat *are kept in the MML?* can arise. Reasons for that are various, e.g.: a) terms of the type `Element of NAT` are sub-terms of operations whose arguments must be elements of some sets, b) terms of the type `Element of NAT` are arguments of adjectives or types whose arguments must be elements of some sets, c) authors of Mizar works sometimes use `Element of NAT` simply "by chance".

To achieve a good classification of types concerning their universes a bigger number of sethood registrations will be required in Mizar articles. Again, the example with numbers nicely illustrates the issue. Let us consider a chain of adjectives representing numbers: `natural` → `integer` → `rational` → `real` → `complex` already defined in the MML. In the old treatment of the sethood property, to know that all these types are included within an universe it was enough to register only one sethoodness for the biggest set, that is for `complex` numbers. The property was inherited by all adjectives which expand to complex numbers. In the new implementation of the sethood property such one registration for complex numbers will result with treatment all `natural`, `integer`, `rational`, `real`, and `complex` numbers as objects of the type `Element of COMPLEX`. To declare more precise universes for each mentioned type the following chain of sethood registrations is needed:

```
registration                  registration                      registration
 sethood of Nat wrt NAT;       sethood of Integer wrt INT;       sethood of Rat wrt RAT;
end;                          end;                              end;

registration                  registration
 sethood of Real wrt REAL;     sethood of Complex wrt COMPLEX;
end;                          end;
```

These registrations allow Checker to treat all discussed kinds of numbers as elements of the most appropriate sets, that is naturals as elements of $\mathbb{N}$, integers as elements of $\mathbb{Z}$, rationals as elements of $\mathbb{Q}$, reals as elements of $\mathbb{R}$, and finally complex numbers as elements of $\mathbb{C}$. But, because the sethoodness is still inherited by sub-types, naturals will be treated not only as elements of $\mathbb{N}$, but also as elements of $\mathbb{Z}$, as elements of $\mathbb{Q}$, as elements of $\mathbb{R}$, and as elements of $\mathbb{C}$. Consequently, integers, rationals, and reals will be treated as elements of universes declared in their sethood registrations, but also as elements of all their super-sets, what in some applications can be seen as undesirable.

3 Conclusions and Future Work

In this paper we report about an extension of the Mizar system with a possibility of using universes of types presented in proofs of the sethood property of chosen types and adjectives to collect more types of terms which basic types have the sethood property registered. It may move work from the world of adjectives to the world of elements of appropriate universes.

This extension induces work for the Library Committee, who is responsible for the management, developing and revisions of the MML, concerning detection of cases where it is worth to introduce more sethoodness registrations in already stored Mizar documents, and in the case of approval, a refactoring of the MML [GS07] will be required while maintaining licensing its content [AKM+11].

References

[AKM+11] Jesse Alama, Michael Kohlhase, Lionel Mamane, Adam Naumowicz, Piotr Rudnicki, and Josef Urban. Licensing the Mizar Mathematical Library. In James H. Davenport, William M. Farmer, Josef Urban, and Florian Rabe, editors, *Proceedings of the 18th Calculemus and 10th International Conference on Intelligent Computer Mathematics*, volume 6824 of *MKM'11, Lecture Notes in Computer Science*, pages 149–163, Berlin, Heidelberg, 2011. Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-22673-1_11

[BBG+15] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, Karol Pąk, and Josef Urban. Mizar: State-of-the-art and beyond. In Manfred Kerber, Jacques Carette, Cezary Kaliszyk, Florian Rabe, and Volker Sorge, editors, *Intelligent Computer Mathematics*, volume 9150 of *Lecture Notes in Computer Science*, pages 261–279. Springer International Publishing, 2015. http://dx.doi.org/10.1007/978-3-319-20615-8_17

[BBG+18] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pąk. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *Journal of Automated Reasoning*, 61(1):9–32, 2018. `https://doi.org/10.1007/s10817-017-9440-6`

[GKN15] Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. Four decades of Mizar. *Journal of Automated Reasoning*, 55(3):191–198, 2015. `http://dx.doi.org/10.1007/s10817-015-9345-1`

[GS07] Adam Grabowski and Christoph Schwarzweller. Revisions as an essential tool to maintain mathematical repositories. In *Proceedings of the 14th Symposium on Towards Mechanized Mathematical Assistants: 6th International Conference*, Calculemus '07 / MKM '07, pages 235–249, Berlin, Heidelberg, 2007. Springer-Verlag. `http://dx.doi.org/10.1007/978-3-540-73086-6_20`

[INKK17] Ievgen Ivanov, Mykola Nikitchenko, Andrii Kryvolap, and Artur Korniłowicz. Simple-named complex-valued nominative data – definition and basic operations. *Formalized Mathematics*, 25(**3**):205–216, 2017. `https://doi.org/10.1515/forma-2017-0020`

[Jec02] Thomas. J. Jech. *Set Theory*. Springer-Verlag, Berlin Heidelberg New York, 2002. `http://dx.doi.org/10.1007/3-540-44761-X`

[Kor13] Artur Korniłowicz. On rewriting rules in Mizar. *Journal of Automated Reasoning*, 50(2):203–210, February 2013. `http://dx.doi.org/10.1007/s10817-012-9261-6`

[NB04] Adam Naumowicz and Czesław Byliński. Improving Mizar texts with properties and requirements. In Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec, editors, *Mathematical Knowledge Management, Third International Conference, MKM 2004 Proceedings*, volume 3119 of *MKM'04, Lecture Notes in Computer Science*, pages 290–301, 2004. `http://dx.doi.org/10.1007/978-3-540-27818-4_21`