Communication Papers of the Federated Conference on
Computer Science and Information Systems pp. 23–26
DOI: 10.15439/2018F106
ISSN 2300-5963 ACSIS, Vol. 17

Mizar Set Comprehension in Isabelle Framework

Karol Pąk
University of Białystok,
Ciołkowskiego 1M, 15-245 Białystok, Poland
Email: pakkarol@uwb.edu.pl

***Abstract*—The Mizar project from its beginning aimed to make a highly human oriented proof environment where the proof style closely reflects the informal proofs style. The support is reflected in the size of the largest consistent formal library — Mizar Mathematical Library (MML). However, the Mizar system is the only tool that provides full verification and further development of the MML. In this paper, we present the progress in the development of the Isabelle/Mizar project whose main goal is independent cross-verification of the MML in Isabelle. We focus on Mizar set comprehension operators that allow defining sets that satisfy a given predicate. The development already covers simple cases where the arity of predicates is limited to two. We propose an infrastructure that provides a more elegant and recursive approach to construct and to provide the main property of set comprehension operators.**

I. Introduction

Mizar Mathematical Library (MML) [1] is one of the most recognizable features of the Mizar system. Developed for almost three decades the library contains today more than 1300 articles, 60000 proved theorems and covers many areas of today's mathematics from algebra, analysis, topology including topological manifolds [2] and lattice theory [3] that have not been formalized elsewhere. Therefore, it is not a surprise that there exists a number of external tools that explore the content of the MML to ensure human-readable access, starting with automatically generated articles in the Journal of Formalized Mathematics, searching tools as MML Query [4], variants of XML format [5] and MMT logical framework [6].

On the other hand, the MML is often used as an extensive theorems database, for instance, in the process of comparing the performance of leading systems of Automatic Theorem Proving (ATP) as well as a training data in machine learning, especially for developing and testing premise selection methods [7]. However, the Mizar logic is a serious problem for today's efficient first-order ATP systems. It is important to note that the Mizar is essentially a first-order system that is based on the set theory, but the Mizar logic goes a little bit beyond first-order in two cases:

- the Mizar schemes that are second-order theorems parameterized by the predicates and functions,
- the Mizar set comprehensions (referred to as *Fraenkel* in the Mizar literature [8]) that allow defining sets of terms whose arguments have given types and satisfy a given predicate.

Therefore, to translate and further to cross-verify the content of the MML we have to choose between first and higher order logic. Obviously, first-order logic is welcome from the ATP point of view, but currently existing translations omit each problem where second-order constructions occur or they need to be expressed in first-order logic with the support of a potentially infinite number of axioms [9]. On the other hand, second-order Mizar problems have been cross-verified by C. Brown [10] using higher-order automated theorem provers Satallax and LEO-II with the support of only a few additional axioms.

Isabelle/Mizar is a project whose main goal is an automatic translation of the Mizar proof scripts from the MML to the Isabelle framework, enabling cross-verification of the obtained scripts, but in contrast to the existing translations it tries to preserve types, commands and the structure of proofs originally used [11], [12]. The project is also a unique from the point of view of the order of logic. Namely, our object logic created in Isabelle that expresses that the foundations of the Mizar logic can be both an extension of first-order and higher-order logic, that is, a user can switch between the dependency on relatively poor Isabelle/FOL and the most developed Isabelle object logic Isabelle/HOL [13].

In this paper, we discuss the progress in the Isabelle/Mizar project in relation to the development of set comprehensions. In our previous work [14] we proposed an equivalent of these sets that can be defined as a meta-functor independently for every arity of relevant predicates. Unfortunately, proofs of such n-arity functor correctness require a lot of effort especially in the case of predicates with many arguments.We will, therefore, propose an infrastructure for a more elegant *recursive* proof of correctness that is able to apply the proven property of n-ary meta-functor to justify corresponding property of $(n+1)$-ary one. We investigate the efficiency of our procedure up to the maximum arity of the set comprehension used in the MML. Currently, the maximum required n is 6.

In Section II we discuss existing methods that try to express more advanced Mizar concepts in first-order and higher-order systems. We mainly focus on solutions used to express the Mizar set comprehension operators and the number of additional axioms introduced for this purpose. After a short introduction of the axiomatization used in our Isabelle/Mizar project in Section III, we describe our concept of the Mizar set comprehension in Section IV. The particular contributions of this paper are:

- We propose a concept of the product of Mizar types that

The paper has been supported by the resources of the Polish National Science Center granted by decision n° DEC-2015/19/D/ST6/01473.

is expressed in our semantics that is slightly more liberal than the Mizar one. We use the concept in a new approach to define Mizar set comprehension in a clear and elegant way.

- We investigate the possibilities of our approach to prove recursively the main property of the Mizar set comprehension operators, , i.e. , every set comprehension determined by given functor, universe and predicate can be replaced by a new constant whose members are exactly the values of the function at each element of the universe that satisfies the predicate.

II. Solutions in Existing Mizar Translations

A lot of work has been done to explore the MML by external tools that struggle with many Mizar problems. J. Urban [15] created the largest and the most comprehensive export of MML, initially to the TPTP untyped first-order language where each higher-order problems related to the set comprehension and schemes have been omitted. To cover omitted cases he uses the standard set-theoretic elimination procedure and introduces a dedicated extension of the TPTP language to make the entire MML available for first-order ATPs as a part of the Mizar Problems for Theorem Proving (MPTP) project [9]. Theoretically, all second-order problems could be completely removed from the representation of the MML using the following two rules:

- every reference to a given scheme can be redirected to a copy of the scheme where the occurring second-order variables have been instantiated by the corresponding predicates and functions determined in the context of the reference,
- every set comprehension can be replaced by a new constant with an appropriate property that is guaranteed by the Replacement axiom of Tarski-Grothendieck set theory.

Obviously, the first solution generates a very large expansion, since schemes in most cases refer to other schemes in their justification. Additionally, the Replacement axiom that is originally formulated as a scheme in the MML

```
scheme ::TARSKI_0:sch 1
 Replacement {A() -> set,P[object,object]}:
  ex X being set st for x being object holds
      x in X iff ex y being object st y in A() & P[y,x]
 provided
  for x,y,z being object st P[x,y] & P[x,z]
     holds y = z;
```

has to be replaced by a potentially infinite number of instances of the axiom. These are necessary to decode the information. The expression `A() -> set` declares a “second-order” 0-arity functor that, in this case, trivializes to a constant and can be instantiated by a term of the type `set`; and the expression `P[object,object]` that declares a “second-order” 2-arity predicate that semantically can be instantiated by a formula with two free variables of the type `object`. The second rule also generates a potentially infinite number of axioms, since the property of the new constant that replaces a given set comprehension can be introduced as an axiom or proven using the Replacement axiom.

A different approach to solve second-order Mizar problems has been proposed by Kunčar [16] who tried to express the content of the MML in the type system of HOL Light. Obviously, the set comprehension operators and schemes can be naturally expressed in higher-order logic. However, the approach proposed by Kunčar was not able to cover more advanced features of the Mizar type system and finally was only sufficient to translate the first few simpler theories. A successful attempt to cover second-order Mizar problems has been done by C. Brown and J. Urban [10] where second-order Mizar problems have been cross-verified using higher-order automated theorem provers Satallax and LEO-II. However, even in this case the set comprehension operators have been axiomatized instead of defined, using a family of constants $\mathsf{replSep}_n$ that correspond to the n-arity set comprehension operators.

III. Mizar Foundations in Isabelle

In our previous work [14], we defined a unique equivalent of the Mizar foundations as an object logic in the Isabelle logical framework that includes several definitional mechanisms, the Mizar dependent type system including the structure types as well as the second-order concepts. This equivalent is a result of many experiments whose main goal was to simultaneously express each Mizar components and minimize the number of additional axioms and constants.

The current version of our semantic model of Mizar based on the following Isabelle meta-level types and meta-level constants:

```
typedecl Set
typedecl Ty
consts
  ty_membership :: Set ⇒ Ty ⇒ o               (infix be 90)
  define_ty :: Ty ⇒ (Set ⇒ o) ⇒ (Set ⇒ o) ⇒Ty
  choice :: Ty ⇒ Set                                (the _)
```

where Set corresponds to Mizar terms, Ty corresponds to Mizar types, ty_membership specifies the relation between terms and types, define_ty allows to define types, and choice is the choice operator. Note that Mizar distinguishes syntactically types for two kinds: modes that require the existence and adjectives that can restrict modes. We have provided this division in our logical framework before [17], but we have combined these types to simplify our model. To preserve the Mizar semantics we define a meta-predicate

$$\mathsf{inhabited}(\mathsf{D}) \longleftrightarrow (\exists_M \mathsf{x}.\ \mathsf{x}\ \mathsf{be}\ \mathsf{D})$$

and assume it defining the bounded quantifiers

$$\mathsf{inhabited}(\mathsf{D}) \Longrightarrow \\ \quad \mathsf{Ball}(\mathsf{D},\ \mathsf{P}) \longleftrightarrow (\forall_M \mathsf{x}.\ \mathsf{x}\ \mathsf{be}\ \mathsf{D} \longrightarrow \mathsf{P}(\mathsf{x}))$$

$$\mathsf{inhabited}(\mathsf{D}) \Longrightarrow \\ \quad \mathsf{Bex}(\mathsf{D},\ \mathsf{P}) \longleftrightarrow (\exists_M \mathsf{x}.\ \mathsf{x}\ \mathsf{be}\ \mathsf{D} \land \mathsf{P}(\mathsf{x}))$$

where $\forall_M$, $\exists_M$ correspond to the standard universal and existential quantifiers of the logic (either Isabelle/FOL or Isabelle/HOL), respectively.

Then to specify all necessary dependencies between terms and types as well as the standard axiom of choice we introduce *only* two axioms that extend the MML axioms, that is, are defined in three axiomatic Mizar articles and are HIDDEN, TARSKI_0, and TARSKI_A, are sufficient to introduce a full semantic model of Mizar. It is important to note that keeping such a small number of axioms is one of the main goals of our project.

```
axiomatization where
  def_ty_property: T ≡ define_ty(parent, cond, property) ⟹
    (x be T ⟶ x be parent ∧ (cond(x) ⟶ property(x))) ∧
    (x be parent ∧ cond(x) ∧ property(x) ⟶ x be T) ∧
    (x be parent ∧ ¬cond(x) ⟶ inhabited(T))  and
  choice_ax: inhabited(M) ⟹ (the M) be M
```

Note that the def_ty_property axiom seems to be unnecessarily complicated and could be replaced by a stronger formula T ≡ define_ty(property) ⟹x be T ⟷ property(x). However, our experience has shown that our formulation is weaker but sufficient to define all the necessary concepts. For example, we use the def_ty_property axiom to define the negation of type, the intersection of types but also in the case of more advanced concepts, for instance, the conditional functor definitions where meaning (prop) of defined functor (df) is formulated under some assumption (as).

```
definition NON (non _ )
  where non A ≡ define_ty(object, λ_. True,λ x . ¬ x is A)

definition ty_intersection (infixl | 100) where
  t1 | t2 ≡ define_ty(object,λ_.True, λx. x be t1 ∧ x be t2)

abbreviation func_assume_means_prefix
  (assume _ func _ → _ means _ [0,0,0,0] 10)
where assume as func df → ty means prop ≡
 df = the define_ty(ty, λ_. as, prop)
```

It is also important to note that in our approach we use the MML axioms or even the first few re-formalized articles of the MML to define as well as to provide properties of selected concepts, for instance, we use the root of the Mizar type (object) in the above definitions.

IV. Mizar Set Comprehensions in Isabelle

As it has been shown in Section II the Mizar set comprehension is one of the two second-order Mizar concepts that require a lot of effort in any attempt to cross-verify the MML.

Generally, it allows to use a set of terms $\mathsf{F}(v_1,\ldots,v_n)$ whose arguments have given types (v_i be Θ_i for $i = 1,2,\ldots,n$) and satisfies the formula $\mathsf{P}[v_1,\ldots,v_n]$. Note that the Mizar semantic does not allow to define this operator directly in a Mizar script (for more detail see [18]). Therefore, the operator is built-in and is automatically expanded in terms of set membership as follows:

$$x\ \mathsf{in}\ \{\mathsf{F}(v_1,\ldots,v_n)\ \mathsf{where}\ v_1\ \mathsf{is}\ \Theta_1,\ldots,v_n\ \mathsf{is}\ \Theta_n{:}\mathsf{P}[v_1,\ldots,v_n]\}$$
$$\mathsf{iff}$$
$$\mathsf{ex}\ v_1\ \mathsf{be}\ \Theta_1,\ldots,v_n\ \mathsf{be}\ \Theta_n \mathsf{st}\ x=\mathsf{F}(v_1,\ldots,v_n)\ \&\ \mathsf{P}[v_1,\ldots,v_n]$$

Obviously such a set is guaranteed to exist by the Replacement axiom but only if every type Θ_i has sethood property to avoid Russell's paradox.

```
definition sethood_prop where
  sethood_prop(M) ≡ ∃X:set. ∀x: M. x in X
```

For example, if a type Θ has sethood property, then the existence of the set $\{\mathsf{F}(v)\ \mathsf{where}\ v\,\mathsf{is}\,\Theta:\mathsf{P}[v]\}$ is a direct consequence of the Replacement axiom substituted by the set of all objects of the type Θ and the predicate λx y. x = F(y) & P[y]. However, the construction of the suitable set is generally a laborious process, since we need to construct the Cartesian product of sets that cover particular types directly from axioms. By using our re-formalization of the MML in the Isabelle/Mizar system we can reduce the size of such a justification using directly the Cartesian product defined originally in the Mizar script ZFMISC_1 but the justification is still quite tedious.

A. Recursive Justification of Freankel Obligations

A naive approach to constructing $(n+1)$-ary set comprehension operators using n-ary one fails in the original Mizar semantics since we cannot define there the product types. However, our semantics is slightly more liberal than that of Mizar and it can be done using the def_ty_property axiom as follows

```
definition ProdType_prefix (_ × _)
  where A × B ≡
define_ty(object,λ_. True, λx. x be pair ∧ x‘1 be A ∧ x‘2 be B)
```

where the pair type corresponds to the Mizar attribute pair and x‘1, x‘2 correspond to the left and right projection of a given term x that can be represented as a pair. Note that the attribute and projections are originally defined in the Mizar article XTUPLE_0. We give as an example of our re-formulation definitions of the pair and the left projection.

```
mdef xtuple_0_def_1 (pair) where
 attr pair for object means
   (λX. ex x1,x2 be object st X=[x1,x2])

mdef xtuple_0_def_2 (_ ‘1) where
  mlet x be object
  assume x is pair func x ‘1 → object means
    (λit. for y1,y2 be object st x=[y1,y2] holds it = y1)
```

Then, based on the selected re-formalized theorems including properties of the Cartesian product we provide that the product of inhabited types is inhabited as well as that the product of types that have sethood property also has the property

```
lemma PT_inhabited:
  assumes inhabited(A) inhabited(B)
  shows inhabited(A×B)

lemma PT_sethood:
  assumes inhabited(A) inhabited(B)
          sethood_prop(A) sethood_prop(B)
  shows sethood_prop(A×B)
```

Next, we prove the PT_rule lemma that specifies a dependence between a formula where an existential quantifier binds a variable of a product type $\Theta_1 \times \Theta_2$ and a corresponding formula where two quantifiers have been used to bind separately variables of types Θ_1, Θ_2

```
lemma PT_rule:
  assumes inhabited(T1) inhabited(T2)
  shows (∃x:T1×T2. uncurry(P)(x)) ⟷
        (∃x1:T1. ∃x2:T2. P(x1,x2))
```

where the uncurry operator is defined as follows:

```
abbreviation
  uncurry(P) ≡ λx. P(x`1,x`2)
```

The PT_rule lemma can now be practically used to provide a basic property of 2-arity set comprehension operator based on the corresponding property of 1-arity ones.

```
theorem Fraenkel2E:
  assumes inhabited(T1) inhabited(T2)
          sethood_prop(T1) sethood_prop(T2)
  shows x in Fraenkel1(uncurry(F),T1×T2,uncurry(P))
       ⟷ (∃y1:T1. ∃y2:T2. x = F(y1,y2) ∧ P(y1,y2))
  by (rule Ifft[OF _ Fraenkel1],  rule Ifft[OF _ PT_rule],
      auto simp add: assms PT_sethood PT_inhabited)
```

It is important to note the justification is a single Isabelle tactic where we use all lemmas formulated above. Additionally, modifying only the reference to the theorem we can easily cover all cases up to the arity equals 6 (for more detail see our formalization). Then it is easy to see that every set comprehension operator can be defined as abbreviations for an appropriately substituted 1-arity operator

$$\mathsf{Fraenkel}_n(\mathsf{F},\Theta_1,\Theta_2,\ldots,\Theta_n,\mathsf{Q}) \equiv \mathsf{Fraenkel1}(\,\underbrace{\mathsf{uncurry}(\ldots(\mathsf{uncurry}(}_{n\text{–}1\text{-times}}\mathsf{F}))\ldots),\ \Theta_1\times\Theta_2\times\ldots\times\Theta_n,\ \underbrace{\mathsf{uncurry}(\ldots(\mathsf{uncurry}(}_{n\text{–}1\text{-times}}\mathsf{Q}))\ldots)\,)$$

V. Conclusion

We have presented the progress in our project aiming to cross-verification the MML in Isabelle. In relation to our previous work [14] the proposed recursive approach is an important step forward in defining more clearly Mizar set comprehension operators. We have improved our solution in two aspects. Namely, we indicate how to define more advanced cases based only on the simplest case, i.e., the 1-arity set comprehension operator and we reduce the proof of the main property of Mizar set comprehension in more advanced cases to a single Isabelle tactic. The detail of our formalization is available at:

http://alioth.uwb.edu.pl/~pakkarol/fedcsis2018/

References

[1] G. Bancerek, C. Byliński, A. Grabowski, A. Korniłowicz, R. Matuszewski, A. Naumowicz, and K. Pąk, "The role of the Mizar Mathematical Library for interactive proof development in Mizar," *Journal of Automated Reasoning*, 2017. doi: 10.1007/s10817-017-9440-6. [Online]. Available: https://doi.org/10.1007/s10817-017-9440-6

[2] K. Pąk, "Topological manifolds," *Formalized Mathematics*, vol. 22, no. **2**, pp. 179–186, 2014. doi: 10.2478/forma-2014-0019

[3] G. Bancerek and P. Rudnicki, "A Compendium of Continuous Lattices in MIZAR," *J. Autom. Reasoning*, vol. 29, no. 3–4, pp. 189–224, 2002.

[4] ——, "Information retrieval in MML," in *Mathematical Knowledge Management, MKM 2003*, ser. LNCS, A. Asperti, B. Buchberger, and J. H. Davenport, Eds., vol. 2594. Springer, 2003. doi: 10.1007/3-540-36469-2_10. ISBN 3-540-00568-4 pp. 119–132. [Online]. Available: https://doi.org/10.1007/3-540-36469-2_10

[5] J. Urban, "XML-izing Mizar: Making semantic processing and presentation of MML easy," in *Mathematical Knowledge Management (MKM 2005)*, ser. LNCS, M. Kohlhase, Ed., vol. 3863. Springer, 2005. ISBN 3-540-31430-X pp. 346–360.

[6] M. Iancu, M. Kohlhase, F. Rabe, and J. Urban, "The Mizar Mathematical Library in OMDoc: Translation and applications," *J. Autom. Reasoning*, vol. 50, no. 2, pp. 191–202, 2013. doi: 10.1007/s10817-012-9271-4

[7] C. Kaliszyk and J. Urban, "MizAR 40 for Mizar 40," *J. Autom. Reasoning*, vol. 55, no. 3, pp. 245–256, 2015. doi: 10.1007/s10817-015-9330-8

[8] A. Grabowski, A. Korniłowicz, and A. Naumowicz, "Four decades of Mizar," *Journal of Automated Reasoning*, vol. 55, no. 3, pp. 191–198, 2015. doi: 10.1007/s10817-015-9345-1

[9] J. Urban and G. Sutcliffe, "ATP-based cross-verification of Mizar proofs: Method, systems, and first experiments," *Math. in Computer Science*, vol. 2, no. 2, pp. 231–251, 2008. doi: 10.1007/s11786-008-0053-7

[10] C. E. Brown and J. Urban, "Extracting higher-order goals from the Mizar Mathematical Library," in *Intelligent Computer Mathematics (CICM 2016)*, ser. LNCS, M. Kohlhase, M. Johansson, B. R. Miller, L. de Moura, and F. W. Tompa, Eds., vol. 9791. Springer, 2016. doi: 10.1007/978-3-319-42547-4_8 pp. 99–114.

[11] C. Kaliszyk, K. Pąk, and J. Urban, "Towards a Mizar environment for Isabelle: Foundations and language," in *Proc. 5th Conference on Certified Programs and Proofs (CPP 2016)*, J. Avigad and A. Chlipala, Eds. ACM, 2016. doi: 10.1145/2854065.2854070 pp. 58–65.

[12] C. Kaliszyk and K. Pąk, "Progress in the independent certification of Mizar Mathematical Library in Isabelle," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2017. doi: 10.15439/2017F289 pp. 227–236.

[13] L. C. Paulson and J. C. Blanchette, "Three years of experience with Sledgehammer, a Practical Link Between Automatic and Interactive Theorem Provers," in *Workshop on the Implementation of Logics, IWIL 2010*, ser. EPiC Series, G. Sutcliffe, S. Schulz, and E. Ternovska, Eds., vol. 2. EasyChair, 2010, pp. 1–11.

[14] C. Kaliszyk and K. Pąk, "Isabelle formalization of set theoretic structures and set comprehensions," in *Mathematical Aspects of Computer and Information Sciences, MACIS 2017*, ser. LNCS, J. Blamer, T. Kutsia, and D. Simos, Eds., vol. 10693. Springer, 2017, pp. 163–178.

[15] J. Urban, "MPTP - motivation, implementation, first experiments," *J. Autom. Reasoning*, vol. 33, no. 3-4, pp. 319–339, 2004. doi: 10.1007/s10817-004-6245-1. [Online]. Available: https://doi.org/10.1007/s10817-004-6245-1

[16] O. Kunčar, "Reconstruction of the Mizar type system in the HOL Light system," in *WDS Proceedings of Contributed Papers: Part I – Mathematics and Computer Sciences*, J. Pavlu and J. Safrankova, Eds. Matfyzpress, 2010, pp. 7–12.

[17] C. Kaliszyk and K. Pąk, "Presentation and manipulation of Mizar properties in an Isabelle object logic," in *Intelligent Computer Mathematics - CICM 2017*, ser. LNCS, H. Geuvers, M. England, O. Hasan, F. Rabe, and O. Teschke, Eds., vol. 10383. Springer, 2017. doi: 10.1007/978-3-319-62075-6_14 pp. 193–207.

[18] A. Grabowski, A. Korniłowicz, and A. Naumowicz, "Mizar in a nutshell," *J. Formalized Reasoning*, vol. 3, no. 2, pp. 153–245, 2010. doi: 10.6092/issn.1972-5787/1980