Noname manuscript No.
(will be inserted by the editor)

Combining Higher-order Logic with Set Theory Formalizations

Cezary Kaliszyk · Karol Pąk

the date of receipt and acceptance should be inserted later

Abstract The Isabelle Higher-order Tarski-Grothendieck object logic includes in its foundations both higher-order logic and set theory, which allows importing the libraries of Isabelle/HOL and Isabelle/Mizar. The two libraries, however, define all the basic concepts independently, which means that the results in the two are disconnected. In this paper, we align significant parts of these two libraries, by defining isomorphisms between their concepts, including the real numbers and algebraic structures. The isomorphisms allow us to transport theorems between the foundations and use the results from the libraries simultaneously.

1 Introduction

Among the various foundations for formal proofs, set theory on top of higher-order logic has been tried a number of times in systems such as HOLZF [Obu06], ProofPeer [OFSA14], Egal [Bro14], and Isabelle/Mizar [KP18]. This foundation is attractive for formalization, as it offers a natural mathematical foundation combined with the automation present in HOL.

The formal proof libraries of Isabelle/HOL [WPN08] and that of Mizar [GKN15, BBG$^+$17] are among the largest proof libraries in existence today. Indeed, the HOL library together with the Archive of Formal Proofs consist of more than 100,000 theorems [BHMN15], while the Mizar Mathematical Library (MML) contains 59,000 theorems. Furthermore, the results contained in the libraries are incomparable: Almost all of the Mizar library concerns itself with mathematics, while the majority of the Isabelle/AFP library are results closer to computer science [BHMN15]. For example, the Mizar library includes results about lattice theory [BR02], topology, and mani-

Cezary Kaliszyk
Department of Computer Science, University of Innsbruck, Innsbruck, Austria and
INDRC, International Neurodegenerative Disorders Research Center, Prague, Czech Republic
cezary.kaliszyk@uibk.ac.at

Karol Pąk
Institute of Computer Science, University of Białystok, Białystok, Poland
pakkarol@uwb.edu.pl

folds [Pąk14b] not present in the Isabelle library, while the Isabelle library has many results related to algorithms not in the MML [EHN20, Lam19, LSBM19].

In our previous work [BKP19], we have presented a model of higher-order Tarski-Grothendieck, which justifies the use of higher-order logic formalizations with set theory-based ones simultaneously. This model will allow us to combine the results present in these two major Isabelle libraries. We will specify isomorphisms between various basic types present in the libraries, such as functions and lists, leading to isomorphisms between various number structures including the real numbers, and algebraic structures. The last requires mappings between extensible soft record types and Isabelle type classes [HW07].

We will use the isomorphisms to transport proved theorem including the theorems of Lagrange, Bertrand, cases of Fermat's last theorem and the Intermediate Value Theorem. We will also merge the formalizations of groups and rings in the two libraries.

This paper is an extended version of our paper presented at ITP 2019 [BKP19]. In particular the new content presented is as follows:

- we specify the alignments between many more complex types in the two proof libraries including the rationals and the real numbers;
- we transfer more advanced theorems between the two foundations, including the intermediate value theorem in the merged HOL-Set theory library, together with a large set of theorems that connect Dedekind cuts with Cauchy sequences; and
- we complete the model of higher-order Tarski-Grothendieck presented in our previous work [BKP19], by justifying that the Grothendieck-style axioms are equivalent to the Tarski style (for example used in the Mizar Mathematical Library), formalizing the relationship between them in Isabelle.

The rest of the paper is structured as follows. In Section 2, we introduce the Isabelle HOTG foundations, which will be the basis for all the work, we describe the various axiomatizations of higher-order Tarski-Grothendieck (HOTG) and prove some of them to be equivalent. The basics of the aligned libraries are presented in Section 3. The subsequent Sections 4, 5, 6 discuss our isomorphisms between the different types concerning functions, numbers, and algebra respectively. Section 7 shows practical examples of theorems we can move using the isomorphisms. Section 8 discusses the Tarski-Grothendieck equivalence proofs. Finally, Section 9 discusses the related work on combining foundations and Section 10 presents the existing automated transfer methods in higher-order logic and discusses the limitations of the current work in this respect.

2 Isabelle and Isabelle/Mizar

The Isabelle logical framework's meta-logic *Pure* is a variant of simple type theory with shallow polymorphism. The framework provides functionality that makes it convenient to define object logics, namely allowing easily defining their types, objects, and inference rules as well as their notations. Isabelle/HOL is today the most developed Isabelle object logic. Further Isabelle object logics [Pau90] include constructive type theory or untyped set theory [Pau93].

As Isabelle/HOL is relatively well known and documented, we assume that the reader is familiar with the HOL foundations, Isabelle's basic commands (such as *definition* and *theorem*) and the basic Isabelle objects (numbers and lists). For details, we refer the reader to the Isabelle Manual [Wen21].

The details of Isabelle/Mizar's design and implementation have been presented previously [KP18], therefore, we present only the main commands needed for understanding the current paper. Isabelle/Mizar can be loaded on top of Isabelle/FOL or Isabelle/HOL. It re-uses the type of propositions of the underlying basic logic (o of FOL or bool of HOL) and its basic propositional connectives (negation, conjunction, disjunction, implication), as well as the polymorphic equality present there. However, as the intention of Isabelle/Mizar is to provide a sofly-typed set theory, the universal and existential quantifiers are actually bounded quantifiers that for each quantified object require the type over which it ranges (e.g. $\forall x$ *being Nat.* ...). These propositional and predicate quantifiers together with quality are sufficient for representing firest-order logic with quality and to represent Jaśkowski [Jaś34] style natural deduction proofs present in Mizar.

To introduce the soft type system, a meta logic type of soft-types ty is declared together with the an infix operator **is** that corresponds to the element satisfying the predicate associated with a type. Types can be combined with an intersection operator (e.g. x **is** even | number) and can be negated (e.g. y **is** non-negative) with natural semantics to these operations. The meta-logic abstractions can be used to parametrize the types by other types or even by terms (e.g. A **is** m,n-matrix corresponds to m-by-n matrices). To improve automation, the user can prove properties of types, including *inhabited* and *sethood*. The first one is useful for eliminating quantifiers, whereas the latter is useful for forming compregension operators. Finally, a choice operator (denoted **the** on the level of types allows for getting a term of a given type). For example, given the type of sets, that is intersected with empty, it is possible to define the empty set as **the** empty | set.

The Isabelle/Mizar object logic subsequently introduces the axioms of set theory, specifically, the Tarski-Grothendieck axioms. In particular, the Fraenkel axiom is sufficient to construct set comprehensions written as {F(x) **where** x **be** Element-**of** X: P(x)} (called *Fraenkel terms*) for a given set X, function F and predicate P. In the Mizar language, it is not always possible to define such a functor for arbitrary X, F, P, to avoid inconsistency (variants of Russell's paradox), however, with the help of sethood safe comprehension terms can be interpreted. In Isabelle/Mizar the semantics of comprehension are defined with sethood as a precondition, which means that the property is only valid for terms for which sethood has been proved. This completes the axiomatic part of the object logic, and subsequent parts are introduced as definitional extensions. In particular, the possibility for users to define all kinds types and objects, as well as syntax that allows an easier interaction with softly-typed set theory will be added in this way.

Isabelle/Mizar allows four kinds of user-level definitions corresponding to the same four kinds of user-level definitions in Mizar [GKN15]. Defining predicates is not different from the usual Isabelle definitions. We present the definition of a set theoretic functor by the example of the set theoretic union of two sets[1]:

mdef *xboole_0_def_3* (**infixl** $\cup$ *65*) **where**
  mlet *X be set, Y be set*
  func $X \cup Y \to$ *set means* $\lambda it.$
    $\forall x.$ *x in it* $\longleftrightarrow$ *x in X* $\vee$ *x in Y*

[1] The Isabelle definitions and lemmas that directly correspond to the definitions and lemmas in the Mizar Mathematical Library have been names with the same identifiers in order to ease comparison. For example the Isabelle/Mizar definition *xboole-0-def-3* directly corresponds to the MML definition *XBOOLE_ 0:def_ 3* (colon is not allowed in Isabelle labels).

The **mdef** command starts with the handle used to refer to the definition, followed by an optional notation (union denoted by infix $\cup$), a typing environment in which the definition is made (**mlet**) and then the actual defined operator is given after the keyword *func*. The return type is given after the keyword $\rightarrow$. A definition by *means* is supposed to correspond to a concept where the *it* has the desired property. The user needs to show the existence and the uniqueness as proof obligations. When the user completes these proofs, the Isabelle/Mizar definition package introduces the identifier together with the theorems corresponding to the property of the object and its type for further use. Functors can also be defined by *equals* where the term is given directly in a given environment and with a given return type of the defined term. There, the obligation is to show that the result has the return type.

Type definitions are similar. In order to make type inference and checking automatable, types are divided into modes (more primitive types that are known to be inhabited) and attributes (the types that are used to restrict other types with intersection). Consider for example the definition of the type of a finite sequences over the type D (which are the set-theoretic equivalents of polymorphic lists used are often used in formal proofs):

mdef *finseq_1_def_4* (*FinSequence-of* _) **where**
mlet *D be object*
mode FinSequence-of D $\rightarrow$ *FinSequence means*
(λ*it*. *rng it* $\subseteq$ *D*)

Again **mlet** introduces an environment (these are preconditions for the definitional theorems but can be used in the proofs) and the definition can describe the desired properties that all objects of the defined type must have. After the proof obligation (non-emptiness) is proved, definitional theorems are derived and given to the user. The already mentioned attributes are also similar. They restrict a given type to a subtype. An example type introduced with the help of an attribute is the type of relations. First, the attribute *Relation_like* is introduced, which can be later used to define the type of relations as just an abbreviation, as follows.

mdef *relat_1_def_1* (*Relation_like*) **where**
attr Relation_like for set means
(λ*it*. $\forall$ *x*. *x in it* $\longrightarrow$ ($\exists$ *y, z*. *x* $=_{\mathcal{S}}$ [*y, z*]))..

This approach allows for all definitions and operations defined for a *Relation* to also immediately be available for a *Function*, which is defined as a type restriction using the attribute *Function_like*. The type *FinSequence* is similarly defined by the attribute *FinSequence_like* as follows:

mdef *funct_1_def_1* (*Function_like*) **where**
attr Function_like for set means
(λ*it*. $\forall$ *x,y1,y2 being object*.
[*x,y1*] *in it* $\wedge$ [*x,y2*] *in it* $\longrightarrow$ *y1* $=_{\mathcal{S}}$ *y2*)..

mdef *finseq_1_def_2* (*FinSequence-like*) **where**
attr FinSequence-like for Relation means
(λ*it*. $\exists$ *n be Element-of NAT*. *dom it* $=_{\mathcal{S}}$ *Seg n*)..

abbreviation *Relation* $\equiv$ *Relation_like* | *set*

abbreviation *Function* $\equiv$ *Function_like* | *Relation*

abbreviation *FinSequence* $\equiv$ *FinSequence-like* | *Function*

Finally, Isabelle/Mizar introduces the **mtheorem** command, that is similar to the standard theorem command, but additionally allows the introduction of soft-type assumptions with the **mlet** keyword and hiding these from the user as long as the automated type inference can handle these. Additionally to imitate the Mizar automation the **mby** proof method has been included, that combines type inference with Isabelle's `auto` proof method.

Parallel to the system development, the Mizar community puts a significant effort into building the Mizar Mathematical Library (MML) [BBG+17]. Parts of the MML library (including numbers or parts of algebra) have been translated to Isabelle/Mizar [KP19a] and are being used in the current paper.

3 Proof Integration

The Isabelle higher-order Tarski-Grothendieck foundations allow the import of results proved in higher-order logic and in set theory. This is possible both theoretically (we have previously presented a model that supports the combined foundation [BKP19] and discussed its adequacy more in Section 8) and practically, that is the Isabelle logical framework allows us to import various results from the two libraries of Isabelle/HOL and Isabelle/Mizar in the same environment. Note, however, that the imported developments are initially disconnected. In this and the next sections, we will define transfer methods between these results. These will allow us to use theorems proved in one of the foundations using the term language of the other.

All the definitions and theorems presented in these sections have been formalized in Isabelle and will be presented close to the Isabelle notation. The Isabelle environment will import both Isabelle/HOL [NPW02] and Isabelle/Mizar [KP18] object logics along with a number of results formalized in the standard libraries of the two. Isabelle distinguishes between meta-level implication ($\Longrightarrow$) and object-level implication ($\longrightarrow$) and our notation in examples below reflects this distinction. The remaining notations will follow first-order conventions. In particular, the symbols $=_{\mathcal{H}}$ and $=_{\mathcal{S}}$ will refer to the HOL and set-theoretic equality operations respectively. Then, *be* is the Mizar infix operator for specifying the type of a set in the Mizar intersection type system [KPU16].

In order to transfer results between the foundations, we will first define bijections between types that are isomorphic. We will next show that these bijections preserve various constants and operators. This will allow us to transfer results using higher-order rewriting, in the style of quotient packages for HOL [Hom05, KU11] and the Isabelle transfer package [HK13]. Note, that we are not able to use these packages directly. We discuss this in section 10.

In the Mizar set theory there are often two ways to express domains of objects. It is already the case for the natural numbers, where it is common to reason both about the type of the natural numbers and the members of the set of natural numbers. This is necessary since the arguments of all operations must be sets, while the reasoning engine allows more advanced reasoning steps for types [BBG+17]. We, therefore, define two operators, one that specifies a bijection between a HOL type and a set-theoretic set and one that specified a bijection between a HOL type and a set-theoretic type. The definitions are analogous and we show only the former one here. We will define an isomorphism between a type σ and a set $d \in \Lambda_\iota$ to be a pair (f, g) of functions (at the type theory level) where f maps sets to objects of type σ and g maps objects of type

σ to sets in such a way that objects of type σ (in the type theory) correspond uniquely to elements of d (in the set theory).

Definition 3.1 Let σ be a type, $d \in \Lambda_\iota$ be a set and $\mathrm{s2h} \in \Lambda_{\iota \Rightarrow \sigma}$ and $\mathrm{h2s} \in \Lambda_{\sigma \Rightarrow \iota}$ be functions. The predicate $\mathrm{beIso}_S\langle \mathrm{h2s}, \mathrm{s2h}, d\rangle$ holds whenever all of the following hold:

- $\forall x : \sigma.\mathrm{s2h}(\mathrm{h2s}(x)) =_{\mathcal{H}} x$,
- $\forall x : \iota.x \in d \longrightarrow \mathrm{h2s}(\mathrm{s2h}(x)) =_{\mathcal{S}} x$,
- $\forall x : \sigma.\mathrm{h2s}(x) \in d$.

In Isabelle the definition appears as follows:

definition *beIsoS(h2s,s2h,d)* $\longleftrightarrow$ $((\forall_L y.\ s2h(h2s(y)) =_{\mathcal{H}} y) \wedge$
$(\forall x{:}\textit{Element-of } d.\ h2s(s2h(x)) =_{\mathcal{S}} x) \wedge (\forall_L y.\ h2s(y)\ \textit{in}\ d))$

The existence of a bijection does not immediately imply the inhabitation of the type/set. However, as types need to be non-empty in both formalisms, we can derive this result as below. For space reasons we only present the statements, all the theorems are proved in our formalization.

theorem *beIsoS_d*:
beIsoS(h2s,s2h,d) $\Longrightarrow$ *d is non empty*

4 Integrating Basic Infrastructure: Functions and Lists

We will denote the morphisms from set theory to HOL with the prefix `s2h` and the inverse ones with the prefix `h2s`. We will initially give the complete types for readability, omitting them later, where the types are clear. The first type, for which we build an isomorphism, is the type of functions. In order to transfer a function of the type $\alpha \to \beta$ between set theory and HOL, we will require isomorphisms for the types α and for the type β.

In order to transfer a set-theoretic function (set of pairs) to HOL, given transfer functions on the range, on the domain, and the function itself, we return the lambda expression, that given a HOL input to the function, transfers it, applies the function to it and transfers it back. The formal definition is as follows.

definition *s2hf* :: $(Set \Rightarrow b) \Rightarrow (a \Rightarrow Set) \Rightarrow Set \Rightarrow (a \Rightarrow b)$ $(s2h_f(\_,\_,\_))$ **where**
$s2h_f(s2hr,h2sd,f) =_{\mathcal{H}} (\lambda x.\ s2hr(f.(h2sd(x))))$

Similarly, to build a set-theoretic function (set of pairs) given a HOL function and the transfer operations, and the domain, we directly build this set:

definition *h2sf* :: $(Set \Rightarrow a) \Rightarrow (b \Rightarrow Set) \Rightarrow Set \Rightarrow (a \Rightarrow b) \Rightarrow Set$ $(h2s_f(\_,\_,\_,\_))$ **where**
$h2s_f(s2hd,h2sr,d,f) =_{\mathcal{S}}$ *the set-of-all* $[x,h2sr(f(s2hd(x)))]$ *where x be Element-of d*

We are then able to directly show that these two functions are inverses of each other on their domains. We also show the existence of an isomorphism, and show that this isomorphism preserves the function application operation:

theorem *beIsoT_Function*:
assumes *beIsoS(h2sd,s2hd,d)* *beIsoS(h2sr,s2hr,r)*
shows $beIsoT(\lambda f.\ h2s_f(s2hd,h2sr,d,f),\lambda f.\ s2h_f(s2hr,h2sd,f),\textit{Function-of } d,r)$

theorem *HtoSappl*:
assumes *beIsoS(h2sd,s2hd,d)* **and** *beIsoS(h2sr,s2hr,r)*

shows $h2s_f(s2hd,h2sr,d,f).h2sd(x) =_{\mathcal{S}} h2sr(f(x))$

Isabelle/HOL lists are realized as a polymorphic algebraic datatype, corresponding to functional programming language lists. MML lists (called finite sequences, FinSequence) are functions from an initial segment of the natural numbers. Higher-order lists behave like stacks, with access to the top of the stack, whereas for the set-theoretic ones the natural operations are the restriction or extension of the domain.

To build a bijection between these types, we note that the *Cons* operator corresponds to the concatenation of a singleton list and the second argument. Since the list type is polymorphic (in the shallow polymorphism sense used in HOL), in order to build this bijection, we also need to map the actual elements of the list. Therefore the bijection on lists will be parametric on a bijection on elements:

fun *h2sfs* :: $(a \Rightarrow Set) \Rightarrow a\ List.list \Rightarrow Set$ $(h2s_L(\_,\_))$**where**
$h2s_L(h2s,\ Nil) =_{\mathcal{S}} <*>$
| $h2s_L(h2s,\ Cons(h,\ t)) =_{\mathcal{S}} ((<*h2s(h)*>)\ \hat{}M\ (h2s_L(h2s,\ t)))$

Where $<*>$ and $\hat{}M$ represent the Mizar empty sequence and the concatenation of sequences respectively. The converse operation needs to decompose a sequence into its first element $x.1_{\mathcal{S}}$ and the remainder of the sequence shifted by one $/\hat{}M1_{\mathcal{S}}$. We define this operation in Isabelle/Mizar and complete the definition. Isabelle will again require us to show the termination of the function, which can be done by induction on the length of the list/sequence:

function *s2hl* :: $(Set \Rightarrow a) \Rightarrow Set \Rightarrow a\ List.list$ $(s2h_L(\_,\_))$ **where**
$\neg\ x\ be\ FinSequence \Longrightarrow s2h_L(s2h,x) =_{\mathcal{H}} undefined$
| $s2h_L(s2h,<*>) =_{\mathcal{H}} Nil$
| $x\ be\ FinSequence \Longrightarrow x \neq <*> \Longrightarrow s2h_L(s2h,x) =_{\mathcal{H}} Cons\ (s2h(x.1_{\mathcal{S}}),\ s2h_L(s2h,x/\hat{}M1_{\mathcal{S}}$
))

For the transformation introduced above, we can show that if we have a good homomorphism between the elements of the lists, then lists over this type are homomorphic with finite sequences.

We can again show that this homomorphism preserves various basic operations, such as concatenation, the selection of n-th element, length, etc.

theorem *s2hL_Prop*:
assumes *p be FinSequence* **and** *q be FinSequence* **and** *n be Nat* **and** *n in len p*
shows $length(s2h_L(s2h,p)) =_{\mathcal{H}} s2h_{\mathbb{N}}(len\ p)$
$s2h_L(s2h,p\hat{}Mq) =_{\mathcal{H}} s2h_L(s2h,p)$ @ $s2h_L(s2h,q)$
$s2h_L(s2h,p)$! $s2h_{\mathbb{N}}(n) =_{\mathcal{H}} s2h(p.\ (succ\ n))$

Note, that the sequences in the Mizar library, *FinSequence*, are indexed starting at 1, whereas Isabelle/HOL's *nth* starts from 0, which justifies the usage of a shift $(succ\ n)$. Furthermore, since Mizar Mathematical Library uses natural numbers in the Peano sense, the expression $n\ in\ len\ p$ actually means $n < len\ p$. To actually use these in order to move theorems between the libraries we show how the morphisms interact with the operations. For example, for reverse these are:

theorem *rev_Rev*:
assumes *p be FinSequence*
shows $rev(s2h_L(s2h,p)) =_{\mathcal{H}} s2h_L(s2h,Rev\ p)$

theorem *Rev_rev*:
$Rev(h2s_L(h2s,p)) =_{\mathcal{S}} h2s_L(h2s,rev(p))$

Moving a polymorphic statement from the Isabelle/HOL library to Isabelle/Mizar requires an additional assumption about the existence of an isomorphism on the parametrized type. The usual statement about the length of a reversed list, therefore becomes (of course this simple statement is already available in the Isabelle/Mizar library, and can be used by referring to *finseq_5_def_3*, but its simplicity is good to demonstrate moving polymorphic statements):

theorem
assumes *p be FinSequence-of d* **and** *beIsoS(h2s,s2h,d)*
shows *len Rev p* $=_S$ *len p*
using *Rev_rev*[*of h2s s2h*$_L$*(s2h, p)*]
len_length[*of h2s s2h*$_L$*(s2h, p)*]
len_length[*of h2s rev(s2h*$_L$*(s2h, p))*]
by (*simp only*: *length_rev FLF_prop*[*OF assms*])

We also show the proof here. It is still straightforward, just like the other proofs of the moved statements given the morphisms, but with polymorphism it no longer follows by higher-order rewriting.

5 Numbers

The way numbers are constructed in set-theory based libraries is very different from the majority of the libraries based on HOL or type-theory. In particular, in Isabelle/Mizar subsequently defined number types are extended (in the sense of set-theoretic subset) by new elements. This is as opposed to hard-type-based systems, in which subsequently defined number types are independent and projections or coercions which preserve the functions are necessary. In particular, Isabelle/Mizar's real numbers are constructed as Dedekind cuts. Note, however, that the cuts corresponding to the rational numbers are replaced by the rational numbers themselves, in order to preserve the inclusion $\mathbb{Q} \subset \mathbb{R}$.

A second, less important, distinction is the fact that in the Mizar library the non-negative types $(\mathbb{N}, \mathbb{Q}^{\geq 0}, \mathbb{R}^{\geq 0})$ are constructed first. After this, the negative reals are built as Kuratowski pairs of the singleton zero and the positive element. Finally, the rationals and integers are subsets of the set of all reals. In particular, the sets $\mathbb{N}, \mathbb{Q}^{\geq 0}, \mathbb{R}^{\geq 0}, \mathbb{R}$ are already constructed with the basic operations on these sets and addition, subtraction, multiplication directly re-use the real operations. The only additional thing to prove is that the types are preserved, so for example the addition of integers returns a real that is also an integer.

The inclusions, together with the order of the construction are depicted in Figure 5.1. In order to realize this construction in Isabelle/Mizar, we first define the set of the natural numbers, as the smallest limit ordinal. The formal definition is as follows:

mdef *ordinal1_def_11* (*omega*) **where**

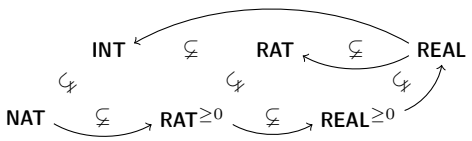

Fig. 5.1 The inclusions between the sets in the Mizar Mathematical Library. The arrows show the construction order between the sets in the MML and our Isabelle set formalization.

func omega → set means (λit.
$0_{\mathcal{S}}$ *in it* $\wedge$ *it be limit_ordinal* $\wedge$ *it be Ordinal* $\wedge$
($\forall A$:*Ordinal*. $0_{\mathcal{S}}$ *in* $A \wedge A$ *is limit_ordinal* $\longrightarrow$ *it* $\subseteq A$))

The definition introduces the constant (zero-argument functor) *omega* of the Mizar type *set*, which satisfies the condition specified after the keyword *means*, that is, the defined constant *it* is a limit ordinal with $0_{\mathcal{S}}$ as a member, and it is the smallest such set (considering set inclusion). As a reminder, the **mdef** command requires the formalization to specify the existence of the constant (proof is only included in the formalization), which is a consequence of the Tarski universe property and its uniqueness.

On the other hand, the Isabelle natural numbers are a subtype of the type of individuals. In order to merge these two different approaches, we specified a functor that preserves zero and the successor. Note that the functor is specified only for the type of the natural numbers which in Isabelle/HOL is implicit, but in the softly-typed set theory needs to be written and checked explicitly. This is the reason for having an undefined case, which as we will see later, still gives an isomorphism.

$$\mathrm{h2s}_{\mathbb{N}}(n) =_{\mathcal{S}} \begin{cases} 0_{\mathcal{S}} & \text{if } n =_{\mathcal{H}} 0_{\mathcal{H}}, \\ \mathrm{S}_{\mathcal{S}}(\mathrm{h2s}_{\mathbb{N}}(k)) & \text{if } n =_{\mathcal{H}} \mathrm{S}_{\mathcal{H}}(k) \text{ for some } \mathcal{H}\text{-natural } k. \end{cases}$$

$$\mathrm{s2h}_{\mathbb{N}}(n) =_{\mathcal{H}} \begin{cases} 0_{\mathcal{H}} & \text{if } n =_{\mathcal{S}} 0_{\mathcal{S}}, \\ \mathrm{S}_{\mathcal{H}}(\mathrm{s2h}_{\mathbb{N}}(k)) & \text{if } n =_{\mathcal{S}} \mathrm{S}_{\mathcal{S}}(k) \text{ for some } \mathcal{S}\text{-natural } k, \\ \textit{undefined} & \text{otherwise.} \end{cases}$$

The functor and its inverse are formally defined in Isabelle as follows

fun *h2sn* :: *nat* ⇒ *Set* (*h2s*$_{\mathbb{N}}$(_)) **where**
h2s$_{\mathbb{N}}$(*0*::*nat*) =$_{\mathcal{S}}$ $0_{\mathcal{S}}$ | *h2s*$_{\mathbb{N}}$(*Suc*(*x*)) =$_{\mathcal{S}}$ *succ h2s*$_{\mathbb{N}}$(*x*)

function *s2hn* :: *Set* ⇒ *nat* (*s2h*$_{\mathbb{N}}$(_)) **where**
¬*x be Nat* ⟹ *s2h*$_{\mathbb{N}}$(*x*) =$_{\mathcal{H}}$ *undefined*
| *s2h*$_{\mathbb{N}}$($0_{\mathcal{S}}$) =$_{\mathcal{H}}$ *0*
| *x be Nat* ⟹ *s2h*$_{\mathbb{N}}$(*succ*(*x*)) =$_{\mathcal{H}}$ *Suc*(*s2h*$_{\mathbb{N}}$(*x*))

Note that $\mathrm{h2s}_{\mathbb{N}}$ is defined only on the HOL natural numbers (*nat*), while $\mathrm{s2h}_{\mathbb{N}}$ is defined on all sets and its definition is only meaningful for arguments that are of the type *Nat*. The soft-type system of Mizar requires us to give this assumption explicitly here, but it can normally be hidden in the contexts where the argument type is restricted appropriately. Isabelle requires us to prove the termination of the definition, which can be done using the proper subset relation defined on natural numbers in the Peano sense.

Using the induction principles for natural numbers present in both libraries, we can show the property *beIsoS*($\mathrm{h2s}_{\mathbb{N}}$, $\mathrm{s2h}_{\mathbb{N}}$,*NAT*), where *NAT* is the set of all *Nat*. In particular, it gives a bijection (note the hidden type restriction to sets of type *nat*). We show also that the functors $\mathrm{h2s}_{\mathbb{N}}$, $\mathrm{s2h}_{\mathbb{N}}$ preserve all the basic operations.

theorem *Nat_to_Nat*:
fixes *x*::*nat* **and** *y*::*nat*
assumes *n be Nat* **and** *m be Nat*
shows *h2s*$_{\mathbb{N}}$(*x* +$_{\mathcal{H}}$ *y*) =$_{\mathcal{S}}$ *h2s*$_{\mathbb{N}}$(*x*) +$_{\mathcal{S}}{}^{\mathbb{N}}$ *h2s*$_{\mathbb{N}}$(*y*)
s2h$_{\mathbb{N}}$(*n* +$_{\mathcal{S}}{}^{\mathbb{N}}$ *m*) =$_{\mathcal{H}}$ *s2h*$_{\mathbb{N}}$(*n*) +$_{\mathcal{H}}$ *s2h*$_{\mathbb{N}}$(*m*)
h2s$_{\mathbb{N}}$(*x* *$_{\mathcal{H}}$ *y*) =$_{\mathcal{S}}$ *h2s*$_{\mathbb{N}}$(*x*) *$_{\mathcal{S}}{}^{\mathbb{N}}$ *h2s*$_{\mathbb{N}}$(*y*)
s2h$_{\mathbb{N}}$(*n* *$_{\mathcal{S}}{}^{\mathbb{N}}$ *m*) =$_{\mathcal{H}}$ *s2h*$_{\mathbb{N}}$(*n*) *$_{\mathcal{H}}$ *s2h*$_{\mathbb{N}}$(*m*)
x < *y* ⟷ *h2s*$_{\mathbb{N}}$(*x*) ⊂ *h2s*$_{\mathbb{N}}$(*y*)
n ⊂ *m* ⟷ *s2h*$_{\mathbb{N}}$(*n*) < *s2h*$_{\mathbb{N}}$(*m*)
x dvd y ⟷ *h2s*$_{\mathbb{N}}$(*x*) *divides h2s*$_{\mathbb{N}}$(*y*)

n *divides* $m \longleftrightarrow s2h_{\mathbb{N}}(n)$ *dvd* $s2h_{\mathbb{N}}(m)$
$prime(x) \longleftrightarrow h2s_{\mathbb{N}}(x)$ *is* $prime_{\mathcal{S}}$
n *is* $prime_{\mathcal{S}} \longleftrightarrow prime(s2h_{\mathbb{N}}(n))$

5.1 Isabelle/Mizar Number Hierarchy

After the natural numbers, MML constructs the non-negative rationals as pairs of relatively prime naturals. Additionally, to preserve the set-theoretic inclusion of the set of natural numbers, not only pairs with the denominator zero but also those with denominator one are excluded and the original natural numbers added. We follow the same construction in Isabelle/Mizar.

mdef *arytm_3_def_7* ($RAT^{\geq 0}$) **where**
func $RAT^{\geq 0} \rightarrow$ *set equals*
({[*i,j*] *where i be Element-of NAT, j be Element-of NAT*:
i,j are-coprime & $j \neq 0_{\mathcal{S}}$} \ *the set-of-all* [*k,*$1_{\mathcal{S}}$] *where k be Element-of NAT*) $\cup$ *NAT*

Non-negative real numbers are constructed in a similar way. To the set of non-negative rationals, we add Dedekind cuts corresponding to the positive irrational numbers. A standard definition of Dedekind cuts is used, only restricted to non-negative rationals. We assume that a proper subset A of non-negative rationals is a cut, if it is closed under smaller elements ($\forall r,s$: *Element-of* $RAT^{\geq 0}$. r *in* $A \wedge s \leq^{\mathbb{Q}^{\geq 0}} r \longrightarrow s$ *in* A) and for every element in the set A there is a larger element in the set A ($\forall r$: *Element-of* $RAT^{\geq 0}$. r *in* $A \longrightarrow (\exists s$: *Element-of* $RAT^{\geq 0}$. s *in* $A \wedge r <^{\mathbb{Q}^{\geq 0}} s$)). Note that $RAT^{\geq 0}$ fulfills this condition, however, it is not a proper subset of non-negative rationals. In contrast, in this approach, the empty set is a Dedekind cut, but we do not need to add it in the construction of $REAL^{\geq 0}$, since empty corresponds to zero.

mdef *arytm_2_def_1*(*DEDEKIND_CUTS*) **where**
func DEDEKIND_CUTS $\rightarrow$ *Subset-Family-of* $RAT^{\geq 0}$ *equals*
{ A *where* A *be Subset-of* $RAT^{\geq 0}$:
$\forall r$: *Element-of* $RAT^{\geq 0}$. r *in* $A \longrightarrow$
($\forall s$: *Element-of* $RAT^{\geq 0}$. $s \leq^{\mathbb{Q}^{\geq 0}} r \longrightarrow s$ *in* A) $\wedge$
($\exists s$: *Element-of* $RAT^{\geq 0}$. s *in* $A \wedge r <^{\mathbb{Q}^{\geq 0}} s$)} \ {$RAT^{\geq 0}$}

In order to preserve the inclusion between the rationals and reals, again the non-negative real numbers are obtained as a union of the non-negative rationals as defined above and the Dedekind cuts corresponding to the irrational numbers, that is cuts that cannot be realized in the form {s *where* s *be Element-of* $RAT+$: $s <^{\mathbb{Q}^{\geq 0}} q$} where q is rational.

mdef *arytm_2_def_2* ($REAL^{\geq 0}$) **where**
func $REAL^{\geq 0} \rightarrow$ *set equals* ($RAT^{\geq 0} \cup$ *DEDEKIND_CUTS*) \
{{s *where* s *be Element-of* $RAT^{\geq 0}$: $s <^{\mathbb{Q}^{\geq 0}} t$} *where* t *be Element-of* $RAT^{\geq 0}$: $t \neq 0_{\mathcal{S}}$}

Finally, the complete reals (*REAL*) are constructed by adding the negative real numbers. In the Mizar set theory the negative numbers are represented by the pairs [$0_{\mathcal{S}}$,r], where r is a positive real number. For this, we add the pairs corresponding to r, where r is a non-negative real and then remove the pair [$0_{\mathcal{S}}$,$0_{\mathcal{S}}$] to avoid duplicating 0. The sets of rationals and integers are then appropriate subsets of the set *REAL*. Of course, it would be possible to build these sets directly, together with their respective

arithmetic operations, however, this would require the introduction of different symbols for these operations in the different datatypes. The Isabelle/Mizar formalization only temporarily introduces the operations $\mathbb{Q}^{\geq 0}, \mathbb{R}^{\geq 0}$ which will almost never be used in the library, and the operations for the type $\mathbb{R}$, which will be directly reused for $\mathbb{Z}$ and $\mathbb{Q}$. In particular, this allows using the operations in the context of homomorphisms between integers, rationals, and reals.

mdef *numbers_def_1* (*REAL*) **where**
func REAL $\rightarrow$ *set equals*
$REAL^{\geq 0} \cup [:\{0_{\mathcal{S}}\}, REAL^{\geq 0}:] \setminus \{[0_{\mathcal{S}}, 0_{\mathcal{S}}]\}$

mdef *numbers_def_3* (*RAT*) **where**
func RAT $\rightarrow$ *set equals*
$RAT^{\geq 0} \cup [:\{0_{\mathcal{S}}\}, RAT^{\geq 0}:] \setminus \{[0_{\mathcal{S}}, 0_{\mathcal{S}}]\}$

mdef *numbers_def_4* (*INT*) **where**
func INT $\rightarrow$ *set equals*
$NAT \cup [:\{0_{\mathcal{S}}\}, NAT:] \setminus \{[0_{\mathcal{S}}, 0_{\mathcal{S}}]\}$

5.2 Integrating Numbers

Given the Isabelle/Mizar number hierarchy specified in the previous section, we can start building bridges between the types. We start with the integers. The set-theoretic definition is again different from the one used in Isabelle/HOL. There, an equivalence relation (equal modulo the difference) is defined on pairs of natural numbers, and the quotient package [KU11] is used to construct the new type. Still, it is straightforward to define a bijection between the two, using the constructed bijections between natural numbers. We also show that these bijections preserve all the basic operators.

function *h2sZ* :: *int* $\Rightarrow$ *Set* ($h2s_{\mathbf{Z}}(\_)$)**where**
$x \geq 0 \Longrightarrow h2s_{\mathbf{Z}}(x) =_{\mathcal{S}} h2s_{\mathbb{N}}(nat(x))$
| $x < 0 \Longrightarrow h2s_{\mathbf{Z}}(x) =_{\mathcal{S}} \text{-}_{\mathcal{S}}^{\mathbb{R}}\ h2s_{\mathbb{N}}(nat(\text{-}_{\mathcal{H}}(x)))$

function *s2hZ* :: *Set* $\Rightarrow$ *int* ($s2h_{\mathbf{Z}}(\_)$)**where**
$\neg x$ *is Integer* $\Longrightarrow s2h_{\mathbf{Z}}(x) =_{\mathcal{H}}$ *undefined*
| *x is natural* $\Longrightarrow s2h_{\mathbf{Z}}(x) =_{\mathcal{H}} int(s2h_{\mathbb{N}}(x))$
| *x is Integer & not x is natural* $\Longrightarrow s2h_{\mathbf{Z}}(x) =_{\mathcal{H}} \text{-}_{\mathcal{H}}\ (int(s2h_{\mathbb{N}}(\text{-}_{\mathcal{S}}^{\mathbb{R}}\ x)))$

theorem *beIsoS_INT*:
$beIsoS(h2s_{\mathbf{Z}}, s2h_{\mathbf{Z}}, INT)$

For the rational numbers, we construct the natural bijection $\mathrm{h2s}_{\mathbb{Q}}$, $\mathrm{s2h}_{\mathbb{Q}}$ using the bijections between the integers and the unique representation of any rational as an irreducible fraction. We again show that the operations behave well on arbitrary (including reducible) fractions.

theorem *s2hQI*:
fixes *n*::*nat*
shows $n \neq_{\mathcal{H}} 0 \longrightarrow Fract(i,n) =_{\mathcal{H}} s2h_{\mathbb{Q}}(h2sZ(i)\ /_{\mathbb{Q}}\ h2sZ(n))$

theorem *s2hQM*:
i is Integer $\wedge$ *n is Nat* $\wedge$ $n \neq \{\} \longrightarrow s2h_{\mathbb{Q}}(i\ /_{\mathbb{Q}}\ n) =_{\mathcal{H}} Fract(s2h_{\mathbf{Z}}(i), s2h_{\mathbf{Z}}(n))$

The constructions of the real numbers are significantly different in the two considered proof libraries. Indeed, in Isabelle/HOL reals are quotients of Cauchy sequences whereas the MML one uses Dedekind cuts. More precisely, in the MML, Dedekind cuts are used to construct the irrational, and operations on them are defined on the cuts. To build a homomorphism between the two definitions and to use it for all the operators requires considering cases, namely whether the given argument is a rational number of a cut. The same is true for the results of the operators.

To ease these constructions we first introduce two operators: *DEDEKIND_CUT* which transform a real number to a Dedekind cut, i.e. for positive rationals it associates to the number r the cut $\{s$ *where* s *be Element-of* $RAT^{\geq 0}$: $s <^{\mathbb{Q}^{\geq 0}} r\}$, and for irrational numbers, which are already cuts, it is the identity. We also define the inverse operator *GLUE*, which transforms cuts that can be represented in the form $\{s$ *where* s *be Element-of* $RAT^{\geq 0}$: $s <^{\mathbb{Q}^{\geq 0}} r\}$ for a rational r, returns r, and is the identity otherwise.

mdef *arytm_2_def_3* (*DEDEKIND_CUT* _) **where**
mlet *x be Element-of* $REAL^{\geq 0}$
func DEDEKIND_CUT x $\rightarrow$
Element-of DEDEKIND_CUTS means
$\lambda it.\ \exists\, r$:*Element-of* $RAT^{\geq 0}$. $x =_{\mathcal{S}} r \wedge$
$it = \{s$ *where* s *be Element-of* $RAT^{\geq 0}$: $s <^{\mathbb{Q}^{\geq 0}} r\}$
if x in $RAT^{\geq 0}$
otherwise $\lambda it.\ it =_{\mathcal{S}} x$

mdef *arytm_2_def_4* (*GLUED* _) **where**
mlet *x be Element-of DEDEKIND_CUTS*
func GLUED x $\rightarrow$ *Element-of* $REAL^{\geq 0}$ *means*
$\lambda it.\ \exists\, r$: *Element-of* $RAT^{\geq 0}$. $it = r \wedge$
$(\forall\, s$: *Element-of* $RAT^{\geq 0}$. s *in* $x \longleftrightarrow s <^{\mathbb{Q}^{\geq 0}} r)$
if $\exists\, r$: *Element-of* $RAT^{\geq 0}$. $\forall\, s$: *Element-of* $RAT^{\geq 0}$.
s *in* $x \longleftrightarrow s <^{\mathbb{Q}^{\geq 0}} r$
otherwise λ *it*. $it = x$

We will now construct the homomorphism between the real number representations. Consider a non-empty Dedekind cut A. We observe, that by multiplying all the elements of A by a positive rational q, we obtain a non-empty Dedekind cut. We denote this cut by $q *_D A$. Next, we denote by $max_{\mathbb{N}}\ A$ the largest natural number in the set A. Consider the sequence of non-negative rationals $\left\{\frac{max_{\mathbb{N}}\ (2^n *_D A)}{2^n}\right\}_{n\in\mathbb{N}}$. It easily follows that this sequence is non-decreasing and that for every $n \leq k$ it is true that

$$\frac{max_{\mathbb{N}}\ (2^n *_D A)}{2^n} \leq \frac{max_{\mathbb{N}}\ (2^k *_D A)}{2^k} \leq \frac{max_{\mathbb{N}}\ (2^n *_D A)}{2^n} + \frac{1}{2^n}$$

which shows that this sequence is a Cauchy sequence.

This allows us to associate any positive real number with a Cauchy sequence of rationals:

mdef *Rat2C*(*rC* _ *110*) **where**
mlet *r be Element-of* $REAL^{\geq 0}$
func rC r $\rightarrow$ *Function-of NAT,RAT means*
$\lambda it.\ \forall\ n$:*Nat*. *it*. $n = (max_{\mathbb{N}}\ (\ (\ 2_{\mathcal{S}}\ |\hat{}\ n)\ *_D\ (DEDEKIND\_CUT\ r))) /_{\mathbb{Q}}\ (\ 2_{\mathcal{S}}\ |\hat{}\ n)$

Using the previously defined homomorphisms between the naturals and rationals as well as between the types of functions (Section 4 and previous subsections of Sec-

tion 5), we can transform this set-theoretic function to a HOL one. We show that this transformation preserves Cauchy convergence:

definition *s2hseq* :: *Set* ⇒ (*nat* ⇒ *rat*) (*s2hseq*(_)) **where**
s2hseq(*f*) =$_{\mathcal{H}}$ *s2hf*(*s2hQ*,*h2sn*,*f*)

theorem *MICauchy*:
assumes *f is Function-of NAT*,*RAT*
shows *f is Cauchy*$_{\mathcal{S}}$ ⟷ *Real.cauchy* (*s2hseq*(*f*))

Which allows us to define the final homomorphism that given a set-theoretic real transforms it to a HOL real.

function *s2hR* :: *Set* ⇒ *real* (*s2h*$_{\mathbb{R}}$(_))**where**
¬*x is MReal* ⟹ *s2h*$_{\mathbb{R}}$(*x*) =$_{\mathcal{H}}$ *undefined*
| *x is Element-of REAL*$^{\geq 0}$ ∧ *x*≠*0*$_{\mathcal{S}}$ ⟹ *s2h*$_{\mathbb{R}}$(*x*) =$_{\mathcal{H}}$ *Real.Real*(*s2hseq*(*rC x*))
| *x is Element-of REAL*$^{\geq 0}$ ∧ *x*=*0*$_{\mathcal{S}}$ ⟹ *s2h*$_{\mathbb{R}}$(*x*) =$_{\mathcal{H}}$ *0*
| *x is MReal* ∧ *x*≠*0*$_{\mathcal{S}}$ ∧ ¬ *x is Element-of REAL*$^{\geq 0}$ ⟹
s2h$_{\mathbb{R}}$(*x*) =$_{\mathcal{H}}$ -$_{\mathcal{H}}$ *Real.Real*(*s2hseq*(*rC*(-$_{\mathcal{S}}^{\mathbb{R}}$ *x*)))

where for non-negative real number x, we use it to produce the sequence of rational numbers $rC\,x$, which are subsequently transformed to a sequence of HOL reals $\mathrm{s2h}seq(rC\ x)$, and finally we return the abstraction of the Cauchy sequence class to which the sequence belongs. For negative real numbers, we use minus twice, analogously to the integer and rational constructions. -$_{\mathcal{H}}$ (...(-$_{\mathcal{S}}^{\mathbb{R}}$ *x*))

In order to build the inverse transformation, we will construct the Dedekind cut based on a real number. First, for any real number r, we start with one of the Cauchy sequence *real2seqL*(r) belonging to its equivalence class r. We consider the equivalence of this sequence in set theory: $\mathrm{h2s}seq(r)$. This sequence is non-decreasing and has non-negative values if r is non-negative. Additionally, if r is positive, this sequence $\mathrm{h2s}seq(r)$ is also positive starting from some index. This means that for any positive real r, the sequence $\{s$ *where s be Element-of RAT*$^{\geq 0}$: $s <^{\mathbb{Q}^{\geq 0}} \mathrm{h2s}seq(r).n\ \}_{n\in\mathbb{N}}$ is non-empty (from some position, to be precise when $\mathrm{h2s}seq(r).n \neq 0_{\mathcal{S}}$) and non-decreasing and its union (*seq2Dedekind*) is a Dedekind cut.

definition *real2seqL* :: *real* ⇒ (*nat*⇒*rat*) **where**
real2seqL(*r*) =$_{\mathcal{H}}$ (λ*n*::*nat*. *Fract*(⌊*r* *$_{\mathcal{H}}$ (*2*^*n*)⌋,*2*^*n*))

definition *h2sseq* :: *real* ⇒ *Set* (*h2sseq*(_)) **where**
h2sseq(*r*) = *h2sf*(*s2hn*,*h2sQ*,*NAT*,*real2seqL*(*r*))

mdef *seq2Dedekind* **where**
mlet *f be Function-of NAT, RAT*
func seq2Dedekind(*f*) → *Subset-of RAT*$^{\geq 0}$ *means*
λ*it*. ∀ *x*:*Element-of RAT*$^{\geq 0}$. *x in it* ⟷ (∃ *k*:*Nat*. *x* <$^{\mathbb{R}}$ (*f* .*k*))

The final transformation that given a HOL real number extracts its Cauchy sequence and transforms it to an Isabelle/Mizar real is:

function *h2sR* :: *real* ⇒ *Set* (*h2s*$_{\mathbb{R}}$(_))**where**
x > *0* ⟹ *h2s*$_{\mathbb{R}}$(*x*) = *GLUED*(*seq2Dedekind*(*h2sseq*(*x*)))
| *x* =$_{\mathcal{H}}$ *0* ⟹ *h2s*$_{\mathbb{R}}$(*x*) = *0*$_{\mathcal{S}}$
| *x* < *0* ⟹ *h2s*$_{\mathbb{R}}$(*x*) = -$_{\mathcal{S}}^{\mathbb{R}}$ *GLUED*(*seq2Dedekind*(*h2sseq*(-$_{\mathcal{H}}$ *x*)))

The two defined operations $\mathrm{s2h}_{\mathbb{R}}$ and $\mathrm{h2s}_{\mathbb{R}}$ are not as straightforward as for the naturals or rationals. We do nonetheless prove (details are only in the formalization)

that they do indeed give an isomorphism and that this isomorphism preserves the basic arithmetic operations and the standard less than order.

theorem *beIsoS_Real*:
beIsoS(*h2sR*,*s2hR*,*REAL*)

theorem *Real_to_Real*:
fixes x::*real* **and** y::*real*
assumes r *be MReal* **and** s *be MReal*
shows $h2s_{\mathbb{R}}(x +_{\mathcal{H}} y) =_{\mathcal{S}} h2s_{\mathbb{R}}(x) +_{\mathcal{S}}{}^{\mathbb{R}} h2s_{\mathbb{R}}(y)$
$s2h_{\mathbb{R}}(r +_{\mathcal{S}}{}^{\mathbb{R}} s) =_{\mathcal{H}} s2h_{\mathbb{R}}(r) +_{\mathcal{H}} s2h_{\mathbb{R}}(s)$
$h2s_{\mathbb{R}}(x *_{\mathcal{H}} y) =_{\mathcal{S}} h2s_{\mathbb{R}}(x) *_{\mathcal{S}}{}^{\mathbb{R}} h2s_{\mathbb{R}}(y)$
$s2h_{\mathbb{R}}(r *_{\mathcal{S}}{}^{\mathbb{R}} s) =_{\mathcal{H}} s2h_{\mathbb{R}}(r) *_{\mathcal{H}} s2h_{\mathbb{R}}(s)$
$x \leq y \longleftrightarrow h2s_{\mathbb{R}}(x) \leq^{\mathbb{R}} h2s_{\mathbb{R}}(y)$
$r \leq^{\mathbb{R}} s \longleftrightarrow s2h_{\mathbb{R}}(r) \leq s2h_{\mathbb{R}}(s)$

We are now ready to practically move proved theorems about numbers between HOL and Isabelle/Mizar.

6 Algebra

The structure representations used in higher-order logic and set theory are usually different. This will be particularly visible when it comes to algebraic structures. In the Isabelle/HOL formalization, algebraic structures are type-classes while in set theory a common approach would be partial functions. We will illustrate the difference on the example of groups. A type α forms a group when we can indicate a binary function on this type that will serve as the group operation satisfying the group axioms. On the other hand, in the usual set-theoretic approach a group in set theory would consist of an explicitly given set (the carrier), and the group operation. With an intersection type system, the fact that the given set with an operation is a group is specified by intersecting the type of structures with the types that specify their individual properties (i.e. a group is a non-empty associative Group-like multMagma)

There are two more differences in the particular formalizations we consider, that we will not focus on, but we will only mention them in this paragraph and consider them only in the formalization. First, the existence and uniqueness of the neutral element can be either assumed in the group specification or derived from the axioms. We will not focus on that, as this is only the choice of a group axiomatization. Second, in the Mizar library, there are two theories of groups: additive groups and multiplicative groups. Rings and fields inherit the latter, while some group-theoretic results are derived only for the former. Even if the Isabelle/HOL group includes a field for the unit, we will ignore it in the morphism, since the set-theoretic definition does not use one. The neutral element along with the other properties is, however, necessary to justify that the result of the morphism is a group in the set-theoretic sense.

definition *h2sg* ($h2s_G(\_,\_,\_,\_)$) **where**
$h2s_G(s2hc,h2sc,c,g) =_{\mathcal{S}}$ [#
carrier $\mapsto$ c;
multF $\mapsto$ $h2s_{BinOp}(s2hc,h2sc,c,mult(g))$ #]

definition *s2hg* ($s2h_G(\_,\_,\_)$) **where**
$s2h_G(s2hc,h2sc,g) =_{\mathcal{H}}$ *Igroup*(
Collect(λx. *h2sc*(x) *in the carrier of* g),
$s2h_{BinOp}$(*s2hc*,*h2sc*,*the multF of* g),

s2hc(1_g))

For the dual morphism, we indicate the result of the operation selecting the neutral element (1_g) as the field needed in the construction of the type-class element. With its help, we can justify that the fields of the translated structure are translations of the fields.

theorem *s2hg_Prop*:
  assumes *beIsoS*(*h2sc*,*s2hc*,*c*) **and** *g be Group*
    and *the carrier of g* $=_{\mathcal{S}}$ *c*
    and $x \in$ *carrierI*(*s2h*$_G$(*s2hc*, *h2sc*, *g*))
      $y \in$ *carrierI*(*s2h*$_G$(*s2hc*, *h2sc*, *g*))
  shows *one*(*s2h*$_G$(*s2hc*,*h2sc*,*g*)) $=_{\mathcal{H}}$ *s2hc*(1_g)
      $x \otimes_{s2h_G(s2hc,h2sc,g)} y =_{\mathcal{H}}$ *s2hc*(*h2sc*(x) $\otimes_g$ *h2sc*(y))
      group (*s2h*$_G$(*s2hc*,*h2sc*,*g*))

A number of proof assistant systems based both on higher-order logic (including Isabelle/HOL) and set theory (including Mizar) support inheritance between their algebraic structures. As part of our work aligning the libraries we also want to verify that such inheritance is supported in the combined library. For this, we align the ring structures present in the two libraries. The isomorphism between the structures is defined in a similar way to the one for groups, we refer the interested reader to our formalization.

We can show that the morphisms form an isomorphism and derive some basic preservation properties. The most basic one is the fact that the isomorphism preserves being a ring.

theorem *s2hr_Prop*:
  assumes *beIsoS*(*h2sc*,*s2hc*,*c*) **and** *r be Ring*
    and *the carrier of r* $=_{\mathcal{S}}$ *c*
    and $x \in$ *carrierI*(*s2h*$_R$(*s2hc*,*h2sc*,*r*)) $y \in$ *carrierI*(*s2h*$_R$(*s2hc*,*h2sc*,*r*))
  shows *zero*(*s2h*$_R$(*s2hc*,*h2sc*,*r*)) $=_{\mathcal{H}}$ *s2hc*(0_r)
      one(*s2h*$_R$(*s2hc*,*h2sc*,*r*)) $=_{\mathcal{H}}$ *s2hc*(1_r)
      $x \oplus_{s2h_R(s2hc,h2sc,r)} y =_{\mathcal{H}}$ *s2hc*(*h2sc*(x) $\oplus_r$ *h2sc*(y))
      $x \otimes_{s2h_R(s2hc,h2sc,r)} y =_{\mathcal{H}}$ *s2hc*(*h2sc*(x) $\otimes_r$ *h2sc*(y))
      ring (*s2h*$_R$(*s2hc*,*h2sc*,*r*))

Finally, we introduce the equivalent of the definition of the integer ring introduced in the MML in [Sch99]. We have previously discussed the semantics of Mizar structures and the way they are represented in Isabelle/Mizar in [KP17]. Here, with the previously defined isomorphisms for the subfields, we can show that $\mathtt{s2h}_R$ and $\mathtt{h2s}_R$ determine an isomorphism between the fields of the rings developed in Isabelle/HOL and the Mizar Mathematical Library.

mdef *int_3_def_3* (**ℤ**-*ring*) **where**
  func **ℤ**-*ring* $\rightarrow$ *strict*(*doubleLoopStr*) *equals* [#
    carrier $\mapsto$ *INT*;
    addF $\mapsto$ *addint*;
    ZeroF $\mapsto$ $0_{\mathcal{S}}$;
    multF $\mapsto$ *multint*;
    OneF $\mapsto$ $1_{\mathcal{S}}$#]

theorem *H_Zring_to_S_Zring*:
  h2s$_R$(*s2h*$_{\mathbf{Z}}$, *h2s*$_{\mathbf{Z}}$,*INT*,$\mathcal{Z}$) $=_{\mathcal{S}}$ **ℤ**-*ring*
  s2h$_R$(*s2h*$_{\mathbf{Z}}$, *h2s*$_{\mathbf{Z}}$, **ℤ**-*ring*) $=_{\mathcal{H}}$ $\mathcal{Z}$

7 Integrated Libraries: Practical Examples

We are now ready to use the existence of isomorphisms to automatically transform theorems about continuity of functions, including the Intermediate Value Theorem and the theorem that states that the image of a closed interval is a closed interval:

theorem *continuous_atM*:
fixes *f a*
assumes *f be Function-of REAL,REAL a is MReal*
shows $isCont(s2h_f\,\mathbb{R}(f),s2hR(a)) \longleftrightarrow f$ *is_continuous_in* a

theorem *continuous_atI*:
fixes f::*real*⇒*real*
shows $isCont(f,a) \longleftrightarrow (h2s_f\,\mathbb{R}(f))$ *is_continuous_in* $(h2sR(a))$

theorem *IVTmiz*:
$\forall f$:*Function-of REAL,REAL*. $\forall a,b,v$:*MReal*. $f.\ a \leq^{\mathbb{R}} v\ \&\ v \leq^{\mathbb{R}} f.\ b\ \&\ a \leq^{\mathbb{R}} b\ \&$
f *is_continuous_on* $[.a,\ b.] \longrightarrow$
$(\exists\, x$:*MReal*. $a \leq^{\mathbb{R}} x\ \&\ x \leq^{\mathbb{R}} b\ \&\ f.\ x = v)$

theorem *IVT_img*:
$\forall f$:*Function-of REAL,REAL*. $\forall\ a,b$:*MReal*.
$a \leq^{\mathbb{R}} b \wedge f$ *is_continuous_on* $[.a,\ b.] \longrightarrow$
$(\exists\, c,d$:*MReal*. $c \leq^{\mathbb{R}} d \wedge f\ .:\ [.\ a,\ b\ .] =_{\mathcal{S}}\ [.\ c,\ d\ .])$

We also show the projection theorem, which again states that the homomorphisms agree and do not require any projections:

theorem
n *is Nat* $\Longrightarrow$ $\mathit{of\_nat}(s2h_{\mathbb{N}}(n)) =_{\mathcal{H}} \mathit{of\_int}(s2h_{\mathbb{Z}}(n))$
i *is Integer* $\Longrightarrow$ $\mathit{of\_int}(s2h_{\mathbb{Z}}(i)) =_{\mathcal{H}} \mathit{of\_rat}(s2h_{\mathbb{Q}}(i))$
q *is Rat* $\Longrightarrow$ $\mathit{of\_rat}(s2h_{\mathbb{Q}}(q)) =_{\mathcal{H}} s2h_{\mathbb{R}}(q)$

It is now possible to translate the Lagrange's Four Squares theorem and Bertrand's postulate between the libraries. We can prove the Isabelle/Mizar counterpart of the Isabelle/HOL theorem only using higher-order rewriting and the above properties.

theorem *LagrangeFourSquares*:
$\forall\, n$:*Nat*. $\exists\, a,b,c,d$:*Nat*.
$a *_{\mathcal{S}}^{\mathbb{N}} a +_{\mathcal{S}}^{\mathbb{N}} b *_{\mathcal{S}}^{\mathbb{N}} b +_{\mathcal{S}}^{\mathbb{N}} c *_{\mathcal{S}}^{\mathbb{N}} c +_{\mathcal{S}}^{\mathbb{N}} d *_{\mathcal{S}}^{\mathbb{N}} d =_{\mathcal{S}} n$

theorem *Bertrand*:
$\forall\, n$:*Nat*. $1_{\mathcal{S}} \subset n \longrightarrow$
$(\exists\, p$:*Nat*. p *be prime*$_{\mathcal{S}} \wedge n \subset p \wedge p \subset (2_{\mathcal{S}} *_{\mathcal{S}}^{\mathbb{N}} n))$

This allows translating the proved Fermat's last theorem for powers divisible by 3 and 4 from Isabelle/HOL to Isabelle/Mizar. The original proof involved quite some computation and therefore has not been attempted in Mizar so far. However, thanks to the isomorphisms, the translated version can be proved automatically (higher-order rewriting combined with Isabelle/Mizar type automation):

theorem *Fermat_divides_3_4*:
$\forall\, x,y,z$:*Integer*. $\forall\, n$:*Nat*.
$(3_{\mathcal{S}}$ *divides* $n \vee 4_{\mathcal{S}}$ *divides* $n) \wedge (x\ I\hat{}\ n) +_{\mathcal{S}}^{\mathbb{R}} (y\ I\hat{}n) =_{\mathcal{S}} z\ I\hat{}n$
$\longrightarrow x *_{\mathcal{S}}^{\mathbb{R}} y *_{\mathcal{S}}^{\mathbb{R}} z =_{\mathcal{S}} 0_{\mathcal{S}}$

theorem *Fermat_3*:
$\forall\, x$:*Integer*. $\forall\, y$:*Integer*. $\forall\, z$:*Integer*.

$(x\ I\hat{}\ 3_{\mathcal{S}})\ +_{\mathcal{S}}^{\mathbb{R}}\ (y\ I\hat{}3_{\mathcal{S}}) = z\ I\hat{}\ 3_{\mathcal{S}} \longrightarrow x\ *_{\mathcal{S}}^{\mathbb{R}}\ y\ *_{\mathcal{S}}^{\mathbb{R}}\ z =_{\mathcal{S}}\ 0_{\mathcal{S}}$

theorem *rev_Rev*:
assumes *p be FinSequence*
shows $rev(s2h_L(s2h,p)) =_{\mathcal{H}}\ s2h_L(s2h,Rev\ p)$

8 Tarski's axiom vs. Grothendieck Universes

The theoretical part of our previous work [BKP19] formally introduced a foundation for computer verified proofs based on higher-order Tarski-Grothendieck set theory (HOTG) and prove that this theory has a model if a 2-inaccessible cardinal exists. Referring to the former as the axioms of Tarski-Grothendieck is, however, slightly misleading, as there are two not immediately equivalent families of axioms. In particular, the two axiom families are equivalent assuming the axiom of choice. Additionally, the axiom of choice is a consequence of the Tarski axioms, but it is not the case for the Grothendieck formulation. Both of these facts are now also formalized in Isabelle, and shortly discussed in this section.

The formalization done in this section is done independently from Isabelle/HOL or Isabelle/Mizar as its goal is to formally justify that Tarski's axiom A is valid in the model proposed in [BKP19]. Recall, that Tarski's axiom A is used in the Mizar library and in Isabelle/Mizar, whereas the existence of a Grothendieck universe is used for example in Egal.

Tarski's Axiom A states that every set N is a member of some Tarski universe M which is closed under subsets, powersets, and every subset of the universe is either a member of the universe or is equipotent with that universe. To state this formally, the equipotence between the sets X and Y can be defined by a set of Kuratowski pairs, which defines a bijection from X to Y using only a minimal set of definitions, as it is done for example in the MML:

definition *Tarski_axiom_A* **where**
$Tarski\_axiom\_A \equiv \forall\ N.\ \exists\ M.$
$N \in M \wedge$
$(\forall\ X\ Y.\ X \in M \wedge Y \subseteq X \longrightarrow Y \in M) \wedge$
$(\forall\ X.\ X \in M \longrightarrow Pow\ X \in M) \wedge$
$(\forall\ X.\ X \subseteq M \longrightarrow (\exists\ b.\ b\text{: } bij\ X\ M) \vee X \in M)$

In the Grothendieck approach, for an arbitrary set X, we can explicitly obtain the Grothendieck universe $\mathsf{Univ}X$. The universe $\mathsf{Univ}X$ is transitive ($Trans\ (Univ\ X)$), closed under union, powerset, and replacement ($ZFclosed\ (Univ\ X)$) and it is the smallest set (w.r.t. set inclusion) having these properties.

axiomatization
$Univ :: set \Rightarrow set$ **where**
UnivIn: $X \in Univ\ X$ **and**
UnivTransSet: $Trans\ (Univ\ X)$ **and**
UnivZF: $ZFclosed\ (Univ\ X)$ **and**
UnivMin: $X \in U \wedge Trans\ U \wedge ZFclosed\ U \Longrightarrow Univ\ X \subseteq U$

To compare these two axiomatizations, we have previously shown in the higher-order logic of Egal that every Grothendieck universe, under the axiom of choice assumption, satisfies Tarski's Axiom A (see [BP19]), but, not vice versa. Tarski universes, as opposed to Grothendieck universes, might not be transitive. We constructed such a

Tarski universe of a set N that is a proper subset of $\mathsf{Univ}N$ in [Pąk20] in the first-order logic of Mizar, as well as proved that $\mathsf{Univ}N$ included in every Tarski universe of a set N if N is transitive.

In particular, using these properties, we proved in Isabelle that assuming HOTG and the axiom of choice, $\mathsf{Univ}\,N$ is a Tarski universe, i.e., that in the model [BKP19], Tarski's Axiom A is valid. Rather than repeat the proofs already described in [BP19] we show the final statement that we proved under the axiom of choice as rendered by Isabelle:

definition *AC_axiom* **where**
$AC\_axiom \equiv \forall\ X.\ \{\} \notin\ X \longrightarrow (\exists\ f.\ (f \in X \to \bigcup\ X) \wedge (\forall A.\ A \in X \longrightarrow f`\ A \in A))$

theorem
$AC\_axiom \longrightarrow Tarski\_axiom\_A$

In order to even more closely show the adequacy of the HOTG model for importing the Isabelle/HOL proofs, one might also consider polymorphism, which is present in the foundations of the HOL families of provers. Andrew Pitts has provided a custom semantics to HOL that factors in polymorphism [Pit93]. We however believe, that since the polymorphism in HOL is shallow (rank-one), it can be considered a notation for monomorphic HOL, namely all proofs can be translated to monomorphic ones and that the Grothendieck universes offer enough room for the quantification incurred by polymorphism. Extending the model to support all the custom extensions present in Isabelle/HOL (such as e.g. type classes [HN10] or local type definitions [KP19b]) is left as future work.

9 Related Work

Since proof assistants based on plain higher-order logic lack the full expressivity of set theory, the idea of adding set theory axioms on top of HOL has been tried multiple times. Gordon [Gor96] discusses approaches to combine the power of HOL and set theory. Obua has proposed HOLZF [Obu06], where Zermelo-Fraenkel axioms are added on top of Isabelle/HOL. With this, he was able to show results on partisan games, that would be hard to show in plain higher-order logic. Later, as part of the ProofPeer project [OFSA14], the combination of HOL with ZF became the basis for an LCF system, reducing the proofs in the higher-order logic part to a minimum (again, since there was no guarantee, that combining the results is safe). Kunčar [Kun10] attempted to import the Tarski-Grothendieck-based library into HOL Light. Here, the set-theoretic concepts were immediately mapped to their HOL counterparts, but it soon came out that without adding the axioms of set theory the system was not strong enough. Brown [Bro14] proposed the Egal system which again combines a specification of higher-order logic with the axioms of set theory. The system uses explicit universes, which is in fact the same presentation as given in this work. This work therefore also gives a model for the Egal system. Finally, we have specified [KP18] and imported [KP19a] significant parts of the Mizar library into Isabelle. In this work, we only use the specification of Mizar in Isabelle and the re-formalized parts of the MML.

The idea to combine proof assistant libraries across different foundations also arose in the Flyspeck project [HAB+17] formalizing the proof of the Kepler conjecture [HHM+10]. Krauss and Schropp [KS10] specified and implemented a translation from Isabelle/HOL proof terms to set-theoretic proved theorems. The translation is sound

and only relies on the Isabelle/ZF logic, however, it is too slow to be useful in practice, in fact, it is not possible to translate the basic Main library of Isabelle/HOL into set theory in reasonable time[2] It is also possible to deep embed multiple libraries in a single meta-theory. Rabe [Rab17] does this practically in the MMT framework deep embedding various proof assistant foundations and providing category-theoretic mappings between some foundations. Logical frameworks allow importing multiple libraries at the same time. In the Dedukti framework, Assaf and Cauderlier [AC15, Ass15] have combined properties originating from the Coq library and the HOL library. Both were imported in the same system, based on the λ_{Π} calculus modulo, however, the two parts of the library relied on different rewrite rules.

Most implementations of set theory in logical frameworks could implicitly use some higher-order features of the framework, as this is already used for the definition of the object logic. The definition of the Zermelo-Fraenkel object logic [Pau93] in Isabelle uses lambda abstractions and higher-order applications for example to specify the quantifiers. This is also the case in Isabelle/TLA [Mer95]. These object logics are normally careful to restrict the use of higher-order features to a minimum, however, the system itself does not restrict this usage.

The first author together with Gauthier [GK19] has previously proposed heuristics for automatically finding alignments across proof assistant libraries. Such alignments, even without merging the libraries can be useful for conjecturing new properties [MGK+17] as well as improving proof assistant automation [GK15].

The fact that Grothendieck universes are the same as transitive Tarski classes has been formalized by Carneiro in Metamath[3].

10 Automated Transfer and Limitations of Current Work

In this section, we discuss transfer in higher-order logic based systems, transport in intuitionistic type theory, and the limitations of the current work when it comes to automating the transfer of theorems between the foundations.

Automating the transfer of theorems between different types in higher-order logic has a long history. Today, higher-order rewriting-based packages for the creation of quotient types are present in the libraries of most HOL-based proof assistants. These packages can automatically translate theorems from the raw types to the quotient types.

For example, HOL Light [Har09] includes the `quot.ml` package already since the nineties. This package defines two ML functions: `lift_function` and `lift_theorem`. The former automatically defines constants (often of higher-order function types) in a quotient type based on corresponding constants in a raw type. The latter ML function uses higher-order rewriting to transfer theorems that use the lifted constants to raw ones.

The procedure has been further improved by Homeier [Hom05] in HOL4. The HOL4 quotient package allows an explicit declaration of properties of functions and relations (preserves and respects properties). These allow for quotients for polymorphic types. A similar architecture has been considered in the initial quotient package

[2] As part of an ongoing project to export Isabelle proof to Dedukti and the project exporting Isabelle to MMT [KRW20] some of the proofs in Isabelle/Main are being currently optimized.

[3] http://us.metamath.org/mpeuni/grutsk.html

for Isabelle/HOL co-developed by the first author [KU11]. By further considering the interplay between the transfer in the outside and inside types it is possible to automatically quotient lists into finite sets with operations such as concatenation of a list of lists automatically translated into a finite set union.

The Isabelle/HOL quotient package has been modularized by Huffman and Kunčar [HK13]. The functionality has been separated into two packages: `lifting` and `transfer`. Lifting allows the automated translation of definitions in a source type to definitions in a target type (including quotient-based definitions). Transfer uses higher-order rewriting to move theorems between types. This modular construction allows the use of transfer also for cases of isomorphic types (including almost isomorphic ones, as was already the case for example with quotients), but where the target is actually not defined as a quotient of the source type.

A further improvement to the transfer mechanism in Isabelle/HOL has been developed by Kuncar and Popescu [KP19b] in their work on local type definitions. There, the transfer package is extended to allow relativizing type-based statements to more set-based forms in a principled way.

In the context of intuitionistic type theory, translating theorems from types to their quotients is much more complex. This is because of the more intricate nature of equality in type theories, which in particular does not allow replacing equal things in all contexts (all above HOL packages rely not only on the axiom of choice but also on extensionality). An traditional approach to moving theorems between types that allows computation has been the use of setoids. This allows moving some theorems to quotients for example in the CoRN project [CFGW04].

More recently, foundations based on homotopy type theory [Awo12] have been proposed. There, propositional equality between terms is interpreted as homotopy. The univalence axiom of Voevodsky [Voe11] assumed in such foundations allows transporting properties and structures expressed over isomorphisms and equivalences. In its simplest variant, transport in HoTT/UF is an operation that takes a type family $P : A \to U$, a path $a = b$ in A, and returns a function $Pa \to Pb$ [Mör21]. This allows transport between isomorphic types but does not take computation into account. This is further extended in cubical type theories [CCHM17]. There, it is possible to directly manipulate n-dimensional cubes based on an interpretation of dependent type theory in a cubical set model. Cubical type theories furthermore are specified in a way that allows Voevodsky's axiom to be provable. Transport in cubical type theories [BCH19] can take as input a line of types $A : I \to U$. This more primitive transport operation can however take computation into account. We are not aware of any automated tactics/packages allowing for transport of theorems between types in the same way as it is possible in Isabelle/HOL's transfer package.

The work presented here, similar to the higher-order automated transfer packages, uses higher-order rewriting to translate the statements between the HOL types and the set-based representation, however, we have not been able to use the Isabelle transfer package for this. The reason for this is that on the Mizar side additional typing predicates are needed to express soft types and reasoning about these types is necessary. The Mizar soft types are additionally dependent. As such, we combine higher-order rewriting with our dedicated Isabelle/Mizar tactic for proving the Mizar type obligations (the mty tactic). As the tactic is responsible for Prolog-style type inference on the predicate level integrating its use with the existing Isabelle transfer package would be rather involved.

In principle, the equivalences provided by the isomorphisms allow translating the statements both in the assumptions and in the conclusions, however, we cannot directly use the transfer package, since type constraints not present on the term level in HOL correspond to explicit typing judgments in the set-theoretic types. Consider the isomorphism between the Mizar finite sequences and Isabelle/HOL lists. All the proved statements require the Mizar dependently typed assumptions stating that an argument is of a finite sequence type over some Mizar domain `l be FinSequence-of t` as well as an additional isomorphism for the domain. We have added the necessary assumptions to the theorems, and in the automated proofs, the Isabelle/Mizar type inference (including the automated proof of Mizar type inhabitation) is necessary to fulfill these obligations. We believe, that is it possible to augment the lifting and transfer packages to add soft type constraints on the term level and fulfill them wherever possible. The details are however unclear and are left as future work.

11 Conclusion

We have used Isabelle HOTG to combine results proved in TG set theory with results proved in higher-order logic. This allows us to combine large parts of two major proof assistant libraries: the Mizar Mathematical library and the Isabelle/HOL library. Supplementary to the theorems and proofs coming from both, we define a number of isomorphisms that allow us to translate theorems proved in part of one of these libraries and use them in the corresponding part of the other library.

As part of the library merging, we have formally defined and proved in Isabelle the necessary concepts. Apart from porting proofs to Isabelle/Mizar, the isomorphism formalizations and the theorems moved using those amount to 10179 lines of proofs. The formalization is available at:

`http://cl-informatik.uibk.ac.at/cek/ckkp-jar2022-hotg.tgz`

Apart from higher-order and set-theoretic foundations, the third most commonly used foundation is dependent type theory. The most important future work direction would investigate combining the results proved here with those proved in such type-theoretic foundations.

So far, we have mostly moved results that have been proved in HOL to set theory. It could be also interesting to transfer the Brouwer's theorem for n-dimensional case (the fixed point theorem [Pąk11], the topological invariance of degree, and the topological invariance of dimension [Pąk14a]) that are essential to define and develop topological manifolds since the Mizar library results on manifolds are much developed than those in Isabelle/HOL [IZ18].

Acknowledgements

This work has been supported by the European Research Council (ERC) Starting Grant Number 714034 *SMART*, the Polish National Science Center granted by decision n DEC-2015/19/D/ST6/01473, and the COST Action CA20111 Number E-COST-GRANT-CA20111-9d20b2ad.

References

AC15. Ali Assaf and Raphaël Cauderlier. Mixing HOL and Coq in Dedukti. In Cezary Kaliszyk and Andrei Paskevich, editors, *Proof eXchange for Theorem Proving (PxTP 2015)*, volume 186 of *EPTCS*, pages 89–96, 2015.

Ass15. Ali Assaf. *A framework for defining computational higher-order logics. (Un cadre de définition de logiques calculatoires d'ordre supérieur)*. PhD thesis, École Polytechnique, Palaiseau, France, 2015.

Awo12. Steven Awodey. Type theory and homotopy. In Peter Dybjer, Sten Lindström, Erik Palmgren, and Göran Sundholm, editors, *Epistemology versus Ontology - Essays on the Philosophy and Foundations of Mathematics in Honour of Per Martin-Löf*, volume 27 of *Logic, Epistemology, and the Unity of Science*, pages 183–201. Springer, 2012.

BBG+17. 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*, 2017.

BCH19. Marc Bezem, Thierry Coquand, and Simon Huber. The univalence axiom in cubical sets. *J. Autom. Reason.*, 63(2):159–171, 2019.

BHMN15. Jasmin Christian Blanchette, Maximilian Haslbeck, Daniel Matichuk, and Tobias Nipkow. Mining the Archive of Formal Proofs. In Manfred Kerber, Jacques Carette, Cezary Kaliszyk, Florian Rabe, and Volker Sorge, editors, *Intelligent Computer Mathematics (CICM 2015)*, volume 9150 of *LNCS*, pages 3–17. Springer, 2015.

BKP19. Chad Brown, Cezary Kaliszyk, and Karol Pąk. Higher-order Tarski Grothendieck as a foundation for formal proof. In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *LIPIcs*, pages 9:1–9:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

BP19. Chad E. Brown and Karol Pąk. A tale of two set theories. In Cezary Kaliszyk, Edwin C. Brady, Andrea Kohlhase, and Claudio Sacerdoti Coen, editors, *Intelligent Computer Mathematics - 12th International Conference, CICM 2019, Prague, Czech Republic, July 8-12, 2019, Proceedings*, volume 11617 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2019.

BR02. Grzegorz Bancerek and Piotr Rudnicki. A Compendium of Continuous Lattices in MIZAR. *J. Autom. Reasoning*, 29(3-4):189–224, 2002.

Bro14. Chad E. Brown. *The Egal Manual*, 2014.

CCHM17. Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. *FLAP*, 4(10):3127–3170, 2017.

CFGW04. Luís Cruz-Filipe, Herman Geuvers, and Freek Wiedijk. C-corn, the constructive coq repository at nijmegen. In Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec, editors, *Mathematical Knowledge Management (MKM 2004)*, volume 3119 of *LNCS*, pages 88–103. Springer, 2004.

EHN20. Manuel Eberl, Max W. Haslbeck, and Tobias Nipkow. Verified analysis of random binary tree structures. *J. Autom. Reason.*, 64(5):879–910, 2020.

GK15. Thibault Gauthier and Cezary Kaliszyk. Sharing HOL4 and HOL Light proof knowledge. In Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov, editors, *20th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2015)*, volume 9450 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2015.

GK19. Thibault Gauthier and Cezary Kaliszyk. Aligning concepts across proof assistant libraries. *J. Symbolic Computation*, 90:89–123, 2019.

GKN15. Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. Four decades of Mizar. *Journal of Automated Reasoning*, 55(3):191–198, 2015.

Gor96. Michael Gordon. Set theory, higher order logic or both? In Joakim von Wright, Jim Grundy, and John Harrison, editors, *Theorem Proving in Higher Order Logics, TPHOLs'96*, volume 1125 of *LNCS*, pages 191–201. Springer, 1996.

HAB+17. Thomas C. Hales, Mark Adams, Gertrud Bauer, Tat Dat Dang, John Harrison, Le Truong Hoang, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Tat Thang Nguyen, Quang Truong Nguyen, Tobias Nipkow, Steven Obua, Joseph Pleso, Jason M. Rute, Alexey Solovyev, Thi Hoai An Ta, Nam Trung Tran, Thi Diep Trieu,

Josef Urban, Ky Vu, and Roland Zumkeller. A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi*, 5, 2017.

Har09. John Harrison. HOL light: An overview. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings*, volume 5674 of *Lecture Notes in Computer Science*, pages 60–66. Springer, 2009.

HHM+10. Thomas C. Hales, John Harrison, Sean McLaughlin, Tobias Nipkow, Steven Obua, and Roland Zumkeller. A revision of the proof of the kepler conjecture. *Discret. Comput. Geom.*, 44(1):1–34, 2010.

HK13. Brian Huffman and Ondřej Kunčar. Lifting and transfer: A modular design for quotients in Isabelle/HOL. In Georges Gonthier and Michael Norrish, editors, *Certified Programs and Proofs - Third International Conference, CPP 2013, Melbourne, VIC, Australia, December 11-13, 2013, Proceedings*, volume 8307 of *LNCS*, pages 131–146. Springer, 2013.

HN10. Florian Haftmann and Tobias Nipkow. Code generation via higher-order rewrite systems. In Matthias Blume, Naoki Kobayashi, and Germán Vidal, editors, *Functional and Logic Programming, 10th International Symposium, FLOPS 2010*, volume 6009 of *LNCS*, pages 103–117. Springer, 2010.

Hom05. Peter V. Homeier. A design structure for higher order quotients. In Joe Hurd and Thomas F. Melham, editors, *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005, Oxford, UK, August 22-25, 2005, Proceedings*, volume 3603 of *Lecture Notes in Computer Science*, pages 130–146. Springer, 2005.

HW07. Florian Haftmann and Makarius Wenzel. Constructive type classes in Isabelle. In Thorsten Altenkirch and Conor McBride, editors, *Types for Proofs and Programs, International Workshop, TYPES 2006*, volume 4502 of *LNCS*, pages 160–174. Springer, 2007.

IZ18. Fabian Immler and Bohua Zhan. Smooth manifolds. *Archive of Formal Proofs*, October 2018. `https://isa-afp.org/entries/Smooth_Manifolds.html`, Formal proof development.

Jaś34. Stanisław Jaśkowski. On the rules of suppositions. *Studia Logica*, 1, 1934.

KP17. Cezary Kaliszyk and Karol Pąk. Isabelle formalization of set theoretic structures and set comprehensions. In Johannes Blamer, Temur Kutsia, and Dimitris Simos, editors, *Mathematical Aspects of Computer and Information Sciences, MACIS 2017*, volume 10693 of *LNCS*. Springer, 2017.

KP18. Cezary Kaliszyk and Karol Pąk. Semantics of Mizar as an Isabelle object logic. *Journal of Automated Reasoning*, 2018.

KP19a. Cezary Kaliszyk and Karol Pąk. Declarative proof translation (short paper). In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *LIPIcs*, pages 35:1–35:7, 2019.

KP19b. Ondrej Kuncar and Andrei Popescu. From types to sets by local type definition in higher-order logic. *J. Autom. Reason.*, 62(2):237–260, 2019.

KPU16. Cezary Kaliszyk, Karol Pąk, and Josef Urban. Towards a Mizar environment for Isabelle: Foundations and language. In Jeremy Avigad and Adam Chlipala, editors, *Proc. 5th Conference on Certified Programs and Proofs (CPP 2016)*, pages 58–65. ACM, 2016.

KRW20. Michael Kohlhase, Florian Rabe, and Makarius Wenzel. Making isabelle content accessible in knowledge representation formats. *CoRR*, abs/2005.08884, 2020.

KS10. Alexander Krauss and Andreas Schropp. A mechanized translation from higher-order logic to set theory. In Matt Kaufmann and Lawrence C. Paulson, editors, *Interactive Theorem Proving (ITP 2010)*, volume 6172 of *LNCS*, pages 323–338. Springer, 2010.

KU11. Cezary Kaliszyk and Christian Urban. Quotients revisited for Isabelle/HOL. In William C. Chu, W. Eric Wong, Mathew J. Palakal, and Chih-Cheng Hung, editors, *Proc. of the 26th ACM Symposium on Applied Computing (SAC'11)*, pages 1639–1644. ACM, 2011.

Kun10. Ondřej Kunčar. Reconstruction of the Mizar type system in the HOL Light system. In Jiri Pavlu and Jana Safrankova, editors, *WDS Proceedings of Contributed Papers: Part I – Mathematics and Computer Sciences*, pages 7–12. Matfyzpress, 2010.

Lam19. Peter Lammich. Refinement to imperative HOL. *J. Autom. Reason.*, 62(4):481–503, 2019.

LSBM19. Andreas Lochbihler, S. Reza Sefidgar, David A. Basin, and Ueli Maurer. Formalizing constructive cryptography using crypthol. In *32nd IEEE Computer Security Foundations Symposium, CSF 2019, Hoboken, NJ, USA, June 25-28, 2019*, pages 152–166. IEEE, 2019.

Mer95. Stephan Merz. Mechanizing TLA in Isabelle. In Robert Rodošek, editor, *Workshop on Verification in New Orientations*, pages 54–74, Maribor, 1995. Univ. of Maribor.

MGK+17. Dennis Müller, Thibault Gauthier, Cezary Kaliszyk, Michael Kohlhase, and Florian Rabe. Classification of alignments between concepts of formal mathematical systems. In Herman Geuvers, Matthew England, Osman Hasan, Florian Rabe, and Olaf Teschke, editors, *10th International Conference on Intelligent Computer Mathematics (CICM'17)*, volume 10383 of *LNCS*, pages 83–98. Springer, 2017.

Mör21. Anders Mörtberg. Cubical methods in homotopy type theory and univalent foundations. *Math. Struct. Comput. Sci.*, 31(10):1147–1184, 2021.

NPW02. Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

Obu06. Steven Obua. Partizan games in Isabelle/HOLZF. In Kamel Barkaoui, Ana Cavalcanti, and Antonio Cerone, editors, *Theoretical Aspects of Computing - ICTAC 2006*, volume 4281 of *LNCS*, pages 272–286. Springer, 2006.

OFSA14. Steven Obua, Jacques D. Fleuriot, Phil Scott, and David Aspinall. ProofPeer: Collaborative theorem proving. *CoRR*, abs/1404.6186, 2014.

Pąk11. Karol Pąk. Brouwer Fixed Point Theorem in the General Case. *Formalized Mathematics*, 19(**3**):151–153, 2011.

Pąk14a. Karol Pąk. Brouwer Invariance of Domain Theorem. *Formalized Mathematics*, 22(**1**):21–28, 2014.

Pąk14b. Karol Pąk. Topological manifolds. *Formalized Mathematics*, 22(**2**):179–186, 2014.

Pąk20. Karol Pąk. Grothendieck universes. *Formalized Mathematics*, 28(**2**):211–215, 2020.

Pau90. Lawrence C. Paulson. Isabelle: The next 700 theorem provers. In Piergiorgio Odifreddi, editor, *Logic and Computer Science (1990)*, pages 361–386, 1990.

Pau93. Lawrence C. Paulson. Set theory for verification: I. From foundations to functions. *J. Autom. Reasoning*, 11(3):353–389, 1993.

Pit93. A. Pitts. The HOL logic. In M. J. C. Gordon and T. F. Melham, editors, *Introduction to HOL: a theorem proving environment for higher order logic*. Cambridge University Press, Cambridge, 1993.

Rab17. Florian Rabe. How to identify, translate and combine logics? *J. Log. Comput.*, 27(6):1753–1798, 2017.

Sch99. Christoph Schwarzweller. The ring of integers, Euclidean rings and modulo integers. *Formalized Mathematics*, 8(**1**):29–34, 1999.

Voe11. Vladimir Voevodsky. Univalent semantics of constructive type theories. In Jean-Pierre Jouannaud and Zhong Shao, editors, *Certified Programs and Proofs - First International Conference, CPP 2011, Kenting, Taiwan, December 7-9, 2011. Proceedings*, volume 7086 of *Lecture Notes in Computer Science*, page 70. Springer, 2011.

Wen21. Makarius Wenzel. *The Isabelle/Isar Reference Manual*, 2021.

WPN08. Makarius Wenzel, Lawrence C. Paulson, and Tobias Nipkow. The Isabelle framework. In Otmane Aït Mohamed, César A. Muñoz, and Sofiène Tahar, editors, *Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008*, volume 5170 of *LNCS*, pages 33–38. Springer, 2008.