Higher-order Tarski Grothendieck as a Foundation for Formal Proof

Chad E. Brown
Czech Technical University in Prague, Czech Republic

Cezary Kaliszyk[1]
University of Innsbruck, Austria
University of Warsaw, Poland
cezary.kaliszyk@uibk.ac.at

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

Abstract

We formally introduce a foundation for computer verified proofs based on higher-order Tarski-Grothendieck set theory. We show that this theory has a model if a 2-inaccessible cardinal exists. This assumption is the same as the one needed for a model of plain Tarski-Grothendieck set theory. The foundation allows the co-existence of proofs based on two major competing foundations for formal proofs: higher-order logic and TG set theory. We align two co-existing Isabelle libraries, Isabelle/HOL and Isabelle/Mizar, in a single foundation in the Isabelle logical framework. We do this by defining isomorphisms between the basic concepts, including integers, functions, lists, and algebraic structures that preserve the important operations. With this we can transfer theorems proved in higher-order logic to TG set theory and vice versa. We practically show this by formally transferring Lagrange's four-square theorem, Fermat 3-4, and other theorems between the foundations in the Isabelle framework.

2012 ACM Subject Classification General and reference → General literature; General and reference

Keywords and phrases model; higher-order; Tarski Grothendieck; proof foundation

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding *Chad E. Brown*: the European Research Council (ERC) grant nr. 649043 *AI4REASON*
Cezary Kaliszyk: European Research Council (ERC) grant no. 714034 *SMART*
Karol Pąk: the Polish National Science Center granted by decision n°DEC-2015/19/D/ST6/01473

1 Introduction

Various formal proof foundations combine higher-order logic with set theory [10, 23, 33, 34]. Such a combination offers a familiar mathematical foundation, while at the same time offering more powerful automation present in HOL. All the combinations have been presented without a model, even though models for the two separate foundations are well known and studied. In this paper we will give a model of such a combination and show some consequences of the existence of the model for practical formalizations.

Today the libraries of proof assistants based on the two separate foundations are among the largest proof libraries available. The library of higher-order logic based Isabelle/HOL [43] together with the Archive of Formal Proofs consist of more than 100,000 theorems [9], while the Mizar Mathematical Library (MML) [6, 15] based on set theory contains 59,000 theorems.

[1] Optional footnote, e.g. to mark corresponding author

A number of results in the libraries are incomparable, for example among the theorems present in Wiedjik's list of 100 important theorems to formalize Isabelle has 16 theorems not formalized in Mizar, while Mizar has 5 theorems absent in Isabelle (64 are formalized in both). The Mizar library includes results about lattice theory [7], topology, and manifolds [38] not present in the Isabelle library.

A model for the higher-order Tarski-Grothendieck allows merging the results in the two libraries. This merging will be performed mostly manually. The reason for this, is that definitions for isomorphic concepts may be quite different in the usual approaches in these system. Consider the real numbers. In the MML their definition is performed in multiple steps. First, natural numbers are introduced using the set-theoretic successor. Next, positive rationals are created by adding fractions as pairs of irreducible naturals $\langle n, k\rangle$ (with $k > 1$). Finally, Dedekind cuts are used to obtain positive reals. The Isabelle approach is fundamentally different. Natural numbers are a subtype of the axiomatic type of individuals. Pairs of naturals are quotiented into integers and rationals. Finally, Cauchy sequences of rationals grant reals. The differences in the construction also imply differences in their behaviours. Every Mizar natural number is also an integer or real, while in Isabelle coercions are required. It is similar when it comes to mathematical structures (used by over 70% of the Mizar library). Their semantics [21] in Mizar is close to partial functions specified on named fields, which enables for example inheritance and this is used to realize the main algebraic structures. Isabelle records are quite similar, but it is type classes that are used to express algebra.

We will propose a way to lift the merged elementary concepts to the more involved ones. By associating the Isabelle number 0 and the empty set and the corresponding successor operations, we will show a homomorphism between the set theoretic and higher-order natural numbers and later integers. We will show that this homomorphism preserves the basic operations, which will allow transporting basic number theorems, including Lagrange, and Bertrand, and cases of Fermat's last theorem.

We will also show that it is possible to show a mapping between the Isabelle type classes and the set theoretic structures corresponding basic algebra. This will allow merging the formalizations of groups and rings in the two libraries. This will again use some merged basic concepts, namely functions and binary operators. This brings us to Euclidean spaces where the Brouwer theorem for n-dimensional case (the fixed point theorem [36], the topological invariance of degree, and the topological invariance of dimension [37]), which are used to define topological manifolds.

The rest of the paper is structured as follows. In Section 2 we review the higher-order logic foundations used later. Section 3 gives an axiomatization of higher-order Tarski-Grothendieck (HOTG). We first define it in a higher-order setting and then relate to the actual proof assistants based on this foundation. Section 4 presents our model of HOTG. Next, in Section 5 we show the implications of the existence of the model for practical formalization: we align the proof libraries of Isabelle/HOL and Isabelle/Mizar by building isomorphisms between the various concepts present in these libraries and by translating theorems via the isomorphism. Section 6 discusses related work.

2 Preliminaries

We begin by reviewing the syntax and semantics of higher-order logic. The original presentation of higher-order logic using simple type theory was due to Church [12] with a corresponding notion of semantics due to Henkin [18] (with an important correction by Andrews [2]). We largely follow the notation and presentation style used in [5].

Let $\mathcal{B}$ be a set of base types. We use β to range over the types in $\mathcal{B}$. We next define *types* and use σ, τ to range over types. The set $\mathcal{T}$ of types is given by inductively extending $\mathcal{B}$ to include the type o (of truth values) and the type $\sigma \to \tau$ (of functions from σ to τ) for all $\sigma, \tau \in \mathcal{T}$. We assume $o \notin \mathcal{B}$ and that types are freely generated.

For each type σ let $\mathcal{V}_\sigma$ be a countable set of variables of type σ, where we assume $\mathcal{V}_\sigma \cap \mathcal{V}_\tau = \emptyset$ whenever $\sigma \neq \tau$. We use x, y, z to range over variables. For each type σ let $\mathcal{C}_\sigma$ be a set of constants of type σ, where again $\mathcal{C}_\sigma \cap \mathcal{C}_\tau = \emptyset$ whenever $\sigma \neq \tau$. Furthermore, we assume $\mathcal{V}_\sigma \cap \mathcal{C}_\tau = \emptyset$. We use c, d to range over constants. A *name* is either a variable or a constant. We use ν to range over names.

We now inductively define a family of sets Λ_σ of *terms*, using s, t, u to range over terms. For the base cases, $\mathcal{V}_\sigma \subseteq \Lambda_\sigma$ and $\mathcal{C}_\sigma \subseteq \Lambda_\sigma$. There are two inductive cases: application and abstraction. If $s \in \Lambda_{\sigma\to\tau}$ and $t \in \Lambda_\sigma$, then $(st) \in \Lambda_\tau$. If $x \in \mathcal{V}_\sigma$ and $t \in \Lambda_\tau$, then $(\lambda x.t) \in \Lambda_{\sigma\to\tau}$. We often omit parenthesis with the convention that application associates to the left, so that stu means $((st)u)$. Terms of type o are also called *formulas*.

We insist on the inclusion of certain constants called *logical constants* in the family $\mathcal{C}$ of constants. For simplicity of presentation, we take every logical constant we will use as a constant. In particular, we assume:

- $\neg$ is a logical constant in $\mathcal{C}_{o\to o}$. We write $\neg(st)$ as $\neg st$.
- $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$ are logical constants in $\mathcal{C}_{o\to o\to o}$. We use infix notation for $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$, with priority in this order, and each one associating to the right.
- For each type σ Π_σ and Σ_σ are a logical constants in $\mathcal{C}_{(\sigma\to o)\to o}$. We write $\Pi_\sigma(\lambda x_1.\cdots\Pi_\sigma(\lambda x_n.t))$ as $\forall x_1 \cdots x_n : \sigma.t$ and write $\Sigma_\sigma(\lambda x_1.\cdots\Sigma_\sigma(\lambda x_n.t))$ as $\exists x_1 \cdots x_n : \sigma.t$
- For each type σ $=_\sigma$ is a logical constant in $\mathcal{C}_{\sigma\to\sigma\to o}$. We write $=_\sigma\ s\ t$ in infix as $s = t$.
- For each type σ ε_σ is a logical constant in $\mathcal{C}_{(\sigma\to o)\to\sigma}$.

It is well-known that smaller sets of logical constants would be sufficient. For example, it is known that in (extensional) higher-order logic equality is sufficient to define the propositional constants and connectives as well as the existential and universal quantifiers at each type [1].

We next turn to a review of Henkin semantics for our language [18] closely following the presentation style in [5]. A *frame* is a family $\mathcal{D}_\sigma$ of nonempty sets such that $\mathcal{D}_o = \{0, 1\}$ and $\mathcal{D}_{\sigma\to\tau} \subseteq (\mathcal{D}_\tau)^{\mathcal{D}_\sigma}$ for each $\sigma, \tau \in \mathcal{T}$. A frame is called *standard* if $\mathcal{D}_{\sigma\to\tau} = (\mathcal{D}_\tau)^{\mathcal{D}_\sigma}$ for every $\sigma, \tau \in \mathcal{T}$. An *assignment* is a function $\mathcal{I}$ mapping every name of type σ to an element in $\mathcal{D}_\sigma$. Given a variable $x \in \mathcal{V}_\sigma$ and element $a \in \mathcal{D}_\sigma$ let $\mathcal{I}^x_a$ be the assignment agreeing with $\mathcal{I}$ except possibly on x where $\mathcal{I}^x_a(x) = a$. An assignment $\mathcal{I}$ is *logical* if for each $\sigma \in \mathcal{T}$ the following conditions hold:

- for $a \in \mathcal{D}_o$ $\mathcal{I}(\neg)(a) = 1$ if and only if $a = 0$,
- for $a, b \in \mathcal{D}_o$ $\mathcal{I}(\wedge)(a)(b) = 1$ if and only if $a = 1$ and $b = 1$,
- for $a, b \in \mathcal{D}_o$ $\mathcal{I}(\vee)(a)(b) = 1$ if and only if $a = 1$ or $b = 1$,
- for $a, b \in \mathcal{D}_o$ $\mathcal{I}(\Rightarrow)(a)(b) = 1$ if and only if $a = 0$ or $b = 1$,
- for $a, b \in \mathcal{D}_o$ $\mathcal{I}(\Leftrightarrow)(a)(b) = 1$ if and only if $a = b$,
- for $f \in \mathcal{D}_{\sigma\to o}$ $\mathcal{I}(\Pi_\sigma)(f) = 1$ if and only if $f(a) = 1$ for all $a \in \mathcal{D}_\sigma$,
- for $f \in \mathcal{D}_{\sigma\to o}$ $\mathcal{I}(\Sigma_\sigma)(f) = 1$ if and only if there is some $a \in \mathcal{D}_\sigma$ such that $f(a) = 1$,
- for $a, b \in \mathcal{D}_\sigma$ $\mathcal{I}(=_\sigma)(a)(b) = 1$ if and only if $a = b$, and
- for $f \in \mathcal{D}_{\sigma\to o}$ $f(\mathcal{I}(\varepsilon_\sigma)(f)) = 1$ if and only if there is some $a \in \mathcal{D}_\sigma$ such that $f(a) = 1$.

In other words, $\mathcal{I}$ is logical if it interprets the logical constants appropriately.

We lift an assignment $\mathcal{I}$ to be a partial function $\hat{\mathcal{I}}$ on terms as follows:

- For names ν, $\hat{\mathcal{I}}(\nu) = \mathcal{I}(\nu)$.
- For $s \in \Lambda_{\sigma\to\tau}$ and $t \in \Lambda_\sigma$, $\hat{\mathcal{I}}(st) = f(a)$ if $\hat{\mathcal{I}}(s) = f \in \mathcal{D}_{\sigma\to\tau}$ and $\hat{\mathcal{I}}(t) = a \in \mathcal{D}_\sigma$.

- For $x \in \mathcal{V}_\sigma$ and $t \in \Lambda_\tau$, $\hat{\mathcal{I}}(\lambda x.t) = f$ if $f \in \mathcal{D}_{\sigma\to\tau}$ and $\widehat{\mathcal{I}_a^x}(t) = f(a)$ for all $a \in \mathcal{D}_\sigma$.

Note that for all $s \in \Lambda_\sigma$ if $\hat{\mathcal{I}}(s)$ is defined, then $\hat{\mathcal{I}}(s) \in \mathcal{D}_\sigma$. If $\hat{\mathcal{I}}$ is a total function with domain $\bigcup_{\sigma\in\mathcal{T}}$, then $\mathcal{I}$ is called an *interpretation*.

A *(Henkin) model* is a pair $(\mathcal{D}, \mathcal{I})$ where $\mathcal{D}$ is a frame and $\mathcal{I}$ is a logical interpretation. A model is called *standard* if the frame is standard. We say $(\mathcal{D}, \mathcal{I})$ satisfies a formula s if $\hat{\mathcal{I}}(s) = 1$ and say $(\mathcal{D}, \mathcal{I})$ is a model for a set $\mathcal{A}$ of formulas if $(\mathcal{D}, \mathcal{I})$ satisfies every $s \in \mathcal{A}$.

To simplify the presentation above, some dependencies were left implicit. For each set $\mathcal{B}$ of base types (with $o \notin \mathcal{B}$), we obtain a set $\mathcal{T}^{\mathcal{B}}$ of types. Additionally, for each set $\mathcal{B}$ of base types and each family $\mathcal{C}$ of constants indexed by $\mathcal{T}^{\mathcal{B}}$, we obtain a family $\Lambda^{\mathcal{B},\mathcal{C}}$ of terms. The definition of a frame above technically depends on the set $\mathcal{B}$ of base types and we say $\mathcal{D}$ is a *frame over* $\mathcal{B}$ when this dependency needs to be explicit. Futhermore an assignment depends on both $\mathcal{B}$ and $\mathcal{C}$ and we say $\mathcal{I}$ is an *assignment over* $\mathcal{B}$ *for* $\mathcal{C}$ when these dependencies need to be explicit.

A *theory* is a triple $(\mathcal{B}, \mathcal{C}, \mathcal{A})$ where $\mathcal{B}$ is a set of base types, $\mathcal{C}$ is a family of sets of constants (which must include the logical constants) over the types $\mathcal{T}^{\mathcal{B}}$ and $\mathcal{A} \subseteq \Lambda_o^{\mathcal{B},\mathcal{C}}$ is a set of formulas called the *axioms* of the theory. A pair $(\mathcal{D}, \mathcal{I})$ is a *model of a theory* $(\mathcal{B}, \mathcal{C}, \mathcal{A})$ if $\mathcal{D}$ is a frame over $\mathcal{B}$, $\mathcal{I}$ is a logical interpretation over $\mathcal{B}$ for $\mathcal{C}$ and $(\mathcal{D}, \mathcal{I})$ is a model of the set $\mathcal{A}$ of formulas.

It is known that the notion of a Henkin model provides a sound and complete semantics for a variety of proof calculi [5, 8, 11]. Our concern in this article is not with proof calculi directly, but with consistency of certain axiom sets for higher-order set theory. In this paper we will only consider one axiomatization of higher-order Tarski Grothendieck set theory. Soundness implies it is sufficient to find models of these axiom sets to infer consistency, and for this purpose constructing a standard model is enough. In future work we plan to consider different axiomatizations of higher-order Tarski Grothendieck (e.g., the one in [23]) and plan to use soundness and completeness with respect to Henkin models to prove the two versions of Tarski Grothendieck are equivalent.

3 An Axiomatization of Higher-Order Tarski Grothendieck

In this section we give a formulation of higher-order Tarski Grothendieck (HOTG) set theory by giving a theory **HOTG**. The theory is identical to the one implemented by the first author in the Egal system [10] and specified an operator that explicitly gives the Grothenieck universe of a set [16]. In the presence of the axiom of choice, this is equivalent to specifying that such a universe exists for every set, which is the approach used in the Mizar system as specified by Trybulec [42]. In the below axiomatization and in the model in the next section, we will use the explicit universe operation, as it makes the presentation simpler, but our intention is to use it both for explicit universes and implicit ones, as specified in Isabelle/Mizar by Kaliszyk and Pąk [23] using Tarski's Axiom A [41] and used in Section 5.

We first describe the theory **HOTG** as given by the triple $(\mathcal{B}, \mathcal{C}, \mathcal{A})$. Here $\mathcal{B}$ be the singleton $\{\iota\}$ and the base type ι is intended to be the type of sets. The typed constants $\mathcal{C}$ consists precisely of the logical constants and the following additional constants:

- In in $\mathcal{C}_{\iota\to\iota\to o}$. We write $\mathsf{In}\ s\ t$ in infix as $s \in t$.
- Empty in $\mathcal{C}_\iota$.
- Un in $\mathcal{C}_{\iota\to\iota}$.
- Pow in $\mathcal{C}_{\iota\to\iota}$.
- Repl in $\mathcal{C}_{\iota\to(\iota\to\iota)\to\iota}$.
- Univ in $\mathcal{C}_{\iota\to\iota}$.

To state the axioms, we will use three abbreviations. Let Subq be the term

$$\lambda X.\lambda Y.\forall z:\iota.z \in X \Rightarrow z \in Y$$

of type $\iota \to \iota \to o$. We write $\mathsf{Subq}\ s\ t$ as $s \subseteq t$. Let $\mathsf{TransSet}$ be the term

$$\lambda U.\forall X:\iota.X \in U \Rightarrow X \subseteq U$$

of type $\iota \to o$. Let $\mathsf{ZFclosed}$ be the term

$$\begin{aligned}\lambda U.\ & (\forall X:\iota.X \in U \Rightarrow \mathsf{Un}\ X \in U) \wedge (\forall X:\iota.X \in U \Rightarrow \mathsf{Pow}\ X \in U) \\ & \wedge(\forall X:\iota.\forall F:\iota\to\iota.X \in U \Rightarrow (\forall x:\iota.x \in X \Rightarrow F\ x \in U) \Rightarrow \mathsf{Repl}\ X\ F \in U)\end{aligned}$$

of type $\iota \to o$.

The set $\mathcal{A}$ of axioms consists of the following formulas:

Extensionality: $\forall XY:\iota.X \subseteq Y \to Y \subseteq X \to X = Y$.

$\in$-Induction: $\forall P:\iota \to o.(\forall X:\iota.(\forall x:\iota.x \in X \to Px) \to PX) \to \forall X:\iota.PX$.

Empty: $\neg\exists x:\iota.x \in \mathsf{Empty}$.

Union: $\forall X:\iota.\forall x:\iota.x \in \mathsf{Un}\ X \Leftrightarrow \exists Y:\iota.x \in Y \wedge Y \in X$.

Power: $\forall XY:\iota.Y \in \mathsf{Pow}\ X \Leftrightarrow Y \subseteq X$.

Replacement: $\forall X:\iota.\forall F:\iota \to \iota.\forall y:\iota.y \in \mathsf{Repl}\ X\ F \Leftrightarrow \exists x:\iota.x \in X \wedge y = Fx$.

UnivIn: $\forall N:\iota.N \in \mathsf{Univ}N$

UnivTransSet: $\forall N:\iota.\mathsf{TransSet}\ (\mathsf{Univ}N)$.

UnivZF: $\forall N:\iota.\mathsf{ZFclosed}\ (\mathsf{Univ}N)$.

UnivMin: $\forall N\,U:\iota.N \in U \to \mathsf{TransSet}\ U \to \mathsf{ZFclosed}\ U \to \mathsf{Univ}N \subseteq U$.

4 A Model of Higher-Order Set Theory

We will make heavy use of the von Neumann hierarchy (see for example [27]). By ordinal induction we define the set V_α for ordinals α as $V_\emptyset = \emptyset$, $V_{\alpha+1} = \wp(V_\alpha)$ and $V_\lambda = \bigcup_{\alpha<\lambda} V_\alpha$. Since we work in a well-founded set theory, for every set X there is some ordinal α such that $X \subseteq V_\alpha$ (and so $X \in V_{\alpha+1}$).

A cardinal κ is *inaccessible* if it is regular and $\lambda < \kappa$ implies $2^\lambda < \kappa$. A cardinal κ is *2-inaccessible* if it is a regular limit of inaccessible cardinals. Note that if κ is *2-inaccessible*, then for every $\lambda < \kappa$ there is some inaccessible κ' with $\lambda < \kappa' < \kappa$. It easily follows every 2-inaccessible is also inaccessible.

The following proposition can be found in Kanamori (see Proposition 2.1 in [26]).

▶ **Proposition 1.** *Let κ be inaccessible.*

1. *$x \subseteq V_\kappa$ implies $x \in V_\kappa$ iff $|x| < \kappa$.*
2. *$V_\kappa \models$ ZFC*

We define universes following Grothendieck [16].

▶ **Definition 2.** *Let U be a set. We say U is a* universe *if four conditions hold:*

- *U is transitive.*
- *If $x, y \in U$, then $\{x, y\} \in U$.*
- *If $X \in U$, then $\wp(X) \in U$.*
- *If $I \in U$ and $X_i \in U$ for each $i \in I$, then $\bigcup_{i\in I} X_i \in U$.*

The fact that every inaccessible yields a universe follows easily from Proposition 1.

▶ **Proposition 3.** *If κ is inaccessible, then V_κ is a universe.*

The following proposition will ensure that universes satisfy the properties in the definition of ZFclosed.

▶ **Proposition 4.** *Let U be a universe.*
1. *If $X \in U$, then $\bigcup X \in U$.*
2. *If $X \in U$ and $f : X \to U$, then $\{f(x)|x \in X\} \in U$.*

Proof. Suppose $X \in U$. We know $\bigcup X \in U$ since $\bigcup X = \bigcup_{x\in X}\{x\}$. Now suppose $X \in U$ and $f : X \to U$. We know $\{f(x)|x \in X\} \in U$ since $\{f(x)|x \in X\} = \bigcup_{x\in X}\{f(x)\}$. ◀

To interpret the constant Univ we will not only need universes, but a global function uniformly giving the least universe containing a given set.

▶ **Definition 5.** *Let $\alpha > 0$ be an ordinal. A* universe function for α *is a function $\mathcal{U} : V_\alpha \to V_\alpha$ such that for all $A \in V_\alpha$ we have $A \in \mathcal{U}(A)$, $\mathcal{U}(A)$ is a universe and $\mathcal{U}(A) \subseteq U$ for all universes $U \in V_\alpha$ with $A \in U$.*

▶ **Definition 6.** *Let $\alpha > 0$ be an ordinal and $\mathcal{U}$ be a universe function for α. Let $\mathcal{D}^\alpha_\iota$ be V_α, $\mathcal{D}^\alpha_o = \{0, 1\}$ and $\mathcal{D}^\alpha_{\sigma\to\tau} = (\mathcal{D}^\alpha_\tau)^{\mathcal{D}^\alpha_\sigma}$ for each $\sigma, \tau \in \mathcal{T}^{\mathcal{B}}$. Note that $V_\alpha \neq \emptyset$ since $\alpha > 0$ and so $\mathcal{D}^\alpha$ is a standard frame over $\mathcal{B}$. We call $\mathcal{D}^\alpha$ the* standard set-theoretic frame for α. *An assignment $\mathcal{I}$ over $\mathcal{B}$ for $\mathcal{C}$ into $\mathcal{D}^\alpha$ is called a* standard set-theoretic interpretation for α and $\mathcal{U}$ *if $\mathcal{I}$ is a logical interpretation and the following properties hold:*
- $\mathcal{I}(\mathsf{In})(a)(A) = 1$ *if and only if $a \in A$ for $a, A \in \mathcal{D}^\alpha_\iota$.*
- $\mathcal{I}(\mathsf{Empty}) = \emptyset$
- $\mathcal{I}(\mathsf{Un})(A) = \bigcup A$ *for every $A \in \mathcal{D}^\alpha_\iota$.*
- $\mathcal{I}(\mathsf{Pow})(A) = \wp(A)$ *for every $A \in \mathcal{D}^\alpha_\iota$.*
- $\mathcal{I}(\mathsf{Repl})(A)(f) = \{f(a)|a \in A\}$ *for every $A \in \mathcal{D}^\alpha_\iota$ and $f \in \mathcal{D}^\alpha_{\iota\to\iota}$.*
- $\mathcal{I}(\mathsf{Univ}) = \mathcal{U}$.

▶ **Theorem 7.** *Let $\alpha > 0$ be an ordinal, $\mathcal{U}$ be a universe function for α and $\mathcal{D}^\alpha$ be the standard set-theoretic frame for α. If $\mathcal{I}$ is a standard set-theoretic interpretation for α and $\mathcal{U}$, then $(\mathcal{D}^\alpha, \mathcal{I})$ is a model of the theory* **HOTG**.

Proof. Assume $\mathcal{I}$ is a standard set-theoretic interpretation for α and $\mathcal{U}$. We only need to prove $\mathcal{I}$ maps every formula in $\mathcal{A}$ to 1.

Extensionality: The fact that

$$\mathcal{I}(\forall XY : \iota.X \subseteq Y \to Y \subseteq X \to X = Y) = 1$$

follows easily from the fact that $A = B$ whenever $A \subseteq B$ and $B \subseteq A$ for $A, B \in V_\alpha$.

$\in$-Induction: In order to prove

$$\mathcal{I}(\forall P : \iota \to o.(\forall X : \iota.(\forall x : \iota.x \in X \to Px) \to PX) \to \forall X : \iota.PX) = 1$$

it suffices to prove that $C = V_\alpha$ for every $C \subseteq V_\alpha$ such that $A \in C$ for every $A \in V_\alpha$ with $A \subseteq C$. Let $C \subseteq V_\alpha$ be given and assume $A \in C$ for every $A \in V_\alpha$ with $A \subseteq C$. Consider $V_\alpha \setminus C$. Assume $V_\alpha \neq C$. In this case $V_\alpha \setminus C$ must be nonempty. By regularity there is an element $A \in V_\alpha \setminus C$ such that $A \cap (V_\alpha \setminus C) = \emptyset$. Since V_α is transitive $A \subseteq V_\alpha$ and so $A \cap (V_\alpha \setminus C) = \emptyset$ implies $A \subseteq C$. By our assumption about C, we must have $A \in C$, contradicting $A \in V_\alpha \setminus C$.

Empty: We know $\mathcal{I}(\neg\exists x : \iota.x \in \mathsf{Empty}) = 1$ since $\mathcal{I}(\mathsf{Empty}) = \emptyset$.

Union: We know $\mathcal{I}(\forall X : \iota.\forall x : \iota.x \in \mathsf{Un}\ X \Leftrightarrow \exists Y : \iota.x \in Y \wedge Y \in X) = 1$ since $\mathcal{I}(\mathsf{Un})(A) = \bigcup A$.

Power: We know $\mathcal{I}(\forall XY : \iota.Y \in \mathsf{Pow}\ X \Leftrightarrow Y \subseteq X) = 1$ since $\mathcal{I}(\mathsf{Pow})(A) = \wp A$.

Replacement: We can easily prove $\mathcal{I}(\forall X : \iota.\forall F : \iota \to \iota.\forall y : \iota.y \in \mathsf{Repl}\ X\ F \Leftrightarrow \exists x : \iota.x \in X \wedge y = Fx) = 1$ using the fact that $\mathcal{I}(\mathsf{Repl})(A)(f) = \{f(a)|a \in A\}$ for every $A \in V_\alpha$ and every $f : V_\alpha \to V_\alpha$.

UnivIn: Since $\mathcal{U}$ is a universe function we know $A \in \mathcal{U}(A)$ for every $A \in V_\alpha$. Hence $\mathcal{I}(\forall N : \iota.N \in \mathsf{Univ}N) = 1$.

UnivTransSet: Since $\mathcal{U}$ is a universe function, $\mathcal{U}(A)$ is a universe (and hence transitive) for every $A \in V_\alpha$. Hence $\mathcal{I}(\forall N : \iota.\mathsf{TransSet}\ (\mathsf{Univ}N)) = 1$.

UnivZF: It is easy to see $\mathcal{I}(\forall N : \iota.\mathsf{ZFclosed}\ (\mathsf{Univ}N)) = 1$ using Definitions 2 and 5 and Proposition 4.

UnivMin: Suppose $A, U \in V_\alpha$ where $A \in U$, U is transitive and $\mathcal{I}(\mathsf{ZFclosed})(U) = 1$. We argue that U is a universe. We know U is transitive. The fact that $\wp(X) \in U$ whenever $X \in U$ follows directly from $\mathcal{I}(\mathsf{ZFclosed})(U) = 1$. In particular, since $A \in U$, we know $\wp(A) \in U$ and $\wp(\wp(A)) \in U$. Let $x, y \in U$ be given. Let $f : \wp(\wp(A)) \to U$ be the function

$$f(X) = \begin{cases} x & \text{if } A \in X \\ y & \text{otherwise} \end{cases}$$

Since $f(A) = x$ and $f(\emptyset) = y$, we know $\{x, y\} = \{f(X)|X \in \wp(\wp(A))\}$. Using $\mathcal{I}(\mathsf{ZFclosed})(U) = 1$ we conclude $\{x, y\} \in U$. Now let $I \in U$ and a family $X_i \in U$ for each $i \in I$ be given. Let $g : I \to U$ be the function $g(i) = X_i$. Using $\mathcal{I}(\mathsf{ZFclosed})(U) = 1$ we know $\{g(i)|i \in I\} \in U$ and then $\bigcup_{i \in I} X_i = \bigcup\{g(i)|i \in I\} \in U$. Hence U is a universe. Since U is a universe with $A \in U$, we conclude $\mathcal{U}(A) \subseteq U$ from Definition 5.

◀

For a general ordinal α there will be no universe function $\mathcal{U}$. For 2-inaccessible cardinals there is a universe function and a corresponding standard set-theoretic interpretation.

▶ **Theorem 8.** *Let κ be 2-inaccessible and $\mathcal{D}^\kappa$ be the standard set-theoretic frame for κ. There is a universe function $\mathcal{U}$ for κ and there is a standard set-theoretic interpretation $\mathcal{I}$ for κ and $\mathcal{U}$.*

Proof. We first construct the universe function. For each $A \in V_\kappa$, let $A' = \{U \in V_\kappa | U \text{ is a universe and } A \in U\}$. We argue A' is always nonempty. Since $A \in V_\kappa$ there must be some $\alpha < \kappa$ such that $A \in V_\alpha$. Since κ is 2-inaccessible there must be some inaccessible $\kappa' < \kappa$ with $\alpha < \kappa'$. By Proposition 3 $V_{\kappa'}$ is a universe and so $V_{\kappa'} \in A'$. Since A' is a nonempty set, $\bigcap A'$ is well-defined and we can take $\mathcal{U}(A)$ to be $\bigcap A'$. A simple inspection of Definition 2 reveals that the intersection of a nonempty set of universes is itself a universe. Thus $\mathcal{U}(A)$ is the least universe with A as a member and $\mathcal{U}$ is a universe function for κ.

Next we turn to the interpretation $\mathcal{I}$. The axiom of choice states that there is a function $\mathfrak{c} : \wp(V_{\kappa+\omega}) \setminus \{\emptyset\} \to V_{\kappa+\omega}$ such that $\mathfrak{c}(A) \in A$ for every $A \in \wp(V_{\kappa+\omega}) \setminus \{\emptyset\}$. An easy induction on types shows $\mathcal{D}^\kappa_\sigma \in V_{\kappa+\omega}$ for each $\sigma \in \mathcal{T}^\mathcal{B}$. Hence $\mathcal{D}^\kappa_\sigma \in \wp(V_{\kappa+\omega}) \setminus \{\emptyset\}$ for each $\sigma \in \mathcal{T}^\mathcal{B}$ since $V_{\kappa+\omega}$ is transitive. We can simply define $\mathcal{I}(x) = \mathfrak{c}(\mathcal{D}^\kappa_\sigma) \in \mathcal{D}^\kappa_\sigma$ for each variable $x \in \mathcal{V}_\sigma$. For the logical constants c other than ε_σ we take the obvious value $\mathcal{I}(c)$ so that $\mathcal{I}$ will be a logical interpretation. In each case this value is in $\mathcal{D}^\kappa_\sigma$ since $\mathcal{D}^\kappa$ is a standard frame. We take $\mathcal{I}(\varepsilon_\sigma)$ to be the function $g \in \mathcal{D}^\kappa_{(\sigma\to o)\to\sigma}$ such that for $f \in \mathcal{D}^\kappa_{\sigma\to o}$ we have

$$g(f) = \begin{cases} \mathfrak{c}(\{a \in \mathcal{D}^\kappa_\sigma | f(a) = 1\}) & \text{if } f(a) = 1 \text{ for some } a \in \mathcal{D}^\kappa_\sigma \\ \mathfrak{c}(\mathcal{D}^\kappa_\sigma) & \text{otherwise.} \end{cases}$$

It only remains to give values $\mathcal{I}(c)$ for the nonlogical constants in $\mathcal{C}$. For In, Empty, Un, Pow and Repl there is at most one corresponding value that might possibly satisfy the conditions in Definition 6. Since we know $\mathcal{D}^{\kappa}_{\iota} = V_\kappa$ is a universe, each of these values is in $\mathcal{D}^{\kappa}_{\sigma}$ in each respective case. Finally we take $\mathcal{I}(\mathsf{Univ})$ to be the universe function $\mathcal{U}$ constructed above. By the choice of $\mathcal{I}$ it is easy to see that $\mathcal{I}$ is a standard set-theoretic interpretation for κ. ◀

As an easy corollary of Theorems 7 and 8 we have the following relative satisfiability result.

▶ **Theorem 9.** *If there is a 2-inaccessible cardinal, then* **HOTG** *is satisfiable.*

5 Proof Integration

The model defined in the previous section allows us to use the higher-order library and set theoretic library simultaneously. We will do this in the Isabelle logical framework, by importing various results from the two libraries in the same environment and define transfer methods between these results. This 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 this section have been formalized in Isabelle and will be presented close to the Isabelle notation. The Isabelle environment will import both Isabelle/HOL [32] and Isabelle/Mizar [23] object logics along with a number of results formalized in the standard libraries of the two. The notations will follow first-order style notations. In particular the symbols $=_{\mathcal{H}}$ and $=_{\mathcal{S}}$ will refer to the HOL and set-theoretic equality operations respectively. Finally *be* is the Mizar infix operator for specifying the type of a set in the Mizar intersection type system [24].

To combine two types we will first define bijections between these types. We will next show that the bijection preserves various constants and operators. This will allow us to transfer results using higher-order rewriting, in the style of quotient packages for HOL [19,25] and the Isabelle transfer package [20]. In the MML set theory it is common to reason both about the type of the natural numbers and the elements of the set of natural numbers. This is necessary, since the arguments of all operations must be sets, while at the same inference can be performed efficiently for types [6]. We therefore define operators for specifying bijections between HOL types and either set theoretic types and sets. The definitions are analogous and we show only the one for types. An isomorphism between X and Y will be defined as a pair of two morphisms $f : X \to Y$, $g : Y \to X$ such that $f \circ g = id_Y$ and $g \circ f = id_X$.

▶ **Definition 10.** *Let σ be type, $d \in \Lambda_\iota$ be set, function $\mathfrak{s}2\mathfrak{h} \in \Lambda_{\iota\to\sigma}$, function $\mathfrak{h}2\mathfrak{s} \in \Lambda_{\sigma\to\iota}$. The predicate $beIso_S\langle \mathfrak{h}2\mathfrak{s}, \mathfrak{s}2\mathfrak{h}, \sigma, d\rangle$ holds whenever all of the following hold:*

- $\forall x : \sigma.\mathfrak{s}2\mathfrak{h}(\mathfrak{h}2\mathfrak{s}(x)) = x$,
- $\forall x : \iota.x \in d \to \mathfrak{h}2\mathfrak{s}(\mathfrak{s}2\mathfrak{h}(x)) = x$,
- $\forall x : \iota.\mathfrak{h}2\mathfrak{s}(\mathfrak{s}2\mathfrak{h}(x)) \in \Lambda_\sigma$,
- $\forall x : \sigma.\mathfrak{s}2\mathfrak{h}(x) \in d$.

We will derive the above predicate to show an isomorphisms between a higher-order type and a set-theoretic type. The existence of a bijection does not immediately imply the inhabitation of the 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 have proofs in our formalization.

theorem *beIsoS_d*:
  beIsoS($\mathfrak{h}2\mathfrak{s}$,$\mathfrak{s}2\mathfrak{h}$,$d$) $\Longrightarrow$ *d is non empty*

5.1 Natural numbers and integers

The Isabelle/Mizar natural numbers are defined as the smallest limit ordinal. The existence of this set is a consequence of the Tarski universe property. The formal definition is as follows:

mdef *ordinal1_def_11* (*omega*) **where**
func omega $\rightarrow$ *set means* (λ*it.*
$0_{\mathcal{S}}$ *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*))

While Isabelle naturals are a subtype of the type of individuals. In order to merge these two different approaches we specified an isomorphisms that preserves zero and the successor. Note that the isomorphism 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.

$$\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(n) =_{\mathcal{S}} \begin{cases} 0_{\mathcal{S}} & \text{if } n =_{\mathcal{H}} 0_{\mathcal{H}}, \\ \mathrm{S}_{\mathcal{S}}(\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(k)) & \text{if } n =_{\mathcal{H}} \mathrm{S}_{\mathcal{H}}(k) \text{ for some } \mathcal{H}\text{-natural } k. \end{cases}$$

$$\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n) =_{\mathcal{H}} \begin{cases} 0_{\mathcal{H}} & \text{if } n =_{\mathcal{S}} 0_{\mathcal{S}}, \\ \mathrm{S}_{\mathcal{H}}(\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(k)) & \text{if } n =_{\mathcal{S}} \mathrm{S}_{\mathcal{S}}(k) \text{ for some } \mathcal{S}\text{-natural } k, \\ \textit{undefined} & \text{otherwise.} \end{cases}$$

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

fun *h2sn* :: *nat* $\Rightarrow$ *Set* ($\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(\_)$) **where**
$\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(0::nat) =_{\mathcal{S}} 0_{\mathcal{S}}$ | $\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(Suc(x)) =_{\mathcal{S}} succ\ \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x)$

function *s2hn* :: *Set* $\Rightarrow$ *nat* ($\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(\_)$) **where**
$\neg x$ *be Nat* $\Longrightarrow \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(x) =_{\mathcal{H}}$ *undefined*
| $\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(0_{\mathcal{S}}) =_{\mathcal{H}} 0$
| *x be Nat* $\Longrightarrow \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(succ(x)) =_{\mathcal{H}} Suc(\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(x))$

Note that $\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}$ is defined only on the HOL natural numbers, while $\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}$ 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 induction principles (both the HOL one and the Mizar one) we can show various properties of the isomorphism. In particular it gives a bijection (note the hidden type restriction to sets of type *nat*) and preserve the basic operations on the natural numbers including addition, multiplication, comparison operators, division, primality, etc. Excerpt of the statements proved in the Isabelle formalization is as follows:

theorem *Nat_to_Nat*:
fixes *x*::*nat* **and** *y*::*nat*
assumes *n be Nat* **and** *m be Nat*
shows $\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x +_{\mathcal{H}} y) =_{\mathcal{S}} \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x) +_{\mathcal{S}}{}^{\mathbb{N}}\ \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(y)$
$\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n +_{\mathcal{S}}{}^{\mathbb{N}}\ m) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n) +_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(m)$
$\mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x *_{\mathcal{H}} y) =_{\mathcal{S}} \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x) *_{\mathcal{S}}{}^{\mathbb{N}}\ \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(y)$
$\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n *_{\mathcal{S}}{}^{\mathbb{N}}\ m) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n) *_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(m)$
$x < y \longleftrightarrow \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x) \subset \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(y)$

$$
\begin{aligned}
n \subset m &\longleftrightarrow \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n) < \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(m)\\
x\ dvd\ y &\longleftrightarrow \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x)\ divides\ \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(y)\\
n\ divides\ m &\longleftrightarrow \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n)\ dvd\ \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(m)\\
prime(x) &\longleftrightarrow \mathfrak{h}2\mathfrak{s}_{\mathbb{N}}(x)\ is\ prime_{\mathcal{S}}\\
n\ is\ prime_{\mathcal{S}} &\longleftrightarrow prime(\mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n))
\end{aligned}
$$

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

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

Integers can be handled in an analogous way: the definitions are again different but it is straightforward to define a bijection between the two, and show that is preserves all the basic operators. For operators that are missing in one of the libraries, it is possible to actually lift their definitions. For example the exponentiation operation, which has not been considered in the Isabelle/Mizar library so far, can be defined as $TransformHS(\mathfrak{s}2\mathfrak{h}_{\mathbf{Z}},\mathfrak{s}2\mathfrak{h}_{\mathbb{N}},\mathfrak{h}2\mathfrak{s}_{\mathbf{Z}},(\hat{\ }))$, where

definition *TransformHS* **where**
$func\ TransformHS(\mathfrak{s}2\mathfrak{h}X1,\mathfrak{s}2\mathfrak{h}X2,\mathfrak{h}2\mathfrak{s}Y,HFun,x1,x2) \to set\ equals$
$\quad \mathfrak{h}2\mathfrak{s}Y(HFun(\mathfrak{s}2\mathfrak{h}X1(x1),\mathfrak{s}2\mathfrak{h}X2(x2)))$

This allows translating the proved Fermat's last theorem for powers divisible by 3 and 4 from Isabelle/HOL to Isabelle/Mizar. The proof involved quite some computation and therefore has not been attempted in Mizar so far.

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

5.2 Polymorphic types and lists

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. At the type of lists is a polymorphic type, 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* :: $(a \Rightarrow Set) \Rightarrow a\ List.list \Rightarrow Set\ (\mathfrak{h}2\mathfrak{s}_L(\_,\_))$**where**
$\quad \mathfrak{h}2\mathfrak{s}_L(\mathfrak{h}2\mathfrak{s},\ Nil) =_{\mathcal{S}} <*>$
$|\ \mathfrak{h}2\mathfrak{s}_L(\mathfrak{h}2\mathfrak{s},\ Cons(h,\ t)) =_{\mathcal{S}} ((<*\mathfrak{h}2\mathfrak{s}(h)*>)\ \hat{}\ (\mathfrak{h}2\mathfrak{s}_L(\mathfrak{h}2\mathfrak{s},\ t)))$

The converse operation needs to separate the first element of a sequence from the rest and shift it by one. 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 $\mathit{s2hl} :: (\mathit{Set} \Rightarrow a) \Rightarrow \mathit{Set} \Rightarrow a\ \mathit{List.list}\ (\mathfrak{s}2\mathfrak{h}_L(\_,\_))$ **where**
  $\neg\ x\ \mathit{be}\ \mathit{FinSequence} \Longrightarrow \mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},x) =_{\mathcal{H}} \mathit{undefined}$
  $|\ \mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},<\!*\!>) =_{\mathcal{H}}\ \mathit{Nil}$
  $|\ x\ \mathit{be}\ \mathit{FinSequence} \Longrightarrow x \neq <\!*\!> \Longrightarrow$
    $\mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},x) =_{\mathcal{H}} \mathit{Cons}\ (\mathfrak{s}2\mathfrak{h}(x.1_{\mathcal{S}}),\ \ \mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},x/\hat{\ }1_{\mathcal{S}}\ \ ))$

For the transformation introduced as 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 $p\ \mathit{be}\ \mathit{FinSequence}$ **and** $q\ \mathit{be}\ \mathit{FinSequence}$
    and $n\ \mathit{be}\ \mathit{Nat}$ **and** $n\ \mathit{in}\ \mathit{len}\ p$
  shows $\mathit{size}(\mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},p)) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(\mathit{len}\ p)$
      $\mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},p\hat{\ }q) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},p)\ @\ \mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},q)$
    $\mathfrak{s}2\mathfrak{h}_L(\mathfrak{s}2\mathfrak{h},p)\ !\ \mathfrak{s}2\mathfrak{h}_{\mathbb{N}}(n) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}(p.\ (\mathit{succ}\ n))$

One of the most general polymorphic types is the type of functions. We can naturally show that set theoretic functions (sets of pairs) correspond to higher-order function and that this homomorphism preserves function application.

theorem *HtoSappl*:
  assumes $\mathit{beIsoS}(\mathfrak{h}2\mathfrak{s}d,\mathfrak{s}2\mathfrak{h}d,d)$ **and** $\mathit{beIsoS}(\mathfrak{h}2\mathfrak{s}r,\mathfrak{s}2\mathfrak{h}r,r)$
  shows $\mathfrak{h}2\mathfrak{s}_f(\mathfrak{s}2\mathfrak{h}d,\mathfrak{h}2\mathfrak{s}r,d,f).\mathfrak{h}2\mathfrak{s}d(x) =_{\mathcal{S}} \mathfrak{h}2\mathfrak{s}r(f(x))$

5.3 Algebra

The structure representation used in higher-order logic and set theories 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 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 hint 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. 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* ($\mathfrak{h}2\mathfrak{s}_G(\_,\_,\_,\_)$) **where**
$\mathfrak{h}2\mathfrak{s}_G(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,c,g) =_{\mathcal{S}}$ [#
$\quad carrier \mapsto c;$
$\quad multF \mapsto \mathfrak{h}2\mathfrak{s}_{BinOp}(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,c,mult(g))$ #]

definition *s2hg* ($\mathfrak{s}2\mathfrak{h}_G(\_,\_,\_)$) **where**
$\mathfrak{s}2\mathfrak{h}_G(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,g) =_{\mathcal{H}} Igroup($
$\quad Collect(\lambda x.\ \mathfrak{h}2\mathfrak{s}c(x)\ in\ the\ carrier\ of\ g),$
$\quad \mathfrak{s}2\mathfrak{h}_{BinOp}(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,the\ multF\ of\ g),$
$\quad \mathfrak{s}2\mathfrak{h}c(1_{.g}))$

For the dual morphism, we indicate the result of the operation selecting the neutral element ($1_{.g}$) as the element needed in the construction of the type-class element. With its help, we can justify that the fields of the translated structure are translation of the fields.

theorem *s2hg_Prop*:
assumes $beIsoS(\mathfrak{h}2\mathfrak{s}c,\mathfrak{s}2\mathfrak{h}c,c)$ **and** g *be Group*
and $the\ carrier\ of\ g =_{\mathcal{S}} c$
and $x \in carrierI(\mathfrak{s}2\mathfrak{h}_G(\mathfrak{s}2\mathfrak{h}c,\ \mathfrak{h}2\mathfrak{s}c,\ g))$
$y \in carrierI(\mathfrak{s}2\mathfrak{h}_G(\mathfrak{s}2\mathfrak{h}c,\ \mathfrak{h}2\mathfrak{s}c,\ g))$
shows $one(\mathfrak{s}2\mathfrak{h}_G(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,g)) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}c(1_{.g})$
$x \otimes_{\mathfrak{s}2\mathfrak{h}_G(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,g)} y =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}c(\mathfrak{h}2\mathfrak{s}c(x) \otimes_g \mathfrak{h}2\mathfrak{s}c(y))$
$group\ (\mathfrak{s}2\mathfrak{h}_G(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,g))$

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 $beIsoS(\mathfrak{h}2\mathfrak{s}c,\mathfrak{s}2\mathfrak{h}c,c)$ **and** r *be Ring*
and $the\ carrier\ of\ r =_{\mathcal{S}} c$
and $x \in carrierI(\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r))$
$y \in carrierI(\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r))$
shows $zero(\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r)) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}c(0_r)$
$one(\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r)) =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}c(1_r)$
$x \oplus_{\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r)} y =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}c(\mathfrak{h}2\mathfrak{s}c(x) \oplus_r \mathfrak{h}2\mathfrak{s}c(y))$
$x \otimes_{\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r)} y =_{\mathcal{H}} \mathfrak{s}2\mathfrak{h}c(\mathfrak{h}2\mathfrak{s}c(x) \otimes_r \mathfrak{h}2\mathfrak{s}c(y))$
$ring\ (\mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}c,\mathfrak{h}2\mathfrak{s}c,r))$

Finally, we introduce the equivalent of the definition of the integer ring introduced in the MML in [40]. We show that $\mathfrak{s}2\mathfrak{h}_R$ and $\mathfrak{h}2\mathfrak{i}_R$ determine an isomorphism between the fields of the rings developed in Isabelle/HOL and the Mizar Mathematical Library.

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

$\quad multF \mapsto multint;$
$\quad OneF \mapsto 1_{\mathcal{S}}\#]$

theorem *H_Zring_to_S_Zring*:
$\quad \mathfrak{h}2\mathfrak{s}_R(\mathfrak{s}2\mathfrak{h}_{\mathbb{Z}}, \mathfrak{h}2\mathfrak{s}_{\mathbb{Z}}, INT, \mathcal{Z}) =_{\mathcal{S}} \mathbb{Z}\text{-}ring$
$\quad \mathfrak{s}2\mathfrak{h}_R(\mathfrak{s}2\mathfrak{h}_{\mathbb{Z}}, \mathfrak{h}2\mathfrak{s}_{\mathbb{Z}}, \mathbb{Z}\text{-}ring) =_{\mathcal{H}} \mathcal{Z}$

6 Related Work

As 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 (without a model) has been tried multiple times. Obua has proposed HOLZF [33], 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 [34], the combination of HOL with ZF became the basis for an LCF system, reducing the proofs in higher-order logic part to a minimum (again, since there was no guarantee, that combining the results is safe). Kunčar [29] 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 they system was not strong enough. The first author, Brown [10] 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, second and third authors [23] have specified and imported [22] 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 [17] formalizing the proof of the Kepler conjecture. There, the dependency on Coq has been eliminated and an ad-hoc justification for the concepts moved between Isabelle and HOL was specified. Logical frameworks allow importing multiple libraries at the same time, again without a model. In the Dedukti framework, Assaf and Cauderlier [3, 4] 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. Krauss and Schropp [28] 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. It also possible to deep embed multiple libraries in a single meta-theory. Rabe [39] does this practically in the MMT framework deep embedding various proof assistant foundations and providing category-theoretic mappings between some foundations.

Most implementation 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 [35] in Isabelle uses lambda abstractions and higher-order applications for example to specify the quantifiers. This is also the case in Isabelle/TLA [30]. 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 second author together with Gauthier [14] 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 [31] as well as to improve proof assistant automation [13].

7 Conclusion

We have defined a model of higher-order Tarski-Grothendieck. The model relies on a 2-inaccessible cardinal, which is the same assumption as the one required for a model of a TG set theory. This model shows that it is safe to combine higher-order features with the axioms of set theory, which has already been done by a number of developments [10, 23, 33, 34].

Moreover, thanks to the model we can safely combine results proved in TG set theory with ones proved in plain higher-order logic. We benefit from this, by combining two of the largest proof assistant libraries: the Mizar Mathematical library and the Isabelle/HOL library. Above the theorems and proofs coming from both, we define a number of isomorphisms that allow us to translate theorems proved in of these part of the library and use them in the other part.

As part of the library merging we have formally defined and proved in Isabelle the necessary concepts. This involved 18 definitions and 135 theorems, which amounts to 2667 lines of proofs. The formalization is available at:

CK will pack at the end

`http://cl-informatik.uibk.ac.at/cek/itp19/`

Apart from higher-order and set-theoretic foundations, the third most commonly used foundation is dependent type theory. The most important future work would be to investigate the consistency of a theory that imports such foundations as well.

References

1 P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Kluwer Academic Publishers, 2nd edition, 2002.

2 Peter B. Andrews. General models and extensionality. *J. Symb. Log.*, 37:395–397, 1972.

3 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. URL: `https://tel.archives-ouvertes.fr/tel-01235303`.

4 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.

5 Julian Backes and Chad E. Brown. Analytic tableaux for higher-order logic with choice. *Journal of Automated Reasoning*, 47(4):451–479, 2011.

6 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. `doi:10.1007/s10817-017-9440-6`.

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

8 Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Higher-order semantics and extensionality. *J. Symb. Log.*, 69:1027–1088, 2004.

9 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. `doi:10.1007/978-3-319-20615-8_1`.

10 Chad E. Brown. *The Egal Manual*, 2014. URL: `http://grid01.ciirc.cvut.cz/~chad/egalmanual.pdf`.

11 Chad E. Brown and Gert Smolka. Analytic tableaux for simple type theory and its first-order fragment. *Logical Methods in Computer Science*, 6(2), Jun 2010.

12 Alonzo Church. A formulation of the simple theory of types. *J. Symb. Log.*, 5:56–68, 1940.

13 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. `doi:10.1007/978-3-662-48899-7_26`.

14 Thibault Gauthier and Cezary Kaliszyk. Aligning concepts across proof assistant libraries. *J. Symbolic Computation*, 90:89–123, 2019. `doi:10.1016/j.jsc.2018.04.005`.

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

16 A. Grothendieck and J.-L. Verdier. *Théorie des topos et cohomologie étale des schémas - (SGA 4) - vol. 1*, volume 269 of *Lecture notes in mathematics*. Springer-Verlag, 1972.

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. `doi:10.1017/fmp.2017.1`.

18 Leon Henkin. Completeness in the theory of types. *J. Symb. Log.*, 15:81–91, 1950.

19 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. `doi:10.1007/11541868_9`.

20 Brian Huffman and Ondrej Kuncar. 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. `doi:10.1007/978-3-319-03545-1_9`.

21 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. `doi:10.1007/978-3-319-72453-9_12`.

22 Cezary Kaliszyk and Karol Pąk. Isabelle import infrastructure for the Mizar mathematical library. In Florian Rabe, William M. Farmer, Grant O. Passmore, and Abdou Youssef, editors, *11th International Conference on Intelligent Computer Mathematics (CICM 2018)*, volume 11006 of *LNCS*, pages 131–146. Springer, 2018. `doi:10.1007/978-3-319-96812-4\_13`.

23 Cezary Kaliszyk and Karol Pąk. Semantics of Mizar as an Isabelle object logic. *Journal of Automated Reasoning*, 2018. `doi:doi.org/10.1007/s10817-018-9479-z`.

24 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. `doi:10.1145/2854065.2854070`.

25 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.

26 Akihiro Kanamori. *The higher infinite: Large cardinals in set theory from their beginnings*. Springer Monographs in Mathematics. Springer-Verlag Berlin Heidelberg, 2 edition, 2003.

27 Dominik Kirst and Gert Smolka. Large model constructions for second-order zf in dependent type theory. *Certified Programs and Proofs - 7th International Conference, CPP 2018, Los Angeles, USA, January 8-9, 2018*, Jan 2018.

28 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.

29 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.

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

31 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. `doi:10.1007/978-3-319-62075-6_7`.

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

33 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.

34 Steven Obua, Jacques D. Fleuriot, Phil Scott, and David Aspinall. ProofPeer: Collaborative theorem proving. *CoRR*, abs/1404.6186, 2014. URL: `http://arxiv.org/abs/1404.6186`.

35 Lawrence C. Paulson. Set theory for verification: I. From foundations to functions. *J. Autom. Reasoning*, 11(3):353–389, 1993. `doi:10.1007/BF00881873`.

36 Karol Pąk. Brouwer Fixed Point Theorem in the General Case. *Formalized Mathematics*, 19(**3**):151–153, 2011. `doi:10.2478/v10037-011-0024-3`.

37 Karol Pąk. Brouwer Invariance of Domain Theorem. *Formalized Mathematics*, 22(**1**):21–28, 2014. `doi:10.2478/forma-2014-0003`.

38 Karol Pąk. Topological manifolds. *Formalized Mathematics*, 22(**2**):179–186, 2014. `doi:10.2478/forma-2014-0019`.

39 Florian Rabe. How to identify, translate and combine logics? *J. Log. Comput.*, 27(6):1753–1798, 2017. `doi:10.1093/logcom/exu079`.

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

41 Alfred Tarski. Über unerreichbare Kardinalzahlen. *Fundamenta Mathematica*, 30:68–89, 1938. URL: `http://matwbn.icm.edu.pl/ksiazki/fm/fm30/fm30113.pdf`.

42 Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 2002. Released 1989.

43 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. `doi:10.1007/978-3-540-71067-7_7`.