JFP **13** (2): 261–293, March 2003. © 2003 Cambridge University Press
DOI: 10.1017/S0956796802004501 Printed in the United Kingdom

Setoids in type theory

GILLES BARTHE, VENANZIO CAPRETTA
INRIA Sophia-Antipolis, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France
(*e-mail:* {Gilles.Barthe}{Venanzio.Capretta}@inria.fr)

OLIVIER PONS
CNAM, France
(*e-mail:* pons@cnam.fr)

Abstract

Formalising mathematics in dependent type theory often requires to represent sets as setoids, i.e. types with an explicit equality relation. This paper surveys some possible definitions of setoids and assesses their suitability as a basis for developing mathematics. According to whether the equality relation is required to be reflexive or not we have total or partial setoid, respectively. There is only one definition of total setoid, but four different definitions of partial setoid, depending on four different notions of setoid function. We prove that one approach to partial setoids in unsuitable, and that the other approaches can be divided in two classes of equivalence. One class contains definitions of partial setoids that are equivalent to total setoids; the other class contains an inherently different definition, that has been useful in the modeling of type systems. We also provide some elements of discussion on the merits of each approach from the viewpoint of formalizing mathematics. In particular, we exhibit a difficulty with the common definition of subsetoids in the partial setoid approach.

1 Introduction

Proof-development systems such as Agda (Coquand & Coquand, 1999), Coq (2002) and Lego (Luo & Pollack, 1992) rely on powerful type systems and have been successfully used in the formalization of mathematics. Nevertheless, their underlying type theories – Martin-Löf's Type Theory (Nordström *et al.*, 1990) and the Calculus of Inductive Constructions (Werner, 1994) – fail to support extensional concepts such as quotients and subsets, which play a fundamental role in mathematics. While significant efforts have been devoted to embed subset and quotient types in type theory (Altenkirch, 1999; Barthe, 1995a; Courtieu, 2001; Hofmann, 1994; Hofmann, 1995b; Hofmann, 1995a; Jacobs, 1999; Maietti, 1999; Salvesen & Smith, 1988), all proposals to date are unsatisfactory, mostly because they introduce non-canonical elements or lead to undecidable type-checking. Thus current versions of Agda, Coq and Lego do not implement subset or quotient types. Instead, mathematical formalizations usually rely on *setoids*, i.e. mathematical structures packaging a carrier: the "set", its equality: the "book equality" and a proof component ensuring that the book equality is well-behaved. This notion was introduced in constructive mathematics by Bishop (1967).

While setoids have been extensively used in the formalization of mathematics, there does not seem to be any consensus on their precise definition. Instead, setoids come in several flavours: for example, they can be total (the book equality is an equivalence relation) or partial (the book equality is a partial equivalence relation); classical (apartness is defined as the logical negation of equality) or constructive (setoids come equipped with an apartness relation independent from the equality relation). Worse, literature about setoids fails to compare the respective merits of existing approaches, especially from the viewpoint of formalising mathematics.

The purpose of this paper is four-fold:

- in section 2, we review existing approaches to define (the category of) setoids. It turns out that there are several alternatives to define morphisms of partial setoids, leading to different definitions of the category of partial setoids;
- in section 3, we show that there are, up to equivalence of categories, two approaches to setoids. Further, we show that one approach to partial setoids, that appears in the literature, uses a definition of function setoid that does not give a correct exponent object for a cartesian closed category;
- in section 4, we assess the suitability of the different approaches by considering choice principles. We show that both partial and total setoids can be turned into a model of intuitionistic set theory by assuming the axiom of unique choice. However, the axiom of unique choice for partial setoids is too weak, in that it does not permit us to define some very natural functions on partial setoids;
- in section 5, we introduce some basic constructions on setoids, such as subsets and quotients, and assess the relative advantages of existing approaches w.r.t. these constructions.

Setting and notations To fix ideas, we shall be working with an extension of the Calculus of Constructions with dependent record types and universes. Dependent record types are used to formalize mathematical structures and universes are used to form the type of categories. Note that we do not need record subtyping and cumulativity between universes and that equality between records is neither extensional nor typed. However, our results are to a large extent independent from the choice of a type system.

Following Luo (1994), we use **Prop** for the universe of propositions, $\mathbf{Type}_i$ for the i-th universe of types. By abuse of notation, we write **Type** for $\mathbf{Type}_0$ so we have **Prop**:**Type** and $\mathbf{Type}_i : \mathbf{Type}_{i+1}$. Moreover, we use the notation $\langle \mathsf{l} : L,\ \mathsf{r} : R\rangle$ for a record type with two fields l of type L and r of type R and $\langle \mathsf{l} = a,\ \mathsf{r} = b\rangle$ for an inhabitant of that type. Finally, we let $\doteq$ denote Leibniz equality, defined as

$$\lambda A : \mathbf{Type}.\lambda x, y : A.\ \Pi P : A \to \mathbf{Prop}.\ (P\ x) \to (P\ y)$$

Proof scripts Most of the results presented in the paper have been formalized in the proof assistant Coq V7.3 and can be obtained from the following web page: `http://www-sop.inria.fr/lemme/Venanzio.Capretta/setoids/index.html`.

2 Setoids

This section gathers some existing definitions of setoids. Here we focus on classical setoids, i.e. setoids that do not carry an apartness relation. Similarly, we ignore issues related to the decidability of equality and do not require equality to be decidable. In this setting, there is a single reasonable definition for total setoids and morphisms of total setoids. Also, there is a single reasonable definition for partial setoids, but there are at least four possible definitions for morphisms of partial setoids.

Below we give these possible definitions of setoids. None of them is original. The first definition has been used, for example, in the formalization of basic algebra (Aczel, 1993; Barthe, 1995b) and of constructive category theory (Huet & Saïbi, 2000). The second definition has been used, for instance, in the formalization of polynomials (Bailey, 1993). The other definitions have been used by Hofmann (1994, 1995a, 1995b) to interpret extensional concepts in intensional type theory.

2.1 Total setoids

A total setoid consists of a type T (the carrier), a binary relation R on T (the book equality), and a proof that R is an equivalence relation over T.

Definition 1
The type of *total setoids* is defined as the record type

$$\textsc{Set}_t = \langle \mathsf{el}_t : \mathbf{Type},\ \mathsf{eq}_t : \mathsf{el}_t \to \mathsf{el}_t \to \mathbf{Prop},\ \mathsf{er} : \textsc{Er}\ \mathsf{el}_t\ \mathsf{eq}_t \rangle$$

where

$$\begin{array}{llll} \textsc{Er} = & \lambda A : \mathbf{Type}.\lambda R : A \to A \to \mathbf{Prop}. & & \\ & \langle\ \ \mathsf{refl}_t & : & \forall x : A.\ R\ x\ x, \\ & \quad\ \mathsf{sym}_t & : & \forall x, y : A.\ (R\ x\ y) \to (R\ y\ x), \\ & \quad\ \mathsf{trans}_t & : & \forall x, y, z : A.\ (R\ x\ y) \to (R\ y\ z) \to (R\ x\ z)\rangle \end{array}$$

By abuse of notation, we write $\mathsf{el}_t\ A$ for $A \cdot \mathsf{el}_t$ and $=_A$ for $A \cdot \mathsf{eq}_t$.

Each type T induces a setoid $\ddagger T$ defined as

$$\langle \mathsf{el}_t = T,\ \mathsf{eq}_T = \lambda x, y : T.\ x \doteq y,\ \mathsf{er} = \ldots \rangle$$

A map of total setoids is a map between the underlying carriers which preserves equality. So, if A and B are two total setoids, a map of total setoids from A to B consists of a function $f : \mathsf{el}_t\ A \to \mathsf{el}_t\ B$ and a proof that f preserves equality.

Definition 2
Let A and B be two total setoids.

- The type $\textsc{map}_t\ A\ B$ of *morphisms of total setoids* from A to B is defined as the record type

  $$\begin{array}{llll} \textsc{map}_t\ A\ B = \langle & \mathsf{ap}_t & : & \mathsf{el}_t\ A \to \mathsf{el}_t\ B, \\ & \mathsf{ext}_t & : & \forall x, y : \mathsf{el}_t\ A.\ (x =_A y) \to (\mathsf{ap}_t\ x =_B \mathsf{ap}_t\ y)\rangle \end{array}$$

  By abuse of notation, we write $\mathsf{ap}_t\ f\ a$ for $f \cdot \mathsf{ap}_t\ a$.

- The *function space setoid* $\textsc{Map}_t\ A\ B$ of maps from A to B is defined as the record

$$\begin{aligned}\textsc{Map}_t\ A\ B = \langle\ \ \mathsf{el}_t &= \textsc{map}_t\ A\ B,\\ \mathsf{eq}_t &= \lambda f, g : \textsc{map}_t\ A\ B.\ \forall x : \mathsf{el}_t\ A.\ (\mathsf{ap}_t\ f\ x) =_B (\mathsf{ap}_t\ g\ x),\\ \mathsf{er} &= \ldots\rangle\end{aligned}$$

We conclude this paragraph by observing that it would have been equivalent to define equality between morphisms from A to B as

$$\lambda f, g : \textsc{map}_t\ A\ B.\ \forall x, y : \mathsf{el}_t\ A.\ (x =_A y) \to (\mathsf{ap}_t\ f\ x) =_B (\mathsf{ap}_t\ g\ y)$$

This alternative definition will be used later for partial setoids, but in that case the two definitions will not be equivalent.

2.2 Partial setoids

A partial setoid consists of a type T (the carrier), a binary relation R on T (the book equality) and a proof that R is a partial equivalence relation over T.

Definition 3
The type of *partial setoids* is defined as the record type

$$\textsc{Set}_p = \langle \mathsf{el}_p : \mathbf{Type},\ \mathsf{eq}_p : \mathsf{el}_p \to \mathsf{el}_p \to \mathbf{Prop},\ \mathsf{per} : \textsc{Per}\ \mathsf{el}_p\ \mathsf{eq}_p\rangle$$

where

$$\begin{aligned}\textsc{Per} = \ &\lambda A : \mathbf{Type}.\lambda R : A \to A \to \mathbf{Prop}.\\ &\langle\ \ \mathsf{sym}_p : \forall x, y : A.\ (R\ x\ y) \to (R\ y\ x),\\ &\quad\ \mathsf{trans}_p : \forall x, y, z : A.\ (R\ x\ y) \to (R\ y\ z) \to (R\ x\ z)\rangle\end{aligned}$$

By abuse of notation, we write $\mathsf{el}_p\ A$ for $A \cdot \mathsf{el}_p$ and $=_A$ for $A \cdot \mathsf{eq}_p$.

In the framework of partial setoids, one distinguishes between defined and undefined elements. The defined elements of a partial setoid A are those expressions $x : \mathsf{el}_p\ A$ such that $x =_A x$; they form the domain of the partial setoid.

Definition 4

- The *domain* of a partial setoid A is defined as the record type

$$\textsc{domain}\ A = \langle \mathsf{cont} : \mathsf{el}_p\ A, \mathsf{def} : \mathsf{cont} =_A \mathsf{cont}\rangle$$

- The *domain setoid* of a partial setoid A is defined as

$$\begin{aligned}\textsc{Domain}\ A = \langle\ \ &\mathsf{el}_p = \textsc{domain}\ A,\\ &\mathsf{eq}_p = \lambda x, y : \textsc{domain}\ A.\ x \cdot \mathsf{cont} =_A y \cdot \mathsf{cont},\\ &\mathsf{per} = \ldots\rangle\end{aligned}$$

Note that the underlying equality of domain setoids is a total equivalence relation. In the next section, we will use domain setoids to relate partial setoids to total setoids.

We now turn to the definition of morphism of partial setoids. It turns out that there are several possible alternatives for this notion; below we present four alternatives that appear in the literature. The alternatives are determined by the following two issues:

1. What is the status of extensionality? Following the definition of morphism of total setoids, a morphism of partial setoids from A to B can be defined as a pair $\langle f, \phi_f \rangle$ where f is a type-theoretical function mapping "elements" of A to "elements" of B and ϕ_f is a proof that f preserves equality; this definition is similar to the one for total setoids. However, one can also take advantage of the possibility to restrict the defined elements by using a partial equivalence relation and choose (1) to define morphisms of setoids as type-theoretical functions, (2) to embed extensionality in the definition of equality for morphisms of setoids – in such a way that a morphism is defined w.r.t. the equality of the setoid $\textsc{Map}\ A\ B$ iff it preserves equality.
2. What is the domain of the function? A morphism of partial setoids from A to B may either take as inputs elements of A, or elements of $\textsc{Domain}\ A$ – in the latter case, one will require that the morphism is constant in the def field of the record.

This leaves us with four alternatives, which are summarized and described below.

Extensionality vs. inputs	Elements of A	Elements of $\textsc{Domain}\ A$
In the definition of morphism	$\textsc{Set}_p$	$\textsc{Set}_q$
In the definition of equality	$\textsc{Set}_r$	$\textsc{Set}_s$

- The first alternative, which appears in Bailey (1993), is to adapt to partial setoids the definition of map of total setoids. Indeed, one can define a map of partial setoids as a map between the underlying carriers which preserves equality.

  Definition 5
  Let A and B be two partial setoids.

  - The type $\textsc{map}_p\ A\ B$ of *P-morphisms of partial setoids* from A to B is defined as the record type

    $$\begin{aligned}\textsc{map}_p\ A\ B = \langle\ \mathsf{ap}_p &: \mathsf{el}_p\ A \to \mathsf{el}_p\ B,\\ \mathsf{ext}_p &: \forall x, y : \mathsf{el}_p\ A.\ (x =_A y) \to (\mathsf{ap}_p\ x =_B \mathsf{ap}_p\ y)\rangle\end{aligned}$$

    By abuse of notation, we write $\mathsf{ap}_p\ f\ a$ for $f \cdot \mathsf{ap}_p\ a$.
  - The *P-function space setoid* $\textsc{Map}_p\ A\ B$ of maps from A to B is defined as the record

    $$\begin{aligned}\textsc{Map}_p\ A\ B = \langle\ \mathsf{el}_p &= \textsc{map}_p\ A\ B,\\ \mathsf{eq}_p &= \lambda f, g : \textsc{map}_p\ A\ B.\ \forall x : \mathsf{el}_p\ A.\\ &\quad\ (x =_A x) \to (\mathsf{ap}_p\ f\ x) =_B (\mathsf{ap}_p\ g\ x),\\ \mathsf{per} &= \ldots\rangle\end{aligned}$$

  Note that $f =_{\textsc{Map}_p\ A\ B} f$ for every $A, B : \textsc{Set}_p$ and $f : \textsc{map}_p\ A\ B$.
- The second alternative requires that a function from A to B takes two arguments, namely an element $a : \mathsf{el}_p\ A$ and a proof $\phi : a =_A a$. The second argument is here to prevent some anomalies with empty sets (see section 3.2), but the result of the application of the function does not depend on it.

Definition 6
Let A and B be two partial setoids.

- The type $\textsc{map}_q\ A\ B$ of *Q-morphisms of partial setoids* from A to B is defined as the record type

$$\begin{aligned}\textsc{map}_q\ A\ B = \langle\ \mathsf{ap}_q &: \Pi a : \mathsf{el}_p\ A.\ (a =_A a) \to \mathsf{el}_p\ B,\\ \mathsf{ext}_q &: \forall x, y : \mathsf{el}_p\ A. \forall \phi : x =_A x. \forall \psi : y =_A y.\\ &\quad (x =_A y) \to (\mathsf{ap}_q\ x\ \phi =_B \mathsf{ap}_q\ y\ \psi)\rangle\end{aligned}$$

- The *Q-function space setoid* $\textsc{Map}_q\ A\ B$ of maps from A to B is defined as the record

$$\begin{aligned}\textsc{Map}_q\ A\ B = \langle\ \mathsf{el}_p &= \textsc{map}_q\ A\ B,\\ \mathsf{eq}_p &= \lambda f, g : \textsc{map}_q\ A\ B.\ \forall x : \mathsf{el}_p\ A.\ \forall \phi : (x =_A x).\\ &\quad (\mathsf{ap}_q\ f\ x\ \phi) =_B (\mathsf{ap}_q\ g\ x\ \phi),\\ \mathsf{per} &= \ldots\rangle\end{aligned}$$

This approach makes function application awkward; perhaps for this reason it has never been used in practice. Note that $f =_{\textsc{Map}_q\ A\ B} f$ for every $A, B : \textsc{Set}_p$ and $f : \textsc{map}_q\ A\ B$.

- The third alternative, which appears in Hofmann (1995b) and has been used extensively in Čubrić *et al.* (1998) and Qiao (2000), does not require inhabitants of the carrier type of the function setoid to preserve equality: instead, the function space between A and B is defined as a partial setoid with carrier $\mathsf{el}_p\ A \to \mathsf{el}_p\ B$. Equality is defined in the obvious way; as a consequence, the defined elements of this partial setoids are those type-theoretical functions preserving equality.

Definition 7
Let A and B be two partial setoids.

- The type $\textsc{map}_r\ A\ B$ of *R-morphisms of partial setoids* from A to B is defined as the type

$$\textsc{map}_r\ A\ B = \mathsf{el}_p\ A \to \mathsf{el}_p\ B$$

- The *R-function space setoid* $\textsc{Map}_r\ A\ B$ of maps from A to B is defined as the record

$$\begin{aligned}\textsc{Map}_r\ A\ B = \langle\ \mathsf{el}_p &= \textsc{map}_r\ A\ B,\\ \mathsf{eq}_p &= \lambda f, g : \textsc{map}_r\ A\ B.\ \forall x, y : \mathsf{el}_p\ A.\\ &\quad (x =_A y) \to f\ x =_B g\ y,\\ \mathsf{per} &= \ldots\rangle\end{aligned}$$

Note that we need not have $f =_{\textsc{Map}_r\ A\ B} f$ for $A, B : \textsc{Set}_p$ and $f : \textsc{map}_r\ A\ B$: in other words, $\textsc{Map}_r\ A\ B$ may be a partial setoid. Also, note that transitivity of equality for $\textsc{Map}_r\ A\ B$ uses that

$$\forall x, x' : \mathsf{el}_p\ A.\ (x =_A x') \to x =_A x$$

which is provable from the symmetry and transitivity of $=_A$.

- The fourth alternative, which appears in Hofmann (1994), takes as inputs defined elements of A and does not require inhabitants of the carrier type of the function setoid to preserve equality.

Definition 8
Let A and B be two partial setoids.

- The type $\textsc{map}_s\ A\ B$ of *S-morphisms of partial setoids* from A to B is defined as the type
$$\textsc{map}_s\ A\ B = \Pi a : \mathsf{el}_p\ A.\ x =_A x \rightarrow \mathsf{el}_p\ B$$
- The *S-function space setoid* $\textsc{Map}_s\ A\ B$ of maps from A to B is defined as the record
$$\begin{array}{rcl} \textsc{Map}_s\ A\ B = \langle\ \ \mathsf{el}_p & = & \textsc{map}_s\ A\ B, \\ \mathsf{eq}_p & = & \lambda f, g : \textsc{map}_s\ A\ B.\ \forall x, y : \mathsf{el}_p\ A. \\ & & \forall \phi : x =_A x.\ \forall \psi : y =_A y. \\ & & (x =_A y) \rightarrow f\ x\ \phi =_B g\ y\ \psi, \\ \mathsf{per} & = & \ldots\rangle \end{array}$$

2.3 Total functional relations as morphisms?

All previous definitions introduce morphisms of setoids as (structures with underlying) type-theoretical functions. In contrast, set theory views morphisms of sets as graphs. One may therefore wonder about this departure from mainstream mathematics. Two points need to be emphasized:

- first, our type theory is expressive enough to formalize the notion of total functional relation and thus one needs not depart, at least in principle, from mainstream mathematics;
- secondly, our type theory does make a difference between the two approaches: every function has an associated total functional relation but the converse needs not be true.

In section 4 we provide some choice axioms under which the two approaches coincide, and briefly discuss the validity of our results/claims in other type-theoretical settings, but for the time being, let us focus on the relative benefits of the two approaches:

- Using type-theoretical functions as the underlying concept for morphisms of setoids is very much in line with the philosophy of type theory because it provides a computational meaning to functions. In effect, most formalizations of mathematics in type theory follow the first approach.
- Using total functional relations as the underlying concept for morphisms of setoids avoids some of the difficulties with choice principles, see Section 4. On the other hand, total functional relations do not have a computational meaning, which is a weakness from a type-theoretical perspective, and their use complicates the presentation of formal proofs, because it becomes impossible to write $f\ a$ for the result of applying the function f to a.

While we are strongly in favour of using type-theoretical functions as the the underlying concept for morphisms of setoids, we would like to conclude this section by observing that it is possible to use a monadic style to manipulate total functional relations. Concretely, our suggestion is to use the ι-monad[1], which assigns to every setoid $A : \textsc{Set}_t$ the setoid $\iota\ A$ defined as the predicates over A that are satisfied by exactly one element. Formally, we need to introduce the setoid Ω of propositions, defined as

$$\langle \mathsf{el}_t = \mathbf{Prop},\ \mathsf{eq}_t = \lambda P, Q : \mathbf{Prop}.\ P \leftrightarrow Q,\ \mathsf{er} = \ldots\rangle$$

the quantifier $\exists!\ x \in A.\ P\ x$, where A is a total setoid, defined as

$$\exists x : \mathsf{el}_t\ A.\ (P\ x) \wedge (\forall y : \mathsf{el}_t\ A.\ P\ y\ \rightarrow x =_A y)$$

the type of predicates over A that are satisfied by exactly one element

$$\underline{\iota}\ A \quad = \quad \langle \mathsf{up} :\ \textsc{map}_t\ A\ \Omega,\ \mathsf{pp} :\ \exists!\ x \in A.\ \mathsf{ap}_t\ \mathsf{up}\ x\rangle$$

and finally the setoid $\iota\ A$ itself

$$\langle \mathsf{el}_t = \underline{\iota}\ A,\ \mathsf{eq}_t = \lambda P, Q : \underline{\iota}\ A.\ P \cdot \mathsf{up} =_{\mathrm{MAP}_t\ A\ \Omega} Q \cdot \mathsf{up},\ \mathsf{er} = \ldots\rangle$$

It is easy to turn ι into a monad. For example, the unit η_ι of the monad is defined as

$$\begin{aligned}\lambda A : \textsc{Set}_t.\ \langle\ \mathsf{ap}_t = \lambda x : \mathsf{el}_t\ A.\ \langle\ & \mathsf{up} = \langle \mathsf{ap}_t = \lambda y : \mathsf{el}_t\ A.\ x =_A y,\ \mathsf{ext}_t = \ldots\rangle, \\ & \mathsf{pp} = \ldots\rangle, \\ \mathsf{ext}_t = \ldots\rangle &\end{aligned}$$

To our knowledge, this approach has not been pursued before, and we have no practical experience with it; yet we feel that it is likely to be less cumbersome than manipulating total functional relations directly.

We conclude this section by observing that it is possible to treat total relations likewise, i.e. by defining an ϵ-monad which maps every setoid A to the setoid $\epsilon\ A$ of non-empty predicates over A. Concretely he type of non-empty predicates over A is defined as

$$\underline{\epsilon}\ A \quad = \quad \langle \mathsf{np} :\ \textsc{map}_t\ A\ \Omega,\ \mathsf{ne} :\ \exists x :\ \mathsf{el}_t\ A.\ \mathsf{ap}_t\ \mathsf{np}\ x\rangle$$

and the setoid $\epsilon\ A$ is defined as

$$\langle \mathsf{el}_t = \underline{\epsilon}\ A,\ \mathsf{eq}_t = \lambda P, Q : \underline{\epsilon}\ A. P \cdot \mathsf{np} =_{\mathrm{MAP}_t\ A\ \Omega} Q \cdot \mathsf{np},\ \mathsf{er} = \ldots\rangle$$

Again, it is a simple matter to turn ϵ into a monad.

3 Categories of setoids

The purpose of this section is to associate to every notion of setoid its corresponding category, and show that all categories defined in the previous section form a model of the simply typed λ-calculus. However, it turns out that the function space setoid for **PSet** does not correspond to the exponent that turns **PSet** into a cartesian closed category. Further, we compare the five categories of setoids; it turns out that there are essentially two categories of setoids: **TSet**, which is equivalent to **QSet** and **SSet**, and **RSet**, which is equivalent to **PSet**.

[1] A *monad* in a category $\mathbb{C}$ is a triple $\langle M, \eta, \mu\rangle$ where $M : \mathbb{C} \rightarrow \mathbb{C}$ is a functor, $\eta : \mathsf{id}_{\mathbb{C}} \rightarrow M$ and $\mu : M \circ M \rightarrow M$ are natural transformations such that $\mu \circ \mu M = \mu \circ M\mu$ and $\mu \circ \eta M = \mu \circ M\eta = \mathsf{id}_M$.

3.1 The category of total setoids

In this subsection, we define the category of total setoids and show that it forms a cartesian closed category. Following previous work on the formalization of category theory in type theory (Aczel, 1993; Huet & Saïbi, 2000; Saïbi, 1998), a T-category (or simply a category) consists of

- a type obj of objects (in $\mathbf{Type}_1$);
- a polymorphic setoid of morphisms $\mathsf{hom} : \mathsf{obj} \to \mathsf{obj} \to \textsc{Set}_t$;
- a polymorphic composition operator

  $$\bullet : \Pi A, B, C : \mathsf{obj}.\ \textsc{bmap}_t\ (\mathsf{hom}\ A\ B)\ (\mathsf{hom}\ B\ C)\ (\mathsf{hom}\ A\ C),$$

  where $\textsc{bmap}_t\ X\ Y\ Z$ is defined as $\textsc{map}_t\ X\ (\textsc{Map}_t\ Y\ Z)$;
- a polymorphic identity $\mathsf{id} : \Pi A : \mathsf{obj}.\ \mathsf{el}_t\ (\mathsf{hom}\ A\ A)$;
- a proof that composition is associative and identity acts as a unit.

Note that objects of a category are required to form a type, but the morphisms between two objects are required to form a setoid because we need to identify equal morphisms: Leibniz equality, which is the default equality relation in type theory, is too rigid for this purpose.

Definition 9

The type $\textsc{Cat}_t$ of *T-categories* is defined as the record type

$$\begin{array}{rcl}
\langle \mathsf{obj} & : & \mathbf{Type}_1, \\
\mathsf{hom} & : & \mathsf{obj} \to \mathsf{obj} \to \textsc{Set}_t, \\
\bullet & : & \Pi A, B, C : \mathsf{obj}.\ \textsc{bmap}_t\ (\mathsf{hom}\ A\ B)\ (\mathsf{hom}\ B\ C)\ (\mathsf{hom}\ A\ C), \\
\mathsf{id} & : & \Pi A : \mathsf{obj}.\ \underline{\mathsf{hom}}\ A\ A, \\
\mathsf{catlaw} & : & \phi_{\mathrm{cat}} \rangle
\end{array}$$

where ϕ_{cat} is

$$\begin{array}{rl}
& (\forall A, B, C, D : \mathsf{obj}.\ \forall f : \underline{\mathsf{hom}}\ A\ B.\ \forall g : \underline{\mathsf{hom}}\ B\ C.\ \forall h : \underline{\mathsf{hom}}\ C\ D. \\
& \quad f \underline{\bullet} (g \underline{\bullet} h) =_{(\mathsf{hom}\ A\ D)} (f \underline{\bullet} g) \underline{\bullet} h) \\
\wedge & (\forall A, B : \mathsf{obj}.\ \forall f : \underline{\mathsf{hom}}\ A\ B. \\
& \quad (\mathsf{id}\ A) \underline{\bullet} f =_{(\mathsf{hom}\ A\ B)} f \wedge f \underline{\bullet} (\mathsf{id}\ B) =_{(\mathsf{hom}\ A\ B)} f)
\end{array}$$

using $\underline{\mathsf{hom}}\ Y\ Z$ as a shorthand for $\mathsf{el}_t\ (\mathsf{hom}\ Y\ Z)$ and $x \underline{\bullet} y$ as a shorthand for $\mathsf{ap}_t\ (\bullet\ x)\ y$. In the sequel, we use obj_C, hom_C and $\underline{\mathsf{hom}}_C$ as shorthand for $C \cdot \mathsf{obj}$, $C \cdot \mathsf{hom}$ and $\mathsf{el}_t\ (\mathsf{hom}_C\ Y\ Z)$ respectively.

Total setoids can be made into a category that plays in $\textsc{Cat}_t$ the role that **Set** plays in standard category theory.

Definition 10

The category **TSet** of total setoids takes as objects elements of $\textsc{Set}_t$ and as homset between A and B the setoid $\textsc{Map}_t\ A\ B$.

Functors are defined in a similar way; informally, a functor from $C : \mathrm{Cat}_t$ to $C' : \mathrm{Cat}_t$ consists of

- a function $\mathsf{fobj} : \mathsf{obj}_C \to \mathsf{obj}_{C'}$;
- a polymorphic map of setoids

$$\mathsf{fmor} : \Pi o, o' : \mathsf{obj}_C.\ \mathrm{map}_t\ (\mathsf{hom}_C\ o\ o')\ (\mathsf{hom}_{C'}\ (\mathsf{fobj}\ o)\ (\mathsf{fobj}\ o'));$$

- a proof that fobj preserves identities and composition.

Definition 11
The parametric type Func_t of *T-functors* is defined as

$$\begin{array}{lrcl}
\lambda C, C' : \mathrm{Cat}_t.\ \langle & \mathsf{fobj} & : & \mathsf{obj}_C \to \mathsf{obj}_{C'}, \\
& \mathsf{fmor} & : & \Pi o, o' : \mathsf{obj}_C.\ \mathrm{map}_t\ (\mathsf{hom}_C\ o\ o')\ (\mathsf{hom}_{C'}\ (\mathsf{fobj}\ o)\ (\mathsf{fobj}\ o')), \\
& \mathsf{flaw} & : & \ldots\rangle
\end{array}$$

The parametric type BFunc_t of *T-bifunctors* is defined as

$$\begin{array}{lrcl}
\lambda C, C', C'' : \mathrm{Cat}_t.\ \langle & \mathsf{bfobj} & : & \mathsf{obj}_C \to \mathsf{obj}_{C'} \to \mathsf{obj}_{C''}, \\
& \mathsf{bfmor} & : & \Pi o, o' : \mathsf{obj}_C.\ \Pi u, u' : \mathsf{obj}_{C'}. \\
& & & \quad \mathrm{bmap}_t\ (\mathsf{hom}_C\ o\ o')\ (\mathsf{hom}_{C'}\ u\ u') \\
& & & \qquad (\mathsf{hom}_{C''}\ (\mathsf{bfobj}\ o\ u)\ (\mathsf{bfobj}\ o'\ u')), \\
& \mathsf{bflaw} & : & \ldots\rangle
\end{array}$$

We now proceed towards the definition of cartesian closedness and define the notions of terminal object, products and exponents. Recall that o is a terminal object if for every object o' there exists a unique morphism from o' to o.

Definition 12
The parametric type TObj_t of *terminal objects* is defined as:

$$\begin{array}{lrcl}
\lambda C : \mathrm{Cat}_t.\ \langle & \mathsf{tobj} & : & \mathsf{obj}_C, \\
& \mathsf{tarr} & : & \Pi o : \mathsf{obj}_C.\ \underline{\mathsf{hom}}_C\ o\ \mathsf{tobj}, \\
& \mathsf{tlaw} & : & \forall o : \mathsf{obj}_C.\ \forall f : \underline{\mathsf{hom}}_C\ o\ \mathsf{tobj}.\ f =_{(\mathsf{hom}_C\ o\ \mathsf{tobj})} \mathsf{tarr}\ o\rangle
\end{array}$$

As appears from the above definition, terminal objects are understood constructively. This constructive reading of categorical notions is in line, for example, with Huet & Saïbi (2000) and Saïbi (1998), and is more appropriate for the issues tackled here.

Definition 13
The parametric (record) type Prod_t is defined as

$$\begin{array}{lrcl}
\lambda C : \mathrm{Cat}_t.\ \langle & \mathsf{prodo} & : & \mathsf{obj}_C \to \mathsf{obj}_C \to \mathsf{obj}_C, \\
& \mathsf{proda} & : & \Pi o, o', o'' : \mathsf{obj}_C.\ \mathrm{bmap}_t\ (\mathsf{hom}_C\ o\ o')\ (\mathsf{hom}_C\ o\ o'') \\
& & & \qquad (\mathsf{hom}_C\ o\ (\mathsf{prodo}\ o'\ o'')), \\
& \mathsf{prodl} & : & \Pi o, o' : \mathsf{obj}_C.\ \underline{\mathsf{hom}}_C\ (\mathsf{prodo}\ o\ o')\ o, \\
& \mathsf{prodr} & : & \Pi o, o' : \mathsf{obj}_C.\ \underline{\mathsf{hom}}_C\ (\mathsf{prodo}\ o\ o')\ o', \\
& \mathsf{prodlaw} & : & \cdots\rangle
\end{array}$$

Given a category $\mathscr{C}$ with a product structure $\mathsf{prod} : \mathrm{Prod}_t\ \mathscr{C}$, we use the notation $o \times o'$ for $\mathsf{prod} \cdot \mathsf{prodo}\ o\ o'$. We also use the notation $f \times f' : \underline{\mathsf{hom}}_{\mathscr{C}}\ o_1 \times o'_1\ o_2 \times o'_2$ to denote the product morphism of $f : \underline{\mathsf{hom}}_{\mathscr{C}}\ o_1\ o_2$ and $f' : \underline{\mathsf{hom}}_{\mathscr{C}}\ o'_1\ o'_2$.

We can now define a Cartesian Closed Category (CCC) as a structure consisting of:

- a category $\mathscr{C}$;
- a terminal object $\bullet$;
- a bifunctor for products $\times : \mathscr{C} \times \mathscr{C} \rightarrow \mathscr{C}$;
- a bifunctor for exponents $\Rightarrow : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \rightarrow \mathscr{C}$;
- an evaluation map $\mathsf{eval}_{o,o'} : \underline{\mathsf{hom}}\ ((o \Rightarrow o') \times o)\ o'$ for every pair of objects o and o';
- an abstraction map $\mathsf{abst}_{o,o',o''} : \textsc{map}_t\ (\mathsf{hom}\ (o \times o')\ o'')\ (\mathsf{hom}\ o\ (o' \Rightarrow o''))$ for every three objects o, o', and o'';
- a proof that for every $f : \underline{\mathsf{hom}}\ (o \times o')\ o''$, $\mathsf{ap}_t\ \mathsf{abst}_{o,o',o''}\ f$ is the unique morphism that gives back f when composed with $\mathsf{eval}_{o',o''}$.

Definition 14
The type of *cartesian closed categories* CCC_t is the record type

$$\begin{array}{rcl}
\langle \mathsf{cccat} & : & \textsc{Cat}_t, \\
\mathsf{terminal} & : & \mathrm{T}\textsc{Obj}_t\ \mathsf{cccat}, \\
\mathsf{ccprod} & : & \textsc{Prod}_t\ \mathsf{cccat}, \\
\mathsf{ccexp} & : & \mathsf{cccat}\cdot\mathsf{obj} \rightarrow \mathsf{cccat}\cdot\mathsf{obj} \rightarrow \mathsf{cccat}\cdot\mathsf{obj}, \\
\mathsf{cceval} & : & \Pi o, o' : \mathsf{cccat}\cdot\mathsf{obj}.\underline{\mathsf{cccat}\cdot\mathsf{hom}}\ ((o \Rightarrow o') \times o)\ o', \\
\mathsf{ccabst} & : & \Pi o, o', o'' : \mathsf{cccat}\cdot\mathsf{obj}. \\
 & & \textsc{map}_t\ (\mathsf{cccat}\cdot\mathsf{hom}\ (o \times o')\ o'')\ (\mathsf{ccat}\cdot\mathsf{hom}\ o\ (o' \Rightarrow o'')), \\
\mathsf{cceq} & : & \forall o, o', o'' : \mathsf{cccat}\cdot\mathsf{obj}.\forall f : \underline{\mathsf{cccat}\cdot\mathsf{hom}}\ (o \times o')\ o''. \\
 & & ((\mathsf{ap}_t\ \mathsf{ccabst}_{o,o',o''}\ f) \times (\mathsf{cccat}\cdot\mathsf{id}\ o')) \bullet \mathsf{cceval}_{o',o''} \\
 & & \quad =_{(\mathsf{cccat}\cdot\mathsf{hom}\ (o\times o')\ o'')} f, \\
\mathsf{ccunique} & : & \forall o, o', o'' : \mathsf{cccat}\cdot\mathsf{obj}. \\
 & & \forall f : \underline{\mathsf{cccat}\cdot\mathsf{hom}}\ (o \times o')\ o''.\forall g : \underline{\mathsf{cccat}\cdot\mathsf{hom}}\ o\ (o' \Rightarrow o''). \\
 & & (g \times (\mathsf{cccat}\cdot\mathsf{id}\ o')) \bullet \mathsf{cceval}_{o',o''} =_{(\mathsf{cccat}\cdot\mathsf{hom}\ (o\times o')\ o'')} f \\
 & & \rightarrow g =_{(\mathsf{cccat}\cdot\mathsf{hom}\ o\ (o'\Rightarrow o''))} \mathsf{ap}_t\ \mathsf{ccabst}_{o,o',o''}\ f\rangle
\end{array}$$

where we use the notations $o \times o'$ for $(\mathsf{cccprod}\cdot\mathsf{prodo}\ o\ o')$ and $o \Rightarrow o'$ for $(\mathsf{ccexp}\ o\ o')$ and we write the object parameters as indexes, for example we write $\mathsf{cceval}_{o',o''}$ for $(\mathsf{cceval}\ o'\ o'')$.

Note that this is a constructive definition of cartesian closed category: a cartesian closed category is a category with extra structure. So what does it mean for a category of setoids to be a cartesian closed category? The following definition provides two possible answers to the question.

Definition 15

- A category $C : \textsc{Cat}_t$ is a *cartesian closed category* if there is a $\mathscr{C} : \mathrm{CCC}_t$ such that $\mathscr{C}\cdot\mathsf{cccat} = C$.
- A pair (C, E) with $C : \textsc{Cat}_t$, $E : C\cdot\mathsf{obj} \rightarrow C\cdot\mathsf{obj} \rightarrow C\cdot\mathsf{obj}$ is a *canonical cartesian closed category* if there is a $\mathscr{C} : \mathrm{CCC}_t$ such that $\mathscr{C}\cdot\mathsf{cccat} = C$ and $\mathscr{C}\cdot\mathsf{ccexp} = E$.

We use the terminology "*canonical* cartesian closed category" to specify that exponents are given canonically by an exponent function, and not just required to exist. Clearly C is a cartesian closed category iff (C, E) is a canonical cartesian closed category for some exponent function E.

Lemma 16
$(\mathbf{TSet}, \lambda A, B : \textsc{Set}_t.\ \textsc{Map}_t\ A\ B)$ is a canonical cartesian closed category.

3.2 The categories of partial setoids

We now turn to categories of partial setoids. Their definition is similar to that of **TSet**, but rely on a different formalism. Indeed, we have seen in the previous subsection that setoids and morphisms of setoids must come before and be the base of category theory. Consequently, we will have different notions of category according to the notion of setoid and setoid morphism we assume. More precisely, we will have four versions of category theory that we call P-category theory, Q-category theory, R-category theory and S-category theory; and inside each of these settings we can define the categories **PSet**, **QSet**, **RSet** and **SSet**, respectively. These categories will play a similar role as the one played by the category **Set** in standard category theory.

Definition 17
Let X range over P, Q, R and S. The type $\textsc{Cat}_x$ of *X-categories* is defined as the record type

$$\begin{array}{rcl} \langle \mathsf{obj} & : & \mathbf{Type}_1, \\ \mathsf{hom} & : & \mathsf{obj} \to \mathsf{obj} \to \textsc{Set}_x, \\ \bullet & : & \Pi A, B, C : \mathsf{obj}.\ \textsc{bmap}_x\ (\mathsf{hom}\ A\ B)\ (\mathsf{hom}\ B\ C)\ (\mathsf{hom}\ A\ C), \\ \mathsf{id} & : & \Pi A : \mathsf{obj}.\ \mathsf{el}_x\ (\mathsf{hom}\ A\ A), \\ \mathsf{catlaw} & : & \phi_{\mathrm{cat}} \rangle \end{array}$$

where ϕ_{cat} is suitably defined.

Other notions, and in particular the notion of cartesian closed category can be adapted likewise.

Definition 18

- The P-category **PSet** takes as objects elements of $\textsc{Set}_p$ and as homset between A and B the setoid $\textsc{Map}_p\ A\ B$.
- The Q-category **Qset** takes as objects elements of $\textsc{Set}_p$ and as homset between A and B the setoid $\textsc{Map}_q\ A\ B$.
- The R-category **RSet** takes as objects elements of $\textsc{Set}_p$ and as homset between A and B the setoid $\textsc{Map}_r\ A\ B$.
- The S-category **SSet** takes as objects elements of $\textsc{Set}_p$ and as homset between A and B the setoid $\textsc{Map}_s\ A\ B$.

In the following lemma, we are interested in determining whether the categories **PSet**, **QSet**, **RSet** and **SSet** with their associated function space setoid, as defined in the previous section, form canonical cartesian closed categories.

Lemma 19
The following pairs are canonical cartesian closed categories:

- $(\mathbf{PSet}, \lambda A, B : \textsc{Set}_p.\ \textsc{Map}_r\ A\ B)$
- $(\mathbf{QSet}, \lambda A, B : \textsc{Set}_p.\ \textsc{Map}_q\ A\ B)$
- $(\mathbf{RSet}, \lambda A, B : \textsc{Set}_p.\ \textsc{Map}_r\ A\ B)$
- $(\mathbf{SSet}, \lambda A, B : \textsc{Set}_p.\ \textsc{Map}_s\ A\ B)$

It is interesting to note that $(\mathbf{PSet}, \lambda A, B : \textsc{Set}_p.\ \textsc{Map}_p\ A\ B)$ is not a canonical cartesian closed category. This can be established by exploiting anomalies related to "empty" partial setoids.

Definition 20
For any type T, the empty partial setoid $\emptyset\ T$ over T is defined by

$$\begin{array}{rlcl} \emptyset\ T = \langle & \mathsf{el}_p & = & T, \\ & \mathsf{eq}_p & = & \lambda x, y : T.\ \bot, \\ & \mathsf{per} & = & \langle\ \ \mathsf{sym}_p\ = \ \lambda x, y : T.\ \lambda\ p : \bot.\ p, \\ & & & \quad\ \mathsf{trans}_p\ = \ \lambda x, y : T.\ \lambda\ p, q : \bot.\ p\rangle\rangle \end{array}$$

Now let $A = \emptyset$ Unit and let $\mathbf{1}_p = \langle \mathsf{el}_p = \mathsf{Unit}, \mathsf{eq}_p = \lambda x, y : \mathsf{Unit}.\ \top, \mathsf{per} = \cdots\rangle$. The type $\textsc{map}_p\ A\ (\textsc{Map}_p\ \mathbf{1}_p\ A)$ is empty: indeed, let $f : \textsc{map}_p\ A\ (\textsc{Map}_p\ \mathbf{1}_p\ A)$. Then $((f \cdot \mathsf{ap}_p\ *) \cdot \mathsf{ext}_p\ *\ *\ !) : \bot$, where $*$ is the only inhabitant of Unit and $!$ is the only proof of $\top$; this is impossible by consistency of the system. On the other hand, if there was a cartesian closed category of the form

$$\langle \mathsf{cccat} = \mathbf{PSet},\ \mathsf{ccexp} = \lambda A, B : \textsc{Set}_p.\ \textsc{Map}_p\ A\ B,\ \ldots\rangle$$

then the type $\textsc{map}_p\ A\ (\textsc{Map}_p\ \mathbf{1}_p\ A)$ would be inhabited, a contradiction. On the basis of this observation, it does not seem adequate to formalize mathematics in type theory using partial setoids and the function space setoid $\textsc{Map}_p$.

3.3 Equivalence between categories

The purpose of this paragraph is to establish whether the categories defined in the previous section are equivalent[2]. The basic conclusion, summarized in Table 1, is that there are, up to equivalences, two cartesian closed categories of setoids: **TSet**, equivalent to **QSet** and **SSet**, and **RSet**, equivalent to **PSet**. The $*$ in the comparison of **RSet** and **PSet** means that the isomorphism does not preserve function setoids.

Before proceeding any further, we would like to clarify what we mean by *comparing categories*. Formally, the standard notion of equivalence can only be used to compare categories that live in the same formalism. In particular, a standard definition of equivalence would not be appropriate to compare categories that do not belong

[2] An equivalence between two categories $\mathscr{C}$ and $\mathscr{D}$ is a tuple $\langle F, G, \eta, \epsilon\rangle$ where

- F is a functor from $\mathscr{C}$ to $\mathscr{D}$;
- G is a functor from $\mathscr{D}$ to $\mathscr{C}$;
- η is a natural isomorphism from $\mathsf{id}_{\mathscr{C}}$ to $G \circ F$;
- ϵ is a natural isomorphism from $F \circ G$ to $\mathsf{id}_{\mathscr{D}}$.

Table 1. *Comparisons between categories (EQ = equivalence; NOEQ = not equivalence; ISO = isomorphism)*

	TSet	**PSet**	**QSet**	**RSet**	**SSet**
TSet		NOEQ	EQ	NOEQ	EQ
PSet	NOEQ		NOEQ	ISO*	NOEQ
QSet	EQ	NOEQ		NOEQ	ISO
RSet	NOEQ	ISO*	NOEQ		NOEQ
SSet	EQ	NOEQ	ISO	NOEQ	

to the same framework, e.g. **TSet** which is a T-category and **RSet** which is an R-category. One remedy to this problem is to define transformations that map categories in one formalism to categories in another formalism. Going back to **TSet** and **RSet**, one can for example transform **TSet** into an R-category **TSet**$_r$ and compare it to **RSet**. Here we prefer to remain at an informal level of discussion, since our main purpose is to stress that not all existing alternatives are equivalent.

3.3.1 Comparing TSet *and* RSet

We begin by showing that **TSet** and **RSet** are not equivalent.

Theorem 21
The categories **TSet** and **RSet** are not equivalent.

Proof
Call an object o a *weak initial* (respectively, *initial*) object if for every object o' there is at most one (respectively, exactly one) morphism from o to o'. Equivalence of categories preserves weak initial and initial objects, hence the following two observations are sufficient to conclude that **TSet** and **RSet** are not equivalent:

- In **RSet**, there is a weak initial object that is not initial.
- In **TSet**, every weak initial object is initial.

For the first observation, take $\emptyset$ Unit. Since this partial setoid does not have any defined element, all functions from it to any other setoid are equal, so it is an weak initial object. On the other hand, there is no morphism from $\emptyset$ Unit to $\emptyset$ Empty, because the underlying type function should have type Unit $\rightarrow$ Empty and there is no such function.

For the second observation, let o be a weakly initial object in **TSet**. We prove that its carrier type must be the empty type. It is immediate to prove that if the carrier type of a total setoid is empty then the setoid is initial. From the fact that o is weakly initial it follows that the two constant functions $c_{\mathsf{true}}, c_{\mathsf{false}} : \text{MAP}_t\ o\ (\ddagger \mathsf{Bool})$, where Bool is the type with two elements true and false, must be equal. It immediately follows that the carrier of o is empty. □

Nevertheless, there are two obvious functors $\mathscr{TR}$: **TSet** → **RSet** and $\mathscr{RT}$: **RSet** → **TSet**. The functor $\mathscr{TR}$ is defined in the obvious way: its object part forgets the reflexivity of equality, and its arrow part forgets the extensionality of the morphism. As for $\mathscr{RT}$, it is defined below.

Definition 22

The functor $\mathscr{RT}$: **RSet** → **TSet** is defined as follows:

- Object part: if A is a partial setoid, then $\textsc{Total}\ A$ is its corresponding total setoid:

$$\begin{array}{rcrcl} \textsc{Total}\ A & = & \langle\ \mathsf{el}_t & = & \textsc{domain}\ (\textsc{Partial}\ A) \\ & & & = & \langle \mathsf{cont} : \mathsf{el}_t\ A, \mathsf{def} : \mathsf{cont} =_A \mathsf{cont}\rangle, \\ & & \mathsf{eq}_t & = & \cdots \\ & & \mathsf{er} & = & \cdots\rangle \end{array}$$

- Arrow part: if g is a defined morphism of partial setoids from A to B, i.e. $g : \textsc{map}_r\ A\ B$ and $\phi : g =_{\textsc{Map}_r\ A\ B} g$ then $\mathscr{RT}\ g$ is defined as the record

$$\begin{array}{rcl} \mathscr{RT}\ g = \langle\ \mathsf{ap}_t & = & \lambda x : \textsc{domain}\ A.\ \langle\ \mathsf{cont} \ = \ g\ x \cdot \mathsf{cont}, \\ & & \qquad\qquad\qquad\quad \mathsf{def} \ = \ \phi\ x \cdot \mathsf{cont}\ x \cdot \mathsf{cont}\ x \cdot \mathsf{def}\rangle \\ \mathsf{ext}_t & = & \lambda x, y : \textsc{domain}\ A. \phi\ x \cdot \mathsf{cont}\ y \cdot \mathsf{cont}\rangle \end{array}$$

One can show that the obvious functors between **TSet** and **RSet** cannot give an equivalence, e.g. we can prove that $\mathscr{RT}$ is not full as in Lemma 31.

3.3.2 Comparing **TSet** *and* **QSet**

There are two obvious functors $\mathscr{TQ}$: **TSet** → **QSet** and $\mathscr{QT}$: **QSet** → **TSet**. They form an equivalence pair.

Definition 23

The functor $\mathscr{TQ}$: **TSet** → **QSet** is defined as follows:

- Object part: if A is a total setoid, then $\mathscr{TQ}\ A = \textsc{Partial}\ A$ is its corresponding partial setoid (just forget the proof of reflexivity):

$$\begin{array}{rcrcl} \textsc{Partial}\ A & = & \langle\ \mathsf{el}_p & = & \mathsf{el}_t\ (\textsc{Total}\ A) = \textsc{domain}\ B, \\ & & \mathsf{eq}_p & = & \cdots \\ & & \mathsf{per} & = & \cdots\rangle \end{array}$$

- Arrow part: let A and B be total setoids and f : $\textsc{map}_t\ A\ B$. Then $\mathscr{T}\mathscr{Q}\ f$: $\textsc{map}_q\ (\mathscr{T}\mathscr{Q}\ A)\ (\mathscr{T}\mathscr{Q}\ B)$ is defined by the components

$$\begin{array}{lcl}
(\mathscr{T}\mathscr{Q}\ f)\cdot\mathsf{ap}_q & : & \Pi a : \mathsf{el}_p\ (\mathscr{T}\mathscr{Q}\ A).\ a =_{(\mathscr{T}\mathscr{Q}\ A)} a \to \mathsf{el}_p\ (\mathscr{T}\mathscr{Q}\ B) \\
 & : & \Pi a : \mathsf{el}_t\ A.a =_A a \to \mathsf{el}_t\ B \\
(\mathscr{T}\mathscr{Q}\ f)\cdot\mathsf{ap}_q & = & \lambda a : \mathsf{el}_t\ A.\ \lambda\ \phi : a =_A a.\ \mathsf{ap}_t\ f\ a \\
 & & \\
(\mathscr{T}\mathscr{Q}\ f)\cdot\mathsf{ext}_q & : & \forall x, y : \mathsf{el}_p\ (\mathscr{T}\mathscr{Q}\ A). \\
 & & \forall\phi : x =_{(\mathscr{T}\mathscr{Q}\ A)} x.\forall\psi : y =_{(\mathscr{T}\mathscr{Q}\ A)} y.\ x =_{(\mathscr{T}\mathscr{Q}\ A)} y \\
 & & \to (\mathsf{ap}_q\ (\mathscr{T}\mathscr{Q}\ f)\ x\ \phi) =_{(\mathscr{T}\mathscr{Q}\ B)} (\mathsf{ap}_q\ (\mathscr{T}\mathscr{Q}\ f)\ y\ \psi) \\
 & : & \forall x, y : \mathsf{el}_t\ A.\forall\phi : x =_A x.\forall\psi : y =_A y\ x =_A y \\
 & & \to (\mathsf{ap}_t\ f\ x) =_B (\mathsf{ap}_t\ f\ y) \\
(\mathscr{T}\mathscr{Q}\ f)\cdot\mathsf{ext}_q & = & \lambda x, y : \mathsf{el}_p\ (\mathscr{T}\mathscr{Q}\ A).\ \lambda\phi : x =_A x.\ \lambda\psi : y =_A y. \\
 & & \quad f\cdot\mathsf{ext}_t\ x\ y
\end{array}$$

The functor $\mathscr{Q}\mathscr{T}$: **QSet** → **TSet** is defined as follows:

- Object part: if A is a partial setoid, then $\mathscr{Q}\mathscr{T}\ A = \textsc{Total}\ A$ is its corresponding total setoid.
- Arrow part: let A and B be partial setoids and f : $\textsc{map}_q\ A\ B$. Then $\mathscr{Q}\mathscr{T}\ f$: $\textsc{map}_t\ (\mathscr{Q}\mathscr{T}\ A)\ (\mathscr{Q}\mathscr{T}\ B)$ is defined by the components

$$\begin{array}{lcl}
(\mathscr{Q}\mathscr{T}\ f)\cdot\mathsf{ap}_t & : & \mathsf{el}_t\ (\mathscr{Q}\mathscr{T}\ A) \to \mathsf{el}_t\ (\mathscr{Q}\mathscr{T}\ B) \\
 & : & \textsc{domain}\ A \to \textsc{domain}\ B \\
(\mathscr{Q}\mathscr{T}\ f)\cdot\mathsf{ap}_t & = & \lambda x : \textsc{domain}\ A.\langle\ \mathsf{cont} = \mathsf{ap}_q\ f\ x\cdot\mathsf{cont}\ x\cdot\mathsf{def}, \\
 & & \qquad\qquad\qquad\quad \mathsf{def} = f\cdot\mathsf{ext}_q\ x\cdot\mathsf{cont}\ x\cdot\mathsf{cont} \\
 & & \qquad\qquad\qquad\qquad\qquad\quad x\cdot\mathsf{def}\ x\cdot\mathsf{def}\ x\cdot\mathsf{def}\rangle \\
 & & \\
(\mathscr{Q}\mathscr{T}\ f)\cdot\mathsf{ext}_t & : & \forall x, y : \mathsf{el}_t\ (\mathscr{Q}\mathscr{T}\ A).(x =_{(\mathscr{Q}\mathscr{T}\ A)} y) \\
 & & \to \mathsf{ap}_t\ (\mathscr{T}\mathscr{Q}\ f)\ x =_{(\mathscr{T}\mathscr{Q}\ B)} \mathsf{ap}_t\ (\mathscr{T}\mathscr{Q}\ f)\ y \\
 & : & \forall x, y : \textsc{domain}\ A.x\cdot\mathsf{cont} =_A y\cdot\mathsf{cont} \to \\
 & & \quad (\mathsf{ap}_q\ f\ x\cdot\mathsf{cont}\ x\cdot\mathsf{def}) =_B (\mathsf{ap}_q\ f\ y\cdot\mathsf{cont}\ y\cdot\mathsf{def}) \\
(\mathscr{Q}\mathscr{T}\ f)\cdot\mathsf{ext}_t & = & \lambda x, y : \textsc{domain}\ A.f\cdot\mathsf{ext}_q\ x\cdot\mathsf{cont}\ y\cdot\mathsf{cont}\ x\cdot\mathsf{def}\ y\cdot\mathsf{def}
\end{array}$$

To show, that these functors form an equivalence we look at their two compositions and show that they are naturally isomorphic to the identity functor in the respective categories.

Theorem 24

The categories **TSet** and **QSet** are equivalent.

Proof

We shall define $\langle\mathscr{T}\mathscr{Q}, \mathscr{Q}\mathscr{T}, \eta, \epsilon\rangle$ is an equivalence between the categories.

Let A : **TSet**, then we have

$$\begin{array}{lcl}
\mathscr{Q}\mathscr{T}\ (\mathscr{T}\mathscr{Q}\ A) & : & \mathbf{TSet} \\
 & = & \textsc{Total}\ (\textsc{Partial}\ A)
\end{array}$$

Let B : **QSet**, then we have

$$\begin{array}{rcl} \mathscr{T}\mathscr{Q}\ (\mathscr{Q}\mathscr{T}\ B) & : & \mathbf{QSet} \\ & = & \textsc{Partial}\ (\textsc{Total}\ B) \end{array}$$

The unit of the equivalence is then defined, for every A : **TSet** as the morphism

$$\begin{array}{lcllcl} \eta_A & : & \multicolumn{4}{l}{\textsc{map}_t\ A\ (\mathscr{Q}\mathscr{T}\ (\mathscr{T}\mathscr{Q}\ A))} \\ \eta_A & = & \langle & \mathsf{ap}_t & : & \mathsf{el}_t\ A \rightarrow \mathsf{el}_t\ (\mathscr{Q}\mathscr{T}\ (\mathscr{T}\mathscr{Q}\ A)) \\ & & & & : & \mathsf{el}_t\ A \rightarrow \langle \mathsf{cont} : \mathsf{el}_t\ A, \mathsf{def}\ : \mathsf{cont} =_A \mathsf{cont}\rangle \\ & & & & = & \lambda x : \mathsf{el}_t\ A.\langle \mathsf{cont} = x, \mathsf{def}\ : A \cdot \mathsf{er} \cdot \mathsf{refl}_t\ x\rangle, \\ & & & \mathsf{ext}_t & = & \cdots\rangle \end{array}$$

η is a natural isomorphism between the identity functor $\mathscr{I}_{\mathbf{TSet}}$ and the functor $\mathscr{Q}\mathscr{T} \circ \mathscr{T}\mathscr{Q}$.

The counit of the equivalence is defined, for every B : **QSet** as the morphism

$$\begin{array}{lcllcl} \epsilon_B & : & \multicolumn{4}{l}{\textsc{map}_q\ (\mathscr{T}\mathscr{Q}\ (\mathscr{Q}\mathscr{T}\ B))\ B} \\ \epsilon_B & = & \langle & \mathsf{ap}_q & : & \Pi x : \mathsf{el}_p\ (\mathscr{T}\mathscr{Q}\ (\mathscr{Q}\mathscr{T}\ B)).x =_{(\mathscr{T}\mathscr{Q}\ (\mathscr{Q}\mathscr{T}\ B))} x \rightarrow_p\ \mathsf{el}_p\ B \\ & & & & : & \Pi x : \textsc{domain}\ B.x =_{(\mathscr{T}\mathscr{Q}\ (\mathscr{Q}\mathscr{T}\ B))} x \rightarrow_p\ \mathsf{el}_p\ B \\ & & & & = & \lambda x : \textsc{domain}\ B.\lambda\phi : x =_{(\mathscr{T}\mathscr{Q}\ (\mathscr{Q}\mathscr{T}\ B))} x.x \cdot \mathsf{cont}, \\ & & & \mathsf{ext}_q & = & \cdots\rangle \end{array}$$

ϵ is a natural isomorphism between the functor $\mathscr{T}\mathscr{Q} \circ \mathscr{Q}\mathscr{T}$ and the identity functor $\mathscr{I}_{\mathbf{QSet}}$. □

3.3.3 *Comparing* TSet *and* SSet

The categories **TSet** and **SSet** are equivalent. The shortest way to show this fact is to prove that **SSet** is equivalent to **QSet**, and then we obtain the equivalence to **TSet** by Theorem 24. There are two obvious functors $\mathscr{Q}\mathscr{S}$: **QSet** → **SSet** and $\mathscr{S}\mathscr{Q}$: **SSet** → **QSet**. They are simply the identity on objects. On morphisms, $\mathscr{Q}\mathscr{S}$ just forgets the proof component ext_q, while $\mathscr{S}\mathscr{Q}$ on the defined elements of $\textsc{Map}_s\ A\ B$, that is the functions $f : \textsc{map}_s\ A\ B$ for which there is a proof $\xi : f =_{\textsc{Map}_s\ A\ B} f$, is

$$\mathscr{S}\mathscr{Q}\ f\ \xi = \langle \mathsf{ap}_q = f, \mathsf{ext}_q = \xi \rangle$$

$\mathscr{Q}\mathscr{S}$ and $\mathscr{Q}\mathscr{S}$ are inverse of each other, so

Theorem 25
The categories **SSet** and **QSet** are isomorphic. The categories **SSet** and **TSet** are equivalent.

3.3.4 *Comparing* PSet *and* RSet

We proved in Lemma 19 that **PSet** cannot be completed to a cartesian closed category having $\textsc{Map}_p\ A\ B$ as exponent object. Therefore there are no equivalences between **PSet** and any of the other four categories that preserves the setoid of functions. However, there is an equivalence between **PSet** and **RSet** that does not preserve the setoid of functions.

Definition 26
The functor $\mathscr{PR}$: **PSet** → **RSet** is defined as follows:

- Object part: the identity.
- Arrow part: if f is a morphism of partial setoids from A to B in **PSet**, i.e. f : $\textsc{map}_p\ A\ B$, then $\mathscr{PR}\ f$ is simply $f \cdot \mathsf{ap}_p$.

The functor $\mathscr{RP}$: **RSet** → **PSet** is defined as follows:

- Object part: the identity.
- Arrow part: if f is a morphism of partial setoids from A to B in **PSet**, i.e. f : $\textsc{map}_r\ A\ B$ such that $f =_{\textsc{Map}_r\ A\ B} f$, then $\mathscr{RP}\ f$ is the record

$$\mathscr{RP}\ f = \langle \mathsf{ap}_p = f, \mathsf{ext}_p = \xi \rangle$$

where ξ is a proof that f is a defined element of $\textsc{Map}_r\ A\ B$, i.e. ξ : $(f =_{\textsc{Map}_r\ A\ B} f) = \forall x, y : \mathsf{el}_p\ A.\ (x =_A y) \to (\mathsf{ap}_p\ f\ x) =_B (\mathsf{ap}_p\ f\ y)$.

It is trivial to verify that $\mathscr{PR}$ and $\mathscr{RP}$ with the identity natural transformations form an equivalence, actually even an isomorphism, between **PSet** and **RSet**.

Theorem 27
The categories **PSet** and **RSet** are isomorphic.

3.3.5 *Comparing* **TSet** *and* **PSet**

That **TSet** and **PSet** are not equivalent follows from the equivalence of **PSet** to **RSet**, Theorem 27, and the fact that **TSet** is not equivalent to **RSet**, Theorem 21. There are, however, two obvious functors $\mathscr{TP}$: **TSet** → **PSet** and $\mathscr{PT}$: **PSet** → **TSet**. As expected, neither $\mathscr{TP}$ nor $\mathscr{PT}$ yields an equivalence of categories.

The functor $\mathscr{TP}$: **TSet** → **PSet** is defined in the obvious way: its object part turns a total setoid into a partial setoid simply by forgetting about the reflexivity of equality, whereas its arrow part is the "identity".

Definition 28
The functor $\mathscr{TP}$: **TSet** → **PSet** is defined as follows:

- Object part: if A is a total setoid, then

$$\mathscr{TP}\ A = \textsc{Partial}\ A = \langle \begin{array}{lcl} \mathsf{el}_p & = & A \cdot \mathsf{el}_t, \\ \mathsf{eq}_p & = & A \cdot \mathsf{eq}_t, \\ \mathsf{per} & = & \ldots \rangle \end{array}$$

is its corresponding partial setoid.
- Arrow part: the arrow part of $\mathscr{TP}$ is the "identity".

$\mathscr{TP}$ cannot induce any equivalence between **TSet** and **PSet**, as shown by the following lemma.

Lemma 29
There exists a partial setoid A : $\textsc{Set}_p$ that is not isomorphic to the image of any total setoid under $\mathscr{TP}$.

Proof
We exhibit a partial setoid $A : \textsc{Set}_p$ such that for every $B : \textsc{Set}_t$, it must be that either $\textsc{Map}_p\ A\ (\mathscr{TP}\ B)$ or $\textsc{Map}_p\ (\mathscr{TP}\ B)\ A$ is empty. Indeed, assume $\vdash t : T$ and let $A = \emptyset\ T$. Now assume $B : \textsc{Set}_t$. We claim that in the empty context:

- $\textsc{Map}_p\ (\mathscr{TP}\ B)\ A$ is not empty iff B is empty. Indeed, if $b : \mathsf{el}_t\ B$ and $f : \textsc{map}_p\ (\mathscr{TP}\ B)\ A$ then $f \cdot \mathsf{ext}_p\ b\ b\ (B \cdot \mathsf{er} \cdot \mathsf{refl}_t\ b) : \bot$, which is impossible by consistency of the system.
- $\textsc{Map}_p\ A\ (\mathscr{TP}\ B)$ is not empty iff B is not empty.

The result follows. □

The functor $\mathscr{PT}$ mapping partial setoids to total setoids takes as object part the function mapping a partial setoid to its domain setoid, and as function part the corresponding transformation described below.

Definition 30
The functor $\mathscr{PT} : \mathbf{PSet} \to \mathbf{TSet}$ is defined as follows:

- Object part: if A is a partial setoid, then

$$
\begin{array}{l}
\mathscr{PT}\ A = \textsc{Total}\ A = \\
\quad \begin{array}{rcl}
\langle\ \mathsf{el}_t & = & \textsc{domain}\ A, \\
\mathsf{eq}_t & = & \lambda x, y : \textsc{domain}\ A.\ x \cdot \mathsf{cont} =_A y \cdot \mathsf{cont}, \\
\mathsf{er} & = & \langle \begin{array}[t]{rcl}
\mathsf{refl}_t & = & \lambda x : \textsc{domain}\ A.\ x \cdot \mathsf{def}, \\
\mathsf{sym}_t & = & \lambda x, y : \textsc{domain}\ A.\ A \cdot \mathsf{per} \cdot \mathsf{sym}_p\ x \cdot \mathsf{cont}\ y \cdot \mathsf{cont}, \\
\mathsf{trans}_t & = & \lambda x, y, z : \textsc{domain}\ A. \\
 & & A \cdot \mathsf{per} \cdot \mathsf{trans}_p\ x \cdot \mathsf{cont}\ y \cdot \mathsf{cont}\ z \cdot \mathsf{cont}\rangle\rangle
\end{array}
\end{array}
\end{array}
$$

  is its corresponding total setoid—$\textsc{Domain}\ A$ differs from $\textsc{Total}\ A$ by the name of its fields.
- Arrow part: if g is a morphism of partial setoids from A to B, i.e. $g : \textsc{map}_p\ A\ B$ then $\mathscr{PT}\ g$ is defined as the record

$$
\begin{array}{rcl}
\mathscr{PT}\ g = \langle\ \mathsf{ap}_t & = & \lambda x : \textsc{domain}\ A. \\
 & & \langle \begin{array}[t]{rcl}
 \mathsf{cont} & = & \mathsf{ap}_p\ g\ x \cdot \mathsf{cont}, \\
 \mathsf{def} & = & g \cdot \mathsf{ext}_p\ x \cdot \mathsf{cont}\ x \cdot \mathsf{cont}\ x \cdot \mathsf{def}\rangle,
 \end{array} \\
\mathsf{ext}_t & = & \lambda x, y : \textsc{domain}\ A.\ g \cdot \mathsf{ext}_p\ x \cdot \mathsf{cont}\ y \cdot \mathsf{cont}\rangle
\end{array}
$$

$\mathscr{PT}$ cannot induce any equivalence between **TSet** and **PSet**, as shown by the following lemma.

Lemma 31
The functor $\mathscr{PT}$ is not full, i.e. there exist two partial setoids A and B such that $\textsc{map}_p\ A\ B$ is empty but $\textsc{map}_t\ (\mathscr{PT}\ A)\ (\mathscr{PT}\ B)$ is not.

Proof
Assume $\vdash t : T$ and let $A = \emptyset\ T$ and $B = \emptyset\ (\textsc{domain}\ A)$. We claim that in the empty context:

- there is no $g : \textsc{map}_p\ A\ B$. If there were such a g, then $(\mathsf{ap}_p\ g\ t) \cdot \mathsf{def} : \bot$. This is impossible by consistency of the system;

- there is a $g : \text{MAP}_t\ (\textsc{Total}\ A)\ (\textsc{Total}\ B)$. Indeed, consider

$$\begin{array}{rcl} g = \langle\ \mathsf{ap}_t & = & \lambda x : \text{DOMAIN}\ A.\ \langle \mathsf{cont} = x,\ \mathsf{def} = x \cdot \mathsf{def}\rangle, \\ \mathsf{ext}_t & = & \lambda x, y : \text{DOMAIN}\ A.\ \lambda p : \bot.\ p\rangle \end{array}$$

The expression g is indeed of type $\text{MAP}_t\ (\textsc{Total}\ A)\ (\textsc{Total}\ B)$ since we have

$$\begin{array}{rcl} \mathsf{el}_t\ (\textsc{Total}\ A) & =_\beta & \text{DOMAIN}\ A \\ \mathsf{el}_t\ (\textsc{Total}\ B) & =_\beta & \text{DOMAIN}\ (\text{DOMAIN}\ A) \\ (x =_{(\textsc{Total}\ A)} y) & =_\beta & \bot \\ (\mathsf{ap}_t\ x =_{(\textsc{Total}\ B)} \mathsf{ap}_t\ y) & =_\beta & \bot \end{array}$$

□

3.4 Discussion

We have seen that there are up to equivalence two possible choices for a category of setoids, namely **TSet** and **RSet**. In the coming sections, we compare these approaches from the point of view of formalizing mathematics.

One could also compare these categories from other perspectives, e.g. one could check which of the categories **TSet** and **RSet** do form a model of dependent type theory. This issue has been investigated in depth by Hofmann (Hofmann, 1995b). It turns out that the category **RSet** does form a model of dependent type theory, whereas there are some difficulties with **TSet**. In particular, it is problematic to define a family of setoids depending on a setoid. One could also check which of the categories **TSet** and **RSet** do form a model of intuitionistic set theory, i.e. a topos (Lambek & Scott, 1986). It turns out that none of the categories forms a topos because of the distinction between total functional relations and functions. In Section 4, we study choice principles which turn these categories into toposes.

4 Choice principles

To pursue the analysis of setoids as a type-theoretic formalization of the notion of set, we study their behaviour in relation to two choice principles: the axiom of choice and the axiom of description, also called axiom of unique choice.

Before proceeding any further, we dispose of a possible criticism regarding the relevance of this enquiry: one may argue that the use of axioms, and in particular choice axioms, inside the theory of setoids is methodologically unjustified. In fact, setoids where devised to develop mathematics in type theory without external assumptions; if we are to assume axioms, we may as well assume all the axioms of set theory and dispense with setoids. While we agree that one should try to develop mathematics using only the constructions available in type theory, we still defend the importance of knowing the relation of the notion of setoid with choice principles, because this relation tells us much about the nature of setoids. Besides, as we will show, some seemingly natural choice principles are inconsistent in **RSet**. While one may not be interested in adding choice axioms, the fact that some choice principles are provably false is undesirable.

We consider the full axiom of choice and the axiom of descriptions, or unique choice – the former leads to classical logic, whereas the second is constructively valid. The axiom of descriptions is consistent in **TSet**, while in **RSet** there are two versions of it, the stronger one is inconsistent and the weaker one is too weak in that some very natural functions cannot be defined in it. Our conclusion is that **TSet** behaves better than **RSet** with respect to choice principles.

4.1 The axiom of choice for types

The axiom of choice for types expresses that every total relation from U to V yields a type-theoretic function of type $U \to V$.

Definition 32

Let U and V be two types. The type $\textsc{tr}\ U\ V$ of total relations from U to V is defined as the record type

$$\begin{array}{rrcl} \textsc{tr}\ U\ V = \langle & \mathsf{rel} & : & U \to V \to \mathbf{Prop}, \\ & \mathsf{total} & : & \forall x : U. \exists y : V.\ \mathsf{rel}\ x\ y \rangle \end{array}$$

The axiom of choice for types is given by the context Γ_{ACT}

$$\begin{array}{lcl} \mathrm{ACT}_{make} & : & \Pi U, V : \mathbf{Type}.\ (\textsc{tr}\ U\ V) \to U \to V, \\ \mathrm{ACT}_{check} & : & \forall U, V : \mathbf{Type}.\ \forall R : \textsc{tr}\ U\ V.\ \forall x : U.\ R \cdot \mathsf{rel}\ x\ (\mathrm{ACT}_{make}\ U\ V\ R\ x) \end{array}$$

The following result is well-known (Coquand, 1990; Werner, 1997).

Proposition 33

Γ_{ACT} is consistent, but not inhabited in the Calculus of Inductive Constructions.

4.2 The axiom of choice for total setoids

The axiom of choice for total setoids states that every total relation between total setoids induces a map of total setoids. A relation from A to B, where A and B are total setoids, consists of a type-theoretical relation $R : (\mathsf{el}_t\ A) \to (\mathsf{el}_t\ B) \to \mathbf{Prop}$ and a proof that R is compatible.

Definition 34

Let A and B be two total setoids. The type $\textsc{Rel}_t\ A\ B$ of relations from A to B is defined as the record type

$$\begin{array}{rrcl} \textsc{rel}_t\ A\ B = \langle & \mathsf{rel}_t & : & \mathsf{el}_t\ A \to \mathsf{el}_t\ B \to \mathbf{Prop}, \\ & \mathsf{compat}_t & : & \forall x, x' : \mathsf{el}_t\ A.\ \forall y, y' : \mathsf{el}_t\ B. \\ & & & x =_A x' \to y =_B y' \to \mathsf{rel}_t\ x\ y \to \mathsf{rel}_t\ x'\ y' \rangle \end{array}$$

An alternative definition of binary relations can be given as setoid functions with result in Ω, with Ω defined as in section 2.3. A total relation from A to B is a relation R such that for every $a : \mathsf{el}_t\ A$, there exists $b : \mathsf{el}_t\ B$ satisfying $R \cdot \mathsf{rel}_t\ a\ b$.

Definition 35

Let A and B be two total setoids. The type $\textsc{trel}_t\ A\ B$ of total relations from A to B is defined as the record type

$$\begin{array}{rlcl} \textsc{trel}_t\ A\ B = \langle & \mathsf{trel}_t & : & \textsc{rel}_t\ A\ B, \\ & \mathsf{total}_t & : & \forall x : \mathsf{el}_t\ A.\ \exists y : \mathsf{el}_t\ B.\ \mathsf{trel}_t \cdot \mathsf{rel}_t\ x\ y \rangle \end{array}$$

The axiom of choice for total setoids is given by the context Γ_{AC}

$$\begin{array}{lcl} \mathrm{AC}_{make} & : & \Pi A, B : \textsc{Set}_t.\ (\textsc{trel}_t\ A\ B) \rightarrow (\textsc{map}_t\ A\ B), \\ \mathrm{AC}_{check} & : & \forall A, B : \textsc{Set}_t.\ \forall R : \textsc{trel}_t\ A\ B.\ \forall x : \mathsf{el}_t\ A. \\ & & \quad R \cdot \mathsf{trel}_t \cdot \mathsf{rel}_t\ x\ (\mathsf{ap}_t\ (\mathrm{AC}_{make}\ A\ B\ R)\ x) \end{array}$$

The following result is well-known.

Lemma 36

The context Γ_{AC}:

1. is consistent;
2. is not instantiable in the context Γ_{ACT};
3. entails that **TSet** is a topos[3] with Ω as subobject classifier;
4. entails excluded middle, i.e. $\forall A : \mathbf{Prop}.\ A \vee \neg A$;
5. entails proof-irrelevance, i.e. $\forall A : \mathbf{Prop}. \forall x, y : A.\ x \doteq y$.

Proof sketch

The first statement is derivable from the fact that the axiom of choice for setoids holds in the proof-irrelevance model – see also Hofmann (1995b); the second statement is derivable from the fourth and the non-provability of classical logic in the context Γ_{ACT}. The third item states that **TSet** is a topos with Ω as subobject classifier and is proved by a simple calculation. The fourth item states the provability of classical logic from the axiom of choice for setoids and follows from Diaconescu's construction, e.g. see Lacas & Werner (1999) and Lambek & Scott (1986). The last item establishes that proof-irrelevance, i.e. the property that all proofs of a proposition are equal (the property was first considered by de Bruijn in the Automath project (Nederpelt *et al.*, 1994)), is derivable from the axiom of choice for setoids and can be established from Barbanera & Berardi (1996). □

We conclude this section by mentioning principles that are equivalent to the axiom of choice for setoids. It is well-known that the axiom of choice is equivalent to stating that every surjective function has a right-inverse. It is routine to define a context equivalent to Γ_{AC} that constructs for every surjective function f from A to B a function g from B to A and a proof that g is right-inverse to A. One can also give a formulation of the axiom of choice that makes use of the ϵ-monad defined in subsection 2.3; in this form the axiom states that we can exhibit an element of every non-empty predicate. The interested reader is referred to Capretta (2002).

[3] Informally, a topos $\mathscr{T}$ is a cartesian closed category with a subobject classifier, i.e. with an object that acts as a set of truth values, e.g. see Lambek & Scott (1986) for a precise definition.

4.3 The axiom of descriptions for total setoids

As seen above, the axiom of choice for setoids is extremely powerful. A weaker form of choice principle, acceptable intuitionistically, is the axiom of descriptions, a.k.a. the axiom of unique choice, which states that every total functional relation induces a map.

4.3.1 The axiom and its consistency

A total functional relation from A to B, where A and B are total setoids, consists of a relation R on A and B and proofs that R is total and functional.

Definition 37
Let A and B be two total setoids. The type $\textsc{tfrel}_t\ A\ B$ of total functional relations from A to B is defined as the record type

$$\begin{array}{rrcl} \textsc{tfrel}_t\ A\ B = \langle & \mathsf{tfrel}_t & : & \textsc{Rel}_t\ A\ B, \\ & \mathsf{ttotal}_t & : & \forall x : \mathsf{el}_t\ A.\ \exists y : \mathsf{el}_t\ B.\ \mathsf{tfrel}_t \cdot \mathsf{rel}_t\ x\ y, \\ & \mathsf{fun}_t & : & \forall x : \mathsf{el}_t\ A.\ \forall y, y' : \mathsf{el}_t\ B. \\ & & & (\mathsf{tfrel}_t \cdot \mathsf{rel}_t\ x\ y) \to (\mathsf{tfrel}_t \cdot \mathsf{rel}_t\ x\ y') \to y =_B y' \rangle \end{array}$$

The axiom of descriptions for total setoids is given by the context Γ_{AD}

$$\begin{array}{rcl} \mathrm{AD}_{make} & : & \Pi A, B : \textsc{Set}_t.\ (\textsc{tfrel}_t\ A\ B) \to (\textsc{map}_t\ A\ B), \\ \mathrm{AD}_{check} & : & \forall A, B : \textsc{Set}_t.\ \forall R : \textsc{tfrel}_t\ A\ B.\ \forall x : \mathsf{el}_t\ A. \\ & & R \cdot \mathsf{tfrel}_t \cdot \mathsf{rel}_t\ x\ (\mathsf{ap}_t\ (\mathrm{AD}_{make}\ A\ B\ R)\ x) \end{array}$$

The following result is well known.

Lemma 38
The context Γ_{AD}:

1. is instantiable in the context Γ_{ACT};
2. is consistent;
3. entails that **TSet** is a topos;
4. does not entail classical logic nor proof-irrelevance.

Proof sketch
The first item is proved by easy logical manipulations. The second item follows immediately from the consistency of Γ_{ACT}. The third item is proved by simple calculations. The last item follows from the fact that classical logic and proof-irrelevance are not derivable from Γ_{ACT}. □

As with the axiom of choice, we can express the axiom of descriptions by stating that every bijection (i.e. injective and surjective function) has an inverse, but also in terms of the ι-monad. Again, the interested reader is referred to Capretta (2002).

4.4 Choice principles for partial setoids

In this section, we focus on the axiom of descriptions for partial setoids. In fact, there are two possible formulations of the axiom, depending on the notion of total relation one adopts. It turns out that one formulation is inconsistent, and that the other is too weak. More precisely, we show that some very natural functions that can be defined on a total setoid with the axiom of descriptions cannot be defined in the corresponding partial setoid, even in presence of (the second formulation of) the axiom of descriptions. This observation leads us to the position that total setoids are better suited for the development of mathematics in type theory than partial setoids. Notice that in the following we will not use exponents for partial setoids, so our results hold both for **PSet** and **RSet**.

As suggested above, we first need to decide about the notion of total relation. A total relation from a partial setoid A to a partial setoid B can be defined in two different ways:

- (definedness-irrelevant) as a relation R such that for every $a : \mathsf{el}_p\ A$ there exists $b : \mathsf{el}_p\ B$ such that $R\ a\ b$;
- (definedness-relevant) as a relation R such that for every $a : \mathsf{el}_p\ A$ such that a is defined, i.e. $a =_A a$, there exists $b : \mathsf{el}_p\ B$ such that $R\ a\ b$ and $b =_B b$.

Each definition yields its variant of the axiom of descriptions.

4.4.1 Axiom of descriptions, definedness-relevant version

In this section, we define the definedness-relevant version of the axiom of descriptions and show it is inconsistent.

Definition 39
Let A and B be two partial setoids.

1. The type $\text{REL}_p\ A\ B$ of relations from A to B is defined as the record type

$$\begin{array}{rrcl} \text{REL}_p\ A\ B = \langle & \mathsf{rel}_p & : & \mathsf{el}_p\ A \to \mathsf{el}_p\ B \to \mathbf{Prop}, \\ & \mathsf{compat}_p & : & \forall x, x' : \mathsf{el}_p\ A.\ \forall y, y' : \mathsf{el}_p\ B. \\ & & & x =_A x' \to y =_B y' \to \mathsf{rel}_p\ x\ y \to \mathsf{rel}_p\ x'\ y' \rangle \end{array}$$

2. The type $\text{TFREL}_p\ A\ B$ of total functional relations from A to B is defined as the record type

$$\begin{array}{rrcl} \text{TFREL}_p\ A\ B = \langle & \mathsf{tfrel}_p & : & \text{REL}_p\ A\ B, \\ & \mathsf{ttotal}_p & : & \forall x : \mathsf{el}_p\ A.\ (x =_A x) \\ & & & \to \exists y : \mathsf{el}_p\ B.\ (\mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ y \wedge y =_B y), \\ & \mathsf{fun}_p & : & \forall x : \mathsf{el}_p\ A.\ \forall y, y' : \mathsf{el}_p\ B. \\ & & & (\mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ y) \to (\mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ y') \\ & & & \to x =_A x \to y =_B y' \rangle \end{array}$$

The axiom of descriptions for partial setoids is given by the context Γ_{AD}

$$\begin{array}{rcl} \text{AD}_{make} & : & \Pi A, B : \text{SET}_p.\ (\text{TFREL}_p\ A\ B) \to (\text{MAP}_p\ A\ B), \\ \text{AD}_{check} & : & \forall A, B : \text{SET}_p.\ \forall R : \text{TFREL}_p\ A\ B.\ \forall x : \mathsf{el}_p\ A. \\ & & R \cdot \mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ (\mathsf{ap}_p\ (\text{AD}_{make}\ A\ B\ R)\ x) \end{array}$$

Lemma 40
The context Γ_{AD} is inconsistent.

Proof
For every type A and partial setoid B, one can prove that the empty relation $r : \textsc{rel}_p\ (\emptyset\ A)\ B$ yields a total functional relation $r' : \textsc{tfrel}_p\ (\emptyset\ A)\ B$ and by the axiom of descriptions, a map $f : \textsc{map}_p\ (\emptyset\ A)\ B$. Now take A to be inhabited. It follows that every $\mathsf{el}_p\ B$ is inhabited, so every type B is inhabited. □

4.4.2 Axiom of descriptions, definedness-irrelevant version

In this section, we adopt the definedness-irrelevant definition of total functional relations.

Definition 41
Let A and B be two partial setoids. The type $\textsc{tfrel}_p\ A\ B$ of total functional relations from A to B is defined as the record type

$$\begin{array}{rrcl} \textsc{tfrel}_p\ A\ B = \langle & \mathsf{tfrel}_p & : & \textsc{Rel}_p\ A\ B, \\ & \mathsf{ttotal}_p & : & \forall x : \mathsf{el}_p\ A.\ \exists y : \mathsf{el}_p\ B.\ \mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ y, \\ & \mathsf{fun}_p & : & \forall x : \mathsf{el}_p\ A.\ \forall y, y' : \mathsf{el}_p\ B. \\ & & & (\mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ y) \to (\mathsf{tfrel}_p \cdot \mathsf{rel}_p\ x\ y') \\ & & & \to x =_A x \to y =_B y' \rangle \end{array}$$

Then the context Γ_{AD} is defined exactly as in the previous paragraph.

Lemma 42
The context Γ_{AD} is instantiable in the context Γ_{ACT}, and hence consistent. Further, in context Γ_{AD}, **RSet** does form a topos.

However, we show that some very natural functions that can be defined on a total setoid with the axiom of descriptions cannot be defined in the corresponding partial setoid with the axiom of descriptions. We will construct a counterexample, i.e. a function that is definable on a total setoid but not on the corresponding partial setoid.

The counterexample is given by a length function that computes the length of eventually null sequences of natural numbers (we use N to denote the type of natural numbers). We use the extensional equality on sequences: If $\sigma_1, \sigma_2 : \mathsf{N} \to \mathsf{N}$, then $(\sigma_1 =_{ext} \sigma_2) = (\forall i : \mathsf{N}.\sigma_1\ i = \sigma_2\ i)$. The total version of the setoid is

$$\begin{array}{rrcl} \mathrm{Z}\textsc{Seq}_t = \langle & \mathsf{el}_t & = & \langle \mathsf{seq} : \mathsf{N} \to \mathsf{N}, \mathsf{evz} : \exists m : \mathsf{N}.\forall i : \mathsf{N}.\ i > m \to \mathsf{seq}\ i = 0 \rangle, \\ & \mathsf{eq}_t & = & \lambda \sigma_1, \sigma_2 : \mathsf{el}_t.\sigma_1 \cdot \mathsf{seq} =_{ext} \sigma_2 \cdot \mathsf{seq}, \\ & \mathsf{er}_t & = & \cdots \rangle \end{array}$$

In the context Γ_{AD} we can define a function $\textsc{length}_t : \textsc{map}_t\ \mathrm{Z}\textsc{Seq}_t\ \mathbb{N}_t$ that gives the length of the part of a sequence that is nonzero (we use $\mathbb{N}_t$ to denote $\ddagger\mathsf{N}$, i.e. the setoid derived from N by taking Leibniz equality as book equality).

In contrast, we claim that there cannot be a version of this function if we use partial setoids. First, we let $\mathbb{N}_p = \langle \mathsf{el}_p = \mathsf{N},\ \mathsf{eq}_p = \lambda x, y : \mathsf{N}.\ x \doteq y,\ \mathsf{per} = \ldots \rangle$ be the partial setoid of natural numbers and we define the partial counterpart of $\text{ZS}\textsc{eq}_t$ as

$$\begin{aligned} \text{ZS}\textsc{eq}_p = \langle\ \ \mathsf{el}_p &= \mathsf{N} \to \mathsf{N}, \\ \mathsf{eq}_p &= \lambda\sigma_1, \sigma_2 : \mathsf{N} \to \mathsf{N}.\exists m : \mathsf{N}.\forall i : \mathsf{N}. \\ &\ (i \leqslant m \to (\sigma_1\ i) = (\sigma_2\ i)) \wedge (i > m \to (\sigma_1\ i) = 0 \wedge (\sigma_2\ i) = 0), \\ \mathsf{per} &= \cdots\rangle \end{aligned}$$

Of course we could define $\text{ZS}\textsc{eq}_p$ from $\text{ZS}\textsc{eq}_t$ by simply forgetting the proof of reflexivity, but what we want to stress here is that the idea of using the book equality to restrict the domain, which is the main advantage of partial setoids, does not always work as desired.

Now we claim that there cannot be a version of the length function for $\text{ZS}\textsc{eq}_p$ in context Γ_{AD}.

Proposition 43
The existence of the function $\textsc{length}_p : (\textsc{map}_p\ \text{ZS}\textsc{eq}_p\ \mathbb{N}_p)$ is not derivable in context Γ_{AD}.

The remaining of the paragraph is devoted to a proof of the proposition. We proceed by defining a context Γ_{CP} such that $\Gamma_{\text{AD}}, \Gamma_{\text{CP}}$ is consistent and in which one can prove the length function does not exist. The context Γ_{CP}, which captures the continuity principle[4], is defined as

$$\begin{aligned} \text{CP}_{make} &: ((\mathsf{N} \to \mathsf{N}) \to \mathsf{N}) \to (\mathsf{N} \to \mathsf{N}) \to \mathsf{N} \\ \text{CP}_{check} &: \forall F : (\mathsf{N} \to \mathsf{N}) \to \mathsf{N}.\ \forall \alpha, \beta : \mathsf{N} \to \mathsf{N}. \\ &\ (\forall i : \mathsf{N}.\ (i \leqslant \text{CP}_{make}\ F\ \alpha) \to \beta\ i = \alpha\ i) \to F\ \beta = F\ \alpha \end{aligned}$$

Lemma 44
The context $\Gamma_{\text{AD}}, \Gamma_{\text{CP}}$ is consistent.

Proof
The context is valid in the realizability model. □

Further, we show that in the context Γ_{CP} it is contradictory to assume the existence of the length function.

Proposition 45

$$\begin{aligned} \Gamma_{\text{CP}} \vdash\ & \neg\exists\textsc{length}_p : (\textsc{map}_p\ \text{ZS}\textsc{eq}_p\ \mathbb{N}_p).\ \forall\sigma : (\text{ZS}\textsc{eq}_p \cdot \mathsf{el}_p).\sigma =_{\text{ZS}\textsc{eq}_p} \sigma \to \\ & \sigma\ (\textsc{length}_p \cdot \mathsf{app}_p\ \sigma) \neq 0\ \wedge \\ & \forall i : \mathsf{N}.i > (\textsc{length}_p \cdot \mathsf{app}_p\ \sigma) \to \sigma\ i = 0 \end{aligned}$$

[4] The continuity principle is a well-known principle in intuitionistic mathematics. In constructive recursion theory, it follows from the Kreisel–Lacombe–Shoenfield theorem (see Troelstra & van Dalen (1988) and Chapter 16 of Beeson (1985)). In type theory, the continuity principle is stated as follows: For every operator $F : (\mathsf{N} \to \mathsf{N}) \to \mathsf{N}$ and for every sequence $\alpha : \mathsf{N} \to \mathsf{N}$, there exists a natural number m such that, for every other sequence $\beta : \mathsf{N} \to \mathsf{N}$ that is equal to α for indexes up to m, that is, $(\beta\ i) = (\alpha\ i)$ for $i \leqslant m$, we have that $F(\beta) = F(\alpha)$. Notice that the Continuity Principle is not provable in type theory, but it is a meta-result that holds for the operators definable in type theory. For our purpose, however, it is sufficient to know that the continuity principle is valid in the realizability model, in which the axiom of descriptions is also valid.

Proof

If there were a function $\textsc{length}_p : \textsc{map}_p\ \mathrm{ZS\textsc{eq}}_p\ \mathbb{N}_p$, then $\textsc{length}_p{\cdot}\mathsf{ap}_p : (\mathsf{N} \to \mathsf{N}) \to \mathsf{N}$ would be an extension of $\textsc{length}$ to all sequences. We can apply the Continuity Principle to the operator $\textsc{length}_p \cdot \mathsf{ap}_p$ and the constantly zero sequence $\bar{0}$. Hence, there must exist a natural number m such that, for all $\beta : \mathsf{N} \to \mathsf{N}$ satisfying $(\forall i : \mathsf{N}.\ (i \leqslant m) \to \beta\ i = \bar{0}\ i = 0)$, we have $(\mathsf{ap}_p\ \textsc{length}_p\ \beta = \mathsf{ap}_p\ \textsc{length}_p\ \bar{0} = 0)$. Now consider the sequence γ defined by

$$\gamma\ i = \begin{cases} 0 & \text{if } i \leqslant m \\ 1 & \text{if } i = m+1 \\ 0 & \text{if } i > m+1. \end{cases}$$

The sequence γ coincides with $\bar{0}$ on the first m elements, and thus, for the consequence of the continuity principle $\mathsf{ap}_p\ \textsc{length}_p\ \gamma = 0$. On the other hand, γ becomes eventually zero only after the element $m{+}1$, so we should have $\mathsf{ap}_p\ \textsc{length}_p\ \gamma = m{+}1$. We reached a contradiction, so our assumption that $\textsc{length}_p$ could be constructed is confuted. □

Notice that the crux of the counterexample is that, in the total setoid $\mathrm{ZS\textsc{eq}}_t$, the carrier type already contains information on when the sequence becomes eventually zero. We can use this information to define $\textsc{length}_t$. This information is not present in the domain of the partial setoid $\mathrm{ZS\textsc{eq}}_p$, making it impossible to define the function $\textsc{length}_p$.

5 Mathematical constructions with setoids

In this section, we study the type-theoretical analogous of two basic set-theoretical constructions, namely subsets and quotients, using setoids as an implementation of the intuitive idea of sets. In particular, we will show that the partial setoid methodology runs into practical difficulties when dealing with subsetoids. More specifically, we will see that the canonical definition of subsetoid in **RSet** is too weak, in the sense that if we use this definition, some natural functions on subsetoids cannot be defined, while they are definable in the corresponding total subsetoid.

5.1 Subsetoids

Total setoids, as embodied by **TSet**, and partial setoids, as embodied in **RSet**, are based on two distinct ways of restricting the domain of a structure, that is, of defining subsetoids. In the first case, restriction is achieved by modifying the underlying carrier of the setoid, while in the second case, restriction is achieved by modifying the setoid's underlying equality relation. However, we will show that total setoids are unavoidable, in the sense that, even if we use partial setoids, we will be forced to restrict the carrier type of a setoid to obtain certain subsetoids.

We begin by reviewing the definition of subsetoids in the context of total setoids. Intuitively a subsetoid is that part of a setoid whose elements satisfy a predicate. Predicates on setoids are defined as type-theoretic predicates on the carrier sets that are invariant for the setoid equality.

Definition 46
Let A be a total setoid. *Setoid predicates* over A are the elements of the record type

$$\begin{aligned}\textsc{pred}_t\ A = \langle\ \ \mathsf{pf}_t &: \mathsf{el}_t\ A \to \mathbf{Prop},\\ \mathsf{inv}_t &: \forall x, y : \mathsf{el}_t\ A.\ (x =_A y) \to (\mathsf{pf}_t\ x \to \mathsf{pf}_t\ y)\rangle\end{aligned}$$

Note that an equivalent definition would be $\textsc{pred}_t\ A = \textsc{map}_t\ A\ \Omega$. In **TSet** the subsetoid of a setoid A defined by a predicate $P : \textsc{pred}_t\ A$ is obtained by first restricting the carrier type, and then constructing the setoid over this carrier by projecting the equality of A on the first component.

Definition 47
Let $A : \textsc{Set}_t$ and $P : \textsc{pred}_t\ A$. The carrier of the subsetoid selected by P from A is

$$\textsc{subCarrier}\ A\ P = \langle \mathsf{subel} : \mathsf{el}_t\ A, \mathsf{insub} : P \cdot \mathsf{pf}_t\ \mathsf{subel}\rangle$$

and the subsetoid is

$$\begin{aligned}\textsc{subSetoid}_t\ A\ P = \langle\ \ &\mathsf{el}_t = \textsc{subCarrier}\ A\ P,\\ &\mathsf{eq}_t = \lambda x, y : \mathsf{el}_t.(\mathsf{subel}\ x) =_A (\mathsf{subel}\ y),\\ &\mathsf{er} = \cdots\rangle\end{aligned}$$

Sometimes (if the axiom of descriptions is not supposed to be true) it is necessary to use a constructively stronger notion of predicate and subsetoid: the carrier of the predicate has type $\mathsf{el}_t\ A \to \mathbf{Type}$ so that its proofs can be used to construct elements of types. The rest of the definition is in this case the same, except for the substitution of **Type** for **Prop**. For the examples given below we assume either that this constructive definition is used or that the axiom of description (equivalently, the axiom of choice for types) is assumed.

On the other hand, when using partial setoids, we do not change the underlying type, but we modify the equality. Predicates over partial setoids are defined in the same way as predicates over total setoids.

Definition 48
Let A be a partial setoid. *Setoid predicates* over A are the elements of the record type

$$\begin{aligned}\textsc{pred}_p\ A = \langle\ \ \mathsf{pf}_p &: \mathsf{el}_p\ A \to \mathbf{Prop},\\ \mathsf{inv}_p &: \forall x, y : \mathsf{el}_p\ A.\ (x =_A y) \to (\mathsf{pf}_p\ x \to \mathsf{pf}_p\ y)\rangle\end{aligned}$$

The propositional function pf_p must be defined on the whole carrier type $\mathsf{el}_p\ A$, even on elements x for which $x =_A x$ is not true.

Definition 49
If A is a partial setoid and $P : \textsc{pred}_p\ A$, then we define the subsetoid of A selected by P as

$$\textsc{subSetoid}_p\ A\ P = \langle \mathsf{el}_p = \mathsf{el}_p\ A, \mathsf{eq}_p = \lambda x, y : \mathsf{el}_p\ A.(P \cdot \mathsf{pf}_p\ x) \wedge x =_A y, \mathsf{per} = \cdots\rangle$$

In the definition of eq_p we do not require $(P \cdot \mathsf{pf}_p\ y)$ because it is derivable from $(P \cdot \mathsf{pf}_p\ x)$, $x =_A y$, and $P \cdot \mathsf{inv}_p$.

This definition has the nice property that an element of the carrier of the subsetoid is automatically an element of the carrier of the setoid. However, a serious drawback of this approach consists in the fact that a function defined on a subsetoid of A must be a type-theoretic function defined on the entire carrier type of A. In some cases, this cannot be done and the use of $\textsc{subSetoid}_t\ A\ P$ is unavoidable. A first example of the above is given by the length function in the example of eventually null sequences that we developed in Subsection 4.4. Indeed, one can define the setoid of sequences as $\textsc{Map}_r\ \mathbb{N}_p\ \mathbb{N}_p$, define the predicate of being eventually zero and then form the subsetoid of eventually zero sequences. Using the results of the previous section, this gives us a first example of a function that cannot be defined using subsetoids *à la* partial setoid.

Below we develop a second example based on the real numbers. Here the idea is to define a setoid of real numbers and then restrict the setoid to smaller systems, say, the rationals or the natural numbers, for example. Of course, one would hope that the number systems defined in this fashion enjoy the same properties and have the same definable functions as their more standard counterparts. It turns out that this is not possible in the framework of partial setoids.

It is well known that in a constructive setting there are several possible implementations of real numbers (e.g. see Chirimar & Howe (1992), Ciaffaglione & Gianantonio (2000), Geuvers *et al.* (2001), Harrison (1998) and Jones (1993) for some works on the formalization of reals in type theory). Here we choose to define the setoid of real numbers $\mathbb{R}$ using Cauchy sequences. The total setoid is defined as

$$\begin{aligned}\mathbb{R}_t = \langle\ & \mathsf{el}_t = \langle \mathsf{seq} : \mathsf{N} \to \mathsf{Q}, \mathsf{con} : \textsc{Cauchy}\ \mathsf{seq}\rangle,\\ & \mathsf{eq}_t = \lambda r_1, r_2 : \mathsf{el}_t.\ r_1 \cdot \mathsf{seq} =_{\mathsf{conv}} r_2 \cdot \mathsf{seq},\\ & \mathsf{er} = \cdots\rangle\end{aligned}$$

where Q is the type of rational numbers, $\textsc{Cauchy}$ is the the property of being a Cauchy sequence of rationals:

$$\textsc{Cauchy}\ s = \forall i : \mathsf{N}.\ \exists k : \mathsf{N}.\ \forall j_1, j_2 : \mathsf{N}.\ j_1 > k \to j_2 > k \to |(s\ j_1) - (s\ j_2)| < 1/i$$

and $=_{\mathsf{conv}}$ is the equality on sequences of rational numbers that holds whenever two sequences are co-convergent:

$$(s_1 =_{\mathsf{conv}} s_2) = \forall i : \mathsf{N}.\ \exists k : \mathsf{N}.\ \forall j : \mathsf{N}.\ j > k \to |(s_1\ j) - (s_2\ j)| < 1/i$$

The corresponding partial setoid is

$$\begin{aligned}\mathbb{R}_p = \langle\ & \mathsf{el}_p = \mathsf{N} \to \mathsf{Q},\\ & \mathsf{eq}_p = \lambda r_1, r_2 : \mathsf{el}_t.\ (\textsc{Cauchy}\ r_1) \wedge (\textsc{Cauchy}\ r_2) \wedge (r_1 \cdot \mathsf{seq} =_{\mathsf{conv}} r_2 \cdot \mathsf{seq}),\\ & \mathsf{er} = \cdots\rangle\end{aligned}$$

As emphasized above, it is convenient to consider smaller number systems, like the natural or rational numbers, as subsetoids of the real numbers—an alternative would be to consider implicit coercions, see e.g. (Saïbi, 1997) but this falls beyond the scope of this paper. We have a type Q of rational numbers, that is not a subsetoid of $\mathbb{R}$. We are going to define the subset of real numbers corresponding to the rationals.

To this end we use the relation $\rightsquigarrow$ between $\mathsf{N} \to \mathsf{Q}$ and Q, such that $s \rightsquigarrow q$ holds if s converges to q:

$$s \rightsquigarrow q = \forall i : \mathsf{N}.\ \exists k : \mathsf{N}.\ \forall j : \mathsf{N}.\ i > k \to |(s\ j) - q| < 1/i$$

We define the predicate $\textsc{isRational}_t$: $\textsc{pred}_t\ \mathbb{R}_t$ by $(\textsc{isRational}_t \cdot \mathsf{pf}_t\ r) = \exists q$: $\mathsf{Q}.\ s \cdot \mathsf{seq} \rightsquigarrow q$. The predicate $\textsc{isRational}_p$: $\textsc{pred}_p\ \mathbb{R}_p$ is defined correspondingly. Now we want to define the subsetoid of the reals whose elements are the real numbers that satisfy $\textsc{isRational}_t$. This is $\mathbb{Q}_t = (\textsc{subSetoid}_t\ \mathbb{R}_t\ \textsc{isRational}_t)$ in **TSet** and $\mathbb{Q}_p = (\textsc{subSetoid}_p\ \mathbb{R}_p\ \textsc{isRational}_p)$ in **RSet**. The problem now arises if we want to define a function on these subsetoids that depends strongly upon the satisfaction of the condition. For example, consider the function $\textsc{Num}$ that gives the numerator of the reduced fraction representing a rational number.

In the framework of partial setoids, defining $\textsc{Num}$ on $\mathbb{Q}_p$ requires that we define it on the whole type $\mathsf{N} \to \mathsf{Q}$, without any information on convergence. This is impossible because we cannot constructively compute whether a sequence converges to a rational value and, in such case, to which one. Further, it is also impossible to define $\textsc{Num}$ with the axiom of descriptions. On the other hand, if we work in the framework of total setoids and use the axiom of descriptions for total setoids, we can easily define such a function for $\mathbb{Q}_t$ since we can extract from one of its elements r the proof $\mathsf{insub}\ r$ containing the rational value of r.

5.2 Quotients

Both when working with total and partial setoids, quotients can be realized by just substituting the setoid equality with a stronger equivalence relation—below we refer to the latter as the quotienting relation. In either case the quotienting relation must preserve the setoid equality: if two elements are equal according to the book equality, they must be equivalent w.r.t. the quotienting relation. However, when working with partial setoids, a problem arises: the quotienting relation may hold for elements that are not equal to themselves according to the setoid equality, i.e. that are not in the domain of the setoid. In this case, taking the equivalence relation as the book equality of the quotient would add elements to the domain, which is incorrect. A solution, proposed by Hofmann (1995a), is to take as book equality in the quotient setoid the restriction of the equivalence relation on the original setoid to the domain of the setoid. We briefly develop this point below.

Recall that an equivalence relation R over a setoid A is an element of $\textsc{rel}_t\ A\ A$ that satisfies reflexivity, symmetry, and transitivity. Now assume that A is a total setoid and that R is such an equivalence relation with ϕ_R to witness that R is indeed an equivalence relation. For the sake of readability, we write $x \equiv_R y$ as a shorthand for $R \cdot \mathsf{rel}_t\ x\ y$. In **TSet** the quotient setoid A/R is defined as

$$\langle \mathsf{el}_t = \mathsf{el}_t\ A,\ \mathsf{eq}_t = \lambda x, y : \mathsf{el}_t\ A.\ x \equiv_R y, \mathsf{er} = \phi_R \rangle$$

But in the same situation in **RSet**, i.e. with A is a partial setoid, R a partial equivalence relation with ϕ_R to witness that R is indeed an partial equivalence relation – and using $x \equiv_R y$ as a shorthand for $R \cdot \mathsf{rel}_p\ x\ y$ it would be wrong to define (A/R) as

$$\langle \mathsf{el}_p = \mathsf{el}_p\ A,\ \mathsf{eq}_p = \lambda x, y : \mathsf{el}_p\ A.\ x \equiv_R y, \mathsf{per} = \phi_R \rangle$$

because it may happen that $x \equiv_R x$ holds when $x =_A x$ does not hold, so we are introducing new elements in the setoid. Instead, we must define first $R' : \textsc{rel}_p\ A\ A$ as

$$\langle \mathsf{rel}_p = \lambda x, y : \mathsf{el}_p\ A. x =_A x \wedge y =_A y \wedge x \equiv_R y,\ \mathsf{compat}_p = \cdots \rangle$$

and then define (A/R) as

$$\langle \mathsf{el}_p = \mathsf{el}_p\ A,\ \mathsf{eq}_p = \lambda x, y : \mathsf{el}_p\ A.\ x \equiv_{R'} y, \mathsf{per} = \ldots \rangle$$

6 Conclusion

Type-theoretical frameworks are used as a foundation for mathematics in several ongoing efforts to develop large libraries of formalized mathematics with proof-assistants such as Agda, Coq and Lego. It is therefore natural to study the relationship between type theory and the standard foundational framework for mathematics, i.e. set theory. Recently, several authors (Aczel, 1999; Werner, 1997) have undertaken a systematic comparison between set theory and intensional type theory (see also Aczel (1978, 1982, 1986) for earlier work).

This paper studies a related issue, namely the use of set-theoretic notions in type theory. More precisely, we focused on the use of setoids in the formalization of mathematics. We analyzed the different approaches to setoids that can be found in the literature, compared them, and drew some conclusions in regard to their appropriateness. Specifically, we showed that existing approaches can be classified into two equivalence classes: the first equivalence class contains total setoids, **TSet**, and some equivalent versions of partial setoids, **QSet** and **SSet**. The second equivalence class contains an essentially different way of using partial setoids, **RSet**, and **PSet**; for the latter, we have shown that a previously used approach to function space is inadequate and needs to be redefined as in **RSet**.

In addition, we compared the two classes under the aspect of suitability for the formalization of mathematics. In particular, we showed that the partial setoid methodology runs into practical difficulties when dealing with subsetoids.

Acknowledgements

The authors are indebted to the referees for their constructive comments on earlier versions of the paper.

References

Aczel, P. (1978) The type theoretic interpretation of constructive set theory. In: MacIntyre, A., Pacholski, A. and Paris, J., editors, *Proceedings of Logic Colloqium 77.*

Aczel, P. (1982) The type theoretic interpretation of constructive set theory: choice principles. In: Troelstra, A. S. and van Dalen, D., editors, *Proceedings L. E. J. Brouwer Centenary Symposium.* Studies in Logic and the Foundations of Mathematics, vol. 110, pp. 1–40. North-Holland.

Aczel, P. (1986) The type theoretic interpretation of constructive set theory: inductive definitions. In: Marcus, R. B., Dorn, G. J. W. and Weingartner, P., editors, *Proceedings of Logic, Methodology and Philosphy of Science VII*. Studies in Logic and the Foundations of Mathematics, vol. 114, pp. 17–49. North-Holland.

Aczel, P. (1993) *GALOIS: A theory development project*. Manuscript.

Aczel, P. (1999) On relating type theories and set theories. In: Altenkirch, T., Naraschewski, W. and Reus, B., editors *Proceedings of TYPES'98: Lecture Notes in Computer Science 1657*, pp. 1–18. Springer-Verlag.

Altenkirch, T. (1999) Extensional equality in intensional type theory. *Proceedings of LICS'99*, pp. 412–420. IEEE Computer Society Press.

Bailey, A. (1993) *Representing Algebra in LEGO*. MPhil thesis, University of Edinburgh.

Barbanera, F. and Berardi, S. (1996) Proof-irrelevance out of excluded-middle and choice in the calculus of constructions. *J. Functional Program.* **6**(3), 519–525.

Barthe, G. (1995a) Extensions of pure type systems. In: Dezani-Ciancaglini, M. and Plotkin, G., editors, *Proceedings of TLCA'95: Lecture Notes in Computer Science 902*, pp. 16–31. Springer-Verlag.

Barthe, G. (1995b) *Formalising Mathematics in Type Theory: Fundamentals and Case Studies.* Technical Report CSI-R9508, University of Nijmegen.

Beeson, M. (1985) *Foundation of Constructive Mathematics*. Springer-Verlag.

Bishop, E. (1967) *Foundations of Constructive Analysis*. McGraw-Hill.

Capretta, V. (2002) *Abstraction and Computation*. PhD thesis, University of Nijmegen, Computing Science Institute.

Chirimar, J. and Howe, D. J. (1992) Implementing constructive real analysis (Preliminary report). In: Myers, J. P. and O'Donnel, M. J., editors, *Constructivity in Computer Science: Lecture Notes in Computer Science 613*, pp. 165–178. Springer-Verlag.

Ciaffaglione, A. and Di Gianantonio, P. (2000) A coinductive approach to real numbers. In: Coquand, Th., Dybjer, P., Nordström, B. and Smith, J., editors, *Proceedings of TYPES'99: Lecture Notes in Computer Science 1956*, pp. 114–130. Springer-Verlag.

Coquand, C. and Coquand, T. (1999) Structured type theory. *Proceedings of LFM'99* (*held in conjunction with PLI'99*).

Coquand, T. (1990) Metamathematical investigations of a calculus of constructions. In: Odifreddi, P., editor, *Logic and Computer Science*, pp. 91–122. Academic Press.

Courtieu, P. (2001) *Représentation d'Algèbres Non Libres en Théorie des Types*. PhD thesis, Université Paris 11.

Čubrić, D., Dybjer, P. and Scott, P. (1998) Normalization and the Yoneda embedding. *Math. Struct. in Comput. Sci.* **8**(2), 153–192.

Geuvers, H., Pollack, R., Wiedijk, F. and Zwanenburg, J. (2001) The algebraic hierarchy of the FTA project. *Informal Proceedings of CALCULEMUS'01.*

Harrison, J. (1998) *Theorem Proving with the Real Numbers*. Distinguished Dissertations. Springer-Verlag.

Hofmann, M. (1994) Elimination of extensionality in Martin-Löf's type theory. In: Barendregt, H. and Nipkow, T., editors, *Proceedings of TYPES'93: Lecture Notes in Computer Science 806*, pp. 166–190. Springer-Verlag.

Hofmann, M. (1995a) A simple model for quotient types. In: Dezani-Ciancaglini, M. and Plotkin, G., editors, *Proceedings of TLCA'95: Lecture Notes in Computer Science 902*, pp. 216–234. Springer-Verlag.

Hofmann, M. (1995b) *Extensional Concepts in Intensional Type Theory*. PhD thesis, Laboratory for the Foundations of Computer Science, University of Edinburgh.

Huet, G. and Saïbi, A. (2000) Constructive category theory. In: Plotkin, G., Stirling, C. and Tofte, M., editors, *Proof, Language and Interaction – Essays in honour of Robin Milner*, pp. 239–275. MIT Press.

Jacobs, B. (1999) *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics, no. 141. North Holland.

Jones, C. (1993) Completing the rationals and metric spaces in LEGO. In: Huet, G. and Plotkin, G., editors, *Logical Environments*, pp. 297–316. Cambridge University Press.

Lacas, S. and Werner, B. (1999) *Which choices imply the excluded middle?* Manuscript.

Lambek, J. and Scott, P. (1986) *Introduction to Higher Order Categorial Logic*. Cambridge University Press.

Luo, Z. (1994) *Computation and Reasoning: A Type Theory for Computer Science*. International Series of Monographs on Computer Science, no. 11. Oxford University Press.

Luo, Z. and Pollack, R. (1992) *LEGO proof development system: User's manual*. Technical report ECS-LFCS-92-211, LFCS, University of Edinburgh.

Maietti, M.-E. (1999) About effective quotients in constructive type theory. In: Altenkirch, T., Naraschewski, W. and Reus, B., editors, *Proceedings of TYPES'98: Lecture Notes in Computer Science 1657*, pp. 164–178. Springer-Verlag.

Nederpelt, R., Geuvers, H. and de Vrijer, R. (eds). (1994) *Selected Papers on Automath*. Studies in Logic and the Foundations of Mathematics, vol. 133. North-Holland.

Nordström, B., Petersson, K. and Smith, J. (1990) *Programming in Martin-Löf's Type Theory. An Introduction*. International Series of Monographs on Computer Science, no. 7. Oxford University Press.

Qiao, H. (2000) Formalising formulas-as-types-as-objects. In: Coquand, T., Dybjer, P., Nordström, B. and Smith, J., editors, *Proceedings of TYPES'99: Lecture Notes in Computer Science 1956*, pp. 174–193. Springer-Verlag.

Saïbi, A. (1997) Typing algorithm in type theory with inheritance. *Proceedings of POPL'97*, pp. 292–301. ACM Press.

Saïbi, A. (1998) *Algèbre Constructive en Théorie des Types, Outils génériques pour la modélisation et la démonstration, Application á la théorie des Catégories*. PhD thesis, Université Paris VI.

Salvesen, A. and Smith, J. (1988) The strength of the subset type in Martin-Löf's type theory. In: *Proceedings of LICS'88*, pp. 384–391. IEEE Press.

The Coq Development Team (2002) *The Coq Proof Assistant User's Guide. Version 7.2.*

Troelstra, A. S. and van Dalen, D. (1988) *Constructivism in Mathematics*. North-Holland.

Werner, B. (1994) *Méta-théorie du Calcul des Constructions Inductives*. PhD thesis, Université Paris 7.

Werner, B. (1997) Sets in types, types in sets. In: Abadi, M. and Ito, T., editors, *Proceedings of TACS'97: Lecture Notes in Computer Science 1281*, pp. 530–546. Springer-Verlag.