HAL
archives-ouvertes.fr

Implementing Cantor's Paradise

Furio Honsell, Marina Lenisa, Luigi Liquori, Ivan Scagnetto

▶ **To cite this version:**

Furio Honsell, Marina Lenisa, Luigi Liquori, Ivan Scagnetto. Implementing Cantor's Paradise. 14th Asian Symposium on Programming Languages and Systems, Nov 2016, Hanoi, Vietnam. hal-01362819

HAL Id: hal-01362819
https://hal.inria.fr/hal-01362819

Submitted on 9 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Implementing Cantor's Paradise*

Furio Honsell[1], Marina Lenisa[1], Luigi Liquori[2], Ivan Scagnetto[1]

[1] Università di Udine, Italy, {*furio.honsell, marina.lenisa, ivan.scagnetto*}*@uniud.it*
[2] INRIA, Sophia Antipolis Méditerranée, France, *Luigi.Liquori@inria.fr*

dedicated to Marco Forti for his 70^{th} birthday

Aus dem Paradies, das Cantor uns geschaffen hat,
soll uns niemand vertreiben können.
David Hilbert

Abstract. Set-theoretic paradoxes have made all-inclusive self-referential Foundational Theories almost a taboo. The few daring attempts in the literature to break this taboo avoid paradoxes by restricting the class of formulæ allowed in Cantor's naïve *Comprehension Principle*. A different, more intensional approach was taken by Fitch, reformulated by Prawitz, by restricting, instead, the shape of deductions, namely allowing only normal(izable) deductions. The resulting theory is quite powerful, and *consistent by design*. However, *modus ponens* and Scotus *ex contradictione quodlibet* principles fail. We discuss Fitch-Prawitz Set Theory (FP) and implement it in a Logical Framework with so-called *locked types*, thereby providing a "Computer-assisted Cantor's Paradise": an interactive framework that, unlike the familiar Coq and Agda, is closer to the familiar informal way of doing mathematics by delaying and consolidating the required normality tests. We prove a *Fixed Point Theorem*, whereby all partial recursive functions are definable in FP. We establish an intriguing connection between an extension of FP and the Theory of Hyperuniverses: the bisimilarity quotient of the coalgebra of closed terms of FP satisfies the Comprehension Principle for Hyperuniverses.
Keywords: Fitch-Prawitz set theory, logical frameworks, paradoxes, coalgebras, hyperuniverses

1 Introduction

The discovery of set-theoretic paradoxes at the turn of last century, such as Russell's, Burali-Forti's and Curry's, inhibited mainstream foundational research from exploring self-referential, all-inclusive Foundational Theories. There are a very few exceptions in the literature. Quine's NF and the Theory of *Hyperuniverses* by Forti and Honsell [FH96] avoid paradoxes while preserving *extensionality*, by restricting the class of formulæ allowed in Cantor's Comprehension Principle to *stratified* or to *generalized positive formulæ*, respectively.

In 1952, Frederic Brenton Fitch [Fit52] introduced a Foundational Set-Theory, consistent by design, which has a more *intensional* flavour. It compensates the

* Work supported by the COST Action IC1201 BETTY "Behavioural Types for Reliable Large-Scale Software Systems".

potentially paradoxical effects of an un-constrained *naïve* Comprehension Principle by restricting the class of deductions. Fitch introduced two possible conditions which are rather idiosyncratic and unnecessarily restrictive, see [Fit52]. It was not until Prawitz in 1966 [Pra06], who gave a *natural deduction* presentation of Fitch's Theory, that a more principled restriction on deductions was introduced, namely that the deduction *be normal*.

Apart from the restriction on the shape of deductions, Fitch-Prawitz Set Theory, FP, is otherwise a standard first order theory with classical negation. Sets, *i.e.* abstractions, are introduced and eliminated in the natural way, and equality is expressed by *Leibniz equality*. FP subsumes higher-order logic for all orders. Fitch himself showed how a considerable part of the theory of Real Numbers can be developed in FP. The theory however is only *paraconsistent*, in that Scotus principle *ex contradictione quodlibet* fails. Moreover the standard rules, such as *modus ponens* or *extensionality* are not admissible.

In this paper, we discuss FP and give a *Fixed Point Theorem*, whereby all partial recursive functions are definable in FP as one would in *functional programming languages*.

Furthermore we show how to encode the highly unorthodox *side condition* of FP in a Logical Framework based on Constructive Set Theory featuring *locked types*, [HLMS16]. This allows to build an extremely flexible, *all-inclusive* interactive foundational environment for developing Mathematics and its foundations. This is indeed an interactive, computer-assisted *Cantor's Paradise*, where one can *optimistically* use the unrestricted Comprehension Principle, in the style of [CSW14,DHJG06] and of *optimistic concurrency* control in distributed systems. I.e., the nuisance of checking consistency is done automatically at the end!

Finally, we provide an intriguing set-theoretic connection between an extension of FP and the Theory of Hyperuniverses, [FHL94]. Namely we show that the *strongly extensional* quotient, *i.e.* the *bisimilarity quotient*, of the coalgebra of closed terms of Fitch-Prawitz Theory satisfies the restricted Comprehension Principle of Hyperuniverses. The relevance of this result is twofold. It provides a purely proof-theoretic consistency proof for the Theory of Hyperuniverses. Moreover, it shows that, if we insist on extensionality, a consistent Comprehension Principle cannot be broadened much beyond *positive formulæ*.

Synopsis. In Section 2, we present the theory FP, and in Section 3 we discuss it. In Section 4, we show how Mathematics can be developed in FP, in particular we prove the Fixed Point Theorem. In Section 5, we show how to encode FP in a Logical Framework featuring locked types. In Section 6, we study the connection between FP and the extensional Theory of Hyperuniverses. In Section 7 we discuss FP as a logical framework and compare it to other "optimistic" frameworks. Final remarks appear in Section 8.

Acknowledgments. The authors are grateful to the anonymous referees, and to Oleg Kiselyov, for many useful remarks and intriguing questions.

2 The Theory of Fitch-Prawitz, FP

We present a classical version of the logical theory of Fitch-Prawitz, which we call FP. We follow essentially [Pra06]. The theory FP includes the usual logical connectives $\wedge, \vee, \rightarrow$, and the $\forall, \exists$ quantifiers, the logical constant $\bot$, together with an unrestricted set constructor. Negation is not a primitive connective, $\neg A$ being expressed as $A \rightarrow \bot$. The crucial non-standard restriction is that only *normal* deductions are allowed in Fitch-Prawitz theory.

2.1 The Language of FP

Definition 1 (Symbols). *The* symbols *consist of the binary constant* $\in$*, the constant* λ *for set abstraction, the logical constant* $\bot$*, the logical connectives* $\neg$, $\wedge$, $\vee$, $\rightarrow$*, the universal and existential quantifiers* $\forall$ *and* $\exists$*. We assume a denumerable set of variables, denoted by lower-case letters* $x, y, z, \ldots$

Definition 2 (Terms and Formulæ). Terms *and* formulæ *are defined by mutual induction:*
$(\mathcal{T} \ni)\ t, u\ ::=\ x \mid \lambda x.A$
$(\mathcal{F} \ni)\ A, B, \ldots, P, \ldots\ ::=\ \bot \mid \neg A \mid A \wedge A \mid A \vee A \mid A \rightarrow A \mid \forall x.A \mid \exists x.A \mid t \in u$,
where $\neg A$ *is an abbreviation for* $A \rightarrow \bot$*.*

We use standard conventions concerning free and bound occurrences of variables, and, of course, *Barendregt's hygiene condition*. Open and closed terms and formulæ are defined as usual. The set of free variables of a term t or a formula A will be denoted by $\mathsf{Fv}(t)/\mathsf{Fv}(A)$. The set of closed terms/formulæ is denoted by $\mathcal{T}^0/\mathcal{F}^0$. Formula contexts $A[\ ]$, where $A[\ ]$ is an incomplete formula with a hole, are defined as usual. We denote by $t[u/x]$, $A[u/x]$ the (capture-avoiding) substitution of the term u for the variable x in the term t or in the formula A.

2.2 Inference Rules and Deductions

We present inference rules in natural deduction style (see Fig. 1). The inference rules consist of an introduction and an elimination rule for each logical connective, for $\forall, \exists$ quantifiers, and for λ, and of a rule for $\bot$.

The rules of FP appear in Fig. 1. In the rules $\forall$I) and $\exists$E), the variable y does not belong to free variables of $A \setminus \{x\}$ and it must not occur free in any hypothesis or undischarged assumptions.

Notice that in FP the rule of *negation introduction* is absorbed by $\rightarrow$I)

We call *quasi-deduction* the standard notion of deduction in Natural Deduction. In FP, not all quasi-deductions are allowed, but only those which essentially correspond to *normal deductions* in Natural Deduction.

Definition 3. *The premisses* A *in the rule* $\rightarrow$E)*,* C *in the rule* $\vee$E)*,* B *in the rule* $\exists$E) *are called* minor premisses*. A premiss that is not minor is called a* major premiss*.*

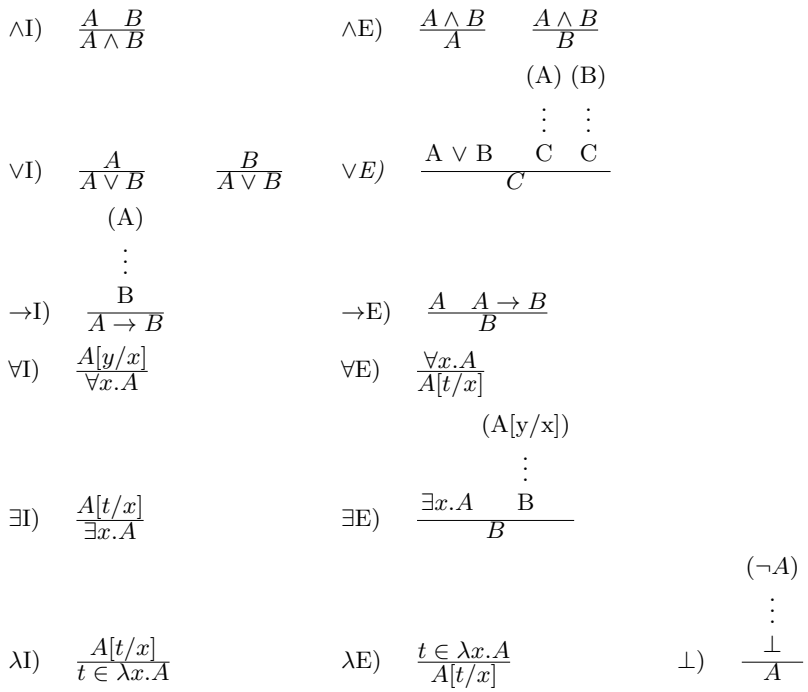

Fig. 1. FP rules in natural deduction style

Definition 4 (Deductions).

- *We call* quasi-deduction *in* FP *a standard deduction in the system* FP*,* i.e. *a formula tree obtained by applying the inference rules.*
- *A formula occurrence in a deduction that is both the consequence of an application of a I-rule or of the ⊥-rule, and major premiss of an application of a (correspondent) E-rule is said to be a* maximum formula *in the deduction.*
- *A* deduction *in* FP *is a quasi-deduction with no maximum formulæ,* i.e. *a* normal *deduction.*
- *We write $\Gamma \vdash_{\mathsf{FP}} A$, for Γ set of formulæ, when there is a deduction of the formula A from the set of formulæ in Γ.*
- *A deduction from no assumptions is called a* proof.

The shape of *normal deductions* manifestly accounts for the fact that no *information* can be logically extracted from a formula which had not been already available at the outset. Normal Deductions apparently satisfy a sort of *nihil fit ex nihilo*. Namely in a main branch β of a deduction Π, the formula occurrences in β that are major premisses of E-rules precede all formula occurrences in β which are premisses of I-rules or the $\bot$-rule. In the *proposition-as-types* analogy, normal proofs correspond to normal forms of λ-calculus. In most popular logics, all deductions can be normalized, although non normal proofs are usually more

concise and more natural. The use of *lemmata*, or *cuts*, produces non-normal proofs. Normalization then amounts to proving all lemmata from first principles within each proof. But cuts, as in FP can lead to circularities. In Section 3.1 we give examples of deductions which cannot be normalized. These arise from the unrestricted use of elimination rules, and therefore might yield deductions of $\bot$.

If we consider the fragment of the system FP where we omit the λ-rules, we obtain a system for classical logic with the property that each derivation is normalizable, *i.e.* for each derivation there exists a corresponding derivation in normal form (see [Pra06], Ch. III, Theorems 1,2). In the full system FP this property fails. In derivations, it is still possible to remove any given maximum formula, but in general it is impossible to remove all maximum formulæ.

Nevertheless, FP is consistent, namely, there is at least a formula which is not derivable, *i.e.* $\bot$. If this were the case, the $\bot$-rule would make the system *trivial*. This is essentially Thm. 3 of [Pra06], but since consistency does not require a tight control on the shape of subformulæ, we shortcircuit some of the arguments.

Proposition 1. *The system* FP *is consistent.*

Proof. We proceed by contradiction. Assume that there is a proof Π (*i.e.*, a deduction in normal form depending on the empty set) of $\bot$. Consider a main branch β of Π. One can easily see that the formula occurrences in β that are major premisses of E-rules precede all formula occurrences in β which are premisses of I-rules or the $\bot$-rule. Otherwise there would be a first formula occurrence in β which is a major premiss of an E-rule but succeeds a premiss of an I-rule or the $\bot$-rule, and such a formula would be a maximum formula contrary to the hypothesis that β is normal. But there are no I-rules for $\bot$ hence there are no premisses of I-rules in β. Hence the first formula in β must be an undischarged assumption, and this contradicts the initial hypothesis that Π is a proof. □

3 The Theory FP: Pros and Cons

Why Normal Deductions? Restricting the class of legitimate deductions in FP to normal deductions yields consistency *by design*, see Proposition 1. Alternately, one could have brutally accepted only deductions which do not derive $\bot$, but then huge complications would arise because we do need *subdeductions* which yield $\bot$, in dealing with negation.

Moreover, since Classical Logic is normalizing, FP is a *conservative extension* with respect to classical theorems which *do not mention* simultaneously both λ and $\in$. This allows to develop in FP a considerable portion of standard Mathematics, see [Fit52].

In [Fit52], Fitch did not introduce the *normal deduction* proviso to qualify proper deductions. He discussed instead two alternate conditions which appear rather idiosyncratic and unnecessarily restrictive, which he called the *simple* and the *special restrictions*. The simple restriction does not allow to derive *e.g.* $A \to (B \to ((A \wedge B) \to C) \to C))$, while the special restriction does not allow to derive $(P \leftrightarrow (P \to Q)) \to Q$ or $((A \to (A \to B) \to B) \to A) \to A$.

Negation. We have already pointed out that *negation* is not a primitive logical operator, but it is encoded using $\bot$, namely $\neg A \stackrel{\Delta}{=} A \to \bot$. The $\bot$)-rule enforces *classical* negation. It clearly subsumes the intuitionistic rule *ex falso sequitur quodlibet*, namely $\dfrac{\bot}{A}$.

Moreover, applying the $\to$E)-rule and provided the overall deduction normalizes, it encompasses also the *double negation* rule, namely $\dfrac{\neg\neg A}{A}$, and hence it proves also *tertium non datur*, *i.e.* $((A \to \bot) \to \bot) \to A$.

Full Elimination Rules are not Admissible. The choice of allowing only normal deductions makes standard elimination rules "unsafe", *i.e.* not admissible. *E.g. Modus Ponens*, *i.e.* the $\to$E)-rule, cannot be applied naïvely, in that if we have normal deductions of $A \to B$ and of A, it is not true in general that the extended deduction obtained by an application of the $\to$E)-rule is still normal.

Normalizable Quasi-Deductions: $\mathsf{FP}^{\#}$. The constraint of considering quasi-deductions to be legal only if already in *normal form* can be weakened to allow for *normalizable* quasi-derivations. In order to define *normalizable* deductions we need to introduce a *calculus of deductions*, a sort of λ-calculus as in the *propositions-as-types* paradigm, and define an appropriate notion of *reduction* which reflects the *inversion principle* underpinning the normalization procedure. One can easily see that this can be done and that the deduction calculus is strongly normalizing. Then, consistency of $\mathsf{FP}^{\#}$ follows from that of FP. This extension of FP, called $\mathsf{FP}^{\#}$ has been recently discussed in [HLMS16], where a type system for characterizing the strongly normalizable λ-terms has been introduced, see Subsection 4.1 below.

Paraconsistency. In FP Duns Scotus rule *ex absurdis sequitur quodlibet*, namely: $\dfrac{A \quad \neg A}{\bot}$ is not admissible, let alone derivable. Since $\neg A$ is encoded as $A \to \bot$, the derivability of *Scotus rule* would require an application of the $\to$E)-rule, which is subject to restrictions on the subdeductions, and hence is not safe in general. In Subsection 3.1 we will see that also Aristotle's *non-contradiction* principle fails, namely we have that for suitable A's $\vdash_{\mathsf{FP}} A \wedge \neg A$. Thus FP is paraconsistent. This is inevitable since, in FP a strong fixed point theorem (Theorem 4.1) holds.

In the original system of Fitch [Fit52], *negation* is a primitive unary connective which behaves differently from our encoding of $\neg$ and satisfies Scotus rule. In [Pra06], Prawitz calls it *constructive negation*. To allow for this notion of negation, Fitch has to give up *excluded middle* and *negation introduction* (which he calls *reductio ad absurdum*). Both restrictions, introduced by Fitch in [Fit52] on quasi-deductions to qualify them as deductions, ensure that the system is consistent but *not* paraconsistent.

3.1 The Taming of Russell's and Curry's Paradoxes

The interest in the theory of Fitch-Prawitz FP lies in its power of taming the naïve *Comprehension Principle*, namely permitting to reason on the set of elements satisfying any formula P.

Thus, in particular, Russell's and Curry's classes are definable in FP, but the deductions, involving these classes, which would allow to derive $\bot$ in classical logic are not normalizable.

Russell's Paradox. Let us define $t \stackrel{\Delta}{=} \lambda x.(x \notin x)$, where $t \notin t$ denotes the formula $\neg(t \in t)$, *i.e.* $t \in t \to \bot$. Then we have the following quasi-deduction:

$$\dfrac{\dfrac{\dfrac{\dfrac{t \in t^{(1)}}{t \notin t} \qquad t \in t^{(1)}}{\bot}}{t \notin t^{(1)}}}{t \in t} \qquad \dfrac{\dfrac{\dfrac{t \in t^{(1)}}{t \notin t} \qquad t \in t^{(1)}}{\bot}}{t \notin t^{(1)}}}{\bot}$$

the index (1) above indicates where hypotheses are discharged. Notice that we have both $\vdash_{\mathsf{FP}} t \in t$ and $\vdash_{\mathsf{FP}} t \notin t$, since the two subderivations are legal in FP. However $\bot$ is not derivable, since the overall quasi-deduction cannot be transformed into a normal deduction. If we perform a step of $\to$-reduction we end up introducing a *new* λ-reduction, indefinitely. We can derive legally instead $\vdash_{\mathsf{FP}} (t \in t) \wedge (t \notin t)$. This amounts to the failure of Aristotle's *Principle of non-contradiction*. However, Scotus rule does not apply, and hence this contradiction does not trivialize the theory, but just makes it paraconsistent.

Curry's Paradox. Let P be any formula, and let $Y \stackrel{\Delta}{=} \lambda x.(x \in x \to P)$. Then we have the following quasi-deduction:

$$\dfrac{\dfrac{\dfrac{Y \in Y^{(1)}}{Y \in Y \to P} \qquad Y \in Y^{(1)}}{P}}{Y \in Y \to P^{(1)}} \qquad \dfrac{\dfrac{\dfrac{Y \in Y^{(1)}}{Y \in Y \to P} \qquad Y \in Y^{(1)}}{P}}{Y \in Y \to P^{(1)}}}{Y \in Y} \qquad \dfrac{\dfrac{\dfrac{Y \in Y^{(1)}}{Y \in Y \to P} \qquad Y \in Y^{(1)}}{P}}{Y \in Y \to P^{(1)}}}{P}$$

Clearly, the above quasi-deduction cannot be transformed into a normal deduction, because the same infinite reduction-chain that occurs in Russell's paradox above would be generated here. Notice that the two quasi-derivations obtained by just dropping the last application of the $\to$E)-rule are in normal form.

Moreover, from the very definition of Y, by applying the λI)-rule, we obtain $(Y \in Y \to P) \to (Y \in Y)$, and by applying the λE)-rule, we obtain $(Y \in Y) \to (Y \in Y \to P)$. Hence we have $\vdash_{\mathsf{FP}} (Y \in Y) \leftrightarrow (Y \in Y \to P)$. This is related to the *Fixed Point Theorem* of Section 4.1, which takes us very close to a paradox but not quite. Russell's class is a special case of Curry's Paradox, if the formula P is taken to be $\bot$.

The Role of Structural Rules in the Paradoxes. In deriving both Russell's and Curry's Paradoxes, we have used the structural rule of *contraction*. In each branch we have discharged two instances of the same assumption. Grishin [Gri82] was the first to show that Naïve Set Theory without contraction is consistent, albeit very weak. To see this it is enough to realize that it amounts to a Set Theory whose logic is Girard's Linear Logic without exponentials, and therefore

all deductions are normalizable even in the presence of λ and $\in$. Hence the "murderer" who chases us away from Cantor's Paradise, namely the "root cause" of the set-theoretic paradoxes, is not *extensionality* or *tertium non datur*, it is not even related to *negation*. It is the structural rule of *contraction* which, via Curry's Paradox, yields inconsistency even in minimal logic.

Incidentally, we point out that the expressive power of J.Y.Girard's Light Linear Logic with abstractions, *LLLs* (see [Gir98], Appendix A.1) lies in between Grishin's Naïve Set Theory without contraction, and the theory of Fitch-Prawitz.

3.2 Equality and Extensionality

Equality in FP is expressed as *Leibniz Equality*, namely

$$t_1 = t_2 \stackrel{\Delta}{=} \forall x.\, t_1 \in x \leftrightarrow t_2 \in x.$$

In Set Theory, it is natural to consider a much stronger version of equality, namely *Extensional Equality*

$$t_1 \simeq t_2 \stackrel{\Delta}{=} \forall x.\, x \in t_1 \leftrightarrow x \in t_2.$$

In FP we can derive $t_1 \simeq t_2 \to t_1 = t_2$. The converse implication amounts to the *Extensionality Axiom* $\quad t_1 = t_2 \to t_1 \simeq t_2$.

Grishin [Gri82] showed in 1982 that, adding Extensionality, the contraction rule becomes derivable. Hence it allows to derive Russell's Paradox already in a Naïve Set Theory based on Linear Logic without exponentials.

Extensionality has a similar impact also on FP. First we need to extend the notion of *normal deduction* to deductions which make use of *axioms*. This is done simply by stipulating that axioms behave as undischarged *assumptions*. Hence, the analogue of Grishin's result for FP is that one can derive a normal deduction of $\bot$ whose only assumptions are instances of Extensionality. Thus, the Extensionality Axiom makes FP inconsistent. We give a direct proof of this:

Proposition 2. $\mathsf{Ext} \vdash_{\mathsf{FP}} \bot$.

Proof. Let $Y \stackrel{\Delta}{=} \{x \mid x \in x\}$, $\emptyset \stackrel{\Delta}{=} \{x \mid \bot\}$, $R \stackrel{\Delta}{=} \{x \mid x \in x \to \bot\}$, $X \stackrel{\Delta}{=} \{x \mid R \in R\}$. Then $R \in R \vdash_{\mathsf{FP}} \forall x. x \in \emptyset \leftrightarrow x \in X$. Namely,

$$\dfrac{\dfrac{\dfrac{\dfrac{x \in X^{(1)}}{R \in R} \qquad \dfrac{R \in R}{R \in R \to \bot}}{\bot}}{x \in \emptyset}}{x \in X \to x \in \emptyset} \qquad\qquad \dfrac{\dfrac{\dfrac{x \in \emptyset^{(1)} \qquad R \in R^{(2)}}{\bot}}{R \in R \to \bot}}{\dfrac{\dfrac{R \in R}{x \in X}}{x \in \emptyset \to x \in X}}$$

Using *Ext*, we have $R \in R \vdash_{\mathsf{FP}} \forall x. \emptyset \in x \leftrightarrow X \in x$. By instantiating x to Y we get $R \in R \vdash_{\mathsf{FP}} \emptyset \in Y \leftrightarrow X \in Y$, hence using λE), we obtain $R \in R \vdash_{\mathsf{FP}} \emptyset \in \emptyset \leftrightarrow X \in X$. Since, by λI) $R \in R \vdash_{\mathsf{FP}} X \in X$, by $\to$E) we get $R \in R \vdash_{\mathsf{FP}} \emptyset \in \emptyset$ and by λE) $R \in R \vdash_{\mathsf{FP}} \bot$. Finally, since $\vdash_{\mathsf{FP}} R \in R$ (see Russell's Paradox at the beginning of Section 3.1), we get a contradiction. One can easily check that all the above arguments are indeed normal deductions. □

Section 6 is devoted to show how Extensionality can be recovered in a weak FP.

4 Developing Mathematics in FP

In this Section we show that even if Extensionality is inconsistent with FP, nevertheless Leibniz Equality allows us to derive a considerable part of Mathematics and Logic in FP. Similar developments can be carried out also in Fitch original Theory [Fit52] and in Girard's *LLLs* [Gir98], Appendix A.1.

First we need to introduce the following fundamental abbreviations:

$\emptyset \triangleq \lambda x.\bot \qquad V \triangleq \lambda x.(x = x) \qquad \{x \mid A\} \triangleq \lambda x.A \qquad \{t\} \triangleq \lambda x.(x = t)$

$\{t_1, \ldots, t_n\} \triangleq \lambda x.(x = t_1 \ \vee \ \ldots x = t_n) \qquad \langle t_1, t_2\rangle \triangleq \{t_1, \{t_2\}\}$

$\langle t_1, \ldots, t_n\rangle \triangleq \boldsymbol{t} \triangleq \langle \ldots \langle t_1, t_2\rangle, \ldots, t_n\rangle \quad \lambda x_1 \ldots x_n.A \triangleq \lambda z.(z = \langle x_1, \ldots, x_n\rangle \wedge A)$.

One can easily see that when any such abbreviation is taken as the definition in FP of the intended notion, it satisfies in FP the standard properties of this notion. *E.g.* two t-ple's are equal if and only if all their components are equal.

4.1 The Fixed Point Theorem

The outstanding expressive power of FP derives from the following logical *Fixed Point Theorem*, which allows us to define entities in FP following a sort of *functional programming paradigm.*

Theorem 1 (Fixed Point (FPT)). *Let A be a formula with free variables $x, z_1, \ldots, z_n$, $n > 0$. Then there exists a term u such that $\vdash_{\mathsf{FP}} \boldsymbol{z} \in u \longleftrightarrow A[u/x]$, where $\boldsymbol{z}$ is a shorthand for $\langle z_1, \ldots, z_n\rangle$.*

Proof. Let $u \triangleq \{\boldsymbol{z} \mid \langle \boldsymbol{z}, t\rangle \in t\}$, where $t \triangleq \{\langle \boldsymbol{z}, y\rangle \mid A[\{\boldsymbol{w} \mid \langle \boldsymbol{w}, y\rangle \in y\}/x]\}$. Then the implication $\boldsymbol{z} \in u \longrightarrow A[u/x]$ and its converse can be derived via two applications, respectively, of the λE-rule, and of the λI-rule. □

Paraconsistency follows immediately from Theorem 1, just take the formula A to be $z \notin x$. Notice that the contradiction, $\bot$, arises from $z \in u \longleftrightarrow z \notin u$, only if we can either use freely the structural rule of *contraction* or a *non-normalizable* proof. The former is precisely what is not allowed in Girard's *LLLs*, while non-normalizable proofs are precisely what are ruled out by FP.

Curry's paradoxical Y as defined in Section 3 is closely related to the fixed point construction but it is not an instance of it. In fact, an alternative Y can be obtained using the Fixed Point Theorem. Namely, consider the formula $A \triangleq z \in x \rightarrow P$. Then, by the Fixed Point Theorem, there exists a term u such that $\vdash_{\mathsf{FP}} z \in u \longleftrightarrow (z \in u \rightarrow P)$. Now, by substituting u for z, we get $u \in u \longleftrightarrow (u \in u \rightarrow P)$. By the proof of the Fixed Point Theorem, u can be taken to be $\{z \mid \langle z, t\rangle \in t\}$. Of course, the Fixed Point Theorem above admits a straightforward generalization to the n-ary case, *i.e.* the case of n formulæ. We will illustrate the power of the Fixed Point Theorem in the following examples.

Selfsingleton Construction. Using the Fixed Point Theorem, one can build the selfsingleton set in FP. Namely, let A be the formula $z = x$. Then, by the Fixed Point Theorem, there exists a term u such that $\vdash_{\mathsf{FP}} z \in u \longleftrightarrow z = u$. By the proof of the Fixed Point Theorem, u can be defined by $u \triangleq \{z \mid \langle z, t\rangle \in t\}$, where $t \triangleq \{\langle z, y\rangle \mid z = \{w \mid \langle w, y\rangle \in y\}\}$.

The natural question arises as to whether there exist more than one selfsingleton. The answer is positive, since any fixed point operator induces a different one. For instance, in the proof of the Fixed Point Theorem, one can take $u \stackrel{\Delta}{=} \{z \mid \langle z, a, t\rangle \in t\}$ and $t \stackrel{\Delta}{=} \{\langle z, a, y\rangle \mid A[\{w \mid \langle w, a, y\rangle \in y\}/x]\}$, for any a, thus getting a different fixed point operator, which thus yields a different selfsingleton.

Recursive Definitions of Functions and Sets. The Fixed Point Theorem, FPT, allows us to define *recursive* sets and functions in FP as in *functional programming* using general recursion, see also [Gir98], Appendix A.1.

Numerals. To define numerals, consider two fixed conventional sets/terms, which we denote by 0 and S, to represent zero and successor. *E.g.* take $\emptyset$ and V. Then apply FPT to the formula A_{Nat}:

$$A_{\mathsf{Nat}}[z, x] \stackrel{\Delta}{=} (\forall A.\ (0 \in A \ \wedge\ \forall y \in A.\ <S, y> \in A)) \longrightarrow\ z \in A) \longrightarrow\ z \in x.$$

By FPT there exists a term Nat such that

$$\vdash_{\mathsf{FP}} z \in \mathsf{Nat} \ \longleftrightarrow A_{\mathsf{Nat}}[z, \mathsf{Nat}].$$

We have enforced *Induction* on Nat by means of *minimality*. In what follows, we use the standard notation $0, 1, \ldots$ to denote numerals.

Subtraction. To define the subtraction function, consider the following formula:

$$A_{\mathsf{Subt}}[\boldsymbol{z}, x] \stackrel{\Delta}{=} (\forall A.$$
$$\forall y_1, y_2, y_3 \in \mathsf{Nat}. \left\{ \begin{array}{c} \langle\langle 0, y_2\rangle, 0\rangle \in A \wedge \\ \langle\langle y_1, 0\rangle, y_1\rangle \in A \wedge \\ \langle\langle y_1, y_2\rangle, y_3\rangle \in A \rightarrow \langle\langle y_1 + 1, y_2 + 1\rangle, y_3\rangle \in A \end{array} \right\} \rightarrow \boldsymbol{z} \in A)$$
$$\longrightarrow \boldsymbol{z} \in x.$$

Then, by the FPT, there exists a term Subt such that

$$\vdash_{\mathsf{FP}} \langle\langle z_1, z_2\rangle, z_3\rangle \in \mathsf{Subt} \ \longleftrightarrow A_{\mathsf{Subt}}[\boldsymbol{z}, \mathsf{Subt}].$$

Lambda terms. The set of closed λ-terms Λ^0 is definable starting from three conventional sets, var the variable marker, app, the application marker, and lam the λ-abstraction marker. For simplicity we omit the "minimality" conditions. Consider the following formula A_{Λ^0}:

$$A_{\Lambda^0} \stackrel{\Delta}{=} (\exists n \in \mathsf{Nat}.\ z = \langle \mathsf{var}, n\rangle) \ \vee\ (\exists y_1, y_2 \in x.\ z = \langle \mathsf{app}, y_1, y_2\rangle) \vee (\exists y \in x.\ \exists n \in \mathsf{Nat}.\ z = \langle \mathsf{lam}, n, y\rangle)\ .$$

Then, by the FPT, there exists a term Λ^0 such that

$$\vdash_{\mathsf{FP}} z \in A_{\Lambda^0} \ \longleftrightarrow\ (\exists n \in \mathsf{Nat}.\ z = \langle \mathsf{var}, n\rangle) \ \vee\ (\exists y_1, y_2 \in \Lambda^0.\ z = \langle \mathsf{app}, y_1, y_2\rangle) \vee (\exists y \in \Lambda^0.\ \exists n \in \mathsf{Nat}.\ z = \langle \mathsf{lam}, n, y\rangle)\ .$$

Given a term N of λ-calculus we denote by $\widetilde{N}$ its FP representation.

Normal λ-terms. Using Theorem 4.1 and the set Λ^0 defined above, we can define the relation R_β consisting of the pairs of terms in Λ^0 such that $\langle \widetilde{M}, \widetilde{N}\rangle \in R_\beta$ iff the λ-terms M and N are β-convertible. Again applying Theorem 4.1 we can now define a predicate Λ^+ such that $x \in \Lambda^+$ is equivalent in FP to $x \in \Lambda^0 \wedge\ \forall y.y \in \Lambda^+ \rightarrow \exists u.\langle u, \langle app, x, y\rangle\rangle \in R_\beta \wedge u \in \Lambda^+$. Then, there is a normal proof in FP of $\widetilde{M} \in \Lambda^+$ only if M is a *closed strongly normalizing term.*

In Section 3, we introduced $\mathsf{FP}^{\#}$, the extension of FP where *normalizable* deductions are legal. In [HLMS16], a type system was suggested for characterizing the strongly normalizable λ-terms. That construction amounts to carrying out the above argument in $\mathsf{FP}^{\#}$ instead of FP. A legal deduction in $\mathsf{FP}^{\#}$ of $\widetilde{M} \in \Lambda^{+}$ would then amount to *typing* M with the type Λ^{+}. There is indeed a natural reflection of the metatheoretic normalizability of the $\mathsf{FP}^{\#}$ deduction of the typing judgement $\widetilde{M} \in \Lambda^{+}$, and the fact that M is indeed strongly normalizable!

Partial Recursive Functions. The above examples can be generalized. Relying on the FPT, we can define objects as in Functional Programming provided we enforce the "minimality" condition, thereby showing that FP is a *Universal Model of Computation*:

Theorem 2. *For any* partial recursive *function h on natural numbers of arity k, there exists a formula P_h with free variables $x_1, \ldots, x_k, y$ such that*

$$h(n_1, \ldots, n_k) \simeq m \iff \vdash_{\mathsf{FP}} P_h[\widetilde{n}_1/x_1, \ldots, \widetilde{n}_k/x_k, \widetilde{m}/y],$$

where $n_1, \ldots n_k, m$ are natural numbers and $\widetilde{n}_1, \ldots, \widetilde{n}_k, \widetilde{m}$ denote the corresponding numerals in FP.

Notice that if we do not enforce the "minimality" condition in the formulæ used in FPT, then we might end up with a lot of "junk". This might be a feature, whereby one can include also infinite and circular objects, *i.e.* introduce co-inductive datatypes.

5 Encoding FP in a Type Theoretic Logical Framework

An implementation of FP in a computer-assisted proof development environment, such as LF, see [HHP93,PS99,WCPW03,COQ], would take us as close as consistently possible to Cantor's Paradise. However, FP is a formal system whose encoding in standard Logical Frameworks is not straightforward. It is indeed very awkward to capture the side-condition which allows only normal deductions.

In this section, we assume the reader familiar with Logical Frameworks and we present the encoding of FP in $\mathsf{LLF}_{\mathcal{P}}$ [HLMS16], a recent extension of the Edinburgh LF which features *lock types*. This encoding provides, in effect, a paramount example of the power of *locks*.

In $\mathsf{LLF}_{\mathcal{P}}$, a new type constructor is introduced and, as costumary in Constructive Type Theory, it is explained through appropriate Introduction, Elimination, and Equality rules. More precisely, in $\mathsf{LLF}_{\mathcal{P}}$ we define objects using a new constructor of the form $\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]$, whose type $\mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]$ is assigned via the type-checking *introduction rule* ($O{\cdot}Lock$). Correspondingly, also an *unlock destructor*, $\mathcal{U}^{\mathcal{P}}_{N,\sigma}[M]$, is introduced whose type is given by the *elimination rule* ($O \cdot Top \cdot Unlock$). This latter rule allows for the elimination of the lock-type constructor, under the condition that a specific predicate $\mathcal{P}$ is verified, possibly *externally*, on a judgement. The rules mentioned above are:

$$\frac{\Gamma \vdash_{\Sigma} M : \rho \quad \Gamma \vdash_{\Sigma} N : \sigma}{\Gamma \vdash_{\Sigma} \mathcal{L}^{\mathcal{P}}_{N,\sigma}[M] : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho]}\ (O{\cdot}Lock) \qquad \frac{\Gamma \vdash_{\Sigma} M : \mathcal{L}^{\mathcal{P}}_{N,\sigma}[\rho] \quad \mathcal{P}(\Gamma \vdash_{\Sigma} N : \sigma)}{\Gamma \vdash_{\Sigma} \mathcal{U}^{\mathcal{P}}_{N,\sigma}[M] : \rho}\ (O{\cdot}Top{\cdot}Unlock)$$

The *equality rule* for lock types amounts to a new form of reduction called *lock reduction* ($\mathcal{L}$-reduction), $\mathcal{U}^{\mathcal{P}}_{N,\sigma}[\mathcal{L}^{\mathcal{P}}_{N,\sigma}[M]] \to_{\mathcal{L}} M$, which allows for the removing of a *lock*, in the presence of an *unlock* with the same superscripts and subscripts. The $\mathcal{L}$-reduction combines with standard β-reduction into $\beta\mathcal{L}$-reduction.

Capitalizing on the monadic nature of the lock constructor ([HLMS16]), one can use locked terms without necessarily establishing the predicate, provided an *outermost* lock is present. This increases the expressivity of the system, and allows for reasoning under the assumption that the verification is successful, as well as for postponing and reducing the number of verifications. The rules which make all this work are:

$$\frac{\Gamma, x{:}\tau \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{S,\sigma}[\rho] : \mathsf{type} \quad \Gamma \vdash_\Sigma N : \mathcal{L}^{\mathcal{P}}_{S',\sigma'}[\tau] \quad \sigma =_{\beta\mathcal{L}} \sigma' \quad S =_{\beta\mathcal{L}} S'}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{S,\sigma}[\rho[\mathcal{U}^{\mathcal{P}}_{S',\sigma'}[N]/x]] : \mathsf{type}} \; (F{\cdot}Guarded{\cdot}Unlock)$$

$$\frac{\Gamma, x{:}\tau \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{S,\sigma}[M] : \mathcal{L}^{\mathcal{P}}_{S,\sigma}[\rho] \quad \Gamma \vdash_\Sigma N : \mathcal{L}^{\mathcal{P}}_{S',\sigma'}[\tau] \quad \sigma =_{\beta\mathcal{L}} \sigma' \quad S =_{\beta\mathcal{L}} S'}{\Gamma \vdash_\Sigma \mathcal{L}^{\mathcal{P}}_{S,\sigma}[M[\mathcal{U}^{\mathcal{P}}_{S',\sigma'}[N]/x]] : \mathcal{L}^{\mathcal{P}}_{S,\sigma}[\rho[\mathcal{U}^{\mathcal{P}}_{S',\sigma'}[N]/x]]} \; (O{\cdot}Guarded{\cdot}Unlock)$$

The second rule is the counterpart of the elimination rule for monads, once we realize that the standard destructor of monads $let_{T_{\mathcal{P}(\Gamma \vdash S:\sigma)}} x = A \; in \; N$ can be replaced in this setting by $N[\mathcal{U}^{\mathcal{P}}_{S,\sigma}[A]/x]$. This is the case since the $\mathcal{L}^{\mathcal{P}}_{S,\sigma}[\cdot]$-monad satisfies the property $let_{T_{\mathcal{P}}} \; x = M \; in \; N \to N$ if $x \notin \mathsf{Fv}(N)$, provided x occurs *guarded* in N, *i.e.* within subterms of the appropriate lock-type. The first rule takes care of elimination at the level of types.

The system $\mathsf{LLF}_{\mathcal{P}}$ can smoothly enforce the *global* normalization constraint of FP *locally* by enforcing a suitable lock on the proof-object. The crucial step is the definition of the predicate involved in the lock, because it needs to be *well-behaved*, see [HLMS16], Definition 2.1. Namely it must be closed under substitution as well as signature and context extension, and this is problematic when dealing with open terms. To overcome these difficulties we need to introduce the notion of *skeleton* of a term in a given signature Σ:

Definition 5. *Given a signature Σ, let Λ_Σ (respectively Λ^o_Σ) be the set of* $\mathsf{LLF}_{\mathcal{P}}$ *terms (respectively* closed $\mathsf{LLF}_{\mathcal{P}}$ *terms) definable using constants from Σ. A term M has a* skeleton *in Λ_Σ if there exists a context $N[\ ,\ldots,\ ] \in \Lambda_\Sigma$ with n holes such that $M \equiv N[M_1,\ldots,M_n]$ for suitable terms $M_1,\ldots,M_n$.*

Furthermore we need to introduce two basic judgements to deal with variables. Namely we make the distinction between *generic* judgements, which cannot be directly utilized in arguments, but which can be assumed, and *apodictic* judgements, which are directly involved in proof rules. In order to make use of generic judgements, one has to downgrade them to an apodictic one, and this is achieved by a suitable coercion function.

The encoding in $\mathsf{LLF}_{\mathcal{P}}$ of the system of Fitch as presented in Section 2.1 is given in the following definition, where (due to lack of space) we focus on the crucial connectives and rules of FP:

Definition 6 ($\mathsf{LLF}_{\mathcal{P}}$ signature Σ_{FP} for Fitch Prawitz Set Theory FP). *The following constants are introduced:*

```
o   : Type                    ι : Type
T   : o -> Type               δ : Π A:o.(V(A) -> T(A))
V   : o -> Type               ⊃ : o -> o -> o          false : o
lam : (ι -> o)-> ι            ϵ : ι -> ι -> o          not: o -> o
⊃_intro: ΠA,B:o.(V(A) -> T(B)) -> (T(A ⊃B))
⊃_elim : ΠA,B:o.Πx:T(A).Πy:T(A⊃B) -> L^Fitch_⟨x,y⟩,T(A)×T(A⊃B)[T(B)]
λ_intro : ΠA:ι ->o.Πt:ι.T(A t) -> T(ϵ t (lam A))
λ_elim  : ΠA:ι ->o.Πt:ι.T(ϵ t (lam A))->T(A t)
bot : ΠA:o.(V(not A) -> T(false)) -> T(A)
```

where o *is the type of propositions,* ⊃ *is the implication connective,* ϵ *is the "membership" predicate,* not *is the negation,* lam *is the "abstraction" operator for building "sets",* T *is the apodictic judgement,* V *is the generic judgement,* δ *is the coercion function, and* ⟨x, y⟩ *denotes the encoding of pairs, whose type is denoted by* σ×τ*, e.g.* λu:σ → τ → ρ. u x y : (σ → τ → ρ) → ρ*. The predicate in the lock is defined as follows:* Fitch(Γ ⊢$_{\Sigma_{\mathsf{FP}}}$ ⟨x, y⟩ : T(A)×T(A ⊃ B)) *holds iff* x *and* y *have skeletons in* $\Lambda_{\Sigma_{\mathsf{FP}}}$*, all the holes of which have either type* o *or are guarded by a* δ*, and hence have type* V(A)*, and, moreover, the proof derived by combining the skeletons of* x *and* y *is normal in the natural sense.*

The notion of *normal* deduction is the standard notion of Definition 4. The predicate Fitch is well-behaved because it considers terms only up-to holes in the skeleton, which can have type o or are generic judgements. Adequacy for this signature can be achieved in the format of [HLLMS13]:

Theorem 3 (Adequacy for FP). *If* $A_1, \ldots, A_n$ *are the atomic formulæ occurring in* $B_1, \ldots, B_m, A$*, then* $B_1 \ldots B_m \vdash_{\mathsf{FP}} A$ *iff there exists a normalizable* M *such that* $\mathtt{A_1}$:o, …, $\mathtt{A_n}$:o, $\mathtt{x_1}$:V($\mathtt{B_1}$), …, $\mathtt{x_m}$:V($\mathtt{B_m}$) ⊢$_{\Sigma_{\mathsf{FP}}}$ M:T(A) *(where* A*, and* $\mathtt{B_i}$ *represent the encodings of, respectively,* A *and* B_i *in* $\mathsf{LLF}_{\mathcal{P}}$*, for* $1 \leq i \leq m$*).*

If in the definition of the well-behaved predicate Fitch we enforce that the deduction is normalizable, we obtain a signature for $\mathsf{FP}^{\#}$. The predicate would then be only semi-decidable.

In the spirit of $\mathsf{LLF}_{\mathcal{P}}$, we do not specify how to enforce the verification of the constraint in the locks. This is left for optimization. The idea underpinning $\mathsf{LLF}_{\mathcal{P}}$ is to specify neatly the interface that this, possibly external, module needs to satisfy in order to be safely plugged in the Logical Framework.

6 The Extensional Quotient of FP

In this section, we relate Fitch-Prawitz Set Theory, FP, to the Theory of Hyperuniverses, TH. Namely, we show that the *extensional quotient* of the closed term model of a suitable extension of FP, called FP^{+}, is a hyperuniverse.

6.1 The Theory of Hyperuniverses TH

The naïve Comprehension Principle can be consistently approximated, by restricting the class of admissible formulæ. In [FH89,FH89a], the *Generalized Positive Comprehension Scheme* has been introduced, namely:

Axiom 1 (Generalized Positive Comprehension Scheme (GPC))
$\{x \mid A\}$ *is a set, if* A *is a* Generalized Positive Formula*, where* Generalized Positive Formulæ (GPF) *are the smallest class of formulæ*
- *including* $u \in t$, $u = t$;
- *closed under the logical connectives* $\wedge, \vee$;
- *closed under the quantifiers* $\forall x, \exists x, \forall x \in y, \exists x \in y$, *where* $\forall x \in y.A$ *(*$\exists x \in y.A$*) is an abbreviation for* $\forall x.(x \in y \to A)$ *(*$\exists x.(x \in y \to A)$*);*
- *closed under the formula* $\forall x.(B \to A)$*, where* A *is a generalized positive formula and* B *is any formula such that* $\mathsf{Fv}(B) \subseteq \{x\}$.

In [FH89,FH89a], the Theory of Hyperuniverses TH, namely GPC + Extensionality, was introduced and proved consistent, together with many extensions which include arbitrary models of Zermelo-Frænkel Set Theory.

The theory TH is a rather expressive Set Theory, in which one can show the existence of many large sets, *e.g.*:
- the universe V, the empty set $\emptyset$;
- $\langle x, y\rangle, \{t\}, \{t, u\}, t \cup u, t \cap u, t \times u, t \circ u, \bigcup t, \bigcap t, \mathsf{dom}(t), \mathsf{cod}(t), t^{-1}, \mathcal{P}(t), \diamond(t) = \{x \mid t \cap x \neq \emptyset\}, \widehat{t}(u) = \{z \mid \exists w \in u.\ \langle w, z\rangle \in t\}, \mathcal{F}(t) = \{y \mid t \in y\}, t_1 \star t_2 = \{\langle u, v, w\rangle \mid \langle u, v\rangle \in t_1 \wedge \langle u, w\rangle \in t_2\}$;
- the equality $\Delta \stackrel{\Delta}{=} \{\langle x, y\rangle | x = y\}$, the membership relation $\in \stackrel{\Delta}{=} \{\langle x, y\rangle | x \in y\}$, the graph of the projection functions π_1, π_2, $\pi_1 \stackrel{\Delta}{=} \{z \mid \exists x, y.\ z = \langle\langle x, y\rangle, x\rangle\}$, the inclusion relation $\subseteq \stackrel{\Delta}{=} \{z \mid \exists x, y.\ (z = \langle x, y\rangle \wedge \forall w.\ y \in w \longrightarrow x \in w)\}$, the graph of the singleton function $\lambda x.\{x\} \stackrel{\Delta}{=} \{z \mid z = \langle x, \{x\}\rangle\}$.

We call *hyperuniverses* the set-theoretic structures which are models of TH, following the terminology of [FH89,FH89a], where many such structures were defined using topological and categorical tools.

6.2 The Extensional Quotient of the Fitch-Prawitz Coalgebra

In this section we study the *extensional quotient*, or *extensional collapse*, of the Fitch-Prawitz coalgebra of closed terms. In particular, we show that a suitable extension of FP, called FP^+, yields an extensional collapse which is (strongly) extensional, and satisfies the GPC scheme, *i.e.* it is a hyperuniverse. This result establishes a connection between FP and TH. For basic definitions and results on coalgebras, we refer to [JR11]. The theory FP^+ is the extension of FP with the following ω-rule:

$$\text{(Bounded-}\omega\text{)} \quad \frac{A[w/x] \text{ for all closed } w \text{ s.t. } B[w/x],\ \mathsf{Fv}(B) \subseteq \{x\}}{\forall x.(B[w/x] \to A)}$$

Even if the (Bounded-ω)-rule has infinitely many premisses, once it is taken as an introduction rule, the notions of quasi-deduction and deduction for FP can be naturally extended to FP^+. Consistency of FP^+ is proved then as for FP.

Notice that in our setting the conclusion of the (Bounded-ω)-rule really amounts to a restricted quantification w.r.t. a closed term. Given that $\mathsf{Fv}(B) \subseteq \{x\}$, the formula $\forall x.(B[w/x] \to A)$ amounts to $\forall x \in \{z \mid B[z]\}.A$, where

$\{z \mid B[z]\}$ is a closed term. Notice that the *Induction Rule* is subsumed by the (Bounded-ω)-rule. Before defining the coalgebra of closed FP^+-terms, we recall the notion of *set-theoretic structure*:

Definition 7 (Set-theoretic Structure). *A* set-theoretic structure $(X, \in)$ *is a first-order structure X together with a binary predicate $\in$ on $X \times X$, denoting the membership relation.*

Notice that set-theoretic structures are coalgebras for the powerset functor $\mathcal{P}(\ )$ on the category *Set*. The following definition will be useful in the sequel.

Definition 8 ((Strongly) Extensional Coalgebra).

- *A $\mathcal{P}(\ )$-coalgebra (X, f_X) is* extensional *if f is injective.*
- *A $\mathcal{P}(\ )$-coalgebra (X, f_X) is* strongly extensional *if the unique coalgebra morphism from (X, f_X) into the final coalgebra is injective.*

Clearly, strong extensionality implies extensionality.

The provable instances of the $\in$-relation on the set of closed FP^+-terms, $\mathcal{T}^0$, naturally induce a coalgebra structure for the powerset functor.

Definition 9 (Fitch-Prawitz Coalgebra). *Let $f_{\mathcal{T}^0} : \mathcal{T}^0 \longrightarrow \mathcal{P}(\mathcal{T}^0)$ be the $\mathcal{P}(\ )$-coalgebra defined by $f_{\mathcal{T}^0}(t) = \{s \mid \vdash_{\mathsf{FP}^+} s \in t\}$, where $\mathcal{P}(\ )$ denotes the standard powerset functor on the category* Set*.*

Given a $\mathcal{P}(\ )$-coalgebra (X, f_X), there is a unique mapping into the *final coalgebra*, $g : (X, f_X) \to (\Omega, f_\Omega)$, where (Ω, f_Ω) denotes the final coalgebra. This latter is clearly extensional, actually it is strongly extensional. The image via g of (X, f_X) into the final coalgebra (Ω, f_Ω) is called the *extensional quotient* of (X, f_X). The extensional quotient is given by the equivalence classes under bisimilarity. In FP^+ (actually already in FP), the notion of bisimilarity can be defined in the theory itself.

Definition 10 (Bisimilarity).

- *Let A_{Bis} be the FP^+formula with free variable x defined by*
  $$A_{\mathsf{Bis}} \stackrel{\Delta}{=} \forall t, t' \ (\langle t, t' \rangle \in x \ \longrightarrow \ \forall s(s \in t \ \longrightarrow \ \exists s'(s' \in t' \ \wedge \ \langle s, s' \rangle \in x)) \wedge \forall s'(s' \in t' \ \longrightarrow \ \exists s.(s \in t \ \wedge \ \langle s, s' \rangle \in x))) \ .$$
  A bisimulation *is a binary relation R such that $\vdash_{\mathsf{FP}^+} A_{\mathsf{Bis}}[R/x]$.*
- *The* bisimilarity *relation $\sim$ is defined by the following FP^+-term:*
  $$\sim \ \stackrel{\Delta}{=} \{\langle t, t' \rangle \mid \exists R. \ (\langle t, t' \rangle \in R \ \wedge \ A_{\mathsf{Bis}}[R/x])\} \ .$$

In the following lemma we show that bisimilarity is a maximal bisimulation equivalence:

Lemma 1. *a) Bisimilarity is an equivalence on FP^+.*
b) $\vdash_{\mathsf{FP}^+} t \sim t' \ \longleftrightarrow \ \forall s(s \in t \ \longrightarrow \ \exists s'(s' \in t' \ \wedge \ s \sim s')) \wedge \forall s'(s' \in t' \ \longrightarrow \ \exists s.(s \in t \ \wedge \ s \sim s')) \ .$

Proof. a) Straightforward.
b) ($\Rightarrow$) This amounts to $\vdash_{\mathsf{FP}^+} A_{\mathsf{Bis}}[\sim /x]$, which can be easily proved.
($\Leftarrow$) This follows by defining $R \stackrel{\Delta}{=} \{(t,t') \mid \forall s(s \in t \ \longrightarrow \ \exists s'(s' \in t' \ \wedge \ s \sim s')) \wedge \ \forall s'(s' \in t' \ \longrightarrow \ \exists s.(s \in t \ \wedge \ s \sim s'))\}$ and $R' \stackrel{\Delta}{=} R \cup \sim$, and proving $\vdash_{\mathsf{FP}^+} A_{\mathsf{Bis}}[R'/x]$. □

We can now quotient the FP^+-coalgebra by the bisimilarity $\sim$.

Definition 11 ($\sim$-quotient of the FP^+-coalgebra). *Let $\mathcal{M} = \mathcal{T}^0/\sim$ be the quotient of $\mathcal{T}^0$ by the bisimilarity $\sim$ on FP^+,* i.e., *for any $t \in \mathcal{T}^0$, we define $\underline{t} \in \mathcal{M}$ by $\underline{t} \stackrel{\Delta}{=} \{t' \mid \vdash_{\mathsf{FP}^+} t \sim t'\}$.*
$\mathcal{M}$ can be endowed with a structure of $\mathcal{P}(\ )$-coalgebra as follows. Let $f_{\mathcal{M}} : \mathcal{M} \to \mathcal{P}(\mathcal{M})$ be defined by $f_{\mathcal{M}}(\underline{t}) = \{\underline{s} \mid \vdash_{\mathsf{FP}^+} s \in t\}$. Then the projection $\pi : \mathcal{T}^0 \to \mathcal{M}$, defined by $\pi(t) = \underline{t}$, is a coalgebra-morphism from $(\mathcal{T}^0, f_{\mathcal{T}^0})$ to $(\mathcal{M}, f_{\mathcal{M}})$, i.e.

$$\begin{array}{ccc} \mathcal{T}^0 & \xrightarrow{f_{\mathcal{T}^0}} & \mathcal{P}(\mathcal{T}^0) \\ {\scriptstyle\pi}\downarrow & & \downarrow{\scriptstyle\mathcal{P}(\pi)} \\ \mathcal{M} & \xrightarrow[f_{\mathcal{M}}]{} & \mathcal{P}(\mathcal{M}) \end{array}$$

Finally we prove strong extensionality of $\mathcal{M}$ w.r.t. FP^+, notice the role of the (Bounded-ω)-rule.

Proposition 3. *The quotient $\mathcal{M}$ is extensional,* i.e. *for all $\underline{t}, \underline{t}' \in \mathcal{M}$,*

$$\underline{t} = \underline{t}' \iff f_{\mathcal{M}}(\underline{t}) = f_{\mathcal{M}}(\underline{t}') \ .$$

Proof. If $f_{\mathcal{M}}(\underline{t}) = f_{\mathcal{M}}(\underline{t}')$, *i.e.* $\{\underline{s} \mid \vdash_{\mathsf{FP}^+} s \in t\} = \{\underline{s}' \mid \vdash_{\mathsf{FP}^+} s' \in t'\}$, then for all s, $(\vdash_{\mathsf{FP}^+} s \in t \implies \exists s' \ (\vdash_{\mathsf{FP}^+} s' \in t' \wedge \vdash_{\mathsf{FP}^+} s \sim s'))$, and vice versa, hence, for all s, $(\vdash_{\mathsf{FP}^+} s \in t \implies \vdash_{\mathsf{FP}^+} \exists s' \ (s' \in t' \wedge \vdash_{\mathsf{FP}^+} s \sim s'))$, and vice versa. Therefore, by applying the bounded-ω-rule, we get
$\vdash_{\mathsf{FP}^+} \forall s(s \in t \ \longrightarrow \ \exists s'(s' \in t' \wedge s \sim s')) \wedge \forall s'(s' \in t' \ \longrightarrow \ \exists s.(s \in t \wedge s \sim s'))$, hence by Lemma 1, $\vdash_{\mathsf{FP}^+} t \sim t'$, *i.e.* $\underline{t} = \underline{t}'$. □

Corollary 1. *The quotient $\mathcal{M}$ is strongly extensional.*

We prove now that $\mathcal{M}$ satisfies the Generalized Positive Comprehension Scheme, namely it is a hyperuniverse. We start with the following definition, which actually defines an *inner model* of TH in $\mathsf{FP}^{\#}$:

Definition 12. *Let A be a formula with constants in $\mathcal{M}$. We define a corresponding formula $\widehat{A}$ by induction on A as follows:*

$$\begin{array}{ll} A \stackrel{\Delta}{=} \bot \implies \widehat{A} \stackrel{\Delta}{=} \bot & A \stackrel{\Delta}{=} A_1 \wedge A_2 \implies \widehat{A} \stackrel{\Delta}{=} \widehat{A}_1 \ \wedge \ \widehat{A}_2 \\ A \stackrel{\Delta}{=} \underline{u} \in \underline{t} \implies \widehat{A} \stackrel{\Delta}{=} \exists u'.u' \sim u \wedge u' \in t & A \stackrel{\Delta}{=} A_1 \vee A_2 \implies \widehat{A} \stackrel{\Delta}{=} \widehat{A}_1 \ \vee \ \widehat{A}_2 \\ A \stackrel{\Delta}{=} \underline{u} = \underline{t} \implies \widehat{A} \stackrel{\Delta}{=} u \sim t & A \stackrel{\Delta}{=} A_1 \to A_2 \implies \widehat{A} \stackrel{\Delta}{=} \widehat{A}_1 \ \to \ \widehat{A}_2 \\ A \stackrel{\Delta}{=} \neg A_1 \implies \widehat{A} \stackrel{\Delta}{=} \neg \widehat{A}_1 & A \stackrel{\Delta}{=} \forall x.A_1 \implies \widehat{A} \stackrel{\Delta}{=} \forall x.\widehat{A}_1 \\ & A \stackrel{\Delta}{=} \exists x.A_1 \implies \widehat{A} \stackrel{\Delta}{=} \exists x.\widehat{A}_1 \end{array}$$

Lemma 2. *For all A, u, t, x, $\widehat{A[\underline{t}/x]} \equiv \widehat{A}[t/x]$ and $\underline{u}[\underline{t}/x] \equiv \underline{u[t/x]}$.*

The following lemma, whose proof which uses (Bounded-ω-rule), is crucial.

Lemma 3. *For all GPF A with free variables $x_1, \ldots, x_n$, for all $t_1, \ldots, t_m \in \mathcal{T}^0$, $m \leq n$, we have: $\mathcal{M} \models A[\underline{t}_1/x_1, \ldots, \underline{t}_m/x_m] \iff \vdash_{\mathsf{FP}^+} \widehat{A}[t_1/x_1, \ldots, t_m/x_m]$.*

Proof. By induction on A, using Lemma 2, and the (Bounded-ω)-rule for dealing with the restricted $\forall$-case.
Base cases. $A \stackrel{\Delta}{=} \underline{u} = \underline{v}$. Let $\mathcal{M} \models (\underline{u} = \underline{v})[\underline{\boldsymbol{t}}/\boldsymbol{x}]$, *i.e.*, using Lemma 2, this holds if and only if $\mathcal{M} \models (\underline{u[\boldsymbol{t}/\boldsymbol{x}]} = \underline{v[\boldsymbol{t}/\boldsymbol{x}]}$, and this amounts to $\vdash_{\mathsf{FP}^+} u[\boldsymbol{t}/\boldsymbol{x}] \sim v[\boldsymbol{t}/\boldsymbol{x}]$.
$A \stackrel{\Delta}{=} \underline{u} \in \underline{v}$. Let $\mathcal{M} \models (\underline{u} \in \underline{v})[\underline{\boldsymbol{t}}/\boldsymbol{x}]$, *i.e.*, using Lemma 2, this amounts to $\vdash_{\mathsf{FP}^+} \exists u' \ (u' \sim u[\boldsymbol{t}/\boldsymbol{x}] \wedge u' \in v[\boldsymbol{t}/\boldsymbol{x}])$.
Induction step. We only deal with two cases: the remaining are similar.
$A \stackrel{\Delta}{=} A_1 \wedge A_2$. Let $\mathcal{M} \models (A_1 \wedge A_2)[\underline{\boldsymbol{t}}/\boldsymbol{x}]$, then $\mathcal{M} \models A_1[\underline{\boldsymbol{t}}/\boldsymbol{x}]$ and $\mathcal{M} \models A_2[\underline{\boldsymbol{t}}/\boldsymbol{x}]$. By induction hypothesis, $\vdash_{\mathsf{FP}^+} \widehat{A}_1[\boldsymbol{t}/\boldsymbol{x}]$ and $\vdash_{\mathsf{FP}^+} \widehat{A}_2[\boldsymbol{t}/\boldsymbol{x}]$, hence $\vdash_{\mathsf{FP}^+} (\widehat{A}_1 \wedge \widehat{A}_2)[\boldsymbol{t}/\boldsymbol{x}]$. The converse implication follows from the standard definition of the interpretation of $\wedge$ in a first-order structure.
$A \stackrel{\Delta}{=} \forall y \in z.A_1$. Unrestricted quantification is clearly a special case of this one, and by our earlier remark the case where $A \stackrel{\Delta}{=} \forall y.(B \to A_1)$, with $\mathsf{Fv}(B) \subseteq \{y\}$, amounts to restricted quantification. So if $\mathcal{M} \models \forall y \in z.\ A_1[\underline{\boldsymbol{t}}/\boldsymbol{x}, \underline{u}/z]$ then for all $\underline{t}'$ such that $\mathcal{M} \models \underline{t}' \in \underline{u}$, we have that $\mathcal{M} \models A_1[\underline{\boldsymbol{t}}/\boldsymbol{x}, \underline{u}/z, \underline{t}'/y]$. Then by induction hypothesis we have that for all $\boldsymbol{t}$ and for all t', such that $\vdash_{\mathsf{FP}^+} \exists y'.y' \sim t' \ \wedge \ y' \in u$ we have that $\vdash_{\mathsf{FP}^+} \widehat{A}[\boldsymbol{t}/\boldsymbol{x}, u/z, t'/y]$, hence applying the (Bounded-ω)-rule, we have that $\vdash_{\mathsf{FP}^+} \forall y.\exists y'.y' \sim u \ \wedge \ y' \in z \to \widehat{A}[\boldsymbol{t}/\boldsymbol{x}, u/z]$. The reverse implication follows from the interpretation of first-order formulæ in a structure. □

Now we are in the position to establish the main theorem of this section:

Theorem 4 ($\mathcal{M}$ satisfies GPC). *For any formula A in GPF with free variable x, $\mathcal{M} \models \underline{t} \in \underline{v} \iff \mathcal{M} \models A[\underline{t}/x]$, where $\underline{v} \stackrel{\Delta}{=} \underline{\{x \mid \widehat{A}\}}$. Hence $\mathcal{M}$ is a hyperuniverse.*

Proof. ($\Rightarrow$) From $\mathcal{M} \models \underline{t} \in \underline{\{x \mid \widehat{A}\}}$ we have $\vdash_{\mathsf{FP}^+} \exists t'.t' \sim t \ \wedge \ t' \in \{x \mid \widehat{A}\}$. Hence $\vdash_{\mathsf{FP}^+} \exists t'.t' \sim t \ \wedge \ \widehat{A}[t'/x]$, which, by Lemma 3, implies $\mathcal{M} \models A[\underline{t}'/x]$, for $t' \sim t$. Hence $\mathcal{M} \models A[\underline{t}/x]$. ($\Leftarrow$) By Lemma 3, from $\mathcal{M} \models A[\underline{t}/x]$ it follows $\vdash_{\mathsf{FP}^+} \widehat{A}[t/x]$. Hence $\vdash_{\mathsf{FP}^+} t \in \{x \mid \widehat{A}\}$, which implies $\mathcal{M} \models \underline{t} \in \underline{\{x \mid \widehat{A}\}}$. □

7 FP as a Logical Framework

FP is essentially Naïve Set-Theory, probably the most natural and straightforward of all Logical Frameworks, which we are familiar with since our schooldays. The reason for considering FP is twofold. The first reason is pragmatic, i.e., to explore how to use it for *fast and loose formal reasoning* on general recursion and datatypes, *i.e.*, as a proper Logical Framework (we borrow from [DHJG06]

this felicitous expression). The second reason is foundational. FP allows for a fine-tuned analysis of paradoxes arising from *diagonal arguments*.

Recently, in the formal methods community, there has been growing interest in logical systems which support convenient and fast, but logically unsound or even invalid features and heuristics [Cap05,CSW14,DHJG06]. Those arise especially in program transformation and program synthesis in non-terminating functional languages when dealing with general recursion. Albeit invalid, these methods are nonetheless extremely useful pragmatically. Furthermore, they can be justified. But this can be done only at the end once there is a good reason for going through the often daunting overhead of checking all the totality and predicativity preconditions [DHJG06,CSW14]. Oleg Kiselyov has remarked that the principled but cautious approach of Coq and Agda is akin to pessimistic concurrency: assuming that shared resources are likely to be contended and hence have to be proactively protected with (often numerous) locks. An alternative is optimistic concurrency, proceeding as if there were no contention – checking for consistency only at the end of a transaction. Optimistic concurrency is akin to *loose and fast programming* and the approach to termination checking, which can be carried out in Twelf and LF [Twelf,WN13].

Using FP as a Logical Framework goes precisely in the direction of *optimistic reasoning*, actually at two different levels. The first is that of using Lock types in the implementation to check the normalizability of deductions. Locks do not amount merely to the postponement of the checks. They rather allow for aggregating and simplifying the checks, so that the final check can be done possibly at some other level, rather than delegated to the metalanguage as in Coq or Agda.

Somewhat more ambitious, and not completely explored yet, is the *pragmatic* value of using a *paraconsistent* system. It was De Bruijn the founding author of AUTOMATH, himself, who first raised the challenging and provocative question: *do we really need a terminating metalanguage?* Of course if we use Scotus rule, then our reasoning is empty. But otherwise we still have plenty of useful arguments to carry out which can make visible *truly* false or missing requirements. So, even paraconsistent systems can increase our confidence in the outcome. After all, absolute certainty cannot be achieved, even with terminating systems.

A sharper understanding of which statements have a paraconsistent cognate is still missing. These arise usually in connection with *diagonal arguments*. Reasoning with small sets or, as we have shown, *generalized positive formulæ* does not lead to *paraconsistencies*. But there are probably many more classes of sentences, for instance in connection with the foundations of Category Theory.

8 Conclusions and Final Remarks

We have discussed the Naïve Set Theory of Fitch-Prawitz [Fit52,Pra06], FP, which is consistent by design, but nevertheless is expressive enough to define all partial recursive functions. Furthermore we have related it to the Theory of Hyperuniverses [FH89a,FHL94]. Foundationally, FP allows for a deeper understanding of the limitations implied by set theoretic paradoxes. In particular, we

have that even if $\vdash_{\mathsf{FP}^+} u \notin v$, then not necessarily $\mathcal{M} \models \underline{u} \notin \underline{v}$. This hints to the fact that, while retaining *Extensionality*, we cannot hope to go significantly beyond GPC in approximating the naïve Comprehension Principle, *e.g.* to include some *negative formulæ*. Pragmatically, FP offers a natural mathematical framework where to develop "optimistically" [CSW14,DHJG06] important branches of Mathematics from Real Numbers [Fit50] to Category Theory. We have encoded FP in the type-theoretic Logical Framework $\mathsf{LLF}_{\mathcal{P}}$, [HLMS16], which is currently under implementation, thereby providing what we called a "Computer Assisted Cantor's Paradise". Further lines of research on FP are the following.

Alternate inner models. In Section 6.2 we have proved that in FP^+ we can define an *Inner Model* for TH, namely, the model $\mathcal{M}$. But there are also *inner models* which have more than one *selfsingleton* and hence satisfy only Extensionality. E.g., the extensional quotient w.r.t. a bisimulation, which is an equivalence but does not equate the two selfsingletons defined in Subsection 4.1, would be an example of a hyperuniverse which is not strongly extensional.

Propositions as types for FP. So far we have based FP on *classical logic*. But we can replace the $\bot$)-rule by its intuitionistic version, namely *ex falso quodlibet*, to get an intuitionistic version of FP. One can then extend the λ-calculus language of proofs with new constructs to account for the rules concerning $\in$ and λ in FP. A simple solution is to extend a typed λ-calculus for intuitionistic proofs with a *1-ple* constructor $< M >$ to account for λI), and correspondingly introduce a π elimination constructor to account for λE):

$$\frac{\Gamma \vdash_{\mathsf{FP}} M : P(t)}{\Gamma \vdash_{\mathsf{FP}} < M > : t \in \lambda x.P(x)}\ \lambda\mathsf{Intro} \qquad \frac{\Gamma \vdash_{\mathsf{FP}} N : t \in \lambda x.P(x)}{\Gamma \vdash_{\mathsf{FP}} \pi(N) : P(t)}\ \lambda\mathsf{Elim}$$

The two constructors are related by the obvious reduction $\pi(\langle N \rangle) \longrightarrow N$. We can then prove that all proof terms corresponding to a contraction-free intuitionistic deduction are *normalizing*, thereby recovering Grishin's result. Notice that in normal deductions, where introduction constructs appear outermost w.r.t. elimination constructs, one can apply π only to variables, *i.e. generic* proof terms.

Escaping Gödel's Second Incompleteness Theorem. Since FP is a *cut free* Set Theory, *i.e.* it is *consistent by design*, within FP one can prove that there is a model of FP. This does not contradict Gödel's second Incompleteness Theorem, since FP is not closed under *modus ponens* which is the, so-called, Hilbert-Bernays third condition necessary for Gödel's result to go through.

FP *and Higher-Order Logics.* The Theory FP, being a *theory of sets*, subsumes higher order logics for any order. For instance $\forall P.Q[P]$ can be expressed as $\forall x.\ Pred(x) \to Q[x]$, for a suitable definition of $Pred$.

The Ubiquitous Hyperuniverse $\mathcal{N}_\omega(\emptyset)$. In [FH89a,FHL94], many hyperuniverses have been introduced. One of these, $\mathcal{N}_\omega(\emptyset)$, arises in many conceptually independent contexts, nicely described by Abramsky in [Abr11]. Namely, $\mathcal{N}_\omega(\emptyset)$ is *Cantor-1* space, the union of Cantor's space (obtained removing the middle thirds of the unit interval) with the centres of the removed intervals. $\mathcal{N}_\omega(\emptyset)$ is the unique solution of the metric domain equation $X \cong \mathcal{P}_{cl}(X_{\frac{1}{2}})$ in the category

of *complete metric spaces*. $\mathcal{N}_\omega(\emptyset)$ is the space of *maximal* points of the solution in Plotkin's category of *SFP domains* of the domain equation $X \cong \mathcal{P}_P(X_\perp) \oplus_\perp 1$, see [ABH03]. $\mathcal{N}_\omega(\emptyset)$ is the free *Stone modal Algebra* over 0 generators. By Theorem 4 we can add a new item to the list, namely: $\mathcal{N}_\omega(\emptyset)$ is the *extensional quotient* of Fitch-Prawitz coalgebra.

References

ABH03. F. Alessi, P. Baldan, F. Honsell. *A category of compositional domain-models for separable Stone spaces. Theor. Comput. Sci.* **290**1, 599–635, 2003.

Abr11. S. Abramsky. *A Cook's Tour of the Finitary Non-Well-Founded Sets CoRR*, abs/1111.7148, 2011, `http://arxiv.org/abs/1111.7148`.

Cap05. V. Capretta. *General recursion via coinductive types.* Logical Methods in Computer Science 1(2), 1-18 (2005)

CSW14. C. Casinghino, V. Sjöberg, S. Weirich. *Combining Proofs and Programs in a Dependently Typed Language.* In POPL '14, pp. 33–45, ACM, 2014.

COQ. Development Team. *The Coq Proof Assistant. Documentation, system download* Contact: `http://coq.inria.fr/`

DHJG06. N.A. Danielsson, J. Hughes, P. Jansson, J. Gibbons. *Fast and Loose Reasoning is Morally Correct.* In POPL'06, pp. 206–217, ACM, 2006.

FH89. M. Forti, R. Hinnion. *The consistency problem for positive comprehension principles. J.Symb.Logic* **54**, 1401–1418, 1989.

FH89a. M. Forti, F. Honsell. *Models of Self-descriptive Set Theories. D*edicated to Ennio De Giorgi on his sixtieth birthday, Birkhäuser, 1989.

FHL94. M. Forti, F. Honsell, M. Lenisa. *Processes and Hyperuniverses.* **MFCS 1994**, Springer LNCS, 352–363, 1994.

FH96. M. Forti, F. Honsell. *A General Construction of Hyperuniverses. Theor. Comput. Sci.* **156(1&2)**, 203–215, 1996.

Fit50. F. B. Fitch. *A Demonstrably Consistent Mathematics. J.S.L.* **15(1)**, 1950.

Fit52. F. B. Fitch. *Symbolic logic - An Introduction.* New York, 1952.

Gir98. J.-Y. Girard. *Light linear logic. Information and Computation*, 143(2):175–204, 1998, `doi:10.1.1.134.4420`.

Gri82. V. N. Grishin. *Predicate and set-theoretic calculi based on logics without contractions.* Math. USSR Izv. 18, 41–59, 1982.

HHP93. R. Harper, F. Honsell, G. Plotkin. *A Framework for Defining Logics. Journal of the ACM (JACM)*, 40(1), 143–184, ACM, 1993.

HLLMS13. F. Honsell, M. Lenisa, L. Liquori, P. Maksimovic, I. Scagnetto. *An open logical framework.* in JLC 26 (1): 293–335, 2016 (first pub. in 2013).

HLMS16. F. Honsell, L. Liquori, P. Maksimovic, I. Scagnetto. $\mathsf{LLF}_\mathcal{P}$*: A Logical Framework for Modeling External Evidence, Side Conditions, and Proof Irrelevance Using Monads.* In LMCS, 2016 (to appear).

JR11. B. Jacobs, J. Rutten. *An introduction to (co)algebras and (co)induction.* In Advanced topics in bisimulation and coinduction, 38–99, 2011.

PS99. F. Pfenning, C. Schürmann. *Twelf-A meta-logical framework for deductive systems (system description).* In CADE'16, Vol. 1632, 1999.

Pra06. D. Prawitz. *Natural Deduction – A proof-theoretical Study.* Dover Publications, New York, 2006.

Twelf. The Twelf Project. `http://twelf.org/wiki/Totality_assertion`

WN13. Y. Wang, G. Nadathur. *Towards extracting explicit proofs from totality checking in twelf.* In LFMTP'13, pp. 55-66, ACM, 2013.

WCPW03. K. Watkins, I. Cervesato, F. Pfenning, D. Walker. *A concurrent logical framework: The propositional fragment.* In Types'03, Springer, 2003.