Half a Syntactic Model to Prove Markov's Principle in CIC

Pierre-Marie Pédrot

October 7, 2019

Abstract

This note shows how to port Coquand-Hofmann's result[2] to CIC in a synthetic way. Namely, we give a model of CIC + Markov's principle in two passes, as an instance of the exceptional translation [10] where the type of exceptions is itself defined inside a presheaf model. The first pass can be described purely syntactically, while the second one requires a set-theoretic construction inside IZF with ω inaccessible cardinals. The final model enjoys semantic canonicity by construction.

1 Introduction

Twenty years ago, Coquand and Hofmann showed that PRA^ω could be consistently extended with Markov's principle, whose statement is usually stated as

$$\mathsf{MP} : \Pi f : \mathbb{N} \to \mathbb{B}.\, \neg\neg(\Sigma n : \mathbb{N}.\, f\ n = \mathtt{true}) \to \Sigma n : \mathbb{N}.\, f\ n = \mathtt{true}$$

Their proof was simply a syntactic interpretation of $\mathsf{PRA}^\omega + \mathsf{MP}$ into PRA^ω itself, which was purportedly a variant of Kripke semantics enriched with an additional layer looking like Friedman's A-translation [3].

While not provable in intuitionistic logic [11], Markov's principle has been long known to be nicely compatible with it, insofar as it does not break the witness property. There are various implementations of MP in the wild, with varying degrees of hackishness. In addition to Coquand-Hofmann's model, let us mention the following.

- The oldest one is probably Kleene realizability [7]. For simplicity, it can be rephrased as an extraction of intuitionistic proofs into a simply-typed λ-calculus with arbitrary fixpoints. MP is then realized via an unbounded loop that checks for every increasing n starting from 0 whether $f\ n = \mathtt{true}$ and stops as soon as it finds such a witness. In particular, the realizer does not use at all the doubly-negated premise, and furthermore its validity critically relies in MP being provable in the meta-theory. As such, it is arguably *cheating*. The requirement of arbitrary fixpoints in the language of realizers is also an issue when trying to justify MP in a sane way, as it brings in undecidability of the realizability relation in general.

- Dialectica [1] also inteprets MP, in a more intricate way. See my PhD [9] for a longer type-theoretical account of this interpretation. In a nutshell, Dialectica is a realizability interpretation over a language that allows to export intensional content from functions. A Dialectica realizer of a negation contains positive data that can be exploited, contrarily to Kleene and Kreisel realizability where such realizers are typically inhabiting a mere proposition. The realizer for MP extracts this data from its double-negated premise to prove the conclusion, allowing it to be at the same time more efficient than a clueless loop, and less demanding logically as it does not require meta-MP to prove that this term realizes object-MP.

- More recently, Herbelin [4] and Iliḱ [5] described how to implement MP using a weak form of delimited continuations. The dynamic content of their realizer is a program that creates a fresh, statically-bound exception and use it to feed the doubly-negated premise with a dummy term that will fail as soon as

the it uses it. The exception carries an integer as an argument, which is the witness of the existential being proved. The realizer thus catches the exception ensuring it does not escape, and returns its carried value. This is valid because the double negation is applied to a hereditarily positive type, which guarantees that the fresh exception cannot be involuntarily caught in a closure and escape its handler.

We argue that Coquand-Hofmann and Herbelin's implementation of MP are essentially the same from the point of view of computation. Coquand-Hofmann corresponds to an A-translation interpretation, i.e. a model with exceptions of type A, where the type A itself is defined w.r.t. a global cell containing a function $p : \mathbb{N} \to \mathbb{B}$, s.t. $A[p] \equiv \Sigma n : \mathbb{N}.\, p\ n = \texttt{true}$. The cell can be updated dynamically, which is used in the proof of MP, where the cell is locally set to contain the pointwise or of its current value and the provided f.

Similarly, as in Iliḱ's presentation, Herbelin's model can be typed more precisely w.r.t. an ordered stack of prompts, where failure is allowed to return at any point in this stack. This intuitively means that the intended meaning of a term of type A in a stack Σ is a sum $A + \Sigma$ where the right case corresponds to an exception. Given a stack Ξ extending Σ, it is always possible to lower a value A in Σ to a value A in Ξ, mimicking the monotonocity from Kripke semantics. Furthermore, prompts are restricted to hereditarily positive types, which can be encoded as decision functions on booleans. The ability to create a new prompt to prove MP is thus tantamount to the action of moving along the order in the Kripke construction.

In this paper, we give a generalization of the Coquand-Hofmann model to CIC, and make the proof more synthetic. The resulting model is the composition of two simpler model transformations, namely a presheaf interpretation followed by an exceptional interpretation. The global picture is

$$\mathsf{CIC} + \mathsf{MP} \xrightarrow{[\cdot]_{\mathcal{E}}} \mathsf{CIC} + \mathcal{E} \xrightarrow{\mathsf{Psh}} \mathsf{Set}$$

where $\mathcal{E}$ is a new type introduced by the presheaf intepretation with a few combinators, and $[\cdot]_{\mathcal{E}}$ is the exceptional translation with the type of exceptions set to $\mathcal{E}$. The presheaf construction can be alternatively targetting an extensional version of CIC rather than Set as in [6]. Unfortunately, as of today, it is still not known whether it is possible to present presheaves as a syntactic model into a type theory without strong forms of uniqueness of identity proofs and function extensionality, so we will have to admit that the second arrow is convincing enough for our purposes.

2 Exceptional Model

2.1 Overview

We briefly summarize the exceptional model [10]. Intuitively, given a type theory $\mathcal{T}$ with enough structure, e.g. containing CIC, together with a type $\vdash_{\mathcal{T}} \mathbb{E} : \Box$ representing exceptions, the exceptional translation builds a new exceptional type theory $\mathcal{T}_{\mathbb{E}}$ where failure is allowed. Such an effect is best represented by the existence in $\mathcal{T}_{\mathbb{E}}$ of terms

$$\begin{array}{l} \vdash_{\mathcal{T}_{\mathbb{E}}} \mathbf{E} : \Box \\ \vdash_{\mathcal{T}_{\mathbb{E}}} \texttt{raise} : \Pi A : \Box.\, \mathbf{E} \to A \end{array}$$

subject to the equations below

$$\begin{array}{rcl} \texttt{raise}\ (\Pi x : A.\, B)\ e & \equiv & \lambda x : A.\, \texttt{raise}\ B\ e \\ \texttt{match}\ (\texttt{raise}\ \mathcal{I}\ e)\ \texttt{return}\ P\ \texttt{with}\ \vec{p} & \equiv & \texttt{raise}\ P\ e. \end{array}$$

Said otherwise, we have a primitive that allows to escape from the surrounding context, i.e. exceptions. They obey a call-by-name equational theory, so they are quite distinct from their realworld counterparts. In particular, contrarly to call-by-value, it is not possible to catch exceptions at an arbitrary type. They

$$
\begin{array}{lcl}
[\Box] & := & (\boxbox, \Omega) \\
[x] & := & x \\
[\lambda x : A.\, M] & := & \lambda x : [\![A]\!].\, [M] \\
[M\ N] & := & [M]\ [N] \\
[\Pi x : A.\, B] & := & (\Pi x : [\![A]\!].\, [\![B]\!], \lambda (e : \mathbb{E})\, (x : [\![A]\!]).\, [B]_{\varnothing}\ e) \\
\boxbox & := & \Sigma A : \Box.\, (\mathbb{E} \to A) \\
[\![A]\!] & := & [A].\pi_1 \\
[A]_{\varnothing} & := & [A].\pi_2 \\
[\![\cdot]\!] & := & \cdot \\
[\![\Gamma, x : A]\!] & := & [\![\Gamma]\!], x : [\![A]\!]
\end{array}
$$

Figure 1: Exceptional Translation

can only be caught over inductive types, in which case the `catch` primitive is a generalization of the usual dependent eliminator. For instance, for booleans we can define a term

$$\vdash_{\mathcal{T}_{\mathbb{E}}} \mathtt{catch}_{\mathbb{B}} : \Pi P : \mathbb{B} \to \Box.\, P\ \mathtt{true} \to P\ \mathtt{false} \to (\Pi e : \mathbf{E}.\, P\ (\mathtt{raise}\ \mathbb{B}\ e)) \to \Pi b : \mathbb{B}.\, P\ b$$

which enjoys the obvious reduction rules. Alternatively, this can be presented as an extension of pattern-matching where branches may optionally refer to an exceptional case, similarly to a feature found in OCaml.

2.2 Translation

We do not dwell on the details here, refer to the previous paper [10] for more information. The model is given a syntactic translation $[\cdot]$ from $\mathcal{T}_{\mathbb{E}}$ into $\mathcal{T}$. The major idea is that we enrich types through the interpretation with their own `raise` function. In particular,

$$
\begin{array}{lcl}
\vdash_{\mathcal{T}_{\mathbb{E}}} A : \Box & \mapsto & \vdash_{\mathcal{T}} [A] : \boxbox \\
\vdash_{\mathcal{T}_{\mathbb{E}}} M : A & \mapsto & \vdash_{\mathcal{T}} [M] : [\![A]\!]
\end{array}
$$

where $\boxbox := \Sigma A : \Box.\, (\mathbb{E} \to A)$ and $[\![A]\!] := [A].\pi_1$. Conversion in $\mathcal{T}_{\mathbb{E}}$ is futhermore defined as conversion in $\mathcal{T}$ through the syntactic translation.

In what follows we fix a theory $\mathcal{T}$ containing CIC, a type $\vdash_{\mathcal{T}} \mathbb{E} : \Box$ and an arbitrary term of type $\vdash_{\mathcal{T}} \Omega : \mathbb{E} \to \boxbox$. Note that the type of this term is always inhabited, hence such a term can always be constructed regardless of the actual value of $\mathbb{E}$. With all this data, the translation of the negative fragment is given at Figure 1. We omit the universe annotations, but they can be added at the cost of a more verbose presentation.

Inductive types can be added straightforwardly, we sketch their translation here. Given an inductive type $\mathcal{I}$, we set $[\mathcal{I}] := (\mathcal{I}_{\mathbb{E}}, \mathcal{I}_{\varnothing})$ where $\mathcal{I}_{\mathbb{E}}$ is an inductive type in $\mathcal{T}$ whose constructors have type the pointwise translation of those of $\mathcal{I}$, plus one additional constructor $\mathcal{I}_{\varnothing} : \mathbb{E} \to \mathcal{I}_{\mathbb{E}}$. Constructors in the source theory are then interpreted by their counterpart from the extended inductive in the target theory. This interpretation is readily adapted to parameters and indices. As an example, we show the translation on lists below.

$$
\begin{array}{l}
\mathtt{Inductive\ list}\ (A : \Box) : \Box := \\
\quad \mid \mathtt{nil} : \mathtt{list}\ A \\
\quad \mid \mathtt{cons} : A \to \mathtt{list}\ A \to \mathtt{list}\ A
\end{array}
\quad \mapsto \quad
\begin{array}{l}
\mathtt{Inductive\ list}_{\mathbb{E}}\ (A : \boxbox) : \Box := \\
\quad \mid \mathtt{nil}_{\mathbb{E}} : \mathtt{list}_{\mathbb{E}}\ A \\
\quad \mid \mathtt{cons}_{\mathbb{E}} : A.\pi_1 \to \mathtt{list}_{\mathbb{E}}\ A \to \mathtt{list}_{\mathbb{E}}\ A \\
\quad \mid \mathtt{list}_{\varnothing} : \mathbb{E} \to \mathtt{list}_{\mathbb{E}}\ A
\end{array}
$$

This fully defines the introduction rules for inductive types. The subtle part lies in the interpretation of dependent elimination. Essentially, we interpret pattern-matching pointwise. This almost works thanks to the preservation of typing, except for the additional constructor which is not accounted for. For this one constructor, we use instead the raising primitive given by the return type of the pattern-matching, which corresponds effectively in reraising the exception in the error case.

Theorem 1. *$\mathcal{T}_{\mathbb{E}}$ is a model of* CIC.

We do not further detail the implementation of the aforementioned exception-related combinators, although they are quite direct.

2.3 Meta-theoretical properties

It is immediate to make the following observation.

Lemma 1. *$\mathcal{T}_{\mathbb{E}}$ is consistent iff $\not\vdash_{\mathcal{T}} \mathbb{E}$.*

Proof. We have $\bot_{\mathbb{E}} \cong \mathbb{E}$ in $\mathcal{T}$. □

It is more generally possible to show a weak form of canonicity for $\mathcal{T}_{\mathbb{E}}$, which is obtained directly from the interpretation of inductive types.

Lemma 2. *If $\mathcal{I}$ is an inductive type in $\mathcal{T}_{\mathbb{E}}$ with constructors $\vec{c}$ and $\vdash_{\mathcal{T}_{\mathbb{E}}} M : \mathcal{I}$, then either $M \equiv c_i$ for some i or $M \equiv \texttt{raise}\ \mathcal{I}\ e$ for some $\vdash_{\mathcal{T}_{\mathbb{E}}} e : \mathbf{E}$.*

The same result can be generalized to general inductive types but we will not discuss it further.

Interestingly, this construction can already be used to show a weak form of MP. Rather than the full fledged internal principle, it is possible to show the admissibility of *Markov's rule* in CIC. This derivation rule is defined as

$$\frac{\vdash_{\mathsf{CIC}} M : \neg\neg P}{\vdash_{\mathsf{CIC}} \langle M \rangle : P}$$

where P is some hereditarily positive type, e.g. $P := \Sigma n : \mathbb{N}.\, f\ n = \texttt{true}$. Note that this rule crucially requires the proof environment to be empty, thus forbidding to derive MP.

Lemma 3 (Friedman's trick)**.** *Markov's rule is admissible in* CIC.

Proof. The proof goes as follows. Let $\vdash_{\mathsf{CIC}} M : \neg\neg P$. By soundness of the exceptional model, for any parameter $\mathbb{E}$ we get a proof $\vdash_{\mathsf{CIC}} [M] : [\![\neg\neg P]\!]$. Unfolding the definitions, we get in CIC an isomorphism $[\![\neg\neg P]\!] \cong ([\![P]\!] \to \mathbb{E}) \to \mathbb{E}$. Now, the crux of the proof lies in the observation that if P is hereditarily positive, then there is actually a closed CIC term $\theta_P : [\![P]\!] \to P + \mathbb{E}$. Such a term is called a *storage operator* [8], and is defined by recursively matching over its argument and reconstructing a purified proof, and propagating the exception in case of error. The ability to do so is fundamentally tied to the positivity of P, which allows us to locally force exceptions on its inhabitants, and intuitively corresponds to enforcing a delimited form of call-by-value.

We now have all the required ingredients. Let us set $\mathbb{E} := P$, in which case we have a term $([\![P]\!] \to P) \to P$ by tinkering with $[M]$, as well as a storage operator $\theta_P : [\![P]\!] \to P + P$. By plugging both together with get a closed proof of P in CIC, hence concluding the proof. □

3 Implementing MP with dynamic types

We argue that knowing to implement Markov's rule is almost sufficient to get the full Markov's principle. Obviously, it is not enough *per se*, and we can actually show that $\mathcal{T}_{\mathbb{E}}$ disproves MP in general, but it already features the necessary basic computational extension to implement MP with a little bit of tweaking. The rationale is the following. Any finite number of closed uses of MP can be eliminated via the above trick. What prevents us from generalizing to MP is the requirement to pick $\mathbb{E}$ beforehand, as it will be fixed once and for all inside the exceptional translation.

A rethorical question: what if we were able to dynamically change the value of $\mathbb{E}$ to track the current set of Markov's rules being applied?

An affirmative answer: let's do it.

Definition 1 (Dynamic prompt)**.** We now assume that the target theory of the exceptional translation is an extension of CIC dubbed $\mathsf{CIC} + \mathcal{E}$ which contains the following data.

- A type $\vdash \mathcal{E} : \square$ called the type of *dynamic prompts*.
- A modality $\vdash \mathtt{local} : (\mathbb{N} \to \mathbb{B}) \to \square \to \square$.
- A term $\vdash \mathtt{return} : \Pi A : \square.\, A \to \Pi \varphi : \mathbb{N} \to \mathbb{B}.\, \mathtt{local}\ \varphi\ A$.
- A term $\vdash \mathtt{of}_\Pi : \Pi (A\, B : \square)\, (\varphi : \mathbb{N} \to \mathbb{B}).\, \mathtt{local}\ \varphi\ (A \to B) \to \mathtt{local}\ \varphi\ A \to \mathtt{local}\ \varphi\ B$.
- A term $\vdash \mathtt{to}_\Pi : \Pi (A\, B : \square)\, (\varphi : \mathbb{N} \to \mathbb{B}).\, (\mathtt{local}\ \varphi\ A \to \mathtt{local}\ \varphi\ B) \to \mathtt{local}\ \varphi\ (A \to B)$.
- A term $\vdash \mathtt{of}_{\mathcal{E}} : \Pi \varphi : \mathbb{N} \to \mathbb{B}.\, \mathtt{local}\ \varphi\ \mathcal{E} \to (\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) + \mathcal{E}$.
- A term $\vdash \mathtt{to}_{\mathcal{E}} : \Pi \varphi : \mathbb{N} \to \mathbb{B}.\, ((\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) + \mathcal{E}) \to \mathtt{local}\ \varphi\ \mathcal{E}$.

We furthermore assume that $\mathcal{E}$ is not inhabited[1] and that the following equations hold.[2]

$$\begin{array}{lcl} \mathtt{local}\ \varphi\ (\Sigma n : \mathbb{N}.\, A) & \equiv & \Sigma n : \mathbb{N}.\, \mathtt{local}\ \varphi\ A \\ \mathtt{local}\ \varphi\ (M = N) & \equiv & M = N \end{array}$$

Intuitively, this extension is defined in a theory where there is an ambient global cell p of type $\mathbb{N} \to \mathbb{B}$, which can be locally mutated. Only types are allowed to really observe this cell, otherwise this would break the behaviour of the pure fragment coming from CIC. For instance, the `local` modality allows to modify the cell by setting it to the pointwise boolean or between p and its argument. Likewise, $\mathcal{E}$ is defined by case analysis on p and morally encodes the fact that there is some $n : \mathbb{N}$ s.t. $p\ n = \mathtt{true}$.

Let $\mathsf{CIC}_{\mathcal{E}}$ be the theory obtained by unrolling the model construction of the previous section by setting $\mathcal{T} := \mathsf{CIC} + \mathcal{E}$ and $\mathbb{E} := \mathcal{E}$. We immediately get the following results.

Lemma 4. *We have the following.*

- *If* $\mathsf{CIC} + \mathcal{E}$ *is consistent then so is* $\mathsf{CIC}_{\mathcal{E}}$.
- *If* $\mathsf{CIC} + \mathcal{E}$ *enjoys canonicity then so does* $\mathsf{CIC}_{\mathcal{E}}$.

Proof. Both properties follow from the uninhabitedness of $\mathcal{E}$. □

[1] That is, that there is no closed term M s.t. $\vdash_{\mathsf{CIC}+\mathcal{E}} M : \mathcal{E}$.

[2] We could axiomatize them as we did for Π-types, but they happen to hold in the model so that it simplifies the proofs.

Note that we are walking a tight rope here. Consistency of $\mathsf{CIC}_{\mathcal{E}}$ only requires the absence of a term M s.t. $\vdash_{\mathsf{CIC}+\mathcal{E}} M : \mathcal{E}$, which is a much weaker property than internally having a term N such that $\vdash_{\mathsf{CIC}+\mathcal{E}} N : \neg\mathcal{E}$. The latter would have been enough to show the degeneracy of the construction, i.e. every type would be isomorphic to its exceptional translation.

Thanks to our fancily uninhabited type of exceptions, we actually get a non-trivial extension of CIC.

Theorem 2. MP *is derivable in* $\mathsf{CIC}_{\mathcal{E}}$.

Proof. The proof is actually quite easy. Up to isomorphism, and by interpreting Σ-types negatively for simplicity[3], MP is translated in $\mathsf{CIC} + \mathcal{E}$ as

$$[\![\mathsf{MP}]\!] := \Pi\varphi : \mathbb{N}_{\mathcal{E}} \to \mathbb{B}_{\mathcal{E}}.\, (((\Sigma n : \mathbb{N}_{\mathcal{E}}.\, (\varphi\ n = \mathtt{true}_{\mathcal{E}} + \mathcal{E})) \to \mathcal{E}) \to \mathcal{E}) \to \Sigma n : \mathbb{N}_{\mathcal{E}}.\, (\varphi\ n = \mathtt{true}_{\mathcal{E}} + \mathcal{E}).$$

By intuitionistic reasoning, it is logically equivalent to prove the simpler statement

$$\Pi\varphi : \mathbb{N}_{\mathcal{E}} \to \mathbb{B}_{\mathcal{E}}.\, (((\Sigma n : \mathbb{N}_{\mathcal{E}}.\, \varphi\ n = \mathtt{true}_{\mathcal{E}}) \to \mathcal{E}) \to \mathcal{E}) \to (\Sigma n : \mathbb{N}_{\mathcal{E}}.\, \varphi\ n = \mathtt{true}_{\mathcal{E}}) + \mathcal{E}.$$

The first part of the proof is to get rid of the exceptional encoding on φ. Let us assume for now some $\varphi : \mathbb{N}_{\mathcal{E}} \to \mathbb{B}_{\mathcal{E}}$. Let $\Uparrow_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}_{\mathcal{E}}$ be the injective function that recursively maps every pure constructor to its exceptional counterpart. Remark now we can easily build a function $\hat{\varphi} : \mathbb{N} \to \mathbb{B}$ s.t.

$$(\Sigma n : \mathbb{N}.\, \varphi\ (\Uparrow_{\mathbb{N}} n) = \mathtt{true}_{\mathcal{E}}) \leftrightarrow (\Sigma n : \mathbb{N}.\, \hat{\varphi}\ n = \mathtt{true})$$

by pattern-matching as

$$\begin{array}{lcll} \hat{\varphi}\ n & := & \mathtt{true} & \text{if } \varphi\ (\Uparrow_{\mathbb{N}} n) \equiv \mathtt{true}_{\mathcal{E}} \\ \hat{\varphi}\ n & := & \mathtt{false} & \text{if } \varphi\ (\Uparrow_{\mathbb{N}} n) \equiv \mathtt{false}_{\mathcal{E}} \\ \hat{\varphi}\ n & := & \mathtt{false} & \text{if } \varphi\ (\Uparrow_{\mathbb{N}} n) \equiv \mathbb{N}_{\varnothing}\ e. \end{array}$$

Moreover, we have both

$$\begin{array}{lcl} (\Sigma n : \mathbb{N}_{\mathcal{E}}.\, \varphi\ n = \mathtt{true}_{\mathcal{E}}) & \to & (\Sigma n : \mathbb{N}.\, \hat{\varphi}\ n = \mathtt{true}) + \mathcal{E} \\ (\Sigma n : \mathbb{N}.\, \hat{\varphi}\ n = \mathtt{true}) & \to & (\Sigma n : \mathbb{N}_{\mathcal{E}}.\, \varphi\ n = \mathtt{true}_{\mathcal{E}}) \end{array}$$

the first one being proved by recursively forcing the integer witness, and the second by trivial injection. By combining these results together, MP is thus reducible to proving

$$\Pi\varphi : \mathbb{N} \to \mathbb{B}.\, (((\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) \to \mathcal{E}) \to \mathcal{E}) \to (\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) + \mathcal{E}.$$

So far we have not used the additional structure on $\mathcal{E}$. The second part of the proof is precisely about the use of dynamic prompt to show the above statement. Let $\varphi : \mathbb{N} \to \mathbb{B}$ and $e : ((\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) \to \mathcal{E}) \to \mathcal{E}$, we show $(\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) + \mathcal{E}$.

By applying `return` to e we get a proof of

$$\mathtt{local}\ \varphi\ (((\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) \to \mathcal{E}) \to \mathcal{E}).$$

Now, by applying the various distribution lemmas and conversion rules, we get a proof of

$$((\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) \to \mathtt{local}\ \varphi\ \mathcal{E}) \to \mathtt{local}\ \varphi\ \mathcal{E}$$

which is further shown logically equivalent through the interaction between $\mathcal{E}$ and `local` to

$$((\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) \to (\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) + \mathcal{E}) \to (\Sigma n : \mathbb{N}.\, \varphi\ n = \mathtt{true}) + \mathcal{E}.$$

We conclude the proof by applying this assumption to the left injection of the sum type.

□

[3]It does not endanger the result, and simply removes translation cruft.

4 Did You Say Presheaf?

In this section, we justify the existence of $\mathsf{CIC} + \mathcal{E}$ via a presheaf construction. We assume that we live in an intuitionistic set theory with a countable number of inaccessible cardinal at hand. First, a generic result.

Lemma 5. *If $(\mathbb{P}, \leq)$ is a set-theoretic preorder, then $\mathsf{Psh}(\mathbb{P})$, the category of presheaves over $\mathbb{P}$, is a model of CIC.*

As already stated before, you need a bit of faith to believe in this statement. Showing the cartesian-closedness of the presheaf category fundamentally relies on function extensionality, and moreover you need the equalities to be strict in order to make any sense of the construction, although most category terrorists pretend not to care. Anyways, the same construction can be carried over in an extensional type theory, which is not-not-marginally better. As a matter of fact, we will use type-theoretic notations to work in the target set theory.

We do not detail the model here as it is fairly standard. Let us insist on one point though.

Lemma 6. *If $\mathbb{P}$ has a maximal element then $\mathsf{Psh}(\mathbb{P})$ enjoys canonicity.*

Proof. Inductive types are interpreted as constant presheaves, so that the value on the maximal element uniquely determines the value everywhere. □

Canonicity in intuitionistic set theory is a bit bland, as the above proposition is saying that given, say, a closed inhabitant of $\mathbb{B}$ in the model implies the constructive existence of a actual boolean which is "equal" to that term. Forget about nice definitional equalities.

Now we pick a specific preorder that we will use to show that we have a model of $\mathsf{CIC} + \mathcal{E}$. No suprise, it is the one from Coquand-Hofmann.

Definition 2. We define the preorder $\mathbb{P}$ as follows.

- Its objects are set-theoretic functions $\mathbb{N} \to \mathbb{B}$.
- We have $p \leq q$ whenever $\Pi n : \mathbb{N}.\, q\ n = \mathtt{true} \to p\ n = \mathtt{true}$.

Going down in the preorder means becoming truer, and the maximal element is the constantly false function. We can define the meet $p \wedge q$ of two forcing conditions p, q as their pointwise boolean or. Now let us implement in the rest of the section the $\mathcal{E}$ extensions in $\mathsf{Psh}(\mathbb{P})$.

A presheaf will be given by a pair (A, θ_A) where $A : \mathbb{P} \to \mathbf{Set}$ and $\theta_A : \Pi(p\, q : \mathbb{P})\, (\alpha : q \leq p).\, A\ p \to A\ q$, satisfying the expected litany of equations. A function from (A, θ_A) to (B, θ_B) will be given by a $\mathbb{P}$-indexed family of pointwise function with again the expected equations.

We will concentrate on the computational content of those objects and skip their functoriality and naturality side-conditions, these proof-irrelevant properties are showed in a companion Coq development.

First, we define $\mathcal{E} : \mathsf{Psh}(\mathbb{P})$.

$$\begin{array}{lcl} \mathcal{E}_p & := & \Sigma n : \mathbb{N}.\, p\ n = \mathtt{true} \\ \theta_{\mathcal{E}}\ p\ q\ \alpha\ ((n, e) : \mathcal{E}_p) & : & \mathcal{E}_q \\ & := & (n, \alpha\ n\ e) \end{array}$$

The presheaf $\mathcal{E}$ has no closed inhabitant, as it would in particular require that for any $p : \mathbb{N} \to \mathbb{B}$, $\mathcal{E}_p$ is inhabited as a set. But it is obvious that $\mathcal{E}_\top$ is empty, where $\top$ denotes the maximal element. Note that such an exotic type is typical of presheaf models, as its contents depends on the modal value of the global state.

Given $\varphi : \mathbb{N} \to \mathbb{B}$ and $A : \mathsf{Psh}(\mathbb{P})$ we need to define $\mathtt{local}\ \varphi\ A : \mathsf{Psh}(\mathbb{P})$. Stating things this way, we are doubly cheating here. First, we are treating an external quantification (in **Set**) as if it were an internal

quantification (in $\mathsf{Psh}(\mathbb{P})$). In practice it does not matter, all set-theoretic functions we will consider are embeddable as presheaf functions. Second, there is no *a priori* reason that $\mathbb{N} \to \mathbb{B}$ through the presheaf is interpreted as $\mathbb{N} \to \mathbb{B}$ in our set theory. Actually, it is not, but for any p, $(\mathbb{N} \to \mathbb{B})_p$ as a presheaf is isomorphic to $\mathbb{N} \to \mathbb{B}$ as a set, so that it does not matter in the end. We will make the same identifications for the other terms defined in this section. We define

$$\begin{array}{lcl} (\texttt{local}\ \varphi\ A)_p & := & A_{p \wedge \varphi} \\ \theta_{(\texttt{local}\ \varphi\ A)}\ p\ q\ \alpha\ (x : A_{p \wedge \varphi}) & : & A_{q \wedge \varphi} \\ & := & \theta_A\ (p \wedge \varphi)\ (q \wedge \varphi)\ (\alpha \wedge \varphi)\ x \end{array}$$

where $\alpha \wedge \varphi : q \wedge \varphi \leq p \wedge \varphi$.

Given $A : \mathsf{Psh}(\mathbb{P})$, $x : A_p$ and $\varphi : \mathbb{N} \to \mathbb{B}$, let us define $(\texttt{return}\ A\ x\ \varphi)_p$ as

$$\begin{array}{lcl} (\texttt{return}\ A\ x\ \varphi)_p & : & A_{p \wedge \varphi} \\ & := & \theta_A\ p\ (p \wedge \varphi)\ \pi_1\ x \end{array}$$

where $\pi_1 : p \wedge \varphi \leq p$.

The other combinators are defined in the Coq file and are straightforward. Special care has to be taken to decide which side of the sum type to choose when implementing $\texttt{of}_{\mathcal{E}}$, namely one must first look at the value for the current forcing condition p before looking at φ.

References

[1] Jeremy Avigad and Solomon Feferman. *The Handbook of Proof Theory*, chapter Gödel's functional ("Dialectica") interpretation, pages 337–405. North-Holland, 1999.

[2] Thierry Coquand and Martin Hofmann. A new method for establishing conservativity of classical systems over their intuitionistic version. *Mathematical Structures in Computer Science*, 9(4):323–333, 1999.

[3] Harvey Friedman. *Classically and intuitionistically provably recursive functions*, pages 21–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[4] Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings of the 25th Annual Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 50–56, 2010.

[5] Danko Ilik. *Constructive Completeness Proofs and Delimited Control. (Preuves constructives de complétude et contrôle délimité)*. PhD thesis, École Polytechnique, Palaiseau, France, 2010.

[6] Guilhem Jaber, Nicolas Tabareau, and Matthieu Sozeau. Extending Type Theory with Forcing. In *LICS 2012 : Logic In Computer Science*, pages 0–0, Dubrovnik, Croatia, June 2012.

[7] Stephen Cole Kleene. On the interpretation of intuitionistic number theory. *J. Symb. Log.*, 10(4):109–124, 1945.

[8] Jean-Louis Krivine. Classical logic, storage operators and second-order lambda-calculus. *Ann. Pure Appl. Logic*, 68(1):53–78, 1994.

[9] Pierre-Marie Pédrot. *A Materialist Dialectica. (Une Dialectica matérialiste)*. PhD thesis, Paris Diderot University, France, 2015.

[10] Pierre-Marie Pédrot and Nicolas Tabareau. Failure is not an option - an exceptional type theory. In Amal Ahmed, editor, *Proceedings of the 27th European Symposium on Programming Languages and Systems (ESOP 2018)*, volume 10801, pages 245–271, Thessaloniki, Greece, April 2018.

[11] A.S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*. Lecture Notes in Mathematics. Springer, 1973.