An Extensional Modified Realizability Topos

MSc Thesis *(Afstudeerscriptie)*

written by

Mees de Vries
(born September 2nd, 1992 in Amsterdam, The Netherlands)

under the supervision of **Dr. Benno van den Berg**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:	**Members of the Thesis Committee:**
March 30, 2017	Dr. Benno van den Berg
	Dr. Nick Bezhanishvili
	Dr. Sam van Gool
	Dr. Jaap van Oosten
	Dr. Jakub Szymanik

Institute for Logic, Language and Computation

Abstract

In this thesis, we construct and investigate a topos for Kreisel's modified realizability. The topos, like Kreisel's modified realizability, is characterized by the axiom of choice in all finite types and the principle of independence of premise. The model is constructed by a general method known as the tripos-to-topos construction. It is closely related to an existing topos, constructed for a modification of modified realizability by Troelstra, usually also called modified realizability.

We pay special attention to the subcategory of our topos on $\neg\neg$-separated objects, constructing it separately. This category is more accessible, and the logical features we look for in our topos are already present in this smaller category.

Contents

Chapter 1

Introduction

1.1 Modified Realizability

1.1. Let us put on our constructivist hats for a moment and consider what it means for the arithmetic sentence

$$\exists y(y + 7 = 12)$$

to be true. As constructivists, we might not be satisfied by observing that 5 *exists*; we might prefer the existence of a program which outputs the number 5. We could subsequently worry about the constructive content of $5 + 7 = 12$, but let us just assume that we can always evaluate arithmetic expressions of naturals, and that naturals have decidable equality. So, what is the content of the statement

$$\forall x \exists y(y + 7 = x),$$

where x, y range over the naturals? Of course, this sentence ought to be false, but what is the constructive content of that? Following our previous example, the statement would be true if there was a program which, given input x, could output a y such that $y + 7 = x$. Of course, such a program does not exist. So, even constructively, this statement is false. On the other hand,

$$\forall x(x \geq 7 \to \exists y(y + 7 = x))$$

ought to be true again. Inequalities of naturals, like equalities, are decidable, so the constructive content of an inequality should be small: essentially, just true or false. The sentence then states that there should be a program which given an x, and some kind of witness that $x \geq 7$ produces a y such that $y + 7 = x$. And indeed, if a function is guaranteed that $x \geq 7$ it can compute $y = x - 7$.

In fact, because $x \geq 7$ is decidable, we could even replace this with

$$\forall x \exists y(x \geq 7 \to (y + 7 = x)).$$

That is, there is a program which, given an x, produces a y such that *if* $x \geq 7$, then $y + 7 = x$. Otherwise, it can just produce a dummy result. The fact

that we can pull out the existential quantifier like this is called *independence of premise*, something which will become important soon.

Now let us expand our scope a little bit. Instead of talking just about natural numbers, we will also talk about functions of natural numbers. Let us write x^0 for variables ranging over the natural numbers, and f^1 for variables ranging over functions. Then, we can state our discussion above *internally* in this language as

$$\forall x^0 \exists y^0 (x \geq 7 \to (y + 7 = x)) \leftrightarrow \exists f^1 (\forall x^0 (x \geq 7 \to (f(x) + 7 = x))).$$

Of course, as per our previous discussion, f ranges not over *all* functions $\mathbb{N} \to \mathbb{N}$, but just the computable ones. This is just as well, as long as we are still being constructivist.

The equivalence we have just written looks a lot like a sort of axiom of choice: if for every x there is a y, then there is a function taking every x to such y. In fact, this form of the axiom of choice will also be a central topic in this thesis.

1.2. The language in which the sentences in the previous example live is the language HA^ω of Heyting arithmetic in finite types. The language of Heyting arithmetic, by itself, has just the usual arithmetical function symbols: it has symbols $0, s, +, \times$. The axioms for Heyting arithmetic are the axioms of Peano arithmetic, but in the context of intuitionistic first order logic.

The logic HA^ω extends HA by adding different sorts, or types. These are constructed inductively as follows: there is a sort called 0, which is the sort ranging over numbers; in particular, the symbols $0, s, +, \times$ all take and produce terms of type 0. Then, if σ, τ are sorts, so are $\sigma \to \tau$ and $\sigma \times \tau$. Thus, some examples of types are $0 \to (0 \to 0)$ and $((0 \times 0) \to 0) \to (0 \to (0 \to (0 \to 0)))$. We adopt the convention that $\times$ binds more strongly than $\to$, and that both $\times$ and $\to$ associate to the right, so that $0 \to 0 \to 0$ means $0 \to (0 \to 0)$. Furthermore, each natural number names a type recursively by defining the type $n + 1$ to be $n \to 0$. For instance, the type 3 is $((0 \to 0) \to 0) \to 0$. Intuitively, the terms of type $\sigma \to \tau$ should be understood as functions from objects of type σ to objects of type τ; and the terms of type $\sigma \times \tau$ should be understood as pairs of terms, one of type σ and one of type τ.

To encode thinking of $\sigma \to \tau$ as a function type, we have for each pair of types σ, τ a special function symbol $\mathrm{ev}_{\sigma,\tau}$ in the language for function evaluation; that is, if f is a term of type $\sigma \to \tau$, and x is a term of type σ, then $\mathrm{ev}_{\sigma,\tau}(f, x)$ is a is a term of type σ. For convenience we will always write $f(x)$ instead of $\mathrm{ev}_{\sigma,\tau}(f, x)$; the notation ev exists only to emphasize that the interpretation of the higher types as function types is not a higher-order feature of our logic, but something encoded in the first-order axioms of the theory. More generally, if f is a term of type $\sigma_1 \to \cdots \to \sigma_n \to \tau$, and $x_1, \ldots, x_n$ are terms of type $\sigma_1, \ldots, \sigma_n$ respectively, we usually write $f(x_1, \ldots, x_n)$ instead of $f(x_1)(x_2) \cdots (x_n)$.

Since $\sigma \times \tau$ ought to be a product type, we have for each pair of types σ, τ

three constants in the language:

$$P_{\sigma,\tau} : \sigma \to \tau \to \sigma \times \tau,$$
$$P^1_{\sigma,\tau} : \sigma \times \tau \to \sigma,$$
$$P^2_{\sigma,\tau} : \sigma \times \tau \to \tau,$$

where the colon should be read as "of type". We will always abbreviate $P_{\sigma,\tau}(x, y)$, which is itself an abbreviation of $\mathrm{ev}(\mathrm{ev}(P_{\sigma,\tau})(x))(y)$, as $\langle x, y\rangle$, and $P^i_{\sigma,\tau}(a)$ as a_i. Furthermore, we abbreviate $\langle a_1, \cdots \langle a_{n-1}, a_n\rangle \cdots \rangle$ as $\langle a_1, \ldots, a_n\rangle$, and when a is a term of $\sigma_1 \times \cdots \times \sigma_n$ we give $a_1, \ldots, a_n$ their obvious meaning as the coordinates of a. To give these symbols their intended meanings, HA^ω has the axioms

$$\forall x^\sigma \forall y^\tau (\langle x, y\rangle_1 = x),$$
$$\forall x^\sigma \forall y^\tau (\langle x, y\rangle_2 = y),$$
$$\forall p^{\sigma\times\tau} (\langle p_1, p_2\rangle = p).$$

We finally abbreviate $f(\langle x_1, \ldots, x_n\rangle)$ also as $f(x_1, \ldots, x_n)$; it should be clear from context when f is applied several arguments in order or to a tuple.

The evaluation function allows us to evaluate functions in the language. We also want to be able to talk about simple functions: in fact, we want to be able to use the (typed) lambda calculus. Rather than axiomatizing it from the ground up, we add some combinators to the language. These are, for types σ, τ, υ

$$I_\sigma : \sigma \to \sigma,$$
$$K_{\sigma,\tau} : \sigma \to \tau \to \sigma,$$
$$S_{\sigma,\tau,\upsilon} : (\upsilon \to \sigma \to \tau) \to (\upsilon \to \sigma) \to \upsilon \to \tau.$$

These correspond to the "ordinary" SKI combinators, which we enforce by adding axioms for their behaviour as follows:

$$\forall x^\sigma (I_\sigma(x) = x),$$
$$\forall x^\sigma \forall y^\tau (K_{\sigma,\tau}(x, y) = x),$$
$$\forall x^\sigma \forall y^\tau \forall z^\upsilon (S_{\sigma,\tau,\upsilon}(x, y, z) = x(z, y(z))).$$

As is well known (see e.g. [2]), any well-typed lambda term can be translated into a term using just these combinators and ev, and we thus have rudimentary computation. Since we have natural numbers at our disposal, we additionally want to be able to build functions recursively. To this end, we add for each type σ another combinator R_σ of type $\sigma \to (\sigma \to 0 \to \sigma) \to 0 \to \sigma$, whose behaviour is defined by the axioms

$$\forall x^\sigma \forall f^{\sigma\to 0\to\sigma} (R_\sigma(x, f, 0) = x),$$
$$\forall x^\sigma \forall f^{\sigma\to 0\to\sigma} \forall n^0 (R_\sigma(x, f, sn) = f(R(x, f, n), n)).$$

With the recursion scheme, we can in particular define and use all primitive recursive functions in HA^ω.

For a much more detailed overview and analysis of HA^ω, see for example the first chapter of [7].

1.3. We can interpret HA^ω in a variety of structures. One obvious choice is to take a classical approach, and interpret the type 0 as the natural numbers, the product of types as the Cartesian product, and $\sigma \to \tau$ as the set of all functions from the set interpreting σ to the set interpreting τ. The symbols all take on their usual meanings.

Of course, this is not very constructive. In particular, it goes against our computability interpretation of the sentences, since the objects of type 1 range over more than just computable functions, which also means that the content of the sentence $\forall x^0 \exists y^0 \phi(x, y)$ can no longer be the existence of a program which computes for each x a y such that $\phi(x, y)$ holds – for instance, just take the formula $f(x) = y$. Thus, we aim for a more computational interpretation. To this end, let us recall some basics of recursion theory.

1.4. We can encode all Turing machines as natural numbers, in such a way that a fixed universal Turing machine can evaluate them, given the code. We fix such a Gödel numbering for the rest of the thesis. Given natural numbers n, m we write $nm \in \mathbb{N}$ for the output of the universal machine if it is run with code n and argument m, if it terminates. If we want to specify that nm exists we write $nm{\downarrow}$. This operation associates to the left, so that $nmk = (nm)k$. If we write an equation with this operation, unless otherwise specified we mean that one side exists if and only if the other side does, and if so they are equal.

A few Turing machines come back often, so their codes get specific names.

- $\mathbf{k}, \bar{\mathbf{k}} \in \mathbb{N}$ are such that $\mathbf{k}ab = a$ and $\bar{\mathbf{k}}ab = b$.
- $\mathbf{p}, \mathbf{p_1}, \mathbf{p_2} \in \mathbb{N}$ are such that $\mathbf{p_1}(\mathbf{p}ab) = a$, $\mathbf{p_2}(\mathbf{p}ab) = b$ and $\mathbf{p}(\mathbf{p_1}a)(\mathbf{p_2}a) = a$.
- $\mathbf{i}$ is such that $\mathbf{i}a = a$.
- $\mathbf{s}$ is such that $\mathbf{s}xyz = xz(yz)$.
- $\mathbf{r}$ is such that $\mathbf{r}xf0 = x$, and $\mathbf{r}xf(n+1) = f(\mathbf{r}xfn)n$.

We can choose our coding in such a way that $0a = 0$ and $\mathbf{p}00 = 0$, and we choose to do so. Also, we typically abbreviate $\mathbf{p}ab$ as $\langle a, b \rangle$, and abbreviate $\mathbf{p_1}a, \mathbf{p_2}a$ as a_1, a_2 respectively.

If t is a closed term in the lambda calculus (for example, $\lambda n.\mathbf{p}nn$), there is a Turing machine for the computation t represents, and we represent the code for that machine by t itself. In particular, we have a code $\mathbf{b} = \lambda nmk.n(mk)$ for composition, and we usually represent it as an infix circle (so that $\mathbf{b}nm = n \circ m$).

1.5 Definition. For each finite type σ we define a set σ_{HEO}, the *hereditarily effective operations of type* σ, together with a function $\rho_{\sigma_{\mathrm{HEO}}} : \sigma_{\mathrm{HEO}} \to \mathcal{P}(\mathbb{N})$. For the type 0, we set $0_{\mathrm{HEO}} = \mathbb{N}$, and $\rho_{0_{\mathrm{HEO}}}(n) = \{n\}$.

If $f : \sigma_{\mathrm{HEO}} \to \tau_{\mathrm{HEO}}$ is a function and r is a natural number, we say that r *realizes* f if for all $x \in \sigma_{\mathrm{HEO}}$ and all $n \in \rho_{\sigma_{\mathrm{HEO}}}(x)$ we have $rn{\downarrow}$ and $rn \in \rho_{\tau_{\mathrm{HEO}}}(f(x))$. Then $(\sigma \to \tau)_{\mathrm{HEO}}$ consists of all functions $f : \sigma_{\mathrm{HEO}} \to \tau_{\mathrm{HEO}}$ which have a realizer, and $\rho_{(\sigma\to\tau)_{\mathrm{HEO}}}(f)$ is the set of those realizers.

We set $(\sigma \times \tau)_{\mathrm{HEO}} = \sigma_{\mathrm{HEO}} \times \tau_{\mathrm{HEO}}$ and define $\rho_{(\sigma\times\tau)_{\mathrm{HEO}}}(x, y) = \{\mathbf{p}nm \mid n \in \rho_{\sigma_{\mathrm{HEO}}}(x), m \in \rho_{(\tau)}(y)\}$.

1.6. An interpretation of HA^ω in HEO sits closer to our intention. For each type σ, we interpret it as σ_{HEO}. The successor, addition and multiplication all keep their interpretation. The symbols S, K, I, R can also keep their original interpretations, since the typed SKI combinators and bounded recursion can only produce computable functions from computable functions.

A realizer of an element of a finite type should be thought of as a representative for that element, and operations on a finite type are typically defined on realizers. This is consistent with how we defined function types in HEO: they are computable functions of the realizers.

From now on, when we refer in the meta-language to a type as if it has been interpreted without explicitly mentioning an interpretation, we mean its interpretation in HEO. Furthermore, we drop the subscript, so that $\sigma = \sigma_{\mathrm{HEO}}$ unless context suggests otherwise.

1.7. So far, we have skirted the subject of what it means for two elements of a type other than 0 to be equal. Of course, equality of natural numbers is nice and decidable. For higher types, this is slightly more complicated. If we take the interpretation in 1.6, equality is no longer decidable; that would require determining for two codes of computable total functions whether or not they represent the same function.

However, in our interpretation in HEO, for terms t, t' of the appropriate type, we do have the equivalences

$$\begin{aligned} t =_{\sigma\to\tau} t' &\iff \forall x^\sigma (tx =_\tau t'x), \\ t =_{\sigma\times\tau} t' &\iff t_1 =_\sigma t'_1 \wedge t'_2 =_\tau t'_2. \end{aligned} \tag{1.1}$$

This means that with recursive translations, we can interpret equality in the higher types in terms of equality for 0. Thus, we make the choice to allow only equality of natural numbers as atomic formulas, and take the "higher equalities" as defined by the above translation. This allows our atomic formulas to always be decidable.

The equivalences 1.1 are not provable from HA^ω themselves. A similar but different interpretation of the finite types, called the *hereditarily recursive operations* or HRO, lets $0_{\mathrm{HRO}} = \mathbb{N}$ and, recursively,

$$\begin{aligned} (\sigma \to \tau)_{\mathrm{HRO}} &= \{r \in \mathbb{N} \mid \forall n \in \sigma_{\mathrm{HRO}}(rn{\downarrow} \wedge rn \in \tau_{\mathrm{HRO}})\} \\ (\sigma \times \tau)_{\mathrm{HRO}} &= \{\langle n, m\rangle \mid n \in \sigma_{\mathrm{HRO}}, m \in \tau_{\mathrm{HRO}}\}. \end{aligned}$$

This structure also has an interpretation of HA^ω, and there distinct elements of $\sigma \to \tau$ can be extensionally equal. In this structure, equality is clearly decidable for all higher types, since it is just equality of natural numbers.

Adding the equivalences 1.1 to HA^ω gives us the new theory $\mathrm{E{-}HA}^\omega$ for *extensional* Heyting arithmetic in all finite types. This is what we work with from now on.

For a deeper analysis of HEO and HRO and their respective differences, see [7], chapter 2 paragraph 4. For a more detailed overview of possible definitions of equality, with their various advantages and problems, see [8].

1.8. Following the interpretation in 1.6, we will define what it means for an element of a finite type in HEO to *realize* a sentence. Recall, when we refer to an element of a higher type, we are referring to an element of that type in HEO. For instance, saying that f is of type 1 means that it is a total computable function of natural numbers.

First, we will assign to each formula ϕ in the language of HA^ω a finite type $\tau(\phi)$; if ϕ is a sentence, this will be the type from which realizers for ϕ may be drawn. If ϕ is an atomic formula, then $\tau(\phi) = 0$. Now we define recursively

$$\begin{aligned}
\tau(\phi \wedge \psi) &= \tau(\phi) \times \tau(\psi),\\
\tau(\phi \vee \psi) &= 0 \times \tau(\phi) \times \tau(\psi),\\
\tau(\phi \to \psi) &= \tau(\phi) \to \tau(\psi),\\
\tau(\forall x^\sigma \phi) &= \sigma \to \tau(\phi),\\
\tau(\exists x^\sigma \phi) &= \sigma \times \tau(\phi).
\end{aligned}$$

The elements of a type $\tau(\phi)$ are referred to as the *potential realizers* of ϕ.

Then, we define a relation $\mathbf{mr}$ between sentences [1] ϕ and elements $a \in \tau(\phi)$, where $a^{\tau(\phi)}$ $\mathbf{mr}$ ϕ is to be read as "a modifiedly realizes ϕ" or just "a realizes ϕ". More commonly, we say that a is an *actual realizer* of ϕ. If ϕ is an atomic sentence, then we have a^0 $\mathbf{mr}$ ϕ if and only if ϕ is true. This is fine, because the truth of atomic sentences is decidable. Then we define recursively:

$$\begin{aligned}
a^{\tau(\phi)\times\tau(\psi)}\ \mathbf{mr}\ \phi \wedge \psi &\iff a_1\ \mathbf{mr}\ \phi \text{ and } a_2\ \mathbf{mr}\ \psi,\\
a^{0\times\tau(\phi)\times\tau(\psi)}\ \mathbf{mr}\ \phi \vee \psi &\iff (a_1 = 0 \text{ and } a_2\ \mathbf{mr}\ \phi) \text{ or}\\
&\qquad (a_1 \neq 0 \text{ and } a_3\ \mathbf{mr}\ \psi),\\
a^{\tau(\phi)\to\tau(\psi)}\ \mathbf{mr}\ \phi \to \psi &\iff \text{for each } k \in \tau(\phi), \text{ if } k\ \mathbf{mr}\ \phi \text{ then } ak\ \mathbf{mr}\ \psi,\\
a^{\sigma\to\tau(\phi)}\ \mathbf{mr}\ \forall x^\sigma \phi &\iff \text{for each } k \in \sigma, \text{ we have } ak\ \mathbf{mr}\ \psi[k/x],\\
a^{\sigma\times\tau(\phi)}\ \mathbf{mr}\ \exists x^\sigma \phi &\iff a_2\ \mathbf{mr}\ \phi[a_1/x].
\end{aligned}$$

Thus, when a $\mathbf{mr}$ ϕ, this should be understood as constructive evidence that ϕ is true. For instance, constructive evidence for an existential statement consists of a witness for the existential statement, together with a proof that the statement actually holds for that witness. Not surprisingly, we have the following theorem.

[1] To be *very* precise, the right element in the relation $\mathbf{mr}$ need not be a sentence of HA^ω, but a sentence of a language extending HA^ω with a finite number of constants, each of which has an intended interpretation as an element of a finite type in HEO.

1.9 Theorem. *If ϕ is a sentence of* HA^ω*, and* $\mathrm{E{-}HA}^\omega \vdash \phi$*, then there is an* $a \in \tau(\phi)$*, given by a closed term in the language of* HA^ω*, such that* a **mr** ϕ*.*

Proof. Clearly if $\phi \to \psi$ and ϕ are both realized, then so is ψ by way of function application. Furthermore, all logical axioms, as well as the axioms of $\mathrm{E{-}HA}^\omega$ are realized by closed terms in the language of HA^ω. By way of example, equational axioms – the axioms about P, P^1, P^2, I, K, S – are all realized by 0, and the recursion scheme

$$\phi(0) \to \forall n(\phi(n) \to \phi(s(n))) \to \forall n(\phi(n))$$

is realized by $\mathbf{r}$. □

1.10. The system we have just described is that of *modified realizability*. It was first introduced by Kreisel in [5], and later in [6], in order to study HA^ω. Modified realizability, as the name suggests, is a variant of an earlier system which goes by the name of *realizability*.

This "original realizability" was invented by Kleene in [4] as a way of formalizing the constructive principle that mathematics should be effective: from a proof about an object we ought to be able to effectively determine, or *compute*, information about the object. Recall the way we considered formulas of the form $\forall x \exists y \phi(x, y)$ in the introduction.

In the original system, which was defined for HA, a realizer was always a number. Where we use a pair, ordinary realizability used a primitive recurisve bijection $\mathbb{N}^2 \to \mathbb{N}$, and where we use a function type, ordinary realizability used a Gödel number for a recursive function. In particular, potential realizers do not come into play. Other than that, the inductive definitions are quite similar. Ordinary realizability will play a minor role in this thesis, mostly as a contrasting example.

Realizability has many variants, most of which we will not go into in this thesis. For an overview of "ordinary" realizability, modified realizability, and many other variants see for example [7]. For a historical overview of realizability, with a focus on categorical developments, see [10].

1.11. Unsurprisingly, the converse to theorem 1.9 does not hold. One example we already discussed is the *axiom of choice for finite types*. For types σ, τ, with a parameter type α, this axiom takes the form

$$\mathrm{AC}_{\sigma,\tau} : \forall a^\alpha (\forall x^\sigma \exists y^\tau (P(x, y, a)) \to \exists f^{\sigma \to \tau} \forall x^\sigma (P(x, f(x), a))).$$

There is a term realizing this: it is given by $\lambda dr.\langle \lambda x.(rx)_1, \lambda x.(rx)_2 \rangle$. However, it cannot be proved from the axioms of $\mathrm{E{-}HA}^\omega$ themselves, something we happen to prove along the way near the end of the next chapter.

Another example is the *independence of premise*. To make our discussion from 1.1 more precise, we need to have formalize what property of $x \geq 7$ allowed us to move the existential quantifier past it. With our conventions from 1.4, that $00 = 0$ and $\langle 0, 0 \rangle = 0$, every finite type is inhabited by an element with code 0. Inductively, 0 is obviously an element of $\mathbb{N}$; if 0 is a code for an element of τ,

then the constant 0 function, which is coded by 0 itself, is an element of $\sigma \to \tau$; and if 0 is a code for elements from both σ, τ then the pair of those elements in $\sigma \times \tau$ will also have 0 for their code. By a small abuse of notation, we will use 0 to refer to the term of any type which has 0 for a code.

Thus, 0 occurs as a sort of special, shared potential realizer between all types. We are looking for a family of formulas $P(\vec{x})$ for which, no matter the values of the variables $\vec{x}$, whenever P is realized, then 0 is an actual realizer of P. Such formulas in particular have the property

$$0 \ \mathbf{mr} \ \forall \vec{x}(\neg\neg P(\vec{x}) \to P(\vec{x})),$$

so under modified realizability, such formulas are all equivalent to negated formulas, namely, their own double negation.

In fact, negated formulas also all have this property: since $\bot$ has no actual realizers, this means $\neg P = P \to \bot$ only has actual realizers when P has no actual realizers. In that case, $0 \ \mathbf{mr} \ \neg P$, and more generally any element of $\tau(\neg P)$ realizes $\neg P$.

Hence, the general form of independence of premise is, for our purposes,

$$\forall x((\neg P(x) \to \exists y(R(x,y))) \to \exists y(\neg P(x) \to R(x,y))),$$

and it is realized by $\lambda xr.\langle (r0)_1, \lambda a.(r0)_2\rangle$. Just like the axiom of choice, it does not follow from the axioms of $\mathrm{E{-}HA}^\omega$.

1.12. The above makes an attempt at characterizing which statements are realizable. Of course, we would never be able to give a complete characterization: since for any sentence ϕ, either ϕ or $\neg\phi$ is realized, in particular for any Turing machine T, either $\mathrm{Halts}(T)$ or $\neg\mathrm{Halts}(T)$ is realized. No actual metatheory is strong enough to determine that for every T.

Despite this, there is a sense in which $\mathrm{E{-}HA}^\omega$ augmented with the axiom of choice for finite types and independence of premise is as strong as modified realizability: $\mathrm{E{-}HA}^\omega$ itself thinks they are the same. Knowing this, it makes sense to focus on AC and IP when considering a potential model for modified realizability.

We will now make the above more precise.

1.13. Our definition of (modified) realizability lies in the metatheory, but there is really no reason for that. For every sentence ϕ in the language, we can define a formula $a^{\tau(\phi)} \ \underline{\mathbf{mr}} \ \phi$ of one free variable $a^{\tau(\phi)}$ as follows. If ϕ is atomic, then $a^{\tau(\phi)} \ \underline{\mathbf{mr}} \ \phi$ is just ϕ. Otherwise, we define recursively

$$\begin{aligned}
a^{\tau(\phi)\times\tau(\psi)} \ \underline{\mathbf{mr}} \ \phi \wedge \psi &= (a_1 \ \underline{\mathbf{mr}} \ \phi) \wedge (a_2 \ \underline{\mathbf{mr}} \ \psi),\\
a^{0\times\tau(\phi)\times\tau(\psi)} \ \underline{\mathbf{mr}} \ \phi \vee \psi &= (a_1 = 0 \wedge a_2 \ \underline{\mathbf{mr}} \ \phi) \vee (a_1 \neq 0 \wedge a_3 \ \underline{\mathbf{mr}} \ \psi),\\
a^{\tau(\phi)\to\tau(\psi)} \ \underline{\mathbf{mr}} \ \phi \to \psi &= \forall k^{\tau(\phi)}(k \ \underline{\mathbf{mr}} \ \phi \to ak \ \underline{\mathbf{mr}} \ \psi),\\
a^{\sigma\to\tau(\phi)} \ \underline{\mathbf{mr}} \ \forall x^\sigma \phi &= \forall k^\sigma(ak \ \underline{\mathbf{mr}} \ \psi[k/x]),\\
a^{\sigma\times\tau(\phi)} \ \underline{\mathbf{mr}} \ \exists x^\sigma \phi &= a_2 \ \underline{\mathbf{mr}} \ \phi[a_1/x].
\end{aligned}$$

Then, since the realizer from theorem 1.9 can be taken to be a closed term in the language of HA^ω, it follows that $\mathrm{E{-}HA}^\omega \vdash \phi$ implies $\mathrm{E{-}HA}^\omega \vdash \exists x^{\tau(\phi)}(x\ \underline{\mathbf{mr}}\ \phi)$.

1.14. As observed, the converse to this is not true. Both the axiom of choice and independence of premise are realized by closed terms of HA^ω, so certainly for each instance ϕ of $\mathrm{AC}_{\sigma,\tau}$ we have $\mathrm{E{-}HA}^\omega \vdash \exists x^{\tau(\phi)}(x\ \underline{\mathbf{mr}}\ \phi)$, but $\mathrm{E{-}HA}^\omega \nvdash \mathrm{AC}_{\sigma,\tau}$. Thus there is the obvious question: how much stronger is modified realizability than provability in $\mathrm{E{-}HA}^\omega$, and in what way?

The following theorem, whose proof can be found in [7] (theorem 3.4.8), states that the axiom of choice and independence of premise are precisely what we need.

1.15 Theorem. *Writing* IP *for the axiom scheme consisting of all instances of independence of premise, and* AC *for the axiom scheme consisting of all instances of the axiom of choice for all finite types,*

$$\mathrm{E{-}HA}^\omega + \mathrm{IP} + \mathrm{AC} \vdash \phi \iff \mathrm{E{-}HA}^\omega \vdash \exists x^{\tau(\phi)}(x\ \underline{\mathbf{mr}}\ \phi).$$

1.2 Heyting Categories and Toposes

1.16. Rather than restricting ourselves to just the finite types, we want to take a wider view, not just of functions of natural numbers but of a wider scope of mathematics. We want to be able to talk about more objects, but still keep the flavour of modified realizability.

Logics can be interpreted in categories with sufficient structure for that particular logic. Then, any object in the category can serve as a sort in the logic. Thus, we might be interested in finding a category which has enough structure to support an interpretation of HA^ω – that is, of first-order intuitionistic logic, with product and function types and something like the natural numbers – which also has a wealth of other objects to work with.

In order for a category to sensibly allow an interpretation of intuitionistic logic, it needs to be a *Heyting category*. That is, for any object, its poset of subobjects needs to be a Heyting algebra; and these Heyting algebras need to interact well with pullback maps. Then the connectives in the logic get interpreted as Heyting algebra operations, and quantifiers as adjoints to the pullback.

A good candidate for a categorical interpretation of the natural numbers is a so-called *natural numbers object*, which is essentially an object that allows recursion.

A function type is most sensibly interpreted as an exponential in a category. Hence, if a category has the right exponentials, we can interpret function sorts in them. In particular, this works if the category is Cartesian closed, that is, has *all* exponentials.

To sum up, we are interested in a Cartesian closed Heyting category with a natural numbers object, whose internal logic reflects that of modified realizability, particularly for the finite types. The first thing we do, once we move beyond

the introductory chapter, will be to define and investigate such a category – this will be the category of *modified assemblies*.

1.17. Although Heyting categories allow us to develop quite a bit of mathematics, from the point of view of a full foundation of mathematics they are lacking. We are able to talk about functions, but we are generally not able to talk about subsets, or predicates, on our objects. Classically, this is less of a problem: a subset of a set A is the same as a function $A \to 2$. In constructive mathematics, this typically no longer holds, and Heyting categories are not powerful enough to be able to talk about subsets of or predicates on objects. This limits them from developing a full scale mathematics.

For that, we move a step "further", to toposes. Toposes distinguish themselves from Cartesian closed Heyting categories by having, internally, power objects, which are objects which function like the power set in set theory. In fact, toposes allow a minimalist definition as finitely complete categories with power objects.

Where Heyting categories allow interpretation of first-order logic, toposes allow interpretations of higher-order logics. When a variable P is meant to range over the subobjects or predicates of a sort, it can instead range over the "elements" of the power object of that sort.

If a topos furthermore has a natural numbers object, functioning somewhat like the axiom of infinity in ZF set theory, this is enough to develop an entire mathematics in its internal logic. Taking this point of view, most mathematicians are investigating the internal structure of Set, but we might like a different topos to work in.

In particular, we are interested in a topos with natural numbers object whose internal logic corresponds to that of modified realizability. That will be the terminal goal of this thesis: to define and investigate a category called MRT, the *modified realizability topos*.

1.18. In fact, an object called "the modified realizability topos" already exists. It was originally defined in an unpublished document by Grayson in 1981; an accessible (and corrected) construction by Van Oosten can be found in [9]. To distinguish it from the object we will define, we call it Gray, the "Grayson topos", but we emphasize that that is not standard terminology. However, because this thesis is about the topos we will call MRT, giving it a longer but more descriptive name (like "extensional modified realizability topos") would be cumbersome.

The topos Gray exists to model something which is *also* called modified realizability. This variant was investigated by Troelstra in [7]. Using the terminology of 1.7, we could informally explain the difference by saying that Troelstra's variant is based on HRO rather than HEO. The difference in construction between the two is sufficiently small, that we can for the most part follow the construction of Gray to construct MRT, and in fact while constructing MRT we also construct Gray "along the way". Due to the similarities in construction, the topos Gray is related MRT by an essential geometric functor. However, these small differences are strong enough that the axiom of choice for finite types does not hold in Gray, while it does in MRT.

What we called "ordinary realizability" earlier, in 1.10, also has an associated topos: the so-called *effective topos* Eff. This topos is arguably the simplest and most famous of the realizability toposes, and is often used as a contrast for geometric toposes.

Similarly, there also already exist categories of assemblies and modified assemblies. The assemblies Asm are a Heyting subcategory of Eff, which has two nice properties: it consists of the so-called $\neg\neg$-separated objects, and furthermore Eff is the ex-reg completion of Asm, which more or less means that every object of Eff is the well-behaved quotient of an assembly.

Our category MAss will share the first property of Asm (with respect to MRT), but not the second; unfortunately, the two properties no longer coincide in MRT. All realizability toposes – at least those constructed by way of triposes – have a nice subcategory with the *second* property, and these are called categories of assemblies in [11]; so our choice of terminology differs from that case.

Furthermore, a category called "modified assemblies" has been investigated before, by Streicher, whose definition is quoted in [9]. It matches neither the general definition from [11] nor our definition.

We will also, along the way, construct an analogue of MAss for Gray we will call GrAss, or *Grayson assemblies*. While this category does not really have an explicit existing name – in particular, it has never been called a category of modified assemblies – it appears in [10] as the subcategory of Gray consisting of the "sub-∇s".

1.19. The remainder of this thesis consists roughly of three parts, divided into three chapters. First, we introduce the category of modified assemblies, and, while developing some general theory about Heyting categories, aim to show that it is a viable candidate for a "category of modified realizability".

In the second part, we generalize the notion of a Heyting category to that of a hyperdoctrine. Certain hyperdoctrines have enough internal structure that from them we can construct toposes: such hyperdoctrines are called triposes. We develop basic theory about triposes, and how they give rise to toposes. In particular, they give rise to the modified realizability topos MRT, as well as Gray and other toposes.

In the third and final part, we investigate MRT in particular. We consider the relation between MRT and the category of modified assemblies – in fact, the latter is a subcategory of the former. In doing so we demonstrate that it is indeed an appropriate category for a mathematics based on modified realizability.

Acknowledgements

First and foremost, I would like to thank Benno van den Berg, my supervisor during this project. His insight and boundless patience guided me through a process that was not always smooth. This thesis is also indebted to some of my fellow thesis writing students, whose company these past months made the process much more enjoyable.

Chapter 2

Modified Assemblies

2.1 Modified Assemblies as a Heyting Category

2.1. In order to define the category of modified assemblies, we first introduce some notation useful for realizability. If A, B are sets of natural numbers, and r is a natural number, we write

$$r : A \to B$$

when for each $a \in A$, we have $ra{\downarrow}$ and $ra \in B$. If furthermore A, B are equipped with equivalence relations $\sim_A, \sim_B$ respectively, then we write

$$r : (A, \sim_A) \to (B, \sim_B)$$

if $r : A \to B$ and whenever $a \sim_A a'$, then also $ra \sim_B ra'$. This last condition is also called *equivariance* of r.

2.2 Definition. A *modified assembly* is a quadruple $(X, P_X, \sim, \rho_X)$ of a set X, a set of natural numbers P_X with $0 \in P_X$, called the *potential realizers*, an equivalence relation $\sim$ on P_X, and a mapping

$$\rho_X : X \to \mathcal{P}(P_X)$$

such that for each $x \in X$, the set $\rho_X(x)$ is not empty. The elements of $\rho_X(x)$ are called *actual* realizers for x. We refer to $(X, P_X, \sim, \rho_X)$ as X whenever this does not cause confusion.

If X, Y are modified assemblies, a function $f : X \to Y$ of the underlying sets is called a *morphism of modified assemblies* if there is an $r \in \mathbb{N}$ such that $r : (P_X, \sim) \to (P_Y, \sim)$ and for each $x \in X$ also $r : \rho_X(x) \to \rho_Y(f(x))$. This r is called a *realizer* for f.

Clearly, morphisms compose to give morphisms and the identity function is a morphism, so we have a category MAss of modified assemblies and morphisms of assemblies.

2.3. If we drop the equivalence relation from 2.2, or equivalently, only consider objects for which the equivalence relation is total, we obtain a similar but distinct category. This is the category GrAss, announced in 1.18, that will play a role with respect to Gray similar to the role MAss plays with respect to MRT. Everything we say in the first two sections of this chapter applies as much to GrAss as it does to MAss, as long as we ignore everything to do with equivalence relations and equivariance. The differences between the categories only become apparent once we start investigating the internal logic of MAss.

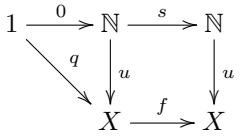
If we drop the sets of potential realizers altogether, and keep only the sets of actual realizers, we obtain the so-called category of assemblies Asm, playing that same role with respect to Eff. This category is still similar to MAss and GrAss in the sense that most of what is constructed in these two sections holds for it; but its logic is further from that of GrAss and MAss than the logics of those categories are from each other.

2.4. Consider the modified assembly $\mathbb{N} = (\mathbb{N}, \mathbb{N}, =, \rho_{\mathbb{N}})$, where $\rho_{\mathbb{N}}(n) = \{n\}$. We interpret the actual realizers of a point in a modified assembly as codes for that point, that is, as information about which point of the structure it is. Hence, this modified assembly seems to represent the natural numbers in the category. If instead we had defined, for instance, $\rho'_{\mathbb{N}}(n) = \mathbb{N}$, we would obtain an object in which the points are "computationally indistinguishable": from this object, we would only be able to map the natural numbers into sets of points which share an actual realizer.

The modified assembly $(1, \mathbb{N}, \mathbb{N} \times \mathbb{N}, * \mapsto \mathbb{N})$ is a terminal object in MAss. The function $0 : 1 \to \mathbb{N}$ is realized by the number 0. The successor function $s : \mathbb{N} \to \mathbb{N}$ is realized by any code for the successor function. Hence, we have a diagram

$$1 \xrightarrow{0} \mathbb{N} \xrightarrow{s} \mathbb{N},$$

and in fact this is a universal such diagram: for any morphisms $1 \xrightarrow{q} X \xrightarrow{f} X$ there is a unique $u : \mathbb{N} \to X$ such that



commutes. Since the choice $u(n) = f^n(q(*))$ is uniquely determined by the underlying sets, we only need to give a realizer for it: if r_f realizes f, and r_q is an actual realizer of $q(*)$, then $\mathbf{r}r_q(\lambda r_x n.r_f x)$ realizes u. This realizer is trivially equivariant because the equivalence relation on $P_{\mathbb{N}}$ is equality.

This property is what makes the modified assembly $\mathbb{N}$ (together with the morphisms $0, s$) a so-called *natural numbers object* in a Cartesian closed category like MAss. (For categories which are not Cartesian closed, a parametrized definition works better.) Natural number objects are essentially those objects which allow an internal notion of recursion. Of course, in a category built on a computable interpretation of arithmetic, this object is central.

2.5. Consider the set $\mathcal{G}$ of (cleverly chosen representatives of isomorphism classes of) finite graphs. A graph $G \in \mathcal{G}$ on n nodes is realized by any number whose binary expansion has n^2+1 digits, the last n^2 of which form an adjacency matrix for the graph. We let $\rho_{\mathcal{G}}(G)$ be the set of these realizers, and

$$P_{\mathcal{G}} = \bigcup_{G \in \mathcal{G}} \rho_{\mathcal{G}}(G).$$

Then $(\mathcal{G}, P_{\mathcal{G}}, =, \rho_{\mathcal{G}})$ is a good candidate for an "object of finite graphs".

Classically, a (numerical) graph invariant is any mapping $\mathcal{G} \to \mathbb{N}$. This set contains many sensible graph properties, like the number of nodes, the number of edges, the number of connected components, the treewidth, or the chromatic number. However, this set also contains ill-behaved mappings, like the one which sends an n node graph to 1 if n lies in the halting set, and to 0 otherwise.

Hence, if you are a practising constructivist – that is, a computer scientist – you might prefer to talk about numerical graph invariants in MAss, rather than in Set: the guarantee of a morphism $\mathcal{G} \to \mathbb{N}$ in MAss is precisely that it is a computable property of the graph.

2.6. Our first goal will be to prove that MAss is a Heyting category, a category in which we can interpret intuitionistic first-order logic. Following [3] book A, we prove in order that MAss is Cartesian, regular, coherent, and finally Heyting, while developing some basics about such categories along the way. This staggered approach shows which properties of MAss allow us to interpret which logical structure into the category: as we restrict ourselves to ever more specific categories, the posets $\mathrm{Sub}(X)$ for objects X of MAss slowly show their structure as Heyting algebras.

To be on our way, we first introduce some more notation.

2.7. Let $A, B \subseteq \mathbb{N}$. We write

$$\begin{aligned} A \times B &= \{\langle a, b\rangle \in \mathbb{N} \mid a \in A, b \in B\}, \\ A + B &= (\{\mathbf{k}\} \times A) \cup (\{\bar{\mathbf{k}}\} \times B), \\ A \to B &= \{r \in \mathbb{N} \mid r : A \to B\}. \end{aligned}$$

In particular, $r : A \to B$ and $r \in A \to B$ are now synonymous.

If we furthermore have equivalence relations $\sim_A, \sim_B$ on A and B respectively, we can also define these operations on $(A, \sim_A)$ and $(B, \sim_B)$. Then $(A, \sim_A) \times (B, \sim_B) = (A \times B, \sim_{A\times B})$, where $(a, b) \sim_{A\times B} (a', b') \iff a \sim_A a' \wedge b \sim_B b'$, similarly $(A, \sim_A) + (B, \sim_B) = (A + B, \sim_{A+B})$ where

$$(x, y) \sim_{A+B} (x', y') \iff (x = x' = \mathbf{k} \wedge y \sim_A y') \vee (x = x' = \bar{\mathbf{k}} \wedge y \sim_B y').$$

We set

$$(A, \sim_A) \to (B, \sim_B) = (\{r \in A \to B \mid r \text{ equivariant.}\}, \sim_{A\to B}),$$

where

$$r \sim_{A\to B} r' \iff \forall a \in A (ra \sim_B r'a).$$

Note that, unlike for $\times$ and $+$, the underlying set of $(A, \sim_A) \to (B, \sim_B)$ may actually be different from $A \to B$.

When we consider two sets $(A, \sim_A)$ and $(B, \sim_B)$ which naturally come with equivalence relations, then $A \to B$ refers to the underlying set of $(A, \sim_A) \to (B, \sim_B)$ unless specified otherwise. In particular, for a modified assembly X this is true for $(P_X, \sim_X)$.

2.8 Lemma. *The category* MAss *is Cartesian; that is, it has all finite limits.*

Proof. We encountered the terminal object before. If X, Y are modified assemblies, then their product is given by

$$(X \times Y, (P_X, \sim_X) \times (P_Y, \sim_Y), \rho_X \times \rho_Y);$$

the realizers for the projections are projections themselves. If $f, g : X \to Y$ are morphisms of modified assemblies, and $X' = \{x \in X \mid f(x) = g(x)\}$, then the equalizer of f, g is given by

$$(X', P_X, \sim_X, \rho_X|_{X'});$$

the realizer for the injection is the identity. □

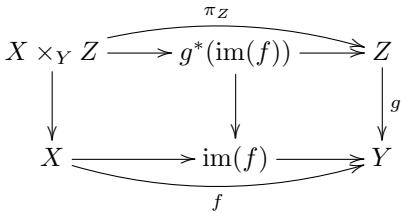
2.9 Lemma. *In a Cartesian category, the posets* $\mathrm{Sub}(A)$ *have finite meets.*

Proof. The identity $A \to A$ gives a top element. If $m_1 : A_1 \to A, m_2 : A_2 \to A$ are subobjects, then the pullback of m_1, m_2 is their meet. □

2.10 Lemma. *If $f : A \to B$ is a morphism of a Cartesian category* C*, we obtain a mapping $f^* : \mathrm{Sub}(B) \to \mathrm{Sub}(A)$, which takes a subobject $m : X \to B$ to the pullback of f and m. Furthermore, f^* preserves the finite meets in* $\mathrm{Sub}(B)$*, and thus also the ordering.*

Proof. The mapping f^* trivially preserves the top element, which is the identity. That f^* preserves binary meets is a number of applications of the pullback pasting lemma to a cube of pullbacks. □

2.11 Lemma. *The category* MAss *is regular. That is, any morphism $f : X \to Y$ has a least subobject* $\mathrm{im}(f)$ *of Y through which f factors, and these images are stable under pullback: if $g : Z \to Y$ is any morphism, then in pulling back the image factorization of f by g,*



the top row is also an image factorization.

Proof. The underlying set of $\mathrm{im}(f)$ is the set-theoretic image of f. Its potential realizers are those of X, and the actual realizers of $y \in \mathrm{im}(f)$ are given by

$$\rho_{\mathrm{im}(f)}(y) = \bigcup_{x : f(x)=y} \rho_X(x).$$

Then $X \to \mathrm{im}(f)$ is realized by the identity, and the inclusion $\mathrm{im}(f) \to Y$ is realized by any realizer for f. This is clearly the minimal subobject of Y through which f factors: if $X \to A \to Y$ is another factorization through a subobject, then the induced mapping $\mathrm{im}(f) \to A$ is realized by the same code as the mapping $X \to A$.

If we have a pullback rectangle as above, then as a set $g^*(\mathrm{im}(f))$ indeed corresponds to the image of π_Z, and the realizers of $z \in g^*(\mathrm{im}(f))$ are precisely those of $(x, z) \in X \times Z$ such that $f(x) = g(z)$, which is the same as the realizers it would get as the image of $X \times_Y Z$. □

2.12 Definition. A morphism $f : X \to Y$ in a regular category is called a *regular epimorphism* if $\mathrm{im}(f) \to Y$ is an isomorphism.

2.13. Note that in a regular category, regular epimorphisms are stable under pullback, since isomorphisms are stable under pullback.

2.14 Lemma. *Let $f : A \to B$ a morphism in a regular category. Then the mapping $\exists_f : \mathrm{Sub}(A) \to \mathrm{Sub}(B)$ which takes $m : X \to A$ to $\mathrm{im}(f \circ m) \in \mathrm{Sub}(B)$ is order-preserving, and a left adjoint to f^*.*

Proof. That $\exists_f$ is order-preserving follows from the definition of the image. To show that $\exists_f \dashv f^*$, first suppose $A_1 \leq f^*(B_1)$, where $A_1 \in \mathrm{Sub}(A), B_1 \in \mathrm{Sub}(B)$. Then $A_1 \xrightarrow{m} A \xrightarrow{f} B$ factorizes through B_1, hence so should $\mathrm{im}(f \circ m) = \exists_f(A_1) \to B$ by definition of the image. Conversely, if $\exists_f(A_1) \leq B_1$, then we have a morphism $A_1 \to B_1$, and the universal property of the pullback gives us the desired morphism $A_1 \to f^*(B_1)$. □

2.15 Proposition (Beck-Chevalley)**.** *If C is a regular category, and*

$$\begin{array}{ccc} A \times_C B & \xrightarrow{\pi_A} & A \\ \downarrow{\scriptstyle \pi_B} & & \downarrow{\scriptstyle f} \\ B & \xrightarrow{g} & C \end{array}$$

is a pullback square, then

$$\begin{array}{ccc} \mathrm{Sub}(B) & \xrightarrow{\pi_B^*} & \mathrm{Sub}(A \times_C B) \\ \downarrow{\scriptstyle \exists_g} & & \downarrow{\scriptstyle \exists_{\pi_A}} \\ \mathrm{Sub}(C) & \xrightarrow{f^*} & \mathrm{Sub}(A) \end{array}$$

commutes.

Proof. Fix $B_1 \in \mathrm{Sub}(B)$. In the diagram

$$\begin{array}{ccccc} \pi_B^*(B_1) & \longrightarrow & A \times_C B & \xrightarrow{\pi_A} & A \\ \downarrow & & \downarrow{\scriptstyle \pi_B} & & \downarrow{\scriptstyle f} \\ B_1 & \longrightarrow & B & \xrightarrow{g} & C \end{array}$$

both squares are pullbacks, hence so is the entire rectangle. Now replacing the bottom row by an image factorization, we have that the top row of

$$\begin{array}{ccccc} \pi_B^*(B_1) & \longrightarrow & f^*(\exists_g(B_1)) & \longrightarrow & A \\ \downarrow & & \downarrow & & \downarrow{\scriptstyle f} \\ B_1 & \longrightarrow & \exists_g(B_1) & \longrightarrow & C \end{array}$$

is also an image factorization, as desired. □

2.16 Lemma. *The regular category* MAss *is coherent. That is, for each object A of* MAss*, the poset* $\mathrm{Sub}(A)$ *has finite coproducts (joins), and these coproducts are stable (preserved by pullback). In particular, f^* is a morphism of bounded lattices.*

Proof. Clearly, it suffices to show that MAss has coproducts which are stable under pullback; then the initial object of MAss is the initial object of $\mathrm{Sub}(A)$, and for $X, Y \to A$ subobjects we have that

$$X \vee Y = \mathrm{im}(X \sqcup Y).$$

The stability of coproducts in $\mathrm{Sub}(A)$ then follows from the stability of image factorization.

The initial object is given by $(\emptyset, \mathbb{N}, \mathrm{id}_{\mathbb{N}}, \emptyset)$. If X, Y are modified assemblies, then their coproduct is given by

$$(X \sqcup Y, (P_X, \sim_X) + (P_Y, \sim_Y), \rho_X + \rho_Y).$$

The inclusion $X \to X \sqcup Y$ is realized by $\mathbf{pk}$, and $Y \to X \sqcup Y$ by $\mathbf{p\bar{k}}$.

Now suppose $f : A \to X \sqcup Y$ is a morphism realized by r_f. The sets $f^{-1}(X), f^{-1}(Y) \subseteq A$ partition A, and the identity $A \to f^*(X) \sqcup f^*(Y)$ is realized by

$$\lambda n.\langle (r_f n)_1, n, (r_f n)_2 \rangle.$$ □

2.17 Lemma. *The coherent category* MAss *is a Heyting category. That is, for any $f : X \to Y$, the functor $f^* : \mathrm{Sub}(Y) \to \mathrm{Sub}(X)$ has a right adjoint $\forall_f$.*

Proof. If $A \to X$ is a subobject, then $\forall_f(A)$ has as its underlying set the points $y \in Y$ for which $f^{-1}(y) \subseteq A$, and furthermore, treating $f^{-1}(y)$ as a modified assembly whose realizers agree with those in X, the inclusion $f^{-1}(y) \to A$ has a realizer. The potential realizers of $\forall_f(A)$ are given by $P_Y \times (P_X \to P_A)$. The

actual realizers of $y \in \forall_f(A)$ are pairs consisting of an element of $\rho_Y(y)$ and a realizer for $f^{-1}(y) \to A$.

Now if $B \to Y$ is a subobject, whose inclusion is realized by r_B, the equivalence

$$f^*(B) \leq A \iff B \leq \forall_f(A)$$

follows. Both inequalities come down to the existence of a function which, given a realizer for $x \in X, f(x) \in B$, can find a realizer for $x \in A$. Formally, if r realizes $f^*(B) \leq A$, then $\lambda n.\langle r_B n, \lambda m.r\langle m, n\rangle\rangle$ realizes $B \leq \forall_f(A)$, while if r realizes $B \leq \forall_f(A)$, then $\lambda\langle n, m\rangle.(rm)_2 n$ realizes $f^*(B) \leq A$. □

2.18. When we work with the logic, we will be especially interested in cases where $f = \pi : X \times Y \to Y$ is a projection, since then $\forall_\pi$ will be used to interpret universal quantification over X. In this case, we can simplify the definition somewhat. The construction from the proof above would give us, for a subobject $A \to X \times Y$, that

$$P_{\forall_\pi(A)} = P_Y \times (P_X \times P_Y \to P_A).$$

Any actual realizer must code in particular an element P_Y and a function $P_X \times P_Y \to P_A$ such that the element of P_Y is always an appropriate input for the function; hence, we can simplify without loss of generality to

$$P_{\forall_\pi(A)} = P_Y \times (P_X \to P_A).$$

Then, we restrict the actual realizers similarly.

2.19 Proposition. *If* C *is a Heyting category, and*

$$\begin{array}{ccc} A \times_C B & \xrightarrow{\pi_A} & A \\ \downarrow{\scriptstyle \pi_B} & & \downarrow{\scriptstyle f} \\ B & \xrightarrow{g} & C \end{array}$$

is a pullback square, then

$$\begin{array}{ccc} \mathrm{Sub}(B) & \xrightarrow{\pi_B^*} & \mathrm{Sub}(A \times_C B) \\ \downarrow{\scriptstyle \forall_g} & & \downarrow{\scriptstyle \forall_{\pi_A}} \\ \mathrm{Sub}(C) & \xrightarrow{f^*} & \mathrm{Sub}(A) \end{array}$$

commutes.

Proof. Take the adjoint of the conclusion of 2.15. □

2.20 Lemma. *In a Heyting category, the lattice* $\mathrm{Sub}(A)$ *is a Heyting algebra. For* $m_1 : A_1 \to A, m_2 : A_2 \to A$,

$$A_1 \to A_2 = \forall_{m_1}(A_1 \wedge A_2).$$

Furthermore, $\to$ *is preserved by pullback.*

Proof. We need to show that in $\mathrm{Sub}(A)$ we have that

$$A_3 \wedge A_1 \leq A_2 \iff A_3 \leq \forall_{m_1}(A_1 \wedge A_2).$$

But this is just the adjunction $m_1^* \dashv \forall_{m_1}$. The fact that $\to$ is preserved by pullback is just 2.19 applied to the pullback square of $m_1 : A_1 \to A$ and $f : B \to A$. □

2.21. Let us briefly take a moment to consider what the Heyting implication looks like in MAss. Consider X_1, X_2 subobjects of X, whose underlying sets are subsets. Then, considering $X_1 \wedge X_2$ as a subobject of X_2, it consists of those elements which are also in X_1, and has realizers of each subobject.

Hence, $X_1 \to X_2 = \forall_{m_1}(X_1 \wedge X_2)$ consists, as a set, of all points in $X_1 \wedge X_2$, together with all points not in X_1. Its potential realizers are given by $P_X \times (P_{X_1} \to P_{X_1} \times P_{X_2})$; for all points x not in X_1, we have

$$\rho_{X_1 \to X_2}(x) = \rho_X(x) \times (P_{X_1} \to P_{X_1} \times P_{X_2})$$

while for the points x which lie in $X_1 \cap X_2$, we have

$$\rho_{X_1 \to X_2}(x) = \rho_X(x) \times ((\rho_{X_1}(x) \to \rho_{X_1}(x) \times \rho_{X_2}(x)) \cap (P_{X_1} \to P_{X_1} \times P_{X_2}))\,.$$

Note that if the second component of a realizer of any point in $X_1 \to X_2$ is $r \mapsto \langle fr, gr\rangle$, then we can create a new realizer by replacing f with the identity. Hence, we can remove the $P_{X_1}\times$ component from all realizers, and obtain an isomorphic object. For ease of use, this is the representation of $X_1 \to X_2$ that we will use.

2.22 Corollary. *For each object A of* MAss*, the poset* $\mathrm{Sub}(A)$ *is a Heyting algebra, and for each morphism $f : A \to B$ the pullback mapping $f^* : \mathrm{Sub}(B) \to \mathrm{Sub}(A)$ is a Heyting algebra morphism. Furthermore, it admits adjoints $\exists_f \dashv f^* \dashv \forall_f$, which satisfy the Beck-Chevalley conditions in 2.15 and 2.19.*

2.23. The relevance of this is that we have now shown all the categorical structure we need to interpret an intuitionistic first-order logic into MAss. However, this is not by itself enough structure to interpret the language HA^ω into it. In HA^ω, the various types are not unrelated: they are formed from other types, and ultimately the type of natural numbers, by the operations $\times, \to$. We already have natural numbers in our category, and products of objects; types formed by the $\to$ operation should be interpreted by an internal notion of function type. That is, we want MAss to have the right exponentials. In fact, since currying is perfectly computable, it has all of them.

2.24 Lemma. *The category* MAss *is Cartesian closed.*

Proof. If X, Y are objects of MAss, then Y^X is given by

$$(\mathrm{Hom}_{\mathsf{MAss}}(X, Y), P_{Y^X}, \sim_{Y^X}, \rho_{Y^X}),$$

where

$$(P_{Y^X}, \sim_{Y^X}) = (P_X, \sim_X) \to (P_Y, \sim_Y)$$

and

$$\rho_{Y^X}(f) = \left(\bigcap_{x \in X} \rho_X(x) \to \rho_Y(f(x)) \right) \cap P_{Y^X}.$$

The evaluation mapping $X \times Y^X \to Y$ is just function evaluation, which is realized by recursive function application mapping $\mathbb{N} \times (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$. □

2.25. Recall that we made the choice to work with E−HA$^\omega$, not just HA$^\omega$. In order for that to work, we need the internal notion of a function in our categories, whose type is represented by the exponent, to be extensional as well. Fortunately, it turns out that extensionality holds internally to any Cartesian closed category. The following lemma sets the internal statement up for us.

2.26 Lemma. *Let A, B be objects of a Cartesian closed Heyting category, and consider the pullback diagram*

$$\begin{array}{ccc} S & \longrightarrow & A \\ \downarrow & & \downarrow{\scriptstyle \Delta} \\ A^B \times A^B \times B & \xrightarrow{\mathrm{ev}_{1,3}, \mathrm{ev}_{2,3}} & A \times A. \end{array}$$

Then for $\pi : A^B \times A^B \times A \to A^B \times A^B$ the projection, the subobject $\forall_\pi(S) \to A^B \times A^B$ factorizes through the diagonal $\Delta : A^B \to A^B \times A^B$.

Proof. We aim to show that the two arrows

$$\forall_\pi(S) \to A^B \times A^B \overset{\pi_1}{\underset{\pi_2}{\rightrightarrows}} A^B$$

are equal. Let us call these two mappings t_1, t_2. It clearly suffices to show that the transposes $\bar{t}_1, \bar{t}_2$ of these two mappings, given by

$$\begin{array}{ccc} \forall_\pi(S) \times B & \xrightarrow{(t_i, \mathrm{id}_B)} & A^B \times B \\ & \searrow^{\bar{t}_i} & \downarrow{\scriptstyle \mathrm{ev}} \\ & & A \end{array}$$

are equal. Now note that if we pull back $\Pi_\pi(S)$ via π we just get $\Pi_\pi(S) \times B$, with inclusion mapping given by $(t_1, t_2, \mathrm{id}_B)$. By the adjunction $\pi^* \dashv \forall_\pi$ it follows that $\forall_\pi(S) \times B$ factorizes through S. In a diagram,

$$\begin{array}{ccccc} \forall_\pi(S) \times B & & \longrightarrow & S & \longrightarrow & A \\ & \searrow^{t_1, t_2, \mathrm{id}_B} & & \downarrow & & \downarrow{\scriptstyle \Delta} \\ & & A^B \times A^B \times B & \xrightarrow{\mathrm{ev}_{1,3}, \mathrm{ev}_{2,3}} & A \times A. \end{array}$$

The bottom arrows compose to $(\bar{t}_1, \bar{t}_2)$, and the top arrows factorize through the diagonal. □

2.2 Logic in Heyting Categories

2.27. Before we continue with MAss specifically in the next section, we look at how to interpret the logic of HA^ω into a category with sufficient structure. This is standard, but spelling it out will prove convenient in the next chapter, when discussing hyperdoctrines. First, we consider more generally how to interpret intuitionistic first-order logic into a Heyting category. Then, the small extension to Cartesian closed Heyting categories with a natural numbers object allows us to interpret the finite types.

The development of the logical language, the proof calculus, and the interpretation in Heyting categories is standard; our discussion mostly follows [3], book D.

2.28. Our logical language $\mathscr{L}$ consists of a set of *sorts* or *types* $\mathrm{S}^{\mathscr{L}}$; a set of relation symbols $\mathrm{R}^{\mathscr{L}}$, with for each $R \in \mathrm{R}^{\mathscr{L}}$ an associated sequence of sorts $\mathrm{sg}(R)$ in $\mathrm{S}^{\mathscr{L}}$, called its signature; and a set of function symbols $\mathrm{F}^{\mathscr{L}}$, with for each $f \in \mathrm{F}^{\mathscr{L}}$ an associated sequence of sorts $\mathrm{sg}(f)$ in $\mathrm{S}^{\mathscr{L}}$, called its signature, and a specified sort $\mathrm{tp}(f) \in \mathrm{S}^{\mathscr{L}}$, called its type. For $R \in \mathrm{R}^{\mathscr{L}}$, we often write $R \subseteq \mathrm{sg}(R)$, and if $f \in \mathrm{F}^{\mathscr{L}}$, we often write $f : \mathrm{sg}(f) \to \mathrm{tp}(f)$, anticipating their intended interpretations. The set $\mathrm{R}^{\mathscr{L}}$ contains at least for each sort X a symbol $=_X \subseteq X \times X$.

Given a language $\mathscr{L}$, we construct the set $\mathrm{Term}(\mathscr{L})$ of $\mathscr{L}$-terms, each of which has a type as well, as the least set containing at least:

- for any sort X, variables $x_1^X, x_2^X, \ldots$ of sort X;
- for each function symbol f and terms $t_1, \ldots, t_n$ such that

$$\mathrm{sg}(f) = (\mathrm{tp}(t_1), \ldots, \mathrm{tp}(t_n))$$

  a term $f(t_1, \ldots, t_n)$ of type $\mathrm{tp}(f)$.

The set of $\mathscr{L}$-formulas $\mathrm{Form}(\mathscr{L})$ is now the least set containing

- the formulas $\bot$ and $\top$;
- for each relation symbol R and each collection of terms $t_1, \ldots, t_n$ such that $\mathrm{sg}(R) = (\mathrm{tp}(t_1), \ldots, \mathrm{tp}(t_n))$, a formula $R(t_1, \ldots, t_n)$;
- for all formulas ϕ, ψ and variables x the formulas $\phi \wedge \psi, \phi \vee \psi, \phi \to \psi, \forall x \phi, \exists x \phi$.

Furthermore, we adopt the abbreviations $\neg\phi$ for $\phi \to \bot$ and $\phi \leftrightarrow \psi$ for $(\phi \to \psi) \wedge (\psi \to \phi)$. Given any term t or formula ϕ with free variables $x_1, \ldots, x_n$, we write $\mathrm{fv}(t), \mathrm{fv}(\phi)$ respectively for the sequence of variables $\vec{x} = (x_1, \ldots, x_n)$.

Generally, a finite sequence of distinct variables $\vec{x} = (x_1, \ldots, x_n)$ is called a *context*. A context $\vec{x}$ is called *appropriate* for a term t or a formula ϕ if all free variables of t respectively ϕ appear in $\vec{x}$, but none of the bound variables do. (In particular, this means that there are no appropriate contexts for formulas

in which a variable occurs both bound and free.) If $\vec{x}$ is appropriate for t or ϕ, then the pairs $\vec{x}.t$ and $\vec{x}.\phi$ are called a *term in context* and a *formula in context* respectively.

If we use a formula ϕ which has no variables which occur both bound and free, where the ambient mathematics suggest that we should be using a formula with context, we always mean $\mathrm{fv}(\phi).\phi$, and similarly for terms. In particular, we usually omit the empty context for sentences. If there is a risk of confusion we are always explicit with the context.

2.29. In any Heyting category C we can give an interpretation of this language. This requires assigning to each sort X an object $[X]$ of C; we extend this notation to sequences of sorts, setting $[(X_1, \ldots, X_n)] = [X_1] \times \cdots \times [X_n]$. Then, we assign each relation symbol R an subobject $[R] \to [\mathrm{sg}(R)]$, and to each function symbol f a morphism $[f] : [\mathrm{sg}(f)] \to [\mathrm{tp}(f)]$. This then allows us to inductively define an interpretation for each formula in context $\vec{x}.\phi$, where we will have $[\vec{x}.\phi] \to [\mathrm{tp}(\vec{x})]$ a subobject.

In particular, a sentence (in the empty context) will have an interpretation in the Heyting algebra $\mathrm{Sub}(1)$, which we can think of as its truth value. We write $\mathsf{C} \models \phi$ if the interpretation $[\phi]$ of a sentence ϕ is equal to $\top \in \mathrm{Sub}(1)$.

2.30 Definition. A term in context $\vec{x}.t$ will be interpreted by a morphism $[\vec{x}.t] : [\mathrm{tp}(\vec{x})] \to [\mathrm{tp}(t)]$. We interpret an atomic term (that is, a variable) by a projection $[\vec{x}.x_i^{X_i}] = \pi_{[X_i]} : [\mathrm{tp}(\vec{x})] \to [X_i]$. For a more complex term in context

$$\vec{x}.t = \vec{x}.f(t_1, \ldots, t_n),$$

we set

$$[\vec{x}.t] = [f] \circ ([\vec{x}.t_1], \ldots, [\vec{x}.t_n]).$$

To interpret formulas in context $\vec{x}.\phi$, we need for each relation symbol R a subobject $[R] \to [\mathrm{sg}(R)]$. We always interpret $[=_X]$ as the diagonal $\Delta : [X] \to [X] \times [X]$. Then we can interpret $\vec{x}.R(t_1, \ldots, t_n)$ as follows:

$$[\vec{x}.R(t_1, \ldots, t_n)] = ([\vec{x}.t_1] \times \cdots \times [\vec{x}.t_n])^*([R]).$$

The two special atomic formulas in context $\vec{x}.\top, \vec{x}.\bot$ are interpreted as the top and bottom elements of $\mathrm{Sub}([\mathrm{tp}(\vec{x})])$. Similarly, the propositional connectives are defined by their counterparts in the Heyting algebra of subobjects: for example, suppose we have $\phi = \phi_0 \wedge \phi_1$. We define

$$[\vec{x}.\phi] = [\vec{x}.\phi_0] \wedge [\vec{x}.\phi_1],$$

and similarly for $\phi_0 \vee \phi_1$ and $\phi_0 \to \phi_1$.

Then, consider $\vec{x}.\phi = \vec{x}.\exists y^Y \psi$. Recall that by definition $\vec{x}$ does not contain the variable y. We have a projection $\pi : [\mathrm{tp}(\vec{x})] \times [Y] \to [\mathrm{tp}(\vec{x})]$. Then we define

$$[\vec{x}.\phi] = \exists_\pi [\vec{x}, y.\psi].$$

We do similarly for $\forall y^Y \psi$.

2.31. We now have an interpretation of logical languages in our Heyting categories. In our specific case, the category of modified assemblies, we would like this to already reflect something about modified realizability.

Indeed it does: for a formula in context $\vec{x}.\phi$ and a (sequence of) elements $\vec{a} \in [\mathrm{tp}(\vec{x})]$, we think of $P_{[\vec{x}.\phi]}$ as the set of potential realizers for the truth of $\phi(a)$ in context $\vec{x}$ – that is, numbers which encode objects of the right type to be realizers for it – and of $\rho_{[\vec{x}.\phi]}(\vec{a})$ as the actual realizers for the truth of $\phi(a)$. Then, the interpretation of $\phi = \phi_0 \wedge \phi_1$, where a realizer for ϕ is a pair of realizers, one for ϕ_0 and one for ϕ_1. This exactly mirrors the way we defined realizability for conjunctions in 1.8 and 1.13. The same holds, mutatis mutandis, for disjunction and implication.

The case of quantifiers is a little more subtle. In our strictly logical treatment of modified realizability, we defined a realizer for an existential statement to be a witness for the statement, together with a realizer for the statement at the witness. In our category of modified assemblies, the set of realizers for an existential statement is the union of realizers for the formula, where the realizers range over possible interpretations of the quantified variable. Hence, we do not encode explicitly for which value of the variable the formula is realized. Of course, we only have natural numbers as codes, so what would it mean to "give" a point from an arbitrary set?

We will come back to this point later.

2.32. In order for this logical language to have any use, it must act like actual logic, and not just an arbitrary heap of symbols. That is, we want to be able to reason with it logically: we need a type of soundness theorem of the form "if ϕ is intuitionistically provable, then ϕ holds in any Heyting category".

There is some subtlety here, which is the reason we have a treatment of formulas that includes contexts. We would like to prove a type of soundness theorem with regards to some intuitionistic logic, but it can not be completely ordinary intuitionistic first-order logic. Proof calculi for ordinary first-order logic (whether classical or constructive) are built with the assumption of interpretation in a non-empty domain. However, this will not work in Heyting categories: for instance, in MAss, we need to deal with formulas like $\forall x^{\emptyset} x \neq x$. We have that $\mathsf{MAss} \models \forall x^{\emptyset}(x \neq x)$, but in fact in most ordinary proof calculuses we would be able to prove $\exists x^{\emptyset}(x = x)$. Hence, we work with *sequents in context*, following [3].

2.33 Definition. If ϕ, ψ are $\mathscr{L}$-formulas, and $\vec{x}$ is a context appropriate for both, then the statement $\phi \vdash_{\vec{x}} \psi$ is called a *sequent*. The informal interpretation of the statement $\phi \vdash_{\vec{x}} \psi$ is that whenever we choose values for each of the variables in the context, if ϕ holds then ψ also holds.

2.34 Definition. We say that a sequent $\phi \vdash_{\vec{x}} \psi$ *holds* in a Heyting category C if

$$[\vec{x}.\phi] \leq [\vec{x}.\psi]$$

as subobjects of $[\mathrm{tp}(\vec{x})]$. In particular, $\mathsf{C} \models \phi$ means that $\top \vdash_{\mathrm{fv}(\phi)} \phi$ holds in C.

2.35. We describe a proof calculus for sequents. We give a collection of rules for when we can derive a sequent $\phi \vdash_{\vec{x}} \psi$ from some finite (possibly empty) list of sequents Γ; we denote this by

$$\frac{\Gamma}{\phi \vdash_{\vec{x}} \psi.}$$

In each of the rules, the sequents are assumed to be well-formed – in particular, each context is assumed to be appropriate for both formulas.

First, we have a set of structural rules. We have the identity axiom,

$$\frac{}{\phi \vdash_{\vec{x}} \phi;}$$

the substitution rule

$$\frac{\phi \vdash_{\vec{x}} \psi}{\phi[\vec{s}/\vec{x}] \vdash_{\vec{y}} \psi[\vec{s}/\vec{x}],}$$

where $\vec{y}$ contains at least all variables that appear in the terms in $\vec{s}$; and the cut rule

$$\frac{\phi \vdash_{\vec{x}} \psi \qquad \psi \vdash_{\vec{x}} \chi}{\phi \vdash_{\vec{x}} \chi.}$$

Next, we have the rules for the propositional connectives. For $\top, \bot$ we have

$$\frac{}{\bot \vdash_{\vec{x}} \phi,} \qquad \frac{}{\phi \vdash_{\vec{x}} \top.}$$

For conjunction we have the rules

$$\frac{}{\phi \wedge \psi \vdash_{\vec{x}} \phi,} \qquad \frac{}{\phi \wedge \psi \vdash_{\vec{x}} \psi,} \qquad \frac{\phi \vdash_{\vec{x}} \psi \qquad \phi \vdash_{\vec{x}} \chi}{\phi \vdash_{\vec{x}} \psi \wedge \chi,}$$

and dually for disjunction

$$\frac{}{\phi \vdash_{\vec{x}} \phi \vee \psi,} \qquad \frac{}{\psi \vdash_{\vec{x}} \phi \vee \psi,} \qquad \frac{\phi \vdash_{\vec{x}} \chi \qquad \psi \vdash_{\vec{x}} \chi}{\phi \vee \psi \vdash_{\vec{x}} \chi.}$$

The two rules for implication are each others' reverse:

$$\frac{\phi \wedge \psi \vdash_{\vec{x}} \chi}{\phi \vdash_{\vec{x}} \psi \to \chi,} \qquad \frac{\phi \vdash_{\vec{x}} \psi \to \chi}{\phi \wedge \psi \vdash_{\vec{x}} \chi.}$$

We have dual pairs of reversible rules for quantification: for existential quantification the rules

$$\frac{\phi \vdash_{\vec{x},y} \psi}{\exists y \phi \vdash_{\vec{x}} \psi,} \qquad \frac{\exists y \phi \vdash_{\vec{x}} \psi}{\phi \vdash_{\vec{x},y} \psi,}$$

and for universal quantification the rules

$$\frac{\phi \vdash_{\vec{x},y} \psi}{\phi \vdash_{\vec{x}} \forall y \psi,} \qquad \frac{\phi \vdash_{\vec{x}} \forall y \psi}{\phi \vdash_{\vec{x},y} \psi.}$$

Finally, we have two rules for equality:

$$\frac{}{\top \vdash_x x = x,} \qquad \frac{}{x = y \wedge \phi \vdash_z \phi[y/x].}$$

Now, if Γ is a set of sequents, we call a sequent $\phi \vdash_{\vec{x}} \psi$ an *immediate consequence* of Γ if there is a subset $\Gamma' \subseteq \Gamma$ such that

$$\frac{\Gamma'}{\phi \vdash_{\vec{x}} \psi,}$$

and we say that a sequent is a *consequence* of Γ if it belongs to the smallest superset of sequents of Γ which contains all its immediate consequences.

2.36. Note that these rules reflect "ordinary intuitionistic reasoning" – really what is different about it is the presence of contexts which may be larger than simply the free variables in the two formulas. Thus, informal constructive arguments (avoiding the law of the excluded middle and reductio ad absurdum) will typically translate into this system, and we will not be too difficult about this.

It is for this logical system that we will prove a soundness theorem for logic interpreted in Heyting categories. Note that $\phi \vdash_{x_1,\ldots,x_n} \psi$ is equivalent in our proof calculus to $\top \vdash \forall x_1 \cdots \forall x_n (\phi \to \psi)$, so we could even restrict ourselves to sequents of the form $\top \vdash \phi$ without loss of generality, and identify these with the sentence ϕ itself – but having the full notion of sequent in context available makes the proof much more palatable.

Of course, our Heyting categories are set up precisely so that $\mathrm{Sub}(A)$ is a Heyting algebra for each object A; so proving a soundness theorem for intuitionistic logic should be easy, and indeed it mostly is. There is one annoying technicality about substitution that we get out of the way in the following lemma, which states that arbitrary formulas behave similarly to atomic formulas under substitution. To illustrate this, consider a formula $x^X.P(x)$. The interpretation as a formula of $[x^X.P(x)] \in \mathrm{Sub}([X])$ is just the same as the interpretation $[P] \to [X]$ of the predicate symbol P. Now if $\vec{y}.t$ is a term in context of type $\mathrm{tp}(x)$, then the interpretation $[\vec{y}.P(t)] \in \mathrm{Sub}([\mathrm{tp}(\vec{y})])$ is equal to $[\vec{y}.t]^*([P])$ by definition. Thus, for the simplest atomic formulas, doing a substitution of a variable for a term results in taking a pullback by the interpretation of the substituting term. Now if $x.\phi(x)$ is a formula of one free variable, one would like this to act like a predicate symbol: we would hope that also $[\vec{y}.\phi(t)] = [\vec{y}.t]^*[x.\phi(x)]$. The following lemma shows this for general formulas.

2.37 Lemma. *Let $\vec{x}.\phi$ be a formula in context. Let $\vec{y}.\vec{t}$ be a sequence of terms in context (that is, the context is appropriate for each term) of the same length as $\vec{x}$ such that $\mathrm{tp}(t_i) = \mathrm{tp}(x_i)$ for each i. Then, writing $[\vec{y}.\vec{t}] = ([\vec{y}.t_1], \ldots, [\vec{y}.t_n])$,*

$$[\vec{y}.\phi(\vec{t}/\vec{x})] = [\vec{y}.\vec{t}]^*[\vec{x}.\phi].$$

Proof. We prove this by induction on the complexity of ϕ. If ϕ is an atomic formula, say $\phi = R(\vec{s})$, then, writing also $[\vec{x}.\vec{s}] = ([\vec{x}.s_1], \ldots, [\vec{x}.s_m])$, we need to prove the equality

$$([\vec{y}.\vec{s}[\vec{t}/\vec{x}]])^*[R] = ([\vec{y}.\vec{t}])^*([\vec{x}.\vec{s}])^*[R].$$

That is, it comes down to the commutativity of the diagram

$$\begin{array}{ccc} [\mathrm{tp}(\vec{y})] & \xrightarrow{[\vec{y}.\vec{t}]} & [\mathrm{tp}(\vec{x})] \\ & \searrow^{[\vec{y}.\vec{s}[\vec{t}/\vec{x}]]} & \downarrow{\scriptstyle [\vec{x}.\vec{s}]} \\ & & [\mathrm{tp}(R)], \end{array}$$

which is hopefully obvious. The cases $\phi = \bot$ and $\phi = \top$ are clear.

The induction step over propositional connectives are similar and simple; let us do the case $\phi = \phi_0 \wedge \phi_1$ for illustration. We have

$$\begin{aligned} [\vec{y}.\phi(\vec{t}/\vec{x})] &= [\vec{y}.\phi_0(\vec{t}/\vec{x})] \wedge [\vec{y}.\phi_1(\vec{t}/\vec{x})] \\ &= [\vec{y}.\vec{t}]^*[\vec{x}.\phi_0] \wedge [\vec{y}.\vec{t}]^*[\vec{x}.\phi_1] \\ &= [\vec{y}.\vec{t}]^*([\vec{x}.\phi_0] \wedge [\vec{x}.\phi_1]) \\ &= [\vec{y}.\vec{t}]^*[\vec{x}.\phi]. \end{aligned}$$

Now suppose $\phi = \exists w^W \psi$. Note first that we have a pullback square

$$\begin{array}{ccc} [\mathrm{tp}(\vec{y})] \times [W] & \xrightarrow{[\vec{y}.\vec{t}]\times \mathrm{id}_W} & [\mathrm{tp}(\vec{x})] \times [W] \\ \downarrow{\scriptstyle \pi_Y} & & \downarrow{\scriptstyle \pi_X} \\ [\mathrm{tp}(\vec{y})] & \xrightarrow{[\vec{y}.\vec{t}]} & [\mathrm{tp}(\vec{x})], \end{array}$$

so we have

$$\begin{aligned} [\vec{y}.\phi(\vec{t}/\vec{x})] &= [\vec{y}.\exists w(\psi(\vec{t}/\vec{x}))] \\ &= \exists_{\pi_Y}[\vec{y}, w.\psi(\vec{t}/\vec{x})] \\ &= \exists_{\pi_Y}([\vec{y}.\vec{t}] \times \mathrm{id}_W)^*[\vec{x}, w.\psi] \\ &= [\vec{y}.\vec{t}]^* \exists_{\pi_X}[\vec{x}, w.\psi] \\ &= [\vec{y}.\vec{t}]^*[\vec{x}.\exists w\psi], \end{aligned}$$

where we apply the Beck-Chevalley condition for the pullback square above in the fourth step. The case for $\phi = \forall w^W \psi$ is exactly similar. □

2.38 Theorem (Soundness Theorem)**.** *Let* C *be a Heyting category, and* Γ *a set of sequents. If each sequent in* Γ *holds, then so does each consequence of* Γ*. In particular, if* $\top \vdash \phi$ *is a consequence of the empty set of sequents, then* $\mathsf{C} \models \phi$*.*

Proof. We prove this, as always, by induction on the length of the derivation. Thus, we simply have to check that "truth in C" is preserved by each of our rules. This is trivial for the identity and cut rules, as they correspond to reflexivity and transitivity of the order in a Heyting algebra. For the substitution rule,

this is precisely lemma 2.37. It is also trivial for each of the propositional connectives, as these correspond precisely to defining properties of the operations in a Heyting algebra.

Then consider the rules for quantifiers. Note that extending a context, which is a special case of substitution, is done by pullback. Write $\pi_X : [\mathrm{tp}(\vec{x})] \times [\mathrm{tp}(y)] \to [\mathrm{tp}(\vec{x})]$ for the projection. We have for a formula in context $\vec{x}.\psi$ that

$$[\vec{x}, y.\psi] = \pi_X^*[\vec{x}.\psi].$$

Then, the two rules for the existential quantifier demand that

$$[\vec{x}, y.\phi] \leq \pi_X^*[\vec{x}.\psi] \iff \exists_{\pi_X}[\vec{x}.\phi] \leq [\vec{x}.\psi],$$

which is simply the adjunction $\exists_{\pi_X} \dashv \pi_X^*$. The rules for universal quantification are treated exactly the same.

The first rule for equality demands that $\Delta^*([=_X])$ is the top element of $\mathrm{Sub}([X])$, and indeed pulling a monomorphism like Δ back along itself gives the identity. For the second equality rule, write the context as $(x, y, \vec{z})$, and $X = \mathrm{tp}(x) = \mathrm{tp}(y), Z = \mathrm{tp}(\vec{z})$, so that we have to show

$$[x, y, \vec{z}.x = y] \wedge [x, y, \vec{z}.\phi] \leq [x, y, \vec{z}.\phi[y/x]] \tag{2.1}$$

as subobjects of $X \times X \times Z$. The interpretation of the very left term is an "extended diagonal" $X \times Z \to X \times X \times Z$, which is to say that the morphism representing the entire left hand side into $X \times X \times Z$ agrees on its first two coordinates. That is, the diagram

$$\begin{array}{ccc} [x, y, \vec{z}.x = y] \wedge [x, y, \vec{z}.\phi] & \longrightarrow & X \times X \times Z \\ \downarrow{\scriptstyle \pi_2} & \searrow & \downarrow{\scriptstyle (\pi_2, \pi_2, \pi_3)} \\ [x, y, \vec{z}.\phi] & \longrightarrow & X \times X \times Z \end{array}$$

commutes. But now note that, by lemma 2.37, the right hand side of equation 2.1 is equal to

$$([x, y, \vec{z}.y], [x, y, \vec{z}.y], [x, y, \vec{z}.\vec{z}])^*([x, y, \vec{z}.\phi]),$$

which is just the pullback of the right and bottom arrow in our diagram. □

2.39. In short, our interpretation of sorted first-order constructive logic into Heyting categories makes sense. As announced, we now turn our attention to the other structure necessary to interpret the language of HA^ω faithfully, in full.

From now on, we assume that our language $\mathscr{L}$ extends the language of HA^ω. That is, we now assume that there is a specified sort symbol 0. Furthermore, we assume that the set of sort symbols comes equipped with operations $\times, \to$ – and in fact that it is freely generated by these operations from some set of atomic sorts, including 0.

We then postulate that, in assigning sorts to sort symbols, we always have that

$$[X \times Y] = [X] \times [Y] \qquad \text{and} \qquad [X \to Y] = [Y]^{[X]}.$$

Furthermore, we always set $[0] = \mathbb{N}$, the natural numbers object.

We also assume, recalling from 1.2, that our set of function symbols contains the symbol $\mathrm{ev}_{\sigma,\tau}$, which gets interpreted as the evalutation mapping $[\tau]^{[\sigma]} \times [\sigma] \to [\tau]$. We similarly assume that it contains for sorts σ, τ the symbols $P_{\sigma,\tau}, P^1_{\sigma,\tau}, P^2_{\sigma,\tau}$, which are interpreted as the transposes of respectively the identity $[\sigma]\times[\tau] \to [\sigma]\times[\tau]$ and the projections $[\sigma]\times[\tau] \to [\sigma]$ and $[\sigma]\times[\tau] \to [\tau]$.

Now to interpret E$-$HA$^\omega$, we in particular need function extensionality. As we remarked in 2.25, function extensionality always holds in a Cartesian closed Heyting category C. Namely, it means that for any types A, B we have

$$\mathsf{C} \models \forall f^{A^B} \forall f'^{A^B} (\forall b^B (f(b) = f'(b)) \to f = f').$$

That is, the subobject $[\forall b^B (f(b) = f'(b))]$ of $A^B \times A^B$ needs to factorize through the subobject $[f = f']$ of $A^B \times A^B$; now the first object is precisely S from 2.26, and the second object is precisely the diagonal.

The language also contains, for sorts σ, τ, ν the symbols $I_\sigma, K_{\sigma,\tau}, S_{\sigma,\tau,\nu}$, and these are interpreted, respectively, as the (repeated) transpose of the identity $[\sigma] \to [\sigma]$, the projection $[\sigma] \times [\tau] \to [\sigma]$, and

$$\mathrm{ev}_{[\tau],[\nu]} \circ (\mathrm{ev}_{[\sigma],[\tau\to\nu]}, \mathrm{ev}_{[\sigma],[\tau]}) \circ \alpha,$$

where α is "the" mapping

$$[\sigma \to \tau \to \nu] \times [\sigma \to \tau] \times [\sigma] \to [\sigma \to \tau \to \nu] \times [\sigma] \times [\sigma \to \tau] \times [\sigma].$$

It seems that we are encoding a lot of redundancy: why do we need multiple ways to interpret projections, identities, and evaluations? The reason is that we are being precise with syntax, but doing category theory "informally" up to isomorphism. For instance, we casually identify

$$[\sigma]^{[\tau]\times[\nu]} \cong \left([\sigma]^{[\tau]}\right)^{[\nu]}.$$

This becomes a clear issue only when we want to actually prove that this is an interpretation of the type system of HA$^\omega$, that is, to verify that all the axioms from 1.2 actually hold in C. This is an exercise we gratefully leave to the reader.

Finally, our language has a symbol R_σ. Intepreting this will take some more work. Consider the morphism

$$(\mathrm{ev}_{[\sigma\times 0],[\sigma]}, \pi_2, s \circ \pi_3) : [\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N} \to [\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N},$$

where $s : \mathbb{N} \to \mathbb{N}$ is the successor, and write

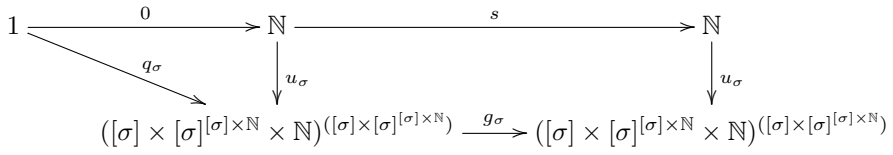
$$g_\sigma : \mathbb{N} \to ([\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N})^{([\sigma]\times[\sigma]^{[\sigma]\times\mathbb{N}})}$$

for its transpose. Then, write

$$q_\sigma : 1 \to ([\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N})^{([\sigma]\times[\sigma]^{[\sigma]\times\mathbb{N}})}$$

for the transpose of the morphism $(\pi_1, \pi_2, 0) : X \times X^{X\times\mathbb{N}} \times 1 \to X \times X^{X\times\mathbb{N}} \times \mathbb{N}$.

The universal property of the natural numbers object gives us a morphism u_σ such that



commutes. The transpose of u_σ is a morphism

$$\hat{u}_\sigma : [\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N} \times [\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N},$$

and finally the interpretation of R_σ is the repeated transpose of the composition of this morphism with the projection $[\sigma] \times [\sigma]^{[\sigma]\times\mathbb{N}} \times \mathbb{N} \to [\sigma]$. Again, we leave checking that this interpretation of R_σ satisfies the axioms for R_σ to the reader if she so wishes.

2.3 The Mathematics of Modified Assemblies

2.40. With our logic for MAss set up, we want to investigate whether or not it indeed reflects modified realizability. First, note that the finite types in MAss are the finite types of HEO: the natural numbers object (from 2.4), the construction of the product (from 2.8) and the construction of the exponent (from 2.24) mirror precisely the definitions from 1.5.

A natural first class of formulas to consider is that of those in the language of HA^ω, with types limited to the finite types, as these are the formulas for which we initially defined realizability. Fortunately, we indeed find that the formulas which hold in MAss are precisely the realized ones.

2.41 Theorem. *Let ϕ be a sentence in the language of HA^ω. Then*

$$\mathsf{MAss} \models \phi \iff \textit{there is an } r \in \tau(\phi) \textit{ such that } r \ \mathbf{mr}\ \phi.$$

Here, $\tau(\phi)$ refers to the HEO-realization of the type $\tau(\phi)$.

Proof. In order to allow a proof by induction, we will prove something stronger. Let $\vec{x}.\phi(\vec{x}, \vec{a})$ be a formula in context in the language of HA^ω extended with finitely many constants $\vec{a}$, each of which is a name for an element of a finite type in HEO, their intended interpretation. Let $\vec{b}$ be a sequence of elements of HEO with the same type as $\vec{x}$. Then we claim that the sets

$$\rho_{[\vec{x}.\phi(\vec{x},\vec{a})]}(\vec{b}) \quad \text{and} \quad \{r \in \tau(\phi[\vec{b}/\vec{x}, \vec{a}]) \mid r \ \mathbf{mr}\ \phi[\vec{b}/\vec{x}, \vec{a}]\} \tag{2.2}$$

are computably equivalent. That is, there is a single function, which given a syntactical encoding of ϕ, and (codes for) the elements of $\vec{a}, \vec{b}$, computes from elements from the left set codes for elements from the right, and a function which does the same from right to left; in such a way that the function is equivariant with respect to the equivalence relation on $P_{[\vec{x}.\phi(\vec{x},\vec{a})]}$ and equality on the right set. Note that this requires canonical choices for the objects $[\vec{x}.\phi(\vec{x},\vec{a})]$, which are really only given up to isomorphism: we assume the choice is given by the constructions in the first section of this chapter.

We prove this, of course, by induction on the complexity of ϕ – we construct the functions as we go along. Thus, suppose first that ϕ is an atomic formula. In other words, disregarding the trivial cases $\phi \in \{\top, \bot\}$, an equation of natural numbers. In this case, our function computes the numbers on either side of the equation. If they are distinct, both sets are empty, and there is nothing to compute. If the numbers agree, then the right hand side is $\mathbb{N}$; the left hand side consists of (codes for) the vector $\vec{b}$ together with the number to which each side of the equation evaluates. The functions we use are clear.

If $\phi = \phi_0 \wedge \phi_1$, then both sets in 2.2 are products of the corresponding sets for ϕ_0, ϕ_1. If $\phi = \phi_0 \vee \phi_1$, then both sets are disjoint unions, coded in almost the same way.

If $\phi = \phi_0 \to \phi_1$, then the right set of 2.2 is equal to

$$\{r \in \tau(\phi[\vec{b}/\vec{x},\vec{a}]) \mid \forall k(k \ \mathbf{mr} \ \phi_0[\vec{b}/\vec{x},\vec{a}] \to rk \ \mathbf{mr} \ \phi_1[\vec{b}/\vec{x},\vec{a}]\}$$

while the left set is given by

$$\rho_{[\mathrm{tp}(\vec{x})]}(\vec{b}) \times (\rho_{[\vec{x}.\phi_0(\vec{x},\vec{a})]}(\vec{b}) \to \rho_{[\vec{x}.\phi_1(\vec{x},\vec{a})]}(\vec{b})).$$

The difference comes down to the factor $\rho_{[\mathrm{tp}(\vec{x})]}(\vec{b})$, but this is actually just a sequence of codes for $\vec{b}$, which the function also takes as input.

If $\phi = \exists y^\sigma \phi_0(y,\vec{x},\vec{a})$, then the right set of 2.2 equals

$$\{r \in \tau(\phi[\vec{b}/\vec{x},\vec{a}]) \mid r_2 \ \mathbf{mr} \ \phi_0[r_1,\vec{x},\vec{a}]\},$$

while the left set equals

$$\bigcup_{c\in\sigma} \rho_{[\vec{x}.\phi_0(c/y,\vec{x},\vec{a})]}(\vec{b}); \tag{2.3}$$

that is, in the first case the element c is explicitly given, while in the second case it is not. We remarked this earlier in 2.31. However, this is no problem: a code for c can be found from the realizer for the inclusion $[y,\vec{x}.\phi_0(y,\vec{x},\vec{a})] \to [\sigma \times \mathrm{tp}(x)]$, which in turn can be computed using just the structure of ϕ, since a realizer from 2.3 is the same as one from $[y,\vec{x}.\phi_0(y,\vec{x},\vec{a})]$.

Finally, suppose $\phi = \forall y^\sigma(\phi_0(y,\vec{x},\vec{a}))$. Then the left set of 2.2 equals

$$\rho_{[\mathrm{tp}(\vec{x})]}(\vec{b}) \times \bigcap_{c\in\sigma} \left(\rho_{[\sigma]}(c) \to \rho_{[\vec{x}.\phi_0(y,\vec{x},\vec{a})]}(c,\vec{b})\right),$$

while the right set equals

$$\{r : \sigma \to \tau(\phi_0(y,\vec{x},\vec{a})) \mid c \in \sigma \implies rc \ \mathbf{mr} \ \phi_0(c/y,\vec{x},\vec{a})\};$$

again, the difference is basically just the factor $\rho_{[\mathrm{tp}(\vec{x})]}(\vec{b})$, which is an input to the algorithm. □

2.42. In particular, this theorem tells us that the forms of the axiom of choice and independence of premise, as they appeared in 1.11, hold in our category MAss. This was a first goal in constructing the category. However, it turns out some generalization is possible: in fact the principle of independence of premise is a logical principle that holds for many more types than just the finite types; and while the axiom of choice in our category is limited to finite types, the parameter a in

$$\forall a^{\alpha}\forall x^{\sigma}\exists y^{\tau}(P(x,y,a)) \to \exists f^{\sigma\to\tau}\forall x^{\sigma}(P(x,f(x),a))$$

can take any type in MAss. Furthermore, it will be interesting to see *why* the axiom of choice holds only for finite types, a fact into which the general theorem does not directly give us insight.

2.43 Theorem. *Let X be any type, and let Y be a type such that for any $n \in P_Y$, there is a $y \in Y$ such that $n \in \rho_Y(y)$. Let P, R predicates on $X, X \times Y$ respectively. Then*

$$\mathsf{MAss} \models \forall x^X((\neg P(x) \to \exists y^Y R(x,y)) \to \exists y(\neg P(x) \to R(x,y))).$$

Proof. For convenience, abbreviate the sentence by $\forall x^X(\phi(x) \to \psi(x))$. We aim to give a section from X to its subobject $[\phi(x) \to \psi(x)]$. That is, we construct an equivariant function which, given an $n \in \rho_X(\bar{x})$, produces a function which, given an actual realizer of $\phi(x)$, gives an actual realizer of $\psi(x)$; and furthermore this function also maps $P_X \to (P_{[\phi(x)]} \to P_{[\psi(x)]})$. First, let r_y be a realizer for the mapping $R \to X \times Y \xrightarrow{\pi} Y$.

We claim that the function $\lambda nrk.\langle n, r_y(r0), r0\rangle$ does the trick; note that the function is trivially equivariant. First, suppose that $n \in P_X, r \in P_{[\phi(x)]}$. We need to show that $\lambda k.\langle n, r_y(r0), r0\rangle \in P_{[\psi(x)]}$. By definition,

$$P_{[\psi(x)]} = P_{[\neg P(x)\to R(x,y)]} = P_{X\times Y} \times P_{[R(x,y)]}^{P_{[\neg P(x)]}}.$$

Now, by assumption $n \in P_X$. Furthermore,

$$r \in P_{[\phi(x)]} = P_{[\exists y(R(x,y))]}^{P_{[\neg P(x)]}},$$

hence $r0 \in P_{[\exists y(R(x,y))]}$ – but this is the same set as $P_{[R(x,y)]}$. In particular then, $r_y(r0) \in P_Y$.

It remains to be shown that our realizer also works on actual realizers. To this end, suppose that $r \in \rho_{[\phi(x)]}(\bar{x})$ for any $\bar{x} \in X$; we have to show that $\lambda k.\langle n, r_y(r0), r0\rangle \in \rho_{[\psi(x)]}(\bar{x})$. Now, by definition, since Y is inhabited,

$$\rho_{[\psi(x)]}(\bar{x}) = \bigcup_{\bar{y}\in Y} \rho_{[\neg P(x)\to R(x,y)]}(\bar{x},\bar{y}).$$

We consider two cases: either $\bar{x} \in [\neg P(x)]$, or $\bar{x} \notin [\neg P(x)]$. In the first case, we in particular have $0 \in \rho_{[x,y.\neg P(x)]}(\bar{x}, \bar{y})$, and thus

$$r0 \in \rho_{[\exists y(R(x,y))]}(\bar{x}) = \bigcup_{\bar{y} \in Y} \rho_{[R(x,y)]}(\bar{x}, \bar{y});$$

in particular, also $r_y(r0) \in \rho_Y(\bar{y})$ for that same $\bar{y}$, as we wanted. If $\bar{x} \notin [\neg P(x)]$, then also for any $\bar{y}$, we have $(\bar{x}, \bar{y}) \notin \rho_{[x,y.\neg P(x)]}(\bar{x}, \bar{y})$, so that

$$\rho_{[\psi(x)]}(\bar{x}) = \bigcup_{\bar{y} \in Y} \rho_{[\neg P(x) \to R(x,y)]}(\bar{x}, \bar{y}) = \bigcup_{\bar{y} \in Y} \rho_X(\bar{x}) \times \rho_Y(\bar{y}) \times P_{[R(x,y)]}^{P_{[\neg P(x)]}};$$

and now $\lambda k.\langle n, r_y(r0), r0\rangle$ lies in this set for the $\bar{y}$ for which $r_y(r0)$ is an actual realizer, which exists by assumption on Y. □

2.44. Note, in particular, that the proof above makes no *use* of equivariance anywhere. We could apply the exact same argument to the category GrAss we defined in 2.3. Thus, independence of premise holds regardless of whether we require extensionality.

However, independence of premise does not hold in Asm. To see this, let $P(x^0)$ be the statement "the machine coded by x does not halt", which we can formulate in the language of HA^ω, and let $R(x, y)$ be the statement "the machine coded by x halts within y steps". Then the antecedent holds in Asm: the realizer for $\neg P(x) \to \exists y^0 R(x, y)$, given a realizer of $P(x)$, just runs the machine given by x until it halts, producing the value y. However, the consequent does not hold: producing for any x a value y such that *if* x halts, then it halts within y steps obviously solves the halting problem.

Modified realizability – that is, the logic of MAss and GrAss – does not suffer from this problem, since a realizer for the antecedent must always terminate on potential realizers, which at least includes 0. This is also visible in the proof, where the realizer is given, essentially, by evaluation at 0.

2.45. To approach this contrast from the other side, we consider another plausible constructive arithemtical statement: Markov's principle. Formally, it is given by

$$\forall x(\forall y^0(R(x, y) \vee \neg R(x, y)) \wedge \neg\forall y^0(\neg R(x, y)) \to \exists y^0(R(x, y)).$$

That is – fixing the parameter x – if $R(x, y)$ is a decidable formula, and it is not true that $R(x, y)$ is always false, then we can find a y so that it is true. The computable interpretation of this is that with the antecedents, we can start an unbounded search on the natural numbers, checking for each of them y whether $R(x, y)$ holds; and this process must terminate by the second assumption.

We can turn this into a realizer for Markov's principle in Asm: it is precisely a function which searches for the first y such that $R(x, y)$ holds. However, by much the same argument as above, we cannot do that within MAss (or GrAss);

let R be the same relation as in 2.44. Clearly R is decidable, so let us focus on the reduced formula

$$\forall y(\neg\forall x(\neg R(x,y)) \to \exists x(R(x,y))).$$

If this statement had an actual realizer in MAss (or GrAss), then in particular that actual realizer would have to be a potential realizer; thus, given the code 0, it would have to terminate. Now if $\neg\forall x(\neg R(x,y))$ were realized, then 0 would in particular be an actual realizer. From this actual realizer we could then extract the realizer for an instance of x such that $R(x,y)$ holds. In other words, this would give us a computable function which, given a natural y number, gives us an upper bound on the number of steps the machine coded by y takes to halt. This would solve the halting problem.

2.46 Theorem. *For all finite types σ,τ, all sorts α, and all interpretations of R,*

$$\mathsf{MAss} \models \forall a^\alpha(\forall x^\sigma \exists y^\tau(R(x,y,a)) \to \exists f^{\sigma\to\tau}\forall x^\sigma(R(x,f(x),a))).$$

Proof. We have to give a section from the object $[\alpha]$ to its subobject

$$[\forall x^\sigma \exists y^\tau(R(x,y,a)) \to \exists f^{\sigma\to\tau}\forall x^\sigma(R(x,f(x),a))], \tag{2.4}$$

so we give a realizer for this statement uniformly in the element $\bar{a} \in A$. Before we do this, let us consider the realizers of $[R(x,f(x),a)]$. That object is the pullback of $[R]$ by the "evaluation"

$$[\sigma]\times[\sigma\to\tau]\times[\alpha] \to [\sigma]\times[\tau]\times[\alpha].$$

Hence, we could take a realizer of a point $(\bar{x},\bar{f},\bar{a}) \in [R(x,f(x),a)]$ to be sequence of realizers for $\bar{x},\bar{f},\bar{a} \in [\sigma]\times[\sigma\to\tau]\times[\alpha]$ and one for $(\bar{x},\bar{f}(\bar{x}),\bar{a}) \in [R] \subseteq [\sigma]\times[\tau]\times[\alpha]$. Of course, using the realizer for $[R] \to [\sigma]\times[\tau]\times[\alpha]$ we can compute the actual realizers for $\bar{x},\bar{a} \in [\sigma]\times[\alpha]$ from those for $(\bar{x},\bar{f}(\bar{x}),\bar{a}) \in R$; thus, we decide that $P_{[R(x,f(x),a]} = P_{[\sigma\to\tau]} \times P_{[R]}$, and

$$\rho_{[R(x,f(x),a)]}(\bar{x},\bar{f},\bar{a}) = \rho_{[\sigma\to\tau]}(\bar{f}) \times \rho_{[R]}(\bar{x},\bar{f}(\bar{x}),\bar{a}).$$

Now let us move on to the proof.

The monomorphism $[R] \to [\sigma]\times[\tau]\times[\alpha]$ composes with a projection to an arrow $[R] \to [\tau]$, a realizer for which we call r_y. Then we claim an actual realizer for 2.4 is given by

$$\lambda\langle r_a, r_R\rangle.\langle r_y \circ r_R, r_a, \lambda r_x.\langle r_y \circ r_R, r_R r_x\rangle\rangle.$$

To prove this, first note that the potential realizers of the antecedent, using the convention from 2.18, are given by

$$P_{[\forall x\exists y R(x,y,a)]} = P_{[\alpha]} \times (P_{[\sigma]} \to P_{\exists y R(x,y,a))} = P_{[\alpha]} \times (P_{[\sigma]} \to P_{[R]}),$$

so that an element of this is indeed a pair. Thus, suppose that $\langle r_a, r_R\rangle$ is a potential realizer of the antecedent.

We have $r_R : P_{[\sigma]} \to P_{[R]}$. Composing this with the realizer $r_y : P_{[R]} \to P_{[\tau]}$ gives a code for an *equivariant* function $r_y \circ r_R : P_{[\sigma]} \to P_{[\tau]}$. In particular, it is a realizer for a function $\hat{f} \in \sigma \to \tau$. Now we have

$$P_{[\exists f \forall x R(x,f(x),a)]} = P_{[\forall x R(x,f(x),a)]} = P_{[\sigma\to\tau]\times[\alpha]} \times (P_{[\sigma]} \to (P_{\sigma\to\tau} \times P_{[R]})),$$

and indeed $r_y \circ r_R \in P_{[\sigma\to\tau]}$, $r_a \in P_{[\alpha]}$ and

$$\lambda r_x.\langle r_y \circ r_R, r_R r_x\rangle \in P_{[\sigma]} \to (P_{[\sigma\to\tau]} \times P_{[R]}).$$

Now suppose furthermore that $\langle r_a, r_R\rangle$ is an actual realizer at the point $\bar{a}$ of the antecedent. Recall that

$$\rho_{[\exists f \forall x R(x,f(x),a)]}(\bar{a}) = \bigcup_{\bar{f}\in\sigma\to\tau} \rho_{[\forall x R(x,f(x),a)]}(\bar{a},\bar{f}),$$

and we claim that

$$\langle r_y \circ r_R, r_a, \lambda r_x.\langle r_y \circ r_R, r_R r_x\rangle\rangle \in \rho_{[\forall x R(x,f(x),a)]}(\hat{f},\bar{a}).$$

An element of this set of actual realizers consists of a pair of actual realizers for $\bar{f}, a$ – which the first two elements of our triple are – together with a realizer for a function which, when given an actual realizer of an $x \in X$, gives an element of

$$\rho_{[R(x,f(x),a)]}(\bar{x},\hat{f},\bar{a}) = \rho_{[\sigma\to\tau]}(\hat{f}) \times \rho_{[R]}(\bar{x},\hat{f}(\bar{x}),\bar{a}),$$

and the third element of our triple does precisely this. □

2.47. Although it may not appear immediately obvious, the usage of equivariance in the proof was crucial. By equivariance, the function $P_{[\sigma]} \to P_{[\tau]}$ that we found was in fact a code for an element $\sigma \to \tau$, which became the witness of the existential statement. This means that the proof does not go through for the analogous category GrAss. In fact, the analogous theorem for GrAss is false.

Recall that GrAss has the same construction as MAss without the equivalence relations, and thus without the equivariance condition on realizers. Since the equivariance of realizers *of finite types* is still forced by the fact that the equivalence classes of realizers are precisely the actual realizers of a single function, the finite types in GrAss have the same underlying sets, and the same sets of realizers, but without equivalence relations on them. Let $R(x^1, y^0)$ get interpreted in MAss as the subobject of $[0] \times [1]$ whose underlying set is

$$\{(x, y) \in 1 \times 0 \mid y \text{ is a code for the function } x\},$$

where the realizers for (x, y) are just a realizer for x and one for y. Then

$$\mathsf{GrAss} \models \forall x^1 \exists y^0 (R(x, y));$$

namely, an actual realizer for this statement is a function which takes a realizer of x (that is, a code for x) to itself. However,

$$\mathsf{GrAss} \not\models \exists f^2 \forall x^1 (R(x, f(x))),$$

since a witness f for this statement would be an actual computable function $1 \to 0$, which assigns codes to computable functions; but this would imply decidability of function equality for codes of total functions.

Thus, the axiom of choice for finite types provides the contrast between the logic of MAss and that of GrAss, where they are otherwise very similar.

In fact, we can carry out the above argument within
$MAss$ as well: if we equip two objects $0', 1'$ with the same underlying sets and sets of realizers as $[0], [1]$, but give the set of potential realizers the total equivalence relation, then the same argument works for these objects in MAss. Of course, these objects are not finite types in MAss; but it shows that the axiom of choice does not hold for *all* objects of MAss, which is why we limit ourselves to finite types in the axiom of choice.

2.48. Now that we know that the axiom of choice does not hold for all types, we are interested in investigating to what types beyond the finite types we can extend it. If we focus on the "domain type" in the axiom of choice, which is $[\sigma]$ in the version above, we arrive at a type of object well-studied in category theory: the projective objects.

2.49 Definition. An object P of a regular category C is called *internally projective* if the functor $(-)^P : \mathsf{C} \to \mathsf{C}$ preserves regular epimorphisms. The object P is called *externally projective* if $\mathrm{Hom}(P, -) : \mathsf{C} \to \mathsf{Set}$ preserves regular epimorphisms.

2.50 Lemma. *In* MAss *(or any category where the object 1 is externally projective) all internally projective objects are externally projective.*

Proof. Note that 1 is indeed externally projective in MAss: if $f : 1 \to A$ is any morphism, and $B \to A$ a regular epimorphism, then we can define a realized mapping $1 \to B$ sending the unique element $*$ of 1 to any element over $f(*)$.

Now, suppose that P is internally projective, and let $e : A \to B$ be an epimorphism. Then $\mathrm{Hom}(P, e)$ is epic if and only if $\mathrm{Hom}(1, e^P)$ is epic, which by internal projectiveness of P and external projectiveness of 1 it is. □

2.51. The converse also holds in MAss, but for this it is easier to first characterize externally projective objects in MAss.

2.52 Lemma. *In* MAss*, the following are equivalent for an object* P*:*

1. *P is externally projective;*
2. *any regular epimorphism onto P splits;*
3. *P is isomorphic to an object P' such that for every $p \in P'$, $\rho_{P'}(p)$ is a singleton.*

Proof. (1) $\implies$ (2): Suppose $e : X \to P$ is a regular epimorphism. Then, $\mathrm{Hom}(P, e) : \mathrm{Hom}(P, X) \to \mathrm{Hom}(P, P)$ is a regular epimorphism, so let s be a point in the preimage of id_P. Then s is a section of e.

(2) $\implies$ (1): Suppose $e : B \to A$ is a regular epimorphism, and consider $f \in \mathrm{Hom}(P, A)$. Then we have a pullback

$$\begin{array}{ccc} P \times_A B & \longrightarrow & B \\ \downarrow & & \downarrow e \\ P & \xrightarrow{f} & A \end{array}$$

where the right arrow is a regular epimorphism because e is. Then we obtain a section $P \to P \times_A B$, and by composition a lift $P \to B$. This arrow is mapped by $\mathrm{Hom}(P, e)$ to f, so $\mathrm{Hom}(P, e)$ is surjective.

(2) $\implies$ (3) Define a new modified assembly $\hat{P}$ by

$$\hat{P} = (P \times P_P, P_P, \sim_P, (p, r) \mapsto \{r\}).$$

That is, $\hat{P}$ contains for every realizer r of every point p a copy of that point. Now, let $\hat{P} \to P$ be the projection, which is realized by the identity. This is clearly a regular epimorphism; its image is exactly P. Hence, this morphism has a section $f : P \to \hat{P}$. But now P is isomorphic to the set-theoretic image of f with the realizers of $\hat{P}$, which are all singleton sets.

(3) $\implies$ (2) Suppose that $e : X \to P$ is a regular epimorphism. This means that there is an isomorphism $P \to \mathrm{im}(e)$. A realizer r_s for this isomorphism computes, for each $p \in P$ for the unique element of $\rho_P(p)$, a realizer of some $x \in X$ such that $e(x) = p$. Now choose for each $p \in P$ any such x, and assemble these into a function $s : P \to X$ which is realized by r_s. By construction, s is a section for e. □

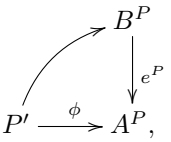
2.53. The construction in the proof of (2) $\implies$ (3) can be done for any object, not just projective objects. This tells us that every object admits a regular epimorphism from a projective objects. This is usually summarized as saying that MAss "has enough projectives". Furthermore, note that the product of two projective objects is still projective: by definition the actual realizer of an element of a product is a pair of actual realizers for each of its coordinates.

2.54 Lemma. *If an object P of MAss is externally projective, then it is internally projective.*

Proof. Let $e : B \to A$ be a regular epimorphism, and let $\phi : P' \to A^P$ be a regular epimorphism from a projective object. Then the object $P' \times P$ is also a projective object, so composition $P' \times P \to A^P \times P \xrightarrow{\mathrm{ev}} A$ admits a lift

$$\begin{array}{ccccc} & & & & B \\ & & & & \downarrow e \\ P' \times P & \xrightarrow{\phi \times \mathrm{id}_P} & A^P \times P & \xrightarrow{\mathrm{ev}} & A. \end{array}$$

Transposing this diagram,



gives a factorization of e^P through ϕ. But ϕ is a regular epimorphism by assumption, hence so must e^P be. □

2.55. Thus, when it comes to MAss, we drop the words "externally" and "internally" and speak only of projective objects. With our characterizations, we are now able to see how these projective objects relate to the axiom of choice. First, we give the obvious logical characterization of regular epimorphisms.

2.56 Lemma. *Let C be a Heyting category. A morphism $e : A \to B$ of C is a regular epimorphism if and only if*

$$\mathsf{C} \models \forall b^B \exists a^A (e(a) = b).$$

Proof. Note that the formula comes down to the existence of a section $B \to [\exists a(e(a) = b)]$. We have a pullback diagram of an image factorization

$$\begin{array}{ccccc} [e(a) = b] & \longrightarrow & \Delta^*(\mathrm{im}(e) \times B) & \longrightarrow & [=_B] \\ \downarrow & & \downarrow & & \downarrow \Delta \\ A \times B & \longrightarrow & \mathrm{im}(e) \times B & \longrightarrow & B \times B; \end{array}$$

the top row is the same mapping as $[e(a) = b] \to A \times B \xrightarrow{\pi} B$, so that the top middle object is equal to $[\exists a(e(a) = b)]$. Hence, a section $B \to \mathrm{im}(e)$ is equivalent to a section $B \to [\exists a(e(a) = b]$. □

2.57 Proposition. *Let P be an object of MAss. Then P is projective if and only if for all objects X of MAss, the statements $\mathrm{AC}_{P,X}$ hold. That is, for all objects A of MAss and all subobjects $R \to P \times X \times A$,*

$$\mathsf{MAss} \models \forall a^A (\forall p^P \exists x^X (R(p, x, a)) \to \exists f^{P \to X} \forall p^P (R(p, f(p), a))).$$

Proof. Suppose first that $\mathrm{AC}_{P,X}$ holds, and suppose that $f : D \to E$ is a regular epimorphism. Then consider the instance of $\mathrm{AC}_{P,X}$

$$\forall g^{P \to D} (\forall p^P \exists e^E (f(g(p)) = e) \to \exists s^{P \to E} \forall p^P (f(g(p)) = s(p))));$$

the antecedent holds since f is a regular epimorphism, and

$$\forall p^P (f(g(p)) = s) \iff f \circ g = s$$

in the internal logic, which means we obtain

$$\forall g^{P \to D} \exists s^{P \to E} (f \circ g = s),$$

which states that f^P is a regular epimorphism.

Now suppose that P is projective, so we can without loss of generality assume that its sets of actual realizers are singletons. Then we claim that our proof of the axiom of choice for finite types, 2.46, goes through for $\mathrm{AC}_{P,X}$ with very small changes. The only difference is the way we justify the existence of what is there called $\hat{f}$: now, if r_R is an actual realizer, the function coded by $r_y \circ r_R$ must map for each $p \in P$ its unique actual realizer to a realizer of some point of X; and choosing any one of these points as a function image gives us a function $\hat{f}$ realized by $r_y \circ r_R$; and this function forms a point in X^P. □

2.58. Note that each modified assembly X admits a monomorphism (a realized injective function) into a projective object: equipping the underlying set of X with the potential realizers $P_{X_0} = \{0\}$ and $\rho_{X_0}(x) = \{0\}$ has the identity $X \to X_0$ trivially realized by 0. It turns out that this, in combination with the existence of enough projectives, is sufficient to reconstruct the category MAss from the full subcategory on the projective objects; this is known as the reg/lex completion. See for instance [1] for details.

2.59. Given that projective objects are a good angle from which to consider the axiom of choice, and that we are especially interested in the axiom of choice for finite types, it becomes a natural question to wonder whether or not there is a nice subcategory of MAss in which the finite types are projective. Note that the finite types (apart from 0, and powers of 0) are not projective in MAss, as by the characterization in 2.52, this would mean that we could compute from the set of realizers for a computable function a single canonical code – that would mean computing equality of computable functions.

2.60 Definition. A modified assembly X is called *modest* if, for all $x, x' \in X$, if $n \in \rho_X(x), n' \in \rho_X(x')$, and $n \sim n'$, then $x = x'$.

2.61. Note that the full subcategory of MAss of modest modified assemblies is still a regular category: the image of a morphism from a modest assembly is naturally modest as well, as are the terminal object and the pullback of two modest assemblies. Thus, it makes sense to consider the notion of externally projective in this category. Furthermore, the finite types are all modest: hence, we can ask whether or not they are (externally) projective in this category.

2.62 Proposition. *A modest modified assembly X is externally projective (in the category of modest modified assemblies) if and only if for each $x \in X$, if $n, n' \in \rho_X(x)$ then $n \sim n'$.*

Proof. Suppose that X, Y are modest modified assemblies, that Y has the described property, and that $g : X \to Y$ is a regular epimorphism; or, equivalently, that $\mathrm{im}(g) \cong Y$. This means that we have a realizer $r_f : P_Y \to P_X$, such that for each $y \in Y$ and $n \in \rho_Y(y)$, there is an $x \in X$ such that $g(x) = y$ and $rn \in \rho_X(x)$. Since for pair $n, n' \in \rho_Y(y)$ we must have $n \sim n'$, and r is equivariant, it follows that $rn \sim rn'$, and thus that the x we found above is unique.

Now let $f : Y \to X$ be the function which assigns to each y this unique x; it is trivially realized by r_f.

Conversely, suppose that Y is projective, and consider the modest modified assembly

$$\hat{Y} = (Y \times P_Y / \sim_Y, P_Y, \sim_Y, (y, r) \mapsto r).$$

Then the projection $\hat{Y} \to Y$ is realized by the identity, and a section $Y \to \hat{Y}$ makes Y isomorphic to its set-theoretic image in $\hat{Y}$ with the realizers of $\hat{Y}$. $\square$

Chapter 3

Realizability Triposes and Toposes

3.1 Hyperdoctrines

3.1. Before we look at the main subject of this chapter, which is triposes, we generalize the notion of Heyting category to that of a hyperdoctrine. Where the predicates of an object in a Heyting category are restricted to being the *categorical* subobjects, a hyperdoctrine allows us to consider other notions of predicate. Triposes are then hyperdoctrines with an extra property that allows us to construct a topos from them.

3.2 Definition. Let C be a finitely complete category, and $\mathsf{P} : \mathsf{C}^{\mathrm{op}} \to \mathsf{Heyt}$, a functor from C to the category of Heyting algebras. We call P a *hyperdoctrine* if for each $f : X \to Y$ of C, the morphism $\mathsf{P}(f) : \mathsf{P}(Y) \to \mathsf{P}(X)$ has both a left adjoint $\exists_f : \mathsf{P}(X) \to \mathsf{P}(Y)$ and a right adjoint $\forall_f : \mathsf{P}(X) \to \mathsf{P}(Y)$ (which are order-preserving but not necessarily Heyting algebra morphisms) and furthermore, these morphisms satisfy the *Beck-Chevalley* condition: if

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ \downarrow{\scriptstyle g} & & \downarrow{\scriptstyle h} \\ Z & \xrightarrow{k} & W \end{array}$$

is a pullback square in C, then the squares

$$\begin{array}{ccc} \mathsf{P}(Z) & \xrightarrow{\mathsf{P}(g)} & \mathsf{P}(X) \\ \downarrow{\scriptstyle \exists_k} & & \downarrow{\scriptstyle \exists_f} \\ \mathsf{P}(W) & \xrightarrow{\mathsf{P}(h)} & \mathsf{P}(Y) \end{array} \qquad \begin{array}{ccc} \mathsf{P}(Z) & \xrightarrow{\mathsf{P}(g)} & \mathsf{P}(X) \\ \downarrow{\scriptstyle \forall_k} & & \downarrow{\scriptstyle \forall_f} \\ \mathsf{P}(W) & \xrightarrow{\mathsf{P}(h)} & \mathsf{P}(Y) \end{array}$$

both (or equivalently: either) commute.

3.3. The motivating example of a hyperdoctrine is the case where C is a Heyting category, and $\mathrm{Sub} : \mathsf{C} \to \mathsf{Heyt}$ is the subobject functor, which sends an object X of C to its poset of subobjects, and a morphism $f : X \to Y$ to the pullback functor $f^* : \mathrm{Sub}(Y) \to \mathrm{Sub}(X)$. Recall that we proved the Beck-Chevalley condition in 2.15 and 2.19.

Since we generally want to think of the elements of $\mathsf{P}(X)$ as generalized subobjects of X, we will continue to write write f^* for $\mathsf{P}(f)$, even when P is a different functor.

Before we move on to other examples for which this generalization was made, note that almost all of our discussion in section 2.2 about Heyting categories carries over to a category equipped with a hyperdoctrine. Substituting "element of $\mathsf{P}(X)$" for "subobject of X" allows everything from the introduction of the logic to the proof of the soundness theorem to go through unchanged, with one exception: equality. Where we interpreted equality in a Heyting category as the diagonal subobject $X \to X \times X$, it might not be immediately obvious what the "equality element" in $\mathsf{P}(X \times X)$ should be. We will come back to this after some examples.

3.4. Hyperdoctrines are often defined instead as functors to $\mathsf{HeytPre}$, the category of Heyting pre-algebras, which are preordered sets whose poset reflections are Heyting algebras (or, equivalently, small thin Cartesian closed finitely cocomplete categories). Many of the important examples – such as 3.10 – naturally give rise to these Heyting prealgebras, rather than Heyting algebras.

In order to work with these examples seamlessly, we will not make too much of a problem of working with Heyting prealgebras rather than algebras, and checking e.g. the commutativity of the Beck-Chevalley square only up to isomorphism rather than strictly.

3.5. Let H be any complete Heyting algebra. For a set X we obtain a new Heyting algebra H^X by ordering pointwise (or, equivalently, applying the Heyting algebra operations pointwise). For a function $f : X \to Y$ we set $f^*(\psi) = \psi \circ f$. Because the operations on H^X, H^Y are given pointwise, f^* is a Heyting algebra morphism.

For $f : X \to Y$ we define

$$\exists_f(\phi)(y) = \bigvee_{x : f(x) = y} \phi(x)$$

and

$$\forall_f(\phi)(y) = \bigwedge_{x : f(x) = y} \phi(x).$$

Note that $\exists_f, \forall_f$ are order-preserving, and recall that we did not require these to be Heyting algebra morphisms. The requirement that $\exists_f \dashv f^*$ states that that for all $\phi \in H^X, \psi \in H^Y, x \in X, y \in Y$,

$$\phi(x) \leq \bigvee_{x' : f(x') = f(x)} \phi(x') \qquad \text{and} \qquad \bigvee_{x : f(x) = y} \psi(f(x)) \leq \psi(y),$$

which clearly hold. The case for $f^* \dashv \forall_f$ is similar.

Now consider a pullback square in Set,

$$\begin{array}{ccc} X \times_Z Y & \xrightarrow{\pi_X} & X \\ \downarrow{\scriptstyle \pi_Y} & & \downarrow{\scriptstyle f} \\ Y & \xrightarrow{g} & Z. \end{array}$$

Then the Beck-Chevalley condition is that

$$\begin{array}{ccc} H^Y & \xrightarrow{\pi_Y^*} & H^{X \times_Z Y} \\ \downarrow{\scriptstyle \forall_g} & & \downarrow{\scriptstyle \forall_{\pi_X}} \\ H^Z & \xrightarrow{f^*} & H^X \end{array}$$

commutes. Fixing $\psi \in H^Y$ and $x \in X$, we have that

$$\begin{aligned} \forall_{\pi_X}(\pi_Y^*(\psi))(x) &= \bigwedge_{\substack{(x,y) \in X \times_Z Y: \\ \pi_X(x,y) = x}} \psi(\pi_Y^*(x,y)) \\ &= \bigwedge_{y : g(y) = f(x)} \psi(y) \\ &= f^*(\forall_g(\psi))(x). \end{aligned}$$

We conclude that $H^- : \mathsf{Set}^{\mathrm{op}} \to \mathsf{Heyt}$ is an example of a hyperdoctrine.

3.6. Before we move on to the more crucial examples, it will help to have some more notation available for operations on sets of natural numbers. Recall the notation in 2.7.

3.7. Let $A = (A_a, A_p, \sim_A)$ and $B = (B_a, B_p, \sim_B)$ pairs of sets of natural numbers with equivalence relations. That is, $A_a \subseteq A_p \subseteq \mathbb{N}$, and $\sim_A$ is an equivalence relation on A_p, and similarly for B. We have

$$\begin{aligned} A \times B &= (A_a \times B_a, (A_p, \sim_A) \times (B_p, \sim_B)), \\ A + B &= (A_a + B_a, (A_p, \sim_A) + (B_p, \sim_B)), \\ A \to B &= ((A_a \to B_a) \cap ((A_p, \sim_A) \to (B_p, \sim_B)), (A_p, \sim_A) \to (B_p, \sim_B)). \end{aligned}$$

Like before, if $r \in A \to B$, we also write this as $r : A \to B$.

3.8. If we have a collection of sets with equivalence relations $\{(A_i, \sim)\}_{i \in I}$, then

$$A = \bigcap_{i \in I} A_i$$

naturally comes with the equivalence relation which is the intersection of the equivalence relations on each of the A_i; it is the greatest relation such that all inclusions $A \to A_i$ are equivariant. Similarly,

$$A' = \bigcup_{i \in I} A_i$$

naturally comes with the equivalence generated by the union of the equivalence relations on each of the A_i; it is the least equivalence relation such that each inclusion $A_i \to A'$ is equivariant. In particular, if the A_i all lie in some ambient set (say $\mathbb{N}$), then the empty union and intersection take on a natural meaning as $(\emptyset, \emptyset)$ and $(\mathbb{N}, \mathbb{N} \times \mathbb{N})$ respectively.

Thus, we can take unions and intersections of collections of sets of naturals with equivalence relations, and by extension of pairs like $(A_a, A_p, \sim)$.

3.9. Now follow the most important examples of hyperdoctrines – and in fact they will all turn out to be triposes. These can all rightfully be called *realizability triposes*, and one in particular will be our modified realizability tripos, the one from which we will construct the modified realizability topos. In the following example and lemmas, Σ refers to one of the following three sets:

$$\Sigma = \mathcal{P}(\mathbb{N});$$
$$\Sigma = \{(A_a, A_p) \mid A_a \subseteq A_p \subseteq \mathbb{N}, 0 \in A_p\};$$
$$\Sigma = \{(A_a, A_p, \sim) \mid A_a \subseteq A_p \subseteq \mathbb{N}, 0 \in A_p, \sim \text{ equivalence relation on } A_p\}.$$

The definitions and proofs work almost exactly the same for all three definitions.

3.10 Definition. For any set X, we define a preorder on Σ^X by reduction: that is, for ϕ, ψ we define

$$\phi \leq \psi \iff \exists r \in \mathbb{N}\, \forall x \in X (r : \phi(x) \to \psi(x)).$$

(We also abbreviate $\forall x (r : \phi(x) \to \psi(x))$ as $r : \phi \to \psi$, and call r a realizer for $\phi \leq \psi$.) We claim that Σ^X is a Heyting prealgebra. For $\phi, \psi \in \Sigma^X$ we set, recalling the definitions in 2.7 and 3.7,

$$(\phi \wedge \psi)(x) = \phi(x) \times \psi(x)$$
$$(\phi \vee \psi)(x) = \phi(x) + \psi(x)$$
$$(\phi \to \psi)(x) = \phi(x) \to \psi(x).$$

Note that what exactly this notation means depends on which of the three sets Σ is.

3.11 Lemma. *For any set X, the pair $(\Sigma^X, \leq)$ is a Heyting prealgebra, and Σ^f is a morphism of Heyting prealgebras.*

Proof. The reflexivity of the preorder on Σ^X is given by the identity realizer $\mathbf{i}$, and its transitivity is given by composition of realizers.

We define $\bot \in \Sigma^X$ by

$$\bot(x) = \emptyset,$$
$$\bot(x) = (\emptyset, \{0\}),$$
$$\bot(x) = (\emptyset, \{0\}, =),$$

depending on which set Σ refers to. Similarly, we define $\top \in \Sigma^X$ by

$$\begin{aligned}\top(x) &= \mathbb{N},\\ \top(x) &= (\mathbb{N},\mathbb{N}),\\ \top(x) &= (\mathbb{N},\mathbb{N},\mathbb{N}\times\mathbb{N}),\end{aligned}$$

Clearly, for any $\chi \in \Sigma^X$, we have $0 : \bot \to \chi$ and $0 : \chi \to \top$, so these are indeed top and bottom elements of Σ^X. For convenience, also use $\top, \bot \in \Sigma$ refer to the unique value these functions take.

Fix $\phi, \psi \in \Sigma^X$. We define

$$(\phi \wedge \psi)(x) = \phi(x) \times \psi(x),$$

so that $\mathbf{p}_1 : \phi\wedge\psi \to \phi$ and $\mathbf{p}_2 : \phi\wedge\psi \to \psi$. If for any $\chi \in \Sigma^X$ we have $r_1 : \chi \to \phi$ and $r_2 : \chi \to \psi$, then $\lambda n.((r_1 n), (r_2 n)) : \chi \to \phi \wedge \psi$.

We define

$$(\phi \vee \psi)(x) = \phi(x) + \psi(x),$$

so $\lambda n.(\mathbf{k}, n) : \phi \to \phi \vee \psi$, and $\lambda n.(\bar{\mathbf{k}}, n) : \psi \to \phi \vee \psi$. If $\chi \in \Sigma^X$ is such that $r_1 : \phi \to \chi$ and $r_2 : \psi \to \chi$ then

$$\lambda n : (\mathbf{p}_1 n)(r_1(\mathbf{p}_2 n))(r_2(\mathbf{p}_2 n)) : \phi \vee \psi \to \chi.$$

Finally, we define

$$(\phi \to \psi)(x) = \phi(x) \to \psi(x).$$

Now, for any χ, if $r : \chi \wedge \phi \to \psi$, then $\lambda nm.r\langle n, m\rangle : \chi \to (\phi \to \psi)$, while if $r : \chi \to (\phi \to \psi)$ then $\lambda n.r(\mathbf{p}_1 n)(\mathbf{p}_2 n) : \chi \wedge \phi \to \psi$.

In the case that Σ consists of pairs with equivalence relations, verifying that each of these realizers is equivariant is nothing but unfolding the definitions, so we have shown that Σ^X is a Heyting prealgebra. Since our operations $0, 1, \wedge, \vee, \to$ are defined pointwise, if $f : X \to Y$ is a function, then $f^* : \Sigma^Y \to \Sigma^X$ preserves these operations and is thus a morphism of Heyting prealgebras. □

3.12 Lemma. *If $f : X \to Y$ is a mapping of sets, then $f^* : \Sigma^Y \to \Sigma^X$ taken as a mapping of posets has a left and right adjoint,*

$$\exists_f \dashv f^* \dashv \forall_f.$$

Proof. We define $\forall_f : \Sigma^X \to \Sigma^Y$ by

$$\forall_f(\phi)(y) = \bigcap_{x : f(x) = y} \top \to \phi(x).$$

(In particular, if y is not in the image of f, then $\forall_f(\phi)(y) = \top$.) Note that $\forall_f(\phi)(y)$ is indeed an element of Σ^Y.

We see that $\forall_f$ is order-preserving: if $r : \phi \to \psi$ is a realizer, then

$$\lambda nk.r(nk) : \forall_f(\phi) \to \forall_f(\psi)$$

is a realizer. In particular, in case Σ is the set of pairs with an equivalence relation, and if $n \sim n'$ as elements of $\forall_f(\phi)_p$, then for each $k \in \mathbb{N}$ we have that $nk \sim n'k$, and thus $r(nk) \sim r(n'k)$. Hence $\lambda k.r(nk) \sim \lambda k.r(n'k)$, so our realizer is equivariant.

Now that we have established that it is a monotone map, let us move on to the adjunction relations $\mathrm{id}_{\Sigma^Y} \leq \forall_f \circ f^*$ and $f^* \circ \forall_f \leq \mathrm{id}_{\Sigma^X}$. If $\psi \in \Sigma^Y$ and $y \in Y$, then

$$(\forall_f \circ f^*)(\psi)(y) = \bigcap_{x:f(x)=y} \top \to \psi(f(x)),$$

so a realizer $r : \psi \to (\forall_f \circ f^*)(\psi)$ is given by $r = \lambda n.\mathbf{k}n$.

Similarly, for $\phi \in \Sigma^X$ and $x \in X$ we have

$$(f^* \circ \forall_f)(\phi)(x) = \bigcap_{x':f(x')=f(x)} \top \to \phi(x')$$

so that in fact $\lambda n.n0 : (f^* \circ \forall_f)(\phi) \to \phi$.

Let $s \in \Sigma$ refer to $\{\mathbf{i}\}, (\{\mathbf{i}\}, \{0, \mathbf{i}\})$, or $(\{\mathbf{i}\}, \{0, \mathbf{i}\}, =)$, depending on the set denoted by the letter Σ. We define $\exists_f$ by

$$\exists_f(\phi)(y) = \begin{cases} \bigcup_{x:f(x)=y} s \times \phi(x) & \text{if } y \in \mathrm{im}(f), \\ \bot & \text{else.} \end{cases}$$

If $r : \phi \to \psi$, then

$$\lambda\langle m, n\rangle.m(r(n_2)) : \exists_f(\phi) \to \exists_f(\psi),$$

so that $\exists_f$ is monotone. The two adjunction relations for $\exists_f$ are $\mathrm{id}_{\Sigma^X} \leq f^* \circ \exists_f$ and $\exists_f \circ f^* \leq \mathrm{id}_{\Sigma^Y}$. Fixing $\phi \in \Sigma^X$ and $x \in X$, we see that

$$(f^* \circ \exists_f)(\phi)(x) = \bigcup_{x':f(x')=f(x)} s \times \phi(x'),$$

so that $\lambda n.(\mathbf{i}, n) : \phi \to (f^* \circ \exists_f)(\phi)$. Then, fixing $\psi \in \Sigma^Y$ and $y \in Y$, we have

$$(\exists_f \circ f^*)(\psi)(y) = \begin{cases} \bigcup_{x:f(x)=y} s \times \psi(y) & \text{if } y \in \mathrm{im}(f), \\ (\emptyset, \{0\}) & \text{else,} \end{cases}$$

so that $\mathbf{p}_2 : (\exists_f \circ f^*)(\psi) \to \psi$. □

3.13. Later on, we will almost exclusively care about $\exists_f$ and $\forall_f$ when f is a surjection. In that case, it is clear that we have isomorphisms

$$\forall_f(\phi)(y) = \bigcap_{x:f(x)=y} \phi(x),$$
$$\exists_f(\phi)(y) = \bigcup_{x:f(x)=y} \phi(x).$$

The symmetry in this definition is more apparent, and it is somewhat less cumbersome to work with.

3.14 Proposition. *The functor* $\mathsf{P} : \mathsf{Set} \to \mathsf{Heyt}$ *which extends the assignment* $X \mapsto \Sigma^X$ *and* $f \mapsto f^*$ *is a hyperdoctrine.*

Proof. In light of lemmas 3.11 and 3.12, we only need to show that the adjoints satisfy the Beck-Chevalley condition.

Consider a pullback square

$$\begin{array}{ccc} X \times_Z Y & \xrightarrow{\pi_X} & X \\ \downarrow{\scriptstyle \pi_Y} & & \downarrow{\scriptstyle f} \\ Y & \xrightarrow{g} & Z \end{array}$$

in Set. Then the Beck-Chevalley condition for left adjoints states that

$$\begin{array}{ccc} \mathsf{P}(Y) & \xrightarrow{\pi_Y^*} & \mathsf{P}(X \times_Z Y) \\ \downarrow{\scriptstyle \exists_g} & & \downarrow{\scriptstyle \exists_{\pi_X}} \\ \mathsf{P}(Z) & \xrightarrow{f^*} & \mathsf{P}(X) \end{array}$$

commutes. Now, for $\psi \in \mathsf{P}(Y)$ and $x \in X$ we have that both $(f^* \circ \exists_g)(\psi)(x)$ and $(\exists_{\pi_X} \circ \pi_Y^*)(\psi)(x)$ are equal to

$$\begin{cases} \bigcup_{y : g(y) = f(x)} s \times \psi(y) & \text{if } f(x) \in \mathrm{im}(g), \\ \bot & \text{else,} \end{cases}$$

so that this holds even strictly. □

3.15. Thus, each of our definitions of Σ corresponds to a hyperdoctrine. In fact, as well shall see shortly, all three are triposes. The most important case is

$$\Sigma = \{(A_a, A_p, \sim) \mid A_a \subseteq A_p \subseteq \mathbb{N}, 0 \in A_p, \sim \text{ equivalence relation on } A_p\}.$$

This is the tripos which corresponds to modified realizability, and will lead us to the modified realizability topos, the central object of study in this thesis. Accordingly, we call this tripos M, the *modified realizability tripos.*

The definition

$$\Sigma = \{(A_a, A_p) \mid A_a \subseteq A_p \subseteq \mathbb{N}, 0 \in A_p\}$$

gives rise to the tripos we call G, the *Grayson tripos* or *non-equivariant modified realizability tripos* – which we give this name as it gives rise to what we call Gray, the Grayson topos, which is the modified realizability topos as it exists in the literature. Its behaviour is very similar to that of M.

Finally, we call the hyperdoctrine corresponding to

$$\Sigma = \mathcal{P}(\mathbb{N})$$

E, the *effective tripos*. We call it this because it will give rise to the famous effective topos.

3.16. Recall that the only thing in the way of translating the proof of the soundness theorem for Heyting algebras to hyperdoctrines outright is the treatment of equality. The definition of equality does not even (immediately) make sense, since it is the only relation symbol we define as a *specific* subobject, rather than a generic element of $\mathrm{Sub}(X)$.

Writing $\Delta : X \to X \times X$ for the diagonal in a Heyting category, a way of writing Δ in the language of hyperdoctrines is

$$\exists_\Delta(\top_{\mathrm{Sub}(X)}).$$

This is redundant for a Heyting category, but it is a notation that makes sense in a general hyperdoctrine: $\top \in \mathsf{P}(X)$ always exists, and we always have the mapping $\exists_\Delta : \mathsf{P}(X) \to \mathsf{P}(X \times X)$. Thus, we take this as our interpretation of equality. To transfer the soundness theorem to hyperdoctrines, we still have to show that this definition of equality works to show that the rules for equality from 2.35 continue to work in a hyperdoctrine. First, a small lemma.

3.17 Lemma. *If $f : X \to Y$ is a morphism in C, and $\psi \in \mathsf{P}(X), \phi \in \mathsf{P}(Y)$, then*

$$\exists_f(\psi \wedge f^*(\phi)) = \exists_i(\psi) \wedge \phi.$$

Proof. By the adjunction $\exists_f \dashv f^*$, we have $\psi \leq f^*\exists_f(\psi)$. Hence,

$$\psi \wedge f^*(\phi) \leq f^*\exists_f(\psi) \wedge f^*(\phi) = f^*(\exists_f\psi \wedge \phi),$$

so that, using the adjunction again,

$$\exists_f(\psi \wedge f(\phi)) \leq \exists_f f^*(\exists_f\psi \wedge \phi) \leq \exists_f\psi \wedge \phi.$$

For the other inequality, note that the adjunction gives us that

$$\psi \wedge f^*(\phi) \leq f^*\exists_f(\psi \wedge f^*(\phi)).$$

Now we can use the adjunction between $\wedge$ and $\to$, to see that this is equivalent to

$$\psi \leq f^*(\phi \to \exists_f(\psi \wedge f^*(\phi))).$$

Using, again, the adjunction $\exists_f \dashv f^*$, and the adjunction between $\wedge$ and $\to$, we obtain

$$\exists_f(\psi) \wedge \phi \leq \exists_f(\psi \wedge f^*(\phi)),$$

which completes the equality. □

3.18 Proposition. *For any category* C *with tripos* P*,*

$$[x.x = x] = \top \in \mathsf{P}(X)$$

and

$$[\vec{z}.x = y \wedge \phi] \leq [\vec{z}.\phi[y/x]] \in \mathsf{P}(Z).$$

Proof. The first equation says that

$$\top = \Delta^* \exists_\Delta \top,$$

which follows from the adjunction $\exists_\Delta \dashv \Delta^*$. The second equation states that

$$(\pi_1, \pi_2)^*(\exists_\Delta \top) \wedge [x, y, \vec{z}.\phi] \leq (\pi_2, \pi_2, \pi_3)^*([x, y, \vec{z}.\phi]), \tag{3.1}$$

where π_1, π_2, π_3 are the projections of $X \times X \times Z$. Applying the Beck-Chevalley condition to the square

$$\begin{array}{ccc} X \times Z & \xrightarrow{\bar{\Delta}} & X \times X \times Z \\ \downarrow & & \downarrow{\scriptstyle (\pi_1, \pi_2)} \\ X & \xrightarrow{\Delta} & X \times X \end{array}$$

shows that

$$(\pi_1, \pi_2)^*(\exists_\Delta \top) = \exists_{\bar{\Delta}}(\top_{\mathsf{P}(X \times Z)})$$

Now, by 3.17, we have

$$\begin{aligned} (\pi_1, \pi_2)^*(\exists_\Delta \top) \wedge [x, y, \vec{z}.\phi] &= \exists_{\bar{\Delta}}(\top_{\mathsf{P}(X \times Z)}) \wedge [x, y, \vec{z}.\phi] \\ &= \exists_{\bar{\Delta}}(\bar{\Delta}^*[x, y, \vec{z}.\phi]) \end{aligned}$$

Hence, by the adjunction $\exists_{\bar{\Delta}} \dashv \bar{\Delta}^*$, we can restate 3.1 as

$$\bar{\Delta}^*[x, y, \vec{z}.\phi] \leq \bar{\Delta}^*(\pi_2, \pi_2, \pi_3)^*[x, y, \vec{z}.\phi],$$

and

$$\bar{\Delta}^*(\pi_2, \pi_2, \pi_3)^* = ((\pi_2, \pi_2, \pi_3) \circ \bar{\Delta})^* = \bar{\Delta}^*. \qquad \square$$

3.19. With this, we have our soundness theorem for hyperdoctrines. The two clauses of proposition 3.18 are precisely the two facts we need to show that the rules for equality in our proof calculus preserve truth in the logic of a hyperdoctrine. If P is a C-hyperdoctrine, and Γ a set of sequents, then if each element of Γ holds in P, so does each consequence of P.

This logical language is the most fundamental part of the hyperdoctrine: from now on we will use it constantly. The informal interpretation of an element $R \in \mathsf{P}(X \times X)$ is that it represents a binary relation on the object X. If we want to express, for example, that this relation is transitive, we can write

$$\mathsf{P} \models \forall x^X \forall y^X \forall z^X (R(x, y) \wedge R(y, z) \to R(x, z)).$$

We no longer have to express this much less legibly by giving projections $\pi_{i,j} : X \times X \times X \to X \times X$ onto the ith and jth factor, and then writing that

$$\pi_{1,2}^* R \wedge \pi_{2,3}^* R \leq \pi_{1,3}^* R$$

holds in $\mathsf{P}(X \times X \times X)$. Similarly, suppose we want to define a predicate P on $\mathsf{P}(X)$ in terms of, say, a relation $R \in \mathsf{P}(X \times Y)$ and a function (morphism) $f : X \to Y$. Writing

$$P(x) \iff R(x, f(x))$$

is now a formal definition of P, instead of having to write

$$P = (\mathrm{id}, f)^* R$$

to make it precise.

3.2 The Category of Partial Equivalence Relations

3.20. From any hyperdoctrine, we can build a so-called *category of partial equivalence relations*, or PERs. This category will in fact be a Heyting category, and its internal logic generalizes the logic of the hyperdoctrine: it adds quotient objects to the objects available in the tripos.

3.21 Definition. Let P be a hyperdoctrine on a category C. We define a category $\mathsf{C}[\mathsf{P}]$ of partial equivalence relations over P as follows. The objects of $\mathsf{C}[\mathsf{P}]$ are pairs $(X, \approx)$ where $\approx \in \mathsf{P}(X \times X)$, such that

$$\mathsf{P} \models \forall x^X \forall y^X (x \approx y \to y \approx x),$$
$$\mathsf{P} \models \forall x^X \forall y^X \forall z^X (x \approx y \wedge y \approx z \to x \approx z).$$

(We use the same notation for the logical symbol $\approx$ and its interpretation. In fact, from now on we will do that often, to various elements of sets $\mathsf{P}(X)$.) The relation $\approx$ should be thought of as a kind of equality. If $x \approx y$, then everything that holds for x must also hold for y. In the setting of realizability this means: there is a uniform way of computing, from realizers for $\phi(x)$, realizers for $\phi(y)$.

A morphism $F : (X, \approx) \to (Y, \approx)$ is an element $F \in \mathsf{P}(X \times Y)$ which is a *functional relation*. That is, F satisfies the following four conditions:

$$\mathsf{P} \models \forall x^X \forall y^Y (F(x, y) \to x \approx x \wedge y \approx y),$$
$$\mathsf{P} \models \forall x^X \forall x'^X \forall y^Y \forall y'^Y (F(x, y) \wedge x \approx x' \wedge y \approx y' \to F(x', y')),$$
$$\mathsf{P} \models \forall x^X \forall y^Y \forall y'^Y (F(x, y) \wedge F(x, y') \to y \approx y'),$$
$$\mathsf{P} \models \forall x^X (x \approx x \to \exists y^Y (F(x, y))).$$

In turn, these requirements express that F is strict, relational, single-valued and total. If $F : (X, \approx) \to (Y, \approx)$ and $G : (Y, \approx) \to (Z, \approx)$ are morphisms, then we define

$$G \circ F = [\exists y (F(x, y) \wedge G(y, z))].$$

Showing that this is again a functional relation is easy using the logic in the tripos. We can put the comments in 3.19 into practice. For instance, to show the single-valuedness of $G \circ F$ we reason as follows. Suppose we have $GF(x,z)$ and $GF(x,z')$. Then there are y, y' such that $F(x,y), G(y,z), F(x,y'), G(y',z')$. From the single-valuedness of F it follows that $y \approx y'$, so that from the relationality of G it follows that $G(y,z')$, and then from the single-valuedness of G it follows that $z \approx z'$.

Associativity of composition comes down to the equivalence

$$\exists z(\exists y(F(x,y) \wedge G(y,z)) \wedge H(z,w)) \iff \exists y(F(x,y) \wedge \exists z(G(y,z) \wedge H(z,w))).$$

The identity morphism on $(X, \approx)$ is just the relation $\approx \in \mathsf{P}(X \times X)$.

By abuse of notation we refer to an object $(X, \approx)$ of $\mathsf{C}[\mathsf{P}]$ as just X. If it adds to clarity, we may refer to $\approx$ as $\approx_X$.

3.22. Let us once more use the logical language of the tripos to show that the functional relations in $\mathsf{P}(X \times Y)$ form an antichain. That is, if $F, G \in \mathsf{P}(X \times Y)$ are functional relations, and $F \leq G$, then $F = G$. Or, in other words, if

$$\mathsf{P} \models \forall x^X \forall y^Y (F(x,y) \to G(x,y)),$$

then $F = G$. The proof proceeds the way one would prove for ordinary functions $f, g : X \to Y$ that if $f \subseteq g$ then $f = g$. Namely, fix arbitrary x, y such that $G(x,y)$. Then by strictness we have $x \approx x$, and thus by totality of F we have that there is some y' such that $F(x,y')$. By our assumption, it follows that $G(x,y')$ holds, so that by single-valuedness of G, we obtain $y \approx y'$. Finally, from the relationality of F we then obtain $F(x,y)$ as desired.

Note, again, that it does not a priori even make sense to say that $G(x,y)$ "holds", especially if the underlying category C is not concrete. But thanks to the soundness theorem, because the above argument can be rendered in our proof calculus, the proof is valid.

3.23 Lemma. $\mathsf{C}[\mathsf{P}]$ *has finite limits.*

Proof. First we construct the terminal object in $\mathsf{C}[\mathsf{P}]$. Let 1 be a terminal object in C, and consider the object $(1, \top_{1\times 1})$. For any object X of $\mathsf{C}[\mathsf{P}]$ define $F(x,*) \in \mathsf{P}(X \times I)$ to be $[x \approx x]$. Then, F is a morphism $(X, \approx) \to (1, \top)$. For instance, to see that F is total, consider the unique morphism $f : X \to 1$ in C. Then

$$\mathsf{P} \models \forall x^X (x \approx x \to F(x, f(x))),$$

from which totality follows.

Next, let us construct the pullback. Suppose that we have functional relations $F \in \mathsf{P}(X \times Z), G \in \mathsf{P}(Y \times Z)$. We define the pullback as $(X \times Y, \approx)$ where $\approx \in \mathsf{P}(X \times Y \times X \times Y)$ is given by

$$(x,y) \approx (x',y') \iff x \approx x' \wedge y \approx y' \wedge \exists z(F(x,z) \wedge G(y,z)).$$

Showing that this is a partial equivalence relation is straightforward using the logic of the tripos, using the relationality of F and G for the symmetry.

We give a projection $\pi_X : X \times_Z Y \to X$ by

$$\pi_X((x,y),x') \iff x \approx x' \wedge \exists z(F(x,z) \wedge G(y,z)).$$

Showing that this is a functional relation is, again, straightforward. Note that we can prove $y \approx y$ (as part of the strictness of π_X) by using the strictness of G. The projection $\pi_Y \in \mathsf{P}(X \times Y \times Y)$ is similar.

To see that the pullback diagram commutes, we can prove the implication

$$\begin{aligned}&\exists x'^X(x \approx x' \wedge \exists z'(F(x,z') \wedge G(y,z')) \wedge F(x',z)) \\ &\quad\implies \exists y'^Y(y \approx y' \wedge \exists z'(F(x,z') \wedge G(y,z')) \wedge G(y',z)),\end{aligned}$$

which is obvious by taking $y' = y$.

Finally, let us see that this morphism is universal. Suppose that we have $A \in \mathsf{P}(W \times X), B \in \mathsf{P}(W \times Y)$ functional relations, such that $F \circ A = G \circ B$. Then, we define $(A,B) \in \mathsf{P}(W \times X \times Y)$ by

$$(A,B)(w,(x,y)) \iff A(w,x) \wedge B(w,y).$$

Note that indeed $\pi_X \circ (A,B) = A$, for trivially

$$\exists(x',y')(A(w,x') \wedge B(w,y') \wedge x' \approx x \wedge \exists z(F(x,z) \wedge G(y,z))) \implies A(w,x),$$

and similarly $\pi_Y \circ (A,B) = B$. Finally, it is clear why (A,B) is unique with this property; if H also has the property that

$$A(w,x) \implies \exists(x',y')(H(w,(x',y')) \wedge \pi_X((x',y'),x),$$

and similarly for B, then $(A,B)(w,(x,y)) \implies H(w,(x,y))$ quickly follows.

Since $\mathsf{C}[\mathsf{P}]$ has pullbacks and a terminal object, it has all finite limits. □

3.24. Since we are interested in interpreting logic in $\mathsf{C}[\mathsf{P}]$, we should characterize subobjects.

3.25 Lemma. *Let $F \in \mathsf{P}(X \times Y)$ be a functional relation. Then $F : X \to Y$ is a monomorphism if and only if*

$$\mathsf{P} \models \forall x^X x'^X y^Y (F(x,y) \wedge F(x',y) \to x \approx x').$$

Proof. Suppose first that the formula above holds for F, and that $G, H : Z \to X$ are such that $F \circ G = F \circ H$ – that is, such that

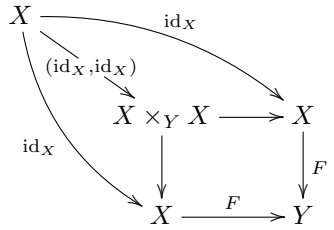
$$\mathsf{P} \models \forall z^Z \forall y^Y (\exists x^X (G(z,x) \wedge F(x,y)) \to \exists x'^X (H(z,x) \wedge F(x',y))).$$

Then, we can conclude that $x \approx x'$, and thus that

$$\mathsf{P} \models \forall z^Z \exists x^X (G(z,x) \wedge H(z,x)),$$

from which it quickly follows that the two are equal.

Conversely, suppose $F : X \to Y$ is a monomorphism. It follows that in the diagram



the outer square is a pullback, so that the diagonal mapping $X \to X \times_Y X$ is an isomorphism. Note that if F is an isomorphism with inverse F^{-1}, then trivially we have

$$F(x, y) \iff F^{-1}(y, x).$$

Thus, the totality of this inverse gives us that

$$\mathsf{P} \models \forall (x, x')^{X \times_Y X}((x, x') \approx (x, x') \implies \exists x''^{X}(x'' \approx x \wedge x'' \approx x')).$$

Since by definition,

$$(x, x') \approx (x, x') \iff x \approx x \wedge x' \approx x' \wedge \exists y(F(x, y) \wedge F(x', y)),$$

clearly $F(x, y) \wedge F(x', y) \implies (x, x') \approx (x, x')$, and the result follows. □

3.26 Definition. Let X be an object of $\mathsf{C}[\mathsf{P}]$. An element $P \in \mathsf{P}(X)$ is a *strict relation* for $(X, \approx)$ if

$$\mathsf{P} \models P(x) \to x \approx x \qquad \text{and} \qquad P \models P(x) \wedge x \approx x' \to P(x').$$

3.27. Given a morphism $F : Y \to X$, the element $P_F = [\exists x F(x, y)] \in \mathsf{P}(Y)$ defines a strict relation. Conversely, if P is a strict relation on X, then we can make an object $(X, \approx_P)$ where

$$x \approx_P x' \iff x \approx x' \wedge P(x);$$

and $\approx_P$ additionally is a morphism $(X, \approx_P) \to (X, \approx)$. With the characterization from 3.25, it is immediate that this morphism is mono, so that this gives a subobject of X.

3.28 Lemma. *These constructions give a one-to-one correspondences between subobjects of* $(X, \approx)$ *and strict relations on it. These correspondences respect the orderings of* $\mathrm{Sub}((X, \approx))$ *and* $\mathsf{P}(X)$.

Proof. Suppose that P is a strict relation on $(X, \approx)$. From this we construct the monomorphism $\approx_P$, from which we construct the new strict relation $[\exists x'(x \approx_P x')]$, which is equivalent to

$$\exists x'(x \approx x' \wedge P(x)),$$

which is trivially equivalent (i.e., equal) to $P(x)$. If Q, P are strict relations, and $Q \leq P$, then $\approx_Q$ also represents a monomorphism $(X, \approx_Q) \to (X, \approx_P)$.

Conversely, suppose that $F : Y \to X$ is a monomorphism. We obtain the relation $\approx'$ on X given bx $x \approx' x' \iff x \approx x' \wedge \exists y^Y (F(y, x))$, which gives a subobject $(X, \approx')$. Then $F \in \mathsf{P}(X \times Y)$ itself is also a morphism $F : Y \to (X, \approx')$. In fact, it is an isomorphism, which is to say that the relation F^{-1} defined by $F^{-1}(y, x) \iff F(x, y)$ is also functional (or, single-valued and total). Checking this is a matter of unfolding definitions.

If we have two composable monomorphisms F, G, then of course

$$\exists z \exists y (G(z, y) \wedge F(y, x)) \implies \exists y (F(y, x)),$$

which is to say that $P_{G \circ F} \leq P_F$ in $\mathsf{P}(Y)$. □

3.29. Summarizing, the subobjects of $(X, \approx)$ are given by the strict relations on $(X, \approx)$, with the same ordering. From now on, we will identify the two whenever notationally convenient, and write $\mathrm{Sub}(X) \subseteq \mathsf{P}(X)$. Note, however, that $\mathrm{Sub}(X)$ does not inherit the Heyting algebra structure from $\mathsf{P}(X)$. For instance, the top element of $\mathrm{Sub}(X)$ is $[x \approx x]$, which is generally not equal to $\top$. However, $\bot \in \mathsf{P}(X)$ is strict, and joints and meets of strict relations are trivially strict. And while, for ϕ, ψ strict, $\phi \to \psi$ need not be strict, $(\phi \to \psi) \wedge [x = x]$ is. Thus, $\mathrm{Sub}(X)$ becomes a Heyting algebra with operations

$$\begin{aligned}
\bot_{\mathrm{Sub}(X)} &= \bot_{\mathsf{P}(X)}, \\
\top_{\mathrm{Sub}(X)} &= [x \approx x], \\
\phi \wedge_{\mathrm{Sub}(X)} \psi &= \phi \wedge_{\mathsf{P}(X)} \psi, \\
\phi \vee_{\mathrm{Sub}(X)} \psi &= \phi \vee_{\mathsf{P}(X)} \psi, \\
\phi \to_{\mathrm{Sub}(X)} \psi &= [x \approx x] \wedge_{\mathsf{P}(X)} (\phi \to_{\mathsf{P}(X)} \psi).
\end{aligned}$$

This gives us a good hint that $\mathsf{C}[\mathsf{P}]$ is itself a Heyting category, and in fact it is. To show that $\mathsf{C}[\mathsf{P}]$ is regular, it is helpful to have another representation of pullbacks of subobjects.

3.30 Lemma. *Let $F : X \to Y$ be a morphism of $\mathsf{C}[\mathsf{P}]$, and let $A \to Y$ be a subobject defined by the strict relation P_A. Then the pullback of A by F is the subobject of X defined by the strict relation*

$$P_{F^*(A)}(x) \iff \exists y^Y (F(x, y) \wedge P_A(y)).$$

Proof. Note that our construction of the pullback in 3.23 gives $X \times_Y A$ as a subobject of the product $X \times A$ given by the strict relation $[\exists y'^Y (F(x, y) \wedge y' \approx_A y)]$, which is equivalent to $[\exists y^Y (F(x, y) \wedge P_A(y))]$. This looks like the exact same relation as $P_{F^*(A)}$, but one is defined on X, and one on $X \times A$. Fortunately, it is easily seen that $G : F^*(A) \to X \times_Y A$ defined by

$$G(x, x', y) \iff x \approx x' \wedge P_A(y) \wedge F(x, y)$$

is an isomorphism. □

3.31 Proposition. *The category* $\mathsf{C}[\mathsf{P}]$ *is regular. If* $F : X \to Y$ *is a morphism,* $\mathrm{im}(F)$ *is the subobject of* Y *given by the strict relation* $[\exists x(F(x,y))]$.

Proof. It is easy to see that F factorizes through a subobject given by strict relation P if and only if

$$\mathsf{P} \models \forall x \forall y(F(x,y) \to P(y)).$$

Now if $G : Z \to Y$ is another morphism, the pullback of $\mathrm{im}(F)$ by G is given by the strict relation

$$[\exists y(G(z,y) \wedge \exists x(F(x,y)))];$$

and the image of the projection $X \times_Y Z \to Z$ is given by the strict relation

$$[\exists x \exists z(z \approx z' \wedge \exists y(F(x,y) \wedge G(z,y)))],$$

which are logically equivalent. □

3.32. Now that $\mathsf{C}[\mathsf{P}]$ is regular, it is also easily seen to be coherent: with our characterization of pullbacks of subobjects in $\mathsf{C}[\mathsf{P}]$ it comes down to the equivalence

$$\exists y(F(x,y) \wedge (P(y) \vee Q(y))) \iff \exists y(F(x,y) \wedge P(y)) \vee \exists y(F(x,y) \wedge Q(y)),$$

which follows from the single-valuedness of F. Hence, it just remains to be shown that the pullback mappings also have right adjoints.

3.33 Lemma. $\mathsf{C}[\mathsf{P}]$ *is a Heyting category.*

Proof. If P_B is a strict relation on Y, and $F : Y \to X$ a morphism, then we define

$$P_{\forall_F(B)}(y) \iff y \approx y \wedge \forall x(F(x,y) \to P_B(x)).$$

Then, the adjunction comes down to, for P_A any strict relation on X, the equivalence

$$\exists y(F(x,y) \wedge P_A(y)) \to P_B(x) \iff P_A(y) \to \forall x(F(x,y) \to P_B(x)). \qquad \square$$

3.34. As a result, we can yet again interpret logic in $\mathsf{C}[\mathsf{P}]$, and have a soundness theorem for this. In particular, equality on $(X, \approx)$ gets interpreted as $\approx \in \mathsf{P}(X \times X)$, in other words, as $\exists_\Delta(\top_{\mathrm{Sub}(X)})$. Note that if $\approx = \exists_\Delta \top_{\mathsf{P}(X)}$, then by our discussion in 3.29, all relations are strict for X. This means that, in particular, the logic of P embeds into the logic of $\mathsf{C}[\mathsf{P}]$: what P proves about X, $\mathsf{C}[\mathsf{P}]$ proves about $(X, \exists_\Delta(\top_{\mathsf{P}(X)}))$. However, in $\mathsf{C}[\mathsf{P}]$, a possibly wider notion of equality is available.

Note that if we are talking about $\mathsf{C}[\mathsf{P}]$, the interpretation brackets $[x.\phi(x)]$ take up a double meaning: they are both used for the interpretation in $\mathsf{P}(\mathrm{tp}(x))$, and for the interpretation as a subobject of $\mathrm{tp}(x)$ as an object of C. From context it should be clear which one we mean.

3.3 Triposes and Toposes

3.35. In this section, we finally define and consider triposes, which are hyperdoctrines with extra structure. This extra structure allows us to show that $\mathsf{C}[\mathsf{P}]$ is not just a Heyting category, it is in fact also a topos. In fact, the name tripos stands for Topos-Representing Indexed Pre-Ordered Set: its definition exists to package together sufficient information to build a topos with a particular logic.

3.36 Definition. Let T be a finitely complete category, and X an object of T. An object $\mathcal{P}(X)$ together with a subobject $\mu \to X \times \mathcal{P}(X)$ is called a *power object* of X if for each object Y and each monomorphism $P \to X \times Y$ there is a morphism $\chi_P : Y \to \mathcal{P}(X)$ with an arrow $P \to \mu$ such that

$$\begin{array}{ccc} P & \longrightarrow & \mu \\ \downarrow & & \downarrow \\ X \times Y & \xrightarrow{\mathrm{id}_X \times \chi_P} & X \times \mathcal{P}(X) \end{array}$$

is a pullback square. If every object of T has a power object, we call T a topos.

3.37. With this definition in mind, it will be fairly clear how a tripos adds the structure needed to define a topos: it adds a sort of proto-power objects. The only difficulty then lies in showing that these proto-power objects also gives rise to *actual* power objects in $\mathsf{C}[\mathsf{P}]$.

3.38 Definition. Let P be a hyperdoctrine over a Cartesian category C. Suppose that we have for each object X of C an object $\pi(X)$ of C with an element $\in_X \in \mathsf{P}(X \times \pi(X))$. Then, if for each element $\phi \in \mathsf{P}(X \times Y)$ there is a morphism $\{\phi\} : Y \to \pi(X)$ such that $\phi = (\mathrm{id}_X \times \{\phi\})^*(\in_X)$, we call P a *tripos*.

3.39. In the case where C is Cartesian closed, which covers most interesting cases as far as we are concerned, we could have simplified this definition to require instead an object Σ with an element $\sigma \in \mathsf{P}(\Sigma)$ such that if $\phi \in \mathsf{P}(X)$, there is a $\{\phi\} : X \to \Sigma$ such that $\phi = \{\phi\}^*(\sigma)$. The object Σ (together with σ) is called a generic object.

This mirrors two possible definitions of a topos: as a Cartesian closed category with a subobject classifier, or as a finitely complete category with power objects. Once the underlying category is Cartesian complete, the tripos "only" needs a proto-subobject classifier.

3.40. If the motivating example for a hyperdoctrine is a Heyting category with its subobject functor, then the motivating example for a tripos is a topos with its subobject functor: the role of $\pi(X)$ is played by the power object $\mathcal{P}(X)$ in the topos, the morphism $\{\phi\}$ is unique, and $\in_X$ is the usual membership predicate associated to such a power object.

The hyperdoctrines defined in 3.10 are all triposes. The underlying category of the hyperdoctrines is Set, so we may use the characterization of 3.39: the sets Σ from 3.9 are the objects Σ, and $\sigma \in \mathsf{P}(\Sigma)$ is given by the element represented

by the identity $\Sigma \to \Sigma$. Similarly, the hyperdoctrines of 3.5 are also triposes: there, the generic object is given by H, together with $\mathrm{id}_H \in H^H$.

Not all hyperdoctrines are triposes. For example, subobject functors on Heyting categories which are not toposes generally will not be. As a very simple example, you can consider a Heyting algebra, which in particular is a Heyting category.

3.41 Proposition. *If* P *is a tripos, then* $\mathsf{C}[\mathsf{P}]$ *has power objects.*

Proof. Let X be any object of $\mathsf{C}[\mathsf{P}]$. The underlying object of $\mathcal{P}(X)$ will be $\pi(X)$. To define $\approx_{\mathcal{P}(X)} \in \mathsf{P}(\pi(X) \times \pi(X))$, we first define $E \in P(\pi(X))$, using a variable Q of type $\pi(X)$, and using the abbreviation $Q(x)$ for $x \in_X Q$, as follows:

$$E(Q) \iff \forall x^X (Q(x) \to x \approx x) \wedge \forall x^X \forall x'^X (Q(x) \wedge x \approx x' \to Q(x'))$$

Then, we define for Q, Q'

$$Q \approx_{\mathcal{P}(X)} Q' \iff E(Q) \wedge \forall x^X (Q(x) \leftrightarrow Q'(x)).$$

The membership relation μ, a subobject of $X \times \mathcal{P}(X)$, is then given by the strict relation P_μ defined by

$$P_\mu(x, Q) \iff Q(x) \wedge E(Q).$$

Now let Y be any object, and $P(x, y)$ a strict relation on $X \times Y$. Then we define $\chi_P : Y \to \mathcal{P}(X)$ by

$$\chi_P(y, Q) \iff \forall x^X (Q(x) \leftrightarrow P(x, y)) \wedge y \approx y.$$

(Note that the notation $P(x)$ just means $P(x)$, while $Q(x)$ is an abbreviation for $x \in_X Q$.) The most difficult part of proving that χ_P is a morphism is the totality, i.e.

$$\mathsf{P} \models \forall y^Y (y \approx y \to \exists Q^{\mathcal{P}(X)} (\chi_P(y, Q))),$$

but now let $\{P\} : Y \to \pi(X)$ be such that $(\mathrm{id}_X \times \{P\})^*(\in_X) = P$, then we find that

$$\mathsf{P} \models \forall y^Y (y \approx y \to \chi_P(y, \{P\}(y))),$$

because in fact $\{P\}(y)(x)$, an abbreviation for $x \in_X \{P\}(y)$, evaluates by definition to $P \in \mathsf{P}(X \times Y)$.

Pulling back $\mu \to X \times \mathcal{P}(X)$ by $\mathrm{id}_X \times \chi_P : X \times Y \to X \times \mathcal{P}(X)$ gives, by 3.30, the subobject of $X \times Y$ defined by the strict relation

$$\exists Q^{\mathcal{P}(X)} (\forall x'^X (Q(x') \leftrightarrow P(x', y)) \wedge Q(x)).$$

That this implies $P(x, y)$ is obvious; and that $P(x, y)$ also implies this follows by taking the Q whose existence is guaranteed by the totality of χ_P. Hence χ_P is a characteristic morphism for the subobject defined by P.

Now, suppose that $\psi_P : Y \to \mathcal{P}(X)$ is another characteristic function of P. Then, again by 3.30, we know that

$$P(x,y) \iff \exists Q^{\mathcal{P}(X)}(\psi_P(y,Q) \wedge Q(x)).$$

We want to show that $\psi_P = \chi_P$ – that is, that

$$\psi_P(y,Q) \implies \forall x^X(Q(x) \leftrightarrow P(x,y)).$$

Substituting $P(x,y)$ with the above equivalence, and noting the single-valuedness of $\psi_P(y,Q)$, makes this implication trivial. □

3.42. The object of primary interest to us is $\mathsf{Set}[\mathsf{M}]$, the topos built out of the modified realizability tripos. We call it, of course, the *modified realizability topos*, or MRT for short. Let us reflect for a second what an object of the modified realizability tripos actually looks like.

An object $(X, \approx)$ consists of a set X, together with an equivalence class of functions $\approx$. If f is a representing function, then we have $f : X \times X \to \Sigma$ – that is, it takes elements x, x' to a pair of sets of natural numbers, with an equivalence relation. That is, we have two sets of naturals $f(x,x')_a \subseteq f(x,x')_p$, and an equivalence relation $\sim$ on $f(x,x')_p$. Recall, we always have $0 \in f(x,x')_p$. Furthermore, we have that

$$\mathsf{M} \models \forall x \forall x'(x \approx x' \to x' \approx x),$$
$$\mathsf{M} \models \forall x \forall x' \forall x''(x \approx x' \wedge x' \approx x'' \to x \approx x'').$$

In our particular case, this comes down to the existence of particular functions; for symmetry, we have a realizer (a natural number) r_s, such that for each x, x', if $n \in f(x,x')_p$ then $r_s n \in f(x',x)_p$, and furthermore this operation is equivariant, and if $n \in f(x,x')_a$ then also $r_s n \in f(x',x)_a$. For short, we wrote this as $r_s : f(x,x') \to f(x',x)$. Similarly, we have a realizer $r_t : f(x,x') \times f(x',x'') \to f(x,x'')$ for transitivity.

This is quite a mouthful, and all we have done is specify some abstract object. We will want to avoid working with MRT directly as much as possible. Fortunately, it will turn out in 4.2 that MAss, a category we are already comfortable with, appears inside MRT as a particularly nice subcategory.

As announced, we call the object $\mathsf{Set}[\mathsf{G}]$ the Grayson topos, Gray, or non-equivariant modified realizability topos. This is the topos which is studied and investigated in [9]. Its objects are the same as those of MRT, but without the equivalence relations.

Finally, as mentioned before, the category $\mathsf{Set}[E]$ is also called Eff, the *effective topos*.

If C is a topos, then $\mathsf{C} \cong \mathsf{C}[\mathrm{Sub}]$. The equivalence takes X to (X, Δ), and a morphism $f : X \to Y$ to $[f(x) = y]$. Unsurprisingly, a topos by itself does not contain enough information to construct *another* topos.

3.43. All toposes are in fact also Heyting categories. This is not something we will aim to prove here; however, we did see that it is true for our toposes built

from triposes. Toposes have even more structure: the power objects allow us to interpret higher-order logic. If we mean a variable R to range, say, over relations on $X \times Y$, then we can let R range over $\mathcal{P}(X \times Y)$. We interpret $R(x, y)$ using the membership predicate. In similar vein, toposes are Cartesian closed: we can find the exponential Y^X as a special subobject of $\mathsf{P}(X \times Y)$, namely as those relations R for which $\forall x^X \exists y^Y (R(x, y) \wedge \forall y'^Y (R(x, y') \to y = y'))$ holds.

As a result, toposes are powerful enough to support an internal mathematics. For a simple example, we are now able to define things like algebraic or relational structure internally (second order logic), or reason about topologies (third order logic). However, the logic still has the flavour of the tripos we started with.

Chapter 4

The Modified Realizability Topos

4.1 $\neg\neg$-separated objects

4.1. In this chapter we will investigate the modified realizability topos. Much like we looked at MAss in section 2.3, we want to investigate what mathematics looks like in MRT, and in particular which constructive principles hold in MRT. There will be no big surprises: again, we will find that independence of premise and the axiom of choice for finite types hold.

Recall from 3.42 that working with objects and morphisms of MRT directly typically requires tending to a large number of details. In order to make the topos easier the work with, we first show that MAss embeds in MRT as a nice subcategory. Then, a lot of our arguments translate (partially) to arguments about MAss, where they are easier to answer.

4.2. We exhibit MAss as a subcategory of MRT. Given a modified assembly $(X, P_X, \sim, \rho_X)$, we define an object $(X, \approx)$ of MRT. The relation $\approx$ is defined by

$$\begin{aligned} [x \approx y](\bar{x}, \bar{x}) &= (\rho_X(\bar{x}), P_X, \sim), \\ [x \approx y](\bar{x}, \bar{y}) &= (\emptyset, P_X, \sim) \qquad \text{if } \bar{x} \neq \bar{y}. \end{aligned}$$

If $f : X \to Y$ is a morphism of assemblies with realizer r_f, we define a functional relation $F \in \mathsf{P}(X \times Y)$ by

$$F(\bar{x}, \bar{y}) = \begin{cases} [x \approx x](\bar{x}) \times [y \approx y](\bar{y}) & \text{if } f(\bar{x}) = \bar{y}, \\ (\emptyset, (P, \sim) \times (Q, \sim)) & \text{else.} \end{cases}$$

Showing that F is relational, single-valued and strict is trivial. To show that it is total, that is,

$$\mathsf{M} \models \forall x (x \approx x \to \exists y (F(x, y))),$$

we use r_f to map a realizer for $x \approx x$ to one for $f(x) \approx f(x)$, which gives us a realizer for $x = x \wedge f(x) = f(x)$, which is the same as a realizer for $F(x, f(x))$.

Together, this gives a functor $\mathsf{MAss} \to \mathsf{MRT}$.

4.3. To avoid cumbersome notation, we will occasionally use the same letter for a variable and a point in the underlying set; for instance, we will say "realizer for $x \approx x$", when what we really mean is "realizer in $[\approx](x)$". When this could cause confusion, we stick to the more formal notation.

4.4 Proposition. *The functor* $\mathsf{MAss} \to \mathsf{MRT}$ *is full and faithful.*

Proof. Suppose the modified assemblies X and Y give rise to $(X, \approx)$ and $(Y, \approx)$ respectively. Suppose $F : (X, \approx) \to (Y, \approx)$ is a functional relation. Then we claim that for each $x \in X$, there is a unique $y \in Y$ such that $F(x, y)$ has an actual realizer. Since X is a modified assembly, $[x = x]$ always has an actual realizer, and from the totality of F we obtain some y such that $F(x, y)$ has an actual realizer; if $F(x, y')$ also has an actual realizer, then the single-valuedness of F gives us an actual realizer of $y \approx y'$. Because Y comes from a modified assembly, this means that $y = y'$.

Thus we define a function $f : X \to Y$ by letting $f(x)$ be the unique $y \in Y$ such that $F(x, y)$ has an actual realizer. This function obtains a realizer by composing the realizer for the totality of F (which turns a realizer for $[x = x]$ into one for $F(x, y)$ for some y) with the realizer for the strictness of F (which turns this into a realizer for $[x = x] \wedge [y = y]$) and a projection (to get a realizer for $[y = y]$). Thus, f is a morphism of assemblies. In particular, it gives rise to $\hat{F} : (X, \approx) \to (Y, \approx)$. And now the realizer for the strictness of F is the same as a realizer for $F(x, y) \to \hat{F}(x, y)$, so $F = \hat{F}$. We see that the functor is full.

Now suppose that $f, g : X \to Y$ are morphisms of assemblies which give rise to F, G respectively. If f and g are distinct, there is an $x \in X$ such that $f(x) \neq g(x)$, and thus $F(x, f(x))$ has an actual realizer while $G(x, f(x))$ does not. This precludes us from giving a realizer for $F(x, f(x)) \to G(x, f(x))$, so that $F \neq G$. Hence, our functor is faithful. □

4.5 Definition. Let X be an object of a topos. If

$$\forall x^X \forall y^X (\neg\neg(x \approx y) \to x \approx y)$$

holds in the topos, then X is called *$\neg\neg$-separated*.

4.6 Proposition. *If $(X, \approx)$ lies in the (essential) image of the functor* $\mathsf{MAss} \to \mathsf{MRT}$*, then $(X, \approx)$ is $\neg\neg$-separated.*

Proof. Recalling our translation in 3.29 and lemma 3.33, rendering the definition of $\neg\neg$-separated in the logic of M we have to show that

$$\mathsf{M} \models \forall x^X \forall y^X (x \approx x \wedge y \approx y \wedge \neg\neg(x \approx y) \to x \approx y)$$

holds, given that $(X, \approx)$ comes from a modified assembly. That is, we have to give a realizer for

$$x \approx x \wedge y \approx y \wedge \neg\neg(x \approx y) \to x \approx y.$$

We claim that the projection onto the first of the three coordinates suffices. Since $(X, \approx)$ comes from a modified assembly, the potential realizers for $[x \approx x]$ and $[x \approx y]$ are identical. If we have an actual realizer for $x \approx x \wedge y \approx y \wedge \neg\neg(x \approx y)$, then in fact $x \approx y$ must also have an actual realizer, and thus since X comes from a modified assembly, we must have $x = y$, and then their actual realizers are also identical. Finally, projections are always equivariant. □

4.7 Proposition. *If an object $(X, \approx)$ of* MRT *is $\neg\neg$-separated, then it lies in the essential image of the functor* $\mathsf{MAss} \to \mathsf{MRT}$.

Proof. Let Y be the set of equivalence classes of $(X, \approx)$, where equivalence of x and y means $[x \approx y]_a \neq \emptyset$. Furthermore, recalling 3.8, define

$$(P_Y, \sim) = \bigcup_{x,x' \in X} [x \approx x']_p,$$

and define for each $y \in Y$

$$\rho_Y(y) = \bigcup_{x,x' \in y} [x \approx x']_a.$$

Then $(Y, P_Y, \sim, \rho_Y)$ is a modified assembly, giving rise to an object $(Y, \approx)$ of MRT. Now define $F \in \mathsf{M}(X \times Y)$ by

$$F(x,y) = \begin{cases} [x \approx x] \times [y \approx y] & \text{if } x \in y, \\ (\emptyset, [x \approx x]_p \times [y \approx y]_p) & \text{else.} \end{cases}$$

We claim that this is an invertible functional relation; that is, that F and F^{-1} defined by $F^{-1}(y,x) = F(x,y)$ are functional relations.

Strictness of F and F^{-1} is realized by the identity; relationality of F and F^{-1} follows by some applications the realizers for transitivity for both objects. Single-valuedness of F is also trivial, since Y comes from a modified assembly. For single-valuedness of F^{-1}, we have to give a realizer for

$$F(x,y) \wedge F(x',y) \to x = x'.$$

The $\neg\neg$-separation of X gives us a realizer $r_{\neg\neg}$ for $x = x \wedge x' = x' \wedge \neg\neg(x = x') \to x = x'$. Applying it to the triple consisting of the first projection of $F(x,y)$, the first projection of $F(x',y)$ and the number 0 gives us the realizer for $x = x'$ we want.

The totality of F is realized by the diagonal. The totality of F^{-1} is realized by taking a realizer for $y = y$, which is a realizer for $x = x'$ for some $x, x' \in y$; and using symmetry and transitivity to produce a realizer for $x = x$ from that. □

4.8. Thus, the modified assemblies are precisely the $\neg\neg$-separated objects. This is convenient, since the $\neg\neg$-separated objects form a very nice class. For instance, it is closed under subobjects: if $F : (X, \approx) \to (Y, \approx)$ is a monomorphism, and Y is $\neg\neg$-separated, then from $\neg\neg(x^X = x'^X)$ we can conclude

$\exists y^Y, y'^Y (F(x,y) \wedge F(x',y') \wedge \neg\neg(y \approx y'))$, so that we can conclude that $y \approx y'$, and thus that $x \approx x'$ by our characterization of monomorphisms from 3.25. (In particular, although we already know, the category MAss is automatically a Heyting category because MRT is.) This in turn means that we can interpret our logical language in MAss instead of MRT whenever we are working with $\neg\neg$-separated objects.

Furthermore, it is closed under the exponents of MRT. For, suppose X, Y are objects, with Y a $\neg\neg$-separated object, and let us reason in the internal logic about $f, f' : X \to Y$. If $\neg\neg(f = f')$, then for each $x \in X$ we have $\neg\neg(f(x) = f'(x))$, and by $\neg\neg$-separation of Y, it follows that $f(x) = f'(x)$. As we noted in 2.39, the internal logic of any Cartesian closed Heyting category has function extensionality, so it follows that $f = f'$.

4.2 Independence of Premise

4.9. We know that independence of premise holds for the appropriate types in MAss; unsurprisingly, the same is true for MRT. In fact, the condition on the object Y is almost exactly the same as in the statement of the theorem for MAss.

4.10 Theorem. *Let X, Y be objects of MRT, where Y has the property that if $n \in [y \approx y]_p$, then there is y' with $n \in [y' \approx y']_a$. Then for any predicates P of type X and R of type (X, Y) we have*

$$\mathsf{MRT} \models \forall x^X((\neg P(x) \to \exists y^Y R(x,y)) \to \exists y^Y(\neg P(x) \to R(x,y))).$$

Proof. Recalling the translation of the logic of MRT into that of M from 3.29 and lemma 3.33, we can equivalently prove that

$$\mathsf{M} \models \forall x^X(x \approx x \wedge (\neg P(x) \to \exists y^Y R(x,y)) \to \exists y^Y(y \approx y \wedge (\neg P(x) \to R(x,y))))$$

for any sets X, Y, symmetric and transitive relations $\approx_X, \approx_Y$ where $\approx_Y$ has the property described above, and strict relations P, R. That is, we have to give a realizer showing that in $\mathsf{M}(X)$,

$$[x \approx x] \wedge [\neg P(x) \to \exists y(R(x,y))] \leq [\exists y(y \approx y \wedge (\neg P(x) \to R(x,y)))]. \quad (4.1)$$

Now, the strictness of R gives us a realizer for $R(x,y) \to y \approx y$; we call this realizer r_y. We claim that the code

$$\lambda e r_a.\langle r_y(r_a 0), \lambda k.r_a 0\rangle$$

realizes 4.1. To this end, suppose that e, r_a are potential realizers for the left hand side at a point $\bar{x} \in X$. Note that because y is inhabited, we can use the characterization of 3.13, so that

$$[\exists y(y \approx y \wedge (\neg P(x) \to R(x,y)))](\bar{x}) = \bigcup_{\bar{y} \in Y} [y \approx y](\bar{x},\bar{y}) \wedge [\neg P(x) \to R(x,y)](\bar{x},\bar{y}).$$

Then, since 0 is always a potential realizer in $[\neg P(x)](\bar{x})$, indeed $r_a 0$ is a potential realizer for

$$[\exists y(R(x,y))](\bar{x}) = \bigcup_{\bar{y}\in Y} [R](\bar{x},\bar{y});$$

we call some $\bar{y}$ for which it is in there $\bar{y}_0$. Then, $r_y(r_a 0)$ also lies in $[y \approx y](\bar{y}_0)_p$, and $\lambda k.r_a 0$ also lies in $[\neg P(x) \to R(x,y)](\bar{x},\bar{y}_0)_p$, showing that this is indeed a potential realizer. Now, it remains to be shown that it is also an actual realizer.

Now suppose that r_a is an actual realizer. There are two possibilities: either $[\neg P](\bar{x})$ has an actual realizer, or it does not. If it does, then $[\neg P(x)](\bar{x})$ in particular always has 0 as an actual realizer, and the proof goes through exactly as for potential realizers.

If it does not, then

$$\begin{aligned}&[\exists y(y \approx y \wedge (\neg P(x) \to R(x,y)))](\bar{x})_a \\ &\qquad = \bigcup_{\bar{y}\in Y} [x,y.y \approx y](\bar{x},\bar{y})_a \times ([x,y.\neg P(x)](\bar{x},\bar{y})_p \to [R(x,y)](\bar{x},\bar{y})_p).\end{aligned}$$

Now, we already know that $\lambda k.r_a 0$ is a potential realizer for $[\neg P(x) \to R(x,y)](\bar{x},\bar{y})$ for any $\bar{y}$; and, by our assumption on Y, since $r_y(r_a 0)$ is a potential realizer of some $\bar{y} \in Y$, it must also be the actual realizer of some $\bar{y} \in Y$. □

4.11. The above construction does not anywhere *use* equivariance, so in particular the same argument works for Gray, in which independence of premise also holds, under the same conditions. However, IP does not hold in Eff; we already saw that it does not hold in Asm, which form the $\neg\neg$-separated objects in Eff.

4.3 The Axiom of Choice for Finite Types

4.12. In order to talk about the axiom of choice for finite types in MRT we need to know what the finite types are. As luck would have it, it will turn out that the natural numbers object in MRT is $\neg\neg$-separated – in fact, it is the natural numbers object of MAss. Since we already know that $\neg\neg$-separated objects are closed under exponentiation in MRT, it follows that the finite types in MRT are just those of MAss. These have a simple description as the types of HEO, while the construction of exponents in MRT invites a lot of technical overhead. Even just checking that the natural numbers object of MAss is indeed a natural numbers object of MRT gets into so much technical detail, that we prefer to just construct the morphism.

4.13 Lemma. *The image under the functor* MAss $\to$ MRT *of the natural numbers object* $\mathbb{N}$ *in* MAss*, which we also denote* $\mathbb{N}$*, is the natural numbers object of* MRT.

Proof. Concretely, let 1 be given by the set $\{*\}$, with $[* \approx *] = (\mathbb{N},\mathbb{N},\mathbb{N}\times\mathbb{N})$. Now suppose we have morphisms $Q : 1 \to X$ and $F : X \to X$ of MRT. Then

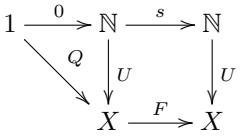
we define $U \in \mathsf{M}(\mathbb{N} \times X)$ by

$$U(0,x) = (\{1\},\{0,1\},=) \times Q(*,x),$$
$$U(n+1,x) = (\{n+2\},\{0,n+2\},=) \times \bigcup_{y \in X} U(n,y) \times F(y,x).$$

Note that any program using a realizer of $U(n,x)$ can recognise whether or not it is an actual or a potential realizer: if the first projection is 0, it is necessarily a potential realizer, and can thus always be sent to 0. If it is non-zero, it is always an actual realizer, and furthermore by subtracting 1 the program can use the value n.

Hence, strictness, relationality, and single-valuedness of U are all trivial using the fact that Q and F are. Totality is realized as follows: 0 is sent to a predetermined realizer of $U(0,x)$, while $n+1$ recursively finds a realizer of $U(n,y)$ and then uses totality of F to find a realizer for $F(y,x)$ allowing us to build a realizer for $U(n+1,x)$.

By construction, U is precisely the unique arrow making the diagram



commute. □

4.14 Theorem. *In* MRT*, the axiom of choice holds for all finite types. That is, for σ, τ finite types, and α any type,*

$$\mathsf{MRT} \models \forall a^{\alpha}(\forall x^{\tau} \exists y^{\sigma} P(x,y,a) \to \exists f^{\sigma \to \tau} \forall x^{\sigma} P(x, f(x), a)).$$

4.15. Since the only thing we have changed from our statement of the axiom of choice in MAss is that the parameter type A is now possibly no longer $\neg\neg$-separated, but the parameter did not figure into the proof in any way, the proof should be very similar to that of 2.46, and indeed it is.

Proof. Translated into the logic of the tripos, we want to demonstrate the existence of an actual realizer, uniformly in A, of

$$\mathsf{M} \models \forall x^{\tau}(x \approx x \to \exists y^{\sigma} P(x,y,a)) \to$$
$$\exists f^{\sigma \to \tau} \forall x^{\sigma}(x \approx x \to \exists y^{\sigma}(\mathrm{ev}(x,f,y) \wedge P(x,y,a))$$

whenever P is a strict relation. In particular, the strictness of P gives us an actual realizer r_s for $P(x,y,a) \to y \approx y$.

Since the underlying sets of $\sigma, \tau, \sigma \to \tau$ are always non-empty, we can use the quantifier definitions from 3.13. Thus, realizers of the antecedent are just realizers for $x \approx x \to \exists y^{\sigma}(P(x,y,a))$ which hold for all x. Now, if r_a is a potential realizer of the antecedent, we claim that $r_s \circ r_a$ is a (code for an)

element of $\sigma \to \tau$. This function takes potential realizers for $x \approx x$ – that is, codes for elements of type σ – as an argument, and it outputs potential realizers for $y \approx y$ – that is, codes for elements of type τ. Because r_s, r_a have to be equivariant, codes for the same element of σ have to be sent to codes for the same element of τ, which gives a well-defined function $\bar{f} : \sigma \to \tau$, which is of course realized by $r_f = r_s \circ r_a$.

Let us use this to build a potential realizer for the consequent. Recall that ev, the evaluation morphism $\tau \times \sigma^\tau \to \sigma$ in MRT, comes from MAss, and thus has the potential realizers of $[x \approx x] \wedge [f \approx f] \wedge [y \approx y]$. The actual realizers agree with those as well as long as $f(x) = y$; otherwise, $\mathrm{ev}(x, f, y)$ has no actual realizers. Thus, from r_a a potential realizer of the antecedent, and r_x a potential realizer for $x \approx x$, we can build a potential realizer for $\mathrm{ev}(x, f, y)$ as

$$\langle r_x, r_f, r_f r_x \rangle.$$

Hence, our realizer for the axiom of choice becomes

$$\lambda r_a r_x . \langle \langle r_x, r_s \circ r_a, r_s(r_a r_x) \rangle, r_f r_x \rangle.$$

Since the potential realizers for $\mathrm{ev}(x, f, y)$ are global, this always works for potential realizers. Furthermore, this realizer is equivariant, since it is just a composition of pairing operators and realizers, all of which must be equivariant.

Finally, let us see that our realizer is an *actual* realizer. Thus, for every x an element of type τ, if r_a and r_x are actual realizers of the antecedent and $x \approx x$, then $r_s \circ r_a$ needs to be the actual realizer of a function $\bar{f}$ such that $r_s(r_a r_x)$ is a code of $f(x)$, and $r_f r_x$ is an actual realizer of P at $(x, \bar{f}(x), a)$. But this is immediate from the definition of $\bar{f}$. □

4.16. Since the axiom of choice for finite types does not hold in GrAss, it does not hold in Gray either. Thus, like with MAss and GrAss, the axiom of choice for finite types distinguishes the two toposes. Also, since the axiom of choice does not even hold for all types in MAss, it certainly will not for all types of MRT either.

4.17. Before we conclude, let us look at how MRT and Gray relate geometrically. Recall that Gray is essentially "MRT without equivalence relations"; this immediately gives us a forgetful functor $\mathsf{MRT} \to \mathsf{Gray}$, which simply removes the equivalence relation on each set of potential realizers. The other way around, we can equip an object or a morphism of Gray with either the total equivalence relation on each set of potential realizers, or with the equality relation; this gives two functors $\mathsf{Gray} \to \mathsf{MRT}$.

It is not difficult to check that the functor adding the equality relation is left adjoint to the forgetful functor, while the functor adding the total relation is its right adjoint. Hence, the three functors assemble into an essential geometric morphism $\mathsf{Gray} \to \mathsf{MRT}$. However, these functors are not logical functors, so from the point of logic this does not tell us something particularly interesting.

Conclusion

In this thesis, we have defined a topos MRT whose internal logic is meant to reflect Kreisel's modified realizability for arithmetic in all finite types. We seem to have succeeded: a sentence in the language of HA^ω is realized (in modified realizability) if and only if its interpretation in the topos is true. Furthermore, we find that independence of premise, one of the characterizing features of modified realizability, holds in the topos more widely that for the finite types.

The construction of MRT is closely analogous to the construction of another topos, also called the "modified realizability topos", which we have named Gray in this thesis after the first person to define it. The topos Gray models a different kind of modified realizability; the crucial logical difference is that the axiom of choice does for finite types does not hold in Gray. Conversely, Church's thesis does not hold (internally) for MRT.

In MRT we have singled out the category MAss based on the $\neg\neg$-separated objects, objects for which the equivalence $(x = y) \leftrightarrow \neg\neg(x = y)$ holds. This category turns out to be closed under subobjects and exponents, and includes many of the interesting objects of MRT, most notably all finite types. Furthermore, this category allows a much simpler description – which in fact we have given and investigated first.

There are some avenues unexplored with regards to MRT in this thesis:

- We build MRT on the computational model of natural numbers with recursive function application. More generally, it seems that the construction would go through on the basis of any PCA. The toposes this creates could have interesting distinct logical properties.
- We have focused on statements in the "first-order" language of Heyting arithmetic in all finite types. Of course, much more mathematics can be developed within an elementary topos like MRT. In particular, we have not considered the real numbers. If we accept the premise that MRT is the "right" topos for Kreisel's modified realizability, then the analysis of MRT is the (as of yet unexplored) analysis of Kreisel's modified realizability.
- While we characterized the projective objects in MAss, we did not yet do so in MRT.
- The objects Y for which the principle of independence of premise holds in MRT seem to be objects from a category analogous to Eff, but where the

union of the sets of actual realizers has an equivalence relation on it. This might be an interesting subcategory of MRT.

- The modest modified assemblies, as described in 2.60, are analogous to a type of object called a "modest set" in the effective topos. These modest sets have been studied relatively extensively; perhaps the study of modest modified assemblies could be an interesting look into MRT.

Each of these could be an avenue for further investigation into the topos MRT and its internal mathematics.

Bibliography

[1] Aurelio Carboni and Enrico M. Vitale. Regular and exact completions. *Journal of pure and applied algebra*, 125(1-3):79–116, 1998.

[2] J. Roger Hindley and Jonathan P. Seldin. *Lambda-Calculus and Combinators: an Introduction*, volume 13. Cambridge University Press Cambridge, 2008.

[3] Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*. Oxford University Press, 2002.

[4] Stephen C. Kleene. On the interpretation of intuitionistic number theory. *The Journal of Symbolic Logic*, 10(4):109–124, 1945.

[5] Georg Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. Amsterdam: North-Holland Pub. Co., 1959.

[6] Georg Kreisel. On weak completeness of intuitionistic predicate logic. *The Journal of Symbolic Logic*, 27(2):139–158, 1962.

[7] Anne S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*. Springer Science & Business Media, 1973.

[8] Benno van den Berg. A note on arithmetic in finite types, 2016. arXiv:1408.3557.

[9] Jaap van Oosten. The modified realizability topos. *Journal of pure and applied algebra*, 116(1-3):273–289, 1997.

[10] Jaap van Oosten. Realizability: An historical essay. *Mathematical Structures in Computer Science*, 12(03):239–263, 2002.

[11] Jaap van Oosten. *Realizability: An Introduction to its Categorical Side*, volume 152. Elsevier, 2008.