Solving Domain Equations using Löb Induction

Julien Marquet-Wagner

August 11, 2025

Abstract

In the context of the theory of the Iris framework, we study the problem of solving recursive domain equations. Throughout this note, we work with a simple such recursive domain equation and build the necessary tools to state and solve it. Our approach follows previous work by Birkedal and Møgelberg (2013), who show that recursive domain equations can be solved by applying Löb induction on a suitable universe. This technique is especially relevant in the context of synthetic guarded domain theory (Birkedal et al., 2012). The present note is meant as a detailed illustration of this approach. Our development takes place within the *topos of trees*.

Introduction

Guarded recursive types are useful to give semantics to programming languages and program logics. In Iris (Jung et al., 2018), to define the semantics of a program logic, one defines *weakest precondition predicates* by recursion (Jung et al., 2018, §6.3). This definition, however, is not structurally recursive. This is where the *guarded* aspect comes into play. The definition of weakest precondition predicates is self-referential, but all recursive references are *guarded* under a modality called *later*. Similarly, the type of Iris propositions is defined recursively, where all its recursive appearences in its defining equation are guarded under a similar later operator (Jung et al., 2018, §4.6). This places the study of guarded recursive definitions at the very heart of Iris, and twice so: once for the definition of certain types like that of Iris propositions, and a second time for the definition of inhabitants of such types, like the weakest precondition predicates which take values in Iris propositions.

Synthetic guarded domain theory (SGDT) (Birkedal et al., 2012) is the study of type theories that feature such a *later* modality. This modality originates in the work of Nakano (2000) and its core property is that it enables *Löb induction*, a general tool to construct fixed points of functions, provided they satisfy a certain guardedness condition. In SGDT, one very desirable property is the existence of fixed points of *guarded recursive domain equations*, *i.e.* type equations in which recursive occurences of a type always appears under the *later* modality.

In this note, we illustrate the problem of solving guarded recursive domain equations by detailing the example of the type `Stream`, which we mean to be the solution of the following equation

$$\texttt{Stream} \simeq \mathbb{N} \times \blacktriangleright\texttt{Stream} \tag{$*$}$$

To construct solutions to guarded recursive domain equations, Iris uses America and Rutten's Theorem (Jung et al., 2018, §4.6, Theorem 2, America and Rutten, 1989, Birkedal et al., 2010), which is concerned with bifunctors that satisfy a certain technical condition. Birkedal and

Møgelberg (2013) provide an alternative to America and Rutten's theorem and show that the existence of solutions to recursive domain equations is a consequence of Löb induction and of the existence of a *universe*. This is the technique we will use to solve equation (∗).

This comes at a cost. Löb induction can only produce elements of some object. Therefore, to feed objects themselves into Löb induction, we need a way to transform objects into elements of some larger object. This is exactly the purpose of universes. Universes were introduced in algebraic geometry by Grothendieck as a tool to study category theory (Artin et al., 1971, Appendice). This notion was later refined by Bénabou (1973), and Streicher (2005) recast it in the context of the categorical semantics of type theory.

Another departure from the standard presentation of Iris is that we don't use *COFEs* (Di Gianantonio and Miculan, 2002, Jung et al., 2018, Fig. 7). The approach of Birkedal and Møgelberg (2013) is in line with work on synthetic guarded domain theory, and we work with the *topos of trees* (Birkedal et al., 2012).

Outline. This note is structured as follows.

- In section 1, we introduce the *topos of trees*. We explain the relevant category-theoretical notions that will come into play in our study.
- In section 2, we introduce the *later* operator. This is where we introduce *Löb induction* (theorem 2.3).
- In section 3, we introduce *universes* and construct one such universe in the topos of trees (definition 3.10, proposition 3.25).
- In section 4, we apply our work to solving equation (∗).

Acknowledgements. This work was conducted from April to August 2025 during an internship at Aarhus University, as part of my Master's degree at ENS Ulm. I was under the supervision of Lars Birkedal and Daniel Gratzer. I would like to thank them for their time, encouragement, and helpful guidance. I would also like to extend my thanks to everyone I met in the LogSem team and in Aarhus University, for their warm welcome to Denmark.

Contents

1 The Topos of Trees

Trees. Trees represent objects that evolve along discrete time steps. Together with a notion of morphism of trees, they assemble in a category.

Notation 1.1. We note ω the first infinite ordinal. Its elements are the natural numbers $0, 1, 2, 3, ...$ We insist on writing ω instead of $\mathbb{N}$ because we are only really interested in the order–theoretic aspects of ω, and not in the algebraic aspects of $\mathbb{N}$.

Definition 1.2. A tree F is given by the following data
— For $t \in \omega$, a set denoted F_t.
— For $t, u \in \omega$, when $u \leq t$, a map $F_{u\leq t} : F_t \to F_u$.
subject to the following constraints
— For $t \in \omega$, $\forall x \in F_t,\ F_{t\leq t}x = x$
— For $t, u, v \in \omega$ such that $u \leq t$ and $v \leq u$, $\forall x \in F_t,\ F_{v\leq u}(F_{u\leq t}\, x) = F_{v\leq t}\, x$

Definition 1.3. Let F, G be trees. A morphism $\varphi : F \longrightarrow G$ is given by the following data
— For $t \in \omega$, a map $\varphi_t : F_t \to G_t$
subject to the following constraint
— For all $t, u \in \omega$ such that $u \leq t$, $\forall x \in F_t, G_{u\leq t}(\varphi_t\, x) = \varphi_u(F_{u\leq t}\, x)$

Drawing trees. We find the following graphical representatioons convenient. These diagrams highlight the flow of information underlying trees and their morphisms.

$$F_0 \xleftarrow{F_{0\leq 1}} F_1 \xleftarrow{F_{1\leq 2}} F_2 \quad \cdots$$

A tree F

$$\begin{array}{ccccccc} F_0 & \longleftarrow & F_1 & \longleftarrow & F_2 & & \cdots \\ \phi_0\downarrow & & \phi_1\downarrow & & \phi_2\downarrow & & \\ G_0 & \longleftarrow & G_1 & \longleftarrow & G_2 & & \cdots \end{array}$$

A morphism of trees $\varphi : F \to G$

Trees are presheaves. Trees are really just presheaves over ω, and morphisms of trees are really just natural transformations between such presheaves. We won't develop a general theory of presheaf toposes, so we will stick with trees and refer to the category of trees and their morphisms as "the topos of trees" without defining toposes in general. We will however see (some of) the structure and properties of the topos of trees that make it deserve to be called a topos. Since trees are presheaves, we will use the following notation.

Notation 1.4. The category of trees (the topos of trees) is noted $\mathbf{Psh}\,\omega$.

Geometry and logic. In the category of sets, "the" point is "the" set with one element. We denote it by $*$, and we also denote its element by $*$. Although there are many sets with only one element, we use the singular to refer to the point as it is unique up to isomorphism. Similarly, in the topos of trees, there is a terminal object.

Construction 1.5. The point $* \in \mathbf{Psh}\,\omega$ is defined by
— $*_t \equiv *$ the terminal set
— $*_{u\leq t} \equiv\, !$ the only map into $*$
This $* \in \mathbf{Psh}\,\omega$ is terminal, as a consequence of the terminality of the set $*$.

The category of sets also features the empty set $\emptyset$ (which is really unique by set extensionality), and constructors like cartesian products $A \times B$ and disjoint sums $A + B$. These sets also have analogs in the topos of trees: there are trees that have the same universal properties.

Construction 1.6. The void $\emptyset \in \mathbf{Psh}\,\omega$ is defined by
— $\emptyset_t \equiv \emptyset$ the initial set
— $\emptyset_t \equiv\, !$ the only map from $\emptyset$.

Construction 1.7. Let A and B be trees. Their product $A \times B$ can be explicitly constructed as
— $(A \times B)_t \equiv A_t \times B_t$
— $(A \times B)_{u \le t}\,(x, y) \equiv (A_{u \le t}\,x, A_{u \le t}\,y)$
This satisfies the universal property of products since the sets $A_t \times B_t$ satisfy this property in the category of sets.

Construction 1.8. Let A and B be trees. Their sum $A + B$ can be explicitly constructed as
— $(A + B)_t \equiv A_t + B_t$
— $(A + B)_{u \le t}\,(x, y) \equiv (A_{u \le t}\,x, A_{u \le t}\,y)$
This satisfies the universal property of sums since the sets $A_t + B_t$ satisfy this property in the category of sets.

The fact that these constructions are available in the topos of trees indicates that the topos of trees behaves like the category of sets. We will briefly develop a geometrical and a logical perspectives on sets, which carry over to trees. These perspectives will make the Yoneda embeddings (definition 1.9) stand out as "nonclassical" objects. First, we focus on the classical case of sets. In the next paragraph, we will see that trees behave non-classically.

We can reinterpret all previous constructions as propositions, along the idea that a set is a proposition that is true if and only if the set is inhabited. We find that since $\emptyset$ is never inhabited it represents falsehood. $A \times B$ is inhabited if and only if both A and B are inhabited, so it represents the logical "and". $A + B$ is inhabited if and only if either A or B, or both, is inhabited, so it represents the logical "or". This is the category of sets seen from the point of view of its internal logic. We find that we have an analog of the excluded middle: a set that is not $\emptyset$ always has an inhabitant.

The geometric point of view on sets is to look at elements. The key idea is that elements of a set X bijectively correspond to maps $* \to X$. From this point of view, we like to call a map $* \to X$ a *point* of X and we like to call $*$ *"the" point*. We find that a set that has no point is necessarily empty. The point can't be "fragmented": there is nothing in between the void and the point.

Nonclassical elements. Now, the topos of trees can also be approached from the logical point of view and the geometrical point of view. However the topos of trees features some new, exotic objects that violate the excluded middle (from the logical perspective) and behave like fragments of the point (from the geometrical perspective). These new objects are the innocent-looking Yoneda embeddings.

Definition 1.9. Let $t \in \omega$. The tree $\mathbb{y}_t$ is defined as follows.
— When $u \le t$, $(\mathbb{y}_t)_u \equiv *$
— When $u > t$, $(\mathbb{y}_t)_u \equiv \emptyset$
— When $u \le t$, $(\mathbb{y}_t)_{v \le u} * \equiv *$
— When $u > t$, $(\mathbb{y}_t)_{v \le u}$ is the only map from $\emptyset$ to $(\mathbb{y}_t)_v$

Proposition 1.10. Let $t \in \omega$. There is no map $\mathbb{y}_t \to \emptyset$. In other words, $\mathbb{y}_t$ is not empty. Equivalently, $\mathbb{y}_t$ is not false.

Proof. A morphism $\varphi : \mathbb{y}_t \to \emptyset$ would map $(\mathbb{y}_t)_t = *$ to $\emptyset$, a contradiction. □

Proposition 1.11. Let $t \in \omega$. There is no map $* \to \mathbb{y}_t$. In other words, $\mathbb{y}_t$ has no point. Equivalently, $\mathbb{y}_t$ is not true.

Proof. A morphism $\varphi : * \to \mathbb{y}_t$ would map into $(\mathbb{y}_t)_{t+1} = \emptyset$ from $*$, a contradiction. □

On the one hand, we will have to pay for this extra structure, and the price is that we won't be able to adapt every idea from set theory. On the other hand, this new structure provides provides us with new ways to reason. Chief among them will be Löb induction which we will encounter later in section 2.1.

1.1 Pullbacks and Fibered Data

Fibered and Indexed Data. In the topos of trees, and more generally in geometry and in categorical semantics of functional languages, there is a way to think about data that depends on some other data. The idea is to define, with just the tools that we have in the topos of tree, an analog of type families.

First, let us look at the familiar context of set theory. We will reformulate the usual set theoretical notions in a way that can be carried over to the topos of trees.

Let I be a set of indices. Let $(X_i)_{i \in I}$ be a family of sets indexed over I. From $(X_i)_i$ we can construct $E \equiv \{(i, x) \mid i \in I, x \in X_i\}$. This E is equiped with a projection $\pi : E \to B$, defined as $\pi(i, x) \equiv i$. The trick here is that we can recover $(X_i)_i$ from E and p. Indeed, we find that X_i is in bijection with $\{y \in E \mid \pi(y) = i\}$. More precisely, we the function $X_i \to \pi^{-1}(\{i\})$ that maps x to (i, x) is a bijection. Conversely, from any pair like (E, π) we can construct a family like $(X_i)_{i \in I}$.

What this means is that the *indexed* point of view — that of $(X_i)_i$ — is equivalent to the *fibered* point of view — that of (E, π). The benefit of the fibered point of view is that fibered data is defined using just a function between two objects, a definition that can be carried over to the topos of trees without change.

Remark 1.12. We will recover the indexed point of view for trees when we introduce the universe in section 3.

Pullacks as a tool to sudy fibered data. Let $B \in \mathbf{Psh}\,\omega$ be a tree, which we note B like *B*ase. Let $E \in \mathbf{Psh}\,\omega$ be another tree, which we want to think of *E*verything that lies over the *B*ase. Finally, let $p : E \to B$, which we will think of as a *p*rojection. Any such (E, p) can be seen as fibered tree over B. We now describe the central tool at our disposal to manipulate families: *pullbacks*. Using pullbacks, we can recover the ideas of reindexing, of inverse images and of various operations on families of data over a base.

Construction 1.13. Let $A, B, C \in \mathbf{Psh}\,\omega$ and $p : A \to B$, $q : C \to B$. We construct $A \times_B C$, which fits in the following cartesian square

$$\begin{array}{ccc} A \times_B C & \longrightarrow & C \\ \downarrow & \lrcorner & \downarrow{\scriptstyle q} \\ A & \xrightarrow[p]{} & B \end{array}$$

We define $A \times_B C$ as follows
— $(A \times_B C)_t \equiv \{(x \in A_t, y \in C_t) \mid p_t(x) = q_t(y)\}$
— $(A \times_B C)_{u \le t}\,(x, y) \equiv (A_{u \le t}\,x, C_{u \le t}\,y)$
The maps on the left and on the top of the diagram are defined as the obvious projections.

The universality of $A \times_B C$ comes from the fact that $(A \times_B C)_t$ is defined as a pullback of A_t and C_t over B_t.

Pullbacks generalize the operation of reindexing. In the context of set theory, a family $(X_i)_{i \in I}$ over i may be reindexed using a function $f : J \to I$, yielding $(X_{f(j)})_{j \in J}$. Using the fibered point of view, this assembles in the following diagram

$$\begin{CD} \{(j, x) \mid x \in X_{f(j)}\} @>>> \{(i, x) \mid x \in X_i\} \\ @V{\pi_1}VV @VV{\pi_1}V \\ J @>>{f}> I \end{CD}$$

This reindexing operation on fibered data has its own notation.

Notation 1.14. Let $A, B \in \mathbf{Psh}\,\omega$ and $f : A \to B$. Let $E \in \mathbf{Psh}\,\omega$ and $p : F \to B$. We note $f^*E \in \mathbf{Psh}\,\omega$ and $f^*p : f^*E \to A$ the data of the pullback as in the following diagram

$$\begin{CD} f^*E @>>> E \\ @V{f^*p}VV @VV{p}V \\ A @>>{f}> B \end{CD}$$

1.2 Global and Local Elements

Global elements. By analogy with the set theoretical case, we will call maps $* \to A$ *points* or *elements* of A. To be more precise, we will call these *global*, as we will later introduce *local* notions.

Definition 1.15. Let $A \in \mathbf{Psh}\,\omega$. A *global point* or *global element* of A is a map $* \to A$.

Let us look at the interaction between points and fibered data. In the context of set theory, when $(X_i)_{i \in I}$ is a family of sets indexed over I, one can take $i \in I$ and have the set X_i. This operation can be seen in the following pullback

$$\begin{CD} X_i @>>> \{(i, x) \mid x \in X_i\} \\ @V{!}VV @VV{\pi_1}V \\ * @>>{* \mapsto i}> I \end{CD}$$

and, from the fibered point of view in sets, pulling back over an element can be done by using inverse images

$$\begin{CD} \pi^{-1}(\{i\}) @>>> E \\ @V{!}VV @VV{\pi}V \\ * @>>{* \mapsto i}> B \end{CD}$$

and we find that, in the topos of trees, pullbacks over points $* \to B$ can also be described in terms of inverse images.

Proposition 1.16. Let $B \in \mathbf{Psh}\,\omega$ and $f : * \to B$. Let $F \in \mathbf{Psh}\,\omega$ and $p : F \to B$. The pullback that fits in the following picture

$$\begin{array}{ccc} f^*F & \longrightarrow & F \\ {\scriptstyle !}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ * & \xrightarrow[*\mapsto i]{} & B \end{array}$$

can be defined as follows

— $(f^*F)_t \equiv p_t^{-1}(\{f_t\,*\})$

— $(f^*F)_{u\leq t}\,x \equiv F_{u\leq t}\,x$

as we find that, by naturality of p, $F_{u\leq t}$ correctly preserves fibers.

Local elements. The $\mathbb{y}_t$ are close relatives of the terminal tree $*$, and we use them to define local elements of trees. Local elements $\mathbb{y}_t \to A$ can be very simply described in terms of the data of A. This fact is the much–celebrated Yoneda lemma.

Definition 1.17. A *local element* of a tree F is a map $\varphi : \mathbb{y}_t \to F$ for some $t \in \omega$.

Proposition 1.18 (Yoneda Lemma)**.** Let $A \in \mathbf{Psh}\,\omega$. There are the following isomorphisms between sets

$$\forall t \in \omega, \quad (\mathbb{y}_t \longrightarrow A) \simeq A_t$$

1.3 Some Constructors

In this section, we define some more structure that we will need for our final application in section 4.

Natural Numbers. There is a tree $\underline{\mathbb{N}}$ that mimics the properties of $\mathbb{N}$. We will need this for our final application in section 4 but we won't need more than its definition.

Definition 1.19. We define $\underline{\mathbb{N}}$ as follows

— $\underline{\mathbb{N}}_t \equiv \mathbb{N}$

— $\underline{\mathbb{N}}_{u\leq t} n \equiv n$

Remark 1.20. More generally, for any set A, we can define $\underline{A}$ as the constant tree that maps all $t \in \omega$ to A. This construction assembles into a functor, which implies some degree of preservation of the structure of $A \in \mathbf{Set}$ when translated to $\mathbf{Psh}\,\omega$.

Exponentials. Let $X, Y \in \mathbf{Psh}^{\mathrm{s}}\,\omega$. The exponential of X and Y is a tree of which (local) elements correspond to maps between Y and X. We note it X^Y. In general, in a category that has cartesian products, like the topos of trees, the exponentials are defined as the right adjoints of the product. This means that we want the following family of isomorphisms

$$(Z \times Y \longrightarrow X) \quad \simeq \quad (Z \longrightarrow X^Y)$$

These isomorphisms are incarnations of currying and decurrying — the notation "$\cdot \longrightarrow \cdot$" for hom-sets helps highlight this analogy. With this property in mind, and with the Yoneda lemma, let us "guess" what the exponential should be.

$$\begin{aligned} (X^Y)_t &\simeq \mathbb{y}_t \longrightarrow X^Y \\ &\simeq \mathbb{y}_t \times Y \longrightarrow X \end{aligned}$$

A natural transformation of the last type in the equations above is given by a family of functions $(\mathsf{y}_t u \times Y_u \longrightarrow X_u)_{u \in \omega}$, which is equivalently a family $(Y_u \longrightarrow X_u)_{u \leq t}$ by currification and because $\mathsf{y}_t u$ is inhabited if and only if $u \leq t$. This motivates the following definition.

Definition 1.21. The exponential presheaf X^Y is defined as follows

- $(X^Y)_t \equiv Y\big|_{\omega_{\leq t}} \longrightarrow X\big|_{\omega_{\leq t}}$ is the set of natural transformations between the functors restricted to $\omega_{\leq t}$.
- $(X^Y)_{u \leq t}$ is defined by restriction.

There is a natural transformation that applies exponentials to arguments.

Definition 1.22. We define $\mathtt{app} : X^Y \times Y \longrightarrow X$ as follows

- $\mathtt{app}_t : (Y\big|_{\omega_{\leq t}} \to X\big|_{\omega_{\leq t}}) \times Y_t \longrightarrow X_t$ maps (ψ, y) to $\psi_t y$.

2 The Later Operator

The Later operator. Let $F \in \mathbf{Psh}\,\omega$ be a tree. It is instructive to think of F_t, $t \in \omega$ as "the data of F after t time steps". There is a special endomorphism in the topos of trees which acts on trees as a delay operator, letting the data lag one step behind.

Definition 2.1. We define $\blacktriangleright : \mathbf{Psh}^{\mathrm{s}}\,\omega \longrightarrow \mathbf{Psh}^{\mathrm{s}}\,\omega$, pronounced "later". This endofunctor maps a tree F to the tree $\blacktriangleright F$ such that

- $(\blacktriangleright F)_0 \equiv *$
- $(\blacktriangleright F)_{t+1} \equiv F_t$
- $(\blacktriangleright F)_{0 \leq t} \equiv !$
- $(\blacktriangleright F)_{u+1 \leq t+1} \equiv F_{u \leq t}$

This construction is functorial in F.

We can see the "delayed" aspect of $\blacktriangleright F$ with respect to F by observing that, when $t \geq 1$, $(\blacktriangleright F)_t = F_{t-1}$: after t time steps, the data that $\blacktriangleright F$ represents is just the data that F used to represent after only $t-1$ time steps. The equation $(\blacktriangleright F)_0 \equiv *$ materializes the idea that there is no information in $\blacktriangleright F$ at time $t = 0$, since there is no "F_{-1}" to get information from.

For a graphical represention of the "$\blacktriangleright$" operator, notice that it takes a tree F such as on the left below to a tree such as on the right below

$$F_0 \xleftarrow{F_{0\leq 1}} F_1 \xleftarrow{F_{1\leq 2}} F_2 \quad \cdots \qquad \longmapsto \qquad * \xleftarrow{\;!\;} F_0 \xleftarrow{F_{0\leq 1}} F_1 \quad \cdots$$

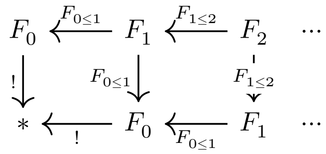
The following construction shows that "if we have some data now, then we can also keep it for later".

Definition 2.2. There is, for all $F \in \mathbf{Psh}^{\mathrm{s}}\,\omega$, a map $\mathtt{next} : F \to \blacktriangleright F$. This map is defined as follows

- $\mathtt{next}_0 : F_0 \to (\blacktriangleright F)_0$ is just the terminal map
- $\mathtt{next}_{t+1} : F_{t+1} \to (\blacktriangleright F)_{t+1}$
  Unfolding the definition of $\blacktriangleright F$, we find that we want a map $F_{t+1} \to F_t$. We take $F_{t \leq t+1}$.

This definition satisfies the naturality condition of natural transformations.

Graphically, this transformation maps F to the following morphism



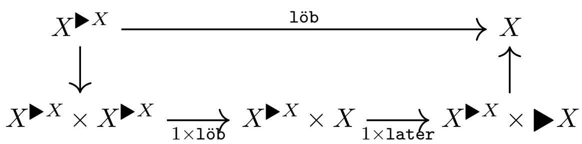
2.1 Löb Induction

Löb and fixed points. In all generality, given a set A, or a A in any other structure, and $f : A \to A$ a function, or any kind of arrow that is relevant to the structure of A, a fixed point of f is $x \in A$ such that $f(x) = x$. An ideal fixed point operator would take any such A and any such f and yield such a x. Of course, such an operator doesn't exist in that much generality. Löb induction provides a general operator that is syntaxically close to this ideal fixpoint operator, up to a constraint on the domain of the function we want to take fixed points of.

Theorem 2.3 (Löb induction)**.** For all $X \in \mathbf{Psh}^{\mathrm{s}}\,\omega$, there is a unique

$$\texttt{löb} : X^{\blacktriangleright X} \longrightarrow X$$

such that the following diagram commutes



where the arrow on the left is duplication, and the arrow on the right is application.

Let us rephrase the diagrammatic statement above in a more familiar language of terms. The arrow on the top of the diagram has to be equal to the composite that passes through the bottom. These arrows are morphisms of trees, hence natural transformations, therefore families of functions indexed by $t \in \omega$. The domains of these functions are the $(X^{\blacktriangleright X})_{t\in\omega}$, which by definition are the $((\blacktriangleright X)\big|_{\omega_{\leq t}} \to X\big|_{\omega_{\leq t}})_{t\in\omega}$. The diagram is then commutative exactly when

$$\forall t \in \omega, \forall f : (\blacktriangleright X)\big|_{\omega_{\leq t}} \to X\big|_{\omega_{\leq t}}, \quad f_t(\texttt{later}_t(\texttt{löb}_t f)) = \texttt{löb}_t f$$

We can further reduce this equation by splitting cases on $t = 0$ and $t > 0$.

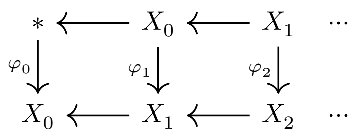
— When $t = 0$ — taking $f : (\blacktriangleright X)\big|_{\omega_{\leq 0}} \to X\big|_{\omega_{\leq 0}}$, we see that actually $f : * \to X_0$, and

$$\texttt{löb}_0 f = f* \tag{2.1}$$

— When $t > 0$ — taking $t \in \omega$ and $f : (\blacktriangleright X)\big|_{\omega_{\leq t+1}} \to X\big|_{\omega_{\leq t+1}}$, we see that

$$\begin{aligned} \texttt{löb}_{t+1} f &= f_{t+1}(\texttt{later}_{t+1}(\texttt{löb}_{t+1} f)) && \text{by definition} \\ &= f_{t+1}(X_{t\leq t+1}(\texttt{löb}_{t+1} f)) && \text{by definition of } \texttt{later} \\ &= f_{t+1}(\texttt{löb}_t(f\big|_{\omega_{\leq t}})) && \text{by naturality of } \texttt{löb} \end{aligned} \tag{2.2}$$

Let us first draw a diagram that helps informally see why there is such an operator. Let $\varphi : \blacktriangleright X \longrightarrow X$. This φ may be represented as follows



and we see that to construct an element of X_0, our only option is to use $\varphi_0 * \in X_0$, then recursively once we have some $x_i \in X_i$ there is only one way to construct an element of X_{i+1} in the diagram.

We now prove theorem 2.3. We split the proof in existence and uniqueness.

Existence of Löb induction. The discussion above motivates the following

Proof. We define $\texttt{löb}\,\varphi$ as follows, recursively over ω

— $(\texttt{löb}\,\varphi)_0 : (X^{\blacktriangleright X})_0 \longrightarrow X_0$
  For $\psi : (\blacktriangleright X)\big|_{\omega_{\leq 0}} \longrightarrow X\big|_{\omega_{\leq 0}}$, we define $(\texttt{löb}\,\varphi)_0 \psi \equiv \psi_0 *$

— $(\texttt{löb}\,\varphi)_{t+1} : (X^{\blacktriangleright X})_{t+1} \longrightarrow X_{t+1}$
  For $\psi : (\blacktriangleright_X)\big|_{\omega_{\leq t+1}} \longrightarrow X\big|_{\omega_{\leq t+1}}$, we define $(\texttt{löb}\,\varphi)_{t+1} \psi \equiv \psi_{t+1}(\texttt{löb}\,\varphi)_t$.
  This is well defined because the domain of ψ_{t+1} is $(\blacktriangleright X)_{t+1}$ which is by definition X_t.

□

Uniqueness of Löb induction. Let $X \in \mathbf{Psh}^{\mathrm{s}}\,\omega$ and $\texttt{fun} : X^{\blacktriangleright X} \longrightarrow X$ be a function that also makes the diagram 2.3 commute, *i.e.* such that

$$\forall t \in \omega, \forall f : (\blacktriangleright X)\big|_{\omega_{\leq t}} \to X\big|_{\omega_{\leq t}}, \quad f(\texttt{later}_t(\texttt{fun}_t f)) = \texttt{fun}_t f$$

We show that $\texttt{fun} = \texttt{löb}$.

Proof. First, notice that eqs. (2.1) and (2.2) can be adapted for $\texttt{fun}$: when f has the right type,

$$\texttt{fun}_0 f = f* \tag{2.3}$$

$$\texttt{fun}_{t+1} f = f_{t+1}(\texttt{fun}_t (f\big|_{\omega_{\leq t}})) \tag{2.4}$$

By function extensionality, it suffices to show that

$$\forall t \in \omega, \forall f : (\blacktriangleright X)\big|_{\omega_{\leq t}} \longrightarrow X\big|_{\omega_{\leq t}}, \quad \texttt{fun}_t f = \texttt{löb}_t f$$

We show the result by induction on $t \in \omega$.

— For $t = 0$, for all f, we want
$$\texttt{fun}_0 f = \texttt{lob}_0 f$$
  which by eqs. (2.1) and (2.3) reduces to $f* = f*$.

— For the inductive step, assume that the result is true for $t \in \omega$. We show that,
$$\forall f : (\blacktriangleright X)\big|_{\omega_{\leq t+1}} \to X\big|_{\omega_{\leq t+1}}, \quad \texttt{fun}_{t+1} f = \texttt{lob}_{t+1} f$$
  which by eqs. (2.2) and (2.4) amounts to
$$f(\texttt{fun}_t (f\big|_{\omega_{\leq t}})) = f(\texttt{löb}_t (f\big|_{\omega_{\leq t}}))$$
  which is true as a consequence of our assumption.

□

3 A Universe in the Topos of Trees

Universes in set theory. It is well-known that trying to reason about a would-be "set of all sets" leads to paradoxes. Grothendieck universes allow to reason about "sets of all small sets", without paradoxes, but at the cost of axiomatizing the existence of such universes. The trick is that although small sets can be collected in a set, this set of small sets isn't small.

Definition 3.1. A *Grothendieck universe* is a set U that such that

(i) *Transitivity.* $\forall u \in U, \forall t \in u, t \in U$
(ii) *Power set.* $\forall u \in U, \mathcal{P}(u) \in U$
(iii) *Empty set.* $\emptyset \in U$
(iv) *Unions.* $\forall I \in U, \forall (u_i)_i \in U^I, \bigcup_{i \in I} u_i \in U$
(v) *Infinity.* $\mathbb{N} \in U$

Definition 3.2. Let U be a Grothendieck universe. A set A is said to be U-small when A is a member of U.

Given a Grothendieck universe U, we may reason about U-small structures too. For instance, a U-small ring is a ring of which carrier set is U-small. This is a consequence of the closure properties of definition 3.1: it can be shown that Grothendieck universes are closed under all the operations of (constructive) set theory. Thanks to this, we find that there is a set of U-small rings. This idea that "small structures may be collected in a set" is crucial in categorically–minded mathematics — think how often one may write about the category of groups and the category of topological spaces, for instance.

Categories of trees. Let us just work with two universes, one for *small* sets and one for (potentially) *large* sets. A set X is *essentially small* when there exists a small A_X in bijection with X, *i.e.* $\exists A_X \in \mathbf{Set}^{\mathrm{s}}, X \simeq A_X$. We will use the following notations to designate categories of sets and sets of sets of various sizes. Using the same notaiton for these two notions is not problematic, as the latter are just the sets of objects of the former.

$\mathbf{Set}^{\mathrm{s}}$	small sets
$\mathbf{Set}^{\mathrm{es}}$	essentially small sets
$\mathbf{Set}$	sets

We also use small and large trees.

Definition 3.3. A tree F is *small* when $\forall t \in \omega, F_t \in \mathbf{Set}^{\mathrm{s}}$.

Definition 3.4. A tree F is *essentially small* when it is isomorphic to a small tree, which is the same as asking for each of the $Ft,\ t \in \omega$, to be essentially small.

We introduce the following notations for categories of trees. We have already defined $\mathbf{Psh}\,\omega$ and we recast this notation to be suitable for our concerns of size. Once again, we use the same notation to designate categories of trees and sets of trees, which is still unproblematic because the latter are the sets of objects of the former.

$\mathbf{Psh}^{\mathrm{s}}\,\omega$	small trees
$\mathbf{Psh}^{\mathrm{es}}\,\omega$	essentially small trees
$\mathbf{Psh}\,\omega$	trees

3.1 Families with Small Fibers

Codes and decoding. We have chosen two Grothendieck universes, one for small sets $\mathbf{Set}^{s}$ and one for (potentially) large sets $\mathbf{Set}$. Let U denote the Grothendieck universe of small sets. There is a function $U \to \mathbf{Set}^{s}$ that maps $X \in U$ to $X \in \mathbf{Set}^{s}$. This tautological function defines a family of sets indexed over U. We turn it into a fibered family over U, by defining $\dot{U}$ the set of pairs $(X \in U, x \in X)$. This comes equiped with a function $p : \dot{U} \to U$, the projection on the first component. We find that for $X \in U$ we have $X \simeq p^{-1}(\{X\})$, so that X fits in the following square

$$\begin{array}{ccc} X & \longrightarrow & \dot{U} \\ \big\downarrow & \lrcorner & \big\downarrow p \\ * & \xrightarrow[* \mapsto X]{} & U \end{array}$$

We take this diagram as the definition of "decoding" a code $* \to U$ for set. We can generalize this idea to codes for families by putting any set on the bottom left of the square in the place of $*$.

The universe of small trees. This diagram can be rewritten in the topos of trees. In section 3.2, we will construct a special tree $\mathcal{U}$ together with a family $\dot{\mathcal{U}} \to \mathcal{U}$ that mimic the properties of U and $(\dot{U}, p)$. First, we define a notion of smallness in the topos of trees.

Definition 3.5. Let $E, B \in \mathbf{Psh}\,\omega$ and $p : E \to B$ be a morphism of trees. p is said to have *small fibers* when for all $\varphi : \mathbb{y}_t \to B$ the pullback $\varphi^* E$ is essentially small.

Notation 3.6. We note $\mathtt{sf}(B)$ the collection of $(X, p : X \to B)$ such that p has small fibers.

Proposition 3.7. Let $E, B \in \mathbf{Psh}\,\omega$. A map $p : E \to B$ has small fibers iff for all $t \in \omega$ and $x \in Bt$, $p_t^{-1}(\{x\})$ is essentially small.

Proof. The reason is that the $p_t^{-1}(\{x\})$ assemble into pullbacks of p along elements $\mathbb{y}_t \to B$. □

We define the following notions of codes for families of trees.

Definition 3.8. Let $A \in \mathbf{Psh}\,\omega$ and $(F, p) \in \mathtt{sf}(A)$. We say that $c : A \to \mathcal{U}$ is a *code* for (F, p) when it is possible to complete the following cartesian square

$$\begin{array}{ccc} F & \overset{\exists}{\dashrightarrow} & \dot{\mathcal{U}} \\ p\big\downarrow & \lrcorner & \big\downarrow p_{\mathcal{U}} \\ A & \xrightarrow[c]{} & \mathcal{U} \end{array}$$

Definition 3.9. Let $A \in \mathbf{Psh}^{s}\,\omega$. We say that $c : * \to \mathcal{U}$ is a code for A when it is a code for $(A, !) \in \mathtt{sf}(*)$.

The property we want to prove. Our goal here will be to construct $\mathcal{U}$ such that every map with small fibers has a code in $\mathcal{U}$. This will be proposition 3.25:

Proposition 3.25. Any map with small fibers $p : E \to B$ arises as a pullback of $p_{\mathcal{U}} : \dot{\mathcal{U}} \to \mathcal{U}$ along a function $\varphi : B \to \mathcal{U}$.

3.2 The Hofmann-Streicher Construction

We now construct the tree $\mathcal{U}$.

Definition 3.10. The tree $\mathcal{U}$ — the $\mathcal{U}$niverse — is defined as follows
- — $\mathcal{U}t \equiv \mathbf{Psh}^{\mathrm{s}}\,\omega_{\leq t}$
- — $\mathcal{U}_{u\leq t} : \mathbf{Psh}^{\mathrm{s}}\,\omega_{\leq t} \longrightarrow \mathbf{Psh}^{\mathrm{s}}\,\omega_{\leq u}$ is restriction

This tree has a canonical map $p : \dot{\mathcal{U}} \to \mathcal{U}$ with small fibers where $\dot{\mathcal{U}}$ where

Definition 3.11. The tree $\dot{\mathcal{U}}$ is defined as follows
- — $\dot{\mathcal{U}}t \equiv \sum_{F\in\mathbf{Psh}^{\mathrm{s}}\,\omega_{\leq t}} Ft$
- — $\dot{\mathcal{U}}_{u\leq t} : \sum_{F\in\mathbf{Psh}^{\mathrm{s}}\,\omega_{\leq t}} Ft \longrightarrow \sum_{F\in\mathbf{Psh}^{\mathrm{u}}\,\omega_{\leq t}} Fu$ is defined on the first component by restriction, and on the second component by using the map $F_{t\leq u}$.

Definition 3.12. The map $p_{\mathcal{U}} : \dot{\mathcal{U}} \longrightarrow \mathcal{U}$ is defined by projecting on the first component of $\mathcal{U}$.

Proposition 3.13. $p_{\mathcal{U}}$ has small fibers.

Proof. This follows from proposition 3.7. Notice that $(p_{\mathcal{U}})_t^{-1}(\{F\}) = F_t$ which is small. □

The category of elements. We need a new technical tool, the category of elements, to prove our proposition 3.25.

Definition 3.14. Let F be a tree. The category of elements of F, noted $\int F$, is the category where
- — objects are pairs (t, x) where $t \in \omega$ and $x \in Ft$
- — there is a morphism $(t, x) \to (u, y)$ when $t \leq u$ and $F_{t\leq u} y = x$.

Remark 3.15. The category of elements can alternatively be defined as the comma category ${}^{\mathbb{y}}\!\!\bullet\!\!\diagup_F$ thanks to the Yoneda lemma — an equivalent description that justifies the name "category of elements".

Our proof will proceed in two step. First, we will show that small families over a base B may be equivalently be seen as small presheaves over the category of elements $\int B$. Then, we will show that presheaves over $\int B$ may equivalently be seen as maps $B \to \mathcal{U}$. Our result will follow by composing these two equivalences.

Construction 3.16. Let B be a tree. Let $X \in \mathbf{Psh}^{\mathrm{s}} \int B$. The tree $\mathtt{tot}\,X$ is defined as follows
- — $(\mathtt{tot}\,X)t \equiv \sum_{x\in Bt} X_{t,x}$
- — $(\mathtt{tot}\,X)u \leq t : \sum_{x\in Bt} X_{t,x} \longrightarrow \sum_{x\in Bu} X_{u,x}$
  is defined using the functions $B_{u\leq t}$ and $X_{(u,B_{u\leq t}x)\leq(t,x)}$.

Construction 3.17. Let B be a tree. Let $X \in \mathbf{Psh}^{\mathrm{s}} \int B$. The map $\mathtt{dis}\,X : \mathtt{tot}\,X \to B$ is defined as follows
- — $(\mathtt{dis}\,X)_t : \sum_{x\in Bt} X_{t,x} \to Bt$ is simply the projection on the first component.

Construction 3.18. Let B be a tree. Let $p : E \to B$ have small fibers. The presheaf $\mathtt{toElm}\,p$ is defined as follows
- — Ideally, $(\mathtt{toElm}\,p)(t, x) \equiv p_t^{-1}(\{x\})$ however it is only *essentially* small according to proposition 3.7. By essential smallness, there is a $A_{p_t^{-1}(\{x\})} \in \mathbf{Set}^{\mathrm{s}}$ that is in bijection with $p_t^{-1}(\{x\})$. We define $(\mathtt{toElm}\,p)(t, x) \equiv p_t^{-1}(\{x\})$.

— $(\texttt{toElm}\, p)_{(u,y)\le(t,x)} : A_{p_t^{-1}(\{x\})} \longrightarrow A_{p_u^{-1}(\{y\})}$
comes from the map $B_{t\le u}$ which correctly maps the x-fiber of p_t to the y-fiber of p_u by consequence definition of the morphisms of the category of elements.

Remark 3.19. If $E, B \in \mathbf{Psh}^{\mathrm{s}}\,\omega$, then we can define $(\texttt{toElm}\, p)(t,x) \equiv p_t^{-1}(\{x\})$ directly since this set is automatically small.

Proposition 3.20. A map $p : E \to B$ has small fibers iff it is isomorphic to $\texttt{dis}\, X : \texttt{tot}\, X \to B$ for some $X \in \mathbf{Psh}^{\mathrm{s}} \int B$.

Proof. First, $\texttt{dis}\, X$ always has small fibers. Using proposition 3.7, it suffices to notice that $(\texttt{dis}\, X)_t^{-1}(\{x\}) = X_{t,x}$ and that this last set is small by definition.

Second, when $p : E \to B$ has small fibers, we can consider the presheaf $\texttt{toElm}\, p$ (Construction 3.18), and we find that p is isomorphic to $\texttt{dis}\,(\texttt{toElm}\, p)$. ☐

Constructions 3.16 and 3.17 together with Construction 3.18 assemble into a correspondance between fibered trees over B and presheaves over the category of elements of B. Although this correspondance isn't a bijection, Proposition 3.20 means that $X \mapsto \texttt{dis}\, X$ is essentially surjective. For reference, we collect our constructions in the following picture:

$$\begin{array}{ccc} \texttt{sf}(B) & \longleftrightarrow & \mathbf{Psh}^{\mathrm{s}} \int B \\ (X,p) & \longmapsto & (c,x) \mapsto (A \text{ small st. } A \simeq p_c^{-1}(x)) \\ (c \mapsto \{(x,y) \mid y \in X(c,x)\}) & \longleftarrow\!\!\shortmid & X \end{array}$$

Construction 3.21. Let $B \in \mathbf{Psh}\,\omega$ and $X \in \mathbf{Psh}^{\mathrm{s}} \int B$. The map $\texttt{toIdx}\, X : B \to \mathcal{U}$ is defined as follows
— $(\texttt{toIdx}\, X)_t x \equiv (u \le t) \mapsto X(u, B_{u\le t} x)$

Construction 3.22. Let $B \in \mathbf{Psh}\,\omega$ and $\varphi : B \to \mathcal{U}$. The presheaf $\texttt{toElm}\, \varphi \in \mathbf{Psh}^{\mathrm{s}} \int B$ is defined as follows
— $(\texttt{toElm}\, \varphi)(t,x) \equiv (\varphi_t x)t$
— $(\texttt{toElm}\, \varphi)_{(u,y)\le(t,x)} : (\varphi_t x)t \to (\varphi_u y)u$ requires some work.
Notice first that, by definition of the morphisms of $\int B$, we actually need a function $(\varphi_t x)t \to (\varphi_u (B_{u\le t} x))u$.
Then, by naturality, this is just a function $(\varphi_t x)t \to (\varphi_t x)u$ and this means that — recall that $\varphi_t x \in \mathbf{Psh}^{\mathrm{s}}\,\omega_{\le t}$ — we can define $(\texttt{toElm}\, \varphi)_{(u,y)\le(t,x)} \equiv (\varphi_t x)_{u\le t}$.

Proposition 3.23. These two maps are mutual inverses.

Proof. First, see that

$$\begin{array}{rcll} (\texttt{toIdx}(\texttt{toElm}\, \varphi))_t x & = & (u \le t) \mapsto (\varphi_u (B_{u\le t}))u & \text{by definition} \\ & = & (u \le t) \mapsto (\varphi_t x)u & \text{by naturality of } \varphi \\ & = & \varphi_t x & \end{array}$$

Then, see that

$$\begin{array}{rcll} (\texttt{toElm}(\texttt{toIdx}\, X))(t,x) & = & ((u \le t) \mapsto X(u, B_{u\le t} x))t & \text{by definition} \\ & = & X(t,x) & \text{by reducing the expression} \end{array}$$

☐

This means that we actually have an isomorphism of sets

$$\begin{array}{ccc} \mathbf{Psh}^{\mathrm{s}} \int B & \simeq & (B \to \mathcal{U}) \\ X & \longmapsto & (x \mapsto (u \le t) \mapsto X(u, B_{u \le t} x))_{t \in \omega} \\ (t, x) \mapsto (\varphi_t x) t & \longleftarrow & \varphi \end{array}$$

Proposition 3.24. The map

$$\begin{array}{ccc} (B \to \mathcal{U}) & \longrightarrow & sf(B) \\ \varphi & \longmapsto & (\mathtt{tot}(\mathtt{toElm}\,\varphi), \mathtt{dis}(\mathtt{toElm}\,\varphi)) \end{array}$$

is a pullback map

$$\begin{array}{ccc} (B \to \mathcal{U}) & \longrightarrow & sf(B) \\ \varphi & \longmapsto & (\varphi^* \dot{\mathcal{U}}, \varphi^* p_{\mathcal{U}}) \end{array}$$

Proposition 3.25. Any map with small fibers $p : E \to B$ arises as a pullback of $p_{\mathcal{U}} : \dot{\mathcal{U}} \to \mathcal{U}$ along a function $\varphi : B \to \mathcal{U}$.

Proof. This follows from the explicit description of pullbacks given in proposition 3.24 together with propositions 3.20 and 3.23 that prove that (E, p) is always isomorphic to this explicit pullback. $\square$

4 Application: Streams

We have gathered enough knowledge about the topos of trees to actually construct a nontrivial example that is relevant to computer science, namely, streams. Our notion of streams will be a formal model of the idea of a sequence of natural numbers that are fed into a system. This notion is time-dependent, and trees are a natural structure with which to reason about time-dependent data and programs.

4.1 Constructors for Families

Products. We have defined products in the category of trees in Construction 1.7. Our construction is easily seen to be functorial in its two arguments. Taking advantage of this functoriality, we now generalize our construction to families. We will see that this generalization is reasonable as, when applied to families over the point $*$, it boils down to our original definition of products.

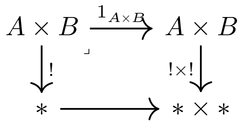
Definition 4.1. Let $A \in \mathbf{Psh}\,\omega$. Let $E, F \in \mathbf{Psh}\,\omega$ and $p : E \to A$, $q : F \to A$. The *product* of (E, p) and (F, q) is the family that fits in the following diagram

$$\begin{array}{ccc} E \times_A F & \longrightarrow & E \times F \\ {\scriptstyle p \times_A q}\downarrow & \lrcorner & \downarrow{\scriptstyle p \times q} \\ A & \xrightarrow[\Delta]{} & A \times A \end{array}$$

where $\Delta : A \to A \times A$ is the diagonal, $\Delta x \equiv (x, x)$.

Proposition 4.2. Let $A, B \in \mathbf{Psh}^{\mathrm{s}}\,\omega$. We see these trees as families over the point $*$. The product $A \times_* B$ following definition 4.1 is isomorphic to the product $A \times B$.

Proof. It suffices to show that $A \times B$ fits in the same diagram as $A \times_* B$ in definition 4.1, *i.e.* that there is the following cartesian square



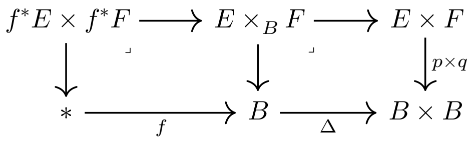
This square is seen to be cartesian because $* \times * \simeq *$. $\square$

More generally, the data of $E \times_B F$ over the global points of B is recovered from the data of E and F over the global points of B. Precisely, we show the following proposition.

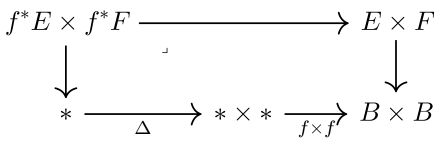
Proposition 4.3. Let $B \in \mathbf{Psh}\,\omega$. Let $E, F \in \mathbf{Psh}\,\omega$ and $p : E \to A$, $q : F \to A$. Let $f : * \to B$. We have the following isomorphism

$$f^*(F \times_B G) \simeq f^*F \times f^*G$$

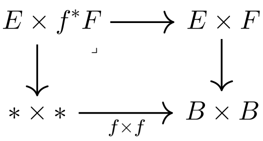
Proof. It suffices to show that $f^*E \times f^*F$ fits in the diagram of definition 4.1. By pullback pasting, it suffices to show that it fits in the outer cartesian square of the following diagram



The bottom composite can be rewritten $\Delta \circ f = (f \times f) \circ \Delta$. Therefore it suffices to fit $f^*E \times f^*F$ in the following cartesian square



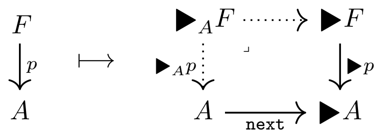
and, because $* \times * \simeq *$, the bottom left arrow is an isomorphism so it suffices to fit $f^*E \times f^*F$ in the following



and this last cartesian square holds by applying the functor $\cdot \times \cdot$ to the two cartesian squares that define f^*E and f^*F. $\square$

Later operator on families. We extends the definition 2.1 to families.

Definition 4.4. Let $A \in \mathbf{Psh}\,\omega$. We define $\blacktriangleright : \mathtt{sf}(A) \longrightarrow \mathtt{sf}(A)$. This endofunctor maps a tree (F, p) to $(\blacktriangleright_A F, \blacktriangleright_A p)$ defined by pulling back as in the following diagram



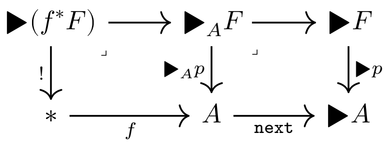
where the arrow on the right hand side of the square comes from definition 2.1.

We can check that this definition of "$\blacktriangleright_A$" makes sense by checking that it behaves correctly above points

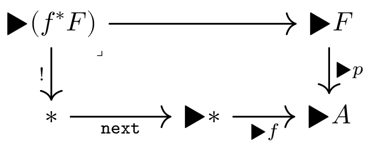
Proposition 4.5. Let $B \in \mathbf{Psh}\,\omega$. Let $(F, p) \in \mathtt{sf}(B)$. Let $f : * \to B$ a global point of B. The following isomorphism holds

$$f^*(\blacktriangleright_B F) \quad \simeq \quad \blacktriangleright(f^* F)$$

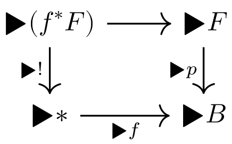
Proof. It suffices to show that $\blacktriangleright(f^*F)$ fits in the outer cartesian square of the following diagram



By naturality of $\mathtt{next}$, we have $\mathtt{next} \circ f = (\blacktriangleright f) \circ \mathtt{next}$. Therefore it suffices to show that $\blacktriangleright(f^*F)$ fits in the following diagram



but $\blacktriangleright * \simeq *$, so we just need to fit $\blacktriangleright(f^*F)$ in the following diagram



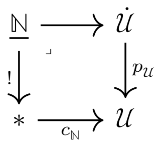
and this last cartesian square is shown to hold by applying $\blacktriangleright$ to the square that defines f^*F. $\square$

4.2 Codes for Families

We make use of our work in section 3 to show that the universe $\mathcal{U}$ has codes for products (Construction 1.7 and definition 4.1), the later operator (definitions 2.1 and 4.4) and natural numbers (definition 1.19). Our arguments hinge on proposition 3.25: to show that an object or a family have a code, it suffices to show that the object is small or the family has small fibers. Since this property is very easy to check, constructing codes is a straightforward process.

Natural numbers. There is a code in $\mathcal{U}$ for $\underline{\mathbb{N}}$.

Construction 4.6. As a consequence of proposition 3.25, there is a code $c_{\mathbb{N}} : * \to \mathcal{U}$ of $\mathbb{N}$ that fits in the following diagram



Products. The universe $\mathcal{U}$ has codes for products.

Construction 4.7. There is a map $\texttt{prod} : \mathcal{U} \times \mathcal{U} \to \mathcal{U}$ which fits in the diagram below

$$\begin{array}{ccc} \dot{\mathcal{U}} \times \dot{\mathcal{U}} & \longrightarrow & \dot{\mathcal{U}} \\ {\scriptstyle p_{\mathcal{U}} \times p_{\mathcal{U}}}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle p_{\mathcal{U}}} \\ \mathcal{U} \times \mathcal{U} & \xrightarrow[\texttt{prod}]{} & \mathcal{U} \end{array}$$

This map is the code of $p_{\mathcal{U}} \times p_{\mathcal{U}}$. This follows from proposition 3.25 (notice that $p_{\mathcal{U}} \times p_{\mathcal{U}}$ has small fibers, as $p_{\mathcal{U}}$ has small fibers).

Proposition 4.8. Let $A \in \mathbf{Psh}\,\omega$ and $(E, p), (F, q) \in \texttt{sf}(A)$. Let $c_1 : A \to \mathcal{U}$ be a code for (E, p), and c_2 be a code for (F, q). The following map

$$A \xrightarrow[\Delta]{} A \times A \xrightarrow[c_1 \times c_2]{} \mathcal{U} \times \mathcal{U} \xrightarrow[\texttt{prod}]{} \mathcal{U}$$

is a code for the family $(E \times_A F, p \times_A p) \in \texttt{sf}(A)$.

Proof.

$$\begin{array}{ccccccc} E \times_A F & \longrightarrow & E \times F & \longrightarrow & \dot{\mathcal{U}} \times \dot{\mathcal{U}} & \longrightarrow & \dot{\mathcal{U}} \\ \big\downarrow & \lrcorner & \big\downarrow & \lrcorner & \big\downarrow & \lrcorner & \big\downarrow \\ A & \xrightarrow[\Delta]{} & A \times A & \xrightarrow[c_1 \times c_2]{} & \mathcal{U} \times \mathcal{U} & \xrightarrow[\texttt{prod}]{} & \mathcal{U} \end{array}$$

We have the pullback squares above, where the left square comes from definition 4.1 and is simply the definition of $E \times_A F$, the middle pullback square is definition 3.8, the fact that c_1 and c_2 are codes, viewed through $\cdot \times \cdot$, which preserves pullbacks, and the right pullback square is proposition 3.25. By pullback pasting, the outer rectangle is cartesian too. □

Later. The universe $\mathcal{U}$ internalizes the operator $\blacktriangleright\cdot$.

Construction 4.9. There is a map $\triangleright : \mathcal{U} \to \dot{\mathcal{U}}$ which fits in the diagram below

$$\begin{array}{ccc} \blacktriangleright\dot{\mathcal{U}} & \longrightarrow & \dot{\mathcal{U}} \\ {\scriptstyle \blacktriangleright p_{\mathcal{U}}}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle p_{\mathcal{U}}} \\ \blacktriangleright\mathcal{U} & \xrightarrow[\triangleright]{} & \mathcal{U} \end{array}$$

This map is the code of $\blacktriangleright p_{\mathcal{U}}$. This follows from proposition 3.25 (notice that $\blacktriangleright p_{\mathcal{U}}$ has small fibers, as $p_{\mathcal{U}}$ has small fibers).

Proposition 4.10. The universe contains codes for the *later* operator on families. Let $A \in \mathbf{Psh}\,\omega$ and $(E, p) \in \texttt{sf}(A)$. Let $c : A \to \mathcal{U}$ be a code for (E, p). The map

$$A \xrightarrow{\texttt{next}} \blacktriangleright A \xrightarrow{\blacktriangleright c} \blacktriangleright\mathcal{U} \xrightarrow{\triangleright} \mathcal{U}$$

is a code for $\blacktriangleright(E, p)$

Proof.

$$
\begin{array}{ccccccc}
\blacktriangleright_A E & \longrightarrow & \blacktriangleright E & \longrightarrow & \blacktriangleright \dot{\mathcal{U}} & \longrightarrow & \dot{\mathcal{U}} \\
\big\downarrow & \lrcorner & \big\downarrow & \lrcorner & \big\downarrow & \lrcorner & \big\downarrow \\
A & \xrightarrow[\texttt{next}]{} & \blacktriangleright A & \xrightarrow[\blacktriangleright c]{} & \blacktriangleright \mathcal{U} & \xrightarrow[\triangleright]{} & \mathcal{U}
\end{array}
\tag{4.1}
$$

We have the pullback squares above, where the left pullback square comes from definition 4.4 and is simply the definition of $\blacktriangleright_A E$, the middle pullback square is definition 3.8, the fact that c is a code, viewed through $\blacktriangleright$, which preserves pullbacks, and the right pullback square is proposition 3.25. By pullback pasting, the outer rectangle is cartesian too. □

4.3 Defining Streams

We construct a solution to the recursive domain equation that we stated in the introduction. We define $\texttt{Stream} : * \longrightarrow \mathcal{U}$ such that

$$
\begin{array}{ccccccc}
* & & & \xrightarrow{\texttt{Stream}} & & & U \\
\big\downarrow & & & & & & \big\uparrow \\
* \times * & \xrightarrow[c_{\mathbb{N}} \times \texttt{Stream}]{} & U \times U & \xrightarrow[1 \times \texttt{later}]{} & U \times \blacktriangleright U & \xrightarrow[1 \times \triangleright]{} & U \times U
\end{array}
$$

where the arrow on the right is our encoding of products (Construction 4.7). In other words,

$$\forall t \in \omega, \quad \texttt{Stream}_t * \simeq \mathbb{N} \times \triangleright_t(\texttt{later}_t(\texttt{Stream}_t *))$$

which reduces to $\texttt{Stream}_0 * \simeq \mathbb{N}$ and for all $t \in \omega$, $\texttt{Stream}_{t+1} * \simeq \mathbb{N} \times \texttt{Stream}_t$, so that

$$\texttt{Stream}_t * \simeq \mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N} = \mathbb{N}^{t+1}$$

It suffices to define $\texttt{Stream}$ by löb induction on the element of $\mathcal{U}^{\blacktriangleright \mathcal{U}}$ corresponding to the following composite

$$\blacktriangleright \mathcal{U} \xrightarrow{\triangleright} \mathcal{U} \xrightarrow{c_{\mathbb{N}} \times 1} \mathcal{U} \times \mathcal{U} \xrightarrow{\texttt{prod}} \mathcal{U}$$

The map $\texttt{Stream} : * \to \mathcal{U}$ is a code for the tree $\texttt{Stream}^* \dot{\mathcal{U}}$. By construction, this tree is a solution to the equation that was stated in the introduction. We can compute its local points (up to isomorphism) using the Yoneda lemma, proposition 1.16 and the characterization above of $\texttt{Stream}_t *$. Its global points are a colimit of its local points.

$$
\begin{array}{lll}
\text{local points} & (\mathbb{y}_t \longrightarrow \texttt{Stream}^* \mathcal{U}) & \simeq \quad \mathbb{N}^{t+1} \\
\text{global points} & (* \longrightarrow \texttt{Stream}^* \mathcal{U}) & \simeq \quad \mathbb{N}^{\omega}
\end{array}
$$

These characterizations reflect the intuitive behavior of streams. A stream, observed with the knowledge of all its extent through time, is a sequence of numbers. On the other hand, if we can only observe the same stream with the knowledge of its contents after t time steps, we find that the data it has yielded is excactly t numbers.

This concludes this note. In the context of the theory of Iris, we have isolated the problem of solving recursive domain equations. We have stated a recursive domain equation, that of streams, we have stated the context in which we meant to solve this equation, and we have developed tools to construct a solution. In doing this, we have illustrated the problem of solving recursive domain equations, and worked out the details of how to do so by using Löb induction on a universe.

References

Pierre America and Jan J. M. M. Rutten. 1989. Solving Reflexive Domain Equations in a Category of Complete Metric Spaces. *J. Comput. Syst. Sci.* 39, 3 (1989), 343–375. doi:10.1016/0022-0000(89)90027-5

Michael Artin, Alexander Grothendieck, and Jean-Louis Verdier. 1971. *Theorie de Topos et Cohomologie Etale des Schemas I, II, III* (2015 reedition ed.). Chapter I, Préfaisceaux. https://www.normalesup.org/~forgogozo/SGA4/01/01.pdf

Jean Bénabou. 1973. Problemes dans les topos: d'apres le cours de questions speciales de mathematique. (1973).

Lars Birkedal and Rasmus Ejlers Møgelberg. 2013. Intensional Type Theory with Guarded Recursive Types qua Fixed Points on Universes. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 213–222. doi:10.1109/LICS.2013.27

Lars Birkedal, Rasmus Ejlers Møgelberg, Jan Schwinghammer, and Kristian Støvring. 2012. First steps in synthetic guarded domain theory: step-indexing in the topos of trees. *Log. Methods Comput. Sci.* 8, 4 (2012). doi:10.2168/LMCS-8(4:1)2012

Lars Birkedal, Kristian Støvring, and Jacob Thamsborg. 2010. The category-theoretic solution of recursive metric-space equations. *Theor. Comput. Sci.* 411, 47 (2010), 4102–4122. doi:10.1016/J.TCS.2010.07.010

Pietro Di Gianantonio and Marino Miculan. 2002. A Unifying Approach to Recursive and Corecursive Definitions. In *Types for Proofs and Programs, Second International Workshop, TYPES 2002, Berg en Dal, The Netherlands, April 24-28, 2002, Selected Papers (Lecture Notes in Computer Science, Vol. 2646)*, Herman Geuvers and Freek Wiedijk (Eds.). Springer, 148–161. doi:10.1007/3-540-39185-1_9

Ralf Jung, Robbert Krebbers, Jacques-Henri Jourdan, Ales Bizjak, Lars Birkedal, and Derek Dreyer. 2018. Iris from the ground up: A modular foundation for higher-order concurrent separation logic. *J. Funct. Program.* 28 (2018), e20. doi:10.1017/S0956796818000151

Hiroshi Nakano. 2000. A Modality for Recursion. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*. IEEE Computer Society, 255–266. doi:10.1109/LICS.2000.855774

Thomas Streicher. 2005. Universes in Toposes. In *From sets and types to topology and analysis - Towards practicable foundations for constructive mathematics*, Laura Crosilla and Peter M. Schuster (Eds.). Oxford logic guides, Vol. 48. Oxford University Press.