Coinduction in Flow: The Later Modality in Fibrations

Henning Basold
CNRS, ENS de Lyon, France
LIACS – Leiden University, The Netherlands
h.basold@liacs.leidenuniv.nl

Abstract

This paper provides a construction on fibrations that gives access to the so-called later modality, which allows for a controlled form of recursion in coinductive proofs and programs. The construction is essentially a generalisation of the topos of trees from the codomain fibration over sets to arbitrary fibrations. As a result, we obtain a framework that allows the addition of a recursion principle for coinduction to rather arbitrary logics and programming languages. The main interest of using recursion is that it allows one to write proofs and programs in a goal-oriented fashion. This enables easily understandable coinductive proofs and programs, and fosters automatic proof search.

Part of the framework are also various results that enable a wide range of applications: transportation of (co)limits, exponentials, fibred adjunctions and first-order connectives from the initial fibration to the one constructed through the framework. This means that the framework extends any first-order logic with the later modality. Moreover, we obtain soundness and completeness results, and can use up-to techniques as proof rules. Since the construction works for a wide variety of fibrations, we will be able to use the recursion offered by the later modality in various context. For instance, we will show how recursive proofs can be obtained for arbitrary (syntactic) first-order logics, for coinductive set-predicates, and for the probabilistic modal μ-calculus. Finally, we use the same construction to obtain a novel language for probabilistic productive coinductive programming. These examples demonstrate the flexibility of the framework and its accompanying results.

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases Coinduction, Fibrations, Later Modality, Recursive Proofs, Up-to techniques, Probabilistic Logic, Probabilistic Programming

Digital Object Identifier 10.4230/LIPIcs.CALCO.2019.8

Related Version `https://arxiv.org/abs/1802.07143`

Funding *Henning Basold*: This work has been funded by the European Research Council (ERC) under the EU's Horizon 2020 programme (CoVeCe, grant agreement No 678157), and was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

1 Introduction

Recursion is one of the most fundamental notions in computer science and mathematics, be it as the foundation of computability, or to define and reason about structures determined by repeated constructions. In this paper, we will focus on the use of recursion as a method for coinductive proofs and coinductive programming.

Usually, coinductive programming is presented by means of coiteration schemes and coinduction as bisimulation proof principle. Coiteration schemes are a syntactic implementation of coalgebras and their coinductive extension to a homomorphism into the final coalgebra [32, 48]. The bisimulation proof principle, on the other hand, asserts that bisimilarity implies equality in the final coalgebra [29, 36, 60]. There are, however, also different approaches that break with this dogma. In coinductive programming, guarded recursion [5, 6, 16, 50, 52], and sets of recursive equations [1, 33, 61] have been used to construct elements of final coalgebras and of coinductive types. On the side of proofs and semantics, several improvements of coinduction

8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019).
Editors: Markus Roggenbach and Ana Sokolova; Article No. 8; pp. 8:1–8:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

have been suggested: simplification of invariants [63] via up-to techniques [19, 54, 58] and the companion [11, 55, 56], incremental techniques [38, 51], games [53, 65], and basic cyclic proofs for stream equality [57]. In this paper, we will focus on guarded recursion because it can be widely applied, and because it leads to clean proof and programming methods.

A concrete appearance of coinduction can be found in the modal μ-calculus Lμ [46, 20] and its quantitative interpretations [39] pLμ or Łμ [49] in form of Park's rule, which assert that if $\psi \to \varphi[\psi/X]$, then $\psi \to \nu X.\, \varphi$. This rule says that an implication with a greatest fixed point as conclusion can be proven by showing that ψ is an *invariant* for φ. Kozen [46] gave an axiomatisation of Lμ based on this rule, and its dual, that turned out to be complete [75]. Thus, this axiomatisation is expressive, but often difficult to use in practice, let alone for proof search. It should be noted that Lμ is decidable if it is interpreted in classical logic. The goal of this work is, however, to develop techniques that can also be used to obtain (constructive) proof objects and can be applied to more general logics. Thus, our focus will be on improving the axiomatisation of Lμ and of coinductive proofs in general.

Coming back to Park's rule, we often find ourselves having to prove $\psi \to \nu X.\, \varphi$ for a formula ψ, which is not an invariant. We are then required to find an invariant ψ', such that, $\psi \to \psi'$. Finding such an invariant can be difficult in general and it does not fit common practice. Instead, it would be preferable if we could incrementally construct the proof for $\psi \to \nu X.\, \varphi$ rather than guessing an invariant ψ'. Such an incremental construction leads to a recursive proof methodology for coinductive proofs. As such incremental methods are valuable in any theory that is based on coiteration or coinduction schemes, we set out in this paper to replace invariant guessing by a general iterative programming and proof method.

The proposed iterative method will be given in form of a framework that introduces recursion into coinductive proofs and programs, while preserving soundness and termination. This framework is centred around the so-called later modality [52], which allows for us to control the use of recursion and thereby avoid the introduction of non-termination and inconsistencies. The later modality has been successfully used in the context of semantics [16, 72], programming [5, 6, 50], and reasoning [23, 14]. Ultimately, we generalise the work of Birkedal et al. [16] on the topos of trees to arbitrary fibrations with the effect of much wider applicability to, for example, quantitative reasoning and probabilistic programming.

In the case of Lμ, we extend the logic with the later modality as a new logical connective. Given a formula φ, we thus obtain a formula $\blacktriangleright \varphi$. This formula should be read as "later φ", which allows us to formulate that knowledge varies over time. The later modality comes with three crucial axioms: $\varphi \to \blacktriangleright \varphi$ (next), $\blacktriangleright(\varphi \to \psi) \to \blacktriangleright \varphi \to \blacktriangleright \psi$ (monotonicity), and $(\blacktriangleright \varphi \to \varphi) \to \varphi$ (fixed point or Löb). It is the Löb rule that introduces recursion into the logic, and it should be read as "if we can prove φ from the assumption that φ holds later, then φ holds at any time". However, the assumption $\blacktriangleright \varphi$ introduced by the Löb rule cannot be used directly. We need one final axiom for that: $\varphi[\blacktriangleright \nu X.\, \varphi/X] \to \nu X.\, \varphi$ (step). These axioms can be combined to obtain recursive proofs, as we will show later. As an appetiser, the reader may have a look already at Figure 3 on Page 15.

The reader may have noticed that the first three axioms, next, monotonicity and Löb, are independent of the logic at hand. Only the step axiom makes use of the structure of formulas. This observation is what enables the topos of trees and the framework presented here to work. More precisely, we will start with a given fibration $p \colon \mathbf{E} \to \mathbf{B}$ and construct a new fibration $\overleftarrow{p} \colon \overleftarrow{\mathbf{E}} \to \overleftarrow{\mathbf{B}}$ out of it. This fibration will have, under mild conditions, the later modality as a map of fibrations $(\triangleright, \blacktriangleright)$ on it. The next and Löb axioms correspond then to certain morphisms in $\overleftarrow{\mathbf{E}}$, while monotonicity says that $\blacktriangleright$ is a strong functor. From a logical perspective, it is more natural to consider another fibration $\overline{p} \colon \overline{\mathbf{E}} \to \mathbf{B}$ over the same base category as the initial fibration. In this fibration, we will not only have access to the later modality and its axioms, but also to quantifiers that are present in the original fibration p.

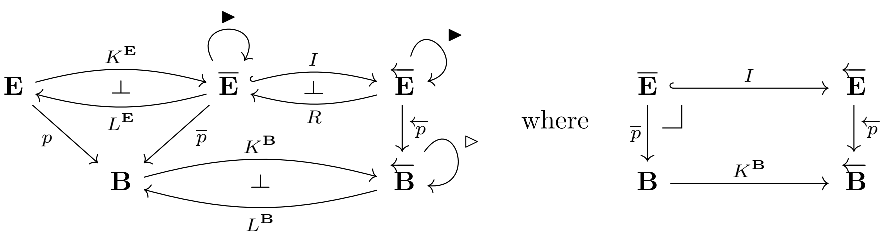

Figure 1 Relation between p (base logic), $\overleftarrow{p}$ (all chains in p) and $\overline{p}$ (chains with constant index).

Contributions. Apart from the applications to the probabilistic modal μ-calculus and to probabilistic programming, the technical contributions of this paper are as follows. Given a fibration p and a well-ordered class $\mathbf{I}$, we let $\overleftarrow{\mathbf{E}}$ be the category of $\mathbf{I}^{\mathrm{op}}$-indexed chains in $\mathbf{E}$, that is, functors $\sigma\colon \mathbf{I}^{\mathrm{op}} \to \mathbf{E}$. The fibration $\overleftarrow{p}$ is given by post-composition with p and thus maps a chain to the chain of its indices given by p. On this fibration, we construct the later modality and find all its good properties. We then restrict our attention to the fibration $\overline{p}\colon \overline{\mathbf{E}} \to \mathbf{B}$, which consists only of chains with constant index. In other words, $\overline{p}$ is given by the change-of-base (pullback) along the functor $K^{\mathbf{B}}\colon \mathbf{B} \to \overleftarrow{\mathbf{B}}$ that maps $I \in \mathbf{B}$ to the chain that is equal to I at every position. This is indicated in the right diagram in Figure 1. The diagram on the left summarises the most important ingredients of the framework:

- the later modality is a map of fibrations $\blacktriangleright\colon \overline{p} \to \overline{p}$ and $(\triangleright, \blacktriangleright)\colon \overleftarrow{p} \to \overleftarrow{p}$ with a natural transformation $\mathrm{next}\colon \mathrm{Id} \Rightarrow \blacktriangleright$ (Theorem 16 and Theorem 17);
- $\overleftarrow{p}$ and $\overline{p}$ are fibred Cartesian closed categories and feature the Löb rule as morphism $\text{löb}_{\sigma}\colon \sigma^{\blacktriangleright \sigma} \to \sigma$ that fulfils a unique solution condition (Theorem 20 and Theorem 27);
- fixed points of so-called locally contractive functors on $\overleftarrow{p}$ and $\overline{p}$ (Theorem 31);
- the final chain construction of final coalgebras via a locally contractive functor (Theorem 34) and up-to techniques as proof rules (Theorem 35);
- if $\mathbf{B}$ has $\mathbf{I}^{\mathrm{op}}$-limits, then there is an adjunction $K^{\mathbf{B}} \dashv L^{\mathbf{B}}$ between $\mathbf{B}$ and $\overleftarrow{\mathbf{B}}$, and an adjunction $I \dashv R$ between $\overline{\mathbf{E}}$ and $\overleftarrow{\mathbf{E}}$ (Theorem 19);
- fibred $\mathbf{I}^{\mathrm{op}}$-limits in p give a fibred adjunction $K^{\mathbf{E}} \dashv L^{\mathbf{E}}$ between $\mathbf{E}$ and $\overline{\mathbf{E}}$ (Theorem 19);
- if p is a first-order fibration, then $\overline{p}$ is a first-order fibration and $L^{\mathbf{E}}$ preserves truth of first-order formulas if disjunction, existentials and equality preserve $\mathbf{I}^{\mathrm{op}}$-limits (Theorem 41).

Particularly interesting is that $\overline{p}$ is a first-order fibration, in other words, models first-order logic. This result can be restricted to any subset of connectives, which allows us to extend any logic with the later modality and its axioms. The adjunction between p and $\overline{p}$ shows then that this yields a sound and complete axiomatisation of coinductive predicates. We leverage this generality to devise a novel proof system for the probabilistic modal μ-calculus and a language for productive probabilistic programming with coinductive types.

Another interesting aspect of the diagram is that one of the central results used by Hasuo et al. [35] (Lem. 3.5) appears here as the composition $L^{\mathbf{E}} \circ R\colon \overleftarrow{\mathbf{E}} \to \mathbf{E}$. In fact, the results in [35] tell us under which conditions we can use the finite ordinals ω as index $\mathbf{I}$ to obtain a sound and complete proof system for coinductive predicates.

Organisation. The framework is introduced in the following steps. First, we provide in Section 2 a brief overview over fibrations, coinductive predicates and well-founded induction. Next, we describe in Section 3 the chain fibrations $\overleftarrow{p}$ and $\overline{p}$, construct the later modality and give some basic results. Section 4 is devoted to show that functor fibrations are fibred Cartesian closed and to the Löb rule. In Section 5 we construct fixed points of so-called locally contractive functors, both, on the whole fibration and on the fibres. Moreover, we show how the final chain arises as locally contractive functor, and how this leads to the proof rule "step" that we saw above. This allows us also to obtain proof rules on the final chain for compatible up-to techniques. As promised, we prove in Section 6 that $\overline{p}$ is a first-order fibration. Furthermore, we give the adjunctions from Figure 1 that relate the various fibrations. The flexibility of the framework is then demonstrated by providing a recursive proof system for probabilistic Lμ and a language for guarded recursive probabilistic programming in Section 7. We conclude with a few remarks and future work in Section 8.

Related Work. To a large part, the present paper generalises the work of Birkedal et al. [16] from the codomain fibration $\mathbf{Set}^{\to} \to \mathbf{Set}$ of sets to arbitrary fibrations. That [16] was so restrictive is not so surprising, as the intention there was to construct models of programming languages, rather than applying the developed techniques to proofs. Going beyond the category of sets also means that one has to involve much more complicated machinery to obtain exponential objects, see Section 4. Later, Bizjak et al. [17] extended the techniques from [16] to dependent type theory, thereby enabling reasoning by means of recursive proofs in a syntactic type theory. However, also this is a very specific setting, which rules out the main examples that we are interested in here. Similarly, also the parameterised coinduction in categories [51] and in lattices [38] is too restrictive, as they only apply to, respectively, propositional and to set-theoretic settings. It might be possible to develop parameterised coinduction in the setting of fibrations by using the companion [11, 55, 56], but we leave this question for another time. Recursion is also central to cyclic proof systems [21, 24, 26, 64]. These are particularly useful in settings that require proofs by induction or coinduction because cyclic proof systems ease proofs enormously compared to the invariant-based method of (co)induction schemes. Nothing comes for free though: In this case checking proofs becomes more difficult, as the correctness conditions are typically global for a proof tree and not compositional. For the same reason, also soundness proofs are often rather complex. The framework we study here gives rise to proof rules that require no further global condition on proofs, which straightforwardly yields proof checking [8] and soundness. Higher-order recursion has also been studied in other categorical settings like topos theory [47, 40] or monoidal categories [30, 34]. Unfortunately, these neither apply to our examples of interest, nor do they provide the logical results and constructions that appear in this paper.

Finally, in the realm of algorithmic proofs, circular proofs have been used to automatically prove identities of streams [57]. Else, computer-supported coinduction is usually limited to proof checking [31, 18, 25]. There have been limited approaches to combine coinduction with resolution [66]. In [10], we were able to go beyond the state of the art by extending uniform proofs to coinduction and using the framework presented in this paper as logical foundation. This shows that the framework of this paper paves the way for algorithmic proof search.

2 Preliminaries

We begin by recalling the terminology of fibrations, coalgebras, coinductive predicates, and well-founded induction. Moreover, we discuss examples that underlie the motivating applications of this paper.

2.1 Fibrations

One of the central notions used in this paper are fibrations [13, 41, 71], as they are an elegant way of capturing (typed) variables in a (higher-order) predicate logic.

▶ **Definition 1.** *Let* $p : \mathbf{E} \to \mathbf{B}$ *be a functor, where* $\mathbf{E}$ *is called the* total *category and* $\mathbf{B}$ *the* base *category. A morphism* $f : A \to B$ *in* $\mathbf{E}$ *is said to be* Cartesian *over* $u : I \to J$*, provided that i)* $pf = u$*, and ii) for all* $g : C \to B$ *in* $\mathbf{E}$ *and* $v : pC \to I$ *with* $pg = u \circ v$ *there is a unique* $h : C \to A$ *such that* $f \circ h = g$*. For* p *to be a* fibration*, we require that for every* $B \in \mathbf{E}$ *and* $u : I \to pB$ *in* $\mathbf{B}$*, there is a cartesian morphism* $f : A \to B$ *over* u*. Finally, a fibration is* cloven*, if it comes with a unique choice for* A *and* f*, in which case we denote* A *by* $u^* B$ *and* f *by* $\overline{u}\, B$*, as displayed in the diagram in Figure 2.* ⌟

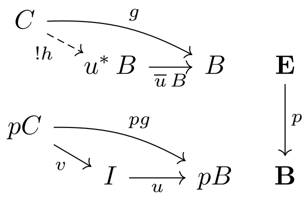

Figure 2 Cartesian Lifting in a Fibration p.

On cloven fibrations, we can define for each $u : I \to J$ in $\mathbf{B}$ a functor, the *reindexing along* u, as follows. Let us denote by $\mathbf{E}_I$ the category having objects A with $p(A) = I$ and morphisms $f : A \to B$ with $p(f) = \mathrm{id}_I$. We call $\mathbf{E}_I$ the *fibre above* I and the morphisms in $\mathbf{E}_I$ *vertical*. The assignment of $u^* B$ to B for a cloven fibration can then be extended to a functor $u^* : \mathbf{E}_J \to \mathbf{E}_I$. Moreover, one can show that there are natural isomorphisms $\mathrm{id}_I^* \cong \mathrm{Id}_{\mathbf{E}_I}$ and $(v \circ u)^* \cong u^* \circ v^*$ subject to some coherence conditions.

For a fibration $p\colon \mathbf{E} \to \mathbf{B}$ and a functor $F\colon \mathbf{C} \to \mathbf{B}$, we can form a new fibration $F^*(p)\colon\ F^*(\mathbf{E}) \to \mathbf{C}$ by pulling p back along F, see [41]. The fibration $F^*(p)$ is said to be obtained by *change-of-base*. Given another fibration $q\colon \mathbf{D} \to \mathbf{A}$, a *map of fibrations* $p \to q$ is a pair (F, G) of functors $F\colon \mathbf{A} \to \mathbf{B}$ and $G\colon \mathbf{D} \to \mathbf{E}$, with $p \circ G = F \circ q$ and such that G preserves Cartesian morphisms. This means in particular for $u\colon I \to J$ and $A \in \mathbf{E}_J$ that for $G(u^* A) \cong (Fu)^*(GA)$. Finally, the fibration p is said to have *fibred* $\Diamond$ (certain limits, colimits, exponentials, etc.), if every fibre has $\Diamond$ and reindexing preserves these.

Let $\mathbf{C}$ be a Cartesian closed category. We denote for $f\colon Y \to X$ by $\ulcorner f \urcorner\colon \mathbf{1} \to X^Y$ the code of f. Recall [45] that a functor $F\colon \mathbf{C} \to \mathbf{C}$ is *strong* if there is natural family of morphisms $\mathrm{st}^F_{X,Y}\colon X^Y \to FX^{FY}$, s.t. $\mathrm{st}^F_{X,Y} \circ \ulcorner f \urcorner = \ulcorner Ff \urcorner$. A map of fibrations $(F, G)\colon p \to p$ is strong if both F and G are strong, and $p\,\mathrm{st}^G = \mathrm{st}^F$.

As the definition of fibrations and the associated notions are fairly abstract, let us give a few examples. There are five examples that we shall use to illustrate different aspects of the theory: predicates over sets, quantitative predicates, syntactic logic, the codomain fibration over the category of sets, and categories as trivial fibrations. The codomain fibration will allow us to recover later the topos of trees. We begin with the simplest example, namely that of predicates. Despite its simplicity, it is already quite useful because it allows us to reason about predicates and relations for arbitrary coalgebras in **Set**.

▶ **Example 2** (Predicates). The fibration $\mathrm{Pred} \to \mathbf{Set}$ of predicates has as objects in its total category Pred predicates $(P \subseteq X)$ over a set X. Each fibre Pred_X has a final object $\mathbf{1}_X = (X \subseteq X)$ and the fibred binary products are given by intersection. We note that fibred constructions, like the above products, are preserved by a change-of-base, see [41, Lem. 1.8.4]. Hence, one can also apply the results in this paper to, for example, the fibration of (binary) relations $\mathrm{Rel} \to \mathbf{Set}$, which is given by pulling $\mathrm{Pred} \to \mathbf{Set}$ back along the diagonal functor $\delta\colon \mathbf{Set} \to \mathbf{Set}$ with $\delta(I) = I \times I$. ⌟

Often, one is not just interested in merely logical predicates, but rather wants to analyse quantitative aspects of system. Such predicates will be the foundation for the probabilistic μ-calculus. The following example extends the predicate fibration from Example 2 to quantitative predicates, which will give us a convenient setting to reason about quantitative properties.

▶ **Example 3.** We define the category of quantitative predicates qPred as follows.

$$\text{qPred} = \begin{cases} \text{objects:} & \text{pairs } (X, \delta) \text{ with } X \in \mathbf{Set} \text{ and } \delta \colon X \to [0,1] \\ \text{morphisms:} & f \colon (X,\delta) \to (Y,\gamma) \text{ if } f \colon X \to Y \text{ in } \mathbf{Set} \text{ and } \delta \leq \gamma \circ f \end{cases}$$

It is easy to show that the first projection qPred $\to$ **Set** gives rise to a cloven fibration, for which the reindexing functors are given for $u \colon X \to Y$ by $u^*(Y, \gamma) = \big(X, \lambda x.\, \gamma(u(x))\big)$. For brevity, let us refer to an object (X, δ) in qPred_X just by its underlying valuation δ. One readily checks that in qPred fibred products can be defined by $(\delta \times \gamma)(x) = \min\{\delta(x), \gamma(x)\}$ and coproducts as maximum. Fibred final objects are given by the constantly 1 valuation. ⌟

The original motivation for the work presented in this paper was to abstract away from the details that are involved in constructing a syntactic logic for a certain coinductive relation in [9]. In [9], the author developed a first-order logic that features the later modality to reason about program equivalences. This logic was given in a very pedestrian way, since the syntax, proof system and models were constructed from scratch. The proofs often involved phrases along the lines of “true because this is an index-wise interpretation of intuitionistic logic”. In the following example, we show how a first-order logic can be presented as a fibration, which allows us to apply the framework to a syntactic logic.

▶ **Example 4** (Syntactic Logic). Suppose we are given a typed calculus, for example the simply typed λ-calculus or even the category **Set** of sets, and a first-order logic, in which the variables range over the types of the calculus. More precisely, let Γ be a context with $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, where the x_i are variables and the A_i are types of the calculus. We write then $\Gamma \Vdash t : A$ if t is a term of type A in context Γ, $\Gamma \Vdash \varphi$ if φ is formula with variables in Γ, and $\Gamma \vdash \varphi$ if φ is provable in the given logic. This allows us to form a fibration as follows. First, we define $\mathcal{C}$ to be the *syntactic category* that has context Γ as objects and tuples t of terms as morphisms $\Delta \to \Gamma$ with $\Delta \Vdash t_i : A_i$. Next, we let $\mathcal{L}$ be the category that has pairs (Γ, φ) with $\Gamma \Vdash \varphi$ as objects, and a morphism $(\Delta, \psi) \to (\Gamma, \varphi)$ in $\mathcal{L}$ is given by a morphism $t \colon \Delta \to \Gamma$ in $\mathcal{C}$ if $\Delta \vdash \psi \to \varphi[t]$, where $\varphi[t]$ denotes the substitution of t in the formula φ. The functor $p \colon \mathcal{L} \to \mathcal{C}$ that maps (Γ, φ) to Γ is then easily seen to be a cloven fibration, see for example [41]. Let us assume that the logic also features a truth formula $\top$, conjunction $\wedge$ and implication $\to$, which are subject to the usual proof rules of intuitionistic logic. We note that p has fibred finite products given by $\top$ and conjunction. ⌟

As promised, the setup of Birkedal et al. [16], the topos of trees, can be recovered as an instance of our framework.

▶ **Example 5** (Codomain Fibration). Let $\mathbf{Set}^{\to}$ be the arrow category over the category of sets and functions. This category has maps as objects and commuting squares as morphisms. The functor cod: $\mathbf{Set}^{\to} \to \mathbf{Set}$ that sends a map to its codomain is a fibration, in which reindexing is given by taking pullbacks, see [41]. ⌟

The final example will allow us to apply the framework of this paper to any category.

▶ **Example 6** (Trivial Fibration). Let $\mathbf{1}$ be the final category with one object $*$ and only the identity on $*$. Then any category $\mathbf{C}$ can be seen as fibration $! \colon \mathbf{C} \to \mathbf{1}$, such that fibred products etc. are normal products. ⌟

2.2 Coalgebras and Coinductive Predicates

Let us now introduce the second central notion of this paper: coinductive predicates. For that, we first need the notion of coalgebra [2, 42, 43, 60, 62].

▶ **Definition 7.** *Let $F\colon \mathbf{C} \to \mathbf{C}$ be a functor. A* coalgebra *is a morphism $c\colon X \to FX$. Given coalgebras $c : X \to FX$ and $d : Y \to FY$, a* homomorphism *from c to d is a morphism $h : X \to Y$ with $Fh \circ c = d \circ h$. We can form a category* $\mathrm{CoAlg}(F)$ *of coalgebras and their homomorphisms and we call a final object in this category a* final coalgebra. ⌟

Coinductive predicates are easiest introduced by taking for a moment a more abstract perspective. Recall that we introduced fibrations as a way to talk abstractly about predicates, relations etc. Now we use this view to define coinductive predicates over a given coalgebra for an arbitrary notion of predicate.

▶ **Definition 8.** *Let $p\colon \mathbf{E} \to \mathbf{B}$ be a cloven fibration and $F\colon \mathbf{B} \to \mathbf{B}$ an endofunctor. We say that a functor $G\colon \mathbf{E} \to \mathbf{E}$ is a* lifting *of F, if $p \circ G = F \circ p$. A G-*invariant *in a coalgebra $c\colon X \to FX$ in $\mathbf{B}$ is a $(c^* \circ G)$-coalgebra in $\mathbf{E}_X$. Further, a G-*coinductive predicate *in c is a final $(c^* \circ G)$-coalgebra. We often denote the carrier of the G-coinductive predicate in c by $\nu(c^* \circ G)$, see [35]. A* compatible up-to technique *for $c^* \circ G$ is a functor $T\colon \mathbf{E} \to \mathbf{E}$ with a natural transformation $T \circ c^* \circ G \Rightarrow c^* \circ G \circ T$, see [19, 59].* ⌟

Let us illustrate the notion of coinductive predicate in an example.

▶ **Example 9.** In this example, we show how the semantics of the modalities of the probabilistic modal μ-calculus (pLμ) can be modelled as liftings. Given a set X, we say that a function $\rho\colon X \to [0,1]$ to the unit interval is a (finitely supported) probability distributions on X, if the support $\operatorname{supp} \rho = \{x \mid \rho(x) \neq 0\}$ is finite and $\sum_{x \in \operatorname{supp}(\rho)} \rho(x) = 1$. One can then define a functor $\mathcal{D}\colon \mathbf{Set} \to \mathbf{Set}$ that maps a set to the set of all probability distributions on X. An (unlabelled) Segala system [69] or probabilistic transition system (PTS) is a coalgebra for the functor $\mathcal{S}$ given by $\mathcal{S} = \mathcal{P} \circ \mathcal{D}$, in which states have non-deterministic transitions into probability distributions. We can now give liftings $\mathcal{S}^{\Box}$ and $\mathcal{S}^{\Diamond}$ of $\mathcal{S}$ to qPred, which correspond to the box and diamond modality, respectively, of pLμ:

$$\mathcal{S}^{\Box}(\delta\colon X \to [0,1])(D \in \mathcal{S}(X)) = \bigwedge_{d \in D} \sum_{x \in \operatorname{supp} d} \delta(x) \cdot d(x)$$
$$\mathcal{S}^{\Diamond}(\delta\colon X \to [0,1])(D \in \mathcal{S}(X)) = \bigvee_{d \in D} \sum_{x \in \operatorname{supp} d} \delta(x) \cdot d(x)$$

Suppose now that we have a PTS $c\colon X \to \mathcal{S}(X)$ at hand, then $c^* \circ \mathcal{S}^{\Box}\colon \mathrm{qPred}_X \to \mathrm{qPred}_X$ yields the expected semantics of the box modality [49]. ⌟

2.3 Well-Founded Induction

The final basic ingredient of this paper is well-founded induction. We will use a rather general form that is based on classes, rather than sets.

▶ **Definition 10.** *Let A be a class and $<$ a binary relation on A. We say that the relation $<$ is* well-founded, *if the* well-founded induction principles *holds for all $P \subseteq A$: If for all $\alpha \in A$ we have that $(\forall \beta < \alpha.\, \beta \in P) \implies \alpha \in P$, then $\forall \alpha \in A.\, \alpha \in P$.*

Given a well-founded order, we can form as usual a category from the induced partial order $\leq$ with $\alpha \leq \beta$ if $\alpha < \beta$ or $\alpha = \beta$. Typical examples, to which the presented framework applies, are the set ω of finite ordinals with the successor relation; the set of ordinals below any limit ordinal with their usual order; and the class of all ordinals Ord.

Recall that ordinals can be constructed as zero, successor and limit ordinals. We say that $\mathbf{I}$ is a *classical ordinal category*[1], if every $\alpha \in \mathbf{I}$ is either zero, a successor or a limit.

3 Descending Chains in Fibrations

It is well-known that a final coalgebra of a functor F, hence also coinductive predicates, can be constructed as limits of α^{op}-chains for some limit ordinal α if such limits exist and are preserved by F [2, 3, 7]. This observation is essential to the proof approach given in this paper, as we rely on the fact that maps into a coinductive predicate, thus proofs, can alternatively be given as maps into this α^{op}-chain. In the following, we introduce the necessary machinery to leverage this fact.

More specifically, we build from a given fibration a new fibration of descending chains. In this fibration, we will be able to construct the final chain as a fixed point of a certain functor on descending chains, see Section 5.1. It should be noted though that the fibration of descending chains allows the construction of fixed points of many more functor, so called locally contractive functors. Thus, the reasoning power of the built fibration extends beyond coinductive predicates as fixed points of, e.g., contravariant functors do also exist, cf. [16, 15].

The fibration of descending chains will then admit recursive proofs for coinductive predicates and will also feature all propositional connectives and quantifiers that are present in the fibration that we started with, see Section 6. This allows us to extend any (higher-order) logic with recursive proofs for coinductive predicates.

3.1 Categories of Diagrams

Before we analyse the final chain of a functor, we introduce general diagrams and establish properties of these. We fix an index category $\mathbf{I}$ and let $[\mathbf{I}, \mathbf{C}]$ for a category $\mathbf{C}$ be the category of functors from $\mathbf{I}$ to $\mathbf{C}$, also called the category of $\mathbf{I}$-*indexed diagrams in* $\mathbf{C}$. Given a functor $F\colon \mathbf{C} \to \mathbf{D}$, we define a functor $[\mathbf{I}, F] : [\mathbf{I}, \mathbf{C}] \to [\mathbf{I}, \mathbf{D}]$ on categories of diagrams by $[\mathbf{I}, F](\sigma) = F \circ \sigma$. Since $[\mathbf{I}, -]$ preserves composition of functors and applies to natural transformations, we obtain a strict 2-functor $[\mathbf{I}, -] : \mathbf{Cat} \to \mathbf{Cat}$. We use this to define for a morphism $f\colon X \to Y$ in $\mathbf{C}$, a morphism $[\mathbf{I}, f] : K_X \Rightarrow K_Y$ in $[\mathbf{I}, \mathbf{C}]$ where K_X is the constant functor sending any object in $\mathbf{I}$ to X: Note that there is a natural transformation $K_f : K_X \Rightarrow K_Y$, which is given by $K_{f,i} = f$. Thus, we can put $[\mathbf{I}, f] = [\mathbf{I}, K_f]$.

The assignment of diagrams and lifting functors not only preserves 2-structure, but also fibrational structure.

▶ **Lemma 11.** *The functor* $[\mathbf{I}, -]$ *extends to an endomap of the fibration* $\mathbf{Fib} \to \mathbf{Cat}$.

Also adjunctions are preserved in the transition to diagrams.

▶ **Lemma 12.** *If* $F\colon \mathbf{C} \to \mathbf{D}$ *and* $G\colon \mathbf{D} \to \mathbf{C}$ *with* $F \dashv G$, *then* $[\mathbf{I}, F] \dashv [\mathbf{I}, G]$.

3.2 Descending Chains and the Later Modality

In this section, we extend the development in [16] to fibrations. We will give some intuition for the later modality and prove some basic results.

[1] We use the term "classical" here because in classical set theory, as opposed to constructive set theory, every ordinal is given in this way.

▶ **Assumption 13.** In the remainder of the paper, we assume that the category $\mathbf{I}$ is induced by a well-founded class I.

In the construction of final coalgebras, one considers $\mathbf{I}^{\mathrm{op}}$-indexed diagrams, which give rise to a functor $\mathbf{Cat} \to \mathbf{Cat}$ with

$$\overleftarrow{(-)} = [\mathbf{I}^{\mathrm{op}}, -]\,, \tag{1}$$

as in the previous section. The *category of descending chains in* $\mathbf{C}$ is then the category $\overleftarrow{\mathbf{C}}$, the objects of which we denote by $\sigma, \tau, \ldots$ More explicitly, $\sigma \in \overleftarrow{\mathbf{C}}$ assigns as a functor $\sigma \colon \mathbf{I}^{\mathrm{op}} \to \mathbf{C}$ to each $\alpha \in I$ an object $\sigma_\alpha \in \mathbf{C}$, and to each pair α and β with $\beta \le \alpha$ a morphism $\sigma(\beta \le \alpha) \colon \sigma_\alpha \to \sigma_\beta$ in $\mathbf{C}$. A morphism in $\overleftarrow{\mathbf{C}}$ is a natural transformation $f \colon \sigma \Rightarrow \tau$, in other words, a family of morphisms $f_\alpha \colon \sigma_\alpha \to \tau_\alpha$ with $\tau(\beta \le \alpha) \circ f_\alpha = f_\beta \circ \sigma(\beta \le \alpha)$.

From Lemma 11 we get that the functor $\overleftarrow{p} \colon \overleftarrow{\mathbf{E}} \to \overleftarrow{\mathbf{B}}$, given by post-composition, is a fibration. The reindexing functors in this fibration will we denoted by $u^{\#}$. Since (co)limits are constructed point-wise in functor categories, the fibration $\overleftarrow{p}$ inherits (co)limits from p. We obtain another fibration by change-of-base along the constant chain functor $K^{\mathbf{B}} \colon \mathbf{B} \to \overleftarrow{\mathbf{B}}$ that sends an object $I \in \mathbf{B}$ to the constant chain $K^{\mathbf{B}}_I \colon \mathbf{I}^{\mathrm{op}} \to \mathbf{B}$ as in the diagram on the right in Figure 1.

We note the following result, which allows us to apply, for example, Lemma 12 and Theorem 27 to functors between fibres.

▶ **Lemma 14.** *We have that* $\overline{\mathbf{E}}_I \cong \overleftarrow{\mathbf{E}_I} \cong \overleftarrow{\mathbf{E}}_{K^{\mathbf{B}}(I)}$.

Many constructions in this paper require only limits over a bounded part of $\mathbf{I}^{\mathrm{op}}$.

▶ **Definition 15.** *Let* $\mathbf{J}$ *be a category and denote for* $i \in \mathbf{J}$ *by* $i \downarrow \mathbf{J}$ *the coslice category under* i. *We say that* $\mathbf{C}$ *has* bounded $\mathbf{J}$-limits, *if for every* $i \in \mathbf{J}$ *all* $(i \downarrow \mathbf{J})$*-limits exist in* $\mathbf{C}$.

As an example, we have for $\mathbf{I} = \omega$ and $n \in \mathbb{N}$ that $n \downarrow \omega^{\mathrm{op}} = (\omega/n)^{\mathrm{op}} = \underline{n}^{\mathrm{op}}$, where $\underline{n}$ is the set of all $k \le n$. Hence, $n \downarrow \omega^{\mathrm{op}}$ is finite and bounded ω^{op}-limits are finite limits.

With this definition, we can now introduce the later modality, which is the central construction that underlies the recursive proofs that we develop in this paper.

▶ **Theorem 16.** *Suppose that* p *has fibred bounded* $\mathbf{I}^{\mathrm{op}}$*-limits. There are functors* $\rhd \colon \overleftarrow{\mathbf{B}} \to \overleftarrow{\mathbf{B}}$ *and* $\blacktriangleright \colon \overleftarrow{\mathbf{E}} \to \overleftarrow{\mathbf{E}}$ *given on objects by*

$$(\rhd c)_\alpha = \lim_{\beta < \alpha} c_\alpha \quad \textit{and} \quad (\blacktriangleright \sigma)_\alpha = \lim_{\beta < \alpha} \sigma_\alpha,$$

together with natural transformations $\mathrm{next}^{\rhd} \colon \mathrm{Id} \Rightarrow \rhd$ *and* $\mathrm{next} \colon \mathrm{Id} \Rightarrow \blacktriangleright$. *The pair* $(\rhd, \blacktriangleright)$ *forms a map of fibrations* $\overleftarrow{p} \to \overleftarrow{p}$ *and we have* $\overleftarrow{p}(\mathrm{next}) = \mathrm{next}^{\rhd}$. *Moreover,* $\blacktriangleright$ *preserves fibred finite limits. Finally, if* $\mathbf{I}$ *is a classical ordinal category, then* $\blacktriangleright$ *has a left-adjoint* $\blacktriangleleft$.

We note that because $\blacktriangleright \colon \overleftarrow{\mathbf{E}} \to \overleftarrow{\mathbf{E}}$ maps $\sigma \in \overleftarrow{\mathbf{E}}_c$ to $\blacktriangleright \sigma \in \overleftarrow{\mathbf{E}}_{\rhd c}$, we can define a restricted version $\blacktriangleright^c \colon \overleftarrow{\mathbf{E}}_c \to \overleftarrow{\mathbf{E}}_c$ of the later modality that leaves the index chain untouched by putting

$$\blacktriangleright^c = (\mathrm{next}^{\rhd}_c)^{\#} \circ \blacktriangleright .$$

Moreover, there is a vertical natural transformation $\mathrm{next}^c \colon \mathrm{Id} \Rightarrow \blacktriangleright^c$, and $\blacktriangleright^c$ has a left-adjoint if $\mathbf{I}$ is classical and if p is a bifibration.

Another special case is obtained for the chains with constant index.

▶ **Theorem 17.** *The later modality is a strong fibred functor* $\blacktriangleright \colon \overline{p} \to \overline{p}$ *with a vertical natural transformation* $\mathrm{next} \colon \mathrm{Id} \Rightarrow \blacktriangleright$*, that is,* $\overline{p}(\mathrm{next}) = \mathrm{id}$.

Since the intention is to use Theorem 17 to extend a logic, let us present the results as proof rules. The first rule is given by the strength of $\blacktriangleright$, the second rule is given by composition with next, and the last rule for product preservation comes from the isomorphism in Theorem 16. This last rule can be applied in both directions, hence the indicated by double lines.

$$\frac{}{\mathrm{mon}_{\sigma,\tau} \colon \sigma^{\tau} \to \blacktriangleright \sigma^{\blacktriangleright \tau}} \qquad \frac{f \colon \tau \to \sigma}{\mathrm{next}_{\sigma} \circ f \colon \tau \to \blacktriangleright \sigma} \qquad \frac{f \colon \tau \to (\blacktriangleright \sigma) \times (\blacktriangleright \sigma')}{\check{f} \colon \tau \to \blacktriangleright(\sigma \times \sigma')}$$

The following assumption ensures that the above proof rules are available throughout the remainder of this paper.

▶ **Assumption 18.** p is cloven with fibred finite limits and fibred bounded $\mathbf{I}^{\mathrm{op}}$-limits.

So far, we have established the fibrations and the later modality in the overview diagram in Figure 1. What remains are the adjunctions that relate the fibrations, cf. [41, Exercise 1.8.8].

▶ **Theorem 19.** *If $\mathbf{E}$ has fibred $\mathbf{I}^{\mathrm{op}}$-limits, then $K^{\mathbf{E}} \colon \mathbf{E} \to \overline{\mathbf{E}}$ has a fibred right adjoint $L^{\mathbf{E}}$, given by $L^{\mathbf{E}}(\sigma) = \lim_{\alpha \in \mathbf{I}} \sigma_\alpha$. If $\mathbf{B}$ has $\mathbf{I}^{\mathrm{op}}$-limits, then $K^{\mathbf{B}} \colon \mathbf{B} \to \overleftarrow{\mathbf{B}}$ and $I \colon \overline{\mathbf{E}} \to \overleftarrow{\mathbf{E}}$ have right adjoints $L^{\mathbf{B}}$ and R, given by $L^{\mathbf{B}}(c) = \lim_{\alpha \in \mathbf{I}} c_\alpha$ and $R = \pi^{\#}$, where $\pi_\beta \colon \lim_{\alpha \in \mathbf{I}} c_\alpha \to c_\beta$ are the limit projections and $(-)^{\#}$ is reindexing in $\overleftarrow{p}$.*

4 Cartesian Closure and the Löb Rule

Up to this point, we have only shown the existence of the next and monotonicity rule that we used in the example in the introduction. What is missing is the recursion given in form of the Löb rule. The goal of this section is to establish the recursion mechanism by utilising so-called Löb induction, which is based on the later modality that we introduced in Section 3.2. To state and prove the Löb induction, we need exponential objects in our fibration $\overleftarrow{p} \colon \overleftarrow{\mathbf{E}} \to \overleftarrow{\mathbf{B}}$ of chains. In the first part of this section, we show how to construct these from exponential objects in $p \colon \mathbf{E} \to \mathbf{B}$. The second part is the devoted to establishing the Löb rule.

4.1 Fibred Cartesian Closure of Diagrams

A *fibred Cartesian closed category (fibred CCC)* is a fibration $p \colon \mathbf{E} \to \mathbf{B}$ in which every fibre is Cartesian closed and reindexing preserves this structure, see [41, Def. 1.8.2]. In a fibred CCC we can model in particular implication, which is what we will need to formulate the Löb rule below. Given a fibred CCC, we show now that the fibration of diagrams is also a fibred CCC. Since the construction of exponential objects in categories of diagrams does not depend on working with a well-founded index category, we will formulate the results in this section for an arbitrary index category $\mathbf{I}$, like we did in Section 3.1.

Let $S \colon \mathbf{I}^{\mathrm{op}} \times \mathbf{I} \to \mathbf{C}$ be a functor. The *end* of S is an object $\int_{i \in \mathbf{I}} S(i,i)$ in $\mathbf{C}$ together with a universal extranatural transformation $\pi \colon \int_{i \in \mathbf{I}} S(i,i) \to S$. This means that π is a family of morphisms indexed by objects in $\mathbf{I}$, such that the diagram below commutes for all $u \colon i \to j$.

$$\begin{array}{ccc} \int_{i \in \mathbf{I}} S(i,i) & \xrightarrow{\pi_j} & S(j,j) \\ \downarrow{\scriptstyle \pi_i} & & \downarrow{\scriptstyle S(u,\mathrm{id})} \\ S(i,i) & \xrightarrow{S(\mathrm{id},u)} & S(i,j) \end{array}$$

Given another extranatural transformation $\alpha \colon X \to S$ there is a unique $f \colon X \to \int_{i \in \mathbf{I}} S(i,i)$ with $\pi_i \circ f = \alpha_i$ for every $i \in \mathbf{I}$. It is well-known that ends can be computed as certain limits in $\mathbf{C}$. By analysing carefully the necessary limits, we obtain the following result.

▶ **Theorem 20.** *Let* $\mathbf{I}$ *be a category and* $p\colon \mathbf{E} \to \mathbf{B}$ *a cloven fibration that has fibred equalisers, fibred exponents and fibred bounded* $\mathbf{I}$*-products. Then* $[\mathbf{I}, p] : [\mathbf{I}, \mathbf{E}] \to [\mathbf{I}, \mathbf{B}]$ *is again a fibred CCC. The exponential object of* $F, G \in [\mathbf{I}, \mathbf{E}]_U$ *is given by*

$$\left(G^F\right)(i) = \int_{v\colon i \to j} \left(U(v)^* \, G(j)\right)^{U(v)^* \, F(j)}.$$

▶ **Assumption 21.** In the remainder we additionally assume that $p\colon \mathbf{E} \to \mathbf{B}$ is a fibred CCC, has fibred equalisers and fibred bounded $\mathbf{I}$-products.

From Assumption 21, we get that $\overleftarrow{p}$ is a fibred CCC. Since change-of-base also preserves fibred exponentials, the fibration $\overline{p}$ that we obtained by pulling $\overleftarrow{p}$ back along the constant chain functor in Section 3.2 is also a fibred CCC, see [41, Lem. 1.8.4] and [71].

▶ **Example 22.** Fibred exponentials exist in Pred_X with $Q^P = \{x \in X \mid x \in P \implies x \in Q\}$. The fibration $\overleftarrow{\mathrm{Pred}}$ consists then of descending chains of predicates. This means that if $\sigma \in \overleftarrow{\mathrm{Pred}}_X$, then σ is a chain with $\sigma_0 \supseteq \sigma_1 \supseteq \cdots$. Note now that each fibre Pred_X is a complete lattice, hence equalisers are trivial and (bounded) limits are just given as (bounded) infima. Hence, Theorem 20 applies and we obtain that $\overleftarrow{\mathrm{Pred}}$ is a fibred CCC. Since equalisers are trivial, it is easy to see that the exponential for $\sigma, \tau \in \overline{\mathrm{Pred}}_X$ can be defined as follows.

$$(\tau^\sigma)_n = \bigcap\nolimits_{m \le n} \tau_m^{\sigma_m}$$

Since fibred exponentials are preserved by a change-of-base, see [41, Lem. 1.8.4], they also exist in the fibration of relations $\mathrm{Rel} \to \mathbf{Set}$ and the associated fibration $\overleftarrow{\mathrm{Rel}} \to \overleftarrow{\mathbf{Set}}$. ⌟

▶ **Example 23.** Recall that we defined in Example 3 a category of quantitative predicates. We note that this fibration is a fibred CCC with exponents given by

$$(\delta \Rightarrow \gamma)(x) = \begin{cases} 1, & \delta(x) \le \gamma(x) \\ \gamma(x), & \text{otherwise} \end{cases}.$$

Again, each fibre qPred_X is a complete lattice and so $\overleftarrow{\mathrm{qPred}}$ is a fibred CCC for any $\mathbf{I}$. ⌟

▶ **Example 24.** In Example 4, we defined a fibration $p\colon \mathcal{L} \to \mathcal{C}$ for a first-order logic with conjunction and implication. From the implication we obtain that p is a fibred CCC. Moreover, since each fibre is a pre-order, equalisers are again trivial. If $\mathbf{I}$ is the poset ω of finite ordinals, then $\overline{p}$ is a fibred CCC. Explicitly, for chains φ, ψ of formulas in $\overline{p}_A$ above a type A, the exponent $\psi \Rightarrow \varphi$ in $\overleftarrow{p}$ is given by a finite conjunction:

$$(\psi \Rightarrow \varphi)_n = \bigwedge_{m \le n} \psi_m \to \varphi_m.$$

⌟

▶ **Example 25.** The trivial fibration is a fibred CCC if and only if $\mathbf{C}$ is Cartesian closed. In this case, the end formula reduces to $\left(G^F\right)(i) = \int_{v\colon j \to i} G(j)^{F(j)}$ for $G, F\colon \mathbf{I}^{\mathrm{op}} \to \mathbf{C}$. ⌟

4.2 The Löb Rule

One purpose of the later modality is that it allows us to characterise maps in $\overleftarrow{p}$, so-called contractive maps, of which we can construct fixed points.

▶ **Definition 26.** *A map* $f\colon \tau \times \sigma \to \sigma$ *in* $\overleftarrow{\mathbf{E}}_c$ *is called* g-contractive *if* $g\colon \tau \times \blacktriangleright^c \sigma \to \sigma$ *with* $f = g \circ (\mathrm{id} \times \mathrm{next}_\sigma)$*. We call* $s\colon \tau \to \sigma$ *a* fixed point *or* solution *for* f*, if* $s = f \circ \langle \mathrm{id}, s \rangle$*.* ⌟

We can now show that there is a operator in $\overleftarrow{p}$ that allows us to construct fixed points.

▶ **Theorem 27.** *For every $\sigma \in \overleftarrow{\mathbf{E}}_c$ there is a unique morphism $\text{löb}^c_\sigma \colon \sigma^{\blacktriangleright^c \sigma} \to \sigma$ in $\overleftarrow{\mathbf{E}}_c$, such that for all g-contractive maps f the map $\text{löb}^c_\sigma \circ \lambda g$ is a solution for f. From this we obtain every for $\sigma \in \overline{\mathbf{E}}_X$ a unique morphism $\text{löb}_\sigma \colon \sigma^{\blacktriangleright \sigma} \to \sigma$ that solves any contractive map in $\overline{\mathbf{E}}_X$.*

▶ **Proposition 28.** *The morphisms löb^c and löb are dinatural transformations.*

From Theorem 27, we obtain the Löb proof rule. This rule allows us to introduce recursion into proofs, by giving us the proof goal σ as an assumption guarded by the later modality.

$$\frac{f \colon \tau \times \blacktriangleright^c \sigma \to \sigma}{\text{löb}^c_\sigma \circ \lambda f \colon \tau \to \sigma} \quad \text{with} \quad \text{löb}^c_\sigma \circ \lambda f = f \circ (\text{id} \times \text{next}_\sigma) \circ \langle \text{id}, \text{löb}^c_\sigma \circ \lambda f \rangle$$

5 Locally Contractive Functors and Coinduction

One of the central notions of Birkedal et al. [16] is that of locally contractive functors. Such functors admit fixed points in the topos of trees and are closed under various constructions like composition and products. Locally contractive functors are used in [16] as a different way of solving recursive domain equations, which is where the name "synthetic domain theory" comes from. In this section, we restate the definition of contractive functors, and generalise the fixed point construction and the closure properties to the fibrations $\overleftarrow{p}$ and $\overline{p}$.

In the following, we use the natural transformation $\text{comp}_{X,Y,Z} \colon X^Y \times Z^X \to Z^Y$ that composes internal morphisms. We will refer to the isomorphism $\blacktriangleright \sigma \times \blacktriangleright \tau \to \blacktriangleright(\sigma \times \tau)$ as $\delta^{\blacktriangleright}$.

▶ **Definition 29.** *A functor $F \colon \overleftarrow{\mathbf{C}} \to \overleftarrow{\mathbf{C}}$ is called* locally contractive *if F is strong, there is a natural transformation $C^F_{\sigma,\tau} \colon \blacktriangleright(\sigma^\tau) \to F\sigma^{F\tau}$ with $\text{st}^F_{\sigma,\tau} = C^F_{\sigma,\tau} \circ \text{next}_{\sigma^\tau}$, and fulfils $C^F_{\sigma,\sigma} \circ \blacktriangleright \ulcorner \text{id} \urcorner = \ulcorner \text{id} \urcorner$ and $\text{comp} \circ (C^F_{\sigma,\tau} \times C^F_{\gamma,\sigma}) = C^F_{\gamma,\tau} \circ \blacktriangleright \text{comp} \circ \delta^{\blacktriangleright}$. A lifting $(F, G) \colon \overleftarrow{p} \to \overleftarrow{p}$ is* locally contractive *if (F, G) is strong, F and G are locally contractive and $\overleftarrow{p}\, C^G = C^F$.*

The next theorem records the essential closure properties of locally contractive functors.

▶ **Theorem 30.** *Let $F, G \colon \mathbf{C} \to \mathbf{C}$ be functors. If F* or *G is locally contractive, then $F \circ G$ is; if F and G are locally contractive, then $F \times G$ is. Both, $(\triangleright, \blacktriangleright) \colon \overleftarrow{p} \to \overleftarrow{p}$ and $\blacktriangleright \colon \overline{p} \to \overline{p}$ are locally contractive. The constant functor $\lambda \tau. \sigma$ is locally contractive for any $\sigma \in \overleftarrow{\mathbf{E}}$.*

The proof of the following theorem proceeds in the same way as the one given in [16] by first establishing for all $\alpha \in \mathbf{I}$ and $\beta < \alpha$ that a locally contractive functor G maps any β-isomorphism f to an α-isomorphism Gf above the corresponding α-iso $F(\overleftarrow{p} f)$. An α-isomorphism is thereby a morphism $f \colon \sigma \to \tau$, such that for all $\beta \leq \alpha$ all f_β are isomorphisms.

▶ **Theorem 31.** *Any locally contractive lifting (F, G) has a unique fixed point in $\overleftarrow{\mathbf{E}}$.*

In Section 7, we will need the following version on fibres for the semantics of pLμ.

▶ **Theorem 32.** *For any $c \in \overleftarrow{\mathbf{B}}$ and locally contractive functor $F \colon \overleftarrow{\mathbf{E}}_c \to \overleftarrow{\mathbf{E}}_c$ a unique fixed point of F exists in $\overleftarrow{\mathbf{E}}_c$. Consequently, also locally contractive functors on $\overline{\mathbf{E}}_X$ for $X \in \mathbf{B}$ have unique fixed points by using that $\overline{\mathbf{E}}_X \cong \overleftarrow{\mathbf{E}}_{K^{\mathbf{B}}(X)}$.*

5.1 The Final Chain and Up-To Techniques

Having laid the ground work, we come to the objects of interest: coinductive predicates. The following definition captures the usual construction of the final chain. Recall that $\overleftarrow{(-)}$ is a functor $\mathbf{Fib} \to \mathbf{Fib}$. Thus, from $\Phi \colon \mathbf{E}_I \to \mathbf{E}_I$, we obtain $\overleftarrow{\Phi} \colon \overline{\mathbf{E}}_I \to \overline{\mathbf{E}}_I$ by Lemma 14. The functor $\overleftarrow{\Phi}$ applies thereby Φ point-wise to chains.

▶ **Definition 33.** *Given a functor* $\Phi\colon \mathbf{E}_I \to \mathbf{E}_I$*, we define the* final chain *of* Φ *to be the fixed point* $\nu(\blacktriangleright \overleftarrow{\Phi})$ *of the locally contractive functor* $\blacktriangleright \circ \overleftarrow{\Phi}$*.*

We can now construct an adjunction between Φ-invariants and coalgebras for $\blacktriangleright \overleftarrow{\Phi}$, cf. [44]. This is a slightly more expressive version of the usual construction of final coalgebras.

▶ **Theorem 34.** *Suppose* $\Phi\colon \mathbf{E}_I \to \mathbf{E}_I$ *preserves* $\mathbf{I}^{\mathrm{op}}$*-limits. Then the adjunction* $K^{\mathbf{E}} \dashv L^{\mathbf{E}}$ *lifts to an adjunction* $\hat{K}^{\mathbf{E}} \dashv \hat{L}^{\mathbf{E}}$ *between the categories* $\mathrm{CoAlg}(\Phi)$ *and* $\mathrm{CoAlg}(\blacktriangleright \overleftarrow{\Phi})$ *of* Φ*- and* $\blacktriangleright \overleftarrow{\Phi}$*-coalgebras. This gives* $\nu\Phi \cong \hat{L}^{\mathbf{E}}(\nu(\blacktriangleright \overleftarrow{\Phi}))$*, where* $\nu(\blacktriangleright \overleftarrow{\Phi})$ *is the unique fixed point of* $\blacktriangleright \overleftarrow{\Phi}$*.*

Theorem 34 will play a central role in recursive proofs, as it allows us to express maps into $\nu\Phi$ in terms of maps into $\nu(\blacktriangleright \overleftarrow{\Phi})$, and it allows us to unfold the final chain and thereby to make progress in a proof. Just as important as unfolding is the ability to reason inside syntactic contexts, use transitivity of relations etc. in a proof. Such properties are captured by up-to techniques, see Definition 8.

▶ **Theorem 35.** *Let* T *and* Φ *be functors* $\mathbf{E}_I \to \mathbf{E}_I$*. If there is a natural transformation* $\rho\colon T\Phi \Rightarrow \Phi T$*, then there is a map* $\widehat{\rho}\colon \overleftarrow{T}\nu(\blacktriangleright \overleftarrow{\Phi}) \to \nu(\blacktriangleright \overleftarrow{\Phi})$ *in* $\overline{\mathbf{E}}_I$*.*

▶ Remark 36. Pous and Rot [56] prove a result similar to Theorem 35, namely that a monotone function T on a complete lattice is below the companion of Φ if and only if there is a map $\overleftarrow{T}\nu(\blacktriangleright \overleftarrow{\Phi}) \to \nu(\blacktriangleright \overleftarrow{\Phi})$. This is equivalent to Theorem 35 because the companion is compatible. ⌟

From Theorems 19, 34 and 35 we obtain the following proof rules, where the first initialises a proof by moving from a coinductive predicate to the final chain.

$$\frac{K_A \longrightarrow \nu(\blacktriangleright \overleftarrow{\Phi})}{A \longrightarrow \nu\Phi} \qquad \frac{f\colon \tau \to \blacktriangleright \overleftarrow{\Phi}(\nu\blacktriangleright \overleftarrow{\Phi})}{f\colon \tau \to \nu\blacktriangleright \overleftarrow{\Phi}} \qquad \frac{\rho\colon T\Phi \Rightarrow \Phi T \qquad f\colon \tau \to \overleftarrow{T}\nu(\blacktriangleright \overleftarrow{\Phi})}{\overleftarrow{\rho} \circ f\colon \tau \to \nu(\blacktriangleright \overleftarrow{\Phi})}$$

The last result in this section, recorded here for completeness, allows us to obtain compatible up-to techniques on fibres from global up-to techniques.

▶ **Proposition 37.** *Let* $(F, G)\colon p \to p$ *be a map of fibrations,* $c\colon I \to FI$ *a coalgebra in* $\mathbf{B}$*, and* $T\colon \mathbf{E} \to \mathbf{E}$ *a lifting of the identity* $\mathrm{Id}_{\mathbf{E}}$*. Define* $\Phi := c^* \circ G\colon \mathbf{E}_I \to \mathbf{E}_I$ *to be the predicate transformer associated to* c*, see Definition 8. If there is a vertical natural transformation* $\rho\colon TG \Rightarrow GT$*, then there is a vertical natural transformation* $\rho^c\colon T\Phi \Rightarrow \Phi T$*.*

6 Chains in First-Order Fibrations

The goal of this section is to show that the fibration $\overline{p}\colon \overline{\mathbf{E}} \to \mathbf{B}$ of $\mathbf{I}^{\mathrm{op}}$-chains with constant index is a first-order fibration (FO fibration) if $p\colon \mathbf{E} \to \mathbf{B}$ is an FO fibration. This allows us to construct out of a given FO logic another FO logic that features the later modality.

6.1 Products, Coproducts and Quantifiers for Descending Chains

Because of Lemma 14, we can apply many construction easily point-wise to chains with constant index. For instance, we can lift products and coproducts in the following sense.

▶ **Theorem 38.** *If for* $u\colon I \to J$ *in* $\mathbf{B}$ *the coproduct* $\coprod_u\colon \mathbf{E}_I \to \mathbf{E}_J$ *along* u *exists, then the coproduct* $\coprod_u\colon \overline{\mathbf{E}}_I \to \overline{\mathbf{E}}_J$ *along* u *is given by* $\overleftarrow{\coprod_u}$*. Similarly, the product* $\prod_u$ *along* u *is* $\overleftarrow{\prod_u}$*.*

▶ **Example 39.** Both Pred and qPred to have products and coproducts along any function in **Set**. For instance, products in qPred along functions $u\colon X \to Y$ are given by

$$\prod_u (\delta\colon X \to [0,1])(y) = \inf\{\delta(x) \mid x \in X, u(x) = y\}.$$

In a syntactic logic, Example 4, one has that $\mathcal{L} \to \mathcal{C}$ products and coproducts along projections $(\Gamma, x : A) \to \Gamma$ are universal and existential quantification over A, respectively. Arbitrary (co)products can then be defined in terms of the equality relation in the logic, cf. [41]. By Theorem 38, all these products and coproducts lift to the fibrations of descending chains. ⌟

Let us denote for $I \in \mathbf{B}$ the later modality on $\overline{\mathbf{E}}_I$ by $\blacktriangleright^I$. We can then establish the following essential properties about the interaction of the later modalities and (co)products, which are analogue to those in [16, Thm. 2.7]. This theorem allows one to distribute in proofs quantifiers over the later modality.

▶ **Theorem 40.** *The following holds for fibred products and coproducts in $\overline{p}$.*

- *There is an isomorphism $\blacktriangleright^J \circ \prod_u \cong \prod_u \circ \blacktriangleright^I$.*
- *There is a natural transformation $\iota\colon \coprod_u \circ \blacktriangleright^I \Rightarrow \blacktriangleright^J \circ \coprod_u$. Moreover, if u is inhabited, that is, has a section $v\colon J \to I$, then ι has a section ι^v.*

For $u\colon I \to J$ in $\mathbf{B}$, we can present the central results of this section as proof rules:

$$\frac{f\colon \tau \to u^* \sigma}{\check{f}\colon \coprod_u \tau \to \sigma} \qquad \frac{f\colon \tau \to \coprod_u (\blacktriangleright^I \sigma)}{\iota \circ f\colon \tau \to \blacktriangleright^J (\coprod_u \sigma)} \qquad \frac{f\colon u^* \tau \to \sigma}{\check{f}\colon \tau \to \prod_u \sigma} \qquad \frac{f\colon \tau \to \blacktriangleright^J (\prod_u \sigma)}{\check{f}\colon \tau \to \prod_u (\blacktriangleright^I \sigma)}$$

6.2 First Order Fibration of Descending Chains

As the name suggests, a first-order fibration models first-order logic with equality. Such an FO fibration is a fibration $p\colon \mathbf{E} \to \mathbf{B}$, which is a fibred pre-ordered lattice and fibred CCC, and has products and coproducts, which satisfy the Beck-Chevalley and Frobenius conditions, along all morphisms in $\mathbf{B}$, see [41, Def. 4.2.1] for details. We now show that not only is the fibration of constant-index chains in p an FO fibration, but is also strongly related to p.

▶ **Theorem 41.** *If $p\colon \mathbf{E} \to \mathbf{B}$ is an FO fibration, then $\overline{p}\colon \overline{\mathbf{E}} \to \mathbf{B}$ is as well an FO fibration. Furthermore, if the fibred coproducts and coproducts along morphisms preserve $\mathbf{I}^{\mathrm{op}}$-limits, then $L^{\mathbf{E}}\colon \overline{\mathbf{E}} \to \mathbf{E}$ preserves all the FO structure except for implication. For implication, truth is preserved, i.e., for all $\sigma, \tau \in \overline{\mathbf{E}}_I$ there is a morphism $L(\sigma^\tau) \to L\sigma^{L\tau}$. If $\tau = K^{\mathbf{E}}_X$ for some $X \in \mathbf{E}_I$, then this morphism is an isomorphism. Finally, $K^{\mathbf{E}}$ is a fully faithful functor.*

That preservation of exponentials fails can be seen by taking $\sigma, \tau \in [\omega^{\mathrm{op}}, \mathrm{Pred}_{\mathbb{N}}]$ to be $\tau_n = \mathbb{N} \setminus \{1, \ldots, n\}$ and $\sigma_n = \{0\}$. Then $L(\sigma^\tau) = \{0\}$ but $L\sigma^{L\tau} = \mathbb{N}$.

7 Examples

In this section, we show the framework in action. Specifically, we show how a novel proof system for the probabilistic modal μ-calculus pLμ can be obtained, and we show a language and its semantics for probabilistic productive coinductive programming.

7.1 Recursive Proofs for the Probabilistic Modal μ-Calculus

The probabilistic modal μ-calculus pLμ has exactly the same syntax as the modal μ-calculus Lμ. However, formulas are interpreted as probability distributions [39]. We extend the

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\blacktriangleright \gamma, \psi \vdash \psi}\ \textbf{(Pr)}
}{\blacktriangleright \gamma, \psi \vdash \varphi(\psi)}\ \text{assumption}
}{\blacktriangleright \gamma, \psi \vdash \varphi(\blacktriangleright \psi)}\ \varphi \text{ positive} + \textbf{(Next)}
\qquad
\dfrac{
\dfrac{}{\blacktriangleright \gamma, \psi \vdash \blacktriangleright(\psi \to \nu X.\, \varphi(X))}\ \textbf{(Pr)}
}{\blacktriangleright \gamma, \psi \vdash \blacktriangleright \psi \to \blacktriangleright \nu X.\, \varphi(X)}\ \textbf{(Mon)}
}{
\dfrac{
\dfrac{\blacktriangleright \gamma, \psi \vdash \varphi(\blacktriangleright \nu X.\, \varphi(X))}{\blacktriangleright \gamma, \psi \vdash \nu X.\, \varphi(X)}\ \textbf{(Step)}
}{
\dfrac{\blacktriangleright \gamma \vdash \gamma}{\vdash \gamma}\ \textbf{(Löb)}
}\ \textbf{(}\to\textbf{-I)}
}\ \varphi \text{ positive}
$$

Figure 3 $\varphi(X)$ positive in X, ψ Lμ-formula, $\gamma = \psi \to \nu X.\, \varphi(X)$ with assumption $\psi \to \varphi(\psi)$.

coinductive fragment of pLμ here with the later modality and thereby obtain the following formulas over sets At and Var of propositional variables P and fixed point variables X:

$$\varphi, \psi ::= P \mid \overline{P} \mid X \mid \top \mid \bot \mid \nu X.\, \varphi \mid \blacktriangleright \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \varphi \sqcap \psi \mid \varphi \sqcup \psi \mid \varphi \to \psi,$$

where X must occur positively in φ when forming $\nu X.\, \varphi$. Given a formula φ with no or one free variable[2] X, a Segala system $c \colon Q \to \mathcal{S}(Q)$ and an interpretation $I \colon Q \to \mathrm{qPred}_{\mathrm{At}}$, we use Theorem 30 to define a locally contractive functor $\langle\!\langle \varphi \rangle\!\rangle \colon \overline{\mathrm{Pred}}_Q^{\,n} \to \overline{\mathrm{Pred}}_Q$ with $n = 0, 1$, where we only display the interesting cases. The remaining cases are given in Appendix A.

$$\langle\!\langle P \rangle\!\rangle = K(I(P)) \qquad \langle\!\langle X \rangle\!\rangle = \blacktriangleright \qquad \langle\!\langle \nu X.\, \varphi \rangle\!\rangle = \nu \langle\!\langle \varphi \rangle\!\rangle \qquad \langle\!\langle \Box\varphi \rangle\!\rangle = c^{\#} \circ \overleftarrow{\mathcal{S}^{\Box}} \circ \langle\!\langle \varphi \rangle\!\rangle$$

This definition and the previous development gives us that the following rules are sound for this interpretation, where double lines are rules that can be used in both directions.

$$\dfrac{\Delta \vdash \varphi[\blacktriangleright \nu X.\, \varphi / X]}{\Delta \vdash \nu X.\, \varphi}\ \textbf{(Step)} \qquad \dfrac{\Delta \vdash \varphi}{\Delta \vdash \blacktriangleright \varphi}\ \textbf{(Next)} \qquad \dfrac{\Delta \vdash \blacktriangleright(\varphi \to \psi)}{\Delta \vdash \blacktriangleright \varphi \to \blacktriangleright \psi}\ \textbf{(Mon)}$$

$$\dfrac{\Delta, \blacktriangleright \varphi \vdash \varphi}{\Delta \vdash \varphi}\ \textbf{(Löb)} \qquad \dfrac{\Delta \vdash \Box \blacktriangleright \varphi}{\Delta \vdash \blacktriangleright \Box\varphi} \qquad \dfrac{\Delta \vdash \Diamond \blacktriangleright \varphi}{\Delta \vdash \blacktriangleright \Diamond\varphi} \qquad + \text{ normal, intuitionistic modal logic}$$

In Figure 3, we show how Park's rule can be proven from these rules. Theorem 41 gives us that these rules are sound and their semantics are complete for the standard semantics of formulas that only have constant premises, i.e. pure modal formulas, in implications.

Let us make two final remarks about this example. First, note that the implication is an internalisation of the ordering on quantitative predicates and thus has, a priori, nothing to do with probabilities. In particular, we have $\langle\!\langle \overline{P} \rangle\!\rangle \neq \langle\!\langle P \to \bot \rangle\!\rangle$. Second, the proof rules give rise to a constructive and recursive proof system for pLμ. This is insofar interesting, as that the completeness proof for Kozen's axiomatisation for Lμ is non-constructive, and a non-probabilistic version of the above presented proof system may give new insights, cf. [27]. Also an analogous version of our cut-free proof system for Horn clause theories [10] may shed new light on cut-free proofs for (p)Lμ, cf. [4].

7.2 Probabilistic Productive Coinductive Programming

In this last example, we show how one can obtain a new programming language for higher-order probabilistic programming with coinductive types, in which all programs are terminating.

[2] We restrict ourselves to this case for simplicity. Supporting several variables is a direct generalisation.

This is in contrast to the language provided in [74], where full recursion is essential to coinductive programming. Full recursion introduces, however, non-terminating and non-productive programs, which makes reasoning about programs unnecessarily difficult [73], especially in the probabilistic setting. As such, the total programming language, which we are about to introduce, provides us with coinductive, probabilistic types, while retaining the good properties of terminating and productive programs.

The essential ingredient are so-called *quasi-Borel spaces* that were introduced by Heunen et al. [37] as a setting for higher-order probabilistic programming. In particular, the category **qBS** of quasi-Borel spaces and their morphisms is (co)complete and Cartesian closed, see [37, 74] for details. From the framework, we obtain that $\overline{\mathbf{qBS}} = [\omega^{\mathrm{op}}, \mathbf{qBS}]$ is as well a (co)complete CCC with later modality and Löb rule. This allows us to provide a probabilistic higher-order programming language with coinductive types.

This language has types and terms that are given in Appendix B. One coinductive example given in [74] is that of a random walk, which produces a stream of random positions for a given standard deviation σ. We may define the type $\mathbb{R}^\omega$ of $\mathbb{R}$-valued streams as fixed point type by $\mathbb{R}^\omega = \mathrm{fix}\, X.\, \mathbb{R} \times \blacktriangleright X$. A random walk can be produced by the following guarded recursive program $\mathrm{RW} : \mathbb{R} \to \mathbb{R} \to \mathbb{R}^\omega$, where $\mathrm{normal}\langle \rho, \sigma \rangle$ draws from a normal distribution with expected value ρ and standard deviation σ.

$$\mathrm{RW} = \lambda\sigma.\, \mathrm{fix}\, f : \blacktriangleright(\mathbb{R} \to \mathbb{R}^\omega).\, \lambda x.\, \mathrm{in}\, \langle x, f \circledast \mathrm{next}\, (\mathrm{normal}\, \langle x, \sigma \rangle) \rangle$$

The details of how the above types and terms can be interpreted in $\overline{\mathbf{qBS}}$ are given in Appendix B. Since **qBS** is complete, we thus obtain an interpretation of the types and terms in **qBS**, which corresponds to the expected final coalgebra semantics, see Theorem 34.

8 Conclusion and Future Work

In this paper, we have established a framework that allows us to reason about coinductive predicates in many cases by using recursive proofs. At the heart of this approach sits the so-called later modality, which comes from provability logic [12, 68, 70] but was later used to obtain guarded recursion in type theories [5, 6, 17, 52] and in domain theory [15, 16]. This modality allows us to control the recursion steps in a proof without having to invoke parity or similar conditions [22, 28, 64, 67], as we have seen in the examples in Section 7. Moreover, even though Birkedal et al. [16] obtained similar results, their framework is limited to **Set**-valued presheaves, while our results are applicable in a much wider range of situations. In particular, we were able to devise a novel probabilistic programming language that guarantees productivity on coinductive types.

So what is there left to do? For once, we have not touched upon how to automatically extract a syntactic logic and models from the fibration $\overleftarrow{L} \to \overleftarrow{\mathcal{C}}$ obtained in Example 24. This would subsume and simplify much of the development in [9]. Next, we only proved the existence of quantifiers that range over fixed domains. It would be useful to extend this construction to indexed domains to, for example, obtain Kripke models abstractly. However, such a construction would be similar to that of exponents in Theorem 20 and thus quite involved. At the same time, also a category theoretical analysis of the delayed implication in [23] is needed. Also a closer analysis of the relation to proof systems obtained through parameterised coinduction, the companion or cyclic proof systems may shed some light on the strength of the proof approach presented in this paper. Such an analysis requires to understand how the causal proofs that the presented framework and the companion support [56], and parameterised coinduction are related. Finally, after a few first step into the direction of proof search for coinductive Horn clause theories in [10], the results of the present paper need to be applied to obtain proof search procedures for other logics and theories.

References

1 Andreas Abel, Brigitte Pientka, David Thibodeau, and Anton Setzer. Copatterns: Programming Infinite Structures by Observations. In *POPL'13*, pages 27–38. ACM, 2013. doi:10.1145/2429069.2429075.

2 Jiří Adámek, Stefan Milius, and Lawrence S. Moss. Introduction to Category Theroy, Algebras and Coalgebra. *A monograph in preparation*, 2010. URL: http://www.tu-braunschweig.de/Medien-DB/iti/survey_full.pdf.

3 Jiŕı Adámek and Vera Trnková. Initial Algebras and Terminal Coalgebras in Many-Sorted Sets. *MSCS*, 21(2):481–509, 2011. doi:10.1017/S0960129510000502.

4 Bahareh Afshari and Graham E. Leigh. Cut-Free Completeness for Modal Mu-Calculus. In *Proc. of LICS'17*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005088.

5 Andrew W. Appel, Paul-André Melliès, Christopher D. Richards, and Jérôme Vouillon. A Very Modal Model of a Modern, Major, General Type System. In *POPL*, pages 109–122. ACM, 2007. doi:10.1145/1190216.1190235.

6 Robert Atkey and Conor McBride. Productive Coprogramming with Guarded Recursion. In *ICFP*, pages 197–208. ACM, 2013. doi:10.1145/2500365.2500597.

7 Michael Barr. Terminal Coalgebras in Well-Founded Set Theory. *TCS*, 114(2):299–315, 1993. doi:10.1016/0304-3975(93)90076-6.

8 Henning Basold. Code Repository, 2018. URL: https://perso.ens-lyon.fr/henning.basold/code/.

9 Henning Basold. *Mixed Inductive-Coinductive Reasoning: Types, Programs and Logic*. PhD Thesis, Radboud University, 2018. URL: https://hdl.handle.net/2066/190323.

10 Henning Basold, Ekaterina Komendantskaya, and Yue Li. Coinduction in Uniform: Foundations for Corecursive Proof Search with Horn Clauses. In *ESOP'19*, volume 11423 of *LNCS*. Springer, 2019. arXiv:1811.07644.

11 Henning Basold, Damien Pous, and Jurriaan Rot. Monoidal Company for Accessible Functors. In *CALCO 2017*, volume 72 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CALCO.2017.5.

12 Lev D. Beklemishev. Parameter Free Induction and Provably Total Computable Functions. *TCS*, 224(1-2):13–33, 1999. doi:10.1016/S0304-3975(98)00305-3.

13 Jean Bénabou. Fibered Categories and the Foundations of Naive Category Theory. *Journal of Symbolic Logic*, 50(1):10–37, 1985. doi:10.2307/2273784.

14 Lars Birkedal, Aleš Bizjak, Ranald Clouston, Hans Bugge Grathwohl, Bas Spitters, and Andrea Vezzosi. Guarded Cubical Type Theory: Path Equality for Guarded Recursion. In *CSL 2016*, volume 62 of *LIPIcs*, pages 23:1–23:17. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.CSL.2016.23.

15 Lars Birkedal and Rasmus Ejlers Møgelberg. Intensional Type Theory with Guarded Recursive Types qua Fixed Points on Universes. In *LICS*, pages 213–222. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.27.

16 Lars Birkedal, Rasmus Ejlers Møgelberg, Jan Schwinghammer, and Kristian Støvring. First Steps in Synthetic Guarded Domain Theory: Step-Indexing in the Topos of Trees. *LMCS*, 8(4), 2012. doi:10.2168/LMCS-8(4:1)2012.

17 Ales Bizjak, Hans Bugge Grathwohl, Ranald Clouston, Rasmus Ejlers Møgelberg, and Lars Birkedal. Guarded Dependent Type Theory with Coinductive Types. In *FoSSaCS*, volume 9634 of *LNCS*, pages 20–35. Springer, 2016. arXiv:1601.01586.

18 Jasmin Christian Blanchette, Johannes Hölzl, Andreas Lochbihler, Lorenz Panny, Andrei Popescu, and Dmitriy Traytel. Truly Modular (Co)Datatypes for Isabelle/HOL. In Gerwin

Klein and Ruben Gamboa, editors, *Proceedings of ITP 2014*, volume 8558 of *LNCS*, pages 93–110. Springer, 2014. doi:10.1007/978-3-319-08970-6_7.

19 Filippo Bonchi, Daniela Petrişan, Damien Pous, and Jurriaan Rot. Coinduction Up-to in a Fibrational Setting. In *LICS '14*, pages 20:1–20:9. ACM, 2014. doi:10.1145/2603088.2603149.

20 Julian C. Bradfield and Colin Stirling. Modal Mu-Calculi. In *Handbook of Modal Logic*, pages 721–756. Elsevier, 2006.

21 James Brotherston. Cyclic Proofs for First-Order Logic with Inductive Definitions. In Bernhard Beckert, editor, *Proceedings of TABLEAUX 2005*, volume 3702 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2005. doi:10.1007/11554554_8.

22 James Brotherston and Alex Simpson. Complete Sequent Calculi for Induction and Infinite Descent. In *Proceedings of LICS 2007*, pages 51–62. IEEE Computer Society, 2007. doi:10.1109/LICS.2007.16.

23 Ranald Clouston and Rajeev Goré. Sequent Calculus in the Topos of Trees. In Andrew M. Pitts, editor, *Proc. of FoSSaCS 2015*, volume 9034 of *LNCS*, pages 133–147. Springer, 2015. doi:10.1007/978-3-662-46678-0_9.

24 J. Robin B. Cockett. Deforestation, Program Transformation, and Cut-Elimination. *Electr. Notes Theor. Comput. Sci.*, 44(1):88–127, 2001. doi:10.1016/S1571-0661(04)80904-6.

25 The Coq Development Team. The Coq Proof Assistant Reference Manual. Technical report, LogiCal Project, 2012. Version 8.4. URL: http://coq.inria.fr.

26 Christian Dax, Martin Hofmann, and Martin Lange. A Proof System for the Linear Time μ-Calculus. In S. Arun-Kumar and Naveen Garg, editors, *Proceedings of FSTTCS 2006*, volume 4337 of *LNCS*, pages 273–284. Springer, 2006. doi:10.1007/11944836_26.

27 Amina Doumane. *On the Infinitary Proof Theory of Logics with Fixed Points*. PhD Thesis, Université Paris Diderot, 2017.

28 Jérôme Fortier and Luigi Santocanale. Cuts for Circular Proofs: Semantics and Cut-Elimination. In *CSL*, pages 248–262, 2013. doi:10.4230/LIPIcs.CSL.2013.248.

29 Clément Fumex, Neil Ghani, and Patricia Johann. Indexed Induction and Coinduction, Fibrationally. In *Proc. of CALCO '11*, volume 6859 of *Lecture Notes in Computer Science*, pages 176–191. Springer, 2011. doi:10.1007/978-3-642-22944-2_13.

30 Sergey Goncharov and Lutz Schröder. Guarded Traced Categories. In Christel Baier and Ugo Dal Lago, editors, *Proc. of FOSSACS'18*, volume 10803 of *LNCS*, pages 313–330. Springer, 2018. doi:10.1007/978-3-319-89366-2_17.

31 Programming Logic group on Agda. Agda Documentation. Technical report, Chalmers and Gothenburg University, 2015. Version 2.4.2.5. URL: http://wiki.portal.chalmers.se/agda/.

32 Tatsuya Hagino. A Typed Lambda Calculus with Categorical Type Constructors. In *Category Theory in Computer Science*, Lecture Notes in Computer Science, pages 140–157. Springer, 1987. doi:10.1007/3-540-18508-9_24.

33 Helle Hvid Hansen, Clemens Kupke, and Jan Rutten. Stream Differential Equations: Specification Formats and Solution Methods. *LMCS*, 13(1), 2017. doi:10.23638/LMCS-13(1:3)2017.

34 Masahito Hasegawa. On Traced Monoidal Closed Categories. *Mathematical Structures in Computer Science*, 19(2):217–244, 2009. doi:10.1017/S0960129508007184.

35 Ichiro Hasuo, Kenta Cho, Toshiki Kataoka, and Bart Jacobs. Coinductive Predicates and Final Sequences in a Fibration. *Electronic Notes in Theoretical Computer Science*, 298:197–214, November 2013. doi:10.1016/j.entcs.2013.09.014.

36 Claudio Hermida and Bart Jacobs. Structural Induction and Coinduction in a Fibrational Setting. *Information and Computation*, 145:107–152, 1997. doi:10.1006/inco.1998.2725.

37 Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. A Convenient Category for Higher-Order Probability Theory. In *Proc. of LICS'17*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005137.

38 Chung-Kil Hur, Georg Neis, Derek Dreyer, and Viktor Vafeiadis. The Power of Parameterization in Coinductive Proof. In *Proc. of POPL'13*, POPL '13, pages 193–206. ACM, 2013. `doi:10.1145/2429069.2429093`.

39 Michael Huth and Marta Z. Kwiatkowska. Quantitative Analysis and Model Checking. In *Proc. of LICS'97*, pages 111–122. IEEE Computer Society, 1997. `doi:10.1109/LICS.1997.614940`.

40 J Martin E Hyland. The Effective Topos. In *Studies in Logic and the Foundations of Mathematics*, volume 110, pages 165–216. Elsevier, 1982.

41 Bart Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

42 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Number 59 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016. `doi:10.1017/CBO9781316823187`.

43 Bart Jacobs and Jan Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *EATCS Bulletin*, 62:62–222, 1997.

44 Henning Kerstan, Barbara König, and Bram Westerbaan. Lifting Adjunctions to Coalgebras to (Re)Discover Automata Constructions. In Marcello M. Bonsangue, editor, *Revised Selected Papers of CMCS'14*, volume 8446 of *LNCS*, pages 168–188. Springer, 2014. `doi:10.1007/978-3-662-44124-4_10`.

45 Anders Kock. Strong Functors and Monoidal Monads. *Archiv der Mathematik*, 23(1):113–120, 1972.

46 Dexter Kozen. Results on the Propositional μ-Calculus. *Theor. Comput. Sci.*, 27:333–354, 1983. `doi:10.1016/0304-3975(82)90125-6`.

47 F William Lawvere. Diagonal Arguments and Cartesian Closed Categories. In *Category Theory, Homology Theory and Their Applications II*, pages 134–145. Springer, 1969.

48 Nex Paul Mendler. Inductive Types and Type Constraints in the Second-Order Lambda Calculus. *Ann. Pure Appl. Logic*, 51(1-2):159–172, 1991. `doi:10.1016/0168-0072(91)90069-X`.

49 Matteo Mio and Alex Simpson. Łukasiewicz (μ)-calculus. *Fundam. Inform.*, 150(3-4):317–346, 2017. `doi:10.3233/FI-2017-1472`.

50 Rasmus Ejlers Møgelberg. A Type Theory for Productive Coprogramming via Guarded Recursion. In *CSL-LICS*, pages 71:1–71:10. ACM, 2014. `doi:10.1145/2603088.2603132`.

51 Lawrence S. Moss. Parametric Corecursion. *Theoretical Computer Science*, 260:139–163, 2001.

52 Hiroshi Nakano. A Modality for Recursion. In *LICS*, pages 255–266. IEEE Computer Society, 2000. `doi:10.1109/LICS.2000.855774`.

53 Damian Niwinski and Igor Walukiewicz. Games for the μ-Calculus. *TCS*, 163(1&2):99–116, 1996. `doi:10.1016/0304-3975(95)00136-0`.

54 Damien Pous. Complete Lattices and Up-To Techniques. In Zhong Shao, editor, *APLAS'07*, volume 4807 of *LNCS*, pages 351–366. Springer, 2007. `doi:10.1007/978-3-540-76637-7_24`.

55 Damien Pous. Coinduction All the Way Up. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of LICS '16*, pages 307–316. ACM, 2016. `doi:10.1145/2933575.2934564`.

56 Damien Pous and Jurriaan Rot. Companions, Codensity, and Causality. In *Proceedings of FOSSACS 2017*, 2017. `doi:10.1007/978-3-662-54458-7_7`.

57 Grigore Roşu and Dorel Lucanu. Circular Coinduction: A Proof Theoretical Foundation. In *CALCO*, volume 5728 of *LNCS*, pages 127–144. Springer, 2009. `doi:10.1007/978-3-642-03741-2_10`.

58 Jurriaan Rot. *Enhanced Coinduction*. PhD, University Leiden, Leiden, 2015.

59 Jurriaan Rot, Filippo Bonchi, Marcello Bonsangue, Damien Pous, Jan Rutten, and Alexandra Silva. Enhanced Coalgebraic Bisimulation. *MSCS*, 27(7):1236–1264, 2017. `doi:10.1017/S0960129515000523`.

60 Jan Rutten. Universal Coalgebra: A Theory of Systems. *TCS*, 249(1):3–80, 2000. `doi:10.1016/S0304-3975(00)00056-6`.

61 Jan Rutten. Behavioural Differential Equations: A Coinductive Calculus of Streams, Automata, and Power Series. *TCS*, 308(1-3):1–53, 2003. `doi:10.1016/S0304-3975(02)00895-2`.

62 Jan Rutten. *The Method of Coalgebra: Exercises in Coinduction*. CWI, Amsterdam, February 2019. URL: `http://persistent-identifier.org/?identifier=urn:nbn:nl:ui:18-28550`.

63 Davide Sangiorgi. On the Bisimulation Proof Method. *Mathematical Structures in Computer Science*, 8(5):447–479, 1998.

64 Luigi Santocanale. A Calculus of Circular Proofs and Its Categorical Semantics. In *FoSSaCS*, pages 357–371, 2002. `doi:10.1007/3-540-45931-6_25`.

65 Luigi Santocanale. μ-Bicomplete Categories and Parity Games. *RAIRO - ITA*, 36(2):195–227, 2002. `doi:10.1051/ita:2002010`.

66 Luke Simon, Ajay Bansal, Ajay Mallya, and Gopal Gupta. Co-Logic Programming: Extending Logic Programming with Coinduction. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Proc. of ICALP'07*, volume 4596 of *LNCS*, pages 472–483. Springer, 2007. `doi:10.1007/978-3-540-73420-8\_42`.

67 Alex Simpson. Cyclic Arithmetic Is Equivalent to Peano Arithmetic. In *Proceedings of FoSSaCS'17*, LNCS, 2017. `doi:10.1007/978-3-662-54458-7_17`.

68 Craig Smoryński. *Self-Reference and Modal Logic*. Universitext. Springer-Verlag, 1985.

69 Ana Sokolova. Probabilistic Systems Coalgebraically: A Survey. *TCS*, 412(38):5095–5110, 2011. `doi:10.1016/j.tcs.2011.05.008`.

70 Robert M. Solovay. Provability Interpretations of Modal Logic. *Israel Journal of Mathematics*, 25(3):287–304, 1976. `doi:10.1007/BF02757006`.

71 Thomas Streicher. Fibred Categories à La Jean Bénabou. *arXiv:math.CT/1801.02927*, 2018. `arXiv:1801.02927`.

72 Kasper Svendsen, Filip Sieczkowski, and Lars Birkedal. Transfinite Step-Indexing: Decoupling Concrete and Logical Steps. In Peter Thiemann, editor, *Proc. of ESOP'16*, volume 9632 of *LNCS*, pages 727–751. Springer, 2016. `doi:10.1007/978-3-662-49498-1_28`.

73 D. A. Turner. Elementary Strong Functional Programming. In Pieter H. Hartel and Marinus J. Plasmeijer, editors, *Proceedings of FPLE'95*, volume 1022 of *LNCS*, pages 1–13. Springer, 1995. `doi:10.1007/3-540-60675-0_35`.

74 Matthijs Vákár, Ohad Kammar, and Sam Staton. A Domain Theory for Statistical Probabilistic Programming. *PACMPL*, 3(POPL):36:1–36:29, 2019. `arXiv:1811.04196`.

75 Igor Walukiewicz. On Completeness of the Mu-Calculus. In *Proceedings of LICS '93*, pages 136–146. IEEE Computer Society, 1993. `doi:10.1109/LICS.1993.287593`.

A Interpretation of the Probabilistic Modal μ-Calculus

Given a formula φ with no or one free variable X, a Segala system $c\colon Q \to \mathcal{S}(Q)$ and an interpretation $I\colon Q \to \mathrm{qPred}_{\mathrm{At}}$, we use Theorem 30 to define a locally contractive functor $\langle\!\langle \varphi \rangle\!\rangle\colon \overline{\mathrm{Pred}}_Q^n \to \overline{\mathrm{Pred}}_Q$ with $n = 0, 1$, where $\heartsuit_Q = \top_Q, \wedge_Q, \dots$ are the corresponding fibred connectives in $\overline{\mathrm{qPred}}_Q$:

$$
\begin{array}{ccc}
\langle\!\langle P \rangle\!\rangle = K(I(P)) & \langle\!\langle \overline{P} \rangle\!\rangle = K(1 - I(P)) & \langle\!\langle X_k \rangle\!\rangle = \blacktriangleright \circ \pi_k \\
\langle\!\langle \top \rangle\!\rangle = \top_Q & \langle\!\langle \bot \rangle\!\rangle = \bot_Q & \langle\!\langle \nu X.\, \varphi \rangle\!\rangle = \nu \langle\!\langle \varphi \rangle\!\rangle \\
\langle\!\langle \blacktriangleright \varphi \rangle\!\rangle = \blacktriangleright \circ \langle\!\langle \varphi \rangle\!\rangle & \langle\!\langle \Box \varphi \rangle\!\rangle = c^{\#} \circ \overleftarrow{\mathcal{S}^{\Box}} \circ \langle\!\langle \varphi \rangle\!\rangle & \langle\!\langle \Diamond \varphi \rangle\!\rangle = c^{\#} \circ \overleftarrow{\mathcal{S}^{\Diamond}} \circ \langle\!\langle \varphi \rangle\!\rangle \\
\langle\!\langle \varphi \sqcap \psi \rangle\!\rangle = \langle\!\langle \varphi \rangle\!\rangle \wedge_Q \langle\!\langle \psi \rangle\!\rangle & \langle\!\langle \varphi \sqcup \psi \rangle\!\rangle = \langle\!\langle \varphi \rangle\!\rangle \vee_Q \langle\!\langle \psi \rangle\!\rangle & \langle\!\langle \varphi \to \psi \rangle\!\rangle = \langle\!\langle \varphi \rangle\!\rangle \Rightarrow_Q \langle\!\langle \psi \rangle\!\rangle
\end{array}
$$

B Types and Terms for Guarded Probabilistic Programming

Type, context and term formation rules for guarded probabilistic programming:

$$\frac{X \in \Delta}{\Delta \Vdash X : \mathbf{Ty}} \qquad \frac{\Delta \Vdash A : \mathbf{Ty}}{\Delta \Vdash \blacktriangleright A : \mathbf{Ty}} \qquad \frac{\Delta, X \Vdash A : \mathbf{Ty} \qquad X \text{ appears under } \blacktriangleright \text{ in } A}{\Delta \Vdash \mathrm{fix}\, X.\, A : \mathbf{Ty}}$$

$$\frac{}{\Delta \Vdash \mathbb{R} : \mathbf{Ty}} \qquad \frac{\Delta \Vdash A : \mathbf{Ty} \qquad \Delta \Vdash B : \mathbf{Ty}}{\Delta \Vdash A \times B : \mathbf{Ty}} \qquad \frac{\Delta \Vdash A : \mathbf{Ty} \qquad \Delta \Vdash B : \mathbf{Ty}}{\Delta \Vdash A \to B : \mathbf{Ty}}$$

$$\frac{}{\cdot\ \mathbf{Ctx}} \qquad \frac{\Gamma\ \mathbf{Ctx} \qquad x \notin \Gamma \qquad \Delta \Vdash B : \mathbf{Ty}}{\Gamma\ \mathbf{Ctx}}$$

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma \vdash t : A}{\Gamma \vdash \mathrm{next}\, t : \blacktriangleright A} \qquad \frac{\Gamma \vdash t : \blacktriangleright(A \to B) \qquad \Gamma \vdash s : \blacktriangleright A}{\Gamma \vdash t \circledast s : \blacktriangleright B}$$

$$\frac{\Gamma, x : \blacktriangleright A \vdash t : A}{\Gamma \vdash \mathrm{fix}\, x.\, t : A} \qquad \frac{\Gamma \vdash t : A[\mathrm{fix}\, X.\, A/X]}{\Gamma \vdash \mathrm{in}\, t : \mathrm{fix}\, X.\, A} \qquad \frac{\Gamma \vdash t : \mathrm{fix}\, X.\, A}{\Gamma \vdash \mathrm{out}\, t : A[\mathrm{fix}\, X.\, A/X]}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash s : A}{\Gamma \vdash t\, s : B}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash s : B}{\Gamma \vdash \langle t, s\rangle : A \times B} \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \mathrm{fst}\, t : A} \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \mathrm{snd}\, t : B}$$

$$\frac{a \in \mathbb{Q}}{\Gamma \vdash a : \mathbb{R}} \qquad \frac{}{\Gamma \vdash \mathrm{normal} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}}$$

Before we come to the semantics, let us single out values. This will simplify the denotational semantics, as values need to be embedded into the monad $\overleftarrow{P}$ that we will use to model the probabilistic effects of the language.

$$v, w, u ::= x \mid a \mid \langle v, w\rangle \mid \mathrm{in}\, v \mid \lambda x.\, t$$

We denote by Tm_A^Γ the terms of type A in context Γ and by Val_A^Γ the values of type A.

Let $P \colon \mathbf{qBS} \to \mathbf{qBS}$ be the strong probability monad on quasi-Borel spaces, see [74]. As monads lift easily point-wise to descending chain, we get a strong monad $\overleftarrow{P}$ on $\overleftarrow{\mathbf{qBS}}$ with unit $\overleftarrow{\eta}$ and bind operator $\overleftarrow{\gg=}$. Note that the unit and multiplication of the monad lift point-wise, while the strength needs to be defined through the unique mapping property of the end that we used to construct exponentials in $\overleftarrow{\mathbf{qBS}}$. In $\mathbf{qBS}$, we also find the normal distribution given as morphism $N \colon \mathbb{R} \times \mathbb{R} \to P(\mathbb{R})$. This gives us a morphism $K(N) \colon K(\mathbb{R}) \times K(\mathbb{R}) \to \overleftarrow{P}(K(\mathbb{R}))$, which we will use below. Next, we need a natural transformation $\iota \colon \blacktriangleright \overleftarrow{P} \Rightarrow \overleftarrow{P} \blacktriangleright$, which is given by $\iota_{\sigma,0} = \mathbf{1} \cong P\mathbf{0} \xrightarrow{P!} P\mathbf{1}$ and $\iota_{\sigma,n+1} = \mathrm{id}_{P\sigma_n}$.

The interpretation of types, context, values and terms over $\overleftarrow{\mathbf{qBS}}$ is then given as follows.

$$\begin{aligned}
&[\![\Delta \Vdash A : \mathbf{Ty}]\!] : \overleftarrow{\mathbf{qBS}}^{\Delta} \to \overleftarrow{\mathbf{qBS}} \\
&\quad [\![\Gamma\ \mathbf{Ctx}]\!] \in \overleftarrow{\mathbf{qBS}} \\
&\qquad [\![-]\!]^{\mathrm{val}} : \mathrm{Val}_A^\Gamma \to \overleftarrow{\mathbf{qBS}}([\![\Gamma]\!], [\![A]\!]) \\
&\qquad\quad [\![-]\!] : \mathrm{Tm}_A^\Gamma \to \overleftarrow{\mathbf{qBS}}\big([\![\Gamma]\!], \overleftarrow{P}[\![A]\!]\big)
\end{aligned}$$

$$[\![X]\!] = \pi_X \qquad [\![\blacktriangleright A]\!] = \blacktriangleright \circ [\![A]\!]$$
$$[\![\mathrm{fix}\, X.\, A]\!] = \nu[\![A]\!] \qquad [\![\mathbb{R}]\!] = K(\mathbb{R})$$
$$[\![A \times B]\!] = [\![A]\!] \times [\![B]\!] \qquad [\![A \to B]\!] = [\![A]\!] \Rightarrow \overleftarrow{P}[\![B]\!]$$

$$[\![\cdot]\!] = \mathbf{1} \qquad [\![\Gamma, x : A]\!] = [\![\Gamma]\!] \times [\![A]\!]$$

$$[\![x]\!]^{\mathrm{val}} = \pi_x \qquad [\![a]\!]^{\mathrm{val}} = \lambda\gamma.\, K(a)$$
$$[\![\langle v, w\rangle]\!]^{\mathrm{val}} = \langle [\![v]\!]^{\mathrm{val}}, [\![w]\!]^{\mathrm{val}}\rangle \qquad [\![\mathrm{in}\, v]\!]^{\mathrm{val}} = \xi^{-1} \circ [\![v]\!]^{\mathrm{val}}$$
$$[\![\lambda x.\, t]\!]^{\mathrm{val}} = \lambda[\![t]\!]$$

$$[\![v]\!]\,\gamma = \overleftarrow{\eta}_{[\![A]\!]}([\![v]\!]^{\mathrm{val}}\,\gamma) \qquad [\![\mathrm{next}\, t]\!]\,\gamma = \overleftarrow{P}(\mathrm{next}_{[\![A]\!]})([\![t]\!]\,\gamma)$$
$$[\![t \circledast s]\!]\,\gamma = [\![t]\!]\,\gamma \overleftarrow{\gg\!=} \lambda f.\, [\![s]\!]\,\gamma \overleftarrow{\gg\!=} \lambda x.\, \mathrm{mon}\, f\, x \qquad [\![\mathrm{fix}\, x.\, t]\!]\,\gamma = \mathrm{l\ddot{o}b}\big(\lambda x.\, \iota(x) \overleftarrow{\gg\!=} \lambda y.\, [\![t]\!]\,(\gamma, y)\big)$$
$$[\![\mathrm{in}\, t]\!]\,\gamma = \overleftarrow{P}(\xi^{-1})([\![t]\!]\,\gamma) \qquad [\![\mathrm{out}\, t]\!]\,\gamma = \overleftarrow{P}(\xi)([\![t]\!]\gamma)$$
$$[\![\mathrm{normal}]\!]\,\gamma = \lambda K(N) \qquad [\![t\, s]\!]\,\gamma = [\![t]\!]\,\gamma \overleftarrow{\gg\!=} \lambda f.\, [\![s]\!]\,\gamma \overleftarrow{\gg\!=} \lambda x.\, f\, x$$
$$[\![\langle t, s\rangle]\!]\,\gamma = [\![t]\!]\,\gamma \overleftarrow{\gg\!=} \lambda x.\, [\![s]\!]\,\gamma \overleftarrow{\gg\!=} \lambda y.\, \eta(x, y) \qquad [\![\mathrm{fst}\, t]\!]\,\gamma = \overleftarrow{P}(\pi_1)([\![t]\!]\,\gamma)$$
$$[\![\mathrm{snd}\, t]\!]\,\gamma = \overleftarrow{P}(\pi_2)([\![t]\!]\,\gamma)$$