Problems with Fixpoints of Polynomials of Polynomials

Cécilia Pradic
Swansea University, UK

Ian Price
Swansea University, UK

Abstract

Motivated by applications in computable analysis, we study fixpoints of certain endofunctors over categories of containers. More specifically, we focus on fibred endofunctors over the fibrewise opposite of the codomain fibration that can be themselves be represented by families of polynomial endofunctors. In this setting, we show how to compute initial algebras, terminal coalgebras and another kind of fixpoint ζ. We then explore a number of examples of derived operators inspired by Weihrauch complexity and the usual construction of the free polynomial monad.

We introduce ζ-expressions as the syntax of μ-bicomplete categories, extended with ζ-binders and parallel products, which thus have a natural denotation in containers. By interpreting certain ζ-expressions in a category of type-2 computable maps, we are able to capture a number of meaningful Weihrauch degrees, ranging from closed choice on $\{0, 1\}$ to determinacy of infinite parity games, via an "answerable part" operator.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Logic and verification; Theory of computation → Constructive mathematics; Theory of computation → Recursive functions; Theory of computation → Automata over infinite objects; Theory of computation → Abstraction

Keywords and phrases Polynomial functors, containers, fixpoints, Weihrauch complexity

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.77

Related Version *Full Version with Appendices*: `https://arxiv.org/abs/2601.15420`

Acknowledgements We thank Arno Pauly and Manlio Valenti for discussions on Weihrauch reducibility, and Zeinab Galal for discussion on fibred fixpoints and references. We also thank the anonymous reviewers for their comments that improved this version of the paper.

1 Introduction

1.1 Internal families, polynomial functors and containers

The fundamental notion of a family of types $(A_i)_{i:I}$ can be interpreted in any category $\mathcal{C}$ with finite limits. Under this interpretation, morphisms $A \to I$ are regarded as I-indexed families and taking a pullback along $f : J \to I$ corresponds to computing a reindexing $(A_{f(j)})_{j:J}$. The most natural category whose objects are families internal to $\mathcal{C}$ is $\mathcal{C}^{\to}$, which has commuting squares for morphisms.

$$\begin{aligned} \varphi :&\quad I \longrightarrow J \\ \psi :&\quad \prod_{i:I} \left(A_i \to B_{\varphi(i)}\right) \end{aligned} \qquad\qquad \begin{tikzcd} A \arrow[r, "\psi"] \arrow[d, "P"'] & B \arrow[d, "Q"] \\ I \arrow[r, "\varphi"'] & J \end{tikzcd}$$

The codomain functor $\mathrm{cod} : \mathcal{C}^{\to} \to \mathcal{C}$ is then a Grothendieck fibration corresponding to the $\mathcal{C}$-indexed category $({}^{\mathcal{C}}/_{I})_{I:\mathsf{Obj}(\mathcal{C})}$. This is the starting point of the rich theory of fibrations and semantics of type theory [32]. That being said, another important notion of morphisms between families is the one induced by the fibrewise opposite of the codomain fibration[1].

$$\begin{aligned} \varphi :&\quad I \longrightarrow J \\ \psi :&\quad \prod_{i:I} \left(B_{\varphi(i)} \to A_i\right) \end{aligned} \qquad\qquad \begin{tikzcd} A \arrow[d, "P"'] & \cdot \arrow[l, "\psi"'] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "Q"] \\ I \arrow[r, equal] & I \arrow[r, "\varphi"'] & J \end{tikzcd}$$

Internal families and these morphisms make up a category $\mathsf{Cont}(\mathcal{C})$, whose objects are sometimes called polynomials or *containers*. The category $\mathsf{Cont}(\mathcal{C})$ and its dependent versions have been studied from a number of viewpoints in mathematics and computer science:

As functors. Assuming $\mathcal{C}$ has Π types, the container $(A_i)_{i:I}$ induces an endofunctor over $\mathcal{C}$:

$$X \qquad \longmapsto \qquad \sum_{i:I} X^{A_i}$$

Endofunctors of this shape are called *polynomial endofunctors*, and we have that strong natural transformations are in one-to-one correspondence with container morphisms. Polynomial functors have excellent structural properties and have thus proved useful in many areas of logic and computer science [25], including semantics [26, 43]. It is also often the case that polynomial endofunctors admit initial algebras and terminal coalgebras [41, 59] (which are categorifications of the notions of least and greatest fixpoints). Intuitively, the initial algebra of $(A_i)_{i:I}$ consists well-founded I-labelled trees where A_i indexes the children of i-labelled nodes. Having initial algebras for polynomial functors allows one to interpret inductive types (also called $\mathcal{W}$-types) of Martin-Löf type theory. Dually the terminal coalgebras ($\mathcal{M}$-types) interpret coinductive types [1].

As polymorphic types. The terminology "containers" was introduced in the context of functional programming [2] to formalize a notion of polymorphic datatypes such as $\alpha \mapsto \mathsf{List}(\alpha)$. The idea there is that a datatype polymorphic in some parameter α is concretely determined by a type of *shapes* I and an I-indexed families of holes in which to put labels taken from α. For instance, the container for $\mathsf{List}(\alpha)$ in Set is the family $(\{0, \dots, n-1\})_{n:\mathbb{N}}$ which says that for every $n : \mathbb{N}$, there is a "generic list" of length n with n holes. Then container morphisms correspond to polymorphic functions that describe an output shape solely in terms of the input shape and provenance of the labels of each hole of the output, which must come from holes in the input. At the level of semantics, instantiating such a polymorphic definition at α then corresponds to applying the corresponding polynomial functor to α.

As problems. Another viewpoint on internal families $(A_i)_{i:I}$ is that they may be viewed as a formalism for input-output specifications which we may call *problems*. The idea is that members $i : I$ are inputs or *questions* and A_i consists of *answers* to i. Then a full solution to such a problem $P : A \to I$ is a section $s : I \to A$. Even if P is "surjective", such full solutions have no reason to exist in general; in our examples of interest it will be the case

[1] See [54, §1 & 5] for a formal definition; informally, this is one of the few constructions which is easier to define from the point of view of indexed categories $\mathcal{C} \to \mathsf{Cat}$, where it simply corresponds to composing with the endofunctor $\mathrm{op} : \mathsf{Cat} \to \mathsf{Cat}$ that reverses all arrows.

Notation	Operation on problems	Fixpoint expression
P^*	finite parallelization	$\mu X.\, \mathbf{I} \,+\, P \otimes X$
$\widehat{P}$	infinite parallelization	$\zeta X.\, P \otimes X$
$\widetilde{P}$	ask ω questions get one answer	$\nu X.\, P \times X$
$P^{\diamond}$	finite iterations (free monad over P)	$\mu X.\, \mathbf{I} + X \star P$
P^{∞}	ω-loop	$\zeta X.\, X \star P$

Figure 1 Endofunctors $\mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$ defined via fixpoint equations. Since the definition of $\star$ requires $\mathcal{C}$ to be a lccc, the last two definitions are not quite the same as the operators on Weihrauch degrees (see [49, §4.4]).

that $\mathcal{C}$ contains only continuous maps and that P typically has no continuous sections. It is then remarkable that container morphisms $P \to Q$ can be regarded as reductions that solve P using exactly one oracle call to Q, a restriction reminiscent of linear logic. This notion is extensively studied in type 2 computability[2], where the objects of $\mathcal{C}$ are subspaces of $2^{\mathbb{N}}$, morphisms are represented by Turing machines operating on infinite bit strings and problems are often given by non-constructive $\forall\exists$ theorems of second-order arithmetic. In this setting, container morphisms are called *Weihrauch reductions* [10] and capture comparative hardness, often in a more fine-grained way than entailment does in reverse mathematics [53]. This viewpoint is also helpful when working with modalities induced by containers [51, 4] in the context of synthetic computability theory [34, 55, 62].

1.2 Fixpoints over the category of containers

Taking the last perspective on containers, their excellent structural properties translate to a multitude of operators to combine problems that find many concrete applications [14]. In particular, we have tensorial products $\otimes$ and $\star$ for the parallel and sequential composition of problems, and the coproduct $P + Q$ allows one to ask a question to either P or Q and get a relevant answer. To relax the linear nature of container morphisms, it is natural to ask for fixpoints of such operators. For instance, a reduction from Q to $P^{\diamond}$, the carrier of the initial algebra of the endofunctor $X \mapsto \mathbf{I} + X \star P$ over $\mathsf{Cont}(\mathcal{C})$, captures the idea of being able to compute solutions to Q questions if we are allowed to use P as an oracle an arbitrary finite number of times. It is not only least fixpoints that are of interest: ω-parallelization is a standard tool [13] and sequential iterations of length ω have recently been introduced [11, 15]. However, in spite of their natural definitions, they are *not* greatest fixpoints! For instance, in the case of the terminal coalgebra of $X \mapsto X \otimes P$, the shape of its carrier consists of streams of P-questions, but it defines no possible answers whatsoever to such streams of questions; ω-parallelization in Weihrauch reducibility contrastingly allows for relevant streams of P-answers.

We are thus compelled to look for a more structural understanding of such fixpoints. In this paper, we focus on fixpoints for endofunctors $F : \mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$ that happen to be components of fibred endofunctors over $\mathrm{cod}^{\mathrm{op}}$; that is, those functors such that there is a (uniquely determined) $F_0 : \mathcal{C} \to \mathcal{C}$ such that $\mathrm{cod}^{\mathrm{op}} \circ F = F_0 \circ \mathrm{cod}^{\mathrm{op}}$. Intuitively it means

[2] But not exclusively there; see [28] for a similar set-up in algebraic complexity theory. This is also similar to intuitions in terms of games given in [29].

that the question space of $F(P)$ only depends on the question space of P and that F induces functors $F_X : \left(\mathcal{C}/X\right)^{\mathrm{op}} \to \left(\mathcal{C}\Big/F_0(X)\right)^{\mathrm{op}}$ at the level of answers. This suggests the following generic recipe to compute a terminal coalgebra of F: first compute a terminal F_0-coalgebra $(\nu F_0, c_{\nu F_0})$, and then, compute a terminal coalgebra for $(c^*_{\nu F_0})^{\mathrm{op}} \circ F_{\nu F_0} : \left(\mathcal{C}/\nu F_0\right)^{\mathrm{op}} \to \left(\mathcal{C}/\nu F_0\right)^{\mathrm{op}}$, that is, an *initial* algebra for $c^*_{\nu F_0} \circ F^{\mathrm{op}}_{\nu F_0} : \mathcal{C}/\nu F_0 \to \mathcal{C}/\nu F_0$. If we thus assume that $\mathcal{C}$ has indexed $\mathcal{W}$-types and $\mathcal{M}$-types and that F_0, as well as every F^{op}_X, are polynomials, then this construction shows us that a terminal F-coalgebra exists. We can also build initial F-algebras in a completely dual manner, by first computing the initial F_0-algebra $(\mu F_0, a_{\mu F_0})$, and then a terminal coalgebra for $(a^{-1}_{\mu F_0})^* \circ F_{\mu F_0}$. More excitingly, we may also mix the two approaches by taking terminal coalgebras for both F_0 and $c^*_{\nu F_0} \circ F^{\mathrm{op}}_{\nu F_0}$, which yields a container ζF that comes with an isomorphism $F(\zeta F) \cong \zeta F$ via Lambek's lemma. This middling fixpoint construction is the one that enables us to capture ω-iterations and parallelizations without trivializing the space of answers (see Figure 1).

Asssuming that $\mathcal{C}$ is lextensive, has indexed $\mathcal{W}$-types and $\mathcal{M}$-types, a niceness notion for endofunctors $\mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$ (i.e. functors we call *fibred polynomial* in Definition 13), an existence theorem for our fixpoints (Theorem 18) and the knowledge that product, coproduct and $\otimes$ are nice (Lemma 14), we have a supply of containers by interpreting the following generalization of the syntax of μ-bicomplete categories [52, 30], which we call *ζ-expressions*:

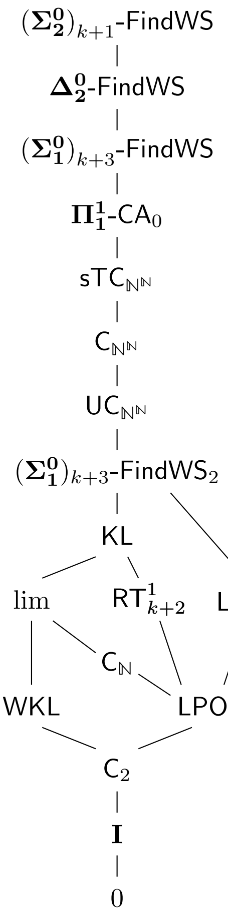
$$E, E' \;::=\; X \mid \mu X.\, E \mid \zeta X.\, E \mid \nu X.\, E \mid E \times E' \mid E \otimes E' \mid E + E'$$

Much like in case of the free μ-bicomplete category, interpreting this syntax does yield some non-trivial results that can be related to strategies in parity games. More specifically, we shall see that such expressions E can be algorithmically turned into regular infinite tree languages L_E describing parity games (as in [5]). Then E gets interpreted as the problem $\langle\!\langle E \rangle\!\rangle$, where a question is a tree $t \in L_E$ which is answered by a winning strategy in the relevant game for the player with, say, the even objective (Theorem 27).

In spite of the non-trivial combinatorics, the degree of the problem $\langle\!\langle E \rangle\!\rangle$ is always trivial. In many situations, we do obtain problems with interesting question-answer pairs that also contain questions which cannot be answered, which means that the problem typically has maximal degree (the terminal object in $\mathsf{Cont}(\mathcal{C})$ is the trivial family $(0)_{i:1}$). This means that we have some way to retrieve interesting information if we have the ability to filter out these problematic unanswerable questions. In the context of Weihrauch reducibility, we can do so via an operator we call Ans (Definition 9). Applying Ans to denotations of ζ-expressions then allows us to describe a healthy number of benchmark Weihrauch problems that include the computation of limits, the infinite pigeonhole principle, closed choice principles on subspaces of $\mathbb{N}^{\mathbb{N}}$ and determinacy for $\boldsymbol{\Delta}^0_1$ and $\boldsymbol{\Delta}^0_2$ objectives (see Figure 2). There are, of course, limits to how many degrees can be expressed this way. The automata-theoretic characterization of $\langle\!\langle E \rangle\!\rangle$ mean we can only get 0 or pointed degrees that are below determinacy for parity games (Proposition 40); this however does not rule out most named degrees studied in Weihrauch complexity so far. We conjecture that some natural degrees, such as RT^2_2 (Ramsey's theorem for pairs), are not equivalent to any $\mathsf{Ans}(\langle\!\langle E \rangle\!\rangle)$.

1.3 Related and further work

Structural aspects of Weihrauch reducibility. The present paper obviously draws on efforts to generalize and "categorify" Weihrauch reducibility and some of its variants [7, 58, 40], which have only recently identified containers as a possible unifying framework [3, 49]. As a result, the algebraic theory of operators on Weihrauch degrees developed separately from the



Weihrauch degree D	ζ-expression E with $\mathsf{Ans}(\langle\!\langle E \rangle\!\rangle) \equiv D$
$(\mathbf{\Sigma}^0_2)_{2k}$-FindWS	$\zeta X_{2k}.\nu X_{2k-1}.\dots.\zeta X_1.\nu X_0.\ \widehat{\widetilde{\sum_{i=0}^{2k}}} X_i$
$\mathbf{\Delta}^0_2$-FindWS	$\mu Z.\ \zeta X.\ \widehat{\widetilde{\widehat{X}}} + \left(\nu Y.\ \widehat{\widetilde{\widehat{Y}}} + Z\right)$
$\mathsf{sTC}_{\mathbb{N}^\mathbb{N}}$	$\left(\nu X.\widehat{X + \mathbf{I}}\right) \times \left(\zeta X.\widetilde{X + 1}\right)$
$\mathsf{C}_{\mathbb{N}^\mathbb{N}}$	$\zeta X.\ \widetilde{X + 1}$
$\mathbf{\Delta}^0_1$-FindWS	$\mu X.\ \widehat{\widetilde{\widehat{X}}} + \mathbf{I} + 1$
KL	$\zeta X.\ (\nu Y.\ X + Y) \times (\nu Y.\ X + Y)$
lim	$(\nu X.\ X + \widehat{\mathbf{I}) \times (}\zeta X.\ X + 1)$
WKL	$\zeta X.\ 1 + X \times X$
RT^1_k	$\underbrace{(\zeta X.\nu Y.\ X + Y) \times \dots}_{k\text{ times}}$
LPO′	$(\nu X.\ \zeta Y.\ X + Y) \times (\zeta X.\ \nu Y.\ X + Y)$
$\mathsf{C}_\mathbb{N}$	$\zeta \widetilde{X.\ X} + 1$
LPO	$(\nu X.\ X + \mathbf{I}) \times (\zeta X.\ X + 1)$
C_k	$\underbrace{(\zeta X.\ X + 1) \times \dots}_{k\text{ times}}$
$\mathbf{I}$	$\zeta X.\ X$
0	$\mu X.\ X$

Figure 2 Problems from the Weihrauch lattice as the answerable part of the interpretation of some ζ-expressions in represented spaces. On the left-hand side is the part of the Weihrauch lattice we capture, where all inequalities are known to be strict below $\mathbf{\Pi}^1_1$-CA_0. $\mathbf{\Gamma}$-FindWS is the answerable part of the problem where a question is code for a $\mathbf{\Gamma}$-set A which is answered by a winning strategy for the player with winning condition A in a Gale-Stewart game over $\mathbb{N}^\mathbb{N}$.

literature on polynomial functors, which led to an apparent duplication of results pertaining to basic properties of sequential composition for instance [14, 61]. Although some care has to be taken as there is a mismatch between the usual sequential composition of polynomial functors and the composition of Weihrauch problems. This is due to the fact that computability theorists always work *intensionally* with Weihrauch problems; they consider containers over a category $\mathsf{rpReprSp}$ which is only weakly (locally) cartesian closed. A consequence of this is that the sequential composition in $\mathsf{Cont}(\mathsf{rpReprSp})$ is only a quasifunctor. While it ends up being closely related to the classical composition of polynomial endofunctors over a supercategory ReprSp, more work is needed to definitely capture what is going on in an abstract setting; see [49, §4] for more details.

Fixpoints of endofunctors over containers. Fixpoints and containers enjoy a strong connection, as polynomial functors do often admit initial algebras and terminal coalgebras [41, 59, 2, 1]. But we are not aware of any systematic study of fixpoints of endofunctors over $\mathsf{Cont}(\mathcal{C})$ that focus on fibred functors comprised of polynomials like we consider here (rather than fixpoints of polynomial enfofunctors over $\mathcal{C}$). That is not to say that such fixpoints have never been considered in the literature on polynomial functors and containers: for instance, the free monad over a polynomial P is computed the initial algebra for $X \mapsto \mathbf{I} + X \star P$ (and serependipitously coincide with the idea of finite iterations of a problem).

While we do focus on those fibred polynomial endofunctors, we can note that we only use the "polynomial" aspect to be able to use the assumption that $\mathcal{C}$ has fixpoints of polynomial functors to begin with. Theorem 18 otherwise only requires the fibredness of the endofunctor $F : \mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$ and could be adapted to settings where fixpoints can be obtained by other means (e.g. via more powerful fixpoint theorems in categories of domains or in settings with impredicative quantifications). This could plausibly lead to the study of more interesting fixpoints, such as countable ordinal iterations of a problem discussed in [48]. Lifting the restriction that F be fibred does break Theorem 18. We only provide a single example of such a functor which does have a sensible initial algebra (Proposition 23).

The definition of fibred polynomial functor that we adopt is useful insofar as it captures examples relevant to us and makes Theorem 18 go through, but it is unclear to us whether it is a rather ad-hoc notion or if there is a more conceptual grounding to this notion, possibly related to generalizing the pseudo-monad Cont to 2-categories beyond Cat.

Finally, the idea of computing (co)algebras over total categories of fibrations, by considering the base and then the fibers, has previously appeared in the literature for posetal fibrations [9, 27, 23]. Here we consider a specific class of non-posetal fibrations, and expect that our straightforward approach generalizes further to other classes of fibrations. However, we are currently not aware of a useful general theorem along those lines.

Weihrauch reductions and ζ-expressions. Aside from the problems $\mathbf{\Gamma}$-FindWS (which essentially correspond to theorems stating that infinite games over $\mathbb{N}^{\mathbb{N}}$ with winning conditions in $\mathbf{\Gamma}$ are determined) with $\mathbf{\Gamma}$ above $(\mathbf{\Sigma}^0_1)_2$, all reductions and non-reductions between problems featured in Figure 2 appear in the literature [13, 37, 39]. To the best of our knowledge, problems around $\mathbf{\Pi}^1_1$-CA_0 or above have rarely been investigated in the Weihrauch lattice thus far (exceptions include [21, 19]). In contrast, determinacy statements have been extensively studied in reverse mathematics since Friedman's celebrated result that Borel determinacy is unprovable without the powerset axiom [24]; references most relevant to the subclasses of parity games appearing in Figure 2 include [56, 57, 42, 35, 47]. Work on the topological complexity of game quantifiers also provides some insights on the logical complexity of determinacy statements [16, 5]. That being said, those results do not straightforwardly translate to results on the Weihrauch degrees $\mathbf{\Gamma}$-FindWS; we therefore limit ourselves to observing strictness of reductions for the difference hierarchy above $\mathbf{\Sigma}^0_1$ using links to finite iterations of $\mathbf{\Pi}^1_1$-CA_0 already present in [56], and leave further studies of $\mathbf{\Gamma}$-FindWS for higher $\mathbf{\Gamma}$s to the future.

While ζ-expressions augmented with Ans are remarkably expressive, the non-functoriality of Ans makes this formalism difficult to integrate in a fixpoint logic to compositionally reason about Weihrauch problems (unlike the equational axiomatizations proposed in [45, 50]). We end by asking whether equivalence of the answerable parts of two ζ-expressions is decidable (Question 41).

1.4 Plan of the paper

Section 2 is dedicated to introducing basic notions and notations pertaining to category theory, containers, type 2 computability, and (co)inductive types (including a sketch of the syntax of μ-bicomplete categories and its automata-theoretic interpretation). Section 3 is chiefly concerned with the existence theorem for the three aforementioned fixpoints for fibred polynomial endofunctors Theorem 18 and its parametric version Theorem 22; it can be read independently of any material on computability. Section 4 then discusses in more detail the operators described in Figure 1 as well as all variations obtained by taking other types of fixpoints, which we hope give a number of concrete examples to see Theorem 18 in action. Finally, Section 5 focuses more specifically on concrete containers definable in the context of Weihrauch reducibility via Ans and ζ-expressions described in Figure 2. Due to the number of examples, it relies more heavily on the material on type 2 computability as well as intuitions built by way of previous examples and the proof of Theorem 27 explaining how to concretely compute the denotation of closed ζ-expressions.

2 Background

2.1 Category theory

We assume familiarity with the basics of category theory [38]. We write $\langle f_1, f_2\rangle : Z \to A_1 \times A_2$ for the pairing of $f_i : Z \to A_i$ ($i \in \{1,2\}$), and $\pi_i : A_1 \times A_2 \to A_i$ for projections ($i \in \{1,2\}$). Dually we write $[f_1, f_2] : A_1 + A_2 \to Z$ and $\mathsf{in}_i : A_i \to A_1 + A_2$ for the copairing and coprojections. If k is a natural number, we may regard it as the k-fold coproduct of 1 in any category.

▶ **Definition 1** ([18]). *A category with finite sums and products is called* extensive *when the canonical functor* $+ : \mathcal{C}/_A \times \mathcal{C}/_B \to \mathcal{C}\big/_{(A+B)}$ *is an isomorphism. We say a category is* lextensive *if it has all finite limits, all finite coproducts and is extensive.*

Henceforth, all categories in sight shall be lextensive. It means in particular that they will be distributive and that $\mathcal{C}/_k \cong \mathcal{C}^k$.

We also assume that the reader is familiar with the notion that morphisms $f : A \to I$ in a category $\mathcal{C}$ can be regarded as *I-indexed families internal to $\mathcal{C}$*, the idea being that for $\mathcal{C} = \mathsf{Set}$, such an f represents the family $(f^{-1}(i))_{i\in I}$. All lextensive categories support reindexing of such families through pullback functors $f^* : \mathcal{C}/_J \to \mathcal{C}/_I$, and have internal sums via its left adjoint $- \circ f = \Sigma_f : \mathcal{C}/_I \to \mathcal{C}/_J$. Lextensivity further ensures that external finite sums and internal sums socialize harmoniously. We say that a category is *locally cartesian closed* when it has all finite limits and the pullback functors admit a right adjoint Π_f that interprets internal products. We will often refer to extensive locally cartesian closed categories as *elcccs* in the sequel.

▶ Remark 2. Extensional Martin-Löf type theory (EMLTT) with $\prod, \sum, 0, 1, +$ can be interpreted in (any splitting of) the codomain fibration over an elcccs $\mathcal{C}$ [20]. For some proofs, we will thus freely use EMLTT as an internal language for such $\mathcal{C}$.

We now discuss inductive and coinductive types in categories, which are modelled by (co)inductive (co)algebras. Given a functor $F : \mathcal{C} \to \mathcal{C}$, an F-algebra is a pair (A, a) where A is an object of $\mathcal{C}$, its *carrier*, and a is a morphism $F(A) \to A$. F-algebra morphisms $(A, a) \to (A', a')$ are morphisms $f : A \to A'$ such that $f \circ a = a' \circ F(f)$. Initial F-algebras (A, a) (and, similarly, terminal F-coalgebras) are fixpoints of F in the sense that a is always an isomorphism; this is called *Lambek's lemma*. We will sometimes write $(\mu F, \mathfrak{in})$ for a fixed initial F-algebra. Carriers of initial algebras $(\mathbb{N}, [0, \mathsf{S}] : 1 + \mathbb{N} \to \mathbb{N})$ for the functor $X \mapsto 1 + X$ are called *natural number objects*. Dually, carriers of terminal coalgebras $(\mathbb{N}_\infty, \mathsf{pred} : \mathbb{N}_\infty \to 1 + \mathbb{N}_\infty)$ are called *conatural number objects*. The coalgebra structure map $\mathbb{N}_\infty \to 1 + \mathbb{N}_\infty$ essentially tells us whether its input is zero, and if it is not, returns a

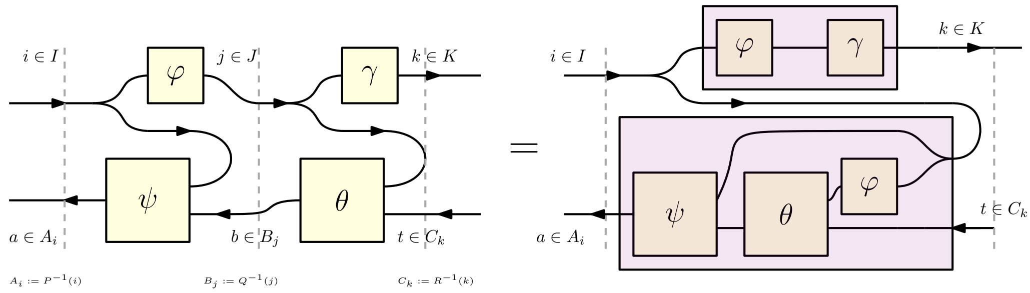


Figure 3 Informal diagram representing the composition of (unary 1-dimensional) container morphisms $(\gamma, \varphi) \circ (\psi, \theta)$. Note that this would be a sound string diagram in a category with cartesian products if we were working with non-dependent families (seen as a pair of objects); it would then be formally interpreted by $(\gamma \circ \psi, \psi \circ \langle \pi_1, \theta \circ (\varphi \times \mathrm{id})\rangle)$.

predecessor. We may define a "point at infinity" $\infty : 1 \to \mathbb{N}_\infty$ as the coalgebra morphism $(1, \mathsf{in}_2 : 1 + 1)$ to the conatural number object. In Set, it can be shown that $\mathbb{N}_\infty$ consists of a copy of the natural number plus this point at infinity; in topological spaces, $\mathbb{N}_\infty$ can be regarded as the subspace of $2^{\mathbb{N}}$ consisting of monotone maps.

2.2 Polynomial functors and containers

Let $\mathcal{C}$ be a category with chosen pullbacks and T and U two objects of $\mathcal{C}$. A container from T to U is a triple of morphisms (P, t, u) fitting in a diagram of the following shape.

$$T \xleftarrow{t} A \xrightarrow{P} I \xrightarrow{u} U$$

We call I the *shape* of the container, A its *directions*, T is its *arity* and U its *dimension*. Containers between T and U are objects of a category $\mathsf{Cont}(\mathcal{C})(T, U)$ such that a morphism $(P, t, u) \to (Q, t', u')$ is determined by a pair (φ, ψ) making the following commute:

$$\begin{tikzcd}
 & A \arrow[r, "P"] \arrow[dl, bend right, "t"'] & I \arrow[d, equal] \arrow[dr, bend left, "u"] & \\
T & \cdot \arrow[u, "\psi"] \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & I \arrow[d, "\varphi"] & U \\
 & B \arrow[r, "Q"'] \arrow[ul, bend left, "t'"] & J \arrow[ur, bend right, "u'"'] &
\end{tikzcd}$$

We call φ the *forward* part of the morphism and ψ the *backward* part. The identity map $(P, t, u) \to (P, t, u)$ is given $(\mathrm{id}_I, \mathrm{id}_A)$ and the composition is definable using pullbacks in $\mathcal{C}$; rather than spelling out the details, we refer the reader to Figure 3 for an intuition of how composition works and to [60] for formal details.

We will often focus our discussions on containers of dimension 1 or containers which have both dimension and arity 1. To make notations lighter, we will omit the third component of containers of dimension 1 (e.g. (P, t) should be regarded as a container of dimension 1) and simply identify containers of dimension and arity 1 as morphisms (e.g. P is a container). We set $\mathsf{Cont}(\mathcal{C}) := \mathsf{Cont}(\mathcal{C})(1, 1)$. In the sequel, containers are considered to be of dimension and arity 1 unless otherwise indicated. For such containers $P : A \to I$, we will also sometimes write $\mathsf{shape}(P)$ for its shape (I in this example) and P_{dir} for its directions (A here). shape

Cartesian product $P \times Q$	Coproduct $P + Q$	Tensor product $P \otimes Q$	Sequential product $Q \star P$
$\begin{array}{c} A \times J + I \times B \\ \big\downarrow{\scriptstyle [P \times \mathrm{id}_J, \mathrm{id}_I \times Q]} \\ I \times J \end{array}$	$\begin{array}{c} A + B \\ \big\downarrow{\scriptstyle P+Q} \\ I + J \end{array}$	$\begin{array}{c} A \times B \\ \big\downarrow{\scriptstyle P \times Q} \\ I \times J \end{array}$	$\begin{array}{c} \sum\limits_{a:A} B^{A_{P(a)}} \\ \big\downarrow \\ \sum\limits_{i:I} J^{P^{-1}(i)} \end{array}$

Figure 4 Definitions of the tensorial products on $\mathsf{Cont}(\mathcal{C})$; we allow ourself to use the internal language of $\mathcal{C}$ as a locally cartesian closed category for $\star$ and let the reader guess what is the unique possibility for the morphism.

extends to a functor $\mathsf{Cont}(\mathcal{C}) \to \mathcal{C}$ by considering the map $(\varphi, \psi) \mapsto \varphi$ on morphisms. In fact, shape is a Grothendieck fibration, and it is straightforward to check that it is exactly the fibrewise opposite of the codomain fibration $\mathrm{cod} : \mathcal{C}^{\to} \to \mathcal{C}$ (see [54, §1 & §5] for details). So given a morphism $P \to Q$ in $\mathsf{Cont}(\mathcal{C})$ given by (φ, ψ), we will call it *cartesian* or *horizontal* if ψ is an isomorphism and *vertical* if $\varphi = \mathrm{id}$.

If $\mathcal{C}$ is locally cartesian closed, containers (P, t, u) with arity T and dimension U induce functors $[\![P, t, u]\!] := \Sigma_u \circ \Pi_P \circ t^* : \mathcal{C}/_T \to \mathcal{C}/_U$. In Set, up to the usual equivalences $\mathsf{Set}/_T \cong \mathsf{Fam}(T)$ and $\mathsf{Set}/_U \cong \mathsf{Fam}(U)$ between slice categories and indexed families and regarding (P, t, u) as the triply-indexed family $(A_{i,\alpha,\beta})_{\alpha \in U, i \in I_\alpha, \beta \in T}$ (by taking $I_\alpha = u^{-1}(\alpha)$ and $A_{i,\alpha,\beta} = P^{-1}(i) \cap t^{-1}(\beta)$), the container (P, t, u) induces the functor

$$\begin{array}{ccc} \mathsf{Fam}(T) & \longrightarrow & \mathsf{Fam}(U) \\ (X_\beta)_{\beta \in T} & \longmapsto & \left(\sum\limits_{i \in I_\alpha} \prod\limits_{\beta \in T} (A_{i,\alpha,\beta} \to X_\beta) \right)_{\alpha \in U} \end{array}$$

In the one-dimensional case, (P, t) induces a functor $[\![P, t]\!] : \mathcal{C}/_T \to \mathcal{C}$, and in the one-dimensional unary case, P induces an endofunctor $[\![P]\!] : \mathcal{C} \to \mathcal{C}$. The $\mathcal{C} = \mathsf{Set}$ case then simplifies to the following expression reminiscent of power series:

$$\begin{array}{ccc} \mathsf{Set} & \longrightarrow & \mathsf{Set} \\ X & \longmapsto & \sum\limits_{i \in I} X^{A_i} \end{array}$$

▶ **Proposition 3** (Corollary of [25, Theorem 2.12]). *$[\![-]\!]$ extends to a functor from $\mathsf{Cont}(\mathcal{C})(I, J)$ to the the category of strong functors $\mathcal{C}/_I \to \mathcal{C}/_J$ and strong natural transformations. Furthermore, $[\![-]\!]$ is full and faithful.*

When $\mathcal{C}$ is lextensive, $\mathsf{Cont}(\mathcal{C})$ is also lextensive and features two additional monoidal products $\otimes$ and $\star$ outlined in Figure 4. The parallel product is essentially the componentwise product of families, while the sequential product is more involved:

$$\begin{array}{ccc} (A_i)_{i:I}, (B_j)_{j:J} & \longmapsto & (A_i \times B_j)_{(i,j):I \times J} \\ (A_i)_{i:I}, (B_j)_{j:J} & \longmapsto & \left(\sum\limits_{b:B_j} A_{f(b)} \right)_{(j,f):\sum\limits_{j:J} I^{B_j}} \end{array}$$

The sequential product we will also sometimes call "composition product" since we have $[\![P \star Q]\!] = [\![Q]\!] \circ [\![P]\!]$. It is the only product in Figure 4 which does not come with a symmetry. All monoidal products come with a unit, with $\star$ and $\otimes$ having the same unit $\mathbf{I}$ with $\mathsf{shape}(\mathbf{I}) = \mathbf{I}_{\mathrm{dir}} = 1$. When $\mathcal{C}$ is additionally locally cartesian closed, it is symmetric monoidal closed for both $\times$ and $\otimes$.

2.3 Weihrauch reducibility and containers

One perspective on (unary and 1-dimensional) containers is that they can be viewed as *problems* in the following way: the shape I of a container $P : A \to I$ should be regarded as a space of *questions*, the directions A should be regarded as *answers* and P specifies what are the answers $P^{-1}(i)$ to some question $i : I$. Computability tells us that some problems cannot be solved computably nor continuously, in which case we are interested in quantifying how large the obstruction is via notions of *reductions*. Container morphisms constitute one such notion of reduction: a morphism $P \to Q$ gives us a forward map φ turning a P-question into a Q-question and the backward map ψ turns a P-question i and a Q-answer to $\varphi(i)$ into a P-answer to i. Intuitively, this means solving P with one, no more, no less, oracle call to Q. In light of this, the monoidal operators from Figure 4, and their units, admit natural interpretations as problem transformers.

If we are strictly interested in the power of problems, we may want to talk about the reduction preorder giving $P \leq Q$ if and only if there exists a morphism $P \to Q$ in $\mathsf{Cont}(\mathcal{C})$. We will freely write $\leq$ for this preorder in the sequel, $\equiv$ for the associated equivalence relation over containers, and call its equivalence classes *degrees*.

To develop examples for this paper, we will be looking at the particular instantiation of this framework in the context of Weihrauch reducibility, previously discussed in [49]. Up to some caveats, this amounts to picking a category for type 2 computability that matches the practice in computable analysis; this will be the category of represented spaces ReprSp.

A represented space (X, δ_X) is a set together with a partial surjection $\delta_X : 2^{\mathbb{N}} \rightharpoonup X$ called its *representation*. A map $f : X \to Y$ between two represented spaces (X, δ_X) and (Y, δ_Y) is called *computable* if it has a (type 2) computable realizer, that is, if there is a multitape Turing M such that, if given $p \in \mathrm{dom}(\delta_X)$ as an oracle, it will write all prefixes of an element of $\delta_Y^{-1}(f(\delta_X(p)))$ on a write-only output tape.

Represented spaces and computable maps form an elccc, which readers familiar with realizability will see to be equivalent the full subcategory $\mathsf{Mod}(\mathcal{K}_2^{\mathrm{rec}}, \mathcal{K}_2)$ of modest sets of the Kleene-Vesley topos studied extensively in [6].

Every Weihrauch problem P can be regarded as an object of $\mathsf{Cont}(\mathsf{ReprSp})$, and every Weihrauch reduction from P to Q correspond to a morphism of $\mathsf{Cont}(\mathsf{ReprSp})$. Let us give an example.

▶ **Definition 4.** *Informally* LPO *is the problem "given an infinite binary sequence $p \in 2^{\mathbb{N}}$, return a boolean saying whether $p = 0^\omega$ or not". Spelling out the formal details,* LPO *is the following container:*

- $\mathsf{shape}(\mathsf{LPO})$ *is Cantor space* $2^{\mathbb{N}}$
- $\mathsf{LPO}_{\mathrm{dir}}$ *is the coproduct*[3] $\{0^\omega\} \quad + \quad 2^{\mathbb{N}} \setminus \{0^\omega\}$.
- *the container is the copairing of the obvious inclusions.*

[3] It is *not* isomorphic to $2^{\mathbb{N}}$, as the first component is topologically isolated.

In the sequel, we will often simply state what problems are, and leave to the reader figuring out the details of the relevant container.

For any set A, we write A^* for the set of finite words over A. Let us write $\sqsubseteq$ for the prefix relation over finite and infinite words and, given any word w of length $\geq n$, write $w[n]$ for its restriction to its first n letters. Recall that (possibly countably-branching infinite) *trees* t can be regarded as maps $t : \mathbb{N}^* \to 2$ (which we sometimes confuse with subsets of $\mathbb{N}^*$) such that $t(\varepsilon) = 1$, which are prefix-closed ($t(un) = 1 \Rightarrow t(u) = 1$).

▶ **Definition 5.** KL *is the problem "given a finitely branching tree $t : \mathbb{N}^* \to 2$, find an infinite path through it".*

▶ **Example 6.** There is a container morphism $\mathsf{LPO} \to \mathsf{KL}$ defined as follows. The forward map φ takes as input a sequence $p \in 2^{\mathbb{N}}$ and outputs a countably branching tree $\varphi(p) : \mathbb{N}^* \to 2$. We may define it by cases:

- $\varphi(p)(0^n) = 1$ whenever 0^n is a prefix of p,
- $\varphi(p)((k+1)0^n) = 1$ whenever $0^k 1$ is a prefix of p (for any $n \in \mathbb{N}$),
- otherwise, $\varphi(p)(q) = 0$ for any other input $q \in \mathbb{N}^*$.

φ can be implemented by a type 2 Turing machine that works as follows. First the machine reads its second input $q \in \mathbb{N}^*$, and determines whether it is of shape 0^n, $(k+1)0^n$ or something else. In the last case it returns 0 immediately. In the first and second cases, it can return the correct result after reading off n and $k+1$ bits of p respectively.

Now the tree given by $\varphi(p)$ always has a single infinite branch: if $p = 0^\omega$, then the branch is 0^ω. If $p \neq 0^\omega$, then $\varphi(p)$ will have an infinite branch $(k+1)0^\omega$ where k is the position of the first non-zero bit of p; it may additionally have another finite branch when $k \neq 0$, but no further infinite branches.

So we have that $\varphi(p) \in \mathsf{shape}(\mathsf{KL})$, and $\varphi(p)$ has a unique infinite branch whose first element is 0 if and only if $p = 0^\omega$. Hence, one can define a backward map

$$\begin{array}{rlcl} \psi : & \sum\limits_{p \in 2^{\mathbb{N}}} \{r \in 2^{\mathbb{N}} \mid r \text{ is a branch of } p\} & \longrightarrow & \{0^\omega\} + 2^{\mathbb{N}} \setminus \{0^\omega\} \\ & (p, r) & \longmapsto & \begin{cases} \mathsf{in}_1(0^\omega) & \text{if } r_0 = 0 \\ \mathsf{in}_2(p) & \text{otherwise} \end{cases} \end{array}$$

so that (φ, ψ) is[4] a container morphism $\mathsf{LPO} \to \mathsf{KL}$.

Conversely, there is no container morphism $\mathsf{KL} \to \mathsf{LPO}$ for continuity reasons. Let us sketch the argument: if there were such a morphism (φ', ψ'), then φ' cannot be constant, as it would otherwise mean that KL has a continuous section. Hence, there is some tree t such that $\varphi'(t) \neq 0^\omega$, which has no infinite path starting with k. By continuity of ψ', there is a finite approximation t_f of t such that $\psi'(u, \mathsf{in}_2(\varphi'(u)))_0 = k$ for some k and all extensions u of t_f. This is a contradiction, as we can pick u with $u(k'0^n) = 1$ for $k' \in \mathbb{N} \setminus \{k\} \cup \mathrm{dom}(t_f)$, and $u(q) = 0$ for all other $q \notin \mathrm{dom}(t_f)$, which has no infinite path starting with k.

While every Weihrauch problem corresponds to a container over ReprSp, the converse is not true for two reasons. Firstly, the official definition of Weihrauch reducibility allows for reductions that may not be extensional, so that more problems are identified. To fix this, we may consider containers over the full subcategory $\mathsf{rpReprSp}$ of ReprSp whose objects are the subspaces of $2^{\mathbb{N}}$ (that is, objects of ReprSp that are isomorphic to spaces represented by a restriction of the identity map $\mathrm{id}_{2^{\mathbb{N}}} : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$). $\mathsf{rpReprSp}$ shares finite limits and colimits

[4] Technically, it rather is one representative of such a morphism. Let us also note that we would have also remarked already that LPO, regarded as a morphism, is a pullback of φ along KL (also regarded as a morphism), so (φ, id) is also a representative of the same morphism.

with ReprSp, however it is only weakly locally cartesian closed. Formally this means that reindexings f^* only have weak right adjoints [17, Remark 3.2]; so dependent function spaces exist, but functions may have multiple representatives (type-theoretically, this means that η laws need not hold).

▶ Remark 7. Certain exponentials do exist in rpReprSp: for instance, for every regular projective represented space A and subspace N of $\mathbb{N}$, A^N can be built. We also have an analogue local statement: call a map $f : X \to Y$ in rpReprSp *locally countable* if there exists some $s : X \to \mathbb{N}$ such that $\langle f, s\rangle : X \to Y \times \mathbb{N}$ is a subspace embedding. For such an f, f^* does have a right adjoint Π_f in rpReprSp.

We will still work with elcccs and ReprSp in the rest of the paper. This will mean that fixpoints involving $\star$ discussed in Section 4 will *not* match those used in Weihrauch reducibility, but can be thought of as extensional analogues which share many properties. That said, Remark 7 will mean that most results of Section 5 do discuss genuine Weihrauch degrees (see Remark 33). Investigating the case of extensive weak locally cartesian closed categories and Cont(rpReprSp) is left for future work; we hope most results carry through modulo the development of a theory of polynomial quasifunctors over weakly locally cartesian closed categories.

The second issue is that the notion of Weihrauch problem is usually restricted so that every question always has an answer. Containers in general have no such constraints, which makes their structural theory a bit nicer: for instance, the terminal object in Cont(ReprSp) is a problem with a question with no answers, and constructing exponentials in Cont(ReprSp) requires such questions with no answers; in contrast, the Weihrauch lattice has no $\top$.

▶ **Definition 8** ([49, Definition 7]). *A container $P : A \to I$ in $\mathsf{Cont}(\mathcal{C})$ will be called* answerable *if it is a pullback-stable epimorphism, that is, an epimorphism such that $f^*(P)$ is always an epimorphism for any $f : J \to I$.*

In represented spaces, this condition is equivalent to P being a set-theoretic surjection. So overall, an official Weihrauch problem will be an answerable container over rpReprSp. In the sequel, we shall often build objects of Cont(rpReprSp) and then build the subobject that restricts to answerable questions.

▶ **Definition 9.** *The* answerable part $\mathsf{Ans}(P)$ *of a container $P : A \to I$ in* Cont(ReprSp) *is the corestriction of P to its image.*

That is, if I is represented by the partial map $\delta_I : 2^{\mathbb{N}} \rightharpoonup I$, the codomain of $\mathsf{Ans}(P)$ is the represented space $(\{P(a) \mid a \in A\}, \delta')$ with δ' defined as the largest possible restriction of δ_I. $\mathsf{Ans}(P)$ is then tracked by the same realizer as P (categorically, this corresponds to taking the left part of a pullback-stable epi-regular mono factorization of P).

It is obvious that the answerable part of a container is answerable. Note that, contrary to the other operations on containers we discussed, it is not functorial, nor monotone when we look at the induced degrees: for instance, the answerable part of 1 is 0, while the answerable part of $\mathbf{I} + 1$ is $\mathbf{I}$, although we have $1 \equiv \mathbf{I} + 1$.

2.4 Fixpoints of polynomial functors

A convenient feature of polynomial functors is that they often admit fixpoints in categories of interest, which then interpret a variety of inductive and coinductive types.

▶ **Definition 10.** *A category $\mathcal{C}$ is said to have dependent $\mathcal{W}$-types (respectively, $\mathcal{M}$-types) if for every object I of $\mathcal{C}$ and container (P, t, u) of $\mathsf{Cont}(\mathcal{C})(I, I)$, $[\![P, t, u]\!] : \mathcal{C}/I \to \mathcal{C}/I$ has an initial algebra (respectively, a terminal coalgebra). A $\Pi\mathcal{W}\mathcal{M}$-category is an extensive locally cartesian closed category which also has dependent $\mathcal{W}$ and $\mathcal{M}$-types.*

We will primarily be concerned with Set and ReprSp, which are $\Pi\mathcal{WM}$-categories [6] and in which one should interpret all results in the reminder of this subsection (and Section 5). $\Pi\mathcal{WM}$ categories have natural and conatural objects, as the polynomial $1 + X$ is induced the container $\mathsf{in}_2 : 1 \to 2$. More generally, they interpret μ-expressions defined as follows:

$$E, E' \quad ::= \quad X \mid \mu X.\, E \mid \nu X.\, E \mid E \times E' \mid E + E'$$

Any such μ-expression E with k free variables thus induce a k-ary container that we write $(\!|E|\!)$; we shall also write $[\![E]\!]$ for $[\![(\!|E|\!)]\!]$. Intuitively, closed μ-expressions denote types of (possibly infinite) trees, which can systematically be characterized by tree automata [52, 30][5].

▶ **Example 11.** We have $(\!|\mu X.1{+}X|\!) \cong \left\{\chi_{\{u \in 2^* \mid u \sqsubseteq w\}} \mid w \in 0(10)^*1\right\}$ where χ_A is the indicator function $2^* \to 2$ of $A \subseteq 2^*$. For the greatest fixpoint, we have that $M_{\nu X.1+X}$ is the closure of $M_{\mu X.1+X}$ in $2^* \to 2$, i.e. $M_{\mu X.1+X} \cup \left\{\chi_{(01)^\omega}\right\}$.

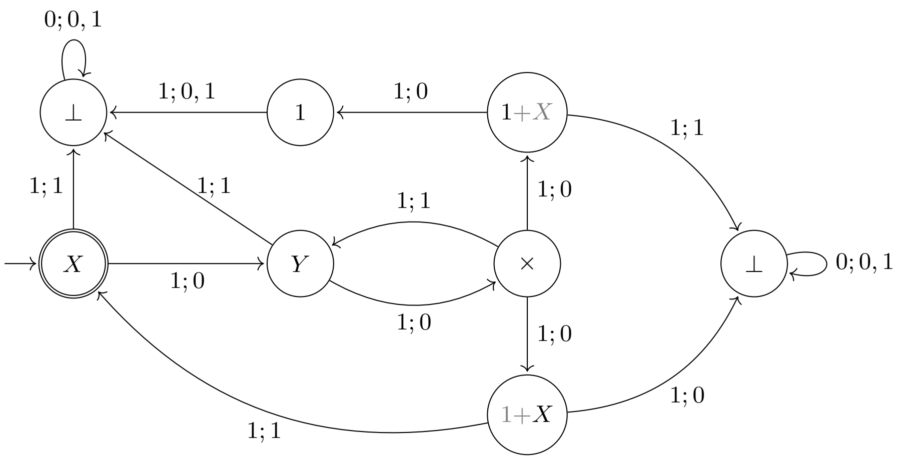
▶ **Example 12.** Let us consider $E = \mu X.\, \nu Y.\, (1 + X) \times Y$. We have that $(\!|E|\!)$ is isomorphic to the set of well-founded $\mathbb{N}$-ary trees $t \subseteq \mathbb{N}^*$. While this can be derived from first principles, a thoughtless automata-theoretic translation also tells us that

$$(\!|E|\!) \cong M_E := \{\ell_t \mid t \subseteq \mathbb{N}^* \text{ and } t \text{ is well-founded}\}$$

with $\ell_t : 2^* \to 2$ defined as the indicator function of

$$L_t = \bigcup_{w \in t} \{u \sqsubseteq h(w)(10)^n 00 \mid wn \notin t\} \cup \{u \sqsubseteq h(w)(10)^n 01 \mid wn \in t\}$$

with $h : t \to 2^*$ defined by $h(\varepsilon) = \varepsilon$ and $h(wn) = h(w)00(10)^n01$. M_E is recognized by the non-deterministic coBüchi infinite tree automaton [8, §14.3] below, where we draw an arrow $q \xrightarrow{a;i} r$ for a transition from state q to r reading a label $a \in 2$ and going down in direction $i \in 2$. The priority of all states are 0 except for X, which has priority 1,hence an accepting run only goes through the state X finitely many times along any branch, ensuring the encoded $\mathbb{N}$-ary tree is well-founded. All transitions are deterministic, except for those starting from state $\times$ along the direction 0: there, a guess of whether a node has a child or not is taking place.



[5] The characterization given in [52, 30] is as winning strategies in parity games over finite arenas. Those winning strategies can be regarded as trees, and the basic theory of automata over infinite trees tell us they form regular tree languages for any given finite arena. We favor using a direct description in terms of regular tree language in the text to minimize cognitive load later. as we will need to regard those strategies/trees as arenas for (other) parity games in Section 5.

3 Fixpoints of fibred polynomial functors

3.1 Fibred polynomial functors

We now turn to introducing fixpoints of endofunctors over categories of containers $\mathsf{Cont}(\mathcal{C})$. Here we focus on what we can achieve when we assume that $\mathcal{C}$ is a $\Pi\mathcal{W}\mathcal{M}$-category on exploiting the fibration $\mathrm{cod}^{\mathrm{op}} \simeq \mathsf{shape} : \mathsf{Cont}(\mathcal{C}) \to \mathcal{C}$ (and more generally its k-fold products $\mathsf{shape}^k : \mathsf{Cont}(\mathcal{C})^k \to \mathcal{C}^k$). This and a number of basic examples lead us to the following convenient definition.

▶ **Definition 13.** *Say that* $F : \mathsf{Cont}(\mathcal{C})^k \to \mathsf{Cont}(\mathcal{C})$ *is a* fibred polynomial functor *when*

1. (F, F_0) *is a fibred functor* $\mathsf{shape}^k \to \mathsf{shape}$ *for some uniquely determined* $F_0 : \mathcal{C}^k \to \mathcal{C}$ *(see [54, Definition 2.2])*
2. F_0 *is a polynomial functor*
3. *for every object* $I = (I_1, \ldots, I_k)$ *in* $\mathcal{C}^k$*, the functors on the fibers*

$$F_I : \mathcal{C}\big/ I_1 + \ldots + I_k \longrightarrow \mathcal{C}\big/ F_0(I)$$

induced by the isomorphism $\mathcal{C}/I_1 + \ldots + I_k \cong \mathcal{C}^k/I$ *and* F *are all polynomial functors.*

Say that $F : \mathsf{Cont}(\mathcal{C})^k \to \mathsf{Cont}(\mathcal{C})^m$ *is a* fibred polynomial functor *if every component* $\pi_i \circ F$ *is a fibred polynomial functor.*

Composition of fibred polynomial functors is defined in the obvious way and does result in fibred polynomial functors. Identities are fibred polynomial functors (as identity functors are polynomial functors [25, Example 1.6 (i)] and they trivially preserve cartesian morphisms). Fibred polynomial functors thus form a category, which also admits obvious finite cartesian products. Now we turn to showing which of the monoidal products from Figure 4 are also fibred polynomial functors.

▶ **Lemma 14.** *Constant functors and the bifunctors* $\times, +, \otimes$ *over* $\mathsf{Cont}(\mathcal{C})$ *are all fibred polynomial.*

▶ **Lemma 15.** *The following functor is fibred polynomial*

$$\begin{array}{ccc} \mathsf{Cont}(\mathcal{C}) & \longrightarrow & \mathsf{Cont}(\mathcal{C}) \\ X & \longmapsto & X \star P \end{array}$$

▶ Remark 16. On the other hand, $X \mapsto P \star X$ is *not* fibred.

With these lemmas, we have gathered all the tools we shall need to produce examples for of fibred polynomial functors in the rest of the paper. For instance, $X \mapsto \mathbf{I} + X \star A$ is seen to be fibred polynomial by combining the above lemmas.

▶ Remark 17. Unlike the usual polynomial endofunctors over a locally cartesian closed category, fibred polynomial functors do not necessarily have a canonical strength. A counter-example is the constant functor that maps any polynomial to $\mathbf{I}$.

3.2 They have fixpoints

We now turn to the main theorem, which states that we can, in a sense, lift fixpoints of functors in $\mathcal{C}$ to fixpoints of fibred polynomials endofunctors over $\mathsf{Cont}(\mathcal{C})$.

base \ total space	initial algebra μ	terminal coalgebra ν
initial algebra μ	initial algebra μ	initial algebra μ
terminal coalgebra ν	terminal coalgebra ν	(co)algebra ζ

Figure 5 Matrix of the kinds of fixpoints we get from applying the recipe outlined after Theorem 18 and Proposition 19.

▶ **Theorem 18.** *If $F : \mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$ is a fibred polynomial functor and $\mathcal{C}$ is an lextensive category with dependent $\mathcal{W}$ and $\mathcal{M}$-types, then*
- *F has an initial algebra $(\mu F, a_{\mu F})$*
- *F has a terminal coalgebra $(\nu F, c_{\nu F})$*
- *F has a (co)algebra $(\zeta F, b_{\zeta F})$ such that $(\mathsf{shape}(\zeta F), \mathsf{shape}(b_{\zeta F}))$ is a final coalgebra for F_0 and it induces a final coalgebra for the endofunctor induced by F over $\mathcal{C}/_{F_0(\zeta F)}$.*

The basic idea behind the proof of Theorem 18 is that we may compute a fixpoint for a fibred polynomial endofunctor F by first computing a fixpoint γF_0 for the induced polynomial functor F_0 in the base, and then take a fixpoint of $i^* \circ F_{\gamma F_0}$ where $i : \gamma F_0 \xrightarrow{\sim} F_0(\gamma F_0)$. In both steps, we have complete freedom over which fixpoint we want to compute, so we may use either initial or terminal coalgebras with the outcomes summarized in Figure 5. Note that there is a repeated entry in the matrix, due to the fact that taking the initial algebra in the base determines all fixpoints of $i^* \circ F_{\gamma F_0}$ (Proposition 19).

Proof. We only treat the first item, i.e. showing that F has an initial algebra. The other algebras are built in a similar fashion, and the terminal coalgebra is also shown to be terminal in a similar way.

The carrier. Since $\mathcal{C} \cong \mathcal{C}/_1$, $F_0 : \mathcal{C} \to \mathcal{C}$ has an initial algebra $(\mu, \mathtt{in} : F_0(\mu) \to \mu)$. By Lambek's lemma, $\mathtt{in}$ is an isomorphism. Similarly, the induced functor $F_\mu : \mathcal{C}/_\mu \to \mathcal{C}/_{F_0(\mu)} \cong \mathcal{C}/_\mu$ has a terminal coalgebra $(\nu, \mathtt{out} : \nu \to F_\mu(\nu))$ in $\mathcal{C}/_\mu$. Together, since $F_\nu = \mathtt{in}^*(F_\mu(\nu))$, these give a representative $(\mathtt{in}, \mathtt{out})$ for a container morphism $F(\nu) \to \nu$.

$$\begin{tikzcd}
F(\nu)_{\mathrm{dir}} \arrow[d, "F(\nu)"] & \nu \arrow[l, "\mathtt{out}"'] \arrow[r, equal] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \nu \arrow[d, "\nu"] \\
F_0(\mu) \arrow[r, equal] & F_0(\mu) \arrow[r, "\mathtt{in}"] & \mu
\end{tikzcd}$$

Weak initiality. Suppose we have an F-algebra with carrier $P : A \to I$ as follows:

$$\begin{tikzcd}
F(P)_{\mathrm{dir}} \arrow[d, "F(P)"] & B \arrow[l, "\beta"'] \arrow[r] \arrow[d, "b"] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "P"] \\
F_0(I) \arrow[r, equal] & F_0(I) \arrow[r, "\alpha"] & I
\end{tikzcd}$$

Let's now construct a container morphism $\nu \to P$. Since $(\mu, \mathfrak{in})$ is an initial F_0-algebra and α is an F_0-algebra, there exists a unique F_0-algebra homomorphism $\mathrm{fold}\,\alpha : \mu \to I$. Pulling back the commuting square for this homomorphism along P, we obtain the cube below where all vertical faces are pullback squares and $C \to D$ is an isomorphism.

$$\begin{tikzcd}
 & D \arrow[rr] \arrow[dd, "R" near start] & & A \arrow[dd, "P"] \\
C \arrow[ur, "\sim"] \arrow[rr, crossing over] \arrow[dd] & & B \arrow[ur] & \\
 & \mu \arrow[rr, "\mathrm{fold}\,\alpha" near start] & & I \\
F_0(\mu) \arrow[ur, "\mathfrak{in}"] \arrow[rr, "F_0(\mathrm{fold}\,\alpha)"] & & F_0(I) \arrow[ur, "\alpha"] \arrow[from=uu, crossing over, "Q" near end] &
\end{tikzcd}$$

The rearmost square is a pullback square, which can be viewed as a (representative of a) cartesian morphism between the containers $R : D \to \mu$ and $P : A \to I$. Since F is a fibred functor, applying it to that square yields another pullback square, and hence a unique arrow $\gamma : C \to F(R)_{\mathrm{dir}}$ making the diagram below commute.

$$\begin{tikzcd}
C \arrow[rrr] \arrow[dr, dashed, "\gamma"] \arrow[ddr, bend right] & & & B \arrow[d, "\beta"] \arrow[dd, bend left=60, "Q"] \\
 & F(R)_{\mathrm{dir}} \arrow[rr] \arrow[d, "F(R)"] \arrow[drr, phantom, "\lrcorner", very near start] & & F(P)_{\mathrm{dir}} \arrow[d, "F(P)"] \\
 & F_0(\mu) \arrow[rr, "F_0(\mathrm{fold}\,\alpha)"] & & F_0(I)
\end{tikzcd}$$

The arrow $D \cong C \xrightarrow{\gamma} F(R)_{\mathrm{dir}}$ in $\mathcal{C}$ yields a coalgebra $\mathfrak{d} : R \to F_\mu(R)$ in ${}^{\mathcal{C}}/_{\mu}$, and hence there is a unique coalgebra morphism $\mathrm{unfold}\,\gamma : R \to \nu$. It follows that we have a container morphism $(\mathrm{fold}\,\alpha, \mathrm{unfold}\,\gamma) : \nu \to P$.

Uniqueness. We now check that the $(\mathrm{fold}\,\alpha, \mathrm{unfold}\,\gamma)$ is the unique algebra morphism from $(\nu, (\mathfrak{in}, \mathfrak{out}))$ to $(P, (\alpha, \beta))$.

The first component of any such algebra morphism is an F_0-algebra morphism from $(\mu, \mathfrak{in})$ to (I, α). Since $(\mu, \mathfrak{in})$ is initial, that component therefore must be $\mathrm{fold}\,\alpha$.

Similarly, the second component must be an F_μ-coalgebra morphism from $(R, \mathfrak{d})$ to $(\nu, \mathfrak{out})$ in ${}^{C}/_{\mu}$. Since the morphisms $C \to F_\mu(\nu)$ are equal in both diagrams below, it follows that the second component must be equal to $\mathrm{unfold}\,\gamma$.

$$\begin{tikzcd}
 & F(R)_{\mathrm{dir}} \arrow[dl, bend right, "F(\mathrm{unfold}\,\gamma)"'] & & C \arrow[ll, "\gamma"] \arrow[d, "\wr"] & \\
F_\mu(\nu) \arrow[d, "F(\nu)"] & \nu \arrow[l, "\sim"'] \arrow[r, equal] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \nu \arrow[d] & D \arrow[l, "\mathrm{unfold}\,\gamma"'] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "P"] \\
F_0(\mu) \arrow[r, equal] & F_0(\mu) \arrow[r, "\sim"] & \mu \arrow[r, equal] & \mu \arrow[r, "\mathrm{fold}\,\alpha"] & I
\end{tikzcd}$$

$$
\begin{tikzcd}
& C \arrow[d, dashed, "\exists!"] \arrow[dd, bend right=60] \arrow[drr, dashed, "\exists!"] \arrow[drrr, bend left] & & & \\
F_\mu(\nu) \arrow[d, "F(\nu)"] & F(R)_{\mathrm{dir}} \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & F(P)_{\mathrm{dir}} \arrow[d, "F(P)"] & B \arrow[l] \arrow[r] \arrow[d, "Q"] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "P"] \\
F_0(\mu) \arrow[r, equal] & F_0(\mu) \arrow[r, "F_0(\mathrm{fold}\,\alpha)"'] & F_0(I) \arrow[r, equal] & F_0(I) \arrow[r, "\alpha"'] & I
\end{tikzcd}
$$

◀

▶ **Proposition 19.** *If (F_0, F) is a fibred polynomial functor over a $\Pi\mathcal{WM}$-category $\mathcal{C}$, $\alpha : F_0(\mu F_0) \to \mu F_0$ is an initial algebra for F_0 and we have $S_0, S_1 \in {}^{\mathcal{C}}/_{\mu F_0}$ with $F_{\mu F_0}(S_i) \cong S_i \circ \alpha$ for $i \in \{0, 1\}$, then $S_0 \cong S_1$.*

▶ **Example 20.** $\mu(X \mapsto X)$ is an initial object, $\nu(X \mapsto X)$ is a terminal object and $\zeta(X \mapsto X)$ is the monoidal unit $\mathbf{I}$ of $\otimes$.

▶ **Example 21.** $X \mapsto 1 + X$ admits the following fixpoints:

- $\mu(X \mapsto 1 + X)$ is the container $0 \to \mathbb{N}$
- $\nu(X \mapsto 1 + X)$ is the container $0 \to \mathbb{N}_\infty$
- $\zeta(X \mapsto 1 + X)$ is the container $\infty : 1 \to \mathbb{N}_\infty$.

Finally, Theorem 18 also behaves well in presence of parameters.

▶ **Theorem 22.** *Let $F : \mathsf{Cont}(\mathcal{C})^{k+1} \to \mathsf{Cont}(\mathcal{C})$ be a fibred polynomial functor. For any $\gamma \in \{\mu, \nu, \zeta\}$, the map*

$$
\begin{array}{ccc}
\mathsf{Obj}\left(\mathsf{Cont}(\mathcal{C})^k\right) & \longrightarrow & \mathsf{Obj}\left(\mathsf{Cont}(\mathcal{C})\right) \\
(f_0, \ldots, f_{k-1}) & \longmapsto & \gamma F(f_0, \ldots, f_{k-1}, -)
\end{array}
$$

extends to a fibred polynomial functor $\gamma_k F : \mathsf{Cont}(\mathcal{C})^k \to \mathsf{Cont}(\mathcal{C})$.

4 Operators definable as fixpoints

Assuming a $\Pi\mathcal{WM}$-category $\mathcal{C}$, we discuss a number of endofunctors over $\mathsf{Cont}(\mathcal{C})$ defined in terms of fixpoints as per Theorem 18. Those useful for the sequel are summarized in Figure 1.

4.1 Parallelizations

Fix a container $P : A \to I$ (write $(A_i)_{i:I}$ for the corresponding family) and let us consider fixpoints of the endofunctor $X \mapsto P \times X$ over $\mathsf{Cont}(\mathcal{C})$. We can first note that we have the following isomorphisms in $\mathcal{C}$.

$$
\begin{array}{ll}
\mathsf{shape}(\mu(X \mapsto P \times X)) \cong \mu(X \mapsto I \times X) \cong 0 & \\
\mathsf{shape}(\gamma(X \mapsto P \times X)) \cong \nu(X \mapsto I \times X) \cong I^{\mathbb{N}} & \text{for } \gamma \in \{\nu, \zeta\}
\end{array}
$$

As a result, we have $\mu(X \mapsto P \times X) \cong 0$. The situation with the other fixpoints is slightly more exciting. The greatest fixpoint yields the family $\left(\sum_{n:\mathbb{N}} A_{s(n)}\right)_{s:I^{\mathbb{N}}}$. Interpreted as a transformation of problems, the corresponding operator takes a problem P to the problem $\widetilde{P}$ where a question is a sequence of P-questions, and an answer picks an index n and answers the nth question.

$$
\begin{array}{cccc}
\widetilde{-} : \mathsf{Cont}(\mathcal{C}) & \longrightarrow & \mathsf{Cont}(\mathcal{C}) \\
P & \longmapsto & \nu(X \mapsto P \times X)
\end{array}
$$

The ζ fixpoint yields the family $\left(\sum_{x:\mathbb{N}_\infty} \prod_{n:\mathbb{N}} \left(x = \underline{n} \to A_{s(n)}\right)\right)_{s:I^{\mathbb{N}}}$. In terms of problems, an answer now consists of a conatural number and, if this conatural number corresponds to a natural number, then an answer is eventually given for the relevant question.

In terms of degrees, $\widetilde{P}$ is not very interesting because $P \equiv \widetilde{P}$: we can reduce P to $\widetilde{P}$ by asking the same question ω times, and conversely, we can reduce $\widetilde{P}$ to P by solving only the first P-question in the input to $\widetilde{P}$. It is thus typically more interesting to consider another operator where one gets a stream of answers for all questions asked; such an operator is called the (infinite) parallelization operator [13, Definition 1.2]. For this we naturally consider a fixpoint of the functor $X \mapsto P \otimes X$ induced by the binary parallelization $\otimes$. Since $X \mapsto P \otimes X$ and $X \mapsto P \times X$ induce the same endofunctor $Y \mapsto I \times Y$ over $\mathcal{C}$, the shapes of the fixpoints are the same and $\mu(X \mapsto P \otimes X) \cong 0$. This time around, the greatest fixpoint is of little interest, but the middle fixpoint ζ gets us what we want; we write $\widehat{P}$ for the infinite parallelization of P defined thus.

$$\begin{array}{lcl} \widehat{-} : \mathsf{Cont}(\mathcal{C}) & \longrightarrow & \mathsf{Cont}(\mathcal{C}) \\ P & \longmapsto & \zeta(X \mapsto P \otimes X) \end{array}$$

It is also customary to consider a finitary version of parallelization by taking the least fixpoint of $X \mapsto \mathbf{I} + P \otimes X$. The finite parallelization of P is typically denoted P^*.

4.2 Iterations

Now, let us turn to fixpoints built with the sequential composition of problems $\star$. As $\star$ is not commutative there are, a priori, two ways of defining functors whose fixpoints are iterations of some problem P. However, one of these is not fibred (Remark 16) and the other is (Lemma 15). We will thus mostly focus on the latter, although it is worth noting that $X \mapsto \mathbf{I} + P \star X$ does have a non-trivial initial algebra with a natural interpretation. It corresponds to the problem whose instances (n, f) are given by a number n of successive questions we want to ask to P and then a recipe f to ask those questions, and where results are the answers to n questions we could have successively asked.

▶ **Proposition 23.** *The carrier of the initial algebra for $X \mapsto \mathbf{I} + X \star P$ is isomorphic to $\sum_{n:\mathbb{N}} \underbrace{P \star \ldots \star P}_{n \text{ times}}$.*

On the other hand, the functor $X \mapsto \mathbf{I} + X \star P$ is fibred polynomial by Lemmas 14 and 15. So Theorem 22 says we have a corresponding polynomial endofunctor $(-)^\diamond : \mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$. Regarding $P^\diamond$ as a problem, inputs are recipes to ask finitely many questions sequentially to P and outputs are possible sequences of answers to that. Note that in general, this is strictly more powerful than the operator from Proposition 23 as the number of questions need not be known in advance.

While finite iterations of problems have been introduced in 2016 to the literature on Weihrauch complexity [44], the endofunctor $(-)^\diamond$ has been known for much longer to category theorists: $[\![P^\diamond]\!]$ was characterized as the free monad over the polynomial functor $[\![P]\!]$ [33]. The construction given in [25, Section 4] correspond to what is given by Theorem 18 using the μ/μ recipe. From that point of view, assuming for a moment $P \in \mathsf{Cont}(\mathsf{Set})$, the endofunctor $[\![P^\diamond]\!]$ may be described as follows. The shape of $P^\diamond$ consists of well-founded trees labelled by some $i : I$ or by a special symbol $\bullet$, and the arity of the node is $P^{-1}(i)$ or 0 if the special symbol is used. Then $[\![P^\diamond]\!](X)$ consists of all those trees from $P^\diamond$ where every leaf ℓ is labelled by some $x_\ell \in X$.

▶ Remark 24. While $\mathsf{shape}(P^\diamond)$ can, a fortiori, be regarded as a sort of well-founded tree structure in $\mathsf{Cont}(\mathsf{ReprSp})$, it is important to note that those trees are very different from the ones we encode in $\mathbb{N}^\mathbb{N}$ for the rest of this paper: any node has potentially continuum-many children and knowing "the arity" of a node corresponds to the ability to answer the P-question of its label, which is typically impossible to do continuously.

A key property of $(-)^\diamond$ is that it associates well with composition of other problems, which is captured by the following.

▶ **Theorem 25.** *$Q \star P^\diamond$ is the carrier of an initial algebra of the fibred polynomial endofunctor $X \mapsto Q + X \star P$.*

One useful application of Theorem 25 is a straightforward construction of a monad multiplication $[\![P^\diamond]\!] \circ [\![P^\diamond]\!] \cong [\![P^\diamond \star P^\diamond]\!] \to [\![P^\diamond]\!]$ by exhibiting an algebra $X \mapsto P^\diamond + X \star P$ whose carrier is $P^\diamond$. We leave that as an exercise, together with the consequences that $P^\diamond$ is the free polynomial monad over P and that $(-)^\diamond$ is a monad. Theorem 25 can also gives an induction principle on degrees:

$$Q \star P \;\leq\; Q \qquad \Longrightarrow \qquad Q \star P^\diamond \;\leq\; Q$$

This scheme corresponds to an induction axiom of the theory of right-handed Kleene algebras [36]. It is also a key part of a complete axiomatization of the Weihrauch lattice augmented with iterated composition [50], or rather, a slightly generalized version of it which may be derived using the cartesian closed structure of $\mathsf{Cont}(\mathcal{C})$.

As finite parallelization has a infinitary countable counterpart, so does sequential iteration by considering a suitable fixpoint of the endofunctor $X \longmapsto X \star P$. Much like in the case of parallelization, the more interesting operator will be derived using the middle fixpoint construction ζ. Then $P^\infty := \zeta X.\ X \star P$ corresponds to the problem whose instances are recipes to ask countably many questions to P in a sequential manner and whose solutions are ω-sequences of possible answers. An analogous operator has recently been introduced in Weihrauch reducibility [11]. In the case of Set, the corresponding endofunctor $[\![P^\infty]\!]$ maps X to the set of pairs (t, x) consisting of a tree t whose nodes are all labelled by P-questions, arities consist of the P-answers, and x maps infinite paths p through t to elements $x(p) \in X$.

When it comes to the less interesting fixpoints, similarly to what happened with $X \mapsto P \times X$, we have $\mu(X \mapsto X \star P) \cong 0$ and $\nu(X \mapsto X \star P) \cong 1 \star P^\infty$. It is clear that since $(1 \star P)^\infty \cong 1 \star P$, those two constructions yield idempotent operators on P. However, $(-)^\infty$ is not idempotent. From the perspective of Weihrauch complexity, we have that $P^{\infty\infty}$ is the problem where one can ask ω^2 questions to P sequentially and then get all of the answers, which is more powerful than asking ω questions sequentially (for instance, $\mathsf{LPO}^\infty < \mathsf{LPO}^{\infty\infty}$ [11, Corollary 23]).

5 Ground problems

5.1 ζ-expressions

As announced before, we may define the following syntax for fibred polynomial functors over $\mathsf{Cont}(\mathcal{C})$ when $\mathcal{C}$ is a $\Pi\mathcal{W}\mathcal{M}$-category.

$$E, E' \;::=\; X \mid \mu X.\ E \mid \zeta X.\ E \mid \nu X.\ E \mid E \times E' \mid E \otimes E' \mid E + E'$$

We write $\zeta\mathrm{Expr}(\vec{X})$ for the set of such ζ-*expressions* with free variables in $\vec{X}$. ζ-expressions E with k free variables have an obvious interpretation as fibred polynomial functors $\langle\!\langle E \rangle\!\rangle : \mathsf{Cont}(\mathcal{C})^k \to \mathsf{Cont}(\mathcal{C})$. When $k = 0$, we regard $\langle\!\langle E \rangle\!\rangle$ as an object of $\mathsf{Cont}(\mathcal{C})$. We will also use the following obvious shorthands for ζ-expressions (where X is a fresh variable):

$$\begin{array}{lll} 0 \ := \mu X.X & \mathbf{I} \ := \zeta X.X & 1 \ := \nu X.X \\ E^* := \mu X.\mathbf{I} + E \otimes X & \widehat{E} := \zeta X.E \otimes X & \widetilde{E} := \nu X.E \times X \end{array}$$

Before turning to examples of ζ-expressions we wish to interpret in $\mathsf{Cont}(\mathsf{ReprSp})$, we will now give an automata-theoretic characterization of their denotations, extending a characterization of μ-expressions alluded to in Section 2. For this, we assume some familiarity with the theory of regular languages of infinite trees [8, §14.3] and write $\mathsf{Tree}(\Gamma, \chi)$ for the set of trees whose internal nodes are binary and labelled by elements of Γ and whose leaves are labelled by elements of χ (if $\chi = \emptyset$, we may write $\mathsf{Tree}(\Gamma)$).

▶ **Definition 26.** *The* gaming alphabet $\mathcal{G}_{i,k}$ *with ranks* (i, k) *(for* $i \le k \in \mathbb{N}$*) is the finite set* $\{\bot\} \uplus \{\mathsf{Even}, \mathsf{Odd}\} \times \{i, \ldots, k\}$*. A* binary game tree *with ranks* (i, k) *and exits* χ *is an element* $\ell \in \mathsf{Tree}(\mathcal{G}_{i,k}, \chi)$ *such that* $\ell(w) = \bot$ *implies* $\ell(wi) = \bot$ *for any* $w \in 2^*$*,* $i \in 2$*. An* Even-strategy *of a binary game tree* ℓ *is a subset* $S \subseteq \mathrm{dom}(\ell)$ *such that* $\varepsilon \in S$*,* $S \cap \ell^{-1}(\bot) = \emptyset$ *and, for every* $w \in S$*:*

- *if* $\pi_1(\ell(w)) = \mathsf{Even}$*, then there exists exactly one* $i \in 2$ *with* $wi \in S$
- *otherwise if* $\pi_1(\ell(w)) = \mathsf{Odd}$*, for every* $i \in 2$ *such that* $\ell(wi) \neq \bot$*,* $wi \in S$*.*

A strategy S *is* winning *for* Even *if for any infinite path* $p \in 2^{\mathbb{N}}$ *through* S*,* $\limsup\limits_{n\to+\infty} \pi_2(\ell(p[n]))$ *is even. Let us call* $\mathsf{Strat}(\ell)$ *the set of such winning strategies.*

For any finite set χ, any game tree language $L \subseteq \mathsf{Tree}(\mathcal{G}_{i,k}, \chi)$ induces a fibred polynomial functor $\langle\!\langle L \rangle\!\rangle : \mathsf{Cont}(\mathsf{ReprSp})^\chi \to \mathsf{Cont}(\mathsf{ReprSp})$. To describe its action on tuples of containers, first recall that L also determines a functor $[\![L]\!] : \mathsf{ReprSp}^\chi \to \mathsf{ReprSp}$ mapping a tuple of spaces $(V_x)_{x\in\chi}$ to the space of pairs $\ell \in L$ and a family of maps $\rho_x : \ell^{-1}(x) \to V_x$ that labels the paths of ℓ ending in $x \in \chi$ with some element of V_x. We compute the shape of $\langle\!\langle L \rangle\!\rangle$ by $\mathsf{shape}(\langle\!\langle L \rangle\!\rangle) = [\![L]\!]$. Intuitively, the answers corresponding to a pair (ℓ, ρ) will then consist of a winning strategy for Even in ℓ together with answers for the questions occurring on maximal finite paths of the strategy. All in all, writing $S_x := \ell^{-1}(x) \cap S$ for $S \in \mathsf{Strat}(\ell)$ and $x \in \chi$, we have

$$\begin{array}{rcl} \langle\!\langle L \rangle\!\rangle : \mathsf{Cont}(\mathsf{ReprSp})^\chi & \longrightarrow & \mathsf{Cont}(\mathsf{ReprSp}) \\ ((A_i)_{i\in I_x})_{x\in\chi} & \longmapsto & \left(\sum\limits_{S\in\mathsf{Strat}(\ell)} \prod\limits_{x\in\chi} \prod\limits_{p\in S_x} A_{\rho(p)} \right)_{(\ell,\rho)\in[\![L]\!]} \end{array}$$

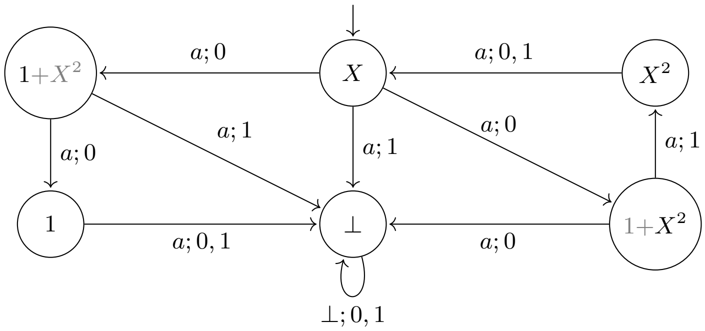
▶ **Theorem 27.** *From a* ζ*-expression* E *with free variables* χ*, we can compute a regular game tree language* L_E *such that* $\langle\!\langle E \rangle\!\rangle$ *and* $\langle\!\langle L_E \rangle\!\rangle$ *are isomorphic in* $\mathsf{Cont}(\mathsf{ReprSp})^\chi \to \mathsf{Cont}(\mathsf{ReprSp})$*.*

Let us now unroll the construction behind Theorem 27 on a number of examples.

▶ **Example 28.** We have $L_{\mu X.X} = \emptyset$ and $L_{\gamma X.X} = \{\ell_\gamma\} \subseteq \mathsf{Tree}(\mathcal{G}_{0,1})$ for $\gamma \in \{\nu, \zeta\}$ where we have $\ell_\gamma(p) = \bot$ if p contains a 1, $\ell_\nu(0^n) = (\mathsf{Even}, 1)$ and $\ell_\zeta(0^n) = (\mathsf{Even}, 0)$.

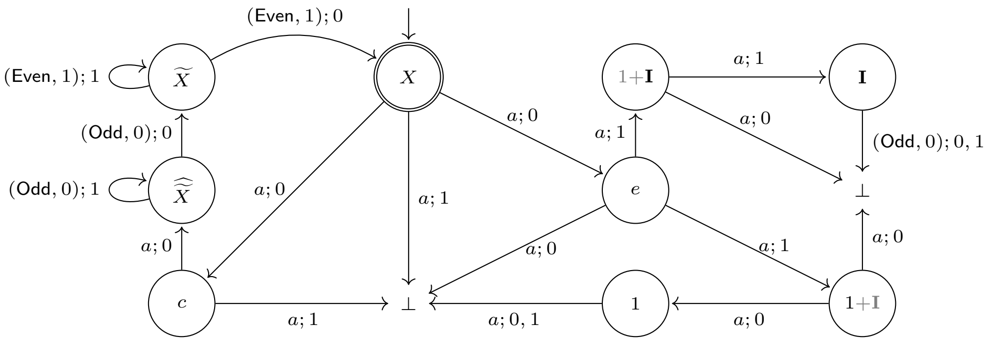
▶ **Example 29.** The language $L_{\zeta X.\nu Y.Y+X}$ includes only trees $\ell \in \mathsf{Tree}(\mathcal{G}_{0,1})$ where ℓ is constantly $\bot$ except along a unique infinite path p belonging to the regular language of infinite words $00((1+\varepsilon)00)^\omega$. The labelling is fully determined by the path in question: $\ell(w1) = (\mathsf{Even}, 0)$ and $\ell(w0) = (\mathsf{Even}, 1)$ for any $w \sqsubset p$. Accordingly, $\mathsf{Strat}(\ell)$ is $\{p\}$ if $\exists^\infty n \in \mathbb{N}.\ p_n = 1$ and $\emptyset$ otherwise.

▶ **Example 30.** The tree language $L_{\zeta X.\ 1+X\times X}$ is recognized by the following Büchi tree automaton assuming $a = (\mathsf{Even}, 0)$ and all states are accepting:



For any $\ell \in L_{\zeta X.\, 1+X\times X}$, we have that $\ell^{-1}(a)$ is a subtree of the full binary tree 2^*. By keeping every third letter of those $w \in \ell^{-1}(a)$, we would obtain a subtree ℓ' of 2^* where every node is either a leaf or has two children; it is not difficult to see that all such subtrees can be obtained in this way. Winning Even strategies in the game tree ℓ are in bijection with infinite paths of ℓ'.

▶ **Example 31.** Let us now consider $D_1 = \mu X.\ \widetilde{\widetilde{X}} + 1 + \mathbf{I}$. The tree language $L_{D_1} \subseteq \mathsf{Tree}(\mathcal{G}_{0,1})$ is recognized by the coBüchi automaton below, where $a = (\mathsf{Even}, 0) \in \mathcal{G}_{0,1}$, X is the only rejecting state and $\bot$ denotes a subautomaton recognizing only the constant labelling by $\bot$ of the full binary tree. The states are labelled with the corresponding subexpressions of D_1, except for e and c which should be respectively read as "end" and "continue".



More intuitively, one can see that on shapes, $\langle\!\langle D_1 \rangle\!\rangle$ is the carrier of an initial algebra $X \mapsto 2 + X^{\mathbb{N}^2}$. It thus consists of well-founded trees where all nodes are either leaves labelled by a boolean or are internal nodes of arity $\mathbb{N}^2$. A direction corresponding to such a tree is a winning strategy for the second player in the following game whose positions are paths in the tree:

- if the path is a leaf, then the first player wins if and only if its label is 0,
- otherwise, the first player first picks a number $n \in \mathbb{N}$, the second player picks a number $m \in \mathbb{N}$, and the game proceeds from the (n, m)th child of the current position.

Questions to $\langle\!\langle D_1 \rangle\!\rangle$ seen as a problem are essentially clopen games over $\mathbb{N}^{\mathbb{N}}$ and answers are winning strategies for the second player.

5.2 Some Weihrauch degrees as answerable parts of ζ-expressions

Let us now sketch how the fragment of the Weihrauch lattice described in Figure 2 can be captured closed ζ-expressions, or more accurately, the *answerable part* (Definition 9) of their denotations. The latter step is important as all ζ-expressions are trivial from a degree-theoretic point of view.

▶ **Proposition 32.** *For any closed ζ-expression E, $\langle\!\langle E\rangle\!\rangle \equiv 0$, $\mathbf{I}$ or 1.*

Proof idea. We generalize the statement inductively to open expressions to state that the set of degrees $\{0, \mathbf{I}, 1\}$ are preserved by any $\langle\!\langle E\rangle\!\rangle$. The more interesting case is the fixpoint case. In that case, we can show that for any fibred polynomial $F : \mathsf{Cont}(\mathcal{C}) \to \mathsf{Cont}(\mathcal{C})$ that preserve the set of degrees $\{0, \mathbf{I}, 1\}$, we have $\mu F \equiv F(F(0))$, $\zeta F \equiv F(\mathbf{I})$ and $\nu F \equiv F(F(1))$. ◀

For instance, the expression D_1 from Example 31 gives $\langle\!\langle D_1\rangle\!\rangle \equiv 1$ because one can compute a clopen game where Even loses. On the other hand, $\mathsf{Ans}(\langle\!\langle D_1\rangle\!\rangle)$ captures $\mathbf{\Delta}^0_1$-determinacy, a non-trivial Weihrauch problem equivalent to $\mathsf{UC}_{\mathbb{N}^\mathbb{N}}$.

▶ Remark 33. By Remark 7, we can inspect the definition of $\langle\!\langle L\rangle\!\rangle$ to check it always restricts to a functor $\mathsf{Cont}(\mathsf{rpReprSp})^{\mathcal{X}} \to \mathsf{Cont}(\mathsf{rpReprSp})$. Hence, by Theorem 27, the answerable parts of denotations of ζ-expressions correspond to genuine Weihrauch degrees (in contrast to the composition operator $\star$ and its iterations in the previous section).

Let us now treat a selection of examples from Figure 2, starting with the first example of a degree we introduced in the paper.

▶ **Proposition 34.** LPO *is Weihrauch-equivalent to* $\mathsf{Ans}\left(\langle\!\langle(\zeta X.\ X+1) \times (\nu X.\ X+\mathbf{I})\rangle\!\rangle\right)$.

Proof. Computing the isomorphism types of $\langle\!\langle \zeta X.\ X+1\rangle\!\rangle$ and $\langle\!\langle \nu X.\ X+\mathbf{I}\rangle\!\rangle$, we obtain the containers corresponding respectively to $\infty : 1 \to \mathbb{N}_\infty$ and to the embedding of $\mathbb{N}$ into $\mathbb{N}_\infty$. We thus have two problems that take as input some $x \in \mathbb{N}_\infty$ which can have a trivial answer. That answer exists only if $x = \infty$ in the first case, and conversely only if $x = \underline{n}$ for some $n \in \mathbb{N}$ in the second case. Hence, questions to $\langle\!\langle(\zeta X.\ X+1) \times (\nu X.\ X+\mathbf{I})\rangle\!\rangle$ are pairs $(x, y) \in \mathbb{N}_\infty^2$ and legal answers are booleans $b \in 2$ such that

$$b = 0 \Longrightarrow x = \infty \qquad \text{and} \qquad b = 1 \Longrightarrow y < \infty$$

Taking the answerable part ensures that we only have inputs such that $y = \infty \Rightarrow x = \infty$. LPO on the other hand is easily seen to be equivalent to the problem taking some $x \in \mathbb{N}_\infty$ and outputting the boolean b with $b = 0 \Leftrightarrow x = \infty$. So we can take the forward part of a reduction $\mathsf{LPO} \leq \langle\!\langle(\zeta X.\ X+1) \times (\nu X.\ X+\mathbf{I})\rangle\!\rangle$ to be the diagonal map $x \mapsto (x, x)$ and its backwards part to be trivial. For the converse reduction, the forward part is simply the second projection, and the backward part is trivial. ◀

The next degree we consider captures *closed choice problems* over (subspaces of) $\mathbb{N}$ with the cofinite topology.

▶ **Definition 35.** *An enumeration $A \subseteq \mathbb{N}$ is a map $e : \mathbb{N} \to \{\bot\} \uplus \mathbb{N}$ such that $A = e(\mathbb{N}) \cap \mathbb{N}$. For $A \subseteq \mathbb{N}$, define C_A to be the following problem: questions are enumerations e of strict subsets of A, and answers are those $m \in A$ which are not enumerated by A.*

▶ **Proposition 36.** $\mathsf{C}_\mathbb{N} \equiv \mathsf{Ans}\left(\langle\!\langle \zeta \widetilde{X.X+1}\rangle\!\rangle\right)$ *and, for $k \in \mathbb{N}$,*

$$\mathsf{C}_k \quad \equiv \quad \mathsf{Ans}\left(\langle\!\langle\underbrace{(\zeta X.\ X+1) \times \ldots \times (\zeta X.\ X+1)}_{k \text{ times}}\rangle\!\rangle\right)$$

Proof. Recall that $\langle\!\langle \zeta X.X+1\rangle\!\rangle$ is isomorphic to the container $\infty : 1 \to \mathbb{N}_\infty$ (Example 21). As a problem, a question is a conatural number, and a question can be (trivially) answered if and only if that conatural number is ∞. This means that $\langle\!\langle \zeta \widetilde{X.X+1}\rangle\!\rangle$ is the problem whose questions are sequences of conatural numbers $x \in \mathbb{N}_\infty^{\mathbb{N}}$ which are answered by any $n \in \mathbb{N}$ such that $x_n = \infty$; a fortiori, $\mathsf{Ans}\langle\!\langle \widetilde{\zeta X. X + 1} \rangle\!\rangle$ restricts the questions so that an $n \in \mathbb{N}$ with $x_n = \infty$ does exist.

The forward part of a reduction $\mathsf{C}_{\mathbb{N}} \leq \mathsf{Ans}\langle\!\langle \widetilde{\zeta X. X + 1} \rangle\!\rangle$ simply sends an enumeration e to the sequence $(\sup\{x \in \mathbb{N}_\infty \mid \forall k \in \mathbb{N}.\ \underline{k} < x \Rightarrow e_k \neq n\})_{n \in \mathbb{N}}$, and the corresponding backwards part is trivial. Conversely, from a sequence $x \in \mathbb{N}_\infty^{\mathbb{N}}$, we can compute

$$e(\langle n, m \rangle) = \begin{cases} n & \text{if } x_n \leq \underline{m} \\ \infty & \text{otherwise} \end{cases}$$

as part of a forward reduction (assuming $\langle -, - \rangle : \mathbb{N}^2 \to \mathbb{N}$ is the standard Cantor pairing function) and have a trivial backwards reduction. The proof for C_k is essentially the same. ◀

Another important instance of closed choice is $\mathsf{C}_{2^{\mathbb{N}}}$, which is equivalent to weak König's lemma.

▶ **Definition 37.** WKL *is the problem whose questions are infinite binary trees $t \subseteq 2^*$, and where such a question is answered by an infinite path $p \in 2^{\mathbb{N}}$ through t.*

▶ **Proposition 38.** WKL *is Weihrauch equivalent to the answerable parts of $\langle\!\langle \zeta X.\ 1 + X \times X \rangle\!\rangle$ and of the infinite parallelization of $\langle\!\langle \zeta X.\ X + 1 \rangle\!\rangle \times \langle\!\langle \zeta X.\ X + 1 \rangle\!\rangle$.*

Proof. The first part was explained in Example 30; the only minor difference with the official definition of WKL that makes one of the reductions different from the identity is that input trees to $\zeta X. 1 + X \times X$ should not have internal unary nodes, which is remedied by adding leaves in the forward reduction. As for the second part, it follows from $\widehat{\mathsf{C}_2} \equiv \mathsf{WKL}$ [12, Theorem 8.2], $\mathsf{Ans}(\widehat{P}) \cong \widehat{\mathsf{Ans}(P)}$ and Proposition 36. ◀

5.3 Problems on game tree languages

We have seen that Ans dramatically increases the expressiveness of closed ζ-expressions $E \in \zeta\mathrm{Expr}$. We may ask whether we can see any limit to this approach, and whether there are any nice structural properties of the set of problems expressible in this way. The following basic observation hints that considering containers $\langle\!\langle L \rangle\!\rangle$ associated to regular game tree languages might be worthwhile.

▶ **Proposition 39.** *For any closed ζ-expression E, we can compute a regular game tree language $L_E \cap W$ such that $\langle\!\langle L_E \cap W \rangle\!\rangle \cong \mathsf{Ans}(\langle\!\langle E \rangle\!\rangle)$.*

Proof. Assuming L_E (from Theorem 27) is over the alphabet $\mathcal{G}_{0,k}$, simply note that the languages $W_{0,k} \subseteq \mathsf{Tree}(\mathcal{G}_{0,k})$ of game trees where Even has a winning strategy is easily seen to be regular, and regular tree languages are closed under finite intersections. ◀

Standard automata theoretic techniques lead to an effective trichotomy that puts a (very high) bound on what we may capture by ζ-expressions.

▶ **Proposition 40.** *For any regular game tree language $L \subseteq \mathsf{Tree}(\mathcal{G}_{0,k})$, we are in one of the following three cases:*

$$\langle\!\langle L \rangle\!\rangle \equiv 0 \qquad \mathbf{I} \leq \langle\!\langle L \rangle\!\rangle \leq (\mathbf{\Sigma}^0_2)_k\text{-}\mathsf{FindWS} \quad \textit{or} \quad \langle\!\langle L \rangle\!\rangle \equiv 1$$

Furthermore, one can compute in which case we are and the appropriate reductions from an automaton recognizing L.

Proof. Given any automaton for $L \subseteq \mathsf{Tree}(\mathcal{G}_{0,k})$, first, decide whether L is empty (in which case $\langle\!\langle L \rangle\!\rangle \cong 0$), or compute a regular tree $\ell \in L$, which yields a reduction $\mathbf{I} \leq \langle\!\langle L \rangle\!\rangle$. Next, decide if $L \setminus W$ is inhabited. If it is, compute a regular tree $\ell' \in L \setminus W$, which yields a reduction $1 \leq \langle\!\langle L \rangle\!\rangle$. Otherwise, $L \cap W = L$, then $\langle\!\langle L \rangle\!\rangle$ is answerable and we have a trivial reduction $\langle\!\langle L \rangle\!\rangle \leq (\mathbf{\Sigma}^0_2)_k\text{-}\mathsf{FindWS}_2$. ◀

This means in particular that $\mathsf{Ans}(\langle\!\langle L \rangle\!\rangle)$ cannot be a problem strictly higher than $(\mathbf{\Sigma}^0_2)_k\text{-}\mathsf{FindWS}_2$s in the Weihrauch lattice. The reductions $\langle\!\langle L \rangle\!\rangle \leq (\mathbf{\Sigma}^0_2)_k\text{-}\mathsf{FindWS}$ are quite trivial and do not guarantee that k is picked optimally. So we feel motivated to ask the following question inspired by previous work on algebraic syntaxes for operators on Weihrauch problems [45, 50].

▶ **Question 41.** *Given automata recognizing $L, M \subseteq \mathsf{Tree}(\mathcal{G}_{0,k}, \chi)$, can we computably decide whether the following hold or not?*

$$\forall P \in \mathsf{Cont}(\mathsf{ReprSp})^{\chi}. \qquad \langle\!\langle L \rangle\!\rangle(P) \leq \langle\!\langle M \rangle\!\rangle(P)$$

What about the cases where L and M arise from ζ-expressions (as per Theorem 27)?

A positive solution to the question above with $\chi = \emptyset$ would in particular allow us to compute, from any automaton for some $L \subseteq \mathsf{Tree}(\mathcal{G}_{0,k})$, the optimal $m \leq k$ such that $\langle\!\langle L \rangle\!\rangle \leq (\mathbf{\Sigma}^0_2)_m\text{-}\mathsf{FindWS}$ when it exists. This is reminiscent of the Rabin-Mostowski index problem asking whether optimizing for indices for infinite tree automata is doable computably. This is a long-standing open problem, although solved positively for interesting subclasses of all tree automata, including deterministic ones [22, 46, 31]. However, Question 41 and its variants also involve thinking about reductions rather than pure language recognition, so a solution might also involve taming some non-trivial continuous infinite tree transductions.

References

1 Michael Abbott, Thorsten Altenkirch, and Neil Ghani. Containers: Constructing strictly positive types. *Theoretical Computer Science*, 342(1):3–27, 2005. doi:10.1016/J.TCS.2005.06.002.

2 Michael Gordon Abbott, Thorsten Altenkirch, and Neil Ghani. Categories of containers. In Andrew D. Gordon, editor, *FOSSACS 2003 proceedings*, volume 2620 of *Lecture Notes in Computer Science*, pages 23–38. Springer, 2003. doi:10.1007/3-540-36576-1_2.

3 Danel Ahman and Andrej Bauer. Comodule representations of second-order functionals. *J. Log. Algebraic Methods Program.*, 2025. doi:10.1016/J.JLAMP.2025.101071.

4 Danel Ahman and Andrej Bauer. Sheaves as oracle computations, 2026. doi:10.48550/arXiv.2602.22135.

5 André Arnold and Damian Niwiński. Continuous separation of game languages. *Fundamenta Informaticae*, 81(1-3):19–28, 2007. doi:10.3233/FUN-2007-811-303.

6 Andrej Bauer. *The realizability approach to computable analysis and topology.* PhD thesis, Carnegie Mellon University, 2000.

7 Andrej Bauer. Instance reducibility and Weihrauch degrees. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/LMCS-18(3:20)2022.

8 Mikołaj Bojańczyk and Wojciech Czerwiński. An automata toolbox. December 30th 2025 version, 2018. URL: https://www.mimuw.edu.pl/~bojan/paper/automata-toolbox-book.

9 Filippo Bonchi, Daniela Petrisan, Damien Pous, and Jurriaan Rot. Coinduction up-to in a fibrational setting. In *CSL-LICS*. ACM, 2014. doi:10.1145/2603088.2603149.

10 Vasco Brattka. Weihrauch complexity and the Hagen school of computable analysis, 2022. arXiv:2203.06166.

11 Vasco Brattka. Loops, inverse limits and non-determinism, 2025. doi:10.48550/arXiv.2501.17734.

12 Vasco Brattka and Guido Gherardi. Weihrauch degrees, omniscience principles and weak computability. *Journal of Symbolic Logic*, 76:143–176, 2011. arXiv:0905.4679.

13 Vasco Brattka, Guido Gherardi, and Arno Pauly. *Weihrauch Complexity in Computable Analysis*, pages 367–417. Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-59234-9_11.

14 Vasco Brattka and Arno Pauly. On the algebraic structure of Weihrauch degrees. *Logical Methods in Computer Science*, 14(4), 2018. doi:10.23638/LMCS-14(4:4)2018.

15 Vasco Brattka and Hendrik Smischliaew. Computability of initial value problems. In *CiE*, 2025. doi:10.1007/978-3-031-95908-0_14.

16 John Burgess. Classical hierarchies from a modern standpoint. part II. R-sets. *Fundamenta Mathematicae*, 115:97–105, 1983.

17 A. Carboni and G. Rosolini. Locally cartesian closed exact completions. *Journal of Pure and Applied Algebra*, 154(1):103–116, 2000. doi:10.1016/S0022-4049(99)00192-9.

18 Aurelio Carboni, Stephen Lack, and R.F.C. Walters. Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra*, 84(2):145–158, 1993. doi:10.1016/0022-4049(93)90035-R.

19 Vittorio Cipriani, Alberto Marcone, and Manlio Valenti. The Weihrauch lattice at the level of $\mathbf{\Pi}^1_1$-CA_0 : the Cantor-Bendixson theorem. *J. Symb. Log.*, 90(2):752–790, 2025. doi:10.1017/JSL.2024.72.

20 Pierre Clairambault and Peter Dybjer. The biequivalence of locally cartesian closed categories and Martin-Löf type theories. *Math. Struct. Comput. Sci.*, 24(6), 2014. doi:10.1017/S0960129513000881.

21 Paul-Elliot Anglès d'Auriac and Takayuki Kihara. A comparison of various analytic choice principles. *The Journal of Symbolic Logic*, 86(4):1452–1485, 2021. doi:10.1017/JSL.2021.37.

22 Alessandro Facchini, Filip Murlak, and Michal Skrzypczak. Index problems for game automata. *ACM Trans. Comput. Log.*, 17(4):24, 2016. doi:10.1145/2946800.

23 Marcelo Fiore, Zeinab Galal, and Farzad Jafarrahmani. Fixpoint constructions in focused orthogonality models of linear logic. In *MFPS*, ENTICS, 2023. doi:10.46298/ENTICS.12302.

24 Harvey Friedman. On the necessary use of abstract set theory. *Advances in Mathematics*, 41(3):209–280, 1981.

25 Nicola Gambino and Joachim Kock. Polynomial functors and polynomial monads. *Mathematical Proceedings of the Cambridge Philosophical Society*, 154(1):153–192, September 2012. doi:10.1017/s0305004112000394.

26 Jean-Yves Girard. Normal functors, power series and λ-calculus. *Annals of pure and applied logic*, 37(2):129–177, 1988. doi:10.1016/0168-0072(88)90025-5.

27 Ichiro Hasuo, Toshiki Kataoka, and Kenta Cho. Coinductive predicates and final sequences in a fibration. *Math. Struct. Comput. Sci.*, 2018. doi:10.1017/S0960129517000056.

28 Michael David Hirsch. *Applications of topology to lower bound estimates in computer science*. PhD thesis, University of California, Berkeley, 1990.

29 Pierre Hyvernat. A linear category of polynomial functors (extensional part). *Log. Methods Comput. Sci.*, 2014. doi:10.2168/LMCS-10(2:2)2014.

30 Pierre Hyvernat. Totality for mixed inductive and coinductive types. *Logical Methods in Computer Science*, 2025. doi:10.46298/lmcs-21(3:19)2025.

31 Olivier Idir and Karoliina Lehtinen. Using games and universal trees to characterise the nondeterministic index of tree languages. In *ICALP*, 2025. doi:10.4230/LIPIcs.ICALP.2025.160.

32 B. Jacobs. *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier, 2001.

33 G Max Kelly. A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on. *Bulletin of the Australian Mathematical Society*, 22(1):1–83, 1980.

34 Takayuki Kihara. Rethinking the notion of oracle: A prequel to Lawvere-Tierney topologies for computability theorists, 2024. `arXiv:2202.00188`.

35 Leszek Aleksander Kołodziejczyk and Henryk Michalewski. How unprovable is Rabin's decidability theorem? In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 788–797, 2016. `doi:10.1145/2933575.2934543`.

36 Dexter Kozen and Alexandra Silva. Left-handed completeness. In Wolfram Kahl and Timothy G. Griffin, editors, *Relational and Algebraic Methods in Computer Science*, pages 162–178, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. `doi:10.1007/978-3-642-33314-9_11`.

37 Stéphane Le Roux and Arno Pauly. Weihrauch degrees of finding equilibria in sequential games. In *Conference on Computability in Europe*, pages 246–257. Springer, 2015. `doi:10.1007/978-3-319-20028-6_25`.

38 Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

39 Alberto Marcone and Manlio Valenti. The open and clopen Ramsey theorems in the Weihrauch lattice. *The Journal of Symbolic Logic*, 86(1):316–351, March 2021. `doi:10.1017/jsl.2021.10`.

40 Samuele Maschio and Davide Trotta. A topos for extended Weihrauch degrees, 2025. `arXiv:2505.08697`.

41 Ieke Moerdijk and Erik Palmgren. Wellfounded trees in categories. *Annals of Pure and Applied Logic*, 104(1-3):189–218, 2000. `doi:10.1016/S0168-0072(00)00012-9`.

42 M. Möllerfeld. *Generalized Inductive Definitions - The μ-calculus and Π^1_2-comprehension*. PhD thesis, Westfälischen Wilhelms-Universität Münster, 2002. Available at `https://miami.uni-muenster.de/Record/9dfa74b6-186b-4e95-a51f-9965d7e1e508`.

43 Sean K. Moss and Tamara von Glehn. Dialectica models of type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 739–748, 2018. `doi:10.1145/3209108.3209207`.

44 Eike Neumann and Arno Pauly. A topological view on algebraic computation models. *J. Complex.*, 44:1–22, 2018. `doi:10.1016/j.jco.2017.08.003`.

45 Eike Neumann, Arno Pauly, and Cécilia Pradic. The equational theory of the Weihrauch lattice with multiplication. *CoRR*, abs/2403.13975, 2024. `doi:10.48550/arXiv.2403.13975`.

46 Damian Niwiński and Michał Skrzypczak. On Guidable Index of Tree Automata. In *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 81:1–81:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.MFCS.2021.81`.

47 Leonardo Pacheco and Keita Yokoyama. Determinacy and reflection principles in second-order arithmetic, 2023. `arXiv:2209.04082`.

48 Arno Pauly. Computability on the space of countable ordinals, 2017. `arXiv:1501.00386`.

49 Cécilia Pradic and Ian Price. Weihrauch problems as containers. In Arnold Beckmann, Isabel Oitavem, and Florin Manea, editors, *Crossroads of Computability and Logic: Insights, Inspirations, and Innovations*, pages 395–409, Cham, 2025. Springer Nature Switzerland. `doi:10.48550/arXiv.2501.17250`.

50 Cécilia Pradic. The equational theory of the Weihrauch lattice with (iterated) composition, 2025. `doi:10.48550/arXiv.2408.14999`.

51 Egbert Rijke, Michael Shulman, and Bas Spitters. Modalities in homotopy type theory. *Log. Methods Comput. Sci.*, 16(1), 2020. `doi:10.23638/LMCS-16(1:2)2020`.

52 Luigi Santocanale. μ-bicomplete categories and parity games. *RAIRO-Theoretical Informatics and Applications*, 36(2):195–227, 2002. `doi:10.1051/ITA:2002010`.

53 Stephen G. Simpson. *Subsystems of second order arithmetic.* Perspectives in Mathematical Logic. CUP, 1999. `doi:10.1007/978-3-642-59971-2`.

54 Thomas Streicher. Fibered categories à la Jean Bénabou, 2023. arXiv:1801.02927.

55 Andrew Swan. Oracle modalities. *CoRR*, abs/2406.05818, 2024. doi:10.48550/arXiv.2406.05818.

56 Kazuyuki Tanaka. Weak axioms of determinacy and subsystems of analysis I: Δ^0_2 games. *Mathematical Logic Quarterly*, 36(6):481–491, 1990.

57 Kazuyuki Tanaka. Weak axioms of determinacy and subsystems of analysis II: Σ^0_2 games. *Ann. Pure Appl. Logic*, 52(1-2):181–193, 1991. doi:10.1016/0168-0072(91)90045-N.

58 Davide Trotta, Manlio Valenti, and Valeria de Paiva. Categorifying computable reducibilities. *Logical Methods in Computer Science*, Volume 21, Issue 1, February 2025. doi:10.46298/lmcs-21(1:15)2025.

59 Benno van den Berg and Federico De Marchi. Non-well-founded trees in categories. *Annals of Pure and Applied Logic*, 146(1):40–59, 2007. doi:10.1016/J.APAL.2006.12.001.

60 Mark Weber. Polynomials in categories with pullbacks. *Theory Appl. Categ*, 30(16):533–598, 2015.

61 Linda Westrick. A note on the diamond operator. *Computability*, 10(2):107–110, 2021. doi:10.3233/COM-200295.

62 K. Yoshimura. General treatment of non-standard realizabilites, 2016. Unpublished.