Solving General Multivariate Recursive Domain Equations in Coq

Sergio Domínguez Cabrera
sergio.dominguezcabrera@ru.nl

Abstract. This paper addresses the challenge of solving recursive domain equations in Coq for type constructors with multiple arguments that appear in both positive and negative positions. Existing frameworks primarily focus on type constructors with a single argument, limiting their applicability to more complex scenarios. While the theoretical foundation for solving such equations with multiple arguments has been established, specific proof details for the general case are missing. This work bridges that gap by providing a comprehensive formalization in Coq. Our formalization is based on the category of ordered families of equivalences (OFEs) and non-expansive functions, a widely adopted and practical model for guarded recursion. This choice enhances the relevance and applicability of our contributions to ongoing research and projects in the field.

1. Introduction

Recursive definitions are pervasive in computer science and mathematics, particularly in programming language semantics, where recursively defined types and relations play a crucial role. However, the usual inductive definitions are not always sufficient to capture the complexity of some recursive constructs, particularly those that are ill-founded in a classical set-theoretical context. These challenges often arise when recursive variables occur in negative positions. Suppose, for example, that A is a set with at least two elements and we want to find a set X satisfying the equation

$$X = F(X), \quad \text{where} \quad F(X) = A^X \quad \text{and}$$

A^X is the set of functions from X to A. For cardinality reasons, this set cannot exist in a classical set-theoretical context.

Step-indexing, first introduced by Appel and McAllester [2], provides a semantic framework for addressing the challenges posed by recursive definitions as described above. The key idea is to represent recursive constructs as sequences of approximations, where each element in the sequence reflects a bounded number of computational steps. This methodology makes the recursive definition well-founded by only needing to consider a finite number of steps at each stage. In this way, we construct a sequence of sets $\{X_n\}_{n\in\mathbb{N}}$, starting from any singleton set, $X_0 = 1$, and defining $X_{n+1} = F(X_n)$.

This methodology has proven effective in a variety of applications, including models for recursive types (e.g. Appel et al. [3]), program logics (e.g. Jung et al. [17]), and programming language semantics involving dynamically allocated higher-order store (e.g. Birkedal et al. [6]), where it offers a structured and reliable approach to complex recursive constructs.

Despite its utility and success, direct application of step-indexing can be cumbersome and error-prone. To address this, Appel et al. [3] introduced the later modality ($\triangleright$) based on the approximation modality by Nakano [22], an operator on predicates that abstracts step-indexing by allowing references to data available only in future computational steps. This innovation simplified reasoning in applications like program logics, such as Iris [17].

Building on the idea of the later modality, Birkedal et al. [5] extended its scope from predicates to type constructors, establishing the foundation for guarded recursion at the type level. To formalize this concept, they introduced an axiomatic characterization of models for guarded recursion. Their framework specifies the requirements a category must meet to serve as a valid model, with a key precondition being the presence of an endofunctor ($\blacktriangleright$) that generalizes the later modality to type constructors. The use of category theory in the context of recursive domain equations dates back to the seminal work of Smyth and Plotkin [24]. Since then, category theory has played a central role in the development of guarded recursion, providing a unified framework for defining and reasoning about models of guarded recursion, without the need for explicit mention of the underlying constructions.

Problem. The investigation into solving recursive domain equations in Coq has been explored in prior works, such as Sieczkowski et al. [23] and Jung et al. [17], has explored solving recursive domain equations in Coq by focusing on type constructors with a single argument, avoiding the complexity of categorical semantics. Their approaches rely on specialized definitions, such as enriched functors, tailored to this restricted case. However, the broader class of type constructors with multiple arguments remains unaddressed. However, generalizing to type constructors with multiple arguments requires categorical constructs, including enriched categories and their products, to define a unified enriched functor applicable to all possible cases. This extension remains a significant challenge, as it introduces substantial theoretical and formalization complexity, particularly in Coq.

Addressing this gap is essential, particularly in light of practical applications. For example, in the Actris project [16], researchers faced substantial challenges when attempting to define a type constructor that necessitated multiple arguments in a recursive position. They were forced to rely on a specific workaround that succeeded in their context, yet this reliance underscores the lack of a general solution, revealing a critical limitation in existing frameworks.

The theoretical foundations for addressing these general recursive domain equations were established by Birkedal et al. [5]. Nonetheless, the proof details for the general case involving multiple arguments are missing. This gap led to an addendum by Gratzer [14], which provided a complete proof of the theorem but required a revised definition of symmetrization that diverged from the original work. Despite these advancements, the new definition gave rise to certain formal issues that were likely overlooked at the time, possibly due to the lack of formalization in a proof assistant.

These unresolved issues highlight the complexity of the challenge at hand and emphasize the pressing need for formalization of this theoretical framework within a proof assistant like Coq.

Solution. In this paper, we present a formalization of the theoretical framework developed by Birkedal et al. [5] and Gratzer [14] within the Coq proof assistant. As part of this effort, we resolve the issues in the definition of symmetrization and provide a complete formal proof of the general existence theorem for recursive domain equations.

Rather than pursuing the full generality of the original theoretical framework, our formalization strategically focuses on a specific and practical model for guarded recursion: the category of ordered families of equivalences (OFEs) and non-expansive functions. While Birkedal et al. [5] utilized the topos of trees — the category of presheaves over the first infinite ordinal, ω— as their primary model, we propose an alternative. The topos of trees is undoubtedly intricate and less accessible for formalization in Coq. In contrast, the category of OFEs has been thoroughly investigated and successfully employed in notable prior research, including the works of Sieczkowski et al. [23] and Jung et al. [17]. Thus, our emphasis on OFEs not only enhances practicality but also fosters seamless integration with ongoing projects that leverage this model for guarded recursion.

Contributions This paper makes the following contributions:

- We clarify and resolve some formal issues in the definition of symmetrization (Subsection 4.4).
- We provide a complete formal proof of the general existence theorem for recursive domain equations (Theorem 4.16).

All results are formalized in Coq and are available in the accompanying repository [9].

2. Ordered Families of Equivalences

In this section, we introduce the basic definitions and results about ordered families of equivalences (OFEs). These structures, along with many other associated concepts, were introduced by Gianantonio and Miculan [13]. Moreover, we will prove the well-known result that OFEs form a category with a rich structure.

OFEs are a generalization of the notion of a setoid —a set equipped with an equivalence relation— to the setting of step-indexing.

Definition 2.1. *An **ordered family of equivalences** (OFE) is a set X equipped with a family of equivalence relations $(\overset{n}{=} \subseteq X \times X)_{n \in \mathbb{N}}$ such that for all $x, y \in X$*

$$\forall n, m \in \mathbb{N},\ x \overset{n}{=} y \to x \overset{m}{=} y \text{ if } m \leq n, \qquad (\text{MONO})$$

$$(\forall n \in \mathbb{N},\ x \overset{n}{=} y) \to x = y \qquad (\text{LIMIT})$$

The intuition behind the definition of an OFE is that if two elements are $\overset{n}{=}$-related for some n, then they are equivalent for n steps of computation. As we increase the number of steps, the equivalence relation becomes more and more refined, that is, the less elements are related. At the limit, the relation becomes the plain equality relation.

Example 2.2. *The following are examples of OFEs:*

1. *Any set X, together with the family of equivalence relations where $\overset{n}{=}$ is the equality relation for all $n \in \mathbb{N}$, is an OFE. We will denote this OFE by ΔX and we will refer to it as the **discrete OFE**.*
2. *The set of infinite sequences X^ω over an OFE X, together with the family of equivalence relations where $\overset{n}{=}$ is the pointwise equality relation, that is,*

$$x \overset{n}{=} y \leftrightarrow \forall i \leq n, x_i \overset{n}{=} y_i$$

is an OFE.

We aim to define a category whose objects are OFEs and our next step is to define the morphisms between them. Following the intuition that equivalence relations capture the notion of computation steps, we want our morphisms to preserve the equivalence up to a certain number of steps. The following definition captures this idea.

Definition 2.3. *Let $(X, (\overset{n}{=}_X)_{n \in \mathbb{N}})$ and $(Y, (\overset{n}{=}_Y)_{n \in \mathbb{N}})$ be two OFEs. A function $f : X \to Y$ is **non-expansive** if for all $n \in \mathbb{N}$ and all $x, y \in X$,*

$$\text{if } x \overset{n}{=}_X y, \text{ then } f(x) \overset{n}{=}_Y f(y).$$

From now on, we will say that a set X is an OFE without giving the equivalence relations explicitly and we will drop the subscripts X from $\overset{n}{=}_X$ since the context is always clear. Moreover, we will denote a non-expansive function f from X to Y by $\mathsf{f} : X \to_{\text{ne}} Y$.

Example 2.4. *The following are examples of non-expansive functions:*

1. *The identity function* $\mathsf{id} : X \to X$ *is non-expansive for any OFE* X.
2. *The constant function* $\Delta_y : X \to Y$ *is non-expansive for any OFEs* X *and* Y, *and any* $y \in Y$.

We will denote the category of OFEs and non-expansive functions by **OFE**. This category forms a *hyperdoctrine*, a concept introduced by Lawvere [20], which provides a framework for defining an internal higher-order logic.

In simple terms, a hyperdoctrine is a cartesian closed category equipped with a special object, known as the *generic object*, which represents the type of propositions. In the category of OFEs, this object is:

$$\mathsf{SProp} = \{P \in \mathcal{P}(\mathbb{N}) \mid \forall n \geq m, n \in P \to m \in P\},$$

the set of downward-closed subsets of $\mathbb{N}$. The usual propositional logic connectives can be defined in SProp. In addition, for this object to form a hyperdoctrine, we need to be able to define quantifiers in the internal logic. More details about categorical logic and the internal logic of this category can be found in the work of Birkedal and Bizjak [4].

Additionally, in our case, there exists a morphism

$$\triangleright : \mathsf{SProp} \to_{\text{ne}} \mathsf{SProp}, \quad P \mapsto \{0\} \cup \{n+1 \mid n \in P\},$$

which corresponds to the *later modality* in the internal logic of OFEs.

2.1. Basic Constructions in OFE

In this section, we introduce some constructions that witness the rich structure of the category of OFEs. Most of these constructions are done pointwise, that is, we define them in terms of the constructions on the category of sets. These constructions include the terminal object, the pullbacks, and the exponentials

Lemma 2.5. *The **terminal object** in* **OFE** *is the discrete OFE of a singleton set,* $\Delta\{*\}$. *From now on, we will denote this object by* 1.

Lemma 2.6. *Let* $f : X \to_{\text{ne}} Z$ *and* $g : Y \to_{\text{ne}} Z$ *be two non-expansive functions. Then, the following set*

$$X \times_Z Y := \{(x, y) \in X \times Y \mid f(x) = g(y)\}$$

together with the equivalence relations

$$(x_1, y_1) \overset{n}{=} (x_2, y_2) \leftrightarrow x_1 \overset{n}{=} x_2 \text{ and } y_1 \overset{n}{=} y_2$$

*is the **pullback** of* f *and* g *in* **OFE**.

The above constructions imply that the category of OFEs has all finite limits (see Borceux [7] Prop. 2.8.2 for a detailed proof). Since we have the pullback construction and the terminal object, we can define the product object in **OFE** as the pullback of the two unique functions from X and Y to the terminal object.

Lemma 2.7. *Let* X *and* Y *be OFEs. Denote by* Y^X *the set of non-expansive functions from* X *to* Y. *The set* Y^X *is an OFE with equivalence relations defined as follows:*

$$f \overset{n}{=} g \leftrightarrow \forall x \in X, f(x) \overset{n}{=} g(x).$$

*and it is the **exponential object** in* **OFE**.

The above constructions imply that the category of OFEs is cartesian closed.

2.2. The Later Construction

The later construction is a fundamental concept in the category of OFEs. It plays a crucial role in defining guarded recursive domain equations introducing a notion of time into the internal logic of the category of OFEs. This temporal aspect enables the definition of fixed points for a specific class of non-expansive functions (as formalized in Theorem 2.21).

Definition 2.8. *Let* X *be an OFE. The **later OFE** on* X, *denoted* $\blacktriangleright X$, *is defined as the set:*

$$\blacktriangleright X = \{\mathsf{next}(x) \mid x \in X\},$$

equipped with the following family of equivalence relations:

$$\mathsf{next}(x) \overset{n}{=} \mathsf{next}(y) \leftrightarrow n = 0 \lor x \overset{n-1}{=} y.$$

This construction provides a mapping on objects. To extend it to a fully defined endofunctor in the category of OFEs, we must also define its action on morphisms. The following definition accomplishes this:

Definition 2.9. *Let* $f : X \to_{\text{ne}} Y$ *be a non-expansive function. The action of the later endofunctor on* f, *denoted* $\blacktriangleright f$, *is the function* $\blacktriangleright f : \blacktriangleright X \to_{\text{ne}} \blacktriangleright Y$ *defined as:*

$$\blacktriangleright f(\mathsf{next}(x)) = \mathsf{next}(f(x)).$$

Remark 2.10. *The definition of the later endofunctor's action on morphisms aligns with the naturality condition. Consequently, the later endofunctor is equipped with a natural transformation* $\mathsf{next} : \mathsf{id} \to \blacktriangleright$.

The later endofunctor also preserves finite limits. Therefore, for any two OFEs X and Y, the following morphisms always exist:

$$K : \blacktriangleright(X \times Y) \to_{\text{ne}} (\blacktriangleright X) \times (\blacktriangleright Y),$$
$$J : \blacktriangleright(Y^X) \to_{\text{ne}} (\blacktriangleright Y)^{(\blacktriangleright X)}.$$

Notably, K is an isomorphism.

2.3. Chains and Complete OFEs

The step-indexed equality of an OFE allows us to define chains of elements in an OFE, which are sequences of elements that become "more and more equal". Chains with limits are interesting because they can be used to define an object as the limit of a sequence of approximations, rather than defining the object directly. Let us define these concepts formally.

Definition 2.11. *A **chain** in an OFE X is a sequence $(c_n)_{n\in\mathbb{N}}$ of elements that satisfy the cauchy property:*

$$\forall n, m.\ n \leq m \Rightarrow c_m \overset{n}{=} c_n$$

*We say that a chain has a **limit** if there exists an element $c_\infty \in X$ such that*

$$\forall n.\ c_n \overset{n}{=} c_\infty$$

*If every chain in X has a limit, then X is said to be **complete** (it is a COFE).*

Complete OFEs form a full subcategory of OFEs, which we denote by **COFE**. Fortunately, this category has all finite limits, products, and exponentials. From now on, we will focus on the category of COFEs, as it has favorable properties for defining fixed points of non-expansive functions. Since the category of COFEs is included in the category of OFEs, the later endofunctor can be restricted to COFEs.

2.4. Contractive Morphisms

Contractive morphisms are a special class of non-expansive functions that are closely related to the later endofunctor. These morphisms are particularly useful since they possess fixed points that can be defined using the Banach fixed-point theorem (as we will see in Theorem 2.21).

Definition 2.12. *A function $f : X \to Y$ is **contractive**, denoted $f : X \to_{\text{ctr}} Y$, if for all $x, y \in X$ and $n \in \mathbb{N}$,*

$$(\forall m < n,\ x \overset{m}{=} y) \to f(x) \overset{n}{=} f(y).$$

It is straightforward to verify that every contractive function is non-expansive, though the converse does not hold in general (take the identity for example). Consequently, each contractive function is a morphism in the category of (C)OFEs.

Example 2.13. *The following are examples of contractive functions:*

1. *Each component of the next natural transformation $\mathsf{next}_X : X \to_{\text{ne}} \blacktriangleright X$ is contractive.*
2. *The internal later modality $\triangleright : \mathsf{SProp} \to_{\text{ne}} \mathsf{SProp}$ is contractive.*

Lemma 2.14 (Properties of Contractive Morphisms)**.**

1. *If either $f : X \to_{\text{ne}} Y$ or $g : Y \to_{\text{ne}} Z$ is contractive, then $g \circ f : X \to_{\text{ne}} Z$ is contractive.*
2. *If $f : X \to_{\text{ne}} Y$ and $g : X' \to_{\text{ne}} Y'$ are both contractive, then $f \times g$ is contractive.*

A central concept in this paper is the notion of the later modality, which we now use to characterize contractive morphisms. The following lemma encapsulates the main idea.

Lemma 2.15. *A morphism $f : X \to_{\text{ne}} Y$ is **contractive** if and only if there exists a morphism $g : \blacktriangleright X \to_{\text{ne}} Y$ such that $f = g \circ \mathsf{next}_X$.*

Remark 2.16. *In the above lemma, we use the $\to_{\text{ne}}$ notation for f to emphasize that every contractive morphism is non-expansive.*

Applying Lemma 2.15, we can easily show that every component of the next natural transformation is contractive (Example 2.13, 1), by taking g to be the identity function.

2.5. Partial Contractive Morphisms

The concept of contractiveness can be sometimes too restrictive and there are some cases where it suffices to require contractiveness in only one argument. The following definition captures this idea.

Definition 2.17. *A function $f : X \times Y \to_{\text{ne}} Z$ is contractive in the first argument if, forall $y \in Y$, $x_1, x_2 \in X$ and $n \in \mathbb{N}$,*

$$(\forall m < n,\ x_1 \overset{m}{=} x_2) \to f(x_1, y) \overset{n}{=} f(x_2, y).$$

The previous definition can be easily generalized to the case where the domain is an arbitrary product and considering contractiveness in the i-th argument.

Remark 2.18. *Note that partial contractiveness is a weaker condition than contractiveness. In particular, every contractive function is also contractive in any of its arguments. However, the converse does not hold in general.*

Example 2.19. *Recall the definitions of the discrete OFE and the OFE of infinite sequences from Example 2.2. Let X be a set with at least two distinct elements x and y. The prepending function,*

$$\begin{aligned} \mathsf{cons} : \Delta X \times (\Delta X)^\omega &\to_{\text{ne}} (\Delta X)^\omega \\ (x, s) &\mapsto x :: s, \end{aligned}$$

is not globally contractive. To see this, consider the elements $(x, \lambda n.x), (y, \lambda n.y) \in \Delta X \times (\Delta X)^\omega$ and let $n = 0$ in the definition of contractiveness. Then, we have to prove that $\mathsf{cons}\ x\ (\lambda n.x)$ and $\mathsf{cons}\ y\ (\lambda n.y)$ are 0-related. Hence, we need to show that $x = y$, which contradicts the assumption that x and y are distinct.

Nevertheless, we can show that cons is contractive in the second argument since now the information about the head elements being equal is provided. Thus, the function $\mathsf{cons}\ x$ becomes globally contractive for every $x \in X$.

The later modality is again useful for characterizing partial contractiveness. The following definition captures this idea.

Lemma 2.20. *A function $f : X \times Y \to_{\text{ne}} Z$ is contractive in the first argument if and only if there exists a $g : \blacktriangleright X \times Y \to_{\text{ne}} Z$ such that $f = g \circ (\mathsf{next}_X \times \mathsf{id}_Y)$.*

2.6. Inhabited COFEs

When working with fixed points, the process often starts with an arbitrary element, which serves as the initial point for an iterative sequence. From there, the sequence is refined step by step by repeatedly applying a function until it converges.

This leads to the idea of inhabited COFEs (iCOFEs), which are COFEs that always include an arbitrary element. Similarly to what we did with COFEs, we can define a category of iCOFEs, denoted by **iCOFE**. The category of iCOFEs has all the properties mentioned earlier for the category of COFEs. However, we must note that we are losing some properties, such as the existence of an initial object. Nonetheless, the trade-off is worthwhile, as it ensures that fixed points always exist.

The following chain of inclusions illustrates the relationship between the categories we have defined so far.

$$\mathbf{iCOFE} \hookrightarrow \mathbf{COFE} \hookrightarrow \mathbf{OFE}$$

Contractive morphisms are particularly useful here because they make it possible to apply the Banach fixed-point theorem to iCOFEs. This theorem provides a simple and elegant way to prove that a unique fixed point exists for any contractive morphism.

Theorem 2.21 (Banach Fixed-Point Theorem)**.** *Let X be a iCOFE and let $f : X \to_{\mathrm{ctr}} X$ be a contractive morphism. Then, there exists a unique element $x \in X$ such that $f(x) = x$.*

3. America-Rutten Theorem

In this section, we present the America-Rutten theorem, originally introduced by America and Rutten [1], in two forms. The first form (Theorem 3.4) corresponds to the original theorem, which applies to type constructors with only positive occurrences of the recursive variable. The second form (Theorem 3.7) extends this result to type constructors with both positive and negative occurrences of the recursive variable. To avoid ambiguity, we refer to the second form as the America-Rutten theorem. This theorem is a simplified instance of the general existence theorem discussed in Section 4.

The proof proceeds in two steps. First, we establish the existence of simple recursive types —those where the recursive variable appears only in positive positions— as stated in America and Rutten [1]. Second, we show how this result extends to type constructors with both positive and negative occurrences of the recursive variable

3.1. Simple Recursive Types

To ensure the existence of simple recursive types in the category of iCOFEs, we need to extend the notion of contractiveness to endofunctors, analogous to morphisms in the category of iCOFEs. Before formalizing this, let us briefly review how functors operate on morphisms.

For any two objects X and Y in **iCOFE**, an endofunctor F induces a function:

$$F_{X,Y} : \mathrm{Hom}(X, Y) \to \mathrm{Hom}(FX, FY).$$

Since **iCOFE** is cartesian closed, the hom-sets $\mathrm{Hom}(X, Y)$ are iCOFEs. Hence, we can write:

$$F_{X,Y} : Y^X \to FY^{FX}.$$

However, $F_{X,Y}$ is not guaranteed to be non-expansive, as it is defined in the category of sets.

Definition 3.1. *A functor F : **iCOFE** $\to$ **iCOFE** is:*

- ***locally non-expansive** if, for all objects X and Y, the map $F_{X,Y} : Y^X \to FY^{FX}$ is non-expansive.*
- ***locally contractive** if, for all objects X and Y, the map $F_{X,Y} : Y^X \to FY^{FX}$ is contractive.*

*We denote these by F : **iCOFE** $\to_{\mathrm{ne}}$ **iCOFE** and F : **iCOFE** $\to_{\mathrm{ctr}}$ **iCOFE**, respectively.*

Example 3.2. *The* later *endofunctor, ($\blacktriangleright$), is locally contractive. This can be shown using the morphism J (Section 2.2) and the characterization of contractive morphisms via the later modality (Lemma 2.15).*

Remark 3.3. *Locally contractive endofunctors inherit the properties of regular contractive morphisms (Lemma 2.14). Notably, the composition of a locally contractive endofunctor with a locally non-expansive endofunctor (in either order) is locally contractive, and every locally contractive endofunctor is locally non-expansive.*

The following theorem ensures the existence of recursive types that reference themselves only in positive positions.

Theorem 3.4. *The category of iCOFEs is contractively complete. That is, every locally contractive endofunctor F : **iCOFE** $\to_{\mathrm{ctr}}$ **iCOFE** has a unique (up to isomorphism) fixed point, i.e., an object $\lim F$ such that $F(\lim F) \cong \lim F$.*

We will not present a detailed proof of Theorem 3.4 here but will outline the main ideas. The construction is based on the process described in Sieczkowski et al. [23]. The central idea is to define an iCOFE $\lim F$ that represents the iterative application of the endofunctor F to the initial object, forming a tower of approximations.

Consider the sequence:

$$1 \xleftarrow{\;!_{F1}\;} F\,1 \xleftarrow{\;F\,!_{F1}\;} F^2\,1 \longleftarrow \;\cdots$$

where $!_{F1}$ is the unique morphism from $F\,1$ to the terminal object 1. To simplify the notation, we will denote by g_n the morphism $F^n\,!_{F1}$. The iCOFE that will serve as the fixed point of F is :

$$\lim F \stackrel{\text{def}}{=} \{x : \Pi_{n\in\mathbb{N}}\, F^n 1 \mid \forall n,\ g_n(x_{n+1}) = x_n\}$$

The elements of this iCOFE are the sequences $(x_n)_{n\in\mathbb{N}}$, where x_n is contained in $F^n 1$ and which are compatible with the morphisms induced by the repeated application of F to $!_{F1}$. This concludes our sketch of the existence, now we shall move on to the uniqueness of the fixed point.

The uniqueness of the fixed point is equivalent to showing that any fixed point of F gives rise to an initial algebra and a final coalgebra for F (corresponding to being the least and greatest fixed points, respectively).

Consider an arbitrary isomorphism $f : FX \cong X$ and an arbitrary algebra structure $g : FY \to Y$. Then, for $h : X \to Y$ to be an algebra homomorphism, we need to show that the following diagram commutes:

$$\begin{tikzcd} FX \arrow[r, "Fh"] \arrow[d, "f"'] & FY \arrow[d, "g"] \\ X \arrow[r, "h"'] & Y \end{tikzcd}$$

This is equivalent to finding a fixed point of the morphism $T = \lambda h : Y^X.\ g \circ Fh \circ f^{-1}$, which is contractive because F is locally contractive. Therefore, the Banach fixed-point theorem guarantees the existence of a unique fixed point for T, which corresponds to the unique algebra homomorphism. The same reasoning applies to final coalgebras. For further details, see Lemma 7.6 in Birkedal et al. [5].

3.2. Mixed-Variance Types

We now address the case of mixed-variance type constructors, where the recursive variable appears in both positive and negative positions. Such type constructors are represented by mixed-variance endofunctors, which are functors of the form $F : \mathbf{iCOFE}^{op} \times \mathbf{iCOFE} \to \mathbf{iCOFE}$.

The notion of locally contractive endofunctors extends naturally to mixed-variance endofunctors.

Definition 3.5. *A mixed-variance endofunctor* $F : \boldsymbol{iCOFE}^{op} \times \boldsymbol{iCOFE} \to \boldsymbol{iCOFE}$ *is said to be* ***locally non-expansive*** *if, for all* X_1, X_2, Y_1, Y_2 *in* $\boldsymbol{iCOFE}$, *the function*

$$F_{X_1,X_2,Y_1,Y_2} : X_1^{Y_1} \times Y_2^{X_2} \to F(Y_1, Y_2)^{F(X_1,X_2)}$$

is non-expansive. Similarly, F is ***locally contractive*** *if this function is contractive.*

Example 3.6. *The following are examples of mixed-variance endofunctors:*

1. *Any locally contractive endofunctor* $F : \boldsymbol{iCOFE} \to \boldsymbol{iCOFE}$ *can be lifted to a mixed-variance endofunctor* F^{ext} *by defining* $F^{ext}(X^-, X^+) = FX^+$.
2. *The functor* $F : \boldsymbol{iCOFE}^{op} \times \boldsymbol{iCOFE} \to \boldsymbol{iCOFE}$ *defined by*

$$(X^-, X^+) \mapsto \mathsf{SProp}^{\left((\blacktriangleright X^- + B)^A\right)}$$

for any iCOFEs A and B is an example of a locally contractive mixed-variance endofunctor. When A represents the type of locations and B the type of values, the fixed point of this endofunctor corresponds to the type of logical predicates in the Iris logic (see Jung et al. [17]).

Now, we can state the America-Rutten theorem.

Theorem 3.7 (America-Rutten)**.** *Every mixed-variance locally contractive endofunctor* $F : \mathbf{iCOFE}^{op} \times \mathbf{iCOFE} \to_{\text{ctr}} \mathbf{iCOFE}$ *has a unique (up to isomorphism) fixed point, i.e., an object* $\lim F$ *such that* $F(\lim F, \lim F) \cong \lim F$.

To sketch the proof of the America-Rutten theorem, the first step is to construct a functor $\mu F : \mathbf{iCOFE}^{op} \to \mathbf{iCOFE}$ that assigns to each object X the fixed point of $F(X, -)$, and to each morphism f, the unique morphism $\mu F\, f$ that satisfies the following commutative diagram:

$$\begin{tikzcd} F(X, \mu F\, X) \arrow[r, "\cong"] \arrow[d, "F(f, \mu F\, f)"] & \mu F\, X \arrow[d, "\mu F\, f"] \\ F(Y, \mu F\, Y) \arrow[r, "\cong"] & \mu F\, Y \end{tikzcd}$$

This construction is done by leveraging the Banach fixed-point theorem. Since F is locally contractive, the functor μF is also locally contractive, that is, for any $X, Y \in \mathbf{iCOFE}^{op}$, the function

$$\mu F_{X,Y} : X^Y \to \mu F\, Y^{\mu F\, X}$$

is contractive.

The next step is to define a locally contractive endofunctor $\zeta F : \mathbf{iCOFE} \to \mathbf{iCOFE}$ by mapping an object X to $F(\mu F\, X, X)$ and a morphism f to $F(\mu F\, f, f)$. We then define $Z = \lim \zeta F$ using Theorem 3.4, and $W = \mu F\, Z$. Using these constructions, we obtain:

$$F(W, Z) = F(\mu F\, Z, Z) = \zeta F(Z) \cong Z,$$
$$F(Z, W) = F(Z, \mu F\, Z) \cong \mu F\, Z = W.$$

Next, we prove that Z and W are isomorphic. Denote by f and g the isomorphisms $F(W, Z) \cong Z$ and $F(Z, W) \cong W$, respectively. Consider the following diagram:

$$\begin{tikzcd} F(W, Z) \arrow[r, "F(h_1, h_1)"] \arrow[d, "f"', "\cong"] & F(Z, W) \arrow[d, "g"', "\cong"] \\ Z \arrow[r, "h_1"] & W \end{tikzcd}$$

Clearly, there is only one possible choice for h that makes the diagram commute, and this choice is the fixed point of the contractive morphism $T = \lambda h_1 : W^Z.\ g \circ F(h_1, h_1) \circ f^{-1}$. We proceed analogously to construct the unique morphism $h_2 : W \to Z$ that makes a similar diagram commute.

To prove that the composition in both directions is the identity, we construct the following contractive morphism

$$\begin{aligned} T' =& \lambda(x_1, x_2) : W^W \times Z^Z. \\ & (g \circ F(x_2, x_1) \circ g^{-1}) \times (f \circ F(x_1, x_2) \circ f^{-1}) \end{aligned}$$

This morphism has two fixed points, namely the pair of identity morphisms $(\mathsf{id}_W, \mathsf{id}_Z)$ and the pair consisting of the morphisms $(h_1 \circ h_2, h_2 \circ h_1)$. Since the fixed points of a contractive morphism are unique, we conclude that the composition in both directions is the identity, and hence $Z \cong W$.

To conclude the sketch of the proof of existence, we can take as the fixed point $\lim F$ the object Z (or W since they are isomorphic), and since all functors preserve isomorphisms, we have $F(\lim F, \lim F) \cong \lim F$.

Finally, to prove that any fixed point of F is isomorphic to $\lim F$, we employ a similar technique as in the proof of Theorem 3.4. In this case, any fixed point of F gives rise to a free F-dialgebra in the sense of Freyd [10] and Freyd [11] and, therefore, they are unique up to isomorphism.

A F-dialgebra is an object X equipped with two morphisms $f_1 : F(X, X) \to X$ and $f_2 : X \to F(X, X)$. Clearly, any fixed point of F is forms an F-dialgebra. Given two F-dialgebras (X, f_1, f_2) and (Y, g_1, g_2), a map of F-dialgebras is a pair of morphisms $h_1 : Y \to Z$ and $h_2 : Z \to Y$ such that the following diagrams commute:

$$
\begin{tikzcd}
F(Y,Y) \arrow[r, "F(h_1,h_1)"] \arrow[d, "f_1"'] & F(X,X) \arrow[d, "g_1"] \\
Y \arrow[r, "h_1"] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "h_2"] \arrow[d, "f_2"'] & Y \arrow[d, "g_2"] \\
F(X,X) \arrow[r, "F(h_2,h_2)"] & F(Y,Y)
\end{tikzcd}
$$

The property of being a free F-dialgebra means that for any F-dialgebra (Y, g_1, g_2), there exists a unique dialgebra homomorphism from the free dialgebra to (Y, g_1, g_2). Leveraging the banach fixed-point theorem we can show that any fixed point of F is a free F-dialgebra, and therefore, it is unique up to isomorphism.

4. General Existence Theorem

In this section, we present the general existence theorem for recursive types in any contractively complete **iCOFE**-enriched category. This theorem guarantees the existence of fixed points for type constructors that have both positive and negative occurrences of the various type arguments.

4.1. iCOFE-Enriched Categories & Functors

In the previous section, we worked with the self-enriched category of iCOFEs without explicitly invoking the concept of enriched categories. As a result, we encountered multiple definitions of non-expansiveness and contractiveness for functors, tailored to specific cases. To formulate and prove the general existence theorem, the framework of enriched categories becomes indispensable. It provides a unified foundation, enabling a single definition of non-expansiveness and contractiveness that applies universally to all type constructors. As we progress through this section, the necessity and advantages of using enriched categories will become clearer, particularly in how they streamline definitions and proofs for the general case.

The notion of enriched categories is a generalization of the classical notion of category (which is enriched over the category of sets). The difference is that the hom-sets are replaced by objects in the category we are enriching over. The idea of changing the hom-sets to hom-objects was first introduced by MacLane [21], but it was not until the work of Kelly [18] that the concept was fully developed.

We can only enrich over monoidal categories, which are categories equipped with a tensor product functor, a unit object, and associativity and unit laws. The category of iCOFEs is monoidal, since we can define a tensor product on it as follows:

$$
\begin{aligned}
\otimes : \mathbf{iCOFE} \times \mathbf{iCOFE} &\to \mathbf{iCOFE} \\
(X, Y) &\mapsto X \times Y \\
(f, g) &\mapsto f \times g
\end{aligned}
$$

The unit object is the terminal object 1, and the associativity and unit laws are satisfied.

Consequently, we can define the notion of an **iCOFE**-enriched category.

Definition 4.1. *An **iCOFE-enriched category** $\mathcal{C}$ (or simply **iCOFE**-category) consists of:*

- *A set of objects $\mathsf{Ob}(\mathcal{C})$.*
- *For each pair of objects X and Y in $\mathcal{C}$, an object $\mathcal{C}(X, Y)$ in **iCOFE**, which we call the hom-object.*
- *For each triple of objects X, Y, and Z in $\mathcal{C}$, a composition operation*

$$\circ_{X,Y,Z} : \mathcal{C}(Y, Z) \otimes \mathcal{C}(X, Y) \to_{\mathrm{ne}} \mathcal{C}(X, Z)$$

  that lives in the category of iCOFEs.
- *For each object X in $\mathcal{C}$, an identity element $\mathsf{ID}_X : 1 \to_{\mathrm{ne}} \mathcal{C}(X, X)$ that also lives in the category of iCOFEs.*
- *The composition and identity operations must satisfy the associativity and unit laws. In the case of the unit laws, we have that for all $A, B \in \mathsf{Ob}(\mathcal{C})$, the following diagrams commute:*

$$
\begin{tikzcd}
\mathcal{C}(B,B) \otimes \mathcal{C}(A,B) \arrow[r, "\circ_{A,B,B}"] & \mathcal{C}(A,B) \\
1 \otimes \mathcal{C}(A,B) \arrow[u, "\mathsf{ID}_B \otimes \mathsf{id}_{\mathcal{C}(A,B)}"] \arrow[ur, "l"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\mathcal{C}(A,B) \otimes \mathcal{C}(A,A) \arrow[r, "\circ_{A,B,B}"] & \mathcal{C}(A,B) \\
\mathcal{C}(A,B) \otimes 1 \arrow[u, "\mathsf{id}_{\mathcal{C}(A,B)} \otimes \mathsf{ID}_A"] \arrow[ur, "r"'] &
\end{tikzcd}
$$

  where l and r are the left and right unit natural transformations of the monoidal structure. The associativity law is defined similarly.

In simple terms, a **iCOFE**-enriched category is a category where the hom-sets are themselves iCOFEs that can be composed and have identities. Let's now see some examples of **iCOFE**-enriched categories.

Example 4.2. *The following are examples of **iCOFE**-enriched categories:*

1. *The category of iCOFEs is self-enriched, as the hom-sets form iCOFEs.*
2. *The one-object category with the hom-object being the terminal object is **iCOFE**-enriched.*
3. *The opposite category of an **iCOFE**-enriched category is also **iCOFE**-enriched.*
4. *The product category of two **iCOFE**-enriched categories is also **iCOFE**-enriched.*

We can now define the notion of enriched functors between **iCOFE**-enriched categories.

Definition 4.3. *An **iCOFE-enriched functor** $F : \mathcal{C} \to \mathcal{D}$ between two **iCOFE**-enriched categories $\mathcal{C}$ and $\mathcal{D}$ consists of:*

- *A function $F : \mathsf{Ob}(\mathcal{C}) \to \mathsf{Ob}(\mathcal{D})$.*
- *For each pair of objects X and Y in $\mathcal{C}$, a morphism in **iCOFE**:*

$$F_{X,Y} : \mathcal{C}(X,Y) \to_{\text{ne}} \mathcal{D}(FX,FY)$$

  where $\mathcal{C}(X,Y)$ and $\mathcal{D}(FX,FY)$ are the hom-objects in $\mathcal{C}$ and $\mathcal{D}$, both of which are iCOFEs.
- *Preservation of composition, but with this composition defined in the **iCOFE**-world. That is, for all objects $A, B, C \in \mathsf{Ob}(\mathcal{C})$, the following diagrams commute:*

$$\begin{tikzcd}
\mathcal{C}(B,C) \otimes \mathcal{C}(A,B) \arrow[r, "\circ_{A,B,C}"] \arrow[d, "F_{B,C} \otimes F_{A,B}"] & \mathcal{C}(A,C) \arrow[d, "F_{A,C}"] \\
\mathcal{D}(FB,FC) \otimes \mathcal{D}(FA,FB) \arrow[r, "\circ_{FA,FB,FC}"] & \mathcal{D}(FA,FC)
\end{tikzcd}$$

- *Preservation of identities in the same way.*

Remark 4.4. *Notice that when we required F to be a locally non-expansive functor in the previous section, we were implicitly saying that F is an **iCOFE**-enriched functor.*

4.2. Locally Contractive Functors

In the previous section, we defined different notions of contractiveness for different types of functors. In this section, we will define a general notion of contractiveness that encompasses all the previous definitions and allows us to state the general existence theorem.

Definition 4.5. *A functor $F : \mathcal{C} \to \mathcal{D}$ between two **iCOFE**-enriched categories is **locally contractive** if, for all objects X and Y in $\mathcal{C}$, the map $F_{X,Y} : \mathcal{C}(X,Y) \to_{\text{ne}} \mathcal{D}(FX,FY)$ is contractive.*

In the same way that we defined partial contractiveness for morphisms, we can define partial contractiveness for functors.

Definition 4.6. *Let $F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ be an **iCOFE**-enriched functor. We say that F is locally contractive in the first argument if for all objects $X, Y \in \mathcal{C} \times \mathcal{D}$, the map $F_{X,Y} : (\mathcal{C} \times \mathcal{D})(X,Y) \to_{\text{ne}} \mathcal{E}(FX,FY)$ is contractive in the first argument.*

This definition can be extended to any argument of the functor and to functors with more than two arguments similarly.

Remark 4.7. *Note that if a functor $F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ is locally contractive in the first argument, then for every object $X \in \mathcal{D}$, the functor $F(-,X) : \mathcal{C} \to \mathcal{E}$ is locally contractive. However, the converse is not necessarily true.*

4.3. Characterizing Contractive Functors

As we did for morphisms in Lemma 2.15, locally contractive functors can be characterized in terms of the later modality. Before we state the lemma, we need to introduce the notion of the *later* category of an **iCOFE**-enriched category.

Definition 4.8. *The **later category** of an **iCOFE**-enriched category $\mathcal{C}$, denoted $\blacktriangleright\mathcal{C}$, has:*

- *The same objects as $\mathcal{C}$,*
- *Morphisms defined by $(\blacktriangleright\mathcal{C})(X,Y) = \blacktriangleright(\mathcal{C}(X,Y))$,*

Remark 4.9. *The natural transformation next (see Remark 2.10) defines an **iCOFE**-enriched functor for any **iCOFE**-enriched category $\mathcal{C}$ to its later category $\blacktriangleright\mathcal{C}$. This functor acts as the identity on objects and lifts morphisms through the later modality.*

Now, we can state the characterization of locally contractive **iCOFE**-enriched functors in terms of the later category.

Lemma 4.10. *An **iCOFE**-enriched functor F between two **iCOFE**-enriched categories $\mathcal{C}$ and $\mathcal{D}$ is locally contractive if and only if there exists an **iCOFE**-enriched functor $G : \blacktriangleright\mathcal{C} \to \mathcal{C}$ such that $F = G \circ \mathsf{next}$.*

Partial contractiveness can also be characterized in terms of the later modality.

Lemma 4.11. *Let $F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$ be an **iCOFE**-enriched functor. We say that F is locally contractive in the first argument if and only if there exists an **iCOFE**-enriched functor $G : \blacktriangleright\mathcal{C} \times \mathcal{D} \to \mathcal{E}$ such that $F = G \circ (\mathsf{next}_{\mathcal{C}} \times \mathsf{id}_{\mathcal{D}})$.*

Analogously, we can define partial contractiveness in terms of the later category for any argument of the functor and for functors with more than two arguments.

4.4. Symmetrization of Functors

Before stating the general existence theorem, we introduce the notion of *symmetrization* of a functor. We will define this notion for regular functors, but this concept can be easily generalized to enriched functors over any enriched category.

Symmetrization, introduced by Birkedal et al. [5], transforms a functor $F : (\mathcal{C}^{op} \times \mathcal{C})^n \to \mathcal{B}$ into a functor $\tilde{F} : (\mathcal{C}^{op} \times \mathcal{C})^n \to \mathcal{B}^{op} \times \mathcal{B}$ that it is defined as

$$\tilde{F}(\vec{X}^{op}, \vec{X}) = \langle F(\vec{X}, \vec{X}^{op}), F(\vec{X}^{op}, \vec{X}) \rangle,$$

where $\vec{X}$ represents a sequence of elements from $\mathcal{C}$, and $\vec{X}^{op}$ represents their counterparts from the opposite category $\mathcal{C}^{op}$.

Remark 4.12. *Note that we are making an abuse of notation in the definition of $\tilde{F}$. To define this functor rigorously, we would also require two additional functors: one that transforms a sequence of pairs of objects into a pair of sequences, and another that performs the reverse operation.*

Although it is conceptually simple, working with this definition directly can be a bit cumbersome, since we need to swap and transform the arguments of the functor. For example, think about the case were $F : (\mathcal{C}^{op} \times \mathcal{C})^{n+1} \to \mathcal{C}$ is contractive in the last argument. In this case, the symmetrization of F would also be contractive in the last argument, but, with this definition of $\tilde{F}$, specifically because of the abuse of notation, a complete proof of this property in a proof assistant would be much more complex than necessary.

For this reason, we introduce an intermediate construction that clarifies the process. This construction was presented in Gratzer [14], to clarify the symmetrization process when the functor has more than one argument. We will call this construction the *parametric symmetrization* of a functor.

Consider a functor $F : \mathcal{D} \times (\mathcal{C}^{op} \times \mathcal{C}) \to \mathcal{B}$. Without any additional structure on $\mathcal{D}$, it is only natural to define its (parametric) symmetrization, as a functor $\delta F : \mathcal{D}^{op} \times \mathcal{D} \times (\mathcal{C}^{op} \times \mathcal{C}) \to \mathcal{B}^{op} \times \mathcal{B}$, as follows:

$$\delta F(D_1, D_2, X, Y) = \langle F(D_1, Y, X), F(D_2, X, Y)\rangle.$$

Just by looking at this definition, it should be clear that the functor δF is contractive in the last argument if F is contractive in the last argument, since the definition does not require any abuse of notation that could hide this property.

When applied to the functor $F : (\mathcal{C}^{op} \times \mathcal{C})^{n+1} \to \mathcal{B}$, this construction does not give us the desired type for the symmetrization of F, but it is a step in the right direction.

To achieve the desired form $(\mathcal{C}^{op} \times \mathcal{C})^{n+1} \to \mathcal{B}^{op} \times \mathcal{B}$, we need to use that $\mathcal{D}$ is not just any category, but the category $(\mathcal{C}^{op} \times \mathcal{C})^{n}$. Leveraging this, we define the following functor:

$$\begin{aligned}\Delta : (\mathcal{C}^{op} \times \mathcal{C})^{n} &\to ((\mathcal{C}^{op} \times \mathcal{C})^{n})^{op} \times (\mathcal{C}^{op} \times \mathcal{C})^{n},\\ (\vec{X}^{op}, \vec{X}) &\mapsto ((\vec{X}, \vec{X}^{op}), (\vec{X}^{op}, \vec{X}))\end{aligned}$$

Note that we are making the same abuse of notation as before. However, the key difference is that we have hidden the complexity of the transformation inside the definition of Δ, while maintaining some properties of the functor in the definition of δF. These properties will be crucial to prove Theorem 4.16.

Now, we can define the symmetrization of a functor as follows:

Definition 4.13. *The* ***symmetrization*** *of a functor $F : (\mathcal{C}^{op} \times \mathcal{C})^{n} \to \mathcal{C}$ is the functor $\tilde{F} : (\mathcal{C}^{op} \times \mathcal{C})^{n} \to \mathcal{C}^{op} \times \mathcal{C}$, defined as follows:*

$$\tilde{F} = \begin{cases} \langle F, F\rangle & \text{if } n = 0,\\ \delta F \circ \langle \Delta, \mathsf{id}\rangle & \text{if } n > 0.\end{cases}$$

The definition presented in Birkedal et al. [5] is equivalent to this one, it is a matter of unfolding the definitions. However, having the intermediate construction δF will make the proof of Theorem 4.16 more straightforward.

4.5. General Existence Theorem

Finally, we can state the general existence theorem for recursive types in any contractively complete **iCOFE**-enriched category. This theorem is a generalization of the America-Rutten theorem presented in Section 3. Let us first revisit the general notion of contractive completness.

Definition 4.14. *A* ***iCOFE****-enriched category $\mathcal{C}$ is* ***contractively complete*** *if every locally contractive endofunctor $F : \mathcal{C} \to \mathcal{C}$ has a unique (up to isomorphism) fixed point, i.e., an object $\lim F$ such that $F(\lim F) \cong \lim F$.*

Example 4.15. *The following are examples of contractively complete* ***iCOFE****-enriched categories:*

1. *The self-enriched category of iCOFEs is contractively complete, as we have seen in Theorem 3.4.*
2. *The opposite category of an* ***iCOFE****-enriched category is also contractively complete.*

Now we can state the general existence theorem, which allows us to reduce the problem of finding fixed points to a simpler one, that is, one where the functor has one argument less.

Theorem 4.16. *Let $\mathcal{C}$ be an* ***iCOFE****-enriched category that is contractively complete, and let $F : (\mathcal{C}^{op} \times \mathcal{C})^{n+1} \to \mathcal{C}$ be locally contractive in the last argument. Then there exists a functor $\mathrm{Fix}F : (\mathcal{C}^{op} \times \mathcal{C})^{n} \to \mathcal{C}$ such that*

$$\forall A,\ F \circ \langle \mathrm{id}, \widetilde{\mathrm{Fix}F}\rangle\ A \cong \mathrm{Fix}F\ A.$$

Moreover, if F is locally contractive in all its arguments, then so is $\mathrm{Fix}F$.

We will provide a sketch to outline the main ideas behind the proof. In the case where $n = 0$ the functor F is of the form

$$F : 1 \times \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C},$$

and since F is contractive in the last argument, the functor $F(\mathrm{tt}, -) : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ is locally contractive. Therefore, by the America-Rutten theorem[1] (Theorem 3.7), there exists a fixed point of $F(\mathrm{tt}, -)$, which we will denote by X. Therefore, the functor $\mathrm{Fix}F$ will be the one that maps the unique object of $(\mathcal{C}^{op} \times \mathcal{C})^{0} = 1$ to X. Now, clearly, for any object $\mathrm{tt} \in 1$,

$$F \circ \langle \mathrm{id}, \widetilde{\mathrm{Fix}F}\rangle\ \mathrm{tt} = F(\mathrm{tt}, (X, X)) \cong X = \mathrm{Fix}F\ \mathrm{tt}.$$

[1] Note that in the proof of the America-Rutten theorem, we used only the fact that the category of iCOFEs is contractively complete and no additional structure on the category of iCOFEs. This is why the proof of this theorem is essentially the same.

The case $n > 0$ is more intricate. The first observation[2] is that for any $A, B \in (\mathcal{C}^{op} \times \mathcal{C})^n$, there exist $X_{A,B}, Y_{A,B} \in \mathcal{C}$ such that

$$F(A, (X_{A,B}, Y_{A,B})) \cong Y_{A,B}, \text{ and}$$
$$F(B, (Y_{A,B}, X_{A,B})) \cong X_{A,B}.$$

Now, consider the parametric symmetrization of the functor F:

$$\delta F : \mathcal{D}^{op} \times \mathcal{D} \times (\mathcal{C}^{op} \times \mathcal{C}) \to \mathcal{C}^{op} \times \mathcal{C},$$

where $\mathcal{D} = (\mathcal{C}^{op} \times \mathcal{C})^n$.

This functor is contractive in the last argument because F is contractive in the last argument. Moreover, for any $(A, B) \in \mathcal{D}^{op} \times \mathcal{D}$, the pair $(Y_{A,B}, X_{A,B})$ forms an initial algebra for $\delta F(A, B, -)$, as it is a fixed point of the contractive enriched endofunctor $\delta F(A, B, -)$:

$$\begin{aligned}
\delta F(A, B, -)(Y_{A,B}, X_{A,B}) \\
&= \langle F(A, (X_{A,B}, Y_{A,B})), F(B, (Y_{A,B}, X_{A,B}))\rangle \\
&\cong (Y_{A,B}, X_{A,B}).
\end{aligned}$$

Using this, we define the functor $\mu\delta F : \mathcal{D}^{op} \times \mathcal{D} \to \mathcal{C}^{op} \times \mathcal{C}$, which assigns to each object (A, B) the fixed point of $\delta F(A, B, -)$, i.e., $(Y_{A,B}, X_{A,B})$, and to each morphism $f : (A, B) \to (C, D)$, the unique morphism $\mu\delta F\ f$ that satisfies the following commutative diagram:

$$\begin{tikzcd}
\delta F(A, B, \mu\delta F\,(A, B)) \arrow[r, "\cong"] \arrow[d, "\delta F(f, \mu\delta F f)"] & \mu\delta F\,(A, B) \arrow[d, "\mu\delta F f"] \\
\delta F(C, D, \mu\delta F\,(C, D)) \arrow[r, "\cong"] & \mu\delta F\,(C, D)
\end{tikzcd}$$

Next, we compose the functor $\mu\delta F$ with the functor Δ defined in Section 4.4 and the projection functor onto the second component, obtaining the functor

$$\mathrm{Fix}F = \pi_2 \circ \mu\delta F \circ \Delta : (\mathcal{C}^{op} \times \mathcal{C})^n \to \mathcal{C}.$$

Lastly, we need show that this functor satisfies the desired property: for every object $A \in (\mathcal{C}^{op} \times \mathcal{C})^n$,

$$(F \circ \langle \mathrm{id}, \widetilde{\mathrm{Fix}F}\rangle)\ A \cong \mathrm{Fix}F\ A.$$

This is a matter of unfolding the definitions and using the properties of the functors involved. The details of this part of the proof can be found in Gratzer [14], Theorem 7.

Corollary 4.17. *Let $\mathcal{C}$ be a contractively complete* ***iCOFE****-enriched category, and let $F : (\mathcal{C}^{op} \times \mathcal{C})^{n+1} \to \mathcal{C}$ be a locally contractive functor. Then there exists a unique (up to isomorphism)* $\lim F \in \mathcal{C}$ *such that*

$$F\big((\lim F, \lim F), \ldots, (\lim F, \lim F)\big) \cong \lim F.$$

To prove this corollary, one could recursively apply Theorem 4.16 to find the fixed point of the functor F and then show that this fixed point is unique up to isomorphism. However, there is a more direct approach that relies solely on the America-Rutten theorem.

First, define the functor $J_n : \mathcal{C}^{op} \times \mathcal{C} \to (\mathcal{C}^{op} \times \mathcal{C})^n$, which maps each object $X \in \mathcal{C}^{op} \times \mathcal{C}$ to the sequence $(X, \ldots, X)$. Using this definition, we can establish the following equality for every object $X \in \mathcal{C}^{op} \times \mathcal{C}$:

$$F(X, \ldots, X) = (F \circ J_n)(X).$$

Since F is locally contractive, the composite $F \circ J_n$ is also locally contractive. By the America-Rutten theorem, this implies that $F \circ J_n$ has a unique fixed point, which corresponds to the desired fixed point of F.

Our main example of a contractively complete **iCOFE**-enriched category is the self-enriched category of iCOFEs, as shown in Theorem 3.4. This corollary allows us to prove the existence and uniqueness of recursive types, using the category of iCOFEs as our model of guarded recursion, for any type constructor that has both positive and negative occurrences of the various type arguments.

5. Coq Formalization

This section describes the methodology and design choices for formalizing the results presented in this paper using the Coq proof assistant. The formalization was developed in a modular fashion, with each concept encapsulated in separate files or folders. The complete Coq development is publicly available at [9].

The formalization is organized into the following components:

- **category-theory**: This module contains the basic concepts of category theory, including categories, functors, and natural transformations. These elements form the foundation of our formalization.
- **ofe**: This module focuses on OFEs and related concepts, encompassing the material discussed in Section 2.
- **ecategory-theory**: This module extends the category-theory formalization to handle **iCOFE**-enriched categories, associated functors, and isomorphisms.
- **solver**: This module contains the formalization of the general existence theorem for recursive types, as discussed in Section 4, and all the concepts of section 3 but generalized to the enriched setting.

The formalization relies on the *Proof Irrelevance* axiom and *Functional Extensionality*. These axioms simplify the treatment of equality for records and functions, offering a pragmatic alternative to the use of setoids and equivalence relations for quotient types. This approach facilitates concise proofs and straightforward reasoning within the Coq framework.

To enhance the performance and modularity of our formalization, we adopted an approach inspired by

[2]See Lemma 6 in Gratzer [14] for more details on the proof of this fact.

the Structure-Mixin methodology introduced by Garillot et al. [12] and popularized by the Mathematical Components library [8]. In this methodology, a mathematical structure typically consists of a representation type (often just one), constants and operations on the type(s), and axioms that these operations must satisfy. While these components can be combined into a single record, doing so often leads to an unwieldy proliferation of arguments, making theorem proving cumbersome.

The Structure-Mixin methodology addresses this by separating the representation type and operations from the axioms, which are encapsulated into a separate dependent record called a mixin. However, unlike the original approach, which relies on Canonical Structures, our implementation uses regular records to split proof terms (axioms) from the data (representation and operations). This adaptation simplifies the formalization while retaining the modularity and clarity benefits of the Structure-Mixin approach.

5.1. Category Theory

The core concepts of categories, functors, natural transformations, and isomorphisms are formalized as Coq records using the previously described methodology. Furthermore, we extended the formalization to include categories with terminal objects, finite products, and cartesian closed categories, utilizing Coq's class mechanism, introduced by Sozeau and Oury [26], to manage these extensions modularly.

5.2. Ordered Families of Equivalences (OFEs)

The formalization of OFEs builds on the work of Jung et al. [17]. In Coq, OFEs are defined as records, using the previously described methodology, and each equivalence relation is expressed using a Coq relation of the Equivalence class.

To define non-expansive functions, we adopted the well-established approach of leveraging the Proper class, as described by Sozeau [25], which enables the use of the setoid_rewrite tactic. This method significantly simplifies proofs involving n-equalities. For instance, consider the following proof scenario:

```
 ...
H1 : z ≡{n}≡ f y
H2 : x ≡{n}≡ y
======================= (1 / 1)
f x ≡{n}≡ z.
```

Although the proof is conceptually simple, without the Proper class, the proof script would require manually invoking the transitivity, symmetry, and the non-expansiveness predicate for f. By contrast, the Proper class reduces the proof to straightforward goal rewriting using the provided hypotheses.

For contractive functions, we followed a similar approach, defining contractive leveraging the Proper class. This design choice ensures that the resulting proofs are more concise and maintainable.

The remainder of the module does not involve any noteworthy design decisions or choices that merit further discussion.

5.3. Enriched Category Theory

The formalization of **iCOFE**-enriched categories extends the foundational category-theory module to enriched settings. We defined a shallow embedding of the monoidal structure of the **iCOFE** category in Coq. This embedding captures the essential properties of the tensor functor, unit object, and associativity and unit laws, enabling the representation of **iCOFE**-enriched structures.

Building on this monoidal structure, we formalized **iCOFE**-enriched categories, enriched functors, enriched natural transformations, and enriched isomorphisms. These concepts are represented as Coq records, maintaining consistency with the style adopted in the category-theory module.

This enriched framework is instrumental in formalizing the existence of recursive types and proving key results about their structure. Notably, the definition of locally contractive functors is made more straightforward and general through the enriched category-theory formalization.

5.4. Solver

The solver module is the most complex part of the formalization, as it incorporates the general existence theorem for recursive types. This module builds on the enriched category-theory module, introducing the necessary definitions and lemmas to prove the theorem.

The design of this module is similar to the previous ones, and there are no significant deviations from the methodology described earlier.

6. Related Work

6.1. Recursive Domain Equations in Coq

Sieczkowski et al. [23] introduced the ModuRes Coq library, a flexible framework for solving recursive domain equations in Coq. Like our approach, ModuRes is based on OFEs and categories enriched over COFEs, and it directly implements the America-Rutten theorem. However, a key distinction lies in the treatment of inhabitedness: while our work introduces a dedicated structure for iCOFEs, ModuRes incorporates the inhabitedness assumption as an additional requirement for its results. This difference is primarily a design choice, and the two approaches are equivalent in terms of expressiveness.

In contrast, our formalization specifically targets iCOFEs because they simplify the definition of fixed points. Additionally, our approach is more modular, drawing from the work of Birkedal et al. [5]. This enables us to prove a simpler version of the theorem and establish the America-Rutten theorem as a corollary (Theorem 3.7). Furthermore, our formalization extends the results to functors with an arbitrary number of arguments, providing a broader framework for reasoning about recursive types. This is a significant advantage over the ModuRes library, which is limited to the case of a single argument with mixed variance.

The Iris library by Jung et al. [17] also provides a proof of the America-Rutten theorem. However, the Iris formalization is more specialized and tailored compared to the ModuRes library (and our work), as it is specifically designed for a particular application. As a result, it incorporates numerous definitions of functors that complicate the generalization of results to more complex settings. In contrast, our formalization is designed to handle more intricate scenarios, and the general nature of our design allows us, for instance, to define a single, unified functor, resulting in a cleaner and more modular design overall.

6.1.1 Category Theory in Coq

Category theory formalizations in Coq are well-established, with several existing libraries, including:

- *UniMath* [15], which adopts a univalent approach,
- The *Categories* library by Timany [27], and
- The *Category Theory* library by Wiegley [28]

Despite the availability of these libraries, we developed a tailored formalization to better align with our specific requirements. Many existing libraries include features and abstractions that were unnecessary for our work, such as univalence, advanced category-theory concepts that affected the interface complexity of the library, and even polymorphism which was not necessary for our work. Instead, we prioritized simplicity and direct applicability to the problems at hand.

7. Future Work

Moving forward, it would be interesting to explore the application of our work to the Iris logic (Jung et al. [17]), which already includes a proof of the America-Rutten theorem. It could be beneficial to integrate results such as the Theorem 4.16 into the library to, for example, handle a broader range of recursive types. While doing so, we could significantly simplify the library we developed. For example, we could eliminate the entire category-theory module and begin directly with the enriched category module. Additionally, we could streamline the record definition for **iCOFE**-enriched categories, as the full categorical structure would no longer be necessary.

Another avenue for future research could be the formalization of the *topos of trees*, as described by Birkedal et al. [5]. This topos is far more general than the category of iCOFEs and could provide a more flexible framework for reasoning about recursive types. However, this formalization would require more advanced category-theory concepts and it would involve many polymorphic definitions, which could be challenging to formalize in Coq. In his internship report, Kocsis [19] has already started to formalize the topos of trees in Coq as a first step towards this goal, but the actual formalization of the general existence theorem for recursive types in the topos of trees is still pending.

8. Acknowledgments

The Iris Mattermost channel played a key role in the progress of this work, providing a platform for asking questions and exchanging ideas with fellow researchers in the field. We especially want to thank Amin Timany and Lars Birkedal for their valuable insights and assistance.

Finally, this work would not have been possible without the weekly meetings with Robbert Krebbers, whose guidance throughout the development of this formalization, along with his insightful suggestions for improving the quality of the report and the formalization itself, was essential to the success of this project. His expertise and dedication were invaluable in overcoming the challenges we encountered.

References

[1] Pierre America and Jan J. M. M. Rutten. "Solving Reflexive Domain Equations in a Category of Complete Metric Spaces". In: *J. Comput. Syst. Sci.* 39.3 (1989), pages 343–375. DOI: `10.1016/0022-0000(89)90027-5`. URL: `https://doi.org/10.1016/0022-0000(89)90027-5` (cited on page 5).

[2] Andrew W. Appel and David A. McAllester. "An indexed model of recursive types for foundational proof-carrying code". In: *ACM Trans. Program. Lang. Syst.* 23.5 (2001), pages 657–683. DOI: `10.1145/504709.504712`. URL: `https://doi.org/10.1145/504709.504712` (cited on page 1).

[3] Andrew W. Appel et al. "A very modal model of a modern, major, general type system". In: *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, Nice, France, January 17-19, 2007*. Edited by Martin Hofmann and Matthias Felleisen. ACM, 2007, pages 109–122. DOI: `10.1145/1190216.1190235`. URL: `https://doi.org/10.1145/1190216.1190235` (cited on page 1).

[4] Lars Birkedal and Aleš Bizjak. "A Taste of Categorical Logic — tutorial notes". In: (2014) (cited on page 3).

[5] Lars Birkedal et al. "First steps in synthetic guarded domain theory: step-indexing in the topos of trees". In: *Log. Methods Comput. Sci.* 8.4 (2012). DOI: `10.2168/LMCS-8(4:1)2012`. URL: `https://doi.org/10.2168/LMCS-8(4:1)2012` (cited on pages 1, 2, 6, 8, 9, 11, 12).

[6] Lars Birkedal et al. "Step-indexed kripke models over recursive worlds". In: *SIGPLAN Not.* 46.1 (Jan. 2011), pages 119–132. ISSN: 0362-1340. DOI: 10.1145/1925844.1926401. URL: https://doi.org/10.1145/1925844.1926401 (cited on page 1).

[7] Francis Borceux. *Handbook of categorical algebra: volume 1, Basic category theory*. Volume 1. Cambridge University Press, 1994 (cited on page 3).

[8] Cyril Cohen et al. *math-comp/math-comp: The Mathematical Components Library 2.3.0*. Version mathcomp-2.3.0. Nov. 2024. DOI: 10.5281/zenodo.14237175. URL: https://doi.org/10.5281/zenodo.14237175 (cited on page 11).

[9] Sergio Domínguez. *sergiodguezc/coq-multivariate-rde-solver: v1.0.0*. Version v1.0.0. Feb. 2025. DOI: 10.5281/zenodo.14879646. URL: https://doi.org/10.5281/zenodo.14879646 (cited on pages 2, 10).

[10] P. Freyd. "Recursive types reduced to inductive types". In: *[1990] Proceedings. Fifth Annual IEEE Symposium on Logic in Computer Science*. 1990, pages 498–507. DOI: 10.1109/LICS.1990.113772 (cited on page 7).

[11] Peter Freyd. "Algebraically complete categories". In: *Category Theory*. Edited by Aurelio Carboni, Maria Cristina Pedicchio, and Guiseppe Rosolini. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pages 95–104. ISBN: 978-3-540-46435-8 (cited on page 7).

[12] François Garillot et al. "Packaging Mathematical Structures". In: *Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings*. Edited by Stefan Berghofer et al. Volume 5674. Lecture Notes in Computer Science. Springer, 2009, pages 327–342. DOI: 10.1007/978-3-642-03359-9_23. URL: https://doi.org/10.1007/978-3-642-03359-9%5C_23 (cited on page 11).

[13] Pietro Di Gianantonio and Marino Miculan. "A Unifying Approach to Recursive and Corecursive Definitions". In: *Types for Proofs and Programs, Second International Workshop, TYPES 2002, Berg en Dal, The Netherlands, April 24-28, 2002, Selected Papers*. Edited by Herman Geuvers and Freek Wiedijk. Volume 2646. Lecture Notes in Computer Science. Springer, 2002, pages 148–161. DOI: 10.1007/3-540-39185-1_9. URL: https://doi.org/10.1007/3-540-39185-1%5C_9 (cited on page 2).

[14] Daniel Gratzer. "An Addendum to Section 7 of first steps in synthetic guarded domain theory". In: (2019) (cited on pages 2, 9, 10).

[15] Daniel R. Grayson et al. *UniMath/UniMath: UniMath 2024-09-23*. Version v20240923. Sept. 2024. DOI: 10.5281/zenodo.13828995. URL: https://doi.org/10.5281/zenodo.13828995 (cited on page 12).

[16] Jonas Kastberg Hinrichsen, Jesper Bengtson, and Robbert Krebbers. "Actris: session-type based reasoning in separation logic". In: *Proc. ACM Program. Lang.* 4.POPL (Dec. 2019). DOI: 10.1145/3371074. URL: https://doi.org/10.1145/3371074 (cited on page 2).

[17] Ralf Jung et al. "Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning". In: *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*. Edited by Sriram K. Rajamani and David Walker. ACM, 2015, pages 637–650. DOI: 10.1145/2676726.2676980. URL: https://doi.org/10.1145/2676726.2676980 (cited on pages 1, 2, 6, 11, 12).

[18] G.M Kelly. "Tensor products in categories". In: *Journal of Algebra* 2.1 (1965), pages 15–37. ISSN: 0021-8693. DOI: https://doi.org/10.1016/0021-8693(65)90022-0. URL: https://www.sciencedirect.com/science/article/pii/0021869365900220 (cited on page 7).

[19] Bálint Kocsis. "Guarded recursion in the topos of trees : Internship report". In: (2023). URL: https://github.com/balint99/gtt_topos_of_trees (cited on page 12).

[20] F William Lawvere. "Adjointness in foundations". In: *Dialectica* (1969), pages 281–296 (cited on page 3).

[21] Saunders MacLane. "Categorical algebra". In: *Bulletin of the American Mathematical Society* 71.1 (1965), pages 40–106 (cited on page 7).

[22] Hiroshi Nakano. "A Modality for Recursion". In: *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*. IEEE Computer Society, 2000, pages 255–266. DOI: 10.1109/LICS.2000.855774. URL: https://doi.org/10.1109/LICS.2000.855774 (cited on page 1).

[23] Filip Sieczkowski, Ales Bizjak, and Lars Birkedal. "ModuRes: A Coq Library for Modular Reasoning About Concurrent Higher-Order Imperative Programming Languages". In: *Interactive Theorem Proving - 6th International Conference, ITP 2015, Nanjing, China, August 24-27, 2015, Proceedings*. Edited by Christian Urban and Xingyuan Zhang. Volume 9236. Lecture Notes in Computer Science. Springer, 2015, pages 375–390. DOI: 10.1007/978-3-319-22102-1_25. URL: https://doi.org/10.1007/978-3-319-22102-1%5C_25 (cited on pages 1, 2, 5, 11).

[24] Michael B. Smyth and Gordon D. Plotkin. "The Category-Theoretic Solution of Recursive Domain Equations". In: *SIAM J. Comput.* 11.4 (1982), pages 761–783. DOI: 10.1137/0211062. URL: https://doi.org/10.1137/0211062 (cited on page 1).

[25] Matthieu Sozeau. "A New Look at Generalized Rewriting in Type Theory". In: *J. Formaliz. Reason.* 2.1 (2009), pages 41–62. DOI: 10.6092/ISSN.1972-5787/1574. URL: https://doi.org/10.6092/issn.1972-5787/1574 (cited on page 11).

[26] Matthieu Sozeau and Nicolas Oury. "First-Class Type Classes". In: *Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August 18-21, 2008. Proceedings*. Edited by Otmane Ait Mohamed, César A. Muñoz, and Sofiène Tahar. Volume 5170. Lecture Notes in Computer Science. Springer, 2008, pages 278–293. DOI: 10.1007/978-3-540-71067-7_23. URL: https://doi.org/10.1007/978-3-540-71067-7%5C_23 (cited on page 11).

[27] Amin Timany. *Categories: FSCD 2016*. Version FSCD16. Apr. 2016. DOI: 10.5281/zenodo.50689. URL: https://doi.org/10.5281/zenodo.50689 (cited on page 12).

[28] John Wiegley. *jwiegley/category-theory: 1.0.0*. Version 1.0.0. Aug. 2022. URL: https://github.com/jwiegley/category-theory (cited on page 12).