Logical Methods in Computer Science
Volume 15, Issue 3, 2019, pp. 1:1–1:47
https://lmcs.episciences.org/

Submitted Dec. 29, 2017
Published Jul. 04, 2019

GUARDED AND UNGUARDED ITERATION FOR GENERALIZED PROCESSES

SERGEY GONCHAROV [a], LUTZ SCHRÖDER [b], CHRISTOPH RAUCH [c], AND MACIEJ PIRÓG [d]

[a,b,c] Friedrich-Alexander-Universität Erlangen-Nürnberg
e-mail address: {sergey.goncharov,lutz.schroeder,christoph.rauch}@fau.de

[d] Uniwersytet Wrocławski
e-mail address: maciej.pirog@cs.uni.wroc.pl

Abstract. Models of iterated computation, such as (completely) iterative monads, often depend on a notion of guardedness, which guarantees unique solvability of recursive equations and requires roughly that recursive calls happen only under certain guarding operations. On the other hand, many models of iteration do admit unguarded iteration. Solutions are then no longer unique, and in general not even determined as least or greatest fixpoints, being instead governed by quasi-equational axioms. Monads that support unguarded iteration in this sense are called (complete) Elgot monads. Here, we propose to equip (Kleisli categories of) monads with an abstract notion of guardedness and then require solvability of abstractly guarded recursive equations; examples of such *abstractly guarded pre-iterative monads* include both iterative monads and Elgot monads, the latter by deeming any recursive definition to be abstractly guarded. Our main result is then that Elgot monads are precisely the iteration-congruent retracts of abstractly guarded *iterative* monads, the latter being defined as admitting *unique* solutions of abstractly guarded recursive equations; in other words, models of unguarded iteration come about by quotienting models of guarded iteration.

1. Introduction

In recursion theory, notions of guardedness traditionally play a central role. Guardedness typically means that recursive calls must be in the scope of certain guarding operations, a condition aimed, among other things, at ensuring progress. The paradigmatic case are recursive definitions in process algebra, which are usually called guarded if recursive calls occur only under action prefixing [5]. A more abstract example are completely iterative theories [11] and monads [23], where, in the latter setting, a recursive definition is guarded

Key words and phrases: Monads, iteration, guarded fixpoints, side-effects.
Research supported by the DFG project *A High Level Language for Monad-based Processes* (GO 2161/1-2, SCHR 1118/8-2).

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.23638/LMCS-15(3:1)2019

if it factors through a given ideal of the monad. Guarded recursive definitions typically have unique solutions; e.g. the unique solution of the guarded recursive definition

$$x = a.\, x$$

is the process that keeps performing the action a.

For unguarded recursive definitions, the picture is, of course, different. For example, to obtain the denotational semantics of an unproductive while loop $\mathsf{while\ true\ do\ skip}$ characterized by circular operational behavior

$$\mathsf{while\ true\ do\ skip} \quad\to\quad \mathsf{skip; while\ true\ do\ skip} \quad\to\quad \mathsf{while\ true\ do\ skip}$$

one will select one of many solutions of this trivial equation, e.g. the least solution in a domain-theoretic semantics.

Sometimes, however, one has a selection among non-unique solutions of unguarded recursive equations that is *not* determined order-theoretically, i.e. by picking least or greatest fixpoints. One example arises from *coinductive resumptions* [16, 31, 30]. In the paradigm of monad-based encapsulation of side-effects [26], coinductive resumptions over a base effect encapsulated by a monad T form a monad T^ν, the *coinductive resumption transform*, given by

$$T^\nu X = \nu\gamma.\, T(X + \gamma) \tag{1.1}$$

– that is, a computation over X performs a step with effects from T, and then returns either a value from X or a resumption that, when resumed, proceeds similarly, possibly ad infinitum. We thus can view coinductive resumptions as processes whose atomic steps are programs over T. We generally restrict to monads T for which (1.1) exists for all X (although many of our results do not depend on this assumption). Functors (or monads) T for which this holds are called *iteratable* [1]. Most computationally relevant monads are iteratable (notable exceptions in the category of sets are the powerset monad and the continuation monad). The last occurrence of γ in (1.1) may be seen as being wrapped in an implicit unary *delay* operation that represents the gap between returning a resumption and resuming it. One thus has a natural *delay map* $T^\nu X \to T^\nu X$ that converts a computation into a resumption, i.e. prefixes it with a delay step. In fact, for $T = \mathsf{id}$, T^ν is precisely Capretta's *partiality monad* [7], also called the *delay monad*. It is not in general possible to equip $T^\nu X$ with an ordered domain structure that would allow for selecting least (or greatest) solutions of unguarded recursive definitions over T^ν. However, one *can* select solutions in a coherent way, that is, such that a range of natural quasi-equational axioms is satisfied, making T^ν into a (complete) *Elgot monad* [2, 18] whenever T is so.

More precisely, we closely follow the perspective advanced by Bloom and Esik [6, 13], who identify as *iteration operators* certain categorical operators with the profile $(f\colon X \to Y + X) \mapsto (f^\dagger\colon X \to Y)$ (which are categorical duals of *parametrized recursion operators* $(f\colon Y \times X \to X) \mapsto (f_\dagger\colon Y \to X)$ [32]). The above-mentioned Elgot monads support iteration operators in this sense, specifically as operators on their Kleisli categories (with coproduct + inherited from the base category). We place total (unguarded) iteration and partial (guarded) iteration on the same footing and thus aim to unify the theories of guarded and unguarded iteration. To this end, we introduce a notion of *abstractly guarded monads*, that is, monads equipped with a distinguished class of *abstractly guarded* equation morphisms satisfying natural closure properties (Section 3). The notion of abstract guardedness can be instantiated in various ways, e.g. with the class of immediately terminating 'recursive' definitions, with the class of guarded morphisms in a completely iterative monad, or with the class of all equation morphisms. We call an abstractly guarded monad *pre-iterative* if all abstractly guarded equation morphisms have a solution, and *iterative* if these solutions are unique. Then completely iterative monads are iterative abstractly guarded in this sense, and (complete) Elgot monads are pre-iterative, where we deem every equation morphism to be abstractly guarded in the latter case.

The quasi-equational axioms of Elgot monads are easily seen to be satisfied when fixpoints are unique, i.e. in iterative abstractly guarded monads, and moreover stable under iteration-congruent retractions in a fairly obvious sense. Our first main result (Section 5, Theorem 5.7) states that the converse holds as well, i.e. *a monad T is a complete Elgot monad iff T is an iteration-congruent retract of an iterative abstractly guarded monad* – specifically of T^ν as in (1.1). As a slogan,

> *monad-based models of unguarded iteration arise by quotienting models of guarded iteration.*

Our second main result (Theorem 5.15) is an algebraic characterization of complete Elgot monads: We show that the construction $(-)^\nu$ mapping a monad T to T^ν as in (1.1) is a monad on the category of monads (modulo existence of T^ν), and *complete Elgot monads are precisely those $(-)^\nu$-algebras T that cancel the delay map on T^ν*, i.e. interpret the delay operation as identity.

As an illustration of these results we discuss various semantic domains of processes equipped with canonical solutions of systems of process definitions under various notions of guardedness (Example 4.5) and show how these domains can be related via iteration-preserving morphisms implementing a suitable coarsening of the underlying equivalence relation, e.g. from bisimilarity to finite trace equivalence (Example 5.8). Moreover, we show (Section 6) that sandwiching a complete Elgot monad between a pair of adjoint functors again yields a complete Elgot monad, in analogy to a corresponding result for completely iterative monads [31]. Specifically, we prove a sandwich theorem for iterative abstractly guarded monads and transfer it to complete Elgot monads using our first main result. For illustration, we then relate iteration in ultrametric spaces using Escardó's metric lifting monad [12] to iteration in pointed cpo's, by noting that the corresponding monads on sets obtained using our sandwich theorems are related by an iteration-congruent retraction in the sense of our first main result.

The material is organized as follows. We discuss preliminaries on monads and their Kleisli categories and on coalgebras in Section 2. Our notion of abstractly guarded monad, derived from a notion of guarded co-Cartesian category, is presented in Section 3, and extended to parametrized monads in the sense of Uustalu [35] in Section 4. We prove our main results on the relationship between Elgot monads and guarded iteration as discussed above in Section 5, and present the mentioned application to sandwiching in Section 6. We discuss related work in Section 7; Section 8 concludes. The present paper extends an earlier conference version [19] by full proofs and additional example material, mostly within Examples 4.5 and 5.8.

2. Preliminaries

We work in a category $\mathbf{C}$ with finite coproducts (including an initial object $\emptyset$) throughout. A pair $\sigma = \langle \sigma_1 \colon Y_1 \to X, \sigma_2 \colon Y_2 \to X \rangle$ of morphisms is a *summand* of X, denoted $\sigma \colon Y_1 \hookrightarrow X$, if it forms a coproduct cospan, i.e. X is a coproduct of Y_1 and Y_2 with σ_1 and σ_2 as coproduct injections. Each summand $\sigma = \langle \sigma_1, \sigma_2 \rangle$ thus determines a *complement summand* $\bar\sigma = \langle \sigma_2, \sigma_1 \rangle \colon Y_2 \hookrightarrow X$. We often shorten a summand $\langle \sigma_1, \sigma_2 \rangle$ to its first component σ_1, in order to use σ as a morphism $Y_1 \to X$. Summands of a given object X are naturally preordered by taking $\langle\sigma_1, \sigma_2\rangle$ to be smaller than $\langle\theta_1, \theta_2\rangle$ if σ_1 factors through θ_1 and θ_2 factors through σ_2. This preorder has a greatest element $\langle \mathsf{id}_X, !\rangle$ and a least element $\langle !, \mathsf{id}_X\rangle$. By writing $X+Y$ we designate the latter as a coproduct of X and Y and assign the canonical names $\mathsf{in}_1\colon X \hookrightarrow X+Y$ and $\mathsf{in}_2\colon Y \hookrightarrow X+Y$ to the corresponding summands. Dually, we write $\mathsf{pr}_1\colon X\times Y \to X$ and $\mathsf{pr}_2\colon X\times Y\to Y$ for canonical *projections* (without introducing a special arrow notation). We do not assume that $\mathbf{C}$ is *extensive* [8], in which case coproduct complements would be uniquely determined.

A *monad* $\mathbb{T}$ over $\mathbf{C}$ can be given in the form of a *Kleisli triple* $(T, \eta, \text{--}^\star)$ where T is an endomap over the objects $|\mathbf{C}|$ of $\mathbf{C}$, the *unit* η is a family of morphisms $(\eta_X\colon X \to TX)_{X\in|\mathbf{C}|}$, *Kleisli lifting* $(\text{--})^\star$ is a family of maps : $\mathsf{Hom}(X, TY) \to \mathsf{Hom}(TX, TY)$, and the *monad laws* are satisfied:

$$\eta^\star = \mathsf{id}, \qquad\qquad f^\star\,\eta = f, \qquad\qquad (f^\star\,g)^\star = f^\star\,g^\star.$$

These laws precisely ensure that taking morphisms of the form $X \to TY$ under $f^\star g$ as the composition and η as identities yields a category, which is also called the *Kleisli category* of $\mathbb{T}$, and denoted $\mathbf{C}_\mathbb{T}$. The standard (equivalent) categorical definition [22] of $\mathbb{T}$ as an endofunctor with natural transformation *unit* $\eta\colon \mathrm{Id} \to T$ and *multiplication* $\mu\colon TT\to T$ can be recovered by taking $Tf = (\eta\, f)^\star$, $\mu = \mathsf{id}^\star$. (We adopt the convention that monads and their functor parts are denoted by the same letter, with the former in blackboard bold.) We call morphisms $X\to TY$ *Kleisli morphisms* and view them as a high level abstraction of sequential programs where $\mathbb{T}$ encapsulates the underlying computational effect as proposed by Moggi [27], with X representing the input type and Y the output type. The Kleisli category inherits coproducts from $\mathbf{C}$, i.e. a coproduct $X+Y$ of objects X, Y in $\mathbf{C}$ remains a coproduct in $\mathbf{C}_\mathbb{T}$, with coproduct injections $\eta\,\mathsf{in}_1$ and $\eta\,\mathsf{in}_2$.

A more traditional use of monads in semantics is due to Lawvere [21], who identified finitary monads on **Set** with *algebraic theories*, hence objects TX can be viewed as sets of terms of the theory over free variables from X, the unit as the operation of casting a variable to a term, and Kleisli composition as substitution. We informally refer to this use of monads as *algebraic monads*. Regardless of this informal convention, for every monad $\mathbb{T}$ we have an associated category of *(Eilenberg-Moore-)algebras* $\mathbf{C}^\mathbb{T}$ whose objects are pairs $(A, a\colon TA\to A)$ satisfying $a\,\eta = \mathsf{id}$ and $\mu\,(Ta) = a\,(Ta)$ and whose morphisms from $(A, a\colon TA\to A)$ to $(B, b\colon TB\to B)$ are maps $f\colon A\to B$ such that $f\,a = b\,(Tf)$.

Given an adjunction $F \dashv G\colon \mathbf{D}\to\mathbf{C}$, we obtain a monad whose functor part is the composite $Gf\colon \mathbf{C}\to\mathbf{C}$, and both the Eilenberg-Moore construction and the Kleisli construction show that every monad has this form. In consequence, we can *sandwich* a monad $\mathbb{T}$ on $\mathbf{D}$ between an adjunction $F\dashv G\colon \mathbf{D}\to\mathbf{C}$, obtaining a monad on $\mathbf{C}$ with functor part GTF.

A(n *F-)coalgebra* for an endofunctor $f\colon \mathbf{C}\to\mathbf{C}$ is a pair $(X, f\colon X\to FX)$ where $X\in|\mathbf{C}|$. Coalgebras form a category, with morphisms $(X,f)\to(Y,g)$ being $\mathbf{C}$-morphisms $h\colon X\to Y$ such that $(Fh)f = g\,h$. A final object of this category is called a *final coalgebra*, and we denote it by

$$(\nu F, \mathsf{out}\colon \nu F\to F\nu F)$$

if it exists. For readability,

> *we will often be cavalier about existence of final coalgebras and silently assume they exist when we need them;*

that is, we hide sanity conditions on the involved functors, such as accessibility (we make an exception to this in parts of Section 5 where we characterize Elgot monads as certain Eilenberg-Moore algebras for a monad on the category of monads). By definition, νF comes with *coiteration* as a definition principle (dual to the iteration principle for algebras): given a coalgebra $(X, f\colon X \to FX)$ there is a unique morphism $(\mathsf{coit}\, f)\colon X \to \nu F$ such that

$$\mathsf{out}\,(\mathsf{coit}\, f) = F(\mathsf{coit}\, f)\, f.$$

This implies that out is an isomorphism (*Lambek's lemma*) and that $\mathsf{coit}\,\mathsf{out} = \mathsf{id}$ (see [36] for more details about coalgebras for coiteration). The category of F-*algebras*, F-algebra morphisms and the notion of *initial* F-*algebra* $(\mu F, \mathsf{in}\colon F\mu F \to \mu F)$ are obtained in a completely dual way. The characteristic properties of final coalgebras and initial algebras can be summarized in the following diagrams:

$$\begin{array}{ccc} F\mu F & \xrightarrow{F(\mathsf{iter}\, f)} & FX \\ {\scriptstyle \mathsf{in}}\downarrow & & \downarrow{\scriptstyle f} \\ \mu F & \xrightarrow[\mathsf{iter}\, f]{} & X \end{array} \qquad\qquad \begin{array}{ccc} X & \xrightarrow{\mathsf{coit}\, f} & \nu F \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \mathsf{out}} \\ FX & \xrightarrow[F(\mathsf{coit}\, f)]{} & F\nu F \end{array}$$

Note that F-algebras should not be confused with Eilenberg-Moore algebras of monads (as we indicated above, those satisfy additional laws).

We generally drop sub- and superscripts, e.g. on natural transformations, whenever this improves readability.

3. Abstractly Guarded Categories and Monads

The notion of guardedness is paramount in process algebra: typically one considers systems of mutually recursive process definitions of the form $x_i = t_i$, and a variable x_i is said to be guarded in t_j if it occurs in t_j only in subterms of the form $a.\, s$ where $a.\,(-)$ is action prefixing. A standard categorical approach is to replace the set of terms over variables X by an object TX where $\mathbb{T}$ is a monad. We then can model separate variables by partitioning X into a sum $X_1 + \ldots + X_n$ and thus talk about guardedness of a morphism $f\colon X \to T(X_1 + \ldots + X_n)$ in any X_i, meaning that every variable from X_i is guarded in f. One way to capture guardedness categorically is to identify the operations of $\mathbb{T}$ that serve as guards by distinguishing a suitable subobject of TX; e.g. the definition of completely iterative monad [23] follows this approach. For our purposes, we require a yet more general notion where we just distinguish some Kleisli morphisms as being guarded in certain output variables. We thus aim to work in a Kleisli category of a monad, but since our formalization and initial results can already be stated in any co-Cartesian category, we phrase them at this level of generality as long as possible.

Definition 3.1 (Abstractly guarded category/monad)**.** A co-Cartesian category $\mathbf{C}$ is *abstractly guarded* if it is equipped with a notion of *abstract guardedness*, i.e. with a relation between morphisms $f\colon X \to Y$ and summands $\sigma\colon Y' \hookrightarrow Y$ closed under the rules in Figure 1 where $f\colon X \to_\sigma Y$ denotes the fact that f and σ are in the relation in question.

A monad is *abstractly guarded* if its Kleisli category is abstractly guarded. A monad morphism $\alpha\colon \mathbb{T} \to \mathbb{S}$ between abstractly guarded monads $\mathbb{T}$, $\mathbb{S}$ is *abstractly guarded* if $f\colon X \to_\sigma TY$ implies $\alpha f\colon X \to_\sigma SY$.

$$\textbf{(trv)}\quad \frac{f\colon X \to Y}{\mathsf{in}_1\, f\colon X \to_{\mathsf{in}_2} Y + Z} \qquad\qquad \textbf{(par)}\quad \frac{f\colon X \to_\sigma Z \qquad g\colon Y \to_\sigma Z}{[f, g]\colon X + Y \to_\sigma Z}$$

$$\textbf{(cmp)}\quad \frac{f\colon X \to_{\mathsf{in}_2} Y + Z \qquad g\colon Y \to_\sigma V \qquad h\colon Z \to V}{[g, h]\, f\colon X \to_\sigma V}$$

FIGURE 1. Axioms of abstract guardedness.

The rules in Figure 1 are designed so as to enable a reformulation of the classical laws of iteration w.r.t. abstract guardedness, as we shall see in Section 5. Intuitively, **(trv)** states that if a program does not output anything via a summand of the output type then it is guarded in that summand. Rule **(par)** states that putting two guarded equation systems side by side again produces a guarded system. Finally, rule **(cmp)** states that guardedness is preserved under composition: if the unguarded part of the output of a program is postcomposed with a σ-guarded program, then the result is σ-guarded, no matter how the guarded part is transformed. That is, guardedness, once introduced, cannot be "undone" through sequential composition, but it can be "forgotten", as the following *weakening rule* indicates:

$$\textbf{(wkn)}\quad \frac{f\colon X \to_\sigma Y}{f\colon X \to_{\sigma\theta} Y},$$

where σ and θ are composable summands. This rule was originally part of our axiomatization [19] but it was later observed to be derivable from the other three [17]:

Proposition 3.2. *Rule **(wkn)** is derivable in the calculus of Figure 1.*

Proof. Let $\bar\sigma\colon Z \to Y$ be the complement of σ, thus $Y = Z + Y'$, $\sigma = \mathsf{in}_2$ and $\bar\sigma = \mathsf{in}_1$. Analogously we present Y' as $Z' + Y''$ with $\theta = \mathsf{in}_2$. In summary, Y is a coproduct of Z, Z' and Y'', $f\colon X \to_{\mathsf{in}_2} Z + (Z' + Y'')$, and we need to show that $f\colon X \to_{\mathsf{in}_2\,\mathsf{in}_2} Z + (Z' + Y'')$. Since $f = [\mathsf{in}_1, \mathsf{in}_2]\, f$, by **(cmp)** we are left to check that $\mathsf{in}_1\colon Z \to_{\mathsf{in}_2\,\mathsf{in}_2} Z + (Z' + Y'')$. Now $Z + (Z' + Y'')$ is also a coproduct of $Z + Z'$ and Y'', with evident injections; so $\mathsf{in}_1\colon Z \to_{\mathsf{in}_2\,\mathsf{in}_2} Z + (Z' + Y'')$ is equivalent to $\mathsf{in}_1\,\mathsf{in}_1\colon Z \to_{\mathsf{in}_2} (Z + Z') + Y''$, which is an instance of **(trv)**. □

Rule **(wkn)** is a weakening principle: If a program is guarded in some summand then it is guarded in any subsummand of that summand. Analogously, we obtain stability of guardedness under isomorphisms:

Proposition 3.3. *The rule*

$$\textbf{(iso)}\quad \frac{f\colon X \to_\sigma Y \qquad h\colon Y \cong Z}{h\, f\colon X \to_{h\sigma} Z},$$

is derivable in the calculus of Figure 1.

Proof. Let $\bar\sigma\colon W \to Y$ be the complement of σ, thus $Y = W + Y'$, $\sigma = \mathsf{in}_2$ and $\bar\sigma = \mathsf{in}_1$. Now, Z is a coproduct of W and and Y' with $h\,\mathsf{in}_1\colon W \to Z$ and $h\,\mathsf{in}_2\colon Y' \to Z$ as the coproduct injections, and h is the copair of $h\,\mathsf{in}_1$ and $h\,\mathsf{in}_2$ w.r.t. this coproduct structure. The rule in question now follows from **(cmp)**, using the fact that by **(trv)**, $h\,\mathsf{in}_1$ is $h\,\mathsf{in}_2$-guarded. □

We write $f\colon X \to_{i_1,\dots,i_k} X_1 + \ldots + X_n$ as a shorthand for $f\colon X \to_\sigma X_1 + \ldots + X_n$ with $\sigma = [\mathsf{in}_{i_1}, \ldots, \mathsf{in}_{i_k}]\colon X_{i_1} + \ldots + X_{i_k} \hookrightarrow X_1 + \ldots + X_n$. More generally, we sometimes need to refer to components of some X_{i_j}. This amounts to replacing the corresponding i_j with a sequence of pairs $i_j n_{j,m}$, and in_{i_j} with $\mathsf{in}_{i_j}[\mathsf{in}_{n_{j,1}}, \dots, \mathsf{in}_{n_{j,k_j}}]$, so, e.g. we write $f\colon X \to_{12,2} (Y + Z) + Z$ to mean that f is $[\mathsf{in}_1\,\mathsf{in}_2, \mathsf{in}_2]$-guarded. Where coproducts $Y + Z$ etc. appear in the rules, we mean any coproduct, not just some selected coproduct.

Recall that we have defined the notion of guardedness as a certain relation between morphisms and summands. Clearly, the *greatest* such relation is the one declaring all morphisms to be σ-guarded for all σ. We call categories (or monads) equipped with this notion of guardedness *totally guarded*. It turns out we also always have a *least* guardedness relation (originally called *trivial* [19]):

Definition 3.4 (Vacuous guardedness)**.** A morphism $f\colon X \to Y$ is *vacuously σ-guarded* for $\sigma\colon Z \hookrightarrow Y$ if f factors through the coproduct complement $\bar\sigma$ of σ.

Intuitively, f is vacuously guarded in $\sigma\colon Z \hookrightarrow Y$ if f does not output anything via the summand Z; observe that by the **(trv)** rule, vacuous guardedness always implies guardedness. Formally, we have:

Proposition 3.5. *By taking the abstractly guarded morphisms to be the vacuously guarded morphisms, we obtain the least guardedness relation making the given category into a guarded category.*

Proof. As indicated above, it is immediate from **(trv)** that every vacuously guarded morphism is guarded under any guardedness relation making the category into a guarded category. It remains to show that vacuous guardedness is closed under the rules in Figure 1; in the following we write $f\colon X \to_\sigma Y$ to mean that f is vacuously σ-guarded.

- **(trv)**: Immediate from the definition of vacuous guardedness.
- **(cmp)**: Suppose $f\colon X \to_2 Y + Z$, i.e. $f = \mathsf{in}_1\, w$ for some $w\,\mathsf{in}_1\colon X \to Y$. Now, for $g\colon Y \to_\sigma V$ and $h\colon Z \to V$, $[g,h]\, f = g\, w$. Let $\bar\sigma\colon W \hookrightarrow V$ be the complement of $\sigma\colon V' \hookrightarrow V$. By assumption, g factors through $\bar\sigma$, i.e. $w = \bar\sigma\, u$ for some u. Therefore $[g,h]\, f = \bar\sigma\, u\, w$, which by definition means that $[g,h]\, f$ is vacuously σ-guarded.
- **(par)**: Suppose that $f\colon X \to_\sigma Z$ and $g\colon Y \to_\sigma Z$, i.e. $f = \bar\sigma\, f'$ and $g = \bar\sigma\, g'$ for some $f'\colon X \to Z'$ and $g'\colon Y \to Z'$ where $\sigma\colon Z' \hookrightarrow Z$ and $\bar\sigma$ is the coproduct complement of σ. Then, of course, $[f,g] = \bar\sigma\,[f', g']$, i.e. $[f,g]\colon X + Y \to_\sigma Z$. □

We call a guarded category (or monad) *vacuously guarded* if its notion of abstract guardedness is given by vacuous guardedness. We note briefly how vacuous guardedness instantiates to Kleisli categories:

Lemma 3.6. *Let $\mathbb{T}$ be a monad on a category $\mathbf{C}$. A morphism $f\colon X \to T(Y + Z)$ is vacuously in_2-guarded iff f factors through $T\,\mathsf{in}_1$ in $\mathbf{C}$.*

Proof. Immediate from the fact that the left injection into the coproduct $Y + Z$ in the Kleisli category of $\mathbb{T}$ is $\eta\,\mathsf{in}_1$, and $(\eta\,\mathsf{in}_1)^\star = T\,\mathsf{in}_1$. □

The notion of abstract guardedness can thus vary on a large spectrum from *vacuous guardedness* to *total guardedness*, possibly detaching it from the initial intuition on guardedness. It is for this reason that we introduced the qualifier *abstract* into the terminology; for brevity, we will omit this qualifier in the sequel in contexts where no confusion is likely, speaking only of guarded monads, guarded morphisms etc.

Remark 3.7. One subtle feature of our axiomatization is that it allows for seemingly counterintuitive situations when a morphism is individually guarded in two disjoint summands, but not in their union. This can be illustrated by the following example. Let $\mathbb{T}$ be the algebraic monad induced by the theory of abelian groups presented, in additive notation, by binary $-$ alone. In this presentation, the zero element is presented by terms of the form $x-x$; the theory thus differs slightly from the more standard presentation in that there is no zero element in the absence of variables, i.e. $T\emptyset = \emptyset$. We equip $\mathbb{T}$ with vacuous guardedness. Now let $z : 1 \to T(\{x\} + \{y\})$ be the map that picks out the zero element. This morphism is both in_1-guarded and in_2-guarded, i.e. it factors both through $T\,\mathsf{in}_2$ and through $T\,\mathsf{in}_1$, since we can write the zero element both as $y-y$ and as $x-x$. However, z fails to be id-guarded, because it does not factor through $T\emptyset = \emptyset$.

Note that, conversely, collective guardedness does always imply individual guardedness, for by **(wkn)**, $f\colon X \to_{1,2} Y + Z$ implies both $f\colon X \to_1 Y + Z$ and $f\colon X \to_2 Y + Z$.

As usual, guardedness serves to identify systems of equations that admit solutions according to some global principle:

Definition 3.8 (Guarded (pre-)iterative category/monad)**.** Given $f\colon X \to_2 Y + X$, we say that $f^\dagger\colon X \to Y$ is a *solution* of f if $f^\dagger$ satisfies the *fixpoint identity* $f^\dagger = [\mathsf{id}, f^\dagger]\, f$. A guarded category is *guarded pre-iterative* if it is equipped with an *iteration operator* that assigns to every in_2-guarded morphism $f\colon X \to_2 Y + X$ a solution $f^\dagger$ of f. If every such f has a unique solution, we call the category *guarded iterative*.

A guarded monad is *guarded (pre-)iterative* if its Kleisli category is guarded (pre-)iterative. A guarded monad morphism $\alpha\colon \mathbb{T} \to \mathbb{S}$ between guarded pre-iterative monads $\mathbb{T}, \mathbb{S}$ is *iteration-preserving* if $\alpha f^\dagger = (\alpha f)^\dagger$ for every $f\colon X \to_2 T(Y + X)$.

We can readily check that the iteration operator preserves guardedness:

Proposition 3.9. *Let* $\mathbf{C}$ *be a guarded pre-iterative category, let* $\sigma\colon Z \hookrightarrow Y$*, and let* $f\colon X \to_{\sigma+\mathsf{id}} Y + X$*. Then* $f^\dagger\colon X \to_\sigma Y$*.*

Proof. Let $\bar\sigma\colon Z' \to Y$ be the complement of $\sigma\colon Z \to Y$, so we proceed under the assumption that $Y = Z' + Z$, $\sigma = \mathsf{in}_2$ and $\bar\sigma = \mathsf{in}_1$. Then

$$f^\dagger = [\mathsf{id}, f^\dagger]f = [[\mathsf{in}_1, \mathsf{in}_2], f^\dagger]\, f = [\mathsf{in}_1, [\mathsf{in}_2, f^\dagger]]\, [[\mathsf{in}_1, \mathsf{in}_2\,\mathsf{in}_1], \mathsf{in}_2\,\mathsf{in}_2]\, f.$$

By assumption, $f\colon X \to_{\mathsf{in}_2+\mathsf{id}} (Z' + Z) + X$. The morphism $h = [[\mathsf{in}_1, \mathsf{in}_2\,\mathsf{in}_1], \mathsf{in}_2\,\mathsf{in}_2]$ is simply an associativity isomorphism, for which $h\,(\mathsf{in}_2 + \mathsf{id}) = \mathsf{in}_2$, hence by **(iso)**, $[[\mathsf{in}_1, \mathsf{in}_2\,\mathsf{in}_1], \mathsf{in}_2\,\mathsf{in}_2]\, f\colon X \to_2 Z' + (Z + X)$. Since by **(trv)**, in_1 is in_2-guarded, we are done by **(cmp)**. ☐

We note that for guarded morphisms into guarded *iterative* monads, preservation of iteration is automatic:

Lemma 3.10. *Let* $\alpha\colon \mathbb{T} \to \mathbb{S}$ *be a guarded morphism between guarded pre-iterative monads* $\mathbb{T}$*,* $\mathbb{S}$ *with* $\mathbb{S}$ *being guarded iterative. Then* α *is iteration-preserving.*

Proof. Indeed, given $f\colon X \to_2 T(Y + X)$,

$$\begin{aligned} \alpha f^\dagger &= \alpha\, [\eta, f^\dagger]^\star f && \text{// fixpoint identity for } f^\dagger \\ &= [\eta, \alpha f^\dagger]^\star \alpha f && \text{// monad morphism} \end{aligned}$$

but this equation has $(\alpha f)^\dagger$ as its unique solution, hence $(\alpha f)^\dagger = \alpha f^\dagger$. ☐

In vacuously guarded categories, there is effectively nothing to iterate, so we have

Proposition 3.11. *Every vacuously guarded category is guarded iterative.*

Proof. Let $f\colon X \to_2 Y + X$, which by assumption means that $f = \mathsf{in}_1\, g$ for some g. Then for any $f^\dagger$ satisfying $f^\dagger = [\mathsf{id}, f^\dagger]\, f$, we have $f^\dagger = [\mathsf{id}, f^\dagger]\, f = [\mathsf{id}, f^\dagger]\, \mathsf{in}_1\, g = g$, which proves uniquenes of solutions. Moreover, $[\mathsf{id}, g]f = [\mathsf{id}, g]\, \mathsf{in}_1\, g = g$, which shows existence. □

We now revisit our motivating considerations on process algebra from the beginning of this section.

Example 3.12 (Generalized processes)**.** A natural semantic domain for finitely branching possibly infinite processes under strong bisimilarity with final results in X and atomic actions in A is the final coalgebra $\nu\gamma.\, \mathcal{P}_\omega(X + A \times \gamma)$ in the category of sets and functions, where $\mathcal{P}_\omega$ is the finite powerset monad. Alternatively, we can view inhabitants of this domain as equivalence classes of possibly non-well-founded terms over variables from X, which can also be thought of as process names, and over the operations $+$ of non-deterministic choice, deadlock $\emptyset$ and action prefixing $a.\,(\text{–})$. The latter view is useful for syntactic presentations of those processes that happen to be finite. Systems of recursive process definitions are naturally represented by morphisms $f\colon X \to \nu\gamma.\, \mathcal{P}_\omega((Y + X) + A \times \gamma)$ where X contains process names being defined and Y contains the remaining process names that can occur freely. For example, the system

$$x = y + a.\, x \tag{3.1}$$

corresponds to the following data: $X = \{x\}$, $Y = \{y\}$, $A = \{a\}$, and

$$f(x) = \mathsf{out}^{-1}\{\mathsf{in}_1\, y, \mathsf{in}_2\langle a, \mathsf{out}^{-1}\{\mathsf{in}_1\, x\}\rangle\}$$

(eliding the isomorphism $Y + X \cong \{x, y\}$). The generalization arising from this example is as follows: Given an endofunctor Σ on a co-Cartesian category $\mathbf{C}$ and a monad $\mathbb{T}$ such that final coalgebras $T_\Sigma X = \nu\gamma.\, T(X + \Sigma\gamma)$ exist, we obtain a corresponding monad $\mathbb{T}_\Sigma$ called the *generalized coalgebraic resumption monad transform* of $\mathbb{T}$. As above, we can view morphisms $f\colon X \to T_\Sigma(Y + X)$ as systems of recursive equations for *generalized processes* with $\mathbb{T}$ capturing the relevant computational effect (such as non-determinism) and Σ capturing *atomic steps* (such as actions $\Sigma = A \times \text{–}$).

Abstract guardedness can be used to effectively distinguish those systems $f\colon X \to T_\Sigma(Y+X)$ for which we can define *desirable* solutions $f^\dagger\colon X \to T_\Sigma Y$. For the moment, we proceed under the assumption that *desirable* means *unique*, for instance (3.1) has the unique solution $x = y + a.\,(y + a.\,(\ldots))$. Let us recall the existing approach to defining guardedness in this context via completely iterative monads [23], which are based on idealized monads [23, Definition 5.5]. To make this precise, recall some definitions.

Definition 3.13 (Monad modules, idealized monads)**.** A *module* over a monad $\mathbb{T}$ on $\mathbf{C}$ is a pair $(M, \text{–}^\circ)$, where M is an endomap over the objects of $\mathbf{C}$, while the lifting $(\text{–})^\circ$ is a map $\mathsf{Hom}(X, TY) \to \mathsf{Hom}(MX, MY)$ such that the following laws are satisfied:

$$\eta^\circ = \mathsf{id}, \qquad\qquad g^\circ f^\circ = (g^\star f)^\circ.$$

Note that M extends to an endofunctor by taking $Mf = (\eta f)^\circ$. A *module-to-monad morphism* is a natural transformation $\xi\colon M \to T$ that satisfies $\xi f^\circ = f^\star \xi$. We call the tuple $(\mathbb{T}, M, \text{–}^\circ, \xi)$ an *idealized monad*; when no confusion is likely, we refer to these data just as $\mathbb{T}$. An *idealized monad morphism* between idealized monads $((T, \eta^T, \text{–}^\star), M, \text{–}^\circ, \xi)$ and $((S, \eta^S, \text{–}^{\maltese}), N, \text{–}^\bullet, \xi')$ is a pair (α, β) where $\alpha\colon T \to S$ is a monad morphism while $\beta\colon M \to N$ is a natural transformation satisfying $\alpha\xi = \xi'\beta$ and $\beta f^\star = f^{\maltese}\beta$.

Example 3.14. It follows from previous results [30, Corollary 3.13] that the monad $\mathbb{T}_\Sigma$ from Example 3.12 is idealized when equipped with the module $T\Sigma T_\Sigma$. In the concrete case where $\mathbb{T} = \mathcal{P}_\omega$ and $\Sigma = A \times (\text{–})$, i.e. $T_\Sigma X = \nu\gamma.\, \mathcal{P}_\omega((Y + X) + A \times \gamma)$, the module $\mathcal{P}_\omega(A \times T_\Sigma)$ contains processes that consist of (finitely many) non-deterministic branches all of which begin with an action.

Milius [23] defines guardedness only for equation morphisms, i.e. morphisms of type $X \to T(Y + X)$. Extending this notion in the obvious way to morphisms of type $X \to T(Y + Z)$ as required in our framework, we obtain the following definition:

Definition 3.15 (Completely iterative monads)**.** Given an idealized monad $(\mathbb{T}, M, \text{–}^\circ, \xi)$, a morphism $f\colon X \to T(Y+Z)$ is *guarded* if it factors via $[\eta\,\mathsf{in}_1, \xi]\colon Y + M(Y+Z) \to T(Y+Z)$. The monad $\mathbb{T}$ is *completely iterative* if every guarded $f\colon X \to T(Y + X)$ in this sense has a unique solution.

It turns out that the above notion of guardedness is not an instance of abstract guardedness; specifically, it does not satisfy our **(par)** rule. Equation (3.1) provides a good illustration of what happens: although both terms y and $a.\,x$ are guarded in x, we cannot factor the corresponding term $X \to T(Y + X)$ through any $[\eta\,\mathsf{in}_1, \xi]\colon Y + M(Y + X) \to T(Y + X)$ due to the top-level nondeterministic choice.

Fortunately, we can fix this by noticing that completely iterative monads actually support iteration for a wider class of morphisms:

Definition 3.16. Let $(\mathbb{T}, M, \text{–}^\circ, \xi)$ be an idealized monad. Given $\sigma\colon Z \hookrightarrow Y$, we say that a morphism $f\colon X \to TY$ is *weakly σ-guarded* if it factors through $[\eta\bar\sigma, \xi]^\star\colon T(Y' + MY) \to TY$ for a complement $\bar\sigma\colon Y' \hookrightarrow Y$ of σ.

Since a morphism that factors as $[\eta\,\mathsf{in}_1, \xi]f$ can be rewritten as $[\eta\,\mathsf{in}_1, \xi]^\star\eta f$, every guarded morphism in an idealized monad is also weakly guarded.

Theorem 3.17. *Let $(\mathbb{T}, M, \text{–}^\circ, \xi)$ be an idealized monad. Then the following hold.*

(1) *$\mathbb{T}$ becomes abstractly guarded when equipped with weak guardedness as the notion of abstract guardedness.*
(2) *If $\mathbb{T}$ is completely iterative, then every weakly in_2-guarded morphism $f\colon X \to T(Y + X)$ has a unique solution.*
(3) *If (α, β) is an idealized monad morphism, then α preserves weak guardedness.*

That is, completely iterative monads are abstractly guarded iterative monads w.r.t. weak guardedness.

Proof. (1)*:* We need to verify that weak guardedness is closed under the rules from Definition 3.1.

- **(trv)** Given a morphism $f\colon X \to TY$, the following holds:

$$\begin{aligned}(T\,\mathsf{in}_1)f &= (\eta\,\mathsf{in}_1)^\star f && /\!/ \text{ Kleisli}\\ &= ([\eta\,\mathsf{in}_1, \xi]\,\mathsf{in}_1)^\star f && /\!/ \text{ coproducts}\\ &= [\eta\,\mathsf{in}_1, \xi]^\star(T\,\mathsf{in}_1)f && /\!/ \text{ Kleisli}\end{aligned}$$

- **(cmp)** Given $f\colon X \to_2 T(Y + Z)$, $g\colon Y \to_\sigma TV$, and $h\colon Y \to TV$, assume that f factors as $[\eta\,\mathsf{in}_1, \xi]^\star f'$, while g factors as $[\eta\bar\sigma, \xi]^\star g'$. Then, the following holds:

$$[g, h]^\star f = [[\eta\bar\sigma, \xi]^\star g', h]^\star[\eta\,\mathsf{in}_1, \xi]^\star f'$$

$$
\begin{aligned}
&= ([[\eta\bar\sigma,\xi]^\star g',h]^\star[\eta\,\mathsf{in}_1,\xi])^\star f' && \text{// Kleisli}\\
&= [[[\eta\bar\sigma,\xi]^\star g',h]^\star\eta\,\mathsf{in}_1,[[\eta\bar\sigma,\xi]^\star g',h]^\star\xi]^\star f' && \text{// coproducts}\\
&= [[[\eta\bar\sigma,\xi]^\star g',h]\,\mathsf{in}_1,[[\eta\bar\sigma,\xi]^\star g',h]^\star\xi]^\star f' && \text{// Kleisli}\\
&= [[\eta\bar\sigma,\xi]^\star g',[[\eta\bar\sigma,\xi]^\star g',h]^\star\xi]^\star f' && \text{// coproducts}\\
&= [[\eta\bar\sigma,\xi]^\star g',\xi[[\eta\bar\sigma,\xi]^\star g',h]^\circ]^\star f' && \text{// module-to-monad morphism}\\
&= [[\eta\bar\sigma,\xi]^\star g',[\eta\bar\sigma,\xi]\,\mathsf{in}_2[[\eta\bar\sigma,\xi]^\star g',h]^\circ]^\star f' && \text{// coproducts}\\
&= [[\eta\bar\sigma,\xi]^\star g',[\eta\bar\sigma,\xi]^\star\eta\,\mathsf{in}_2[[\eta\bar\sigma,\xi]^\star g',h]^\circ]^\star f' && \text{// Kleisli}\\
&= ([\eta\bar\sigma,\xi]^\star[g',\eta\,\mathsf{in}_2[[\eta\bar\sigma,\xi]^\star g',h]^\circ])^\star f' && \text{// coproducts}\\
&= [\eta\bar\sigma,\xi]^\star[g',\eta\,\mathsf{in}_2[[\eta\bar\sigma,\xi]^\star g',h]^\circ]^\star f'. && \text{// Kleisli}
\end{aligned}
$$

- **(par)** Given a morphism $f\colon X\to_\sigma TZ$ and $Y\to_\sigma TZ$ assume that f factors as $[\eta\,\mathsf{in}_1,\xi]^\star f'$, and g factors as $[\eta\,\mathsf{in}_1,\xi]^\star g'$. Then, the following holds:

$$
\begin{aligned}
[f,g] &= [[\eta\bar\sigma,\xi]^\star f',[\eta\bar\sigma,\xi]^\star g'] && \text{// guardedness}\\
&= [\eta\bar\sigma,\xi]^\star[f',g']. && \text{// coproducts}
\end{aligned}
$$

(2)*:* Let $f=[\eta\,\mathsf{in}_1,\xi]^\star j$ for a morphism $j\colon X\to T(Y+M(Y+X))$. We define an auxiliary morphism $g=[\eta\,\mathsf{in}_1,j]^\star\xi\colon M(Y+X)\to T(Y+M(Y+X))$. Note that g is guarded (in the sense of [23]), since it can be rewritten as follows:

$$
\begin{aligned}
[\eta\,\mathsf{in}_1,j]^\star\xi &= \xi[\eta\,\mathsf{in}_1,j]^\circ && \text{// module-to-monad morphism}\\
&= [\eta\,\mathsf{in}_1,\xi]\,\mathsf{in}_2[\eta\,\mathsf{in}_1,j]^\circ. && \text{// coproducts}
\end{aligned}
$$

Thus, g has a unique solution $g^\dagger\colon M(Y+X)\to TY$. We use it to define a solution to f, namely $f^\ddagger=[\eta,g^\dagger]^\star j$. It is left to show that it is indeed a solution and that it is unique:

- *Solution:*

$$
\begin{aligned}
f^\ddagger &= [\eta,g^\dagger]^\star j\\
&= [\eta,[\eta,g^\dagger]^\star g]^\star j && \text{// solution}\\
&= [\eta,[\eta,g^\dagger]^\star[\eta\,\mathsf{in}_1,j]^\star\xi]^\star j\\
&= [\eta,([\eta,g^\dagger]^\star[\eta\,\mathsf{in}_1,j])^\star\xi]^\star j && \text{// Kleisli}\\
&= [\eta,[[\eta,g^\dagger]^\star\eta\,\mathsf{in}_1,[\eta,g^\dagger]^\star j]^\star\xi]^\star j && \text{// coproducts}\\
&= [\eta,[[\eta,g^\dagger]\,\mathsf{in}_1,[\eta,g^\dagger]^\star j]^\star\xi]^\star j && \text{// Kleisli}\\
&= [\eta,[\eta,[\eta,g^\dagger]^\star j]^\star\xi]^\star j && \text{// coproducts}\\
&= [[\eta,[\eta,g^\dagger]^\star j]\,\mathsf{in}_1,[\eta,[\eta,g^\dagger]^\star j]^\star\xi]^\star j && \text{// coproducts}\\
&= [[\eta,[\eta,g^\dagger]^\star j]^\star\eta\,\mathsf{in}_1,[\eta,[\eta,g^\dagger]^\star j]^\star\xi]^\star j && \text{// Kleisli}\\
&= ([\eta,[\eta,g^\dagger]^\star j]^\star[\eta\,\mathsf{in}_1,\xi])^\star j && \text{// coproducts}\\
&= [\eta,[\eta,g^\dagger]^\star j]^\star[\eta\,\mathsf{in}_1,\xi]^\star j && \text{// Kleisli}\\
&= [\eta,f^\ddagger]^\star f.
\end{aligned}
$$

- *Uniqueness:* Let $r\colon X \to TY$ be a solution of f, that is, $r = [\eta, r]^\star f$. First, we calculate:

$$\begin{aligned}
[\eta, r]^\star \xi &= [\eta, [\eta, r]^\star f]^\star \xi \\
&= [\eta, [\eta, r]^\star [\eta\,\mathsf{in}_1, \xi]^\star j]^\star \xi \\
&= [\eta, ([\eta, r]^\star [\eta\,\mathsf{in}_1, \xi])^\star j]^\star \xi && \text{// Kleisli} \\
&= [\eta, [[\eta, r]^\star \eta\,\mathsf{in}_1, [\eta, r]^\star \xi]^\star j]^\star \xi && \text{// coproducts} \\
&= [\eta, [[\eta, r]\,\mathsf{in}_1, [\eta, r]^\star \xi]^\star j]^\star \xi && \text{// Kleisli} \\
&= [\eta, [\eta, [\eta, r]^\star \xi]^\star j]^\star \xi && \text{// coproducts} \\
&= [[\eta, [\eta, r]^\star \xi]\,\mathsf{in}_1, [\eta, [\eta, r]^\star \xi]^\star j]^\star \xi && \text{// coproducts} \\
&= [[\eta, [\eta, r]^\star \xi]^\star \eta\,\mathsf{in}_1, [\eta, [\eta, r]^\star \xi]^\star j]^\star \xi && \text{// Kleisli} \\
&= ([\eta, [\eta, r]^\star \xi]^\star [\eta\,\mathsf{in}_1, j])^\star \xi && \text{// coproducts} \\
&= [\eta, [\eta, r]^\star \xi]^\star [\eta\,\mathsf{in}_1, j]^\star \xi && \text{// Kleisli} \\
&= [\eta, [\eta, r]^\star \xi]^\star g.
\end{aligned}$$

Thus, $[\eta, r]^\star \xi$ is a solution of g. By uniqueness, we obtain that $g^\dagger = [\eta, r]^\star \xi$. With this, we can check the uniqueness of $f^\ddagger$:

$$\begin{aligned}
r &= [\eta, r]^\star f \\
&= [\eta, r]^\star [\eta\,\mathsf{in}_1, \xi]^\star j \\
&= ([\eta, r]^\star [\eta\,\mathsf{in}_1, \xi])^\star j && \text{// Kleisli} \\
&= [[\eta, r]^\star \eta\,\mathsf{in}_1, [\eta, r]^\star \xi]^\star j && \text{// coproducts} \\
&= [[\eta, r]\,\mathsf{in}_1, [\eta, r]^\star \xi]^\star j && \text{// Kleisli} \\
&= [\eta, [\eta, r]^\star \xi]^\star j && \text{// coproducts} \\
&= [\eta, g^\dagger]^\star j && \text{// the above} \\
&= f^\ddagger.
\end{aligned}$$

(3)*:* Let (α, β) be as in Definition 3.13. Let $f\colon X \to TY$ be weakly σ-guarded. This means that f factors as $[\eta^T \bar\sigma, \xi]^\star f'$ for a morphism $f'\colon X \to T(Y' + MY)$. We need to show that $\alpha f\colon X \to SY$ factors as $[\eta^S \bar\sigma, \xi']^{\bigstar} g$ for some $g\colon X \to S(Y' + NY)$. We calculate:

$$\begin{aligned}
\alpha f &= \alpha[\eta^T \bar\sigma, \xi]^\star f' && \text{// factorisation of } f \\
&= (\alpha[\eta^T \bar\sigma, \xi])^{\bigstar} \alpha f' && \text{// monad morphism} \\
&= [\alpha\eta^T \bar\sigma, \alpha\xi]^{\bigstar} \alpha f' && \text{// coproducts} \\
&= [\eta^S \bar\sigma, \alpha\xi]^{\bigstar} \alpha f' && \text{// monad morphism} \\
&= [\eta^S \bar\sigma, \xi'\beta]^{\bigstar} \alpha f' && \text{// idealized monad morphism} \\
&= ([\eta^S \bar\sigma, \xi'](\mathsf{id} + \beta))^{\bigstar} \alpha f' && \text{// coproducts} \\
&= ([\eta^S \bar\sigma, \xi']^{\bigstar} \eta^S(\mathsf{id} + \beta))^{\bigstar} \alpha f' && \text{// Kleisli} \\
&= [\eta^S \bar\sigma, \xi']^{\bigstar} (\eta^S(\mathsf{id} + \beta))^{\bigstar} \alpha f' && \text{// Kleisli}
\end{aligned}$$

□

4. Parametrizing Guardedness

Uustalu [35] defines a *parametrized monad* to be a functor from a category $\mathbf{C}$ to the category of monads over $\mathbf{C}$. We need a minor adaptation of this notion where we allow parameters from a different category than $\mathbf{C}$, and simultaneously introduce a guarded version of parametrized monads:

Definition 4.1 (Parametrized guarded monad)**.** A *parametrized (guarded) monad* is a functor from a category $\mathbf{D}$ to the category of (guarded) monads and (guarded) monad morphisms over $\mathbf{C}$. Alternatively (by uncurrying), it is a bifunctor $\#\colon \mathbf{C} \times \mathbf{D} \to \mathbf{C}$ such that for any $X \in |\mathbf{D}|$, $-\#X\colon \mathbf{C} \to \mathbf{C}$ is a (guarded) monad, and for every $f\colon Z \to V$, $\mathsf{id} \# f\colon X \# Z \to X \# V$ is the X-component of a (guarded) monad morphism $-\#f\colon -\#Z \to -\#V$, explicitly,

$$(\mathsf{id} \# f)\,\eta = \eta \qquad\qquad (\mathsf{id} \# f)\, g^\star = ((\mathsf{id} \# f)\, g)^\star(\mathsf{id} \# f) \tag{4.1}$$

for any $g\colon X \to Y$ and, in the guarded case,

$$g\colon Z \to_\sigma V \# X \quad \text{implies} \quad (\mathsf{id} \# f)\, g\colon Z \to_\sigma V \# Y.$$

A *parametrized (guarded) monad morphism* between parametrized (guarded) monads qua functors into the category of (guarded) monads over $\mathbf{C}$ is a natural transformation that is componentwise a (guarded) monad morphism. In uncurried notation, given parametrized monads $\#, \hat{\#}\colon \mathbf{C} \times \mathbf{D} \to \mathbf{C}$ a natural transformation $\alpha\colon \# \to \hat{\#}$ is a parametrized (guarded) monad morphism if for each $X \in |\mathbf{D}|$, $\alpha_{-,X}\colon -\#X \to -\hat{\#}X$ is a (guarded) monad morphism.

A parametrized guarded monad $\#$ is *guarded (pre-)iterative* if each monad $-\#X$ is guarded (pre-)iterative and the monad morphisms $-\#f$ are iteration-preserving, i.e.

$$(\mathsf{id} \# f)\, g^\dagger = ((\mathsf{id} \# f)\, g)^\dagger. \tag{4.2}$$

Note that by Lemma 3.10, condition (4.2) is automatic for guarded iterative parametrized monads.

In the sequel, we tend to use the same notation for parametrized monads as for the non-parametrized case, assuming that omitted information is understood from the context. For example, the monad unit $\eta_{X,Y}\colon X \to X \# Y$ is additionally parametrized by Y, and both parameters will be occasionally omitted unless confusion arises. Kleisli lifting assigns $f^\star\colon X \# Z \to Y \# Z$ to $f\colon X \to Y \# Z$, and for fixed Z all monad laws can be used for parametrized monads as stated for non-parametrized monads. The connection between Kleisli lifting and the functor part of the monad can now be restated as follows: $(f \# \mathsf{id}_Z) = (\eta_{Y,Z} f)^\star$ where $f\colon X \to Y \# Z$.

Example 4.2. For purposes of the present work, the most important example (taken from [35]) is $\# = T(-+\Sigma -)\colon \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ where $\mathbb{T}$ is a (non-parametrized) monad on $\mathbf{C}$ and Σ is an endofunctor on $\mathbf{C}$. Informally, $\mathbb{T}$ captures a computational effect, e.g. nondeterminism for T being a (bounded) powerset monad, and Σ captures a signature of actions, e.g. $\Sigma X = A \times X$, as in Example 3.12. Specifically, taking $A = 1$ we obtain $X \# Y = T(X + Y)$; in this case, we have only one guard, which can be interpreted as a delay. The second argument of $\#$ can thus be thought of as designated for guarded recursion.

Theorem 4.3. *Let $\#\colon \mathbf{C} \times (\mathbf{C} \times \mathbf{D}) \to \mathbf{C}$ be a parametrized monad, with unit η and Kleisli lifting $(-)^\star$. Then*

$$X \#^\nu Y = \nu\gamma.\, X \# (\gamma, Y)$$

defines a parametrized monad $\#^\nu\colon \mathbf{C}\times\mathbf{D}\to\mathbf{C}$*, whose unit and Kleisli lifting we denote* η^ν *and* $(-)^{\bigstar}$*, respectively. Moreover,*

(1) *If* $\#$ *is guarded, then so is* $\#^\nu$*, with guardedness defined as follows: given* $\sigma\colon Y'\hookrightarrow Y$*,* $f\colon X\to Y\#^\nu Z$ *is* σ*-guarded if* $\mathsf{out}\, f\colon X\to Y\#(Y\#^\nu Z, Z)$ *is* σ*-guarded; the correspondence* $\#\mapsto\#^\nu$ *extends to a functor between the respective categories of guarded parametrized monads.*

(2) *If* $\#$ *is guarded pre-iterative, with an iteration operator* $(-)^\dagger$*, then so is* $\#^\nu$*, with the iteration operator* $(-)^\ddagger$ *sending* $f\colon X\to_2 (Y+X)\#^\nu Z$ *to* $f^\ddagger\colon X\to Y\#^\nu Z$ *as follows:*
$$f^\ddagger = \mathsf{coit}\Big([\eta_Y, (\mathsf{out}\, f)^\dagger]^\star\,\mathsf{out}\colon (Y+X)\#^\nu Z\to Y\#(((Y+X)\#^\nu Z), Z)\Big)\,\eta^\nu_{Y+X,Z}\,\mathsf{in}_2$$

(3) *If* $\#$ *is guarded iterative, then so is* $\#^\nu$*, with solutions described as in the previous clause.*

To better understand the typing in the second clause above, note that

- $\mathsf{out}\, f\colon X\to_2 (Y+X)\#((Y+X)\#^\nu Z, Z)$, so
- $(\mathsf{out}\, f)^\dagger\colon X\to Y\#((Y+X)\#^\nu Z, Z)$;
- the right-most occurrence of out has type
$$\mathsf{out}\colon (Y+X)\#^\nu Z\to (Y+X)\#((Y+X)\#^\nu Z, Z);$$
- the $\mathsf{coit}(\dots)$ subterm has type $(Y+X)\#^\nu Z\to Y\#^\nu Z$.

In case there is only one parameter of type $\mathbf{C}$, i.e. $\#\colon\mathbf{C}\times\mathbf{C}\to\mathbf{C}$, the typing simplifies slightly: Now $\#^\nu$ is just a monad on $\mathbf{C}$, which we denote by $F_\#$ (i.e. $F_\# X = X\#^\nu())$. We write η^ν, $(-)^{\bigstar}$ for the corresponding monad structure. Then given $f\colon X\to_2 F_\#(Y+X)$,

- $\mathsf{out}\, f\colon X\to_2 (Y+X)\# F_\#(Y+X)$;
- $(\mathsf{out}\, f)^\dagger\colon X\to_2 Y\# F_\#(Y+X)$;
- the right-most occurrence of out has type
$$\mathsf{out}\colon F_\#(Y+X)\to (Y+X)\# F_\#(Y+X);$$
- the $\mathsf{coit}(\dots)$ subterm has type $F_\#(Y+X)\to F_\# Y$;
- and, of course, $f^\ddagger\colon X\to F_\# Y$.

Proof (Theorem 4.3). (1): By currying we equivalently view $\#$ as a functor from $\mathbf{D}$ to the category of parametrized guarded monads of type $\mathbf{C}\times\mathbf{C}\to\mathbf{C}$, and the transformation $\#\mapsto\#^\nu$ as given pointwise. It therefore suffices to show that the assignment
$$\#\mapsto F_\# \qquad\text{where}\qquad F_\# X = \nu\gamma.\, X\#\gamma$$
extends to a functor from parametrized guarded monads of type $\mathbf{C}\times\mathbf{C}\to\mathbf{C}$ to guarded monads over $\mathbf{C}$ where guardedness for $F_\#$ is defined as follows: $f\colon X\to F_\# Y$ is σ-guarded iff $\mathsf{out}\, f\colon X\to Y\# F_\# Y$ is σ-guarded w.r.t. $\#$. Uustalu [35] already proves that $F_\#$ is a monad; we proceed to check that his construction is in fact functorial.

As indicated above, we denote the monad structure on $F_\#$ by η^ν, $(-)^{\bigstar}$. These data are uniquely determined by commutation of
$$\begin{array}{ccc} X & \xrightarrow{\eta^\nu_X} & F_\# X \\ {\scriptstyle \eta_{X,F_\# X}}\downarrow & & \downarrow{\scriptstyle \mathsf{out}} \\ X\# F_\# X & = & X\# F_\# X \end{array} \qquad \begin{array}{ccc} F_\# X + F_\# Y & \xrightarrow{[f^{\bigstar},\mathsf{id}]} & F_\# Y \\ {\scriptstyle [\hat f, (Y\#\mathsf{in}_2)\,\mathsf{out}]}\downarrow & & \downarrow{\scriptstyle \mathsf{out}} \\ Y\#(F_\# X + F_\# Y) & \xrightarrow{Y\#[f^{\bigstar},\mathsf{id}]} & Y\#(F_\# Y) \end{array} \tag{4.3}$$

where

$$\hat{f} = \Big(F_{\#}X \xrightarrow{\mathsf{out}} X \# F_{\#}X \xrightarrow{X\#\mathsf{in}_1} X\#(F_{\#}X + F_{\#}Y) \xrightarrow{\bar{f}^{\star}} Y\#(F_{\#}X + F_{\#}Y)\Big)$$
$$\bar{f} = \Big(X \xrightarrow{f} F_{\#}Y \xrightarrow{\mathsf{out}} X\#F_{\#}Y \xrightarrow{Y\#\mathsf{in}_2} Y\#(F_{\#}X + F_{\#}Y)\Big).$$

That is, η^{ν}_X is the unique $(X\#-)$-coalgebra morphism $(X, \eta_{X,F_{\#}X}) \to (F_{\#}X, \mathsf{out})$, and $[f^{\bigstar}, \mathsf{id}]$ is the unique $(Y \# -)$-coalgebra morphism

$$(F_{\#}X + F_{\#}Y,\ [\hat{f}, (Y\#\mathsf{in}_2)\,\mathsf{out}]) \to (F_{\#}Y, \mathsf{out}),$$

the latter being essentially a definition of $f^{\bigstar}$ by primitive corecursion. In the sequel, we will omit the object part of coalgebras when convenient, saying, e.g., that η^{ν}_X is a coalgebra morphism $\eta_{X,F_{\#}X} \to \mathsf{out}$.

We need to define the action of F on morphisms: Let $\#'$ be a further parametrized monad, with all data of $\#'$ and $F_{\#'}$ indicated by primes, and let

$$\alpha\colon \# \to \#'$$

be a parametrized monad morphism. We then define a monad morphism $F_\alpha\colon F_{\#} \to F_{\#'}$ by commutation of

$$\begin{array}{ccc} F_{\#}X & \xrightarrow{(F_\alpha)_X} & F_{\#'}X \\ {\scriptstyle \alpha_{X,F_{\#}X}\,\mathsf{out}}\downarrow & & \downarrow{\scriptstyle \mathsf{out}'} \\ X\#' F_{\#}X & \xrightarrow{X\#(F_\alpha)_X} & X\#' F_{\#'}X, \end{array}$$

i.e. $(F_\alpha)_X$ is the unique $(X \#' -)$-coalgebra morphism $(F_{\#}X, \alpha\,\mathsf{out}) \to (F_{\#'}X, \mathsf{out}')$.

We first check functoriality of F. For preservation of identities, just note that $\mathsf{id}\colon (F_{\#}X, \mathsf{id}\,\mathsf{out}) \to (F_{\#}X, \mathsf{out})$ is a coalgebra morphism. For preservation of composition, we have that if $\beta\colon \#' \to \#''$ is a further parametrized monad morphism then by naturality of β, the $(X \#' -)$-coalgebra morphism $(F_\alpha)_X\colon \alpha\,\mathsf{out} \to \mathsf{out}'$ is also an $(X \#'' -)$-coalgebra morphism $\beta\alpha\,\mathsf{out} \to \beta\,\mathsf{out}'$; so $(F_\beta)_X(F_\alpha)_X$ is a coalgebra morphism $\beta\alpha\,\mathsf{out} \to \mathsf{out}''$, and hence equals $(F_{\beta\alpha})_X$.

It remains to verify that F_α is indeed a monad morphism. First, we show compatibility with the unit, i.e.

$$(F_\alpha)_X\,\eta^{\nu}_X = \eta'^{\nu}_X\colon X \to F_{\#'}X.$$

We note that by naturality of α, the $(X \# -)$-coalgebra morphism $\eta^{\nu}_X\colon \eta_{X,F_{\#}X} \to \mathsf{out}$ is also an $(X\#'-)$-coalgebra morphism $\eta'_{X,F_{\#}X} = \alpha\,\eta_{X,F_{\#}X} \to \alpha\,\mathsf{out}$, so that $(F_\alpha)_X\,\eta^{\nu}_X$ is a coalgebra morphism $\eta'_{X,F_{\#}X} \to \mathsf{out}'$ and hence equals η'^{ν}_X.

For compatibility of F_α with Kleisli lifting, we have to show that for $f\colon X \to F_{\#}Y$,

$$(F_\alpha\,f)^{\bigstar'}F_\alpha = F_\alpha\,f^{\bigstar}.$$

We strengthen this goal to one concerning $[f^{\bigstar}, \mathsf{id}]$, specifically we show that

$$\begin{array}{ccc} F_{\#}X + F_{\#}Y & \xrightarrow{[f^{\bigstar},\mathsf{id}]} & F_{\#}Y \\ {\scriptstyle (F_\alpha)_X + (F_\alpha)_Y}\downarrow & & \downarrow{\scriptstyle (F_\alpha)_Y} \\ F_{\#'}X + F_{\#'}Y & \xrightarrow{[((F_\alpha)_Y f)^{\bigstar'},\mathsf{id}]} & F_{\#'}Y \end{array}$$

commutes. By definition, the bottom arrow is a $(Y\#' -)$-coalgebra morphism

$$[\widehat{(F_\alpha)_Y f}, (Y\#'\,\mathsf{in}_2)\,\mathsf{out}'] \to \mathsf{out}',$$

and by now-familiar arguments, the top and right-hand arrows compose to yield a $(Y \mathbin{\#}' -)$-coalgebra morphism

$$[\alpha\hat{f}, \alpha(Y \mathbin{\#} \mathsf{in}_2)\,\mathsf{out}] \to \mathsf{out}'\,.$$

It therefore suffices to show that

$$(F_\alpha)_X + (F_\alpha)_Y \colon [\alpha\hat{f}, \alpha(Y \mathbin{\#} \mathsf{in}_2)\,\mathsf{out}] \to [\widehat{(F_\alpha)_Y f}, (Y \mathbin{\#}' \mathsf{in}_2)\,\mathsf{out}']$$

is a $(Y \mathbin{\#}' -)$-coalgebra morphism. We first check commutation of the corresponding square on the right-hand summand $F_{\#} Y$:

$$\begin{aligned}
&(Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))\alpha(Y \mathbin{\#} \mathsf{in}_2)\,\mathsf{out} \\
&= (Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))(Y \mathbin{\#}' \mathsf{in}_2)\alpha\,\mathsf{out} && /\!/ \text{ naturality of } \alpha \\
&= (Y \mathbin{\#}' \mathsf{in}_2)(Y \mathbin{\#}'(F_\alpha)_Y)\alpha\,\mathsf{out} \\
&= (Y \mathbin{\#}' \mathsf{in}_2)\,\mathsf{out}'(F_\alpha)_Y. && /\!/ \text{ definition of } F_\alpha
\end{aligned}$$

For commutation on the left-hand summand we have to show that

$$(Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))\,\alpha\hat{f} = \widehat{(F_\alpha)_Y f}(F_\alpha)_X. \tag{4.4}$$

We rewrite the left-hand side of (4.4):

$$\begin{aligned}
&(Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))\,\alpha\hat{f} \\
&= (Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))\,\alpha\bar{f}^{\star}(X \mathbin{\#} \mathsf{in}_1)\,\mathsf{out} && /\!/ \text{ definition of } \hat{f} \\
&= (Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))\,(\alpha\bar{f})^{\star'}\alpha(X \mathbin{\#} \mathsf{in}_1)\,\mathsf{out} && /\!/\ \alpha \text{ a monad morphism} \\
&= (Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))\,(\alpha\bar{f})^{\star'}(X \mathbin{\#}' \mathsf{in}_1)\alpha\,\mathsf{out}\,. && /\!/ \text{ naturality of } \alpha
\end{aligned}$$

We next rewrite the right-hand side of (4.4):

$$\begin{aligned}
&\widehat{(F_\alpha)_Y f}(F_\alpha)_X \\
&= \overline{(F_\alpha)_Y f}^{\star'}(X \mathbin{\#}' \mathsf{in}_1)\,\mathsf{out}'(F_\alpha)_X && /\!/ \text{ definition of } \widehat{(F_\alpha)_Y f} \\
&= \overline{(F_\alpha)_Y f}^{\star'}(X \mathbin{\#}' \mathsf{in}_1)(X \mathbin{\#}'(F_\alpha)_X)\,\alpha\,\mathsf{out} && /\!/ \text{ definition of } \widehat{(F_\alpha)_Y f} \\
&= \overline{(F_\alpha)_Y f}^{\star'}(X \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))(X \mathbin{\#}' F_{\mathsf{in}_1})\,\alpha\,\mathsf{out}
\end{aligned}$$

It thus suffices to show that

$$(Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))(\alpha\bar{f})^{\star'} = \overline{(F_\alpha)_Y f}^{\star'}(X \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y)). \tag{4.5}$$

We further rewrite the right-hand side of (4.5):

$$\begin{aligned}
&\overline{(F_\alpha)_Y f}^{\star'}(X \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y)) \\
&= ((Y \mathbin{\#}' \mathsf{in}_2)\,\mathsf{out}'(F_\alpha)_Y f)^{\star'} \\
&\qquad (X \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y)) && /\!/ \text{ definition of } \overline{(F_\alpha)_Y f} \\
&= ((Y \mathbin{\#}' \mathsf{in}_2)(Y \mathbin{\#}'(F_\alpha)_Y)\alpha\,\mathsf{out}\,f)^{\star'} \\
&\qquad (X \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y)) && /\!/ \text{ definition of } (F_\alpha)_Y \\
&= ((Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))(Y \mathbin{\#}' \mathsf{in}_2)\,\alpha\,\mathsf{out}\,f)^{\star'} \\
&\qquad (X \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y)) \\
&= (Y \mathbin{\#}'((F_\alpha)_X + (F_\alpha)_Y))((Y \mathbin{\#}' \mathsf{in}_2)\,\alpha\,\mathsf{out}\,f)^{\star'}
\end{aligned}$$

where we use in the last step that $Y \#' ((F_\alpha)_X + (F_\alpha)_Y)$ is a monad morphism. We have thus reduced (4.5) to showing that

$$\alpha \bar{f} = (Y \#' \mathsf{in}_2)\alpha\, \mathsf{out}\, f.$$

But this is straightforward:

$$\begin{aligned} \alpha \bar{f} &= \alpha\,(Y \# \mathsf{in}_2)\,\mathsf{out}\, f && /\!\!/ \text{ definition of } \bar{f} \\ &= (Y \#' \mathsf{in}_2)\,\alpha\,\mathsf{out}\, f. && /\!\!/ \text{ naturality of } \alpha \end{aligned}$$

Next, we need to check the axioms of guarded monads for $F_\#$.

- **(trv)** Let $f\colon X \to F_\# Y$. Then

$$\mathsf{out}(F_\# \,\mathsf{in}_1) f = (\mathsf{in}_1 \# (F_\# \,\mathsf{in}_1))\,\mathsf{out}\, f.$$

By **(trv)** for $\#$, $\mathsf{out}(F_\# \,\mathsf{in}_1)f$ is in_2-guarded, and thus, by definition so is $(F_\# \mathsf{in}_1)f$.

- **(cmp)** Let $f\colon X \to_2 F_\#(Y+Z)$, $g\colon Y \to_\sigma F_\# V$, $h\colon Z \to F_\# V$. Then we obtain

$$\mathsf{out}\,[g,h]^\star f = [\mathsf{out}\ g, \mathsf{out}\ h]^\star\,(\mathsf{id} \# [g,h]^\star)\,\mathsf{out}\, f.$$

By assumption $\mathsf{out}\, f$ is in_2-guarded, and therefore, since $\#$ is a parametrized guarded monad, so is $(\mathsf{id} \# [g,h]^\star)\,\mathsf{out}\, f$. Also, by assumption, $\mathsf{out}\, h$ is σ-guarded. By **(cmp)** for $\#$, this implies that the composite $[\mathsf{out}\ g, \mathsf{out}\ h]^\star\,(\mathsf{id} \# [g,h]^\star)\,\mathsf{out}\, f$ is σ-guarded and thus so is $[g,h]^\star f$.

- **(par)** Let $f_i\colon X_i \to_\sigma F_\# Y$ for $i = 1,2$, which by definition means that $\mathsf{out}\, f_i\colon X_i \to_\sigma Y \# (F_\# Y)$. By **(par)** for $\#$, $\mathsf{out}[f_1,f_2] = [\mathsf{out}\, f_1, \mathsf{out}\, f_2]\colon X_i \to_\sigma Y \# (F_\# Y)$, so that $[f_1,f_2]\colon X_1 + X_2 \to_\sigma F_\# Y$ as required.

This shows that $F_\#$ is indeed a guarded monad; it remains to show that given a parametrized guarded monad morphism $\alpha\colon \# \to \#'$ as above, the monad morphism F_α preserves guardedness. That is, for $f\colon Z \to_\sigma F_\# V$ we have to show that $F_\alpha f\colon Z \to_\sigma F_{\#'} V$, i.e. that $\mathsf{out}(F_\alpha f)$ is σ-guarded. Indeed, by definition of F_α,

$$\mathsf{out}\,(F_\alpha\, f) = (\mathsf{id} \# F_\alpha)\,\alpha\,\mathsf{out}\, f.$$

By assumption, $\mathsf{out}\, f$ is σ-guarded and therefore, since $\#$ is a parametrized guarded monad and α is a parametrized guarded monad morphism, so is $(\mathsf{id} \# F_\alpha)\,\alpha\ \mathsf{out}\, f$.

(2): Let $f\colon X \to_2 (Y+X) \#^\nu Z$, and let $g = \mathsf{out}^{-1}\,(\mathsf{in}_1 \# \mathsf{id})\,(\mathsf{out}\, f)^\dagger\colon X \to_2 (Y+X) \#^\nu Z$. Again, using the results of Uustalu [35, Theorem 3.11], $h = \mathsf{coit}([\eta, (\mathsf{out}\, f)^\dagger]^\star\,\mathsf{out})\colon (Y + X) \#^\nu Z \to_2 Y \#^\nu Z$ is the unique solution of equation

$$h = [\eta^\nu, h\, g]^{\bigstar},$$

which implies that $f^\ddagger = h\,\eta^\nu\,\mathsf{in}_2$ is a fixpoint of g. Indeed, $f^\ddagger = h\,\eta^\nu\,\mathsf{in}_2 = [\eta^\nu, h\, g]^{\bigstar}\eta^\nu\,\mathsf{in}_2 = h\, g$, and thus, $[\eta^\nu, f^\ddagger]^{\bigstar}\, g = [\eta^\nu, h\, g]^{\bigstar}\, g = h\, g = f^\ddagger$. We are left to check that $f^\ddagger$ is also a fixpoint of f. First, we record the auxiliary equation

$$\mathsf{out}[\eta^\nu, f^\ddagger]^{\bigstar} = [\eta, \mathsf{out}\, f^\ddagger]^\star(\mathsf{id} \# [\eta^\nu, f^\ddagger]^{\bigstar})\,\mathsf{out}, \tag{4.6}$$

which entails the goal as follows (using the fact that out is an isomorphism):

$$\begin{aligned} \mathsf{out}[\eta^\nu, f^\ddagger]^{\bigstar}\, f &= [\eta, \mathsf{out}\, f^\ddagger]^\star(\mathsf{id} \# [\eta^\nu, f^\ddagger]^{\bigstar})\,\mathsf{out}\, f && /\!\!/ \text{ (4.6)} \\ &= [\eta, \mathsf{out}[\eta^\nu, f^\ddagger]^{\bigstar}\, g]^\star(\mathsf{id} \# [\eta^\nu, f^\ddagger]^{\bigstar})\,\mathsf{out}\, f && /\!\!/ \text{ definition of } f^\ddagger \\ &= [\eta, [\eta, \mathsf{out}\, f^\ddagger]^\star(\mathsf{id} \# [\eta^\nu, f^\ddagger]^{\bigstar})\,\mathsf{out}\, g]^\star \\ &\qquad (\mathsf{id} \# [\eta^\nu, f^\ddagger]^{\bigstar})\,\mathsf{out}\, f && /\!\!/ \text{ (4.6)} \end{aligned}$$

$$
\begin{aligned}
&= [\eta, [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}(\mathsf{in}_1 \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})(\mathsf{out}\, f)^{\dagger}]^{\star} \\
&\qquad (\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, f && /\!\!/ \text{ definition of } g \\
&= [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}[\eta\, \mathsf{in}_1, (\mathsf{in}_1 \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})(\mathsf{out}\, f)^{\dagger}]^{\star} \\
&\qquad (\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, f \\
&= [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}(\mathsf{in}_1 \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})[\eta, (\mathsf{out}\, f)^{\dagger}]^{\star}\, \mathsf{out}\, f && /\!\!/ \text{ (4.1)} \\
&= [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}(\mathsf{in}_1 \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})(\mathsf{out}\, f)^{\dagger} && /\!\!/ \text{ definition of } (\text{–})^{\dagger} \\
&= [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}(\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, g && /\!\!/ \text{ definition of } g \\
&= \mathsf{out}[\eta^{\nu}, f^{\ddagger}]^{\bigstar}\, g && /\!\!/ \text{ (4.6)} \\
&= \mathsf{out}\, f^{\ddagger}. && /\!\!/ \text{ definition of } f^{\ddagger}
\end{aligned}
$$

Equation (4.6) is derived as follows:

$$
\begin{aligned}
\mathsf{out}[\eta^{\nu}, f^{\ddagger}]^{\bigstar} &= \big(\mathsf{id} \mathbin{\#} [[\eta^{\nu}, f^{\ddagger}]^{\bigstar}, \mathsf{id}]\big)\big((\mathsf{id} \mathbin{\#} \mathsf{in}_2)\, \mathsf{out}\, [\eta^{\nu}, f^{\ddagger}]\big)^{\star} \\
&\qquad (\mathsf{id} \mathbin{\#} \mathsf{in}_1)\, \mathsf{out} && /\!\!/ \text{ definition of } (\text{–})^{\bigstar} \\
&= \big((\mathsf{id} \mathbin{\#} [[\eta^{\nu}, f^{\ddagger}]^{\bigstar}, \mathsf{id}])(\mathsf{id} \mathbin{\#} \mathsf{in}_2)\, \mathsf{out}\, [\eta^{\nu}, f^{\ddagger}]\big)^{\star} \\
&\qquad (\mathsf{id} \mathbin{\#} [[\eta^{\nu}, f^{\ddagger}]^{\bigstar}, \mathsf{id}])\, (\mathsf{id} \mathbin{\#} \mathsf{in}_1)\, \mathsf{out} && /\!\!/ \text{ (4.1)} \\
&= (\mathsf{out}\, [\eta^{\nu}, f^{\ddagger}])^{\star}(\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out} \\
&= [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}(\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\,. && /\!\!/ \text{ definition of } \eta^{\nu}
\end{aligned}
$$

Property (4.2) transfers routinely along $\# \mapsto \#^{\nu}$.

(3): We have to show that, given $f\colon X \to_2 (Y + X) \#^{\nu} Z$ and $\hat{f}\colon X \to Y \#^{\nu} Z$ such that $\hat{f} = [\eta^{\nu}, \hat{f}]^{\bigstar} f$, we have $\hat{f} = f^{\ddagger}$, with $f^{\ddagger}$ defined as in Claim (2). Again, let $g = \mathsf{out}^{-1}(\mathsf{in}_1 \mathbin{\#} \mathsf{id})\, (\mathsf{out}\, f)^{\dagger}\colon X \to_2 (Y + X) \#^{\nu} Z$. As we indicated above, $f^{\ddagger}$ is the unique solution of the equation $[\eta^{\nu}, f^{\ddagger}]^{\bigstar}\, g = f^{\ddagger}$, and thus to obtain the desired identity $\hat{f} = f^{\ddagger}$, it suffices to prove the same equation for $\hat{f}$. Note that (4.6) remains valid for $\hat{f}$ instead of $f^{\ddagger}$ and therefore we obtain

$$\mathsf{out}\, \hat{f} = \mathsf{out}[\eta^{\nu}, \hat{f}]^{\bigstar} f = [\eta, \mathsf{out}\, \hat{f}]^{\star}(\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, f,$$

which implies $\mathsf{out}\, \hat{f} = ((\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, f)^{\dagger}$, for $(\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, f$ is in_2-guarded, and therefore has a unique fixpoint. Now, since

$$
\begin{aligned}
\mathsf{out}\, \hat{f} &= ((\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, f)^{\dagger} \\
&= (\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, (\mathsf{out}\, f)^{\dagger} && /\!\!/ \text{ (4.2)} \\
&= [\eta, \mathsf{out}\, f^{\ddagger}]^{\star}(\mathsf{id} \mathbin{\#} [\eta^{\nu}, f^{\ddagger}]^{\bigstar})\, \mathsf{out}\, \mathsf{out}^{-1}\, (\mathsf{in}_1 \mathbin{\#} \mathsf{id})\, (\mathsf{out}\, f)^{\dagger} \\
&= \mathsf{out}[\eta^{\nu}, f^{\ddagger}]^{\bigstar} g && /\!\!/ \text{ (4.6), definition of } g \\
&= \mathsf{out}\, f^{\ddagger}, && /\!\!/ \text{ definition of } f^{\ddagger}
\end{aligned}
$$

we obtain $f^{\ddagger} = \hat{f}$ using the fact that out is an isomorphism. □

Remark 4.4. The definitions figuring in Theorem 4.3 specialize to two generic cases occurring in previous literature:

(1) With $\mathbf{D} = 1$, $\# = T(\text{–} + \Sigma\text{–})$ for an endofunctor Σ and a totally guarded pre-iterative monad $\mathbb{T} = (T, \eta, \text{–}^\star, \text{–}^\dagger)$, we obtain the setting studied by Goncharov et al. [18]: $F_\#$ is isomorphically a monad $\mathbb{T}_\Sigma$ on $\mathbf{C}$ with $T_\Sigma X = \nu\gamma.\, T(X + \Sigma\gamma)$, unit $\eta^\nu = \mathsf{out}^{-1}\eta\,\mathsf{in}_1$, with Kleisli lifting $(f\colon X \to T_\Sigma Y)^{\bigstar}$ uniquely determined by the equation

$$\mathsf{out}\, f^{\bigstar} = [\mathsf{out}\, f,\, \eta\,\mathsf{in}_2\, \Sigma f^{\bigstar}]^\star\, \mathsf{out},$$

and with the total iteration operator

$$\big(f\colon X \to T_\Sigma(Y + X)\big)^\ddagger = \mathsf{coit}\big(\big[[\eta\,\mathsf{in}_1, (T[\mathsf{in}_1 + \mathsf{id}, \mathsf{in}_1\,\mathsf{in}_2]\,\mathsf{out}\, f)^\dagger], \eta\,\mathsf{in}_2\big]^\star\, \mathsf{out}\big)\, \eta^\nu\, \mathsf{in}_2\,.$$

(2) With $\mathbf{D} = 1$, and any vacuously guarded $\#\colon \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, we obtain the setting of Uustalu [35], with the guarded iterative monad $F_\# = \nu\gamma.\, \text{–} \# \gamma$ defined as follows: The monad structure is specified by (4.3), and the iteration operator $(\text{–})^\ddagger$ is uniquely determined by the equation $[\eta^\nu, f^\ddagger]^{\bigstar} = f^\ddagger$ for every in_2-guarded $f\colon X \to F_\#(Y + X)$. According to Theorem 4.3 (2), f is in_2-guarded iff $\mathsf{out}\, f\colon X \to (Y + X) \# F_\#(Y + X)$ factors through $\mathsf{in}_1 \# \mathsf{id}\colon Y \# F_\#(Y + X) \to (Y + X) \# F_\#(Y + X)$, which is precisely the notion of guardedness in [35].

Example 4.5. We proceed to illustrate the use of Theorem 4.3 by various instances of Example 3.12.

(1) By equipping the finite powerset monad $\mathcal{P}_\omega$ on **Set** with vacuous guardedness, we obtain by Theorem 4.3 a notion of guardedness for $\nu\gamma.\, \mathcal{P}_\omega(X + A \times \gamma)$, which allows for systems consisting of equations of the form

$$x = y_1 + \ldots + y_n + a_1.\, t_1 + \ldots + a_m.\, t_m \tag{4.7}$$

where the variables y_i are not allowed to occur on the left-hand side and the terms t_i represent elements of $\nu\gamma.\, \mathcal{P}_\omega(X + A \times \gamma)$, as previously explained in Example 3.12. By Proposition 3.11 and Theorem 4.3, we conclude that these systems have unique solutions, which is of course a known fact in process algebra. This and the following examples are intentionally chosen to be simple for illustrative purposes, but we emphasize that the same principles apply to the examples obtained by replacing $\mathcal{P}_\omega$ with a more general T and $A \times \text{–}$ with a more general Σ. For example, replacing T with a *subdistribution monad* [20], to model probability instead on nondeterminism, would require changing the format of (4.7) to

$$x = p_1 \cdot y_1 + \ldots + p_n \cdot y_n + p'_1 \cdot a_1.\, t_1 + \ldots + p'_m \cdot a_m.\, t_m$$

where the non-negative real coefficients $p_1, \ldots, p_n, p'_1, \ldots, p'_m$, subject to the condition $p_1 + \ldots + p_n + p'_1 + \ldots + p'_m \leqslant 1$, represent the probabilities of choosing the corresponding alternative.

(2) By replacing $\mathcal{P}_\omega$ with countable powerset $\mathcal{P}_{\omega_1}$ in the previous clause, we can relax the format of equation systems that can be solved, at the price of losing uniqueness of solutions. Specifically, let $\mathcal{P}_{\omega_1}$ be totally guarded pre-iterative with solutions of $f\colon X \to \mathcal{P}_{\omega_1}(Y + X)$ calculated via least fixpoints. The derived notion of guardedness for $\nu\gamma.\, \mathcal{P}_{\omega_1}(X + A \times \gamma)$ according to Theorem 4.3 is again total, i.e. allows solving arbitrary systems of equations (we discuss an application of such unguarded recursive process definitions in [18, Section 3]; specifically, they allow defining countably branching systems in basic process algebra). The canonical derived iteration operator makes use of both least fixpoints and unique coalgebraic fixpoints. For example, the canonical solution of

$$x = x + a.\, x \tag{4.8}$$

is the infinite sequence $x = a^\omega$, seen as an element of the final countably branching labelled transition system $\nu\gamma.\,\mathcal{P}_{\omega_1}(A\times\gamma)$ – intuitively, the original system (4.8) is first collapsed to $x = a.\,x$, iterating away the first x in the sum by taking a least fixpoint, and the resulting system is solved uniquely. In detail, the definitions in Theorem 4.3 unfold as follows. We have the case mentioned in Example 4.2 where $X \mathbin{\#} Y = \mathcal{P}_{\omega_1}(X + A \times Y)$ (so $F_{\#}X = \nu\gamma.\mathcal{P}_{\omega_1}(X + A \times \gamma)$). Our example equation (4.8) corresponds to the map $f\colon X \to F_{\#}(\emptyset + X)$ where $X = \{x\}$ and $\mathsf{out}\, f(x) = \{\mathsf{in}_1\, \mathsf{in}_2\, x, \mathsf{in}_2\langle a, \mathsf{out}^{-1}(\{x\})\rangle\} \in \mathcal{P}_{\omega_1}((\emptyset + X) + A \times F_{\#}(\emptyset + X)) \cong \mathcal{P}_{\omega_1}(A \times F_{\#}X + X)$. The definition of $(-)^\ddagger$ according to Theorem 4.3 now tells us to first iterate $\mathsf{out}\, f$ in $\mathcal{P}_{\omega_1}$, by taking a least fixpoint with x seen as a variable, obtaining $(\mathsf{out}\, f)^\dagger\colon X \to \mathcal{P}_{\omega_1}(A \times F_{\#}X)$ where

$$(\mathsf{out}\, f)^\dagger(x) = \{\langle a, \mathsf{out}^{-1}(\{\mathsf{in}_1\, x\})\rangle\}.$$

We next form the map $g = (\mathcal{P}_{\omega_1}\, \mathsf{in}_2)[(\mathsf{out}\, f)^\dagger, \eta]^\star\, \mathsf{out}\colon F_{\#}X \to \mathcal{P}_{\omega_1}(\emptyset + A \times F_{\#}X)$, where η and $(\text{–})^\star$ are the unit and the Kleisli lifting of $\mathcal{P}_{\omega_1}$, so for $t \in F_{\#}X = \nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma)$,

$$\begin{aligned} g(t) &= \{\mathsf{in}_2(\mathsf{out}\, f)^\dagger(x) \mid \mathsf{in}_1\, x \in \mathsf{out}\, t\} \cup \{\mathsf{in}_2\langle b, s\rangle \in A \times F_{\#}X \mid \mathsf{in}_2\langle b, s\rangle \in \mathsf{out}\, t\} \\ &= \{\mathsf{in}_2\langle a, \mathsf{out}^{-1}(\{x\})\rangle \mid \mathsf{in}_1\, x \in \mathsf{out}\, t\} \cup \{\mathsf{in}_2\langle b, s\rangle \in A \times F_{\#}X \mid \mathsf{in}_2\langle b, s\rangle \in \mathsf{out}\, t\}. \end{aligned}$$

We then obtain a final coalgebra morphism $\mathsf{coit}\, g\colon F_{\#}X \to F_{\#}\emptyset = \nu\gamma.\,\mathcal{P}_{\omega_1}(A \times \gamma)$. The solution $f^\ddagger(x)$ is obtained by applying $\mathsf{coit}\, g$ to $\eta^\nu_X(x) = \mathsf{out}^{-1}(\eta_{X, F_{\#}X}(x)) = \mathsf{out}^{-1}(\{\mathsf{in}_1\, x\})$, using the description of η^ν recalled in the proof of Theorem 4.3. Since $g(\mathsf{out}^{-1}(\{\mathsf{in}_1\, x\})) = \{\mathsf{in}_2\langle a, \mathsf{out}^{-1}(\{\mathsf{in}_1\, x\})\rangle\}$, we obtain that $f^\ddagger(x)$ is a^ω, as expected.

(3) Consider a further variation of the same example obtained by replacing A in the previous example by $1 + A$, where the adjoined element is supposed to capture the invisible action τ in the usual sense of process algebra [25]. Applying Theorem 4.3 to $X \mathbin{\#} Y = \mathcal{P}_{\omega_1}(X + (1 + A) \times Y)$ as in the previous example, we would derive a notion of guardedness that identifies as guarded any recursive call preceded by an action, visible or not. We can refine this view by allowing only visible actions as guards, which is in fact standard for CCS [25]. To this end, consider the obvious isomorphism

$$\nu\gamma.\,\mathcal{P}_{\omega_1}(X + (1 + A) \times \gamma) \cong \nu\gamma'.\,\nu\gamma.\,\mathcal{P}_{\omega_1}(X + \gamma + A \times \gamma'),$$

which involves two more parametrized monads: $\mathcal{P}_{\omega_1}(\text{–} + \text{–} + A \times \text{–})\colon \mathbf{Set} \times (\mathbf{Set} \times \mathbf{Set}) \to \mathbf{Set}$ and $\nu\gamma.\,\mathcal{P}_{\omega_1}(\text{–} + \gamma + A \times \text{–})\colon \mathbf{Set} \times \mathbf{Set} \to \mathbf{Set}$. The latter parametrized monad is formed on top of the former. We equip $\nu\gamma.\,\mathcal{P}_{\omega_1}(\text{–} + \gamma + A \times \text{–})$ with the vacuous notion of guardedness. By furthermore forming the fixpoint $\nu\gamma'.\,\nu\gamma.\,\mathcal{P}_{\omega_1}(X + \gamma + A \times \gamma')$, we obtain precisely the notion of guardedness we aimed at for the isomorphic monad $\#^\nu$.

(4) Consider $TX = \mathcal{P}_{\omega_1}(\mu\gamma.\, X + 1 + A \times \gamma)$, which can be understood as a semantic domain for processes with results in X as before, but now modulo *finite trace equivalence* instead of strong bisimilarity as the underlying equivalence relation: the elements of TX are sets of traces from $\mu\gamma.\, X + 1 + A \times \gamma \cong A^\star + A^\star \times X$ consisting of terminating traces (from $A^\star \times X$) and non-terminating traces (from $A^\star$). In order to apply our theory to this example, we make use of Hasuo et al.'s results on *coalgebraic finite trace semantics* [20]. Specifically, we make use of the fact that due to presence of a canonical distributive law

$$X + 1 + A \times \mathcal{P}_{\omega_1} \to \mathcal{P}_{\omega_1}(X + 1 + A \times \text{–})$$

and a suitable order-enrichment of $\mathcal{P}_{\omega_1}$, the object $\mu\gamma.\, X + 1 + A \times \gamma$ computed in $\mathbf{Set}$ carries a final coalgebra $\nu\gamma.\, X + 1 + A \times \gamma$ in the Kleisli category of $\mathcal{P}_{\omega_1}$. In this category we equip the parametrized monad $\# = \text{–} + 1 + A \times \text{–}$ with the vacuous notion of guardedness and thus derive the notion of guardedness for $\#^\nu$, allowing exactly for recursive calls preceded

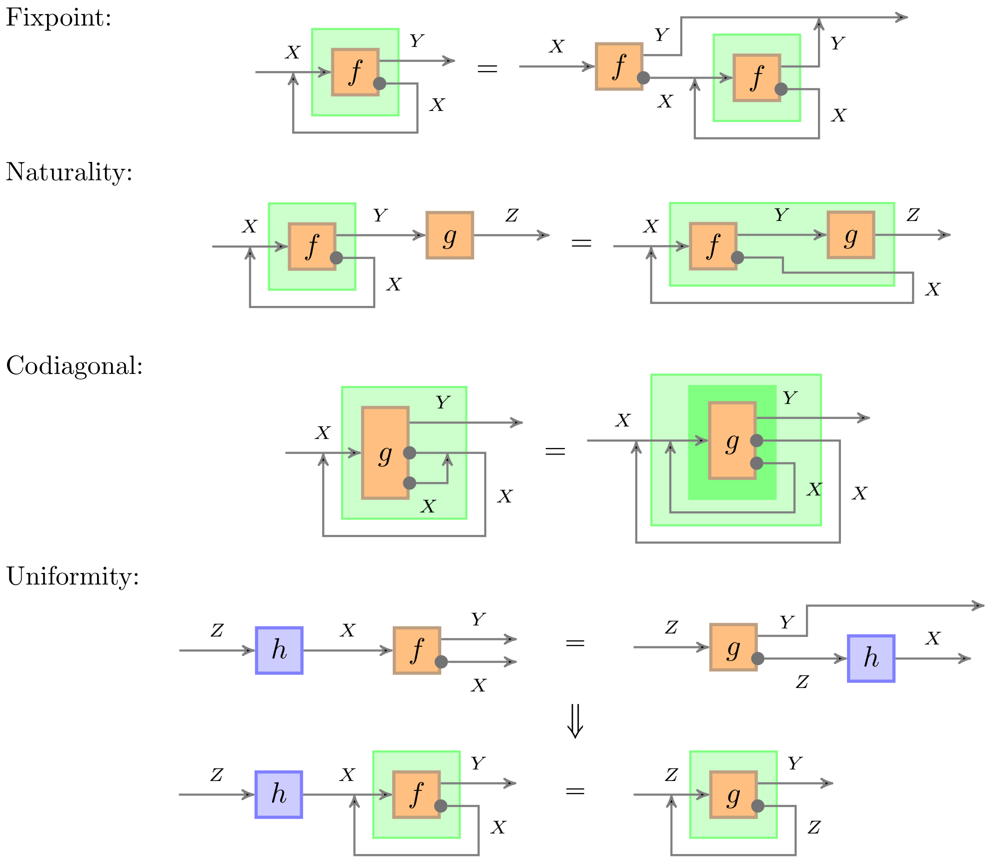

FIGURE 2. Axioms of guarded iteration.

by actions from A. Again, by Proposition 3.11 and by Theorem 4.3 (3), the obtained monad is guarded iterative.

Note that the monad $\mathcal{P}_{\omega_1}(\mu\gamma.\,\text{--}+1+A\times\gamma)$ is arguably too large, as it contains sets of traces not realized by any process from $\nu\gamma.\,\mathcal{P}_{\omega_1}(X+A\times\gamma)$. This can easily be fixed by cutting down to the submonad of $\mathcal{P}_{\omega_1}(\mu\gamma.\,\text{--}+1+A\times\gamma)$ consisting of the *prefix-closed* sets of traces, i.e. such sets S that $st\in S$ implies $s\in S$ and $\langle st,x\rangle\in S$ implies $s\in S$. It is easy to see that this is a guarded pre-iterative submonad of $\mathbb{T}$, and therefore guarded iterative.

5. Complete Elgot Monads and Iteration Congruences

Besides the fixpoint identity we are interested in natural guarded versions of the classical properties of the iteration operator, which we refer to as the *iteration laws* [10, 6, 32]:

- *naturality:* $g^\star f^\dagger = ([(T\,\mathsf{in}_1)\,g, \eta\,\mathsf{in}_2]^\star f)^\dagger$ for $f\colon X\to_2 T(Y+X)$, $g\colon Y\to TZ$;
- *codiagonal:* $(T[\mathsf{id},\mathsf{in}_2]\,f)^\dagger = f^{\dagger\dagger}$ for $f\colon X\to_{12,2} T((Y+X)+X)$;
- *uniformity:* $f\,h = T(\mathsf{id}+h)\,g$ implies $f^\dagger h = g^\dagger$ for $f\colon X\to_2 T(Y+X)$, $g\colon Z\to_2 T(Y+Z)$ and $h\colon Z\to X$.

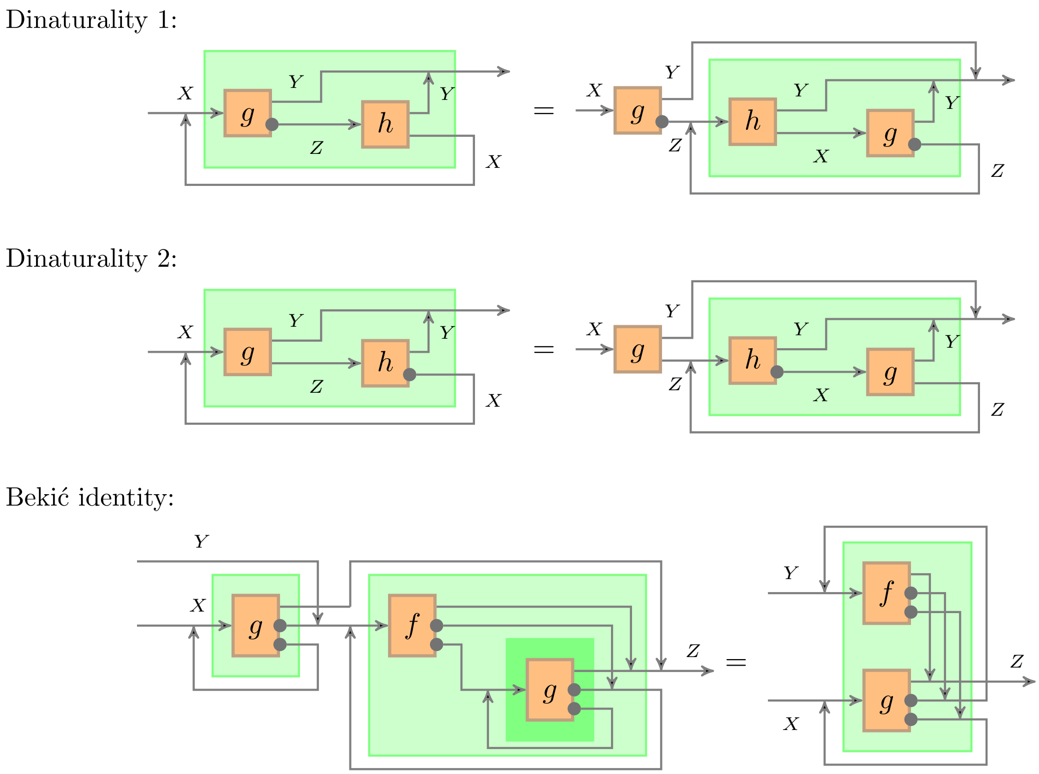

FIGURE 3. Derivable laws of iteration.

Remarkably, this list does not include the well-known *dinaturality law*, as is turns out to be derivable (cf. [18, 14]). We prove this further below. The above axioms are summarized in graphical form in Figure 2, and then become quite intuitive. We indicate the scope of the iteration operator by a shaded box and guardedness by bullets at the outputs of a morphism. Blue boxes indicate morphisms of the base category **C**, to contrast orange boxes referring to Kleisli morphisms.

A guarded pre-iterative monad is called a *complete Elgot monad* if it is totally guarded and satisfies all iteration laws. In the sequel we shorten 'complete Elgot monads' to 'Elgot monads' (to be distinguished from Elgot monads in the sense of [2], which have solutions only for morphisms with finitely presentable domain).

In general, the fact that the iteration laws are correctly formulated relies on the axioms of guardedness. For example, in the codiagonal axiom, this follows by **(cmp)** from the assumption $f\colon X \to_{12,2} T((Y+X)+X)$ that $T[\mathsf{id},\mathsf{in}_2]\,f$ is in_2-guarded, and by Proposition 3.9 that $f^\dagger$ is in_2-guarded. Indeed, the axioms for guarded monads are designed precisely to enable the formulation of the iteration laws.

We show next that for guarded *iterative* monads, all iteration laws are automatic. In preparation, we prove the aforementioned fact that dinaturality follows from the other axioms (thus generalizing corresponding recent observations on iteration theories [18, 14]). Additionally, we show that the well-known Bekić identity is derivable too.

Proposition 5.1. *Any guarded pre-iterative monad satisfying naturality, codiagonal and uniformity also satisfies*

- dinaturality: $([\eta\,\mathsf{in}_1, h]^\star\, g)^\dagger = [\eta, ([\eta\,\mathsf{in}_1, g]^\star\, h)^\dagger]^\star\, g$ *for* $g\colon X \to_2 T(Y+Z)$ *and* $h\colon Z \to T(Y+X)$ *or* $g\colon X \to T(Y+Z)$ *and* $h\colon Z \to_2 T(Y+X)$*;*
- Bekić identity: $(T[\mathsf{id}+\mathsf{in}_1, \mathsf{in}_2\ \mathsf{in}_2]\,[f,g])^\dagger = [h^\dagger, [\eta, h^\dagger]^\star g^\dagger]$ *with* $f\colon X \to_{12,2} T((Y+X)+Z)$*,* $g\colon Z \to_{12,2} T((Y+X)+Z)$*, and* $h = [\eta, g^\dagger]^\star f\colon X \to_2 T(Y+X)$*.*

In axiomatizations of total iteration, the Bekić identity is sometimes taken to replace codiagonal and dinaturality [6, Section 6.8] [2, 18]. Both dinaturality and the Bekić identity are again depicted graphically in Figure 3. The two versions of the dinaturality axiom correspond to the alternative sets of guardedness assumptions in its formulation; basically, we need to distinguish cases on whether the loop over g and h is guarded at g or at h.

Proof. Following [14], we consider a specific instance of uniformity:

$$f^\dagger = ([T(\mathsf{id}+\mathsf{in}_1)f, h] : X+Z \to_2 T(Y+(X+Z)))^\dagger\,\mathsf{in}_1 \tag{5.1}$$

where $f\colon X \to_2 T(Y+X)$ and $h\colon Z \to_2 T(Y+(X+Z))$, and prove the following instance of the Bekić identity:

$$(T(\mathsf{id}+\mathsf{in}_1)\,[f,g]\colon X+Z \to T(Y+(X+Z)))^\dagger = [f^\dagger, [\eta, f^\dagger]^\star g] \tag{5.2}$$

where $f\colon X \to_2 T(Y+X)$ and $g\colon Z \to_2 T(Y+X)$. Indeed, on the one hand, by (5.1),

$$(T(\mathsf{id}+\mathsf{in}_1)\,[f,g])^\dagger\,\mathsf{in}_1 = f^\dagger, \tag{5.3}$$

and on the other hand

$$\begin{aligned}
&(T(\mathsf{id}+\mathsf{in}_1)\,[f,g])^\dagger\,\mathsf{in}_2\\
&\quad= [\eta, (T(\mathsf{id}+\mathsf{in}_1)\,[f,g])^\dagger]^\star\, T(\mathsf{id}+\mathsf{in}_1)[f,g]\,\mathsf{in}_2 && /\!\!/ \text{ fixpoint}\\
&\quad= [\eta, (T(\mathsf{id}+\mathsf{in}_1)\,[f,g])^\dagger]^\star\, T(\mathsf{id}+\mathsf{in}_1)g\\
&\quad= [\eta, (T(\mathsf{id}+\mathsf{in}_1)\,[f,g])^\dagger\,\mathsf{in}_1]^\star\, g\\
&\quad= [\eta, f^\dagger]^\star\, g. && /\!\!/ \text{ (5.3)}
\end{aligned}$$

As the result we obtain (5.2). Analogously, we prove another instance of the Bekić identity, namely

$$(T(\mathsf{id}+\mathsf{in}_2)\,[f,g]\colon X+Z \to T(Y+(X+Z)))^\dagger = [[\eta, g^\dagger]^\star f, g^\dagger] \tag{5.4}$$

with $f\colon X \to_2 T(Y+Z)$ and $g\colon Z \to_2 T(Y+Z)$. We proceed to show that under the other axioms, these two instances imply the full Bekić identity

$$(T[\mathsf{id}+\mathsf{in}_1, \mathsf{in}_2\ \mathsf{in}_2]\,[f,g]\colon X+Z \to T(Y+(X+Z)))^\dagger = [h^\dagger, [\eta, h^\dagger]^\star g^\dagger] \tag{5.5}$$

where $f\colon X \to_{12,2} T((Y+X)+Z)$ and $g\colon Z \to_{12,2} T((Y+X)+Z)$ and $h = [\eta, g^\dagger]^\star f\colon X \to T(Y+X)$. Let us argue briefly that h is in_2-guarded. Note that the assumption for f by **(iso)** implies that $f' = T[\mathsf{id}+\mathsf{in}_1, \mathsf{in}_2]\, f\colon X \to T(Y+(X+Z))$ is in_2-guarded and therefore $h = [\eta\,\mathsf{in}_1, [\eta\,\mathsf{in}_1, g^\dagger]]^\star f'$ is in_2-guarded by **(cmp)**.

Now, the proof of (5.5) runs as follows:

$$\begin{aligned}
&(T[\mathsf{id}{+}\,\mathsf{in}_1, \mathsf{in}_2\ \mathsf{in}_2]\,[f,g])^\dagger\\
&\quad= \big(T[\mathsf{id}, \mathsf{in}_2]\,T((\mathsf{id}+\mathsf{in}_1)+\mathsf{in}_2)\,[f,g]\big)^\dagger\\
&\quad= (T((\mathsf{id}+\mathsf{in}_1)+\mathsf{in}_2)\,[f,g])^{\dagger\dagger} && /\!\!/ \text{ codiagonal}
\end{aligned}$$

$$
\begin{aligned}
&= (T(\mathsf{id}+\mathsf{in}_2)\,[T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,f, T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,g])^{\dagger\dagger}\\
&= [[\eta,(T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,g)^{\dagger}]^{\star}T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,f,\\
&\qquad (T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,g)^{\dagger}]^{\dagger} && /\!/\ (5.4)\\
&= [[\eta(\mathsf{id}+\mathsf{in}_1),(T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,g)^{\dagger}]^{\star}f,\\
&\qquad (T((\mathsf{id}+\mathsf{in}_1)+\mathsf{id})\,g)^{\dagger}]^{\dagger} && /\!/\ \text{naturality}\\
&= [T(\mathsf{id}+\mathsf{in}_1)\,[\eta,g^{\dagger}]^{\star}\,f, T(\mathsf{id}+\mathsf{in}_1)\,g^{\dagger}]^{\dagger}\\
&= (T(\mathsf{id}+\mathsf{in}_1)\,[h,g^{\dagger}])^{\dagger}\\
&= [h^{\dagger},[\eta,h^{\dagger}]^{\star}g^{\dagger}]. && /\!/\ (5.2)
\end{aligned}
$$

Finally, let us derive dinaturality from (5.5). Suppose that $g\colon X \to_2 T(Y+Z)$ and $h\colon Z \to T(Y+X)$ satisfy either guardedness premise of the dinaturality axiom and consider the following instance of (5.5) with f replaced by $T(\mathsf{in}_1 + \mathsf{id})\,g$ and g replaced by $(T\,\mathsf{in}_1)\,h$ (note that by the fixpoint identity, $((T\,\mathsf{in}_1)\,h)^{\dagger} = h$):

$$[T(\mathsf{id}+\mathsf{in}_2)\,g, T(\mathsf{id}+\mathsf{in}_1)\,h]^{\dagger} = [([\eta\,\mathsf{in}_1,h]^{\star}g)^{\dagger},[\eta,([\eta\,\mathsf{in}_1,h]^{\star}g)^{\dagger}]^{\star}h]. \tag{5.6}$$

Let $\gamma_{X,Y}: Y+X \to Y+X$ be the obvious symmetry transformation and note the following simple consequence of uniformity:

$$[T(\mathsf{id}+\mathsf{in}_2)\,g, T(\mathsf{id}+\mathsf{in}_1)\,h]^{\dagger}\gamma = [T(\mathsf{id}+\mathsf{in}_2)\,h, T(\mathsf{id}+\mathsf{in}_1)\,g]^{\dagger}. \tag{5.7}$$

By combining (5.7), (5.6) and the symmetric form of the latter (with h and g switched), we obtain:

$$[([\eta\,\mathsf{in}_1,h]^{\star}g)^{\dagger},[\eta,([\eta\,\mathsf{in}_1,h]^{\star}g)^{\dagger}]^{\star}h] = [[\eta,([\eta\,\mathsf{in}_1,g]^{\star}h)^{\dagger}]^{\star}g,([\eta\,\mathsf{in}_1,g]^{\star}h)^{\dagger}].$$

Dinaturality is now obtained by composing both sides with $\mathsf{in}_1\colon X \to X+Z$. □

The proof of the following result runs in accordance with the original ideas of Elgot for iterative theories [10], except that, by Proposition 5.1, dinaturality is now replaced with uniformity.

Theorem 5.2. *Every guarded iterative monad validates naturality, dinaturality, codiagonal, and uniformity.*

Proof. By Proposition 5.1 we only need to verify naturality, codiagonal and uniformity.

- *Naturality.* Let $f\colon X \to_2 T(Y+X)$ and $g\colon Y \to TZ$. Then

$$
\begin{aligned}
g^{\star}f^{\dagger} &= g^{\star}[\eta,f^{\dagger}]^{\star}f\\
&= [g^{\star}\eta,g^{\star}f^{\dagger}]^{\star}f\\
&= [g,g^{\star}f^{\dagger}]^{\star}f\\
&= [\eta,g^{\star}f^{\dagger}]^{\star}[(T\,\mathsf{in}_1)\,g,\eta\,\mathsf{in}_2]^{\star}f.
\end{aligned}
$$

Since the same equation uniquely characterizes $([(T\,\mathsf{in}_1)\,g,\eta\,\mathsf{in}_2]^{\star}\,f)^{\dagger}$, the latter is equal to $g^{\star}f^{\dagger}$.

- *Codiagonal.* Let $f\colon X \to_{12,2} T((Y+X)+X)$. Then

$$
\begin{aligned}
f^{\dagger\dagger} &= [\eta,f^{\dagger\dagger}]^{\star}f^{\dagger}\\
&= [\eta,f^{\dagger\dagger}]^{\star}[\eta,f^{\dagger}]^{\star}f
\end{aligned}
$$

$$= [[\eta, f^{\dagger\dagger}], [\eta, f^{\dagger\dagger}]^\star f^\dagger]^\star f$$
$$= [[\eta, f^{\dagger\dagger}], f^{\dagger\dagger}]^\star f$$
$$= [\eta, f^{\dagger\dagger}]^\star T[\mathsf{in}_2, \mathsf{id}]\, f.$$

Therefore $f^{\dagger\dagger}$ satisfies the fixpoint identity for $(T[\mathsf{in}_2, \mathsf{id}]\, f)^\dagger$, and thus $f^{\dagger\dagger} = (T[\mathsf{in}_2, \mathsf{id}]\, f)^\dagger$.

- *Uniformity.* Suppose that $f\, h = T(\mathsf{id} + h)\, g$ for some $f\colon X \to_2 T(Y + X)$, $g\colon Z \to_2 T(Y + Z)$ and $h\colon Z \to X$. Then

$$f^\dagger h = [\eta, f^\dagger]^\star f\, h = [\eta, f^\dagger]^\star T(\mathsf{id} + h)\, g = [\eta, f^\dagger h]^\star g,$$

that is, $f^\dagger h$ satisfies the fixpoint equation for $g^\dagger$. Hence $g^\dagger = f^\dagger h$. □

We now proceed to introduce key properties of morphisms of guarded monads that allow for transferring pre-iterativity and the iteration laws, respectively.

Definition 5.3 (Guarded retraction)**.** Let $\mathbb{T}$ and $\mathbb{S}$ be guarded monads. We call a monad morphism $\rho\colon \mathbb{T} \to \mathbb{S}$ a *guarded retraction* if there is a family of morphisms $(\upsilon_X\colon SX \to TX)_{X\in|\mathbf{C}|}$ (not necessarily natural in X!) such that
(1) for every $f\colon X \to_\sigma SY$, we have $\upsilon_Y f\colon X \to_\sigma TY$, and
(2) $\rho_X \upsilon_X = \mathsf{id}$ for all $X \in |\mathbf{C}|$.

Theorem 5.4. *Let $\rho\colon \mathbb{T} \to \mathbb{S}$ be a guarded retraction, witnessed by $\upsilon\colon \mathbb{S} \to \mathbb{T}$, and suppose that $(\mathbb{T}, \text{–}^\dagger)$ is guarded pre-iterative. Then $\mathbb{S}$ is guarded pre-iterative with the iteration operator $(\text{–})^\ddagger$ given by $f^\ddagger = \rho\,(\upsilon f)^\dagger$.*

Proof. Since $\mathbb{T}$ satisfies the fixpoint identity, $[\eta, (\upsilon f)^\dagger]^\star \upsilon f = (\upsilon f)^\dagger$ and therefore,

$$\begin{aligned} f^\ddagger &= \rho\,(\upsilon f)^\dagger \\ &= \rho\,[\eta, (\upsilon f)^\dagger]^\star \upsilon f \\ &= [\rho\eta, \rho(\upsilon f)^\dagger]^\star \rho\upsilon f && /\!/\ \rho \text{ is a monad morphism} \\ &= [\eta, f^\ddagger]^\star f. && /\!/\ \rho\upsilon = \mathsf{id} \end{aligned}$$

□

Definition 5.5 (Iteration congruence)**.** Let $\mathbb{T}$ be a guarded pre-iterative monad and let $\mathbb{S}$ be a monad. We call a monad morphism $\rho\colon \mathbb{T} \to \mathbb{S}$ an *iteration congruence* if for every pair of morphisms $f, g\colon X \to_2 T(Y + X)$,

$$\rho f = \rho g \qquad \text{implies} \qquad \rho f^\dagger = \rho g^\dagger. \tag{5.8}$$

If ρ is moreover a guarded retraction, we call ρ an *iteration-congruent retraction.*

Theorem 5.6. *Under the premises of Theorem 5.4, assume moreover that ρ is an iteration-congruent retraction. Then any property out of naturality, dinaturality, codiagonal, and uniformity that is satisfied by $\mathbb{T}$ is also satisfied by $\mathbb{S}$.*

Proof. The crucial observation is that under our assumptions, (5.8) is equivalent to the condition that for all $f\colon X \to_2 T(Y + X)$,

$$\rho\,(\upsilon\rho\, f)^\dagger = \rho f^\dagger. \tag{5.9}$$

Indeed, (5.8) $\implies$ (5.9), for $\rho\upsilon\rho\, f = \rho\, f$ and therefore $\rho(\upsilon\rho\, f)^\dagger = \rho\, f^\dagger$ and conversely, assuming (5.9) both for f and for g, and $\rho f = \rho g$, we obtain that $\rho f^\dagger = \rho(\upsilon\rho\, f)^\dagger = \rho(\upsilon\rho\, g)^\dagger = \rho g^\dagger$. The proof of transfer of the respective properties then proceeds as follows.

- *Naturality:*

$$
\begin{aligned}
g^\star f^\ddagger &= g^\star \rho\,(\upsilon\, f)^\dagger \\
&= (\rho\upsilon\, g)^\star\, \rho\,(\upsilon\, f)^\dagger && /\!/\ \rho\upsilon = \mathsf{id} \\
&= \rho\,(\upsilon\, g)^\star\,(\upsilon\, f)^\dagger && /\!/\ \rho \text{ is a monad morphism} \\
&= \rho\,([(T\,\mathsf{in}_1)\,\upsilon\, g, \eta\,\mathsf{in}_2]^\star \upsilon\, f)^\dagger && /\!/\ \text{naturality for } (\text{–})^\dagger \\
&= \rho\,(\upsilon\rho\,[(T\,\mathsf{in}_1)\,\upsilon\, g, \eta\,\mathsf{in}_2]^\star \upsilon\, f)^\dagger && /\!/\ (5.9) \\
&= \rho\,(\upsilon[\rho\,(T\,\mathsf{in}_1)\,\upsilon\, g, \eta\,\mathsf{in}_2]^\star \rho\,\upsilon\, f)^\dagger && /\!/\ \rho \text{ is a monad morphism} \\
&= \rho\,(\upsilon[(S\,\mathsf{in}_1)\ g, \eta\,\mathsf{in}_2]^\star\, f)^\dagger && /\!/\ \rho\upsilon = \mathsf{id} \\
&= ([(S\,\mathsf{in}_1)\ g, \eta\,\mathsf{in}_2]^\star f)^\ddagger.
\end{aligned}
$$

- *Dinaturality:* First observe that it follows from the fact that ρ is a monad morphism and $\rho\upsilon = \mathsf{id}$ that $\rho\upsilon[\eta\,\mathsf{in}_1, h]^\star\, g = \rho[\eta\,\mathsf{in}_1, \upsilon h]^\star \upsilon g$, and therefore, by (5.8), that

$$
\rho(\upsilon[\eta\,\mathsf{in}_1, h]^\star\, g)^\dagger = \rho([\eta\,\mathsf{in}_1, \upsilon h]^\star \upsilon g)^\dagger. \tag{5.10}
$$

Then we obtain the goal as follows:

$$
\begin{aligned}
([\eta\,\mathsf{in}_1, h]^\star\, g)^\ddagger &= \rho(\upsilon[\eta\,\mathsf{in}_1, h]^\star\, g)^\dagger \\
&= \rho([\eta\,\mathsf{in}_1, \upsilon h]^\star \upsilon g)^\dagger && /\!/\ (5.10) \\
&= \rho[\eta, ([\eta\,\mathsf{in}_1, \upsilon g]^\star\, \upsilon h)^\dagger]^\star\, \upsilon g && /\!/\ \text{dinaturality for } (\text{–})^\dagger \\
&= [\eta, \rho([\eta\,\mathsf{in}_1, \upsilon g]^\star\, \upsilon h)^\dagger]^\star \rho\upsilon g && /\!/\ \rho \text{ is a monad morphism} \\
&= [\eta, \rho([\eta\,\mathsf{in}_1, \upsilon g]^\star\, \upsilon h)^\dagger]^\star g && /\!/\ \text{since } \rho\upsilon = \mathsf{id} \\
&= [\eta, \rho(\upsilon[\eta\,\mathsf{in}_1, g]^\star h)^\dagger]^\star g && /\!/\ \text{analogous to } (5.10) \\
&= [\eta, ([\eta\,\mathsf{in}_1, g]^\star h)^\ddagger]^\star g.
\end{aligned}
$$

- *Codiagonal:*

$$
\begin{aligned}
(f^\ddagger)^\ddagger &= \rho\,(\upsilon\rho(\upsilon\, f)^\dagger)^\dagger \\
&= \rho\,((\upsilon\, f)^\dagger)^\dagger && /\!/\ (5.9) \\
&= \rho\,(T[\mathsf{id}, \mathsf{in}_2]\,\upsilon\, f)^\dagger && /\!/\ \text{codiagonal for } (\text{–})^\dagger \\
&= \rho\,(\upsilon\rho\, T[\mathsf{id}, \mathsf{in}_2]\,\upsilon\, f)^\dagger && /\!/\ (5.9) \\
&= \rho\,(\upsilon\, S[\mathsf{id}, \mathsf{in}_2]\ f)^\dagger && /\!/\ \rho\upsilon = \mathsf{id} \\
&= (S[\mathsf{id}, \mathsf{in}_2]\, f)^\ddagger.
\end{aligned}
$$

- *Uniformity:* Suppose that $f\, h = S(\mathsf{id} + h)\, g$. Then

$$
\begin{aligned}
\upsilon\, f h &= \upsilon\, S(\mathsf{id} + h)\, g \\
&= \upsilon\, S(\mathsf{id} + h)\, \rho\upsilon\, g \\
&= T(\mathsf{id} + h)\, \upsilon\, g
\end{aligned}
$$

and therefore $(\upsilon\, f)^\dagger\, h = (\upsilon\, g)^\dagger$. This implies $f^\ddagger\, h = g^\ddagger$ by definition. □

Recall from the introduction that a monad $\mathbb{S}$ is *iteratable* if its coinductive resumption transform $\mathbb{S}^\nu$ exists. We make $\mathbb{S}^\nu$ into a guarded monad by applying Theorem 4.3 to $\mathbb{S}$ as a vacuously guarded monad; explicitly: $f\colon X \to S^\nu(Y+X)$ is guarded iff

$$\mathsf{out}\, f = S(\mathsf{in}_1 + \mathsf{id})\, g \qquad \text{for some} \qquad g\colon X \to S(Y + S^\nu(Y+X)).$$

We are now set to prove our first main result, which states that every iteratable Elgot monad can be obtained by quotienting a guarded iterative monad; that is, every choice of solutions that obeys the iteration laws arises by quotienting a more fine-grained model in which solutions are uniquely determined:

Theorem 5.7. *A totally guarded iteratable monad $\mathbb{S}$ is an Elgot monad iff there is a guarded iterative monad $\mathbb{T}$ and an iteration-congruent retraction $\rho\colon \mathbb{T} \to \mathbb{S}$. Specifically, every iteratable Elgot monad $\mathbb{S}$ is an iteration-congruent retract of its coinductive resumption transform $\mathbb{S}^\nu$.*

Proof. 'If' is immediate by Theorems 5.6 and 5.2. We prove 'only if', i.e. that $\mathbb{S} = (S, \eta, {-}^\star, {-}^\dagger)$ is an iteration-congruent retract of $\mathbb{S}^\nu = (\nu\gamma.\, S({-}+\gamma), \eta^\nu, {-}^\star, {-}^\ddagger)$. We define $\upsilon_X = \mathsf{out}^{-1}\, \eta\, \mathsf{in}_2\, \mathsf{out}^{-1}(S\, \mathsf{in}_1)$ and

$$\rho_X = \big(S^\nu X \xrightarrow{\mathsf{out}} S(X + S^\nu X)\big)^\dagger.$$

Clearly, υf is σ-guarded for every $f\colon X \to SY$ and υ is left inverse to ρ, for

$$\begin{aligned}
\rho\upsilon &= \mathsf{out}^\dagger\, \mathsf{out}^{-1}\, \eta\, \mathsf{in}_2\, \mathsf{out}^{-1}(S\, \mathsf{in}_1) \\
&= [\eta, \rho]^\star\, \mathsf{out}\, \mathsf{out}^{-1}\, \eta\, \mathsf{in}_2\, \mathsf{out}^{-1}(S\, \mathsf{in}_1) && /\!/ \text{ fixpoint for } (-)^\dagger \\
&= \rho\, \mathsf{out}^{-1}(S\, \mathsf{in}_1) \\
&= [\eta, \rho]^\star\, \mathsf{out}\, \mathsf{out}^{-1}(S\, \mathsf{in}_1) && /\!/ \text{ fixpoint for } (-)^\dagger \\
&= \eta^\star \\
&= \mathsf{id}.
\end{aligned}$$

It follows straightforwardly by naturality of $(-)^\dagger$ that ρ is a natural transformation. Note the following property of ρ: for any $h\colon X \to S(Y+X)$, $\mathsf{out}(\mathsf{coit}\, h) = S(\mathsf{id} + \mathsf{coit}\, h)\, h$, and hence, by uniformity

$$\rho(\mathsf{coit}\, h) = h^\dagger. \tag{5.11}$$

Let us verify that ρ is a monad morphism. For one thing

$$\rho\eta^\nu = [\eta, \rho]^\star\, \mathsf{out}\, \eta^\nu = [\eta, \rho]^\star \eta\, \mathsf{in}_1 = \eta.$$

Next, we have to check that $\rho f^{\bigstar} = (\rho f)^\star \rho$ for any $f\colon X \to S^\nu Y$. Note that

$$f^{\bigstar} = \mathsf{coit}\big([[\eta\, \mathsf{in}_1, S(\mathsf{id} + S^\nu\, \mathsf{in}_1)\, \mathsf{out}\, f], \eta\, \mathsf{in}_2]^\star\, \mathsf{out}\big)\, (S^\nu\, \mathsf{in}_2).$$

Therefore

$$\begin{aligned}
\rho f^{\bigstar} &= \rho\, \mathsf{coit}\big([[\eta\, \mathsf{in}_1, S(\mathsf{id} + S^\nu\, \mathsf{in}_1)\, \mathsf{out}\, f], \eta\, \mathsf{in}_2]^\star\, \mathsf{out}\big)\, (S^\nu\, \mathsf{in}_2) \\
&= \big([[\eta\, \mathsf{in}_1, S(\mathsf{id} + S^\nu\, \mathsf{in}_1)\, \mathsf{out}\, f], \eta\, \mathsf{in}_2]^\star\, \mathsf{out}\big)^\dagger\, (S^\nu\, \mathsf{in}_2) && /\!/ \text{ (5.11)} \\
&= \big(S[\mathsf{id}, \mathsf{in}_2]\, [(S\, \mathsf{in}_1)\, [\eta\, \mathsf{in}_1, S(\mathsf{id} + S^\nu\, \mathsf{in}_1)\, \mathsf{out}\, f], \eta\, \mathsf{in}_2]^\star\, \mathsf{out}\big)^\dagger\, (S^\nu\, \mathsf{in}_2) \\
&= \big(([(S\, \mathsf{in}_1)\, [\eta\, \mathsf{in}_1, S(\mathsf{id} + S^\nu\, \mathsf{in}_1)\, \mathsf{out}\, f], \eta\, \mathsf{in}_2]^\star\, \mathsf{out})^\dagger\big)^\dagger\, (S^\nu\, \mathsf{in}_2) && /\!/ \text{ codiagonal} \\
&= ([\eta\, \mathsf{in}_1, S(\mathsf{id} + S^\nu\, \mathsf{in}_1)\, \mathsf{out}\, f]^\star\, \mathsf{out}^\dagger)^\dagger\, (S^\nu\, \mathsf{in}_2) && /\!/ \text{ naturality}
\end{aligned}$$

$$
\begin{aligned}
&= ([\eta\,\mathsf{in}_1, S(\mathsf{id} + S^\nu\,\mathsf{in}_1)\,\mathsf{out}\,f]^\star\rho)^\dagger\,(S^\nu\,\mathsf{in}_2)\\
&= [\eta, ([\eta\,\mathsf{in}_1,\rho]^\star S(\mathsf{id} + S^\nu\,\mathsf{in}_1)\,\mathsf{out}\,f)^\dagger]^\star\rho\,(T\,\mathsf{in}_2) && /\!/ \text{ dinaturality}\\
&= [\eta, ([\eta\,\mathsf{in}_1,\rho]^\star S(\mathsf{id} + S^\nu\,\mathsf{in}_1)\,\mathsf{out}\,f)^\dagger]^\star(S\,\mathsf{in}_2)\,\rho\\
&= (([\eta\,\mathsf{in}_1,\rho]^\star S(\mathsf{id} + S^\nu\,\mathsf{in}_1)\,\mathsf{out}\,f)^\dagger)^\star\,\rho\\
&= (((S\,\mathsf{in}_1)\,[\eta,\rho]^\star\,\mathsf{out}\,f)^\dagger)^\star\,\rho\\
&= (((S\,\mathsf{in}_1)\,\rho f)^\dagger)^\star\,\rho\\
&= ([\eta, ((S\,\mathsf{in}_1)\,\rho f)^\dagger]^\star(S\,\mathsf{in}_1)\,\rho f)^\star\,\rho && /\!/ \text{ fixpoint}\\
&= (\rho f)^\star\rho.
\end{aligned}
$$

Finally, let us check that ρ is an iteration congruence. Let $f, g\colon X \to_2 S^\nu(Y + X)$, which means that there are $f', g'\colon X \to S(Y + S^\nu(X + Y))$ such that $\mathsf{out}\,f = S(\mathsf{in}_1 + \mathsf{id})f'$ and $\mathsf{out}\,f = S(\mathsf{in}_1 + \mathsf{id})g'$. Suppose that $\rho f = \rho g$, which amounts to

$$[\eta\,\mathsf{in}_1,\rho]^\star f' = [\eta\,\mathsf{in}_1,\rho]^\star g', \tag{5.12}$$

for

$$\rho f = [\eta,\rho]^\star\,\mathsf{out}\,f = [\eta,\rho]^\star S(\mathsf{in}_1 + \mathsf{id})f' = [\eta\,\mathsf{in}_1,\rho]^\star f'$$

and analogously for g. Our goal is to prove that

$$\rho f^\ddagger = [\eta, ([\eta\,\mathsf{in}_1,\rho]^\star f')^\ddagger]^\star\eta\,\mathsf{in}_2,$$

from which $\rho f^\dagger = \rho g^\dagger$ will follow by the analogous formula for $\rho g^\dagger$ and (5.12). Observe that

$$f^\ddagger = (\mathsf{coit}\,h)\ \eta^\nu\,\mathsf{in}_2$$

where $h = [[\eta\,\mathsf{in}_1, f'], \eta\,\mathsf{in}_2]^\star\,\mathsf{out}$. Now

$$
\begin{aligned}
\rho f^\ddagger &= \rho\,(\mathsf{coit}\,h)\,\eta^\nu\,\mathsf{in}_2\\
&= h^\dagger\,\eta^\nu\,\mathsf{in}_2 && /\!/ \text{ (5.11)}\\
&= ([[\eta\,\mathsf{in}_1, f'], \eta\,\mathsf{in}_2]^\star\,\mathsf{out})^\dagger\,\eta^\nu\,\mathsf{in}_2\\
&= (S[\mathsf{id},\mathsf{in}_2]\,[[\eta\,\mathsf{in}_1\,\mathsf{in}_1, (S\,\mathsf{in}_1)f'], \eta\,\mathsf{in}_2]^\star\,\mathsf{out})^\dagger\,\eta^\nu\,\mathsf{in}_2\\
&= (([[\eta\,\mathsf{in}_1\,\mathsf{in}_1, (S\,\mathsf{in}_1)f'], \eta\,\mathsf{in}_2]^\star\,\mathsf{out})^\dagger)^\dagger\,\eta^\nu\,\mathsf{in}_2 && /\!/ \text{ codiagonal}\\
&= (([(S\,\mathsf{in}_1)[\eta\,\mathsf{in}_1, f'], \eta\,\mathsf{in}_2]^\star\,\mathsf{out})^\dagger)^\dagger\,\eta^\nu\,\mathsf{in}_2\\
&= ([\eta\,\mathsf{in}_1, f']^\star\,\mathsf{out}^\dagger)^\dagger\,\eta^\nu\,\mathsf{in}_2 && /\!/ \text{ naturality}\\
&= ([\eta\,\mathsf{in}_1, f']^\star\rho)^\dagger\,\eta^\nu\,\mathsf{in}_2\\
&= [\eta, ([\eta\,\mathsf{in}_1,\rho]^\star f')^\dagger]^\star\rho\,\eta^\nu\,\mathsf{in}_2 && /\!/ \text{ dinaturality}\\
&= [\eta, ([\eta\,\mathsf{in}_1,\rho]^\star f')^\dagger]^\star\eta^\nu\,\mathsf{in}_2
\end{aligned}
$$

and we are done. □

Example 5.8 (Finite trace semantics)**.** Let us revisit Example 4.5 (4), with A assumed to be finite throughout. Recall that $\mu\gamma.\,(X+1) + A\times\gamma \cong A^\star + A^\star\times X$ is a final $((X+1)+A\times\text{–})$-coalgebra in the Kleisli category $\mathbf{Set}_{\mathcal{P}_{\omega_1}}$ of $\mathcal{P}_{\omega_1}$. Note that $\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A\times\gamma)$ is a coalgebra of the same type in the same category, with

$$\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A\times\gamma) \xrightarrow{\mathcal{P}_{\omega_1}(\mathsf{in}_1 + \mathsf{id})\,\mathsf{out}\,\cup\,\{\cdot\}\,\mathsf{in}_1\,\mathsf{in}_2\,!} \mathcal{P}_{\omega_1}((X+1) + A\times\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A\times\gamma))$$

as the structure morphism, where $\cup$ denotes pointwise union and $\{\cdot\}$ is the map $x \mapsto \{x\}$, i.e. the unit of $\mathcal{P}_{\omega_1}$. Intuitively, we thus add 'non-termination', i.e. the element of the right-hand summand 1 in $X+1$, as a possible result to every state (in the original view of Hasuo et al. [20], this element instead represents acceptance, so the above definition would correspond to converting a labelled transition system into an automaton by making every state accepting). This yields a final coalgebra map $\xi_X \colon \nu\gamma.\, \mathcal{P}_{\omega_1}(X + A \times \gamma) \to \mathcal{P}_{\omega_1}(\mu\gamma.\,(X+1) + A \times \gamma)$ characterized by the diagram

$$\begin{array}{ccc} \nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma) & \xrightarrow{(\mathsf{in}_1 + \mathsf{id})\,\mathsf{out} \,\cup\, \{\cdot\}\,\mathsf{in}_1\,\mathsf{in}_2\,!} & (X+1) + A \times \nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma) \\ {\scriptstyle \xi_X}\downarrow & (\text{in } \mathbf{Set}_{\mathcal{P}_{\omega_1}}) & \downarrow{\scriptstyle X+1+A\times\xi_X} \\ \mu\gamma.\,(X+1) + A \times \gamma & \xrightarrow{\mathsf{in}^{-1}} & (X+1) + A \times \mu\gamma.\,(X+1) + A \times \gamma \end{array}$$

which amounts to the following corecursive definition of ξ_X:

$$\xi_X(t) = \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \mathsf{out}\,t\} \cup \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\} \cup \{\mathsf{in}\,\mathsf{in}_2\langle a, t''\rangle \mid t'' \in \xi_X(t'), \mathsf{in}_2\langle a, t'\rangle \in \mathsf{out}\,t\}.$$

The result of applying ξ_X to a tree is the set of finite traces in it, which are finite sequences from $A^\star$ followed either by an element of X (successfully terminating traces) or by the single inhabitant of 1 (divergent traces). It is easy to see that ξ is a natural transformation; we show that it is in fact a monad morphism. The domain of ξ is a generalized coalgebraic resumption monad $\nu\gamma.\,\mathcal{P}_{\omega_1}(\text{--} + A \times \gamma)$ (on $\mathbf{Set}$) as discussed in Example 3.12, while the codomain $\mathcal{P}_{\omega_1}(\mu\gamma.\,(\text{--}+1) + A \times \gamma)$ is obtained by sandwiching the monad $\nu\gamma.\,(\text{--}+1) + A \times \gamma$ (on $\mathbf{Set}_{\mathcal{P}_{\omega_1}}$) between the adjoint pair $F \dashv G \colon \mathbf{Set}_{\mathcal{P}_{\omega_1}} \to \mathbf{Set}$ generating the monad $\mathcal{P}_{\omega_1}$, and therefore is also a monad (cf. Section 2). The corresponding structure is defined as follows:

$$\eta(x) = \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x\}$$

$$\big(f \colon X \to \mathcal{P}_{\omega_1}(\mu\gamma.\,(Y+1) + A \times \gamma)\big)^\star(p) = \bigcup \{(\mathsf{iter}\,\hat{f})(t) \mid t \in p\}$$

where $\mathsf{iter}\,\hat{f} \colon \mu\gamma.\,(X+1) + A \times \gamma \to \mathcal{P}_{\omega_1}(\mu\gamma.\,(Y+1) + A \times \gamma)$ is the initial algebra morphism to the algebra $(\mathcal{P}_{\omega_1}(\mu\gamma.\,(Y+1) + A \times \gamma), \hat{f})$ whose structure map

$$\hat{f} \colon (X+1) + A \times \mathcal{P}_{\omega_1}(\mu\gamma.\,(Y+1) + A \times \gamma) \to \mathcal{P}_{\omega_1}(\mu\gamma.\,(Y+1) + A \times \gamma)$$

is as follows: $\hat{f}(\mathsf{in}_1\,\mathsf{in}_1\,x) = f(x)$, $\hat{f}(\mathsf{in}_1\,\mathsf{in}_2\,\star) = \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\}$, $\hat{f}(\mathsf{in}_2\langle a, t\rangle) = \{\mathsf{in}\,\mathsf{in}_2\langle a, t\rangle\}$. This results in the following inductive definition of $\mathsf{iter}\,\hat{f}$:

$$\begin{aligned} (\mathsf{iter}\,\hat{f})\,\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x &= f(x) \\ (\mathsf{iter}\,\hat{f})\,\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star &= \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\} \\ (\mathsf{iter}\,\hat{f})\,\mathsf{in}\,\mathsf{in}_2\langle a, t\rangle &= \{\mathsf{in}\,\mathsf{in}_2\langle a, t'\rangle \mid t' \in (\mathsf{iter}\,\hat{f})(t)\} \end{aligned}$$

It is then easy to see that ξ respects η. The fact that ξ respects Kleisli lifting amounts to a rather technical verification of the fact that both $\xi f^\star$ and $(\xi f)^\star \xi$ satisfy the same corecursive definition and are thus equal:

$$\begin{aligned} \xi\, f^\star(p) &= \big\{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \mathsf{out}\,f^\star(p)\big\} \cup \big\{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\big\} \\ &\qquad \cup \big\{\mathsf{in}_2\langle a, t''\rangle \mid t'' \in \xi(t'), \mathsf{in}_2\langle a, t'\rangle \in \mathsf{out}\,f^\star(p)\big\} \\ &= \big\{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \{\mathsf{out}\,f(x) \mid \mathsf{in}_1\,x \in \mathsf{out}\,p\}\big\} \cup \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\} \end{aligned}$$

$$
\begin{aligned}
&\cup \big\{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \{\mathsf{in}_2\langle a, f^\star(p')\rangle \mid \mathsf{in}_2\langle a,p'\rangle \in \mathsf{out}\,p\}\big\}\\
&\cup \big\{\mathsf{in}_2\langle a,t''\rangle \mid t'' \in \xi(t'), \mathsf{in}_2\langle a,t'\rangle \in \{\mathsf{out}\,f(x) \mid \mathsf{in}_1\,x \in \mathsf{out}\,p\}\big\}\\
&\cup \big\{\mathsf{in}_2\langle a,t''\rangle \mid t'' \in \xi(t'), \mathsf{in}_2\langle a,t'\rangle \in \{\mathsf{in}_2\langle a, f^\star(p')\rangle \mid \mathsf{in}_2\langle a,p'\rangle \in \mathsf{out}\,p\}\big\}\\
= {}& \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \mathsf{out}\,f(x), \mathsf{in}_1\,x \in \mathsf{out}\,p\} \cup \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\}\\
&\cup \{\mathsf{in}_2\langle a,t''\rangle \mid t'' \in \xi(t'), \mathsf{in}_2\langle a,t'\rangle \in \mathsf{out}\,f(x), \mathsf{in}_1\,x \in \mathsf{out}\,p\}\\
&\cup \{\mathsf{in}_2\langle a,t''\rangle \mid t'' \in \xi(f^\star(p')), \mathsf{in}_2\langle a,p'\rangle \in \mathsf{out}\,p\}
\end{aligned}
$$

$$
\begin{aligned}
(\xi f)^\star(\xi(p)) = {}& \bigcup \{(\mathsf{iter}\,\widehat{\xi f})(t) \mid t \in \xi(p)\}\\
= {}& \bigcup \big\{(\mathsf{iter}\,\widehat{\xi f})(t) \mid t \in \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \mathsf{out}\,p\}\big\}\\
&\cup \bigcup \big\{(\mathsf{iter}\,\widehat{\xi f})(t) \mid t \in \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\}\big\}\\
&\cup \bigcup \big\{(\mathsf{iter}\,\widehat{\xi f})(t) \mid t \in \{\mathsf{in}\,\mathsf{in}_2\langle a,t''\rangle \mid t'' \in \xi(t'), \mathsf{in}_2\langle a,t'\rangle \in \mathsf{out}\,p\}\big\}\\
= {}& \bigcup \{\xi f(x) \mid \mathsf{in}_1\,x \in \mathsf{out}\,p\} \cup \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\}\\
&\cup \{\mathsf{in}_2\langle a,t\rangle \mid t \in (\mathsf{iter}\,\widehat{\xi f})(t''), t'' \in \xi(t'), \mathsf{in}_2\langle a,t'\rangle \in \mathsf{out}\,p\}\\
= {}& \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \mid \mathsf{in}_1\,x \in \mathsf{out}\,f(x), \mathsf{in}_1\,x \in \mathsf{out}\,p\} \cup \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\}\\
&\cup \{\mathsf{in}_2\langle a,t''\rangle \mid t'' \in \xi(t'), \mathsf{in}_2\langle a,t'\rangle \in \mathsf{out}\,f(x), \mathsf{in}_1\,x \in \mathsf{out}\,p\}\\
&\cup \{\mathsf{in}_2\langle a,t''\rangle \mid t'' \in (\xi f)^\star(\xi(t')), \mathsf{in}_2\langle a,t'\rangle \in \mathsf{out}\,p\}
\end{aligned}
$$

Now consider the situation where guardedness for $\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma)$ is induced by vacuous guardedness for $\mathcal{P}_{\omega_1}(\text{–} + A \times \gamma)$ by Theorem 4.3 (1) and with guardedness for $\mathcal{P}_{\omega_1}(\mu\gamma.\,(X+1) + A \times \gamma)$ defined as follows: $f\colon X \to \mathcal{P}_{\omega_1}(\mu\gamma.\,(Y+1) + A \times \gamma)$ is σ-guarded iff as a morphism $f\colon X \to \nu\gamma.\,(Y+1) + A \times \gamma$ in $\mathbf{Set}_{\mathcal{P}_{\omega_1}}$ it is σ-guarded under the notion of guardedness induced by vacuous guardedness for $(\text{–} + 1) + A \times \text{–}$ in $\mathbf{Set}_{\mathcal{P}_{\omega_1}}$, again by Theorem 4.3 (1). This turns ξ into a guarded monad morphism, and moreover ξ is iteration-preserving by Lemma 3.10, because, as we argued before in Example 4.5 (4), its codomain $\mathcal{P}_{\omega_1}(\mu\gamma.\,(X+1) + A \times \gamma)$ is guarded iterative (a more abstract argument showing that sandwiching a guarded iterative monad between an adjoint pair produces a guarded iterative monad is later given in Theorem 6.1).

In order to obtain a guarded retraction from ξ, let ρ be the epimorphic part of the image factorization of ξ. It is easy to verify that the codomain of ρ consists precisely of the prefix-closed subsets of $\mathcal{P}_{\omega_1}(\mu\gamma.\,(X+1) + A \times \gamma)$, i.e. is the guarded iterative submonad of $\mathcal{P}_{\omega_1}(\mu\gamma.\,(X+1) + A \times \gamma)$ mentioned in Example 4.5 (4). Under the *axiom of choice*, this is sufficient to turn ρ into a retraction because every epi splits. However, the requisite section υ can also be constructed explicitly without choice, for every prefix-closed subset of $\mathcal{P}_{\omega_1}(\mu\gamma.\,(X+1) + A \times \gamma)$ standardly induces an A-branching tree, hence an element of $\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma)$. In summary,

$$\rho_X(t) = \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \,|\, \mathsf{in}_1\,x \in \mathsf{out}\,t\} \cup \{\mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_2\,\star\} \cup \{\mathsf{in}\,\mathsf{in}_2\langle a,t''\rangle \,|\, t'' \in \rho_X(t'), \mathsf{in}_2\langle a,t'\rangle \in \mathsf{out}\,t\}$$

$$\upsilon_X(S) = \mathsf{out}^{-1}\big(\{\mathsf{in}_1\,x \mid \mathsf{in}\,\mathsf{in}_1\,\mathsf{in}_1\,x \in S\} \cup \{\mathsf{in}_2\langle a, \upsilon_X(\{t \mid \mathsf{in}\,\mathsf{in}_2\langle a,t\rangle \in S\})\rangle \mid a \in A\}\big)$$

where $t \in \nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma)$ and S is a countable prefix-closed subset of $\mu\gamma.\,(X+1) + A \times \gamma$. Note that the tree constructed by υ has only very special kind of nondeterminism, not including non-deterministic choice between processes prefixed by actions. Roughly, we can have $x + y$ and $x + a.t$ in the image of υ with $x, y \in X$ and $a \in A$, but not $a.t + b.s$ with $a, b \in A$. The composition $\upsilon\rho$ can therefore be seen as a determinization procedure, pushing the non-deterministic choice downwards along the tree. Of course, non-determinism can not be entirely eliminated, because in the end we arrive at subsets of X, which must remain intact. We conjecture that this effect is generic, i.e. that the same scenario can be run with ω_1 replaced by any other regular infinite cardinal κ; that is, $\upsilon\rho$ pushes κ-branching non-determinism downwards in the same sense as above. We also conjecture that υ is a monad morphism and hence so is $\upsilon\rho$.

The established retraction (ρ, υ) can thus be reused in two further cases.

Guarded iteration for finitely-branching processes We can restrict ρ to the monad $\nu\gamma.\,\mathcal{P}_\omega(\text{–} + A \times X)$ capturing finitely branching processes with outputs in X. As indicated above, we then essentially again obtain countable prefix-closed sets P of traces as the image of ρ, which however now additionally satisfy the condition that for each $w \in A^*$, the set $\{x \in X \mid (w, x) \in P\}$ is finite (while in the countably branching case, and for infinite X, these sets may be countably infinite). The section υ restricts accordingly, and we thus obtain a guarded retraction.

Unguarded iteration for countably-branching processes As discussed in Example 4.5 (2), $\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma)$ supports unguarded iteration, and in fact is an Elgot monad [18]. In the remainder of the example we use the terms "unguarded" for total guardedness and "guarded" for the notion of guardedness on $\nu\gamma.\,\mathcal{P}_{\omega_1}(X + A \times \gamma)$ discussed above. Now, in order to conclude by Theorem 5.7 that the codomain of ρ as above is an Elgot monad, it suffices to check that ρ remains iteration preserving if we equip its domain with total guardedness, i.e. that ρ preserves iteration also of unguarded morphisms. So let $f\colon X \to \nu\gamma.\,\mathcal{P}_{\omega_1}((Y + X) + A \times \gamma)$. The unguarded iterate $f^\ddagger$ is defined as the guarded iterate $\hat{f}^\ddagger$, where $\hat{f}$ has the same profile as f and is defined as the guarded morphism

$$\hat{f} = \mathsf{out}^{-1}\,\mathcal{P}_{\omega_1}(\mathsf{in}_1 + \mathsf{id})\left(\mathcal{P}_{\omega_1}[\mathsf{in}_1 + \mathsf{id}, \mathsf{in}_1\,\mathsf{in}_2]\,\mathsf{out}\,f\right)^\dagger,$$

with iteration $(\text{–})^\dagger$ on $\mathcal{P}_{\omega_1}$ calculated in the expected way using least fixpoints [18]. It is easy to check that

$$\upsilon\rho\hat{f} = \mathsf{out}^{-1}\,\mathcal{P}_{\omega_1}(\mathsf{in}_1 + \mathsf{id})\left(\mathcal{P}_{\omega_1}[\mathsf{in}_1 + \mathsf{id}, \mathsf{in}_1\,\mathsf{in}_2]\,\mathsf{out}\,\upsilon\rho f\right)^\dagger$$

and thus $(\upsilon\rho\hat{f})^\ddagger = (\upsilon\rho f)^\ddagger$ by the above definition of $(\upsilon\rho f)^\ddagger$. Therefore, using (5.9) and the fact that, as we argued above, ρ preserves guarded iteration, $\rho f^\ddagger = \rho\hat{f}^\ddagger = \rho(\upsilon\rho\hat{f})^\ddagger = \rho(\upsilon\rho f)^\ddagger$, which means that ρ is iteration preserving.

Recall from Section 4 that guardedness, guarded iteration, and the coinductive resumption transform work at the level of *parametrized monads*, i.e. functors from a parameter category $\mathbf{D}$ into the category of monads on a category $\mathbf{C}$, typically rearranged into bifunctors $\#\colon \mathbf{C} \times \mathbf{D} \to \mathbf{C}$. The notions of guarded retraction and iteration congruence extend straightforwardly to parametrized monads; explicitly:

Definition 5.9. A parametrized guarded monad morphism is a *guarded retraction* (an *iteration congruence*) if its components are guarded retractions (iteration congruences).

We then can take the claims of Theorem 4.3 further:

Theorem 5.10. *Let* $\#, \hat{\#}\colon \mathbf{C} \times (\mathbf{C} \times \mathbf{D}) \to \mathbf{C}$ *be guarded parametrized monads, and let* $\rho\colon \# \to \hat{\#}$ *be an iteration-congruent retraction. By Theorem 4.3,* $\#^\nu = \nu\gamma.\,\text{–}\,\#(\gamma, \text{–})$

and $\hat{\#}^\nu = \nu\gamma.\,\text{–}\,\hat{\#}(\gamma,\text{–})$ *are also parametrized guarded monads. Then* $\rho^\nu\colon \#^\nu \to \hat{\#}^\nu$*, with components*

$$\rho^\nu_{X,Y} = \mathsf{coit}\big(\nu\gamma.\,X \# (\gamma, Y) \xrightarrow{\rho\,\mathsf{out}} X \mathbin{\hat{\#}} (\nu\gamma.\,X \# (\gamma,Y), Y)\big),$$

is again an iteration-congruent retraction.

Proof. It is already shown in Theorem 4.3 that ρ^ν is a monad morphism.

We define the associated section by $\upsilon^\nu = \mathsf{coit}(\upsilon\,\mathsf{out})$. Indeed it is easy to check that $\rho^\nu\upsilon^\nu = \mathsf{id}$: since

$$\begin{aligned}\mathsf{out}\;\rho^\nu\upsilon^\nu &= (\mathsf{id} \mathbin{\hat{\#}} \rho^\nu)\,\rho\,\mathsf{out}\,\upsilon^\nu\\ &= (\mathsf{id} \mathbin{\hat{\#}} \rho^\nu)\,\rho(\mathsf{id} \# \upsilon^\nu)\,\upsilon\,\mathsf{out}\\ &= (\mathsf{id} \mathbin{\hat{\#}} \rho^\nu)(\mathsf{id} \mathbin{\hat{\#}} \upsilon^\nu)\,\rho\upsilon\,\mathsf{out}\\ &= (\mathsf{id} \mathbin{\hat{\#}} \rho^\nu\upsilon^\nu)\;\mathsf{out}\,.\end{aligned}$$

and also $\mathsf{out}\,\mathsf{id} = (\mathsf{id} \mathbin{\hat{\#}} \mathsf{id})\,\mathsf{out}$, the claim $\rho^\nu\upsilon^\nu = \mathsf{id}$ follows by uniqueness of final coalgebra morphisms.

Next, suppose that $f\colon X \to Y \mathbin{\hat{\#}}^\nu Z$ is σ-guarded, which according to Theorem 4.3 means that $\mathsf{out}\,f$ is σ-guarded. We need to show that so is $\upsilon^\nu f\colon X \to Y \#^\nu Z$. Now

$$\mathsf{out}\,\upsilon^\nu f = (\mathsf{id} \# \upsilon^\nu)\,\upsilon\,\mathsf{out}\,f$$

is σ-guarded because ρ is a guarded retraction and hence $\upsilon\,\mathsf{out}\,f$ is σ-guarded, and $\mathsf{id} \# \upsilon^\nu$ is a parametrized guarded monad morphism. Hence, again, according to Theorem 4.3, $\upsilon^\nu f$ is σ-guarded. We have thus proved that ρ^ν is a guarded retraction.

We are left to check that ρ^ν is an iteration congruence. Suppose that $\rho^\nu f = \rho^\nu g$ for some $f, g\colon X \to_2 (Y + X) \#^\nu Z$. Then

$$\rho\,(\mathsf{id} \# \rho^\nu)\,\mathsf{out}\,f = (\mathsf{id} \mathbin{\hat{\#}} \rho^\nu)\,\rho\;\mathsf{out}\,f = \mathsf{out}\,\rho^\nu f$$

(by naturality of ρ and the definition of ρ^ν) and analogously for g in place of f, so using that ρ is an iteration congruence, we obtain

$$\rho\,((\mathsf{id} \# \rho^\nu)\,\mathsf{out}\,f)^\dagger = \rho\,((\mathsf{id} \# \rho^\nu)\,\mathsf{out}\,g)^\dagger. \tag{5.13}$$

Observe that for suitably typed h,

$$\begin{aligned}\mathsf{out}\,\mathsf{coit}(\rho\,\mathsf{out})\,(\mathsf{coit}\,h) &= (\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}(\rho\,\mathsf{out}))\,\rho\,\mathsf{out}\,(\mathsf{coit}\,h)\\ &= (\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}(\rho\,\mathsf{out}))\,\rho\,(\mathsf{id} \# (\mathsf{coit}\,h))\,h\\ &= (\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}(\rho\,\mathsf{out})\,(\mathsf{coit}\,h))\,\rho\,h,\end{aligned}$$

and therefore, by finality of $\mathsf{coit}(\rho h)$,

$$\mathsf{coit}(\rho\,\mathsf{out})\,(\mathsf{coit}\,h) = \mathsf{coit}(\rho h). \tag{5.14}$$

Therefore,

$$\begin{aligned}\rho^\nu f^\ddagger &= \rho^\nu\,\mathsf{coit}([\eta, (\mathsf{out}\,f)^\dagger]^\star\,\mathsf{out})\,\eta\,\mathsf{in}_2 && /\!/\ \text{Theorem 4.3}\\ &= \mathsf{coit}(\rho\,\mathsf{out})\,\mathsf{coit}([\eta, (\mathsf{out}\,f)^\dagger]^\star\,\mathsf{out})\,\eta\,\mathsf{in}_2\\ &= \mathsf{coit}(\rho[\eta, (\mathsf{out}\,f)^\dagger]^\star\,\mathsf{out})\,\eta\,\mathsf{in}_2 && /\!/\ (5.14)\\ &= \mathsf{coit}([\eta, \rho\,((\mathsf{id} \# \rho^\nu)\;\mathsf{out}\,f)^\dagger]^\star\,\mathsf{out})\,\rho^\nu\eta\,\mathsf{in}_2\,.\end{aligned}$$

The last step is due to uniqueness of the final coalgebra morphism $\mathsf{coit}([\eta, \rho\,(\mathsf{out}\,f)^\dagger]^\star \rho\,\mathsf{out})$ and the following calculation:

$$
\begin{aligned}
&\mathsf{out}\,\mathsf{coit}\big([\eta, \rho\,((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star\, \mathsf{out}\big)\, \rho^\nu \\
&\quad= \big(\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}([\eta, \rho\,((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star\, \mathsf{out})\big) \\
&\qquad\quad [\eta, \rho\,((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star\, \mathsf{out}\,\rho^\nu \\
&\quad= \big(\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}([\eta, \rho\,((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star\, \mathsf{out})\big) \\
&\qquad\quad \rho\,[\eta, ((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star (\mathsf{id} \mathbin{\#} \rho^\nu)\, \mathsf{out} && /\!\!/\ \rho \text{ is a monad morphism} \\
&\quad= \big(\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}([\eta, \rho\,((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star\, \mathsf{out})\big) \\
&\qquad\quad \rho\,(\mathsf{id} \mathbin{\#} \rho^\nu)\,[\eta, (\mathsf{out}\,f)^\dagger]^\star\, \mathsf{out} && /\!\!/\ \mathsf{id} \mathbin{\#} \rho^\nu \text{ is a monad morphism} \\
&\quad= \big(\mathsf{id} \mathbin{\hat{\#}} \mathsf{coit}([\eta, \rho\,((\mathsf{id} \mathbin{\#} \rho^\nu)\ \mathsf{out}\,f)^\dagger]^\star\, \mathsf{out})\,\rho^\nu\big) \\
&\qquad\quad \rho\,[\eta, (\mathsf{out}\,f)^\dagger]^\star\, \mathsf{out}\,.
\end{aligned}
$$

An analogous calculation applies to $\rho^\nu g^\ddagger$, and therefore by (5.13), $\rho^\nu f^\ddagger = \rho^\nu g^\ddagger$. □

Theorems 5.7 and 5.10 jointly provide a simple and structured way of showing that Elgotness extends along the parametrized monad transformer $\# \mapsto \hat{\#}$: If $- \# X$ is Elgot, then by Theorem 5.7 there is an iteration-congruent retraction $\rho\colon \nu\gamma.\, - + \gamma \# X \to - \# X$. By Theorem 5.10, this gives rise to an iteration-congruent retraction

$$\rho^\nu\colon \nu\gamma'.\,\nu\gamma.\, - + \gamma \# (\gamma', X) \to \nu\gamma'.\, - \# (\gamma', X)$$

and by Theorem 5.7, the right-hand side is again Elgot. We have thus proved

Corollary 5.11. *Given a parametrized monad* $\#$ *and* $X \in |\mathbf{C}|$, *if* $- \# X$ *is Elgot then so is* $- \#^\nu X = \nu\gamma.\, - \# (\gamma, X)$.

In particular, we have thus obtained a more structured and simpler proof of one of the main results in [18], which states that the coinductive generalized resumption monad transformer preserves Elgotness.

Theorem 5.7 characterizes iteratable Elgot monads as iteration-congruent retracts of their $(-)^\nu$-transforms. We take this perspective further as follows.

Definition 5.12. We extend the notation $F^\nu = \nu\gamma.F(- + \gamma)$ to functors F. We say that a functor F is

- 1-*iteratable* if F^ν exists,
- $(n + 1)$-*iteratable* if F^ν is n-iteratable,
- ω-*iteratable* if F is n-iteratable for every n.

We apply all these notions mainly to monads $\mathbb{T}$, referring to their underlying functor T.

Remark 5.13. Note that for every natural number n,

$$\nu\gamma'.\,\nu\gamma.\,T(X + \gamma' + n \times \gamma) \cong \nu\gamma.\,T(X + \gamma + n \times \gamma) \cong \nu\gamma.\,T(X + (n + 1) \times \gamma),$$

where $n \times X$ denotes the n-fold sum $X + \ldots + X$. It follows by induction that n-iteratability of T is equivalent to the assumption that all coalgebras $\nu\gamma.\,T(X + n \times \gamma)$ exist, a condition that does not appear much stronger than iteratability of T. Still, the 2-iteratable functors are properly contained in the iteratable functors, as the following example shows. Let $\mathbf{C}$ be the category of countable sets and $T = \mathrm{Id}$. Then, it is easy to see that $T^\nu X$ is isomorphic to

$X \times \mathbb{N} + 1$, hence Id is iteratable. However, it is not 2-iteratable, because $(T^\nu)^\nu\emptyset \cong \nu\gamma.\, 2 \times \gamma$ can be characterized as the object of all infinite bit streams, which does not fit into $\mathbf{C}$ for cardinality reasons. Showing this formally amounts to mimicking Cantor's classical diagonalization argument.

We expect that separating n-iteratability from $(n+1)$-iteratability for $n > 1$ would involve much less natural examples, as the previous cardinality argument typically would not apply.

Consider the functor $\mathbb{T} \mapsto \mathbb{T}^\nu$ on the category of ω-iteratable monads over $\mathbf{C}$. This construction is itself a monad: the unit $\boldsymbol{\eta}$ is the natural transformation with components $\boldsymbol{\eta}_X = \mathsf{out}^{-1}(T\,\mathsf{in}_1)\colon TX \to T^\nu X$, and the multiplication $\boldsymbol{\mu}\colon T^{\nu\nu} \to T^\nu$ has components

$$\boldsymbol{\mu}_X = \mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\colon T^{\nu\nu}X \to T(X + T^{\nu\nu}X)\big).$$

We record explicitly that the relevant laws are satisfied:

Lemma 5.14. *With multiplication $\boldsymbol{\mu}$ and unit $\boldsymbol{\eta}$ as defined above, the construction $(\text{–})^\nu$ becomes a monad on the (overlarge) category of ω-iteratable monads.*

Proof. By coinduction. Using the definitions of $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$, we have

$$\begin{aligned}
\mathsf{out}\,\boldsymbol{\mu\eta} &= \mathsf{out}\,\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\big)\,\mathsf{out}^{-1}(T^\nu\,\mathsf{in}_1)\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\big)\,T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,(T^\nu\,\mathsf{in}_1)\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\big)\,T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,T\big(\mathsf{in}_1 + (T^\nu\,\mathsf{in}_1)\big)\,\mathsf{out}\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\,\mathsf{out}^{-1}\,(T^\nu\,\mathsf{in}_1)\big)\,\mathsf{out}\\
&= T\big(\mathsf{id} + \boldsymbol{\mu\eta}\big)\,\mathsf{out},
\end{aligned}$$

and therefore $\boldsymbol{\mu\eta} = \mathsf{id}$ by uniqueness of final coalgebra morphisms. Analogously,

$$\begin{aligned}
\mathsf{out}\,\boldsymbol{\mu\eta}^\nu &= \mathsf{out}\,\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\big)\,\mathsf{coit}(\boldsymbol{\eta}\,\mathsf{out})\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\big)\\
&\quad\; T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,T(\mathsf{id} + \mathsf{coit}(\boldsymbol{\eta}\,\mathsf{out}))\,\boldsymbol{\eta}\,\mathsf{out}\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\big)\\
&\quad\; T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,(T\,\mathsf{in}_1)\,T\big(\mathsf{id} + \mathsf{coit}(\boldsymbol{\eta}\,\mathsf{out})\big)\;\mathsf{out}\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\big)\,T\big(\mathsf{id} + \mathsf{coit}(\boldsymbol{\eta}\,\mathsf{out})\big)\;\mathsf{out}\\
&= T(\mathsf{id} + \boldsymbol{\mu\eta}^\nu)\,\mathsf{out}
\end{aligned}$$

and therefore $\mathsf{out}\,\boldsymbol{\mu\eta}^\nu = \mathsf{id}$. The remaining law $\boldsymbol{\mu\mu} = \boldsymbol{\mu}\,\boldsymbol{\mu}^\nu$ follows by the same argument from

$$\begin{aligned}
\mathsf{out}\,\boldsymbol{\mu\mu}^\nu &= \mathsf{out}\,\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\big)\,\mathsf{coit}(\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\big)\,\mathsf{out})\\
&= T\big(\mathsf{id} + \mathsf{coit}(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out})\big)\,T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\\
&\quad\; T\big(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\big)\,\mathsf{out})\big)\;\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\big)\,\mathsf{out}\\
&= T(\mathsf{id} + \boldsymbol{\mu})\,T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,T(\mathsf{id} + \boldsymbol{\mu}^\nu)\,\boldsymbol{\mu}\,\mathsf{out}\\
&= T[\mathsf{id} + \boldsymbol{\mu\mu}^\nu, \mathsf{in}_2\,\boldsymbol{\mu}\,\mathsf{out}^{-1}\,T(\mathsf{id} + \boldsymbol{\mu}^\nu)]\,\mathsf{out}\,\boldsymbol{\mu}\,\mathsf{out}\\
&= T[\mathsf{id} + \boldsymbol{\mu\mu}^\nu, \mathsf{in}_2\,\boldsymbol{\mu}\,\mathsf{out}^{-1}\,T(\mathsf{id} + \boldsymbol{\mu}^\nu)\,\boldsymbol{\mu}]\,T[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out}\,\mathsf{out}
\end{aligned}$$

$$= T[\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu}^\nu, \mathsf{in}_2\, \boldsymbol{\mu}\boldsymbol{\mu}^\nu\, \mathsf{out}^{-1}]\, T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\,\mathsf{out}$$
$$= T(\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu}^\nu)\, T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\,\mathsf{out}$$

$$\begin{aligned}
\mathsf{out}\,\boldsymbol{\mu}\boldsymbol{\mu} &= \mathsf{out}\,\mathsf{coit}\big(T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\big)\, \mathsf{coit}\big(T^\nu[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\big)\\
&= T(\mathsf{id} + \boldsymbol{\mu})\, T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\, T^\nu(\mathsf{id} + \boldsymbol{\mu})\, T^\nu[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\\
&= T[\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu}, \mathsf{in}_2\, \boldsymbol{\mu}\, \mathsf{out}^{-1}\, T^\nu(\mathsf{id} + \boldsymbol{\mu})]\;\, \mathsf{out}\, T^\nu[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\\
&= T[(\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu})\, [\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}], \mathsf{in}_2\, \boldsymbol{\mu}\, \mathsf{out}^{-1}\, T^\nu(\mathsf{id} + \boldsymbol{\mu})\, T^\nu[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]]\;\, \mathsf{out}\,\mathsf{out}\,\mathsf{out}\\
&= T[(\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu})\, [\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}], \mathsf{in}_2\, \boldsymbol{\mu}\boldsymbol{\mu}\, \mathsf{out}^{-1}\, \mathsf{out}^{-1}]\;\, \mathsf{out}\,\mathsf{out}\,\mathsf{out}\\
&= T(\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu})\, T[[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}], \mathsf{in}_2\, \mathsf{out}^{-1}\, \mathsf{out}^{-1}]\;\, \mathsf{out}\,\mathsf{out}\,\mathsf{out}\\
&= T(\mathsf{id} + \boldsymbol{\mu}\boldsymbol{\mu})\, T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, T[\mathsf{id}, \mathsf{in}_2\, \mathsf{out}^{-1}]\, \mathsf{out}\,\mathsf{out}\,\mathsf{out} \qquad \square
\end{aligned}$$

For every T we now define the *delay transformation*

$$\triangleright = \mathsf{out}^{-1}\, \eta\, \mathsf{in}_2 \colon T^\nu \to T^\nu.$$

This leads to our second main result:

Theorem 5.15. *The category of ω-iteratable Elgot monads over $\mathbf{C}$ is isomorphic to the full subcategory of the category of $(\text{–})^\nu$-algebras consisting of the $(\text{–})^\nu$-algebras $(\mathbb{S}, \rho\colon \mathbb{S}^\nu \to \mathbb{S})$ (for ω-iteratable $\mathbb{S}$) satisfying $\rho \triangleright = \rho$.*

We refer to the condition $\rho \triangleright = \rho$ as *delay cancellation*.

Remark 5.16. The point of the above result is to systematize the connection between the $(\text{–})^\nu$ construction and Elgot monads previously indicated by Theorem 5.7. Alternative efforts to show that Elgotness is monadic exist (see Section 7) but necessarily involve quite different monads than $(\text{–})^\nu$: Any monad $\mathfrak{M}$ (on a category of monads) whose algebras are precisely the Elgot monads would itself have to produce Elgot monads $\mathfrak{M}\mathbb{T}$, while the point of involving $(\text{–})^\nu$ is to obtain Elgot monads from guarded iterative ones.

Formally, the following simple example shows that the delay cancellation condition $\rho \triangleright = \rho$ cannot be omitted from Theorem 5.15. Let $\mathbf{Mon}(\mathbf{C})^\nu$ be the category of $(\text{–})^\nu$-algebras, and let $\mathbf{Mon}(\mathbf{C})^\nu_\triangleright$ be the full subcategory of $\mathbf{Mon}(\mathbf{C})^\nu$ figuring in Theorem 5.15. Since the identity functor is the initial monad, the initial object of $\mathbf{Mon}(\mathbf{C})^\nu$ is Capretta's delay monad [7] $D = \nu\gamma.\,(\text{–} + \gamma)$. On the other hand, the initial object of $\mathbf{Mon}(\mathbf{C})^\nu_\triangleright$ (if it exists) is the *initial Elgot monad* $\mathbb{L}$, which on $\mathbf{C} = \mathbf{Set}$ is the *maybe monad* $(\text{–}) + 1$.

If $\mathbf{C} = \mathbf{Set}$, then $DX = (X \times \mathbb{N} + 1)$ does turn out to be Elgot [15] (but applying Theorem 5.15 to D qua Elgot monad yields a different $(\text{–})^\nu$-algebra structure than the initial one), and $\mathbb{L}$ is, in this case, a retract of $\mathbb{D}$ in $\mathbf{Mon}(\mathbf{C})^\nu_\triangleright$. The situation is more intricate in categories with a nonclassical internal logic, for which $\mathbb{D}$ is mainly intended. We believe that in such a setting, neither is $\mathbb{D}$ Elgot in general, nor is $\mathbb{L}$ the maybe monad. However, there will still be a unique $(\text{–})^\nu$-algebra morphism $\mathbb{D} \to \mathbb{L}$ in $\mathbf{Mon}(\mathbf{C})^\nu$.

Proof (Theorem 5.15). We fix the notation $(\eta, \text{–}^\star, \dagger)$ for (potential) Elgot monads over $\mathbf{C}$ and $(\eta^\nu, \text{–}^{\bigstar}, \ddagger)$ for their $(\text{–})^\nu$-transforms. We record the following identity, satisfied by any monad morphism ρ for which $\rho\boldsymbol{\eta} = \mathsf{id}$ and $\rho \triangleright = \rho$:

$$\rho = [\eta, \rho]^\star\, \mathsf{out}\,. \tag{5.15}$$

Indeed,

$$\begin{aligned}
[\eta,\rho]^\star\,\mathsf{out} &= [\eta,\rho]^\star\rho\boldsymbol{\eta}\,\mathsf{out} && /\!\!/\ \rho\boldsymbol{\eta}=\mathsf{id}\\
&= [\eta,\rho\,\rhd]^\star\rho\boldsymbol{\eta}\,\mathsf{out} && /\!\!/\ \rho\,\rhd=\rho\\
&= \rho\,[\eta,\rhd]^{\bigstar}\boldsymbol{\eta}\,\mathsf{out} && /\!\!/\ \rho \text{ is a monad morphism}\\
&= \rho\,[\eta,\rhd]^{\bigstar}\,\mathsf{out}^{-1}(S\,\mathsf{in}_1)\,\mathsf{out}\\
&= \rho\,\mathsf{out}^{-1}\,[\mathsf{out}[\eta,\mathsf{id}],\eta\,\mathsf{in}_2[\eta,\mathsf{id}]^{\bigstar}]^\star(S\,\mathsf{in}_1)\,\mathsf{out}\\
&= \rho\,\mathsf{out}^{-1}\,[\eta\,\mathsf{in}_1,\eta\,\mathsf{in}_2]^\star\,\mathsf{out}\\
&= \rho.
\end{aligned}$$

For the inclusion from Elgot monads to $(\text{–})^\nu$-algebras, let $\mathbb{S}$ be an Elgot monad. By Theorem 5.7, $\mathbb{S}$ is an iteration-congruent retract of $\mathbb{S}^\nu$ with $S^\nu X=\nu\gamma.\,S(X+\gamma)$; specifically, $\upsilon=\rhd\boldsymbol{\eta}\colon S\to S^\nu$ is a left inverse to $\rho=\mathsf{out}^\dagger\colon S^\nu\to S$.

First of all, it is easy to see that

$$\rho\,\rhd=[\eta,\rho]^\star\,\mathsf{out}\,\rhd=[\eta,\rho]^\star\eta\,\mathsf{in}_2=\rho.$$

Moreover, we need to show the axioms of $(\text{–})^\nu$-algebras:

$$\rho\boldsymbol{\eta}=\mathsf{id}\qquad\text{and}\qquad\rho\boldsymbol{\mu}=\rho\rho^\nu \tag{5.16}$$

where $\rho^\nu=\mathsf{coit}(\rho\,\mathsf{out})\colon S^{\nu\nu}\to S^\nu$. For the left axiom, we readily have $\mathsf{id}=\rho\upsilon=\rho\rhd\boldsymbol{\eta}=\rho\boldsymbol{\eta}$. The right axiom is shown as follows:

$$\rho\boldsymbol{\mu}\ \overset{(i)}{=\!=}\ \rho[\eta,\,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}\,\mathsf{out}\ \overset{(ii)}{=\!=}\ \rho\big(\mathsf{out}^{-1}\,S(\mathsf{in}_1+\eta\,\mathsf{in}_2)\rho\,\mathsf{out}\big)^\ddagger\ \overset{(iii)}{=\!=}\ \rho\rho^\nu.$$

To show step (i), first observe that on the one hand

$$\boldsymbol{\mu}=\mathsf{coit}\big(S[\mathsf{id}+\mathsf{out},\mathsf{in}_2]\,\mathsf{out}\big)\ \mathsf{out}$$

Indeed, let $t=\mathsf{coit}\big(S[\mathsf{id}+\mathsf{out},\mathsf{in}_2]\,\mathsf{out}\big)$. Then

$$\begin{aligned}
\mathsf{out}\ t\ \mathsf{out} &= S(\mathsf{id}+t)\,S[\mathsf{id}+\mathsf{out},\mathsf{in}_2]\,\mathsf{out}\,\mathsf{out}\\
&= S[\mathsf{id}+t\,\mathsf{out},t\,\mathsf{in}_2]\,\mathsf{out}\,\mathsf{out}\\
&= S(\mathsf{id}+\ t\,\mathsf{out})\,S[\mathsf{id},\mathsf{in}_2\,\mathsf{out}^{-1}]\,\mathsf{out}\,\mathsf{out},
\end{aligned}$$

which means that $t\,\mathsf{out}$ satisfies the equation uniquely characterizing $\boldsymbol{\mu}$, hence $\boldsymbol{\mu}=t\,\mathsf{out}$. On the other hand, $[\eta,\,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}$ satisfies the equation characterizing t. In order to see this, note that

$$\mathsf{out}(\rhd\,\mathsf{out})^\ddagger=\ \eta\,\mathsf{in}_2[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}\,\mathsf{out}, \tag{5.17}$$

witnessed by the following calculation:

$$\begin{aligned}
&\mathsf{out}(\,\rhd\,\mathsf{out})^\ddagger\\
&\quad=\ \mathsf{out}[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}\,\rhd\,\mathsf{out}\\
&\quad= [\mathsf{out}[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger],\eta\,\mathsf{in}_2[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}]^\star\,\mathsf{out}\rhd\mathsf{out}\\
&\quad= [\mathsf{out}[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger],\eta\,\mathsf{in}_2[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}]^\star\eta\,\mathsf{in}_2\,\mathsf{out}\\
&\quad= \eta\,\mathsf{in}_2[\eta^\nu,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}\,\mathsf{out}
\end{aligned}$$

Therefore,

$$\mathsf{out}[\eta^\nu,\,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}=[\mathsf{out}[\eta^\nu,\,(\rhd\,\mathsf{out})^\ddagger],\eta\,\mathsf{in}_2[\eta^\nu,\,(\rhd\,\mathsf{out})^\ddagger]^{\bigstar}]^\star\,\mathsf{out}$$

$$
\begin{aligned}
&= [[\mathsf{out}\,\eta^\nu, \mathsf{out}\,(\rhd\,\mathsf{out})^\ddagger], \eta\,\mathsf{in}_2[\eta^\nu,\ (\rhd\,\mathsf{out})^\ddagger]^\bigstar]^\star\,\mathsf{out}\\
&= [\eta\,(\mathsf{id} + [\eta^\nu,\ (\rhd\,\mathsf{out})^\ddagger]^\bigstar\ \mathsf{out}), \eta\,\mathsf{in}_2[\eta^\nu,\ (\rhd\,\mathsf{out})^\ddagger]^\bigstar]^\star\,\mathsf{out} && /\!\!/\ (5.17)\\
&= [\eta\,(\mathsf{id} + [\eta^\nu,\ (\rhd\,\mathsf{out})^\ddagger]^\bigstar)\,[\mathsf{id} + \mathsf{out}, \mathsf{in}_2]]^\star\,\mathsf{out}\\
&= S(\mathsf{id} + [\eta^\nu,\ (\rhd\,\mathsf{out})^\ddagger]^\bigstar)\,S[\mathsf{id} + \mathsf{out}, \mathsf{in}_2]\,\mathsf{out}\,.
\end{aligned}
$$

In summary we obtain

$$\boldsymbol{\mu} = \mathsf{coit}\big(S[\mathsf{id} + \mathsf{out}, \mathsf{in}_2]\,\mathsf{out}\big)\ \mathsf{out} = [\eta^\nu, (\rhd\,\mathsf{out})^\ddagger]^\bigstar\,\mathsf{out},$$

which justifies (i). Let us check (iii). Let us denote $\mathsf{out}^{-1}\,S(\mathsf{in}_1 + \eta^\nu\,\mathsf{in}_2)\rho\,\mathsf{out}$ by t. Then

$$
\begin{aligned}
\mathsf{out}\,t^\ddagger &= \mathsf{out}[\eta^\nu, t^\ddagger]^\bigstar\,\mathsf{out}^{-1}\,S(\mathsf{in}_1 + \eta^\nu\,\mathsf{in}_2)\rho\,\mathsf{out}\\
&= [\mathsf{out}[\eta^\nu, t^\ddagger], \eta\,\mathsf{in}_2[\eta^\nu, t^\ddagger]^\bigstar]^\star S(\mathsf{in}_1 + \eta^\nu\,\mathsf{in}_2)\rho\,\mathsf{out}\\
&= [\mathsf{out}\,\eta^\nu, \eta\,\mathsf{in}_2\,t^\ddagger]^\star\rho\,\mathsf{out}\\
&= S(\mathsf{id} + t^\ddagger)\rho\,\mathsf{out},
\end{aligned}
$$

and therefore $t^\ddagger$ satisfies the equation characterizing ρ^ν, hence $\rho^\nu = t^\ddagger$. Finally, we proceed with the proof of (ii). Using the fact that ρ is a monad morphism and that it cancels $\rhd$, we obtain that

$$
\begin{aligned}
\rho[\eta, (\rhd\,\mathsf{out})^\ddagger]^\bigstar\ \mathsf{out} &= [\eta, \rho\,(\rhd\,\mathsf{out})^\ddagger]^\bigstar\rho\ \mathsf{out}\\
&= [\eta, \rho\,(\rhd\,\mathsf{out})^\ddagger]^\bigstar\rho\rhd\ \mathsf{out}\\
&= \rho[\eta,\ (\rhd\,\mathsf{out})^\ddagger]^\bigstar\rhd\ \mathsf{out}\\
&= \rho(\rhd\,\mathsf{out})^\ddagger. && /\!\!/\ \text{fixpoint}
\end{aligned}
$$

In order to finish the proof of (ii), it suffices to check that

$$\rho\rhd\mathsf{out} = \rho\,\mathsf{out}^{-1}\,S(\mathsf{in}_1 + \eta\,\mathsf{in}_2)\rho\,\mathsf{out} \tag{5.18}$$

and call the assumption that ρ is an iteration congruence. The proof of (5.18) runs as follows:

$$
\begin{aligned}
\rho\,\mathsf{out}^{-1}\,S(\mathsf{in}_1 + \eta\,\mathsf{in}_2)\rho\,\mathsf{out} &= [\eta, \rho]^\star\,\mathsf{out}\,\mathsf{out}^{-1}\,S(\mathsf{in}_1 + \eta\,\mathsf{in}_2)\rho\,\mathsf{out}\\
&= [\eta, \rho]^\star S(\mathsf{in}_1 + \eta\,\mathsf{in}_2)\rho\,\mathsf{out}\\
&= [\eta\,\mathsf{in}_1, \rho\eta\,\mathsf{in}_2]^\star\rho\,\mathsf{out}\\
&= \rho\,\mathsf{out}\\
&= \rho\rhd\mathsf{out}\,.
\end{aligned}
$$

We have thus proved the claimed inclusion on objects. To extend the claim to morphisms, suppose that $\alpha\colon \mathbb{S} \to \mathbb{T}$ is an Elgot monad morphism, i.e. a monad morphism such that $\alpha f^\dagger = (\alpha f)^\dagger$, and let us show that it is also a morphism of the corresponding $(\text{–})^\nu$-algebras, i.e. $\alpha\rho = \rho\alpha^\nu$. Indeed, on the one hand $\alpha\rho = \alpha\,\mathsf{out}^\dagger = (\alpha\,\mathsf{out})^\dagger$, and also on the other hand, by uniformity of $(\text{–})^\dagger$, $\rho\alpha^\nu = \mathsf{out}^\dagger\,\alpha^\nu = (\alpha\,\mathsf{out})^\dagger$, since $\mathsf{out}\,\alpha^\nu = T(\mathsf{id} + \alpha^\nu)\alpha\,\mathsf{out}$.

We proceed with the converse inclusion, i.e. from $(\text{–})^\nu$-algebras to Elgot monads. So assume that $(\mathbb{S}, \rho)$ is a $(\text{–})^\nu$-algebra, i.e. the laws (5.16) are satisfied, and $\rho\rhd = \rho$. We claim that $\mathbb{S}$ equipped with the iteration operation $f^\dagger = \rho(\mathsf{coit}\,f)$ is an Elgot monad. The corresponding axioms are verified as follows.

- *Fixpoint.* Let $f\colon X \to S(Y+X)$. Then $f^\ddagger = \rho(\mathsf{coit}\, f)$ and hence

$$\begin{aligned} f^\ddagger &= \rho(\mathsf{coit}\, f) \\ &= [\eta,\rho]^\star\, \mathsf{out}\,(\mathsf{coit}\, f) \\ &= [\eta,\rho]^\star S(\mathsf{id} + \mathsf{coit}\, f) f && /\!\!/\ (5.15) \\ &= [\eta, \rho\, \mathsf{coit}\, f]^\star f \\ &= [\eta, f^\ddagger]^\star f. \end{aligned}$$

- *Naturality.* Let $f\colon X \to S(Y+X)$ and $g\colon Y \to SZ$. Then

$$g^\star f^\dagger = g^\star \rho\,(\mathsf{coit}\, f) = \rho\,(\boldsymbol{\eta} g)^{\bigstar}\,(\mathsf{coit}\, f),$$
$$([(S\,\mathsf{in}_1)\, g, \eta\,\mathsf{in}_2]^\star f)^\dagger = \rho\,\mathsf{coit}([(S\,\mathsf{in}_1)\, g, \eta\,\mathsf{in}_2]^\star f).$$

We are left to show that $(\boldsymbol{\eta} g)^{\bigstar}(\mathsf{coit}\, f)$ satisfies the equation for $\mathsf{coit}([(S\,\mathsf{in}_1)\, g, \eta\,\mathsf{in}_2]^\star f)$. This runs as follows:

$$\begin{aligned} \mathsf{out}(\boldsymbol{\eta} g)^{\bigstar}(\mathsf{coit}\, f) &= [\mathsf{out}\,\boldsymbol{\eta}\, g, \eta\,\mathsf{in}_2(\boldsymbol{\eta} g)^{\bigstar}]^\star\, \mathsf{out}\,(\mathsf{coit}\, f) \\ &= [(S\,\mathsf{in}_1) g, \eta\,\mathsf{in}_2(\boldsymbol{\eta} g)^{\bigstar}]^\star S(\mathsf{id} + \mathsf{coit}\, f) f \\ &= [(S\,\mathsf{in}_1) g, \eta\,\mathsf{in}_2(\boldsymbol{\eta} g)^{\bigstar}(\mathsf{coit}\, f)]^\star f \\ &= S(\mathsf{id} + (\boldsymbol{\eta} g)^{\bigstar}(\mathsf{coit}\, f))\, [(S\,\mathsf{in}_1)\, g, \eta\,\mathsf{in}_2]^\star f. \end{aligned}$$

- *Codiagonal.* Let $f\colon X \to S((Y+X)+X)$. Observe that since

$$\begin{aligned} &\mathsf{out}\,(\mathsf{coit}(\rho\ \mathsf{out}))\,(\mathsf{coit}(\mathsf{coit}\, f)) \\ &= S(\mathsf{id} + \mathsf{coit}(\rho\ \mathsf{out}))\,\rho\ \mathsf{out}\,(\mathsf{coit}(\mathsf{coit}\, f)) \\ &= S(\mathsf{id} + \mathsf{coit}(\rho\ \mathsf{out}))\,\rho\, S^\nu(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f))\,(\mathsf{coit}\, f) \\ &= S(\mathsf{id} + \mathsf{coit}(\rho\ \mathsf{out}))\, S(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f))\,\rho\,(\mathsf{coit}\, f) && /\!\!/\ \text{naturality of } \rho \\ &= S(\mathsf{id} + (\mathsf{coit}(\rho\ \mathsf{out}))\,(\mathsf{coit}(\mathsf{coit}\, f)))\,\rho\,(\mathsf{coit}\, f), \end{aligned}$$

we have that

$$\mathsf{coit}(\rho\,(\mathsf{coit}\, f)) = (\mathsf{coit}(\rho\ \mathsf{out}))\,(\mathsf{coit}(\mathsf{coit}\, f)) \tag{5.19}$$

by uniqueness of final morphisms. Thus,

$$\begin{aligned} f^{\dagger\dagger} &= \rho\,(\mathsf{coit}(\rho(\mathsf{coit}\, f))) \\ &= \rho\,(\mathsf{coit}(\rho\ \mathsf{out}))(\mathsf{coit}(\mathsf{coit}\, f)) && /\!\!/\ (5.19) \\ &= \rho\,\rho^\nu\,(\mathsf{coit}(\mathsf{coit}\, f)) && /\!\!/\ \text{definition of } \rho^\nu \\ &= \rho\,\boldsymbol{\mu}\,(\mathsf{coit}(\mathsf{coit}\, f)). && /\!\!/\ (5.16) \end{aligned}$$

Since by definition, $(S[\mathsf{id},\mathsf{in}_2]\, f)^\dagger = \rho\,\mathsf{coit}(S[\mathsf{id},\mathsf{in}_2]\, f)$, we are only left to check that $\mathsf{coit}(S[\mathsf{id},\mathsf{in}_2]\, f) = \boldsymbol{\mu}\,(\mathsf{coit}(\mathsf{coit}\, f))$. This is easy to establish directly by showing that the right-hand side satisfies the equation characterizing the left-hand side:

$$\begin{aligned} &\mathsf{out}\,\boldsymbol{\mu}\,\mathsf{coit}(\mathsf{coit}\, f) \\ &= S(\mathsf{id} + \boldsymbol{\mu})\, S[\mathsf{id}, \mathsf{in}_2\,\mathsf{out}^{\text{-}1}]\,\mathsf{out}\,\mathsf{out}\,\mathsf{coit}(\mathsf{coit}\, f) \\ &= S[\mathsf{id} + \boldsymbol{\mu}, \mathsf{in}_2\,\boldsymbol{\mu}\,\mathsf{out}^{\text{-}1}]\,\mathsf{out}\,\mathsf{out}\,(\mathsf{coit}(\mathsf{coit}\, f)) \\ &= S[\mathsf{id} + \boldsymbol{\mu}, \mathsf{in}_2\,\boldsymbol{\mu}\,\mathsf{out}^{\text{-}1}]\ \mathsf{out}\ S^\nu(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f))\ \mathsf{coit}\, f \\ &= S[\mathsf{id} + \boldsymbol{\mu}, \mathsf{in}_2\,\boldsymbol{\mu}\,\mathsf{out}^{\text{-}1}]\, S((\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f)) + S^\nu(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f)))\ \mathsf{out}\ \mathsf{coit}\, f \end{aligned}$$

$$
\begin{aligned}
&= S[\mathsf{id} + \boldsymbol{\mu}, \mathsf{in}_2\, \boldsymbol{\mu}\, \mathsf{out}^{-1}]\, S((\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f)) + S^{\nu}(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f)))\, S(\mathsf{id} + \mathsf{coit}\, f) f \\
&= S[\mathsf{id} + \boldsymbol{\mu}\, (\mathsf{coit}(\mathsf{coit}\, f)), \mathsf{in}_2\ \boldsymbol{\mu}\ \mathsf{out}^{-1}\, S^{\nu}(\mathsf{id} + \mathsf{coit}(\mathsf{coit}\, f))(\mathsf{coit}\, f)]\, f \\
&= S[\mathsf{id} + \boldsymbol{\mu}\, (\mathsf{coit}(\mathsf{coit}\, f)), \mathsf{in}_2\ \boldsymbol{\mu}\, (\mathsf{coit}(\mathsf{coit}\, f))]\, f \\
&= S(\mathsf{id} + \boldsymbol{\mu}\, (\mathsf{coit}(\mathsf{coit}\, f)))\, S[\mathsf{id}, \mathsf{in}_2]\, f.
\end{aligned}
$$

- *Uniformity.* Let $f\colon X \to T(Y+X)$, $g\colon Z \to S(Y+Z)$, $h\colon Z \to X$ and suppose that $f\,h = T(\mathsf{id} + h)\, g$. If follows standardly by uniqueness of final coalgebra morphisms that $(\mathsf{coit}\, f)\, h = \mathsf{coit}\, g$ and therefore

$$f^{\ddagger}\, h = \rho\, (\mathsf{coit}\, f)\, h = \rho\, (\mathsf{coit}\, g) = g^{\ddagger}.$$

Finally, let us check that every $(\text{–})^{\nu}$-algebra morphism $\alpha\colon \mathbb{T} \to \mathbb{S}$ is an Elgot monad morphism. By assumption we have that $\alpha\rho = \rho\alpha^{\nu}$, and therefore, for every $f\colon X \to S(Y+X)$, $\alpha f^{\dagger} = \alpha\rho(\mathsf{coit}\, f) = \rho\alpha^{\nu}(\mathsf{coit}\, f) = \rho\, \mathsf{coit}(\alpha\, \mathsf{out})(\mathsf{coit}\, f)$. It is then straightforward to verify that $\rho\, \mathsf{coit}(\alpha\, \mathsf{out})(\mathsf{coit}\, f) = \rho\, \mathsf{coit}(\alpha f) = (\alpha f)^{\dagger}$. □

6. A Sandwich Theorem for Elgot Monads

As an application of Theorem 5.7, we show that sandwiching an Elgot monad between a pair of adjoint functors again yields an Elgot monad. A similar result has previously been shown for completely iterative monads [31]; this result generalizes straightforwardly to guarded iterative monads:

Theorem 6.1. *Let $F\colon \mathbf{C} \to \mathbf{D}$ and $U\colon \mathbf{D} \to \mathbf{C}$ be a pair of adjoint functors with associated natural isomorphism $\Phi\colon \mathbf{D}(FX, Y) \to \mathbf{C}(X, UY)$, and let $\mathbb{T}$ be a guarded iterative monad on $\mathbf{D}$. Then the monad induced on the composite functor UTF is guarded iterative, with the guardedness relation defined by taking $f\colon X \to_{\sigma} UTFY$ if and only if $\Phi^{-1} f\colon FX \to_{\sigma} TFY$, and unique solutions given by $f \mapsto \Phi((\Phi^{-1} f)^{\dagger})$.*

Proof. First, we need to verify that the guardedness relation defined in the claim satisfies the rules from Definition 3.1. Note that since left adjoints preserve coproducts (LAPC), we can assume w.l.o.g. that $F(X+Y) = FX + FY$.

- **(trv)** Let $f\colon X \to UTFY$ be a morphism. By **(trv)** for $\mathbb{T}$, we have $(T\,\mathsf{in}_1)(\Phi^{-1} f)\colon FX \to_{\sigma} T(FY + FX)$. Then, the following holds:

$$
\begin{aligned}
(T\,\mathsf{in}_1)(\Phi^{-1} f) &= (TF\,\mathsf{in}_1)(\Phi^{-1} f) && /\!/ \text{ LAPC} \\
&= \Phi^{-1}((UTF\,\mathsf{in}_1) f) && /\!/ \ \Phi \text{ is a natural isomorphism}
\end{aligned}
$$

Thus, $\Phi^{-1}((UTF\,\mathsf{in}_1) f)\colon FX \to_{\sigma} T(FY + FX)$, so $(UTF\,\mathsf{in}_1) f\colon X \to_{\sigma} UTF(Y+X)$.

- **(par)** Let $f\colon X \to_{\sigma} UTFZ$ and $g\colon Y \to_{\sigma} UTFZ$. This means that $\Phi^{-1} f\colon FX \to_{\sigma} TFZ$ and $\Phi^{-1} g\colon FY \to_{\sigma} TFZ$, hence, by **(par)** for $\mathbb{T}$, $[\Phi^{-1} f, \Phi^{-1} g]\colon FX + FY \to_{\sigma} TFZ$. By LAPC, we have $[\Phi^{-1} f, \Phi^{-1} g] = \Phi^{-1}[f, g]\colon F(X+Y) \to_{\sigma} TFZ$, so $[f, g]\colon X + Y \to_{\sigma} UTFZ$.
- **(cmp)** Let $f\colon X \to_2 UTF(Y+Z)$, $g\colon Y \to_{\sigma} UTFV$, and $h\colon Z \to UTFV$ be morphisms. Then, by **(cmp)** for $\mathbb{T}$, we obtain that $[\Phi^{-1} g, \Phi^{-1} h]^{\star}(\Phi^{-1} f)\colon FX \to_{\sigma} TFV$. Then, the following holds:

$$
\begin{aligned}
&[\Phi^{-1} g, \Phi^{-1} h]^{\star}(\Phi^{-1} f) \\
&= \mu^{\mathbb{T}}(T[\Phi^{-1} g, \Phi^{-1} h])(\Phi^{-1} f) \\
&= \mu^{\mathbb{T}}(\Phi^{-1}(UT[\Phi^{-1} g, \Phi^{-1} h]) f) && /\!/ \ \Phi \text{ is a nat. iso.}
\end{aligned}
$$

$$
\begin{aligned}
&= \Phi^{-1}((U\mu^{\mathbb{T}})(UT[\Phi^{-1}g, \Phi^{-1}h])f) && /\!/ \ \Phi \text{ is a nat. iso.}\\
&= \Phi^{-1}((U\mu^{\mathbb{T}})(UT\Phi^{-1}[g,h])f) && /\!/ \ \text{LAPC}\\
&= \Phi^{-1}((U\mu^{\mathbb{T}})(UT\Phi^{-1}\mathsf{id})(UTF[g,h])f) && /\!/ \ \Phi \text{ is a nat. iso.}\\
&= \Phi^{-1}(\mu^{UTF}(UTF[g,h])f)\\
&= \Phi^{-1}([g,h]^{\star}f).
\end{aligned}
$$

Thus, $\Phi^{-1}([g,h]^{\star}f)\colon X \to_\sigma UTF(Y+X)$ in $\mathbb{T}$, so $[g,h]^{\star}f\colon FX \to_\sigma TF(Y+X)$ in the monad on UTF.

This means that if $f\colon X \to_2 UTF(Y+X)$, then $\Phi^{-1}f\colon FX \to_2 T(FY+FX)$, so $\Phi^{-1}f$ has a unique solution due to the fact that $\mathbb{T}$ is guarded iterative. The rest of the proof is the same as for Theorem 3.1 in [31]. □

Now, to obtain a similar result for Elgot monads, we can easily combine Theorems 5.7 and 6.1 without having to verify the equational properties by hand.

Theorem 6.2. *With an adjunction as in Theorem 6.1, let $\mathbb{S}$ be an Elgot monad on $\mathbf{D}$. Then, the monad induced on the composite USF is an Elgot monad.*

Proof. By Theorem 5.7, there exist a guarded iterative monad $\mathbb{T}$ and an iteration-congruent retraction $\rho\colon \mathbb{T} \to \mathbb{S}$. By Theorem 6.1, the monad induced on UTF is guarded iterative. Thus, it is enough to show that $U\rho F : UTF \to USF$ is an iteration-congruent retraction.

- It is a retraction, since retractions are preserved by all functors.
- To see that it is guarded, let $f\colon X \to USFY$ be σ-guarded. By definition, this means that $\Phi^{-1}f\colon FX \to SFY$ is σ-guarded in $\mathbb{S}$. Since ρ is a guarded retraction, it follows that $\upsilon(\Phi^{-1}f)$, for ρ's family of sections υ, is also σ-guarded. By the fact that Φ is a natural isomorphism, we obtain $\upsilon(\Phi^{-1}f) = \Phi^{-1}((U\upsilon)f)$, hence, by definition, $(U\upsilon)f$ is also σ-guarded.
- To see that $U\rho F$ is an iteration congruence, let us denote by $(\text{–})^{\dagger}$ the solution in $\mathbb{T}$, and by $(\text{–})^{\ddagger}$ the solution in the monad on UTF. Let $f, g\colon X \to_2 UTF(X+Y)$ be morphisms such that $(U\rho)f = (U\rho)g$. First, using this and the fact that Φ is a natural isomorphism, we obtain the following:

$$\rho(\Phi^{-1}f) = \Phi^{-1}((U\rho)f) = \Phi^{-1}((U\rho)g) = \rho(\Phi^{-1}g)$$

Thus, by the fact that ρ is an iteration congruence, we obtain that $\rho(\Phi^{-1}f)^{\dagger} = \rho(\Phi^{-1}g)^{\dagger}$. Now, we check that $U\rho F$ is an iteration congruence:

$$
\begin{aligned}
(U\rho)f^{\ddagger} &= (U\rho)(\Phi(\Phi^{-1}f)^{\dagger})\\
&= \Phi(\rho(\Phi^{-1}f)^{\dagger}) && /\!/ \ \Phi \text{ is a natural isomorphism}\\
&= \Phi(\rho(\Phi^{-1}g)^{\dagger}) && /\!/ \ \text{the above}\\
&= (U\rho)(\Phi(\Phi^{-1}g)^{\dagger}) && /\!/ \ \Phi \text{ is a natural isomorphism}\\
&= (U\rho)g^{\ddagger}
\end{aligned}
$$
□

Example 6.3 (From Metric to CPO-based Iteration)**.** As an example exhibiting sandwiching as well as the setting of Theorem 5.7, we compare two iteration operators on **Set** that arise from different fixed point theorems: Banach's, for complete metric spaces, and Kleene's, for complete partial orders, respectively. We obtain the first operator by sandwiching Escardo's *metric lifting monad* $\mathbb{S}$ [12] in the adjunction between sets and bounded complete ultrametric spaces (which forgets the metric in one direction and takes discrete spaces in the other), obtaining a monad $\bar{\mathbb{S}}$ on **Set**. Given a bounded complete metric space (X, d), $S(X, d)$ is a metric on the set $(X \times \mathbb{N}) \cup \{\bot\}$. As we show in the appendix, $\mathbb{S}$ is guarded iterative if we define $f\colon (X, d) \to S(Y, d')$ to be σ-guarded if $k > 0$ whenever $f(x) = (\sigma(y), k)$. By Theorem 6.1, $\bar{\mathbb{S}}$ is also guarded iterative (of course, this can also be shown directly). The second monad arises by sandwiching the identity monad on cpos with bottom in the adjunction between sets and cpos with bottom that forgets the ordering in one direction and adjoins bottom in the other, obtaining an Elgot monad $\mathbb{L}$ on Set according to Theorem 6.2. The latter is unsurprising, of course, as $\mathbb{L}$ is just the maybe monad $LX = X + 1$.

The monad $\bar{\mathbb{S}}$ keeps track of the number of steps needed to obtain the final result. We have an evident extensional collapse map $\rho\colon \bar{\mathbb{S}} \to \mathbb{L}$, which just forgets the number of steps. One can show that ρ is in fact an iteration-congruent retraction, so we obtain precisely the situation of Theorem 5.7. Technical details are in the appendix.

7. Related Work

Alternatively to our guardedness relation on Kleisli morphisms, guardedness can be formalized using type constructors [28] or, categorically, functors, as in *guarded fixpoint categories* [24]. Roughly speaking, in such settings a morphism $X \to Y + Z$ is guarded in Z if it factors through a morphism $X \to Y + \blacktriangleright Z$ where $\blacktriangleright$ is a functor or type constructor to be thought of as isolating the guarded inhabitants of a type. The functorial approach, giving rise to *guarded fixpoint categories*, covers also total guardedness, like we do. Our approach is slightly more fine-grained, and in particular natively supports the two variants of the dinaturality axiom (Figure 2), which, e.g., in guarded fixpoint categories require additional assumptions [24, Proposition 3.15] akin to the one we discuss in Remark 3.7. In our own subsequent work, we have generalized the notion of abstract guardedness from co-Cartesian to symmetric monoidal categories [17], where guardedness becomes a more symmetric concept: among morphisms $X \otimes Y \to Z \otimes W$, where $\otimes$ is the monoidal structure, one distinguishes morphisms that are (simultaneously) *unguarded* in the input A and *guarded* in the output D.

A result that resembles our Theorem 5.15, due to Adámek et al. [3], states roughly that if **C** is locally finitely presentable and hyperextensive (a property imposing certain compatibility constraints between pullbacks and countable coproducts, satisfied, e.g., over sets and over complete partial orders), then the finitary Elgot monads are the algebras for a monad on the category of endofunctors given by $H \mapsto L_H = \rho\gamma.\, (\text{–} + 1 + H\gamma)$ where ρ takes *rational fixpoints* (i.e. final coalgebras among those where every point generates a finite subcoalgebra); that is, in the mentioned setting, finitary Elgot monads are monadic over endofunctors. Besides Theorem 5.15 making fewer assumptions on **C**, the key difference (indicated already in Remark 5.16) is that, precisely by dint of the mentioned result, L_H is already a finitary Elgot monad (namely, the free finitary Elgot monad over H); contrastingly, we characterize Elgot monads as quotients of *guarded iterative* monads, i.e. of monads where guarded recursive definitions have *unique* fixpoints.

8. Conclusions and Further Work

We have given a unified account of monad-based guarded and unguarded iteration by axiomatizing the notion of guardedness to cover standard definitions of guardedness, and additionally, as a corner case, what we call *total guardedness*, i.e. the situation when all morphisms are declared to be guarded. We thus obtain a common umbrella for *guarded* *iterative monads*, i.e. monads with unique iterates of guarded morphisms, and Elgot monads, i.e. totally guarded monads satisfying Elgot's classical laws of iteration. We reinforce the view that the latter constitute a canonical model for monad-based unguarded iteration by establishing the following equivalent characterizations: Provided requisite final coalgebras exist, a monad $\mathbb{T}$ is Elgot iff it satisfies one of the following equivalent conditions:

- it satisfies the quasi-equational theory of iteration [2, 18] (definition);
- it is an iteration-congruent retract of a guarded iterative monad (Theorem 5.7);
- it is an algebra $(\mathbb{T}, \rho)$ of the monad $T \mapsto \nu\gamma.\, T(X + \gamma)$ in the category of monads satisfying a natural delay cancellation condition (Theorem 5.15).

In future work, we aim to investigate further applications of this machinery, in particular to examples which did not fit previous formalizations. One prospective target is suggested by work of Nakata and Uustalu [29], who give a coinductive big-step trace semantics for a while-language. We conjecture that this work has an implicit guarded iterative monad $\mathbb{T}_{\mathbb{R}}$ under the hood, for which guardedness cannot be defined using the standard argument based on a final coalgebra structure of the monad because $\mathbb{T}_{\mathbb{R}}$ is not a final coalgebra. Moreover, we aim to extend the treatment of iteration in finite trace semantics via iteration-congruent retractions (Example 5.8) to infinite traces, possibly taking orientation from recent work on coalgebraic infinite trace semantics [34].

In type theory, there is growing interest in forming an extensional quotient of the delay monad [9, 4]. It is shown in [9] that under certain reasonable conditions, a suitable collapse of the delay monad by removing delays is again a monad; however, the proof is already quite complex, and proving directly that the collapse is in fact an Elgot monad, as one would be inclined to expect, seems daunting. We expect that Theorem 5.15 may shed light on this issue. A natural question that arises in this regard is whether the subcategory of $(\text{--})^\nu$-algebras figuring in the theorem is reflexive. A positive answer would provide a means of constructing canonical quotients of $(\text{--})^\nu$-algebras (such as the delay monad) with the results automatically being Elgot monads.

Acknowledgements. We would like to thank the anonymous referees for their thorough attention to the text and their useful comments on improving the presentation.

REFERENCES

[1] P. Aczel, J. Adámek, S. Milius, and J. Velebil. Infinite trees and completely iterative theories: a coalgebraic view. *Theor. Comput. Sci.*, 300(1–3):1–45, 2003.
[2] J. Adámek, S. Milius, and J. Velebil. Equational properties of iterative monads. *Inf. Comput.*, 208(12):1306–1348, 2010.
[3] J. Adámek, S. Milius, and J. Velebil. Elgot theories: a new perspective of the equational properties of iteration. *Math. Struct. Comput. Sci.*, 21(2):417–480, 2011.
[4] T. Altenkirch, N. Danielsson, and N. Kraus. Partiality, revisited - the partiality monad as a quotient inductive-inductive type. In J. Esparza and A. Murawski, eds., *Foundations of Software Science and Computation Structures, FOSSACS 2017*, vol. 10203 of *LNCS*, pp. 534–549, 2017.
[5] J. Bergstra, A. Ponse, and S. Smolka, eds. *Handbook of Process Algebra*. Elsevier, 2001.
[6] S. Bloom and Z. Ésik. *Iteration theories: the equational logic of iterative processes*. Springer, 1993.
[7] V. Capretta. General recursion via coinductive types. *Log. Meth. Comput. Sci.*, 1(2), 2005.
[8] A. Carboni, S. Lack, and R. Walters. Introduction to extensive and distributive categories. *J. Pure Appl. Algebra*, 84:145–158, 1993.

[9] J. Chapman, T. Uustalu, and N. Veltri. Quotienting the delay monad by weak bisimilarity. In M. Leucker, C. Rueda, and F. Valencia, eds., *Theoretical Aspects of Computing, ICTAC 2015*, vol. 9399 of *LNCS*, pp. 110–125. Springer, 2015.

[10] C. Elgot. Monadic computation and iterative algebraic theories. In H. Rose and J. Shepherdson, eds., *Logic Colloquium 1973*, vol. 80 of *Studies in Logic and the Foundations of Mathematics*, pp. 175–230. Elsevier, 1975.

[11] C. Elgot, S. Bloom, and R. Tindell. On the algebraic atructure of rooted trees. *J. Comput. Syst. Sci.*, 16(3):362–399, 1978.

[12] M. Escardó. A metric model of PCF. In *Realizability Semantics and Applications*, 1999.

[13] Z. Ésik. Axiomatizing iteration categories. *Acta Cybern.*, 14(1):65–82, 1999.

[14] Z. Ésik and S. Goncharov. Some remarks on Conway and iteration theories. *CoRR*, abs/1603.00838, 2016.

[15] S. Goncharov, S. Milius, and C. Rauch. Complete Elgot monads and coalgebraic resumptions. In L. Birkedal, ed., *Mathematical Foundations of Programming Semantics, MFPS 2016*, vol. 325 of *ENTCS*, pp. 147–168. Elsevier, 2016.

[16] S. Goncharov and L. Schröder. A coinductive calculus for asynchronous side-effecting processes. *Inf. Comput.*, 231:204–232, 2013.

[17] S. Goncharov and L. Schröder. Guarded traced categories. In C. Baier and U. Dal Lago, eds., *Foundations of Software Science and Computation Structures, FOSSACS 2018*, vol. 10803 of *LNCS*, pp. 313–330. Springer, 2018.

[18] S. Goncharov, L. Schröder, C. Rauch, and J. Jakob. Unguarded recursion on coinductive resumptions. *Log. Methods Comput. Sci.*, 14(3), 2018.

[19] S. Goncharov, L. Schröder, C. Rauch, and M. Piróg. Unifying guarded and unguarded iteration. In J. Esparza and A. Murawski, eds., *Foundations of Software Science and Computation Structures, FoSSaCS 2017*, vol. 10203 of *LNCS*, pp. 517–533. Springer, 2017.

[20] I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Log. Meth. Comput. Sci.*, 3(4), 2007.

[21] W. Lawvere. Functorial semantics of algebraic theories. *Proc. Natl. Acad. Sci. USA*, 50(5):869–872, 1963.

[22] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2nd edition, 1998.

[23] S. Milius. Completely iterative algebras and completely iterative monads. *Inf. Comput.*, 196(1):1–41, 2005.

[24] S. Milius and T. Litak. Guard your daggers and traces: Properties of guarded (co-)recursion. *Fund. Inform.*, 150:407–449, 2017.

[25] R. Milner. *Communication and concurrency*. Prentice-Hall, 1989.

[26] E. Moggi. A modular approach to denotational semantics. In D. Pitt, P.-L. Curien, S. Abramsky, A. Pitts, A. Poigné, and D. Rydeheard, eds., *Category Theory and Computer Science, CTCS 1991*, vol. 530 of *LNCS*, pp. 138–139. Springer, 1991.

[27] E. Moggi. Notions of computation and monads. *Inf. Comput.*, 93:55–92, 1991.

[28] H. Nakano. A modality for recursion. In *Logic in Computer Science, LICS 2000*, pp. 255–266. IEEE Computer Society, 2000.

[29] K. Nakata and T. Uustalu. A Hoare logic for the coinductive trace-based big-step semantics of while. *Log. Methods Comput. Sci.*, 11(1), 2015.

[30] M. Piróg and J. Gibbons. The coinductive resumption monad. In B. Jacobs, ed., *Mathematical Foundations of Programming Semantics, MFPS 2014*, vol. 308 of *ENTCS*, pp. 273–288, 2014.

[31] M. Piróg and J. Gibbons. Monads for behaviour. In D. Kozen, ed., *Mathematical Foundations of Programming Semantics, MFPS 2013*, vol. 298 of *ENTCS*, pp. 309–324, 2015.

[32] A. Simpson and G. Plotkin. Complete axioms for categorical fixed-point operators. In *Logic in Computer Science, LICS 2000*, pp. 30–41, 2000.

[33] M. Smyth. Topology. In *Handbook of Logic in Computer Science*, vol. 1, pp. 641–761. Clarendon Press, 1992.

[34] N. Urabe and I. Hasuo. Coalgebraic infinite traces and kleisli simulations. In L. Moss and P. Sobocinski, eds., *Algebra and Coalgebra in Computer Science, CALCO 2015*, vol. 35 of *LIPIcs*, pp. 320–335. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[35] T. Uustalu. Generalizing substitution. *ITA*, 37(4):315–336, 2003.

[36] T. Uustalu and V. Vene. Primitive (Co)Recursion and Course-of-Value (Co)Iteration, Categorically. *Informatica (Lithuanian Academy of Sciences)*, 10(1):5–26, 1999.

Appendix A. Details of Example 6.3

As an example of the setting described in Theorem 5.7, we compare two iteration operators on **Set** that arise from different fixed point theorems: Banach's for complete metric spaces and Kleene's for complete partial orders, respectively. The first operator keeps track of the number of steps needed to obtain the final result. We show that its extensional collapse, defined as a morphism that forgets the number of steps, is an iteration-congruent retraction with the second operator as the retract.

First, we consider the category **CbUMet** of complete 1-bounded ultrametric spaces and nonexpansive maps. Note that **CbUMet** has coproducts and is Cartesian closed [33]. Following Escardó [12], one has a monad T on **CbUMet** given on objects as $T(A, d) = ((A \times \mathbb{N}) \cup \{\infty\}, d')$, with d' given by

$$d'(\infty,\infty) = 0 \qquad d'((x,k),\infty) = d'(\infty,(x,k)) = (1/2)^k$$
$$d'((x,k),(y,k)) = (1/2)^k d(x,y) \qquad d'((x,k),(y,t)) = (1/2)^{\min(k,t)} \text{ if } k \neq t$$

The monad structure on T is defined as expected, by $\eta(a) = (a, 0)$ and $f^\star(a, k) = (b, k + t)$ where $f(a) = (b, t)$

Theorem A.1. *The monad T is guarded iterative with $f\colon X \to TY$ being σ-guarded if for all x and y, $f(x) = (\sigma(y), k)$ implies $k > 0$.*

Proof. First, note that the product of (A, d_A) and (B, d_B) in **CbUMet** is given by $(A \times B, d_{A\times B})$, where

$$d_{A\times B}((x_1, y_1),(x_2, y_2)) = \max\{d_A(x_1, x_2), d_B(y_1, y_2)\}.$$

The exponential object is equal to $(B^A, d_{A\Rightarrow B})$ where

$$d_{A\Rightarrow B}(f, g) = \sup\{d_B(f(x), g(x)) \mid x \in A\}.$$

The coproduct is given by $(A + B, d_{A+B})$, where

$$d_{A+B}(p,q) = \begin{cases} d_A(x_1,x_2) & \text{if } p = \mathsf{in}_1\ x_1 \text{ and } q = \mathsf{in}_1\ x_2 \\ d_B(y_1,y_2) & \text{if } p = \mathsf{in}_2\ y_1 \text{ and } q = \mathsf{in}_2\ y_2 \\ 1 & \text{otherwise} \end{cases}$$

Now, we show that the monad $\mathbb{T}$ is guarded. The only nontrivial case is **(cmp)**. So, assume $([g, h]^\star f)(x) = (\sigma(y), k)$. We consider two cases:

- $f(x) = (\mathsf{in}_2\, z, k')$. Then, since f is in_2-guarded, $k' > 0$, so, by definition of $(-)^\star$, $k > 0$.
- $f(x) = (\mathsf{in}_1\, z, k')$. Then, $([g, h]^\star f)(x) = [g, h]^\star(\mathsf{in}_1\, z, k') = (\sigma(y), k' + k'')$, where $g(z) = (\sigma(y), k'')$. Since g is σ-guarded, $k'' > 0$, so $k = k' + k'' > 0$.

Given a guarded morphism $f\colon X \to_1 T(Y + X)$, we define the morphism $f^\dagger\colon X \to TY$ as the unique fixed point of the following map $\psi : (X \to TY) \to (X \to TY)$:

$$\psi(g) = [\eta, g]^\star f$$

One can easily see that any fixed point of ψ satisfies the fixed point identity, and that the uniqueness of such a fixed point gives us that f has a unique solution. We use Banach's theorem to achieve both.

By Banach's theorem, it is enough to show that ψ is contractive, that is, there exists a non-negative real $c < 1$ such that for maps $g, g'\colon X \to TY$, the following holds:

$$d_{X\Rightarrow TY}(\psi(g), \psi(g')) \leqslant c \cdot d_{X\Rightarrow TY}(g, g') \tag{A.1}$$

The left-hand side of the equation (A.1) is equal to:

$$d_{X\Rightarrow TY}(\psi(g),\psi(g')) = \sup\{d_{TY}(\psi(g)(x),\psi(g')(x)) \mid x \in X\}$$

In turn, the right-hand side is as follows:

$$\begin{aligned} c\cdot d_{X\Rightarrow TY}(g,g') &= c\cdot \sup\{d_{TY}(g(x),g'(x)) \mid x\in X\} \\ &= \sup\{c\cdot d_{TY}(g(x),g'(x)) \mid x \in X\}\end{aligned}$$

Thus, it is enough to show that for all $x \in X$, there exists $y \in X$ such that:

$$d_{TY}(\psi(g)(x),\psi(g')(x)) \leqslant c\cdot d_{TY}(g(y),g'(y))$$

We show this for $c = 1/2$. We consider two cases:

- $f(x) = (\mathsf{in}_1\, y, k)$ for some $y \in Y$ and $k \in \mathbb{N}$. Then, for all $g\colon X \to TY$, the following holds:

$$\begin{aligned}\psi(g)(x) &= ([\eta,g]^\star f)(x) \\ &= [\eta,g]^\star(\mathsf{in}_1\, y,k) \\ &= (y,k)\end{aligned}$$

So, the following holds:

$$\begin{aligned} d_{TY}(\psi(g)(x),\psi(g')(x)) &= d_{TY}((y,k),(y,k)) \\ &= 0 \\ &\leqslant (1/2)\cdot d_{TY}(g(x),g'(x))\end{aligned}$$

- $f(x) = (\mathsf{in}_2\, y, k+1)$ for some $y \in X$ and $k \in \mathbb{N}$ (the '+1' part follows from the fact that f is guarded). Assume that $g(y) = (z,t)$ for some z and t. Then

$$\begin{aligned}\psi(g)(x) &= ([\eta,g]^\star f)(x) \\ &= [\eta,g]^\star(\mathsf{in}_2\, y,k+1) \\ &= (z,t+k+1).\end{aligned}$$

Similarly, let $g'(y) = (z',t')$, and so $\psi(g')(x) = (z',t'+k+1)$. Then, it follows that:

$$\begin{aligned} d_{TY}(\psi(g)(x),\psi(g')(x)) &= d_{TY}((z,t+k+1),(z',t'+k+1)) \\ &= (1/2)^{k+1}\cdot d_{TY}((z,t),(z',t')) \\ &= (1/2)^{k}\cdot (1/2)\cdot d_{TY}((z,t),(z',t')) \\ &\leqslant (1/2)\cdot d_{TY}((z,t),(z',t')) \\ &= (1/2)\cdot d_{TY}(g(y),g'(y))\end{aligned}$$

□

We thus obtain a monad $U_\mathrm{D}TF_\mathrm{D}$ on **Set** by sandwiching T in the adjunction $F_\mathrm{D} \dashv U_\mathrm{D}$ where U_D is the forgetful functor **CbUMet** → **Set** and F_D takes discrete metrics. By Theorem 6.1, $U_\mathrm{D}TF_\mathrm{D}$ is guarded iterative.

For the second operator, let $\mathbf{Cpo}_\bot$ be the category of complete partial orders and continuous bottom-preserving functions. The identity on $\mathbf{Cpo}_\bot$ is an Elgot monad, hence, by Theorem 6.2, we obtain an Elgot monad $U_\mathrm{L}F_\mathrm{L}$ on **Set** by sandwiching in the adjunction $F_\mathrm{L} \dashv U_\mathrm{L}$ where U_L is the forgetful functor $\mathbf{Cpo}_\bot \to \mathbf{Set}$ and F_L adjoins bottom. The relation between the two monads on **Set** is an instance of our notion of iteration-congruent retraction:

Theorem A.2. *Define $\rho\colon U_{\mathrm{D}}TF_{\mathrm{D}} \to ULF_{\mathrm{L}}$ by $\rho(a,k) = a$ and $\rho(\infty) = \bot$. Then ρ is an iteration-congruent retraction with the section given by $\upsilon(a) = (a,1)$ and $\upsilon(\bot) = \infty$. Moreover, the respective iteration operators induced by ρ and the sandwich theorem coincide.*

Proof. It is trivial that ρ is a guarded retraction. To see that it is a iteration congruence, we first define an auxiliary relation: given two functions $f, h\colon X \to (B \times \mathbb{N}) \cup \{\infty\}$, we write $f \sim h$ if $f(x) = (a,k)$ for some $a \in B$, $k \in \mathbb{N}$ if and only if $h(x) = (a,k')$ for some $k' \in \mathbb{N}$ and $f(x) = \infty$ if and only if $h(x) = \infty$ (i.e. the two functions differ only in the number of steps needed to obtain the value). We also write $\psi_f(g) = [\eta, g]^\star f$ for the function ψ from the proof of Theorem A.1.

Given a 2-guarded function $f\colon X \to ((Y + X) \times \mathbb{N}) \cup \{\infty\}$, the function $f^\dagger$ can be defined as the unique fixed point of ψ_f (see the proof of Theorem A.1), which by Banach's fixed-point theorem is given by the limit of the sequence $W_0^f = c$ and $W_{(n+1)}^f = \psi_f(W_n^f)$, where c is the constant function $c(x) = \infty$. It is easy to see that for each x the sequence $W_n^f(x)$ stabilizes. Given a function h such that $f \sim h$, it is easy to show by induction that for every x, the sequence $W_n^f(x)$ stabilizes with (a,k) for some k if and only if $W_n^h(x)$ stabilizes with (a,k') for some k' at the same index n. Then, for all x, $f^\dagger(x) = (a,k)$ and $h^\dagger(x) = (a,k')$, so $\rho(f^\dagger(x)) = \rho(h^\dagger(x))$. Then, the result is obtained by noticing that for all f and h, $\rho f = \rho h$ implies $f \sim h$.

It is left to see that the solution operator that follows from the sandwich theorem and the one that follows from the iteration-congruent retraction coincide. Given $f\colon X \to (X + Y) \cup \{\bot\}$, its solution in the Elgot monad $U_{\mathrm{L}}F_{\mathrm{L}}$ is given by the fixed point of the equation $\phi(g) = [\eta, g]^\star f$, that is, by Kleene's theorem, by the limit of the sequence $W_0' = c'$ and $W_{(n+1)}' = \phi_f(W_n')$, where c' is the constant function $c'(x) = \bot$. It is easy to see that $W_n' = \rho W_n^{(\upsilon f)}$, so the solutions coincide. ☐

Forgetting the provenance of the above-mentioned monads on **Set** via sandwiching, we obtain that the maybe monad $(\text{–}) + \{\bot\}$ on **Set** is an iteration-congruent retract of the delay monad $(\text{–}) \times \mathbb{N} + \{\bot\}$, which is, of course, not surprising. In categories beyond sets (where the delay monad, or partiality monad, is more generally defined as $\nu\gamma.(- + \gamma)$ [7]), the situation is more complex, see Remark 5.16.