

Unifying Guarded and Unguarded Iteration

Sergey Goncharov¹, Lutz Schröder¹, Christoph Rauch¹, and Maciej Piróg²

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg

² Department of Computer Science, KU Leuven, Belgium

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 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 [6]. A more abstract example are completely iterative theories [11] and monads [19], where, in the latter setting, a recursive definition is guarded 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. E.g. to obtain the denotational semantics of an unproductive while loop while true do skip characterized by circular operational behavior

$$\text{while true do skip} \rightarrow \text{skip}; \text{while true do skip} \rightarrow \text{while true do skip}$$

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* [16,26,25]. In the paradigm of monad-based encapsulation of side-effects [21], coinductive resumptions over a base effect encapsulated by a monad T form a *coinductive resumption transform monad* T^ν given by

$$T^\nu X = \nu\gamma.T(X + \gamma) \tag{1}$$

– 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 generally restrict to monads T for which (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). Notice that one has a natural *delay map* $T^\nu X \rightarrow T^\nu X$ that converts a computation into a resumption, i.e. prefixes it with a delay step. In fact, for $T = \text{id}$, T^ν is precisely Capretta’s *partiality monad* [8], also called the *delay monad*. It is not in general possible to equip $T^\nu X$ with a 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* [2,15].

In the current work we aim to unify theories of guarded and unguarded iteration. To this end, we introduce a notion of *abstractly guarded monads*, that is, monads T equipped with a distinguished class of *abstractly guarded* equation morphisms satisfying some 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 monads 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 22) 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). As a slogan,

monad-based models of unguarded recursion arise by quotienting models of guarded recursion.

Our second main result (Theorem 26) is an algebraic characterization of complete Elgot monads: We show that the construction $(-)^{\nu}$ mapping a monad T to T^{ν} as in (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 show (Section 6) that sandwiching a complete Elgot monad between adjoint functors again yields a complete Elgot monad, in analogy to a corresponding result for completely iterative monads [26]. 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.

2 Preliminaries

We work in a category \mathbf{C} with finite coproducts. We fix the notation $\text{in}_i : X_i \rightarrow X_1 + \dots + X_n$ for the i -th injection. A morphism $\sigma : Y \rightarrow X$ is a *summand* of X , which we denote $\sigma : Y \hookrightarrow X$, if there is $\sigma' : X' \rightarrow X$ such that X is a coproduct of Y and X' with σ and σ' being coproduct injections. The morphism σ' is called a (*coproduct*) *complement* of σ and by definition is also a summand. We are not assuming that \mathbf{C} is *extensive*, and coproduct complements are not in general uniquely determined.

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

$$\eta^* = \text{id}, \quad f^* \eta = f, \quad (f^* g)^* = f^* g^*.$$

The standard (equivalent) categorical definition [18] of \mathbb{T} as an endofunctor with natural transformation *unit* $\eta : \text{Id} \rightarrow T$ and *multiplication* $\mu : TT \rightarrow T$ can be recovered by taking $Tf = (\eta f)^*$, $\mu = \text{id}^*$. (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 $X \rightarrow TY$ *Kleisli morphisms* and view them as a high level abstraction of sequential programs with \mathbb{T} encapsulating the underlying computational effect as proposed by Moggi [22], with X representing the input type and Y the output type. A more traditional use of monads in semantics is due to Lawvere [17], who identified finitary monads on \mathbf{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*.

A (*n F-*)*coalgebra* is a pair $(X, f : X \rightarrow FX)$ where $X \in |\mathbf{C}|$ and $F : \mathbf{C} \rightarrow \mathbf{C}$ is an endofunctor. Coalgebras form a category, with morphisms $(X, f) \rightarrow (Y, g)$

being \mathbf{C} -morphisms $h : X \rightarrow Y$ such that $(Fh)f = gh$. A final object of this category is called a *final coalgebra*, and we denote it by $(\nu F, \text{out} : \nu F \rightarrow F\nu F)$ if it exists. For brevity, *we will 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. By definition, νF comes with *coiteration* as a definition principle (dual to the iteration principle for algebras): given a coalgebra $(X, f : X \rightarrow FX)$ there is a unique morphism $(\text{coit } f) : X \rightarrow \nu F$ such that

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

This implies that out is an isomorphism (*Lambek's lemma*) and that $\text{coit out} = \text{id}$ (see [30] for more details about coalgebras for coiteration).

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

3 Guarded 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 $X_1 + \dots + X_n$ and thus talk about guardedness of a morphism $f : X \rightarrow T(X_1 + \dots + X_n)$ in any X_i , meaning that every variable from X_i is guarded in f . Since Kleisli morphisms can be thought of as abstract programs we can therefore speak about guardedness of a program in a certain portion of the output type, e.g. $X_i \hookrightarrow X_1 + \dots + X_n$. One way to capture guardedness categorically is to identify the operations of \mathbb{T} that serve as guards by distinguishing a suitable subobject of \mathbb{T} ; e.g. the definition of completely iterative monad [19] 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. This is formalized as follows.

Definition 1 ((Abstractly) guarded monad). We call a monad \mathbb{T} (*abstractly*) *guarded* if it is equipped with a notion of (abstract) guardedness, i.e. with a relation between morphisms $f : X \rightarrow TY$ and summands of Y (by putting the word ‘abstract’ in brackets we mean that we will often omit it later). We call $f : X \rightarrow TY$ (*abstractly*) σ -*guarded* if f and $\sigma : Y' \hookrightarrow Y$ are in this relation, and then write $f : X \rightarrow_{\sigma} TY$. Abstract guardedness is required to be closed under the rules in Figure 1. In rule **(wkn)**, σ and θ are composable summands.

Given guarded monads \mathbb{T}, \mathbb{S} , a monad morphism $\alpha : T \rightarrow S$ is (*abstractly*) *guarded* if whenever $f : X \rightarrow_{\sigma} TY$ then $\alpha f : X \rightarrow_{\sigma} SY$.

Intuitively, **(trv)** says that if a program does not output anything via a summand of the output type then it is guarded in that summand. Rule **(wkn)**

$$\begin{array}{c}
\text{(trv)} \quad \frac{f : X \rightarrow TY}{(T \text{in}_1) f : X \rightarrow_{\text{in}_2} T(Y + Z)} \qquad \text{(wkn)} \quad \frac{f : X \rightarrow_{\sigma} TY}{f : X \rightarrow_{\sigma\theta} TY} \\
\text{(cmp)} \quad \frac{f : X \rightarrow_{\text{in}_2} T(Y + Z) \quad g : Y \rightarrow_{\sigma} TV \quad h : Z \rightarrow TV}{[g, h]^* f : X \rightarrow_{\sigma} TV} \\
\text{(sum)} \quad \frac{f : X \rightarrow_{\sigma} TZ \quad g : Y \rightarrow_{\sigma} TZ}{[f, g] : X + Y \rightarrow_{\sigma} TZ}
\end{array}$$

Fig. 1: Axioms of guardedness.

is a weakening principle: if a program is guarded in some summand then it is guarded in any subsummand of that summand. Rule **(cmp)** asserts that guardedness is preserved by 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. Finally, rule **(sum)** says that putting two guarded equation systems side by side again produces a guarded system. The rules 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.

We write $f : X \rightarrow_{i_1, \dots, i_k} T(X_1 + \dots + X_n)$ as a shorthand for $f : X \rightarrow_{\sigma} T(X_1 + \dots + X_n)$ with $\sigma = [\text{in}_{i_1}, \dots, \text{in}_{i_k}] : X_{i_1} + \dots + X_{i_k} \hookrightarrow X_1 + \dots + X_n$. 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 $\text{in}_{i_j} [\text{in}_{n_{j,1}}, \dots, \text{in}_{n_{j,k_j}}]$, so, e.g. we write $f : X \rightarrow_{12,2} T((Y + Z) + Z)$ to mean that f is $[\text{in}_1 \text{in}_2, \text{in}_2]$ -guarded. Where coproducts $Y + Z$ etc. appear in the rules, we mean any coproduct, not just some selected coproduct. We defined the notion of guardedness as a certain relation over Kleisli morphisms and summands. Clearly, the largest such relation is the one declaring all Kleisli morphisms to be σ -guarded for all σ . We call monads equipped with this notion of guardedness *totally guarded*. It turns out that for every monad we also have a least guardedness relation.

Definition 2. Let \mathbb{T} be a monad. A morphism $f : X \rightarrow TY$ is *trivially σ -guarded* for $\sigma : Z \hookrightarrow Y$ if f factors through $T\sigma'$ for a coproduct complement σ' of σ .

Proposition 3. *Let \mathbb{T} be a monad. Then taking the abstractly guarded morphisms to be the trivially guarded morphisms is the least guardedness relation making \mathbb{T} into a guarded monad.*

We call a guarded monad *trivially guarded* if all its abstractly guarded morphisms are trivially guarded. As we see, the notion of abstract guardedness can vary on a large spectrum from *trivial guardedness* to *total guardedness*, thus somewhat detaching abstract guardedness from the original intuition. 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 4. Although by **(wkn)**, $f : X \rightarrow_{1,2} T(Y + Z)$ implies both $f : X \rightarrow_1 T(Y + Z)$ and $f : X \rightarrow_2 T(Y + Z)$, the converse is not required to be true, and in fact can fail even for trivial guardedness. This is witnessed by the following simple counterexample. Let \mathbb{T} be the algebraic monad given by taking TX to be the free commutative semigroup over X satisfying the additional law $x + y = x$. Now the term $x + y \in T(X + Y)$ (seen as a morphism $1 \rightarrow T(X + Y)$) with $x \in X$ and $y \in Y$ is both in_1 -guarded and in_2 -guarded, being equivalent both to y and to x . But it is not id -guarded, because it does not factor through $T\emptyset = \emptyset$.

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

Definition 5 (Guarded (pre-)iterative monad). Given $f : X \rightarrow_2 T(Y + X)$, we say that $f^\dagger : X \rightarrow TY$ is a *solution* of f if f^\dagger satisfies the *fixpoint identity* $f^\dagger = [\eta, f^\dagger]^* f$. A guarded monad \mathbb{T} is *guarded pre-iterative* if it is equipped with an *iteration operator* that assigns to every in_2 -guarded morphism $f : X \rightarrow_2 T(Y + X)$ a solution f^\dagger of f . If every such f has a unique solution, we call \mathbb{T} *guarded iterative*.

We can readily check that the iteration operator preserves guardedness.

Proposition 6. *Let \mathbb{T} be a guarded pre-iterative monad, let $\sigma : Y' \hookrightarrow Y$, and let $f : X \rightarrow_{\sigma+\text{id}} T(Y + X)$. Then $f^\dagger : X \rightarrow_\sigma TY$.*

In trivially guarded monads, there is effectively nothing to iterate, so we have

Proposition 7. *Every trivially guarded monad is guarded iterative.*

Guardedness in completely iterative monads One instance of our notion of abstract guardedness is found in completely iterative monads [19], which are based on idealised monads. To make this precise, we need to recall some definitions. First, a *module* over a monad \mathbb{T} on \mathbf{C} is a pair $(M, -^\circ)$, where M is an endomap over the objects of \mathbf{C} , while the lifting $-^\circ$ is a map $\text{Hom}(X, TY) \rightarrow \text{Hom}(MX, MY)$ such that the following laws are satisfied:

$$\eta^\circ = \text{id}, \quad g^\circ f^\circ = (g^* f)^\circ.$$

Note that M extends to an endofunctor by taking $Mf = (\eta f)^\circ$. Next, a *module-to-monad morphism* is a natural transformation $\xi : M \rightarrow T$ that satisfies $\xi f^\circ = f^* \xi$. We call the triple $(\mathbb{T}, M, -^\circ, \xi)$ an *idealised monad*; when no confusion is likely, we refer to these data just as \mathbb{T} . Following [19], we can then define guardedness as follows:

Definition 8. Given an idealised monad \mathbb{T} as above, a morphism $f : X \rightarrow T(Y + X)$ is *guarded* if it factors via $[\eta \text{in}_1, \xi] : Y + M(Y + X) \rightarrow T(Y + X)$. Then, \mathbb{T} is a *completely iterative monad* if every such guarded f has a unique solution.

It turns out that this notion of guardedness is not an instance of our notion of abstract guardedness. Fortunately, we can fix this by noticing that completely iterative monads actually support iteration for a wider class of morphisms:

Definition 9. Let $(\mathbb{T}, M, -^\circ, \xi)$ be an idealised monad. Given $\sigma : Z \hookrightarrow Y$, we say that a morphism $f : X \rightarrow TY$ is *weakly σ -guarded* if it factors through $[\eta\sigma', \xi]^* : T(Y' + MY) \rightarrow TY$ for a complement $\sigma' : Y' \hookrightarrow Y$ of σ .

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

Theorem 10. Let $(\mathbb{T}, M, -^\circ, \xi)$ be an idealised 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 $f : X \rightarrow T(Y + X)$ has a unique solution.

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

4 Parametrizing Guardedness

Uustalu [29] 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 11 (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 $\# : \mathbf{C} \times \mathbf{D} \rightarrow \mathbf{C}$ such that for any $X \in |\mathbf{D}|$, $- \# X : \mathbf{C} \rightarrow \mathbf{C}$ is a (guarded) monad, and for every $f : X \rightarrow Y$, $\text{id} \# f : Z \# X \rightarrow Z \# Y$ is the Z -component of a (guarded) monad morphism $- \# f : - \# X \rightarrow - \# Y$.

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

Guardedness of the monad morphisms $- \# f$ means explicitly that $g : Z \rightarrow_\sigma V \# X$ implies $(\text{id} \# f)g : Z \rightarrow_\sigma V \# Y$.

Example 12. For the purposes of the present work, the most important example (taken from [29]) is $\# = T(- + \Sigma -) : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ 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 powerset, and Σ captures a signature of actions, e.g. $\Sigma X = A \times X$, as in process algebra. 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.

Incidentally, our modification of parametrized monads also covers Atkey's parametrized monads [5], which are certain functors $\mathbf{S} \times \mathbf{S}^{op} \times \mathbf{C} \rightarrow \mathbf{C}$ forming a monad in the third argument. The first and the second arguments serve, e.g., to parametrize the computational effect of interest with initial and final states of different types.

Theorem 13. *Let $\# : \mathbf{C} \times (\mathbf{C} \times \mathbf{D}) \rightarrow \mathbf{C}$ be a parametrized monad, with unit η and Kleisli lifting $(-)^*$. Then*

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

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

1. *If $\#$ is guarded then so is $\#^\nu$ with guardedness defined as follows: given $\sigma : Y' \hookrightarrow Y$, $f : X \rightarrow Y \#^\nu Z$ is σ -guarded if $\text{out } f : X \rightarrow Y \# (Y \#^\nu Z, Z)$ is σ -guarded.*
2. *If $\#$ is pre-iterative under an iteration operator $-^\dagger$ then so is $\#^\nu$ with the iteration operator $-^\ddagger$ defined as follows:*

$$(f : X \rightarrow_2 (Y + X) \#^\nu Z)^\ddagger = \text{coit}([\eta, (\text{out } f)^\dagger]^* \text{out}) \eta^\nu \text{in}_2$$

3. *If $\#$ is iterative then so is $\#^\nu$ under the definition from the previous clause.*

Example 14. We spell out one instance of Theorem 13 in case $\mathbf{D} = 1$ and $\# = T(- + -)$ where $\mathbb{T} = (T, \eta, -^*)$ is a monad. Then $\#^\nu$ is isomorphically a monad \mathbb{T}^ν on \mathbf{C} with $T^\nu X = \nu\gamma. T(X + \gamma)$, unit $\eta^\nu = \eta \text{in}_1$ and Kleisli star $(f : X \rightarrow T^\nu Y)^*$ being uniquely determined by the equation

$$\text{out } f^* = [\text{out } f, \eta \text{in}_2 f^*]^* \text{out}.$$

If \mathbb{T} is pre-iterative then so is \mathbb{T}^ν with iteration

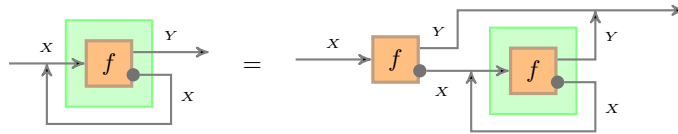
$$(f : X \rightarrow T^\nu(Y + X))^\ddagger = \text{coit}([\eta \text{in}_2, (T[\text{in}_1 + \text{id}, \text{in}_1 \text{in}_2] \text{out } f)^\dagger]^* \text{out}) \eta^\nu \text{in}_2.$$

Example 15. Theorem 13 shows that our notion of guardedness extends along the applications of the final coalgebra transformer $\# \mapsto \#^\nu$ on parametrized monads. This can be used to capture existing notions of guardedness as follows. Consider $X \# Y = T(X + (1 + A) \times Y)$ where \mathbb{T} is some monad. In **Set**, $1 + A$ can be thought of as consisting of a set A of visible actions and a silent action τ . In process algebra we standardly consider a process definition to be guarded if every recursive call is preceded by a visible action from A . In our framework this can be reconstructed as follows. The obvious isomorphism

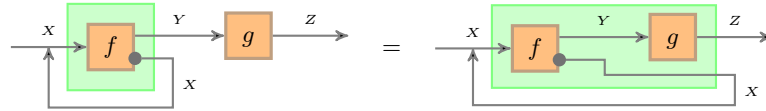
$$\nu\gamma. T(X + (1 + A) \times \gamma) \cong \nu\gamma'. \nu\gamma. T(X + \gamma + A \times \gamma')$$

involves two more parametrized monads: $T(- + - + A \times -) : \mathbf{C} \times \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ and $\nu\gamma. T(- + \gamma + A \times -) : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$. By taking the latter to be trivially guarded and then defining guardedness for $\nu\gamma'. \nu\gamma. T(X + \gamma + A \times \gamma')$ using Theorem 13, we arrive precisely at the notion we expected for the isomorphic monad $\#^\nu$.

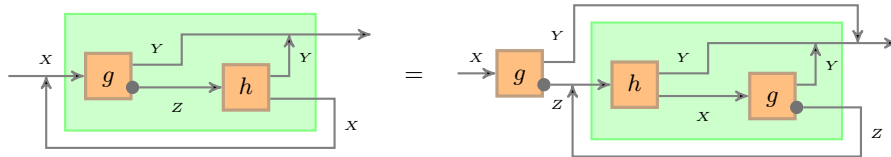
Fixpoint:



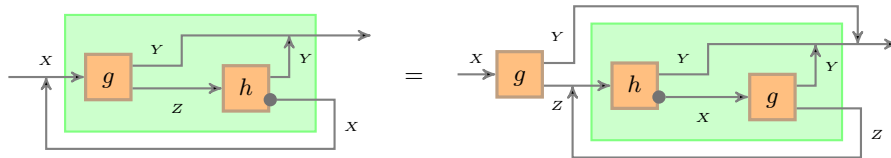
Naturality:



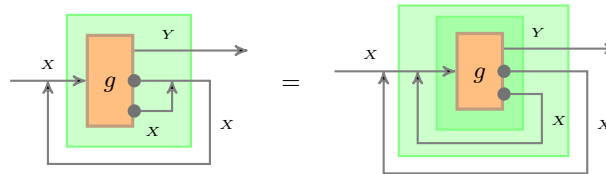
Dinaturality 1:



Dinaturality 2:



Codiagonal:



Uniformity:

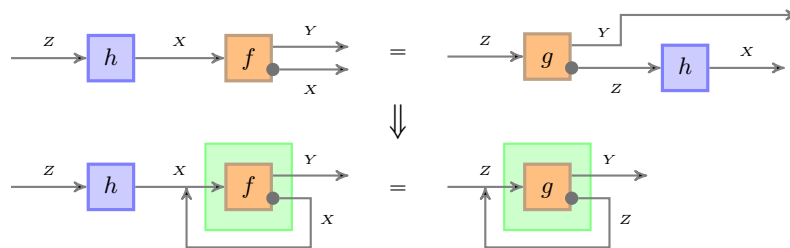


Fig. 2: Axioms of guarded iteration.

5 Complete Elgot Monads and Iteration Congruences

Besides the fixpoint identity we are interested in the following classical properties of the iteration operator, which we refer to as the *iteration laws* [10,7,27]:

- *naturality*: $g^* f^\dagger = ([T \text{in}_1] g, \eta \text{in}_2]^* f)^\dagger$ for $f : X \rightarrow_2 T(Y + X)$, $g : Y \rightarrow TZ$;
- *dinaturality*: $([\eta \text{in}_1, h]^* g)^\dagger = [\eta, ([\eta \text{in}_1, g]^* h)^\dagger]^* g$ for $g : X \rightarrow_2 T(Y + Z)$ and $h : Z \rightarrow T(Y + X)$ or $g : X \rightarrow T(Y + Z)$ and $h : Z \rightarrow_2 T(Y + X)$;
- *codiagonal*: $(T[\text{id}, \text{in}_2] f)^\dagger = f^{\dagger\dagger}$ for $f : X \rightarrow_{12,2} T((Y + X) + X)$;
- *uniformity*: $f h = T(\text{id} + h) g$ implies $f^\dagger h = g^\dagger$ for $f : X \rightarrow_2 T(Y + X)$, $g : Z \rightarrow_2 T(Y + Z)$ and $h : Z \rightarrow X$.

The axioms are summarized in graphical form in Figure 2, and then become quite intuitive. The two versions of the dinaturality axiom correspond to the alternative sets of guardedness assumptions mentioned above. We indicate the scope of the iteration operator by a shaded box and guardedness by bullets at the outputs of a morphism.

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 for guardedness. E.g., in the dinaturality axiom it suffices to assume that $g : X \rightarrow T(Y + Z)$ is in_2 -guarded and this implies that both $[\eta \text{in}_1, h]^* g$ and $[\eta \text{in}_1, g]^* h$ are in_2 -guarded by **(cmp)** and **(trv)**, and additionally **(sum)** in the latter case. Symmetrically, it suffices to make the analogous assumption about h . In the codiagonal axiom, it follows from the assumption $f : X \rightarrow_{12,2} T((Y + X) + X)$ by **(cmp)** that $T[\text{id}, \text{in}_2] f$ is in_2 -guarded and by Proposition 6 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 that for guarded iterative monads all iteration laws are automatic. Prior to that, we show that dinaturality follows from the others (thus generalizing the corresponding observation made recently [15,13]).

Proposition 16. *Any guarded pre-iterative monad satisfying naturality, codiagonal and uniformity also satisfies dinaturality, as well as the Bekić identity*

$$[T[\text{id} + \text{in}_1, \text{in}_2]^* f, T[\text{id} + \text{in}_1, \text{in}_2]^* g]^\dagger = [h^\dagger, [\eta, h^\dagger]^* g^\dagger]$$

where $f : X \rightarrow_{12,2} T((Y + X) + Z)$, $g : Z \rightarrow_{12,2} T((Y + X) + Z)$, and $h = [\eta, g^\dagger]^* f : X \rightarrow_2 T(Y + X)$.

The proof of the following result runs in accordance with the original ideas of Elgot [10] for iterative theories, except that, by Proposition 16, dinaturality is now replaced by uniformity.

Theorem 17. *Every guarded iterative monad validates naturality, dinaturality, codiagonal and uniformity.*

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

Definition 18 (Guarded retraction). Let \mathbb{T} and \mathbb{S} be guarded monads. We call a monad morphism $\rho : \mathbb{T} \rightarrow \mathbb{S}$ a *guarded retraction* if there is a family of morphisms $(v_X : SX \rightarrow TX)_{X \in |\mathbf{C}|}$ (not necessarily natural in X !) such that

1. for every $f : X \rightarrow_{\sigma} SY$, we have $v_Y f : X \rightarrow_{\sigma} TY$,
2. $\rho_X v_X = \text{id}$ for all $X \in |\mathbf{C}|$.

Theorem 19. *Let $\rho : \mathbb{T} \rightarrow \mathbb{S}$ be a guarded retraction, witnessed by $v : \mathbb{S} \rightarrow \mathbb{T}$, and suppose that $(\mathbb{T}, -^{\dagger})$ is guarded pre-iterative. Then \mathbb{S} is guarded pre-iterative with the iteration operator $(-)^{\ddagger}$ given by $f^{\ddagger} = \rho(vf)^{\dagger}$.*

Definition 20 (Iteration congruence). Let \mathbb{T} be a guarded pre-iterative monad and let \mathbb{S} be a monad. We call a monad morphism $\rho : \mathbb{T} \rightarrow \mathbb{S}$ an *iteration congruence* if for every pair of morphisms $f, g : X \rightarrow_2 T(Y + X)$,

$$\rho f = \rho g \implies \rho f^{\dagger} = \rho g^{\dagger}. \quad (2)$$

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

Theorem 21. *Under the premises of Theorem 19, 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 (Sketch). The crucial observation is that under our assumptions, (2) is equivalent to the condition that for all $f : X \rightarrow_2 T(Y + X)$,

$$\rho(v\rho f)^{\dagger} = \rho f^{\dagger}. \quad (3)$$

Indeed, (2) \implies (3), for $\rho v\rho f = \rho f$ and therefore $\rho(v\rho f)^{\dagger} = \rho f^{\dagger}$ and conversely, assuming (3) both for f and for g , and $\rho f = \rho g$, we obtain that $\rho f^{\dagger} = \rho(v\rho f)^{\dagger} = \rho(v\rho g)^{\dagger} = \rho g^{\dagger}$. Using (3), the claim is established routinely. \square

Recall from the introduction that a monad \mathbb{S} is *iteratable* if its coinductive resumption transform \mathbb{S}^{ν} exists. We make \mathbb{S}^{ν} into a guarded monad by applying Theorem 13 to \mathbb{S} as a trivially guarded monad; explicitly: $f : X \rightarrow S^{\nu}(Y + X)$ is guarded iff $\text{out } f = S(\text{in}_1 + \text{id})g$ for some $g : X \rightarrow S(Y + S^{\nu}(Y + X))$. 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 22. *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 $\rho : \mathbb{T} \rightarrow \mathbb{S}$. Specifically, every iteratable Elgot monad \mathbb{S} is an iteration-congruent retract of its coinductive resumption transform \mathbb{S}^{ν} .*

Proof (Sketch). Direction (\Leftarrow) immediately follows from Theorems 17 and 21 .

In order to prove (\Rightarrow), we show that $\mathbb{S} = (S, \eta, -^*, -^\dagger)$ is an iteration-congruent retract of $\mathbb{S}^\nu = (\nu\gamma. S(- + \gamma), \eta^\nu, -^*, -^\dagger)$. Let $v_X = \text{out}^{-1} \eta \text{in}_2 \text{out}^{-1}(S \text{in}_1)$ and

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

Clearly, $v f$ is σ -guarded for every $f : X \rightarrow SY$ and it is easy to verify that v is left inverse to ρ by using the fixpoint identity for $-^\dagger$ twice.

Naturality of ρ is proved straightforwardly from naturality of $-^\dagger$. The remaining calculations showing that ρ is a monad morphism and moreover an iteration congruence make heavy use of the Elgot monad laws. \square

The notions of guarded retraction and iteration congruence straightforwardly extend to parametrized monads. We then can take the claims of Theorem 13 further.

Theorem 23. *Let $\#, \hat{\#} : \mathbf{C} \times (\mathbf{C} \times \mathbf{D}) \rightarrow \mathbf{C}$ be guarded parametrized monads and let $\rho : \# \rightarrow \hat{\#}$ be an iteration-congruent retraction. By Theorem 13, $\#^\nu = -\#(\gamma, -)$ and $\hat{\#}^\nu = -\hat{\#}(\gamma, -)$ are also parametrized guarded monads. Then $\rho^\nu : \#^\nu \rightarrow \hat{\#}^\nu$, with components*

$$\rho_{X,Y}^\nu = \text{coit}(\nu\gamma. X \# (\gamma, Y) \xrightarrow{\rho \text{out}} X \hat{\#} (\nu\gamma. X \# (\gamma, Y), Y)),$$

is again an iteration-congruent retraction.

Theorems 22 and 23 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 22 there is an iteration-congruent retraction $\rho : \nu\gamma. - + \gamma \# X \rightarrow -\#X$; by Theorem 23, it gives rise to an iteration-congruent retraction

$$\rho^\nu : \nu\gamma'. \nu\gamma. - + \gamma \# (\gamma', X) \rightarrow \nu\gamma'. -\#(\gamma', X)$$

and by Theorem 22, the right-hand side is again Elgot. We have thus proved

Corollary 24. *Given a parametrized monad $\#$ and $X \in |\mathbf{C}|$, if $-\#X$ is Elgot then so is $-\#^\nu X = \nu\gamma. -\#(\gamma, X)$.*

This yields a more general and simpler proof of one of the main results in [15].

Example 25. By instantiating $\#$ in Corollary 24 with $\mathcal{P}_\omega(- + A \times -) : \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ where \mathcal{P}_ω is the countable powerset monad, we obtain $\nu\gamma. \mathcal{P}_\omega(X + A \times \gamma)$, which can be viewed as a semantic domain for countably branching processes that possibly terminate with results in X and are taken modulo strong bisimilarity. The simple fact that \mathcal{P}_ω is Elgot [15] implies that so is $\nu\gamma. \mathcal{P}_\omega(X + A \times \gamma)$. This justifies the use of systems of possibly unguarded recursive process algebra equations (as done, e.g., in [6]). It is worth noting that the iteration operator of the transformed monad is neither least nor unique. It arises by introducing an additional delay action that guards all recursive calls and then eliminating these delays from the unique solution of the new recursive definition; the delay elimination is the effect of ρ^ν in Theorem 23.

Theorem 22 characterizes iterable Elgot monads as iteration-congruent retracts of their $(-)^{\nu}$ -transforms. We take this perspective further as follows. Let us call \mathbb{T} *strongly iterable* if every $T^{\nu \dots \nu}$ exists. Consider the functor $\mathbb{T} \mapsto \mathbb{T}^{\nu}$ on the category of strongly iterable monads over \mathbf{C} . This is itself a monad: the unit η is the natural transformation with components $\eta_X = \text{out}^{-1}(T \text{in}_1) : TX \rightarrow T^{\nu} X$ and the multiplication $\mu : T^{\nu \nu} \rightarrow T^{\nu}$ has components

$$\mu_X = \text{coit}(T[\text{id}, \text{in}_2 \text{out}^{-1}] \text{out out} : T^{\nu \nu} X \rightarrow T(X + T^{\nu \nu} X)).$$

For every T we define the *delay transformation* $\triangleright = \text{out}^{-1} \eta \text{in}_2 : T^{\nu} \rightarrow T^{\nu}$. This leads to our second main result:

Theorem 26. *The category of strongly iterable Elgot monads over \mathbf{C} is isomorphic to the full subcategory of the category of $(-)^{\nu}$ -algebras for strongly iterable \mathbb{S} consisting of the $(-)^{\nu}$ -algebras $(S^{\nu}, \rho : S^{\nu} \rightarrow S)$ satisfying $\rho \triangleright = \rho$.*

Proof (Sketch). To show that every strongly iterable Elgot monad is a $(-)^{\nu}$ -algebra, one has to check the equations $\rho \eta = \text{id}$ and $\rho \mu = \rho \rho^{\nu}$ where $\rho^{\nu} = \text{coit}(\rho \text{out}) : S^{\nu \nu} \rightarrow S^{\nu}$. The first equation follows relatively easily. The second one is shown along the following lines:

$$\rho \mu \stackrel{(i)}{=} \rho[\eta, (\triangleright \text{out})^{\ddagger}]^* \text{out} \stackrel{(ii)}{=} \rho(\text{out}^{-1} S(\text{in}_1 + \eta^{\nu} \text{in}_2) \rho \text{out})^{\ddagger} \stackrel{(iii)}{=} \rho \rho^{\nu}.$$

Here, (i) and (iii) only amount to equivalent transformations of μ and ρ^{ν} , respectively, while (ii) makes crucial use of the fact that ρ is an iteration congruence, as implied by Theorem 22.

For the converse implication, we start with a $(-)^{\nu}$ -algebra and verify the Elgot monad laws for the iteration operator $f^{\ddagger} = \rho(\text{coit } f)$. \square

Remark 27. The delay cancellation condition $\rho \triangleright = \rho$ is essential, as can be seen on a simple example. Let $\mathbf{Mon}(\mathbf{C})^{\nu}$ be the category of $(-)^{\nu}$ -algebras and let $\mathbf{Mon}(\mathbf{C})_{\triangleright}^{\nu}$ be its subcategory figuring in Theorem 26. Since the identity functor is the initial monad, the initial object of $\mathbf{Mon}(\mathbf{C})^{\nu}$ is Capretta's delay monad [8] $D = \nu \gamma. (- + \gamma)$. On the other hand, the initial object of $\mathbf{Mon}(\mathbf{C})_{\triangleright}^{\nu}$ (if it exists) is the *initial Elgot monad* \mathbb{L} , which on $\mathbf{C} = \mathbf{Set}$ is the *maybe monad* $(-) + 1$.

If $\mathbf{C} = \mathbf{Set}$, then $DX = (X \times N + 1)$ does turn out to be Elgot [14] (but applying Theorem 26 to D qua Elgot monad yields a different $(-)^{\nu}$ -algebra structure than the initial one), and \mathbb{L} is, in this case, a retract of \mathbb{D} in $\mathbf{Mon}(\mathbf{C})_{\triangleright}^{\nu}$. The situation is more intricate in categories with a non-classical 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 $(-)^{\nu}$ -algebra morphism $\mathbb{D} \rightarrow \mathbb{L}$ in $\mathbf{Mon}(\mathbf{C})^{\nu}$.

6 A Sandwich Theorem for Elgot Monads

As an application of Theorem 22, we show that sandwiching an Elgot monad between adjoint functors again yields an Elgot monad. A similar result has been shown for completely iterative monads [26]; this result generalizes straightforwardly to guarded iterative monads:

Theorem 28. *Let $F : \mathbf{C} \rightarrow \mathbf{D}$ and $U : \mathbf{D} \rightarrow \mathbf{C}$ be a pair of adjoint functors with the associated natural isomorphism $\Phi : \mathbf{D}(FX, Y) \rightarrow \mathbf{C}(X, UY)$, 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 $f : X \rightarrow_{\sigma} UTFY$ if and only if $\Phi^{-1}f : FX \rightarrow_{\sigma} TFY$, and the unique solutions given by $f \mapsto \Phi((\Phi^{-1}f)^{\dagger})$.*

Now, to obtain a similar result for Elgot monads, we can easily combine Theorems 22 and 28 without having to verify the equational properties by hand.

Theorem 29. *With an adjunction as in Theorem 28, let \mathbb{S} be an Elgot monad on \mathbf{D} . Then, the monad induced on the composition USF is an Elgot monad.*

Proof (Sketch). By Theorem 22, there exists a guarded iterative monad \mathbb{T} and an iteration-congruent retraction $\rho : \mathbb{T} \rightarrow \mathbb{S}$. By Theorem 28, the monad induced on UTF is guarded iterative. Again by Theorem 22, it suffices to show that $U\rho F : UTF \rightarrow USF$ is an iteration-congruent retraction, which is straightforward. \square

Example 30 (From Metric to CPO-based Iteration). As an example exhibiting sandwiching as well as the setting of Theorem 22, 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 $(X \times \mathbb{N}) \cup \{\perp\}$. As we show in the appendix, $\bar{\mathbb{S}}$ is guarded iterative if we define $f : (X, d) \rightarrow S(Y, d')$ to be σ -guarded if $k > 0$ whenever $f(x) = (\sigma(y), k)$. By Theorem 28, $\bar{\mathbb{S}}$ 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 29. 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 $\rho : \bar{\mathbb{S}} \rightarrow \mathbb{L}$, 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 22.

7 Related Work

Alternatively to our guardedness relation on Kleisli morphisms, guardedness can be formalized using type constructors [23] or, categorically, functors, as in *guarded fixpoint categories* [20]; the latter cover 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 [20, Proposition 3.15] akin to the one we discuss in Remark 4.

A result that resembles our Theorem 26, due to Adámek et. al. [3], states roughly that if \mathbf{C} is locally finitely presentable and hyperextensive (e.g. $\mathbf{C} = \mathbf{Set}$) then the finitary Elgot monads are the algebras for a monad on the category of endofunctors given by $H \mapsto L_H = \rho\gamma.(- + 1 + H\gamma)$ where ρ takes *rational fixpoints* (i.e. final coalgebras among those where every point generates a finite subcoalgebra). Besides Theorem 26 making fewer assumptions on \mathbf{C} , the key difference is that, precisely by dint of this result, L_H is already an Elgot monad; 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 the quasi-equational theory of iteration [2,15] (definition);
- it is an iteration-congruent retract of a guarded iterative monad (Theorem 22);
- it is an algebra (\mathbb{T}, ρ) of the monad $T \mapsto \nu\gamma.T(X + \gamma)$ in the category of monads satisfying a natural delay cancellation condition (Theorem 26).

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 the work of Nakata and Uustalu [24], 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.

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 26 may shed light on this issue. A natural question that arises in this regard is whether the subcategory of $(-)^{\nu}$ -algebras figuring in the theorem is reflexive. A positive answer would provide a means of constructing canonical quotients of $(-)^{\nu}$ -algebras (such as the delay monad) with the results automatically being Elgot monads.

References

1. P. Aczel, J. Adámek, S. Milius, and J. Velebil. Infinite trees and completely iterative theories: a coalgebraic view. *Theoretical Computer Science*, 300(1–3):1–45, 2003.
2. J. Adámek, S. Milius, and J. Velebil. Equational properties of iterative monads. *Inf. Comput.*, 208:1306 – 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.*, 21:417–480, 2011.
4. T. Altenkirch and N. Danielsson. Partiality, revisited. In *Types for Proofs and Programs, TYPES 2016*, 2016.
5. R. Atkey. Parameterised notions of computation. *J. Funct. Prog.*, 19:335, 2009.
6. J. Bergstra, A. Ponse, and S. Smolka, eds. *Handbook of Process Algebra*. Elsevier, 2001.
7. S. Bloom and Z. Ésik. *Iteration theories: the equational logic of iterative processes*. Springer, 1993.
8. V. Capretta. General recursion via coinductive types. *Log. Meth. Comput. Sci.*, 1(2), 2005.
9. J. Chapman, T. Uustalu, and N. Veltri. Quotienting the delay monad by weak bisimilarity. In *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 *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 structure of rooted trees. *J. Comput. Syst. Sci.*, 16:362–399, 1978.
12. M. H. Escardó. A metric model of PCF. In *Realizability Semantics and Applications*, 1999.
13. Z. Ésik and S. Goncharov. Some remarks on Conway and iteration theories. *CoRR*, abs/1603.00838, 2016.
14. S. Goncharov, S. Milius, and C. Rauch. Complete Elgot monads and coalgebraic resumptions. In *Mathematical Foundations of Programming Semantics, MFPS 2016*, ENTCS, 2016.
15. S. Goncharov, C. Rauch, and L. Schröder. Unguarded recursion on coinductive resumptions. In *Mathematical Foundations of Programming Semantics, MFPS 2015*, ENTCS, 2015.
16. S. Goncharov and L. Schröder. A coinductive calculus for asynchronous side-effecting processes. *Inf. Comput.*, 231:204 – 232, 2013.
17. W. Lawvere. Functorial semantics of algebraic theories. *Proc. Natl. Acad. Sci. USA*, 50:869–872, 1963.
18. S. Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.
19. S. Milius. Completely iterative algebras and completely iterative monads. *Inf. Comput.*, 196:1–41, 2005.
20. S. Milius and T. Litak. Guard your daggers and traces: Properties of guarded (co-)recursion. *Fund. Inform.* To appear.
21. E. Moggi. A modular approach to denotational semantics. In *Category Theory and Computer Science, CTCS 1991*, vol. 530 of *LNCS*, pp. 138–139. Springer, 1991.
22. E. Moggi. Notions of computation and monads. *Inf. Comput.*, 93:55–92, 1991.
23. H. Nakano. A modality for recursion. In *Logic in Computer Science, LICS 2000*, pp. 255–266. IEEE Computer Society, 2000.

24. K. Nakata and T. Uustalu. A Hoare logic for the coinductive trace-based big-step semantics of while. *Log. Meth. Comput. Sci.*, 11(1), 2015.
25. M. Piróg and J. Gibbons. The coinductive resumption monad. In *Mathematical Foundations of Programming Semantics, MFPS 2014*, vol. 308 of *ENTCS*, pp. 273–288, 2014.
26. M. Piróg and J. Gibbons. Monads for behaviour. In *Mathematical Foundations of Programming Semantics, MFPS 2013*, vol. 298 of *ENTCS*, pp. 309–324, 2015.
27. A. Simpson and G. Plotkin. Complete axioms for categorical fixed-point operators. In *Logic in Computer Science, LICS 2000*, pp. 30–41, 2000.
28. M. Smyth. Topology. In *Handbook of Logic in Computer Science*, vol. 1, pp. 641–761. Clarendon Press, 1992.
29. T. Uustalu. Generalizing substitution. *ITA*, 37:315–336, 2003.
30. T. Uustalu and V. Vene. Primitive (Co)Recursion and Course-of-Value (Co)Iteration, Categorically. *Informatika (Lithuanian Academy of Sciences)*, 10(1):5–26, 1999.

A Appendix: Omitted proofs

A.1 Proof of Proposition 3

It is immediate from **(trv)** that every trivially guarded morphism is guarded under any guardedness relation making \mathbb{T} into a guarded monad. It remains to show that trivial guardedness is closed under the rules in Definition 1; in the following we write $f : X \rightarrow_{\sigma} TY$ to mean that f is trivially σ -guarded.

- **(trv)**: Immediate from the definition of trivial guardedness.
- **(wkn)**: Suppose $f : X \rightarrow_{\sigma} TY$, i.e. $f = (T\sigma')g$ for some $g : X \rightarrow TY'$ where $\sigma : Y' \hookrightarrow Y$ and σ' is a coproduct complement of $\sigma : Z \hookrightarrow Y$. Moreover, suppose $\theta : W \hookrightarrow Z$, and let $\theta' : Z' \hookrightarrow Z$ be a coproduct complement of θ . Then $[\sigma\theta', \sigma'] : Z' + Y' \rightarrow Y$ is a coproduct complement of $\sigma\theta$, and f factors as $X \xrightarrow{g} TY' \xrightarrow{T\text{in}_2} T(Z' + Y') \xrightarrow{[\sigma\theta', \sigma']} TY$, so f is trivially $\sigma\theta$ -guarded.
- **(cmp)**: Suppose $f : X \rightarrow_2 T(Y+Z)$, i.e. $f = (T\text{in}_1)w$ for some $w : X \rightarrow TY$. Now, for $g : Y \rightarrow_{\sigma} TV$ and $h : Z \rightarrow TV$, $[g, h]^*f = g^*w$. Let $\sigma' : W \hookrightarrow V$ be a complement of $\sigma : V' \hookrightarrow V$. By assumption, g factors through $T\sigma'$, i.e. $w = (T\sigma')t$ for some t . Therefore $[g, h]^*f = (T\sigma')t^*w$ which by definition means that $[g, h]^*f$ is trivially σ -guarded.
- **(sum)**: Suppose that $f : X \rightarrow_{\sigma} TZ$ and $g : X \rightarrow_{\sigma} TZ$, i.e. $f = (T\sigma')f'$ and $g = (T\sigma')g'$ for some $f' : X \rightarrow TZ'$ and $g' : X \rightarrow TZ'$ where $\sigma : Z' \hookrightarrow Z$ and σ' is a coproduct complement of σ . Then, of course, $[f, g] = (T\sigma')[f', g']$, i.e. $[f, g] : X + Y \hookrightarrow TZ$. \square

A.2 Proof of Proposition 6

Let $\sigma' : Z \rightarrow Y$ be the complement of $\sigma : Y' \rightarrow Y$ and let $h : Y \rightarrow Z + Y'$ be the inverse of $[\sigma', \sigma]$. Then

$$\begin{aligned}
 f^{\dagger} &= [\eta, f^{\dagger}]^* f \\
 &= [\eta, f^{\dagger}]^* T([\sigma', \sigma] + \text{id}) T(h + \text{id}) f \\
 &= [\eta[\sigma', \sigma], f^{\dagger}]^* T(h + \text{id}) f \\
 &= [\eta\sigma', [\sigma, f^{\dagger}]]^* [[\eta\text{in}_1, \eta\text{in}_2\text{in}_1] h, \eta\text{in}_2\text{in}_2]^* f.
 \end{aligned}$$

Note that by **(wkn)**, $[[\eta\text{in}_1, \eta\text{in}_2\text{in}_1] h, \eta\text{in}_2\text{in}_2]^* f : X \rightarrow_2 T(Z + (Y' + X))$. Since by **(trv)**, $\eta\sigma' = (T\sigma')\eta : Y' \rightarrow_{\sigma} TY$ is σ -guarded, we are done by **(cmp)**. \square

A.3 Proof of Theorem 10

1: We need to verify that weak guardedness is closed under the rules from Definition 1.

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

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

- **(wkn)** Let $[\theta, \theta'] : V + V' \rightarrow Z$ be an isomorphism. Assume that a morphism $f : X \rightarrow_{\sigma} TY$ factors as $[\eta\sigma', \xi]^* f'$ for some morphism f' . Then, the following holds:

$$\begin{aligned}
f &= [\eta\sigma', \xi]^* f' \\
&= [\eta[\sigma\theta', \sigma'] \text{ in}_2, \xi]^* f' && // \text{ coproducts} \\
&= [\eta[\sigma\theta', \sigma'], \xi]^*(T(\text{in}_2 + \text{id}))f' && // \text{ Kleisli}
\end{aligned}$$

- **(cmp)** Given $f : X \rightarrow_2 T(Y + Z)$, $g : Y \rightarrow_{\sigma} TV$, and $h : Y \rightarrow TV$, assume that f factors as $[\eta \text{ in}_1, \xi]^* f'$, while g factors as $[\eta\sigma', \xi]^* g'$. Then, the following holds:

$$\begin{aligned}
[g, h]^* f &= [[\eta\sigma', \xi]^* g', h]^* [\eta \text{ in}_1, \xi]^* f' \\
&= ([[\eta\sigma', \xi]^* g', h]^* [\eta \text{ in}_1, \xi])^* f' && // \text{ Klesli} \\
&= [[[\eta\sigma', \xi]^* g', h]^* \eta \text{ in}_1, [[\eta\sigma', \xi]^* g', h]^* \xi]^* f' && // \text{ coproducts} \\
&= [[[\eta\sigma', \xi]^* g', h] \text{ in}_1, [[\eta\sigma', \xi]^* g', h]^* \xi]^* f' && // \text{ Kleisli} \\
&= [[\eta\sigma', \xi]^* g', [[\eta\sigma', \xi]^* g', h]^* \xi]^* f' && // \text{ coproducts} \\
&= [[\eta\sigma', \xi]^* g', \xi[[\eta\sigma', \xi]^* g', h]^{\circ}]^* f' && // \text{ m-to-m morph.} \\
&= [[\eta\sigma', \xi]^* g', [\eta\sigma', \xi] \text{ in}_2[[\eta\sigma', \xi]^* g', h]^{\circ}]^* f' && // \text{ coproducts} \\
&= [[\eta\sigma', \xi]^* g', [\eta\sigma', \xi]^* \eta \text{ in}_2[[\eta\sigma', \xi]^* g', h]^{\circ}]^* f' && // \text{ Kleisli} \\
&= ([\eta\sigma', \xi]^* [g', \eta \text{ in}_2[[\eta\sigma', \xi]^* g', h]^{\circ}])^* f' && // \text{ coproducts} \\
&= [\eta\sigma', \xi]^* [g', \eta \text{ in}_2[[\eta\sigma', \xi]^* g', h]^{\circ}]^* f' && // \text{ Kleisli}
\end{aligned}$$

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

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

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

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

Thus, g is guarded in the sense of [19], so it has a unique solution $g^\dagger : M(Y+X) \rightarrow TY$. We use it to define a solution to f , namely $f^\ddagger = [\eta, g^\dagger]^*j$. It is left to show that it is indeed a solution and that it is unique:

– *Solution:*

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

– *Uniqueness:* Let $r : X \rightarrow TY$ be a solution of f , that is, it is the case that $r = [\eta, r]^*f$. First, we calculate:

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

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

$$r = [\eta, r]^*f$$

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

□

A.4 Proof of Proposition 7

The fact that f is in_2 -guarded by assumption means that $f = (T \text{in}_1)g$ for some $g : X \rightarrow TY$. Then, of course, for any f^\dagger satisfying $f^\dagger = [\eta, f^\dagger]^* f$, $f^\dagger = [\eta, f^\dagger]^* f = [\eta, f^\dagger]^* (T \text{in}_1)g = g$, i.e. f^\dagger can be taken to be g , and this is the unique choice satisfying the fixpoint identity. □

A.5 Proof of Proposition 16

Following [13], we consider a specific instance of uniformity:

$$f^\dagger = [T(\text{id} + \text{in}_1)f, g]^\dagger \text{in}_1 \quad (4)$$

where $f : X \rightarrow_2 T(Y + X)$ and $g : Z \rightarrow_2 T(Y + (X + Z))$ and prove the following instance of *Bekić identity*:

$$[T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g]^\dagger = [f^\dagger, [\eta, f^\dagger]^* g] \quad (5)$$

where $f : X \rightarrow_2 T(Y + X)$ and $g : Z \rightarrow_2 T(Y + X)$. Indeed, on the one hand, by (4),

$$f^\dagger = [T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g]^\dagger \text{in}_1, \quad (6)$$

and on the other hand

$$\begin{aligned}
&[T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g]^\dagger \text{in}_2 \\
&= [\eta, [T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g]^\dagger]^* \\
&\quad [T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g] \text{in}_2 && // \text{ fixpoint} \\
&= [\eta, [T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g]^\dagger]^* T(\text{id} + \text{in}_1)g \\
&= [\eta, [T(\text{id} + \text{in}_1)f, T(\text{id} + \text{in}_1)g]^\dagger \text{in}_1]^* g \\
&= [\eta, h^\dagger]^* g. && // (6)
\end{aligned}$$

As the result we obtain (5). Analogously, we prove another instance of *Bekić identity*, which is

$$[T(\text{id} + \text{in}_2)f, T(\text{id} + \text{in}_2)g]^\dagger = [[\eta, g^\dagger]^* f, g^\dagger] \quad (7)$$

where $f : X \rightarrow_2 T(Y + Z)$ and $g : Z \rightarrow_2 T(Y + Z)$. Under the further axioms we are assuming these two instances imply the full *Bekić identity*:

$$[T[\text{id} + \text{in}_1, \text{in}_2]^* f, T[\text{id} + \text{in}_1, \text{in}_2]^* g]^\dagger = [h^\dagger, [\eta, h^\dagger]^* g^\dagger] \quad (8)$$

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

Now, the proof of (8) runs as follows:

$$\begin{aligned} & [T[\text{id} + \text{in}_1, \text{in}_2] f, T[\text{id} + \text{in}_1, \text{in}_2] g]^\dagger \\ &= (T[\text{id}, \text{in}_2] [T((\text{id} + \text{in}_1) + \text{in}_2) f, T((\text{id} + \text{in}_1) + \text{in}_2) g])^\dagger \\ &= [T((\text{id} + \text{in}_1) + \text{in}_2) f, T((\text{id} + \text{in}_1) + \text{in}_2) g]^\dagger \quad // \text{codiagonal} \\ &= [[\eta, (T((\text{id} + \text{in}_1) + \text{id}) g)^\dagger]^* T((\text{id} + \text{in}_1) + \text{id}) f, \\ &\quad (T((\text{id} + \text{in}_1) + \text{id}) g)^\dagger]^\dagger \quad // (7) \\ &= [[\eta(\text{id} + \text{in}_1), (T((\text{id} + \text{in}_1) + \text{id}) g)^\dagger]^* f, \\ &\quad (T((\text{id} + \text{in}_1) + \text{id}) g)^\dagger]^\dagger \quad // \text{naturality} \\ &= [T(\text{id} + \text{in}_1) [\eta, g^\dagger]^*, T(\text{id} + \text{in}_1) g^\dagger]^\dagger \\ &= [T(\text{id} + \text{in}_1) h, T(\text{id} + \text{in}_1) g^\dagger]^\dagger \\ &= [h^\dagger, [\eta, h^\dagger]^* g^\dagger]. \quad // (5) \end{aligned}$$

Finally, let us derive dinaturality from (8). Suppose that $g : X \rightarrow_2 T(Y + Z)$ and $h : Z \rightarrow T(Y + X)$ satisfy either guardedness premise of the dinaturality axiom and let us form

$$\begin{aligned} s &= [\eta \text{in}_1, g]^* h : Z \rightarrow_2 T(Y + Z), \\ t &= [\eta \text{in}_1, h]^* g : X \rightarrow_2 T(Y + X), \\ w &= T(\text{in}_1 + \text{id}) s : Z \rightarrow_{12,2} T((Y + X) + Z), \\ u &= T(\text{in}_1 + \text{id}) g : X \rightarrow_{12,2} T((Y + X) + Z). \end{aligned}$$

By naturality, $w^\dagger = (T \text{in}_1) s^\dagger$ and therefore,

$$\begin{aligned} ([\eta, w^\dagger]^* u)^\dagger &= ([\eta \text{in}_1, (T \text{in}_1) w^\dagger]^* g)^\dagger \\ &= ([\eta \text{in}_1, (T \text{in}_1) s^\dagger]^* g)^\dagger \\ &= [\eta, ([\eta \text{in}_1, (T \text{in}_1) s^\dagger]^* g)^\dagger]^* [\eta \text{in}_1, (T \text{in}_1) s^\dagger]^* g \\ &= [\eta, s^\dagger]^* g. \end{aligned}$$

By instantiating f in (8) with u and g with w we obtain:

$$[T(\text{id} + \text{in}_1) t, T(\text{id} + \text{in}_2) g]^\dagger = [[\eta, s^\dagger]^* g, s^\dagger].$$

Note that by (4),

$$[T(\text{id} + \text{in}_1)t, T(\text{id} + \text{in}_2)g]^\dagger \text{in}_1 = t^\dagger,$$

and therefore

$$[\eta, s^\dagger]^\star g = t^\dagger,$$

which was our goal. The case of in_2 -guarded h is derived analogously. \square

A.6 Proof of Theorem 17

By Proposition 16 we need to verify naturality, codiagonal and uniformity.

– *Naturality.* Let $f : X \rightarrow_2 T(Y + X)$ and $g : Y \rightarrow TZ$. Then

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

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

– *Codiagonal.* Let $f : X \rightarrow_{12,2} T((Y + X) + X)$. Then

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

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

– *Uniformity.* Suppose that $f h = T(\text{id} + h)g$ for some $f : X \rightarrow_2 T(Y + X)$, $g : Z \rightarrow_2 T(Y + Z)$ and $h : Z \rightarrow X$. Then

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

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

A.7 Proof of Theorem 19

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

$$\begin{aligned} f^\ddagger &= \rho(vf)^\dagger \\ &= \rho[\eta, (vf)^\dagger]^\star vf \\ &= [\rho\eta, \rho(vf)^\dagger]^\star \rho vf && // \rho \text{ is a mon. morph.} \\ &= [\eta, f^\ddagger]^\star f. && // \text{since } \rho v = \text{id} \end{aligned}$$

\square

A.8 Proof of Theorem 21

Recall that under our assumptions (2) is equivalent to (3).

1. *Naturality*:

$$\begin{aligned}
g^* f^\ddagger &= g^* \rho(v f)^\dagger \\
&= (\rho v g)^* \rho(v f)^\dagger && // \text{ since } \rho v = \text{id} \\
&= \rho(v g)^* (v f)^\dagger && // \rho \text{ is a mon. morph.} \\
&= \rho([(T \text{ in}_1) v g, \eta \text{ in}_2]^* v f)^\dagger && // \text{ naturality for } -^\dagger \\
&= \rho(v \rho[(T \text{ in}_1) v g, \eta \text{ in}_2]^* v f)^\dagger && // (3) \\
&= \rho(v[\rho(T \text{ in}_1) v g, \eta \text{ in}_2]^* \rho v f)^\dagger && // \rho \text{ is a mon. morph.} \\
&= \rho(v[(S \text{ in}_1) g, \eta \text{ in}_2]^* f)^\dagger && // \text{ since } \rho v = \text{id} \\
&= [(S \text{ in}_1) g, \eta \text{ in}_2]^* f)^\ddagger.
\end{aligned}$$

2. *Dinaturality*: First observe that it follows from the fact that ρ is a monad morphism and $\rho v = \text{id}$ that $\rho v[\eta \text{ in}_1, h]^* g = \rho[\eta \text{ in}_1, v h]^* v g$, and therefore, by (2) that

$$\rho(v[\eta \text{ in}_1, h]^* g)^\dagger = \rho([\eta \text{ in}_1, v h]^* v g)^\dagger. \quad (9)$$

Then we obtain the goal as follows:

$$\begin{aligned}
([\eta \text{ in}_1, h]^* g)^\ddagger &= \rho(v[\eta \text{ in}_1, h]^* g)^\dagger \\
&= \rho([\eta \text{ in}_1, v h]^* v g)^\dagger && // (9) \\
&= \rho[\eta, ([\eta \text{ in}_1, v g]^* v h)^\dagger]^* v g && // \text{ dinaturality for } -^\dagger \\
&= [\eta, \rho([\eta \text{ in}_1, v g]^* v h)^\dagger]^* \rho v g && // \rho \text{ is a mon. morph.} \\
&= [\eta, \rho([\eta \text{ in}_1, v g]^* v h)^\dagger]^* g && // \text{ since } \rho v = \text{id} \\
&= [\eta, \rho(v[\eta \text{ in}_1, g]^* h)^\dagger]^* g && // \text{ analog. to (9)} \\
&= [\eta, ([\eta \text{ in}_1, g]^* h)^\ddagger]^* g.
\end{aligned}$$

3. *Codiagonal*:

$$\begin{aligned}
(f^\ddagger)^\ddagger &= \rho(v \rho(v f)^\dagger)^\dagger \\
&= \rho((v f)^\dagger)^\dagger && // (3) \\
&= \rho(T[\text{id}, \text{in}_2] v f)^\dagger && // \text{ codiagonal for } -^\dagger \\
&= \rho(v \rho T[\text{id}, \text{in}_2] v f)^\dagger && // (3) \\
&= \rho(v S[\text{id}, \text{in}_2] f)^\dagger && // \text{ since } \rho v = \text{id} \\
&= (S[\text{id}, \text{in}_2] f)^\ddagger.
\end{aligned}$$

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

$$v f h = v S(\text{id} + h) g$$

$$\begin{aligned}
&= v S(\text{id} + h) \rho v g \\
&= T(\text{id} + h) v g
\end{aligned}$$

and therefore $(v f)^\dagger h = (v g)^\dagger$. This implies $f^\ddagger h = g^\ddagger$ by definition. \square

A.9 Proof of Theorem 13

By currying we can equivalently see $\#$ as a functor from \mathbf{D} to the category of parametrized guarded monads of type $\mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$. It therefore suffices to show that the assignment

$$\# \mapsto F_\# \quad \text{where} \quad F_\# X = \nu \gamma. X \# \gamma$$

and $f : X \rightarrow F_\# Y$ is σ -guarded iff $\text{out } f : X \rightarrow Y \# F_\# Y$ is σ -guarded, where

$$\text{out} : F_\# X \rightarrow X \# F_\# X$$

is the coalgebra structure of $F_\# X$, induces a functor N from parametrized guarded monads of type $\mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$ to guarded monads over \mathbf{C} . Uustalu [29] already proves that $F_\#$ is a monad; we proceed to check that his construction is in fact functorial.

We denote the monad structure on $F_\#$ by $\eta^\nu, (-)^\star$. These data are uniquely determined by commutation of

$$\begin{array}{ccc}
X & \xrightarrow{\eta_X^\nu} & F_\# X \\
\eta_{X,X} \downarrow & & \downarrow \text{out} \\
X \# X & \xrightarrow{X \# \eta_X} & X \# F_\# X
\end{array}
\qquad
\begin{array}{ccc}
F_\# X + F_\# Y & \xrightarrow{[f^\star, \text{id}]} & F_\# Y \\
[\hat{f}, (Y \# \text{in}_2) \text{out}] \downarrow & & \downarrow \text{out} \\
Y \# (F_\# X + F_\# Y) & \xrightarrow{Y \# [f^\star, \text{id}]} & Y \# (F_\# Y)
\end{array}$$

where

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

with

$$\bar{f} = X \xrightarrow{f} F_\# X \xrightarrow{\text{out}} X \# F_\# X \xrightarrow{Y \# \text{in}_2} Y \# (F_\# X + F_\# Y).$$

That is, η_X^ν is the unique $(X \# -)$ -coalgebra morphism $(X, \eta_{X,X}) \rightarrow (F_\# X, \text{out})$, and $[f^\star, \text{id}]$ is the unique $(Y \# -)$ -coalgebra morphism

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

the latter being essentially a definition of f^\star by primitive corecursion. In the sequel, we will omit the object part of coalgebras when convenient, saying, e.g., that η_X^ν is a coalgebra morphism $\eta_{X,X} \rightarrow \text{out}$.

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

$$\alpha : \# \rightarrow \#'$$

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

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

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

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

$$(F_\alpha)_X \eta_X^\nu = \eta_{X'}^\nu : X \rightarrow F_{\#'} X.$$

We note that by naturality of α , the $(X \#-)$ -coalgebra morphism $\eta_X^\nu : \eta_{X,X} \rightarrow \text{out}$ is also an $(X \#'-)$ -coalgebra morphism $\eta_{X,X}' = \alpha \eta_{X,X} \rightarrow \alpha \text{out}$, so that $(F_\alpha)_X \eta_X^\nu$ is a coalgebra morphism $\eta_{X,X}' \rightarrow \text{out}'$ and hence equals $\eta_{X'}^\nu$.

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

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

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

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

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

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

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

$$[\alpha \hat{f}, \alpha(Y \#' \text{in}_2) \text{out}] \rightarrow \text{out}'.$$

It therefore suffices to show that

$$(F_\alpha)_X + (F_\alpha)_Y : [\alpha \hat{f}, \alpha(Y \#' \text{in}_2) \text{out}] \rightarrow [(\widehat{F_\alpha})_Y f, (Y \#' \text{in}_2) \text{out}']$$

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

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

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

$$(Y \#'((F_{\alpha})_X + (F_{\alpha})_Y))\alpha \hat{f} = \widehat{(F_{\alpha})_Y} f (F_{\alpha})_X. \quad (10)$$

We rewrite the left hand side of (10):

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

We next rewrite the right hand side of (10):

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

It thus suffices to show that

$$(Y \#'((F_{\alpha})_X + (F_{\alpha})_Y))(\alpha \bar{f})^* = \overline{(F_{\alpha})_Y} f^* (X \#'((F_{\alpha})_X + (F_{\alpha})_Y)). \quad (11)$$

We further rewrite the right hand side of (11):

$$\begin{aligned}
& \overline{(F_{\alpha})_Y} f^* (X \#'((F_{\alpha})_X + (F_{\alpha})_Y)) \\
&= ((Y \#' \text{in}_2) \text{out}'(F_{\alpha})_Y f)^* \\
&\quad (X \#'((F_{\alpha})_X + (F_{\alpha})_Y)) && // \text{ definition of } \overline{(F_{\alpha})_Y} f \\
&= ((Y \#' \text{in}_2)(Y \#'(F_{\alpha})_Y)\alpha \text{out } f)^* \\
&\quad (X \#'((F_{\alpha})_X + (F_{\alpha})_Y)) && // \text{ definition of } (F_{\alpha})_Y \\
&= ((Y \#'((F_{\alpha})_X + (F_{\alpha})_Y))(Y \#' \text{in}_2) \alpha \text{out } f)^* \\
&\quad (X \#'((F_{\alpha})_X + (F_{\alpha})_Y)) \\
&= (Y \#'((F_{\alpha})_X + (F_{\alpha})_Y))((Y \#' \text{in}_2) \alpha \text{out } f)^*
\end{aligned}$$

where we use in the last step that $Y \#' ((F_\alpha)_X + (F_\alpha)_Y)$ is a monad morphism. We have thus reduced (11) to showing that

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

But this is straightforward:

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

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

– (**trv**) Let $f : X \rightarrow F_\# Y$. Then

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

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

– (**wkn**) Let $f : X \rightarrow_\sigma F_\# Y$, which by definition means that $\text{out } f : X \rightarrow_\sigma Y \# (F_\# Y)$. By (**wkn**) for $\#$, $\text{out } f$ is $\sigma\theta$ -guarded and hence, by definition, so is f .

– (**cmp**) Let $f : X \rightarrow_2 F_\# (Y + Z)$, $g : Y \rightarrow_\sigma F_\# V$, $h : Z \rightarrow F_\# V$. Then we obtain

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

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

– (**sum**) Let $f_i : X_i \rightarrow_\sigma F_\# Y$ for $i = 1, 2$, which by definition means that $\text{out } f_i : X_i \rightarrow_\sigma Y \# (F_\# Y)$. By (**sum**) for $\#$, $\text{out}[f_1, f_2] = [\text{out } f_1, \text{out } f_2] : X_1 + X_2 \rightarrow_\sigma Y \# (F_\# Y)$, so that $[f_1, f_2] : X_1 + X_2 \rightarrow_\sigma F_\# Y$ as required.

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

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

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

A.10 Proof of Theorem 22

Direction (\Leftarrow) immediately follows from Theorems 21 and 17.

In order to prove (\Rightarrow) , we show that $\mathbb{S} = (S, \eta, -^*, -^\dagger)$ is an iteration-congruent retract of $\mathbb{S}^\nu = (\nu\gamma. S(-+\gamma), \eta^\nu, -^*, -^\dagger)$. We define $v_X = \text{out}^{-1}\eta \text{in}_2 \text{out}^{-1}(S \text{in}_1)$ and

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

Clearly, $v f$ is σ -guarded for every $f : X \rightarrow SY$ and v is left inverse to ρ , for

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

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

$$\rho(\text{coit } h) = h^\dagger. \quad (12)$$

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

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

Next, we have to check that $\rho f^* = (\rho f)^* \rho$ for any $f : X \rightarrow S^\nu Y$. Note that

$$f^* = \text{coit}([\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f], \eta \text{in}_2]^* \text{out} (S^\nu \text{in}_2).$$

Therefore

$$\begin{aligned} \rho f^* &= \rho \text{coit}([\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f], \eta \text{in}_2]^* \text{out} (S^\nu \text{in}_2) \\ &= ([\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f], \eta \text{in}_2]^* \text{out})^\dagger (S^\nu \text{in}_2) && // (12) \\ &= (S[\text{id}, \text{in}_2] [(S \text{in}_1) [\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f], \\ &\quad \eta \text{in}_2]^* \text{out})^\dagger (S^\nu \text{in}_2) \\ &= (((S \text{in}_1) [\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f], \\ &\quad \eta \text{in}_2]^* \text{out})^\dagger)^\dagger (S^\nu \text{in}_2) && // \text{codiagonal} \\ &= ([\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f]^* \text{out}^\dagger)^\dagger (S^\nu \text{in}_2) && // \text{naturality} \\ &= ([\eta \text{in}_1, S(\text{id} + S^\nu \text{in}_1) \text{out } f]^* \rho)^\dagger (S^\nu \text{in}_2) \\ &= [\eta, ([\eta \text{in}_1, \rho]^* S(\text{id} + S^\nu \text{in}_1) \text{out } f)^\dagger]^* \rho (T \text{in}_2) && // \text{dinaturality} \\ &= [\eta, ([\eta \text{in}_1, \rho]^* S(\text{id} + S^\nu \text{in}_1) \text{out } f)^\dagger]^* (S \text{in}_2) \rho \\ &= ((([\eta \text{in}_1, \rho]^* S(\text{id} + S^\nu \text{in}_1) \text{out } f)^\dagger)^* \rho \\ &= (((S \text{in}_1) [\eta, \rho]^* \text{out } f)^\dagger)^* \rho \end{aligned}$$

$$\begin{aligned}
&= (((S \text{ in}_1) \rho f)^\dagger)^\star \rho \\
&= ([\eta, ((S \text{ in}_1) \rho f)^\dagger]^\star (S \text{ in}_1) \rho f)^\star \rho && // \text{ fixpoint} \\
&= (\rho f)^\star \rho.
\end{aligned}$$

Finally, let us check that ρ is an iteration congruence. Let $f, g : X \rightarrow_2 S^\nu(Y + X)$, which means that there are $f', g' : X \rightarrow S(Y + S^\nu(X + Y))$ such that $\text{out } f = S(\text{in}_1 + \text{id})f'$ and $\text{out } g = S(\text{in}_1 + \text{id})g'$. Suppose that $\rho f = \rho g$, which amounts to

$$[\eta \text{ in}_1, \rho]^\star f' = [\eta \text{ in}_1, \rho]^\star g', \quad (13)$$

for

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

and analogously for g . Our goal is to prove that

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

from which $\rho f^\dagger = \rho g^\dagger$ will follow from the analogous formula for ρg^\dagger and (13). Observe that

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

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

$$\begin{aligned}
\rho f^\ddagger &= \rho (\text{coit } h) \eta^\nu \text{ in}_2 \\
&= h^\dagger \eta^\nu \text{ in}_2 && // (12) \\
&= ([[\eta \text{ in}_1, f'], \eta \text{ in}_2]^\star \text{out})^\dagger \eta^\nu \text{ in}_2 \\
&= (S[\text{id}, \text{in}_2] [[\eta \text{ in}_1 \text{ in}_1, (S \text{ in}_1)f'], \eta \text{ in}_2]^\star \text{out})^\dagger \eta^\nu \text{ in}_2 \\
&= ([[\eta \text{ in}_1 \text{ in}_1, (S \text{ in}_1)f'], \eta \text{ in}_2]^\star \text{out})^\dagger \eta^\nu \text{ in}_2 && // \text{codiagonal} \\
&= (((S \text{ in}_1)[\eta \text{ in}_1, f'], \eta \text{ in}_2]^\star \text{out})^\dagger \eta^\nu \text{ in}_2 \\
&= ([\eta \text{ in}_1, f']^\star \text{out})^\dagger \eta^\nu \text{ in}_2 && // \text{naturality} \\
&= ([\eta \text{ in}_1, f']^\star \rho)^\dagger \eta^\nu \text{ in}_2 \\
&= [\eta, ([\eta \text{ in}_1, \rho]^\star f')^\dagger]^\star \rho \eta^\nu \text{ in}_2 && // \text{dinaturality} \\
&= [\eta, ([\eta \text{ in}_1, \rho]^\star f')^\ddagger]^\star \eta^\nu \text{ in}_2
\end{aligned}$$

and we are done. \square

A.11 Proof of Theorem 23

It is already shown in Theorem 13 that ρ^ν is a monad morphism.

Let us define $v^\nu = \text{coit}(v \text{ out})$. Then it is easy to check that $\rho^\nu v^\nu = \text{id}$: since

$$\text{out } \rho^\nu v^\nu = (\text{id} \# \rho^\nu) \rho \text{ out } v^\nu$$

$$\begin{aligned}
&= (\text{id} \hat{\#} \rho^\nu) \rho(\text{id} \# v^\nu) v \text{ out} \\
&= (\text{id} \hat{\#} \rho^\nu)(\text{id} \hat{\#} v^\nu) \rho v \text{ out} \\
&= (\text{id} \hat{\#} \rho^\nu v^\nu) \text{ out}.
\end{aligned}$$

and also $\text{out id} = (\text{id} \hat{\#} \text{id}) \text{ out}$, $\rho^\nu v^\nu$ must be equal id .

Suppose that $f : X \rightarrow Y \hat{\#}^\nu Z$ is σ -guarded and let us show that so is $v^\nu f : X \rightarrow Y \#^\nu Z$. According to Theorem 13, this means that $\text{out } f$ is σ -guarded. Now

$$\text{out } v^\nu f = (\text{id} \# v^\nu) v \text{ out } f$$

is σ -guarded because by assumption $v \text{ out } f$ is σ -guarded and $\text{id} \# v^\nu$ is a parametrized guarded monads morphism. Hence, again, according to Theorem 13, $v^\nu f$ is σ -guarded. We 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 $f, g : X \rightarrow_2 (Y + X) \hat{\#}^\nu Z$. Then

$$\rho(\text{id} \# \rho^\nu) \text{ out } f = (\text{id} \hat{\#} \rho^\nu) \rho \text{ out } f = \text{out } \rho^\nu f$$

and analogously for g , so by applying the assumption that ρ is an iteration congruence, we obtain

$$\rho((\text{id} \# \rho^\nu) \text{ out } f)^\dagger = \rho((\text{id} \# \rho^\nu) \text{ out } g)^\dagger. \quad (14)$$

Observe that for suitably typed h ,

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

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

$$\text{coit}(\rho \text{ out}) (\text{coit } h) = \text{coit}(\rho h). \quad (15)$$

Therefore,

$$\begin{aligned}
\rho^\nu f^\ddagger &= \rho^\nu \text{coit}([\eta, (\text{out } f)^\dagger]^\star \text{out}) \eta \text{ in}_2 && // \text{Theorem 13} \\
&= \text{coit}(\rho \text{ out}) \text{coit}([\eta, (\text{out } f)^\dagger]^\star \text{out}) \eta \text{ in}_2 \\
&= \text{coit}(\rho[\eta, (\text{out } f)^\dagger]^\star \text{out}) \eta \text{ in}_2 && // (15) \\
&= \text{coit}([\eta, \rho((\text{id} \# \rho^\nu) \text{ out } f)^\dagger]^\star \text{out}) \rho^\nu \eta \text{ in}_2.
\end{aligned}$$

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

$$\begin{aligned}
&\text{out coit}([\eta, \rho((\text{id} \# \rho^\nu) \text{ out } f)^\dagger]^\star \text{out}) \rho^\nu \\
&= (\text{id} \hat{\#} \text{coit}([\eta, \rho((\text{id} \# \rho^\nu) \text{ out } f)^\dagger]^\star \text{out}))
\end{aligned}$$

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

An analogous calculation can be done for $\rho^\nu g^\ddagger$ and therefore by (14), $\rho^\nu f^\ddagger = \rho^\nu g^\ddagger$. \square

A.12 Proof of Theorem 26

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

$$\rho = [\eta, \rho]^\star \text{out}. \quad (16)$$

Indeed,

$$\begin{aligned}
[\eta, \rho]^\star \text{out} &= [\eta, \rho]^\star \rho\eta \text{out} && // \text{for } \rho\eta = \text{id} \\
&= [\eta, \rho \triangleright]^\star \rho\eta \text{out} && // \text{for } \rho \triangleright = \rho \\
&= \rho [\eta, \triangleright]^\star \eta \text{out} && // \rho \text{ is mon. morph.} \\
&= \rho [\eta, \triangleright]^\star \text{out}^{-1}(S \text{in}_1) \text{out} \\
&= \rho \text{out}^{-1}[\text{out}[\eta, \text{id}], \eta \text{in}_2[\eta, \text{id}]^\star]^\star (S \text{in}_1) \text{out} \\
&= \rho \text{out}^{-1}[\eta \text{in}_1, \eta \text{in}_2]^\star \text{out} \\
&= \rho.
\end{aligned}$$

Let \mathbb{S} be an Elgot monad, hence by construction from Theorem 22, it is an iteration-congruent retract of \mathbb{S}^ν with $S^\nu = \nu\gamma. S(X + \gamma)$, and more specifically, $v = \triangleright \eta$ is a left inverse to $\rho = \text{out}^\dagger$.

First of all, it is easy to see that $\rho \triangleright = \rho$:

$$\begin{aligned}
\rho \triangleright &= [\eta, \rho]^\star \text{out} \triangleright \\
&= [\eta, \rho]^\star \eta \text{in}_2 \\
&= \rho.
\end{aligned}$$

We need to show the axioms of $(-)^^\nu$ -algebras:

$$\rho\eta = \text{id} \quad \text{and} \quad \rho\mu = \rho\rho^\nu \quad (17)$$

where $\rho^\nu = \text{coit}(\rho \text{ out}) : S^{\nu\nu} \rightarrow S^\nu$. As for the left one we readily have $\text{id} = \rho\nu = \rho \triangleright \eta = \rho\eta$. The right axiom is shown as follows:

$$\rho\mu \stackrel{(i)}{=} \rho[\eta, (\triangleright \text{out})^\ddagger]^* \text{out} \stackrel{(ii)}{=} \rho(\text{out}^{-1} S(\text{in}_1 + \eta \text{in}_2) \rho \text{out})^\ddagger \stackrel{(iii)}{=} \rho\rho^\nu.$$

To show transition (i), first observe that

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

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

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

which means that $t \text{out}$ satisfies the equation uniquely characterizing μ , hence $\mu = t \text{out}$. On the other hand, $[\eta, (\triangleright \text{out})^\ddagger]^*$ satisfies the equation characterizing t . In order to see that, note that

$$\text{out}(\triangleright \text{out})^\ddagger = \eta \text{in}_2[\eta^\nu, (\triangleright \text{out})^\ddagger]^* \quad (18)$$

and therefore

$$\begin{aligned} \text{out}[\eta^\nu, (\triangleright \text{out})^\ddagger]^* &= [\text{out}[\eta^\nu, (\triangleright \text{out})^\ddagger], \eta \text{in}_2[\eta^\nu, (\triangleright \text{out})^\ddagger]^*]^* \text{out} \\ &= [\text{out}[\eta^\nu, (\triangleright \text{out})^\ddagger], \eta \text{in}_2[\eta^\nu, (\triangleright \text{out})^\ddagger]^*]^* \text{out} \\ &= S(\text{id} + [\eta^\nu, (\triangleright \text{out})^\ddagger]^*) S[\text{id} + \text{out}, \text{in}_2] \text{out} \quad // (18) \end{aligned}$$

In summary we obtain

$$\mu = \text{coit}(S[\text{id} + \text{out}, \text{in}_2] \text{out}) \text{out} = [\eta^\nu, (\triangleright \text{out})^\ddagger]^* \text{out},$$

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

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

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 \triangleright , we obtain that

$$\begin{aligned} \rho[\eta, (\triangleright \text{out})^\ddagger]^* &= [\eta, \rho(\triangleright \text{out})^\ddagger]^* \rho \\ &= [\eta, \rho(\triangleright \text{out})^\ddagger]^* \rho \triangleright \end{aligned}$$

$$\begin{aligned}
&= \rho[\eta, (\triangleright \text{out})^\dagger]^\star \rho \triangleright \\
&= \rho(\triangleright \text{out})^\dagger.
\end{aligned}$$

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

$$\rho \triangleright \text{out} = \rho \text{out}^{-1} S(\text{in}_1 + \eta \text{in}_2) \rho \text{out} \quad (19)$$

and call the assumption that ρ is an iteration congruence. The proof of (19) runs as follows:

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

Suppose that $\alpha : \mathbb{S} \rightarrow \mathbb{T}$ 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 $(-)^{\nu}$ -algebras, i.e. $\alpha \rho = \rho \alpha^{\nu}$. Indeed, on the one hand $\alpha \rho = \alpha \text{out}^\dagger = (\alpha \text{out})^\dagger$, and also on the other hand, by uniformity of $-^\dagger$, $\rho \alpha^{\nu} = \text{out}^\dagger \alpha^{\nu} = (\alpha \text{out})^\dagger$, for $\text{out} \alpha^{\nu} = T(\text{id} + \alpha^{\nu}) \alpha \text{out}$.

We proceed with the other direction of the claim. Assume that (\mathbb{S}, ρ) is a $(-)^{\nu}$ -algebra, i.e. the laws (17) are satisfied, and $\rho \triangleright = \rho$ and let us show that \mathbb{S} equipped with the iteration operation $f^\dagger = \rho(\text{coit } f)$ is an Elgot monad by verifying the axioms.

- *Fixpoint.* Let $f : X \rightarrow S(Y + X)$. Then $f^\ddagger = \rho(\text{coit } f)$ and hence

$$\begin{aligned}
f^\ddagger &= \rho(\text{coit } f) \\
&= [\eta, \rho]^\star \text{out}(\text{coit } f) \\
&= [\eta, \rho]^\star S(\text{id} + \text{coit } f) f && // (16) \\
&= [\eta, \rho \text{coit } f]^\star f \\
&= [\eta, f^\ddagger]^\star f.
\end{aligned}$$

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

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

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

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

- *Coadiagonal.* Let $f : X \rightarrow S((Y + X) + X)$.

$$\begin{aligned}
f^{\dagger\dagger} &= \rho(\text{coit } \rho(\text{coit } f)) \\
&= \rho \rho^\nu(\text{coit}(\text{coit } f)) && // \text{ definition of } \rho^\nu \\
&= \rho \boldsymbol{\mu}(\text{coit}(\text{coit } f)). && // (17)
\end{aligned}$$

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

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

- *Uniformity.* Let $f : X \rightarrow T(Y + X)$, $g : Z \rightarrow S(Y + Z)$, $h : Z \rightarrow X$ and suppose that $f h = T(\text{id} + h) g$. If follows standardly by the finality argument that $(\text{coit } f) h = \text{coit } g$ and therefore

$$\begin{aligned}
f^\ddagger h &= \rho(\text{coit } f) h \\
&= \rho(\text{coit } g) \\
&= g^\ddagger.
\end{aligned}$$

Finally, let us check that a $(-)^{\nu}$ -algebra morphism $\alpha : \mathbb{T} \rightarrow \mathbb{S}$ is an Elgot monad morphism. As a premise we have that $\alpha \rho = \rho \alpha^\nu$, and therefore, for every $f : X \rightarrow S(Y + X)$, $\alpha f^\dagger = \alpha \rho(\text{coit } f) = \rho \alpha^\nu(\text{coit } f) = \rho \text{coit}(\alpha \text{out})(\text{coit } f)$. It is then straightforward to verify that $\rho \text{coit}(\alpha \text{out})(\text{coit } f) = \rho \text{coit}(\alpha f) = (\alpha f)^\dagger$. \square

A.13 Proof of Theorem 28

First, we need to verify that the defined guardedness relation satisfies the rules from Definition 1. Note that since left adjoints preserve coproducts (LAPC), without loss of generality we assume that $F(X + Y) = FX + FY$.

- (**trv**) Let $f : X \rightarrow UTFY$ be a morphism. By (**trv**) for \mathbb{T} , it is the case that $(T \text{in}_1)(\Phi^{-1} f) : FX \rightarrow_\sigma T(FY + FX)$. Then, the following holds:

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

Thus, it is the case that $\Phi^{-1}((UTF \text{in}_1) f) : FX \rightarrow_\sigma T(FY + FX)$, so $(UTF \text{in}_1) f : X \rightarrow_\sigma UTF(Y + X)$.

- **(wkn)** Let $f : X \rightarrow_{\sigma} UTFY$. Then, $\Phi^{-1}f : FX \rightarrow_{\sigma} TFY$ in \mathbb{T} . By **(wkn)** we obtain $\Phi^{-1}f : FX \rightarrow_{\sigma\theta} TFY$ in \mathbb{T} , hence $f : X \rightarrow_{\sigma\theta} UTFY$ in the monad on UTF .
- **(cmp)** Let $f : X \rightarrow_2 UTF(Y + Z)$, $g : Y \rightarrow_{\sigma} UTFV$, and $h : Z \rightarrow UTFV$ be morphisms. Then, by **(cmp)** for \mathbb{T} , we obtain that $[\Phi^{-1}g, \Phi^{-1}h]^*(\Phi^{-1}f) : FX \rightarrow_{\sigma} TFV$. Then, the following holds:

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

Thus, it is the case that $\Phi^{-1}([g, h]^*f) : X \rightarrow_{\sigma} UTF(Y + X)$ in \mathbb{T} , so $[g, h]^*f$ in the monad on UTF .

This means that if $f : X \rightarrow_2 UTF(Y + X)$, then it is the case that $\Phi^{-1}f : FX \rightarrow_2 T(FY + FX)$, 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 [26]. \square

A.14 Proof of Theorem 29

By Theorem 22, there exists a guarded iterative monad \mathbb{T} and an iteration-congruent retraction $\rho : \mathbb{T} \rightarrow \mathbb{S}$. By Theorem 28, the monad induced on UTF is guarded iterative. Thus, it is enough to show that $U\rho F : UTF \rightarrow USF$ is an iteration-congruent retraction.

- It is a retraction, since retractions are preserved by all functors.
- To see that it is guarded, let $f : X \rightarrow USFY$ be σ -guarded. By definition, it means that $\Phi^{-1}f : FX \rightarrow SFY$ is σ -guarded in \mathbb{S} . Since ρ is a guarded retraction, it follows that $v(\Phi^{-1}f)$, for ρ 's family of sections v , is also σ -guarded. By the fact that Φ is a natural isomorphism, we obtain $v(\Phi^{-1}f) = \Phi^{-1}((Uv)f)$, hence, by definition, $(Uv)f$ is also σ -guarded.
- To see that $U\rho F$ is an iteration congruence, let us denote by $(-)^{\dagger}$ the solution in \mathbb{T} , and by $(-)^{\ddagger}$ the solution in the monad on UTF . Let $f, g : X \rightarrow_2 UTF(X + Y)$ 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:

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

□

A.15 Details for Example 30

As an example of the setting described in Theorem 22, 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 non-expansive maps. Note that **CbUMet** has coproducts and is Cartesian closed [28]. Following Escardo [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

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

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

Theorem 31. *The monad T is guarded iterative with $f : X \rightarrow TY$ being σ -guarded if for all x and y , it is the case that $f(x) = (\sigma(y), k)$ implies $k > 0$.*

Proof. First, note that the product of (A, d_A) and (B, d_B) in **Cbus** 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

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

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

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

Given a guarded morphism $f : X \rightarrow_1 T(Y + X)$, we define the morphism $f^\dagger : X \rightarrow TY$ as the unique fixed point of the following map $\psi : (X \rightarrow TY) \rightarrow (X \rightarrow TY)$:

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

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 $g, g' : X \rightarrow TY$, the following holds:

$$d_{X \Rightarrow TY}(\psi(g), \psi(g')) \leq c \cdot d_{X \Rightarrow TY}(g, g') \quad (20)$$

The left-hand side of the equation (20) 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:

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

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

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

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

So, the following holds:

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

- $f(x) = (\text{in}_2 y, k + 1)$ 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 it is the case that:

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

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

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

□

We thus obtain a monad $U_D T F_D$ on **Set** by sandwiching T in the adjunction $F_D \dashv U_D$ where U_D is the forgetful functor **CbUMet** \rightarrow **Set** and F_D takes discrete metrics. By Theorem 28, $U_D T F_D$ is guarded iterative.

For the second operator, let **Cpo** $_{\perp}$ be the category of complete partial orders and continuous bottom-preserving functions. The identity on **Cpo** $_{\perp}$ is an Elgot monad, hence, by Theorem 29, we obtain an Elgot monad $U_L F_L$ on **Set** by sandwiching in the adjunction $F_L \dashv U_L$ where U_L is the forgetful functor **Cpo** $_{\perp}$ \rightarrow **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 32. *Define $\rho : U_D T F_D \rightarrow U_L F_L$ by $\rho(a, k) = a$ and $\rho(\infty) = \perp$. Then ρ is an iteration-congruent retraction with the section given by $v(a) = (a, 1)$ and $v(\perp) = \infty$. 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 retraction congruence, we first define an auxiliary relation: given two functions $f, h : X : (B \times \mathbb{N}) \cup \{\infty\}$, we write $f \sim g$ if it is the case that $f(x) = (a, k)$ for some $a \in B$, and $k \in \mathbb{N}$ 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 $\psi_f(g) = [\eta, g]^* f$ for the function ψ from the proof of Theorem 31.

Given a 2-guarded function $f : X \rightarrow ((Y + X) \times \mathbb{N}) \cup \{\infty\}$, the function f^\dagger can be defined as the unique fixed point of ψ_f (see the proof of Theorem 31), 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 $W_n^f(x)$ stabilises. Given a function h such that $f \sim h$, it is easy to show by induction that for every x , the sequence $W_n^f(x)$ stabilises with (a, k) for some k if and only if $W_n^h(x)$ stabilises 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 , it is the case that $\rho f = \rho h$ 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 $f : X \rightarrow (X + Y) \cup \{\perp\}$, its solution in the Elgot monad $U_L F_L$ is given by the fixed point of the equation $\phi(g) = [\eta, g]^* f$, 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 $c'(x) = \perp$. It is easy to see that $W'_n = \rho W_n^{(vf)}$, so the solutions coincide. \square