A Unifying Categorical View of Nondeterministic Iteration and Tests

Sergey Goncharov
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Tarmo Uustalu
Reykjavik University, Iceland, and Tallinn University of Technology, Estonia

Abstract

We study Kleene iteration in the categorical context. A celebrated completeness result by Kozen introduced Kleene algebra (with tests) as a ubiquitous tool for lightweight reasoning about program equivalence, and yet, numerous variants of it came along afterwards to answer the demand for more refined flavors of semantics, such as stateful, concurrent, exceptional, hybrid, branching time, etc. We detach *Kleene iteration* from *Kleene algebra* and analyze it from the categorical perspective. The notion, we arrive at is that of *Kleene-iteration category* (with coproducts and tests), which we show to be general and robust in the sense of compatibility with programming language features, such as exceptions, store, concurrent behaviour, etc. We attest the proposed notion w.r.t. various yardsticks, most importantly, by characterizing the free model as a certain category of (nondeterministic) *rational trees*.

2012 ACM Subject Classification Theory of computation → Categorical semantics; Theory of computation → Axiomatic semantics

Keywords and phrases Kleene iteration, Elgot iteration, Kleene algebra, coalgebraic resumptions

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2024.25

Related Version *Full Version*: `https://arxiv.org/abs/2407.08688`

Funding *Sergey Goncharov*: German Research Foundation (DFG) project 501369690, Icelandic Research Fund project 228684-052
Tarmo Uustalu: Icelandic Research Fund project 228684-052

1 Introduction

Axiomatizing notions of iteration both algebraically and categorically is a well-established topic in computer science where two schools of thought can be distinguished rather crisply: the first one is based on the inherently nondeterministic *Kleene iteration*, stemming from the seminal work of Stephen Kleene [22] and deeply rooted in automata and formal language theory; the second one stems from another seminal work – by Calvin Elgot [12] – and is based on another notion of iteration, we now call *Elgot iteration*. The most well-known instance of Kleene iteration is the one that is accommodated in the algebra of regular expressions where a^* represents n-fold compositions $a \cdots a$ and n nondeterministically ranges over all naturals. More abstractly, Kleene iteration is an operation of the following type:

$$\frac{p \colon X \to X}{p^* \colon X \to X}$$

Intuitively, we think of p as a program whose inputs and outputs range over X, and of p^* as a result of composing p nondeterministically many times with itself. Elgot iteration, in contrast, is agnostic to nondeterminism, but crucially relies on the categorical notion of binary coproduct, and thus can only be implemented in categorical or type-theoretic setting.

Concretely, the typing rule for Elgot iteration is

$$\frac{p\colon X \to Y + X}{p^\dagger : X \to Y} \tag{†}$$

That is, given a program that receives an input from X, and can output either to Y or to X, $p^\dagger$ self-composes p precisely as long as p outputs to X.

A profound exploration of both versions of iteration and their axiomatizations in the categorical context, more precisely, in the context of *Lavwere theories*, has been done by Bloom and Ésik in a series of papers and subsumed in their monograph [6]. One outcome of this work is that in the context of Lawvere theories, in presence of nondeterminism, Kleene iteration and Elgot iteration are essentially equivalent – the ensuing theory was dubbed *iteration grove theory* [5]. The existing analysis still does not cover certain aspects, which we expressly address in our present work, most importantly, the following.

- Lavwere theories are only very special categories, while iteration is a common ingredient of semantic frameworks, which often involve it directly via an ambient category with coproducts, and not via the associated Lavwere theory.
- Previous results on the equivalence of Elgot iteration and Kleene iteration do not address the connection between control mechanisms involved in both paradigms: Elgot iteration fully relies on coproducts for making decisions whether to continue or to end the loop, while Kleene iteration for the same purpose uses an additional mechanism of *tests* [24], which are specified axiomatically and thus yield a higher degree of flexibility.
- A key feature of Kleene iteration of Kleene algebra, are the quasi-equational laws, which can be recast [16] to a form of the versatile and powerful *uniformity principle* (e.g. [36]). The latter is parameterized by a class of well-behaved elements, which in Kleene algebra coincide with the algebra's entire carrier. However, in many situations, this class has to be restricted, which calls for axiomatizing it, analogously to tests.

Here, we seek a fundamental, general and robust categorical notion of Kleene iteration, which addresses these issues, is in accord with Elgot iteration and the corresponding established laws for it (Elgot iteration operators that satisfy these laws are called *Conway operators*). In doing so, we depart from the laws of Kleene algebra, and relax them significantly. Answering the question how to do this precisely and in a principled way is the main insight of our work.

Let us dwell briefly on the closely related issues of generality and robustness. The laws of Kleene algebra, as originally axiomatized by Kozen [23], capture a very concrete style of semantics, mirrored in the corresponding free model, which is the algebra of regular events, i.e. the algebra of regular sets of strings over a finite alphabet of symbols, with iteration rendered as a least fixpoint. Equations validated by this model are thus shared by the whole class of Kleene algebras. By regarding Kleene algebra terms as programs, the interpretation over the free model can be viewed as *finite trace semantics* of linear-time nondeterminism. A standard example of properly more fine-grained – branching-time – nondeterminism is (bisimulation-based) process algebra, which fails the Kleene algebra's law of distributivity from the left:

$$p \,;\, (q + r) = p \,;\, q + p \,;\, r. \tag{1}$$

Similarly, if we wanted to allow our programs to raise exceptions, the laws of Kleene algebra would undesirably force all exceptions to be equal:

$$\mathsf{raise}\, e_1 = \mathsf{raise}\, e_1 \,;\, 0 = 0 = \mathsf{raise}\, e_2 \,;\, 0 = \mathsf{raise}\, e_2.$$

Here, we combine the law $p\,;0 = 0$ of Kleene algebra with the equation $\mathsf{raise}_i\,; p = \mathsf{raise}_i$ that alludes to the common programming knowledge that raising an exception exits the program instantly and discards any subsequent fragment p. The resulting equality $\mathsf{raise}\,e_1 = \mathsf{raise}\,e_2$ states that raising exception e_1 is indistinguishable from raising exception e_2.

We can interpret these and similar examples as evidence that the axioms of Kleene algebra are not sufficiently robust under extensions by programming language features. More precisely, Kleene algebras can be scaled up to *Kleene monads* [17], and thus reconciled with Moggi's approach to computational effects [30]. An important ingredient of this approach are *monad transformers*, which allow for combining effects in a principled way. For example, one uses the *exception monad transformer* to canonically add exception raising to a given monad. The above indicates that Kleene monads are not robust under this transformer.

Finally, even if we accept all iteration-free implications of Kleene algebra, these will not jointly entail the following identity:

$$1^* = 1, \tag{2}$$

which is however entailed by the Kleene algebra axioms. One setting where (2) is undesirable is domain theory, which insists on distinguishing *deadlock* from *divergence*, in particular, (2) is failed by interpreting programs over the *Plotkin powerdomain* [33]. Intuitively, (2) states that, if a loop *may* be exited, it *will* eventually be exited, while failure of (2) would mean that the left program *may* diverge, while the right program *must* converge, and this need not be the same. Let us call the corresponding variant of Kleene algebra, failing (2), *may-diverge Kleene algebra*. However, it is not a priori clear how the axioms of may-diverge Kleene algebras must look like, given that (2) is not a Kleene algebra axiom, but a consequence of the assumption that Kleene iteration is a least fixpoint. Hence, in may-diverge Kleene algebras Kleene iteration is not a least fixpoint (w.r.t. the order, induced by $+$).

The notion we develop and present here is that of *Kleene-iteration category (with tests) (KiC(T))*. It is designed to address the above issues and to provide a uniform general and robust framework for Kleene iteration in a category. We argue in various ways that KiC(T) is in a certain sense the most basic practical notion of Kleene iteration, most importantly by characterizing its free model, as a certain category of (nondeterministic) rational trees.

Related work. The (finite or ω-complete) partially additive categories (PACs) by Arbib and Manes [2] and the PACs with effects of Cho [8] are similar in spirit to KiCs in that they combine structured homsets and coproducts, but significantly more special; in particular they support relational, sets of traces and similar semantics, but not branching time semantics. A PAC is a category with coproducts enriched in partial commutative monoids (PMC). The PMC structure of homsets and the coproducts are connected by axioms that make the PMC structure unique. In an ω-complete PAC, these axioms also ensure the presence of an Elgot iteration operator, which is computed as a least fixpoint. A PAC with effects comes with a designated effect algebra object; this defines a wide subcategory of total morphisms, with coproducts inherited from the whole category. Effectuses [21] achieve the same as PACs with effects, but starting with a category of total morphisms and then adding partial morphisms. Cockett [9] recently proposed a notion of iteration in a category, based on restriction categories, and analogous to Elgot iteration (†), but avoiding binary coproducts in favor of a suitably axiomatized notion of disjointness for morphisms.

In the strand of Kleene algebra, various proposals were made with utilitarian motivations to weaken or modify the Kleene algebra laws, and thus to cope with process algebra [14], branching behaviour [31], probability [27], statefulness [20], graded semantics [15], without however aiming to identify the conceptual core of Kleene iteration, which is our objective here. A recent move within this tendency is to eliminate nondeterminism altogether, with *guarded Kleene algebras* [37], which replace nondeterministic choice and iteration with conditionals and while-loops. This is somewhat related to our analysis of tests and iteration via while-loops, but largely orthogonal to our main objective to stick to Kleene iteration as nondeterministic operator in the original sense. Our aim to reconcile Kleene algebra, (co)products and Elgot iteration is rather close to that of Kozen and Mamouras [25].

Our characterization of the free KiCT in a way reframes the original Kozen's characterization of the free Kleene algebra [23]. We are not generalizing this result though, essentially because we work in categories with coproducts, while a true generalization would only be achieved via categories without any extra structure (noting that algebras are single-object categories). This distinction becomes particularly important in the context of branching time semantics, which we also cover by allowing a controlled use of programs that fail distributivity from the left (1). An axiomatization for such semantics has been proposed by Milner [29] and was shown to be complete only recently [19]. Again, we are not generalizing this result, since the definability issues, known to be the main obstruction for completeness arguments there, are not effective in presence of coproducts.

Plan of the paper. We review minimal notations and conventions from category theory in Section 2. We then introduce idempotent grove and Kleene-Kozen categories in Section 3 to start off. In Section 4, we formally compare two control mechanisms in categories: decisions and tests. In Sections 5, 6, we establish equivalent presentations of nondeterministic iteration as Kleene iteration, as Elgot iteration and as while-iteration. In Section 7 we construct a free model for our notion of iteration, and then come to conclusions in Section 8.

2 Notations and Conventions

We assume familiarity with the basics of category theory [26, 3]. In a category $\mathbf{C}$, $|\mathbf{C}|$ will denote the class of objects and $\mathbf{C}(X,Y)$ will denote the set of morphisms from X to Y. The judgement $f\colon X \to Y$ will be regarded as an equivalent to $f \in \mathbf{C}(X,Y)$ if $\mathbf{C}$ is clear from the context. We tend to omit indexes at natural transformations for readability. A subcategory $\mathbf{D}$ of $\mathbf{C}$ is called *wide* if $|\mathbf{C}| = |\mathbf{D}|$. We will use diagrammatic composition ; of morphisms throughout, i.e. given $f\colon X \to Y$ and $g\colon Y \to Z$, $f \,;\, g\colon X \to Z$. We will denote by 1_X, or simply 1 the identity morphism on X.

Coproducts. In this paper, by calling $\mathbf{C}$ "a category with coproducts" we will always mean that $\mathbf{C}$ has *selected binary coproducts*, i.e. that a bi-functor $\oplus\colon \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ exists such that $X \oplus Y$ is a coproduct of X and Y. In such a category, we write $\mathsf{in}_0\colon X \to X \oplus Y$ and $\mathsf{in}_1\colon Y \to X \oplus Y$ for the left and right coproduct injections correspondingly. We will occasionally condense $\mathsf{in}_\mathsf{i} \,;\, \mathsf{in}_\mathsf{j}$ to $\mathsf{in}_{\mathsf{ij}}$ for the sake of succinctness.

Monads. A monad $\mathbf{T}$ on $\mathbf{C}$ is determined by a *Kleisli triple* $(T, \eta, (-)^\sharp)$, consisting of a map $T\colon |\mathbf{C}| \to |\mathbf{C}|$, a family of morphisms $(\eta_X\colon X \to TX)_{X\in|\mathbf{C}|}$ and *Kleisli lifting* sending each $f\colon X \to TY$ to $f^\sharp\colon TX \to TY$ and obeying *monad laws*:

$$\eta^\sharp = 1, \qquad \eta \,;\, f^\sharp = f, \qquad (g \,;\, f^\sharp)^\sharp = g^\sharp \,;\, f^\sharp.$$

It follows that T extends to a functor, η extends to a natural transformation – *unit*, $\mu = 1^\sharp \colon TTX \to TX$ extends to a natural transformation – *multiplication*, and that (T, η, μ) is a monad in the standard sense [26]. We will generally use blackboard capitals (such as $\mathbb{T}$) to refer to monads and the corresponding Roman letters (such as T) to refer to their functor parts. Morphisms of the form $f \colon X \to TY$ are called *Kleisli morphisms* and form the *Kleisli category* $\mathbf{C}_{\mathbb{T}}$ of $\mathbb{T}$ under *Kleisli composition* $f, g \mapsto f \,;\, g^\sharp$ with identity η. If $\mathbf{C}$ has binary coproducts then so does $\mathbf{C}_{\mathbb{T}}$: the coproduct injections are Kleisli morphisms of the form $\mathsf{in}_0 \,;\, \eta \colon X \to T(X \oplus Y)$, $\mathsf{in}_1 \,;\, \eta \colon Y \to T(X \oplus Y)$.

Coalgebras. Given an endofunctor $F \colon \mathbf{C} \to \mathbf{C}$, a pair $(X \in |\mathbf{C}|, c \colon X \to FX)$ is called an F-coalgebra. Coalgebras form a category under the following notion of morphism: $h \colon X \to X'$ is a morphism from (X, c) to (X', c') if $h \,;\, c' = c \,;\, Fh$. A terminal object in this category is called a *final coalgebra*. We reserve the notation $(\nu F, \mathsf{out})$ for a selected final coalgebra if it exists. A well-known fact (Lambek's lemma) is that out is an isomorphism.

For a coalgebra $(X, c \colon X \to FX)$ on $\mathbf{Set}$ a relation $\mathcal{B} \subseteq X \times X$ is a *(coalgebraic) bisimulation* if it extends to a coalgebra $(\mathcal{B}, b \colon \mathcal{B} \to F\mathcal{B})$, such that the left and the right projections from $\mathcal{B}$ to X are coalgebra morphisms; $x \in X$ and $y \in X$ are *bisimilar* if $x \mathrel{\mathcal{B}} y$ for some bisimulation $\mathcal{B}$; the coalgebra $(X, c \colon X \to FX)$ is *strongly extensional* [38] if bisimilarity entails equality. Final coalgebras are the primary example of strongly extensional coalgebras.

3 Idempotent Grove and Kleene-Kozen Categories

A monoid is precisely a single-object category. Various algebraic structures extending monoids can be generalized to categories along this basic observation (e.g. a group is a single-object groupoid, a quantale is a single-object quantaloid, etc.). In this section, we consider two classes of categories for nondeterminism and Kleene iteration, which demonstrate our principled categorical approach of working with algebraic structures.

▶ **Definition 1** (Idempotent Grove Category, cf. [4, 5])**.** *Let us call a category* $\mathbf{C}$ *an* idempotent grove category *if the hom-sets of* $\mathbf{C}(X, Y)$ *are equipped with the structure* $(0, +)$ *of bounded join-semilattice such that, for all* $p \in \mathbf{C}(Y, Z)$ *and* $q, r \in \mathbf{C}(X, Y)$,

$$0 \,;\, p = 0, \qquad (q + r) \,;\, p = q \,;\, p + r \,;\, p. \tag{3}$$

In such a category, we call a morphism $p \in \mathbf{C}(X, Y)$ linear *if it satisfies, for all* $q, r \in \mathbf{C}(Y, Z)$,

$$p \,;\, 0 = 0, \qquad p \,;\, (q + r) = p \,;\, q + p \,;\, r. \tag{4}$$

An idempotent grove category with coproducts *is an idempotent grove category with selected binary coproducts and with* in_0 *and* in_1 *linear.*

Given $p, q \in \mathbf{C}(X, Y)$, let $p \leqslant q$ if $p + q = q$. This yields a partial order with 0 as the bottom element, and morphism composition is monotone on the left, while linear morphisms are additionally monotone on the right. The class of all linear morphisms of an idempotent grove category thus forms a sub-category enriched in bounded join-semilattices (equivalently: commutative and idempotent monoids) – thus, an idempotent grove category where all morphisms are linear is an enriched category. However, we are interested in categories where not all morphisms are linear. An instructive example is as follows.

▶ **Example 2** (Synchronization Trees)**.** Let A be some non-empty fixed set of *labels*, and let $TX = \nu\gamma.\, \mathcal{P}_{\omega_1}(X \oplus A \times \gamma)$ where $\mathcal{P}_{\omega_1}$ is the countable powerset functor. By generalities [39], T extends to a monad $\mathbf{T}$ on **Set**. The elements of TX can be characterized as *countably-branching strongly extensional synchronization trees with exit labels in* X. Synchronization trees have originally been introduced by Milner [28] as denotations of process algebra terms, and subsequently generalized to infinite branching and to explicit exit labels (e.g. [1]). A generic element $t \in TX$ can be more explicitly represented using the following syntax:

$$t = \sum\nolimits_{i\in I} a_i.\, t_i + \sum\nolimits_{i\in J} x_i$$

where I and J are at most countable, the a_i range over A, the t_i range over TX, and x_i range over X. The involved summation operators $\sum_{i\in I}$ are considered modulo countable versions of associativity, commutativity and idempotence, and $0 = \sum_{i\in\emptyset} t_i$ and $t_1 + t_2 = \sum_{i\in\{1,2\}} t_i$.

Recall that strong extensionality means that bisimilar elements are equal [38]. The Kleisli category of $\mathbf{T}$ is idempotent grove with 0 and $+$ inherited from $\mathcal{P}_{\omega_1}$ and ensuring (3) automatically. It is easy to see that linear morphisms are precisely those that do not involve actions.

A straightforward way to add a Kleene iteration operator to a category is as follows.

▶ **Definition 3** (Kleene-Kozen Category [16])**.** *An idempotent grove category* $\mathbf{C}$ *is a* Kleene-Kozen category *if all morphisms of* $\mathbf{C}$ *are linear and there is a* Kleene iteration *operator* $(-)^*\colon \mathbf{C}(X,X) \to \mathbf{C}(X,X)$ *such that, for any* $p\colon X \to X$, $q\colon Y \to X$ *and* $r\colon X \to Z$*, the morphism* $q\,;p^*$ *is the least (pre-)fixpoint of* $q + (-)\,;p$ *and the morphism* $p^*\,;r$ *is the least (pre-)fixpoint of* $r + p\,;(-)$*.*

It is known [16] that Kleene algebra is precisely a single-object Kleene-Kozen category.

In idempotent grove categories with coproducts, the following property is a direct consequence of linearity of in_0, in_1, and will be used extensively throughout.

▶ **Proposition 4.** *In idempotent grove categories with coproducts,* $[p, q] + [p', q'] = [p + p', q + q']$*.*

4 Decisions and Tests in Category

We proceed to compare two mechanisms for modeling control in categories: *decisions* and *tests*. The first one is inherently categorical, and requires coproducts. The second one needs no coproducts, but requires nondeterminism. The latter one is directly inspired by tests of the Kleene algebra with tests [24]. We will show that tests and decisions are in a suitable sense equivalent, when it comes to modeling control that satisfies Boolean algebra laws.

▶ **Definition 5** (Decisions [10, 16])**.** *In a category* $\mathbf{C}$ *with binary coproducts, we call morphisms from* $\mathbf{C}(X, X \oplus X)$ decisions.

We consider the following operations on decisions, modeling truth values and logical connectives: $\mathsf{tt} = \mathsf{in}_1$ (true), $\mathsf{ff} = \mathsf{in}_0$ (false), $\sim d = d\,;[\mathsf{in}_1, \mathsf{in}_0]$ (negation), $d \mathbin{||} e = d\,;[e, \mathsf{in}_1]$ (disjunction), $d \mathbin{\&\&} e = d\,;[\mathsf{in}_0, e]$ (conjunction). Even without constraining decisions in any way, certain logical properties can be established, e.g. (not necessarily commutative or idempotent) monoidal structures $(\mathsf{ff}, ||)$, $(\mathsf{tt}, \&\&)$, involutivity of $\sim$, "de Morgan laws" $\sim(d \mathbin{||} e) = \sim d \mathbin{\&\&} \sim e$, $\sim(d \mathbin{\&\&} e) = \sim d \mathbin{||} \sim e$, and the laws $\mathsf{tt} \mathbin{||} d = \mathsf{tt}$, $\mathsf{ff} \mathbin{\&\&} d = \mathsf{ff}$.

Given $d \in \mathbf{C}(X, X \oplus X)$ and $p, q \in \mathbf{C}(X, Y)$, let

$$\underline{\mathsf{if}}\; d\; \underline{\mathsf{then}}\; p\; \underline{\mathsf{else}}\; q = d\,;[q, p]. \tag{5}$$

▶ **Definition 6** (Tests). *Given an idempotent grove category* $\mathbf{C}$*, we call a family of linear morphisms* $\mathbf{C}^? = (\mathbf{C}^?(X) \subseteq \mathbf{C}(X,X))_{X\in|\mathbf{C}|}$ tests *if every* $\mathbf{C}^?(X)$ *forms a Boolean algebra under* ; *as conjunction and* $+$ *as disjunction.*

It follows that $1 \in \mathbf{C}^?(X)$ and $0 \in \mathbf{C}^?(X)$ correspondingly are the top and bottom elements of $\mathbf{C}^?(X)$. Given $b \in \mathbf{C}^?(X)$, $p, q \in \mathbf{C}(X,Y)$, let

$$\mathsf{if}\, b\, \mathsf{then}\, p\, \mathsf{else}\, q = b\,;p + \bar{b}\,;q. \tag{6}$$

In an idempotent grove category $\mathbf{C}$ with coproducts and tests $\mathbf{C}^?$, let $?\colon \mathbf{C}(X, X \oplus X) \to \mathbf{C}^?(X)$ be the morphism $d? = d\,;[0,1]$.

▶ **Proposition 7.** *Let* $\mathbf{C}$ *be an idempotent grove category with coproducts. If a decision* d *is linear, then, for all* p *and* q*, we have* $\underline{\mathsf{if}}\, d\, \underline{\mathsf{then}}\, p\, \underline{\mathsf{else}}\, q = \mathsf{if}\, d?\, \mathsf{then}\, p\, \mathsf{else}\, q$*.*

Let us say that a pair $(b,c) \in \mathbf{C}(X,X) \times \mathbf{C}(X,X)$ satisfies *(the law of) contradiction* if $b\,;c = 0$, and that it satisfies *(the law of) excluded middle* if $b + c = 1$. The following characterization is instructive.

▶ **Proposition 8.** *Given an idempotent grove category* $\mathbf{C}$*, a family of linear morphisms* $\mathbf{C}^? = (\mathbf{C}^?(X) \subseteq \mathbf{C}(X,X))_{X\in|\mathbf{C}|}$ *forms tests for* $\mathbf{C}$ *iff, for every* $b \in \mathbf{C}^?$*, there is* $\bar{b} \in \mathbf{C}^?$ *such that* $(b, \bar{b})$ *satisfies contradiction and excluded middle.*

Note that the smallest choice of tests in $\mathbf{C}$ is $\mathbf{C}^?(X) = \{0,1\}$. We proceed to characterize the smallest possible choice of tests, sufficient for modeling control.

▶ **Definition 9** (Expressive Tests). *We call the tests* $\mathbf{C}^?$ expressive *if every* $\mathbf{C}^?(X)$ *contains* $[\mathsf{in}_0, 0]$ *whenever* $X = X_1 \oplus X_2$*.*

▶ **Lemma 10.** *The smallest expressive family of tests always exists and is obtained by closing tests of the form* 0*,* 1*,* $[\mathsf{in}_0, 0]$*, and* $[0, \mathsf{in}_1]$ *under* $+$ *and* ;*.*

In the sequel, we will use the notation $\top$, $\bot$, $\wedge$, $\vee$ for tests, synonymously to 1, 0, ;, $+$ to emphasize their logical character.

▶ **Lemma 11.** *Let* $\mathbf{C}^?$ *be tests in an idempotent grove category* $\mathbf{C}$ *with binary coproducts.*
1. *The morphisms* $\diamond : \mathbf{C}^?(X) \to \mathbf{C}(X, X \oplus X)$*,* $?\colon \mathbf{C}(X, X \oplus X) \to \mathbf{C}^?(X)$ *defined by* $\diamond\, b = \bar{b}\,;\mathsf{in}_0 + b\,;\mathsf{in}_1$*,* $d? = d\,;[0,1]$ *form a retraction.*
2. *Every morphism* d *in the image of* $\diamond$ *is linear. Moreover, we have* $d\,;\nabla = 1$*,* $d = d \mathbin{\&\&} d$*, and* $d = d \mathbin{||} d$*.*
3. *For all* e *and* d *in the image of* $\diamond$*, it holds that* $(e \mathbin{||} d)? = e? \vee d?$*,* $(e \mathbin{\&\&} d)? = e? \wedge d?$*, and* $({\sim} d)? = \overline{d?}$*.*

Lemma 11 indicates that in presence of coproducts and with linear coproduct injections, instead of Boolean algebras on subsets of $\mathbf{C}(X,X)$, one can equivalently work with Boolean algebras on subsets of $\mathbf{C}(X, X \oplus X)$.

We conclude this section by an illustration that varying tests, in particular, going beyond smallest expressive tests is practically advantageous.

▶ **Example 12.** Consider the *nondeterministic state monad* $\mathbf{T}$ with $TX = \mathcal{P}(S \times X)^S$ on $\mathbf{Set}$, where S is a fixed *global store*, which the programs, represented by Kleisli morphisms of $\mathbf{T}$ are allowed to read and modify. Morphisms of the Kleisli category $\mathbf{Set}_{\mathbf{T}}$ are equivalently (by uncurrying) maps of the form $p\colon S \times X \to \mathcal{P}(S \times Y)$, meaning that $\mathbf{Set}_{\mathbf{T}}$ is equivalent to a full subcategory of $\mathbf{Set}_{\mathcal{P}}$, from which $\mathbf{Set}_{\mathbf{T}}$ inherits the structure of an idempotent grove

$$\mathsf{in}_0 ; [p,q] = p \qquad \mathsf{in}_1 ; [p,q] = q \qquad [\mathsf{in}_0, \mathsf{in}_1] = 1 \qquad [p,q] ; r = [p ; r, q ; r]$$

$$0 + p = p \qquad p + p = p \qquad p + q = q + p \qquad (p+q) + r = p + (q + r)$$

$$0 ; p = 0 \qquad (q + r) ; p = q ; p + r ; p \qquad u ; 0 = 0 \qquad u ; (p + q) = u ; p + u ; q$$

$$(*\textsf{-Fix})\ \ p^* = 1 + p ; p^* \qquad (*\textsf{-Sum})\ \ (p+q)^* = p^* ; (q ; p^*)^* \qquad (*\textsf{-Uni})\ \ \frac{u ; p = q ; u}{u ; p^* = q^* ; u}$$

Figure 1 Axioms of KiCs, including binary coproducts (p, q, r range over $\mathbf{C}$, u ranges over $\bar{\mathbf{C}}$).

category. The tests identified in Lemma 10 are those maps $b\colon S \times X \to \mathcal{P}(S \times X)$ that are determined by decompositions $X = X_1 \oplus X_2$, in particular, they can neither read nor modify the store. In practice, only the second is regarded as undesirable (and indeed would break commutativity of tests), while reading is typically allowed. This leads to a more permissive notion of tests, as those that are determined by the decompositions $S \times X = X_1 \oplus X_2$.

5 Kleene Iteration, Categorically

We now can introduce our central definition by extending idempotent grove categories with a selected class of linear morphisms, called *tame morphisms*, and with Kleene iteration. Crucially, we assume the ambient category $\mathbf{C}$ to have coproducts as a necessary ingredient. Finding a general definition, not relying on coproducts, presently remains open.

▶ **Definition 13** (KiC(T)). *We call a tuple* $(\mathbf{C}, \bar{\mathbf{C}})$ *a* Kleene-iteration category (KiC) *if*

1. $\mathbf{C}$ *is an idempotent grove category with coproducts;*
2. $\bar{\mathbf{C}}$ *is a wide subcategory of* $\mathbf{C}$*, whose morphisms we call* tame *such that*
   - $\bar{\mathbf{C}}$ *has coproducts strictly preserved by the inclusion to* $\mathbf{C}$*;*
   - *the morphisms of* $\bar{\mathbf{C}}$ *are all linear;*
3. *for every* $X \in |\mathbf{C}|$*, there is a* Kleene iteration *operator* $(-)^*\colon \mathbf{C}(X,X) \to \mathbf{C}(X,X)$ *such that the laws* $*$-**Fix**, $*$-**Sum** *and* $*$-**Uni** *in Figure 1, with* u *ranging over* $\bar{\mathbf{C}}$*, are satisfied.*

A functor $F\colon (\mathbf{C}, \bar{\mathbf{C}}) \to (\mathbf{D}, \bar{\mathbf{D}})$ *between KiCs is a coproduct preserving functor* $F\colon \mathbf{C} \to \mathbf{D}$ *such that* $F0 = 0$*,* $F(q+r) = Fq + Fr$ *and* $Fp^* = (Fp)^*$ *for all* $q, r \in \mathbf{C}(X,Y)$*,* $p \in \mathbf{C}(X,X)$*, and* $Fp \in \bar{\mathbf{D}}(FX, FY)$ *for all* $p \in \bar{\mathbf{C}}(X,Y)$*.*

A KiC $(\mathbf{C}, \bar{\mathbf{C}})$ *equipped with a choice of tests* $\mathbf{C}^?$ *in* $\bar{\mathbf{C}}$ *we call a* KiCT *(=KiC with tests). Correspondingly, functors between KiCTs are additionally required to send tests to tests.*

It transpires from the definition that the role of tameness is to limit the power of the uniformity rule $*$-**Uni**. The principal case for $\mathbf{C} \neq \bar{\mathbf{C}}$ is Example 2. As we see later (Example 24), this yields a KiC. More generally, unless we restrict $\bar{\mathbf{C}}$ to programs that satisfy the linearity laws (4), the uniformity principle $*$-**Uni** would tend to be unsound. Very roughly, uniformity is some infinitary form of distributivity from the left and it fails for programs that fail the standard left distributivity. This phenomenon is expected to occur for other flavors of concurrent semantics: as long as $\mathbf{C}$ admits morphisms that fail (4), $\bar{\mathbf{C}}$ would have to be properly smaller than $\mathbf{C}$. Apart from concurrency, if $\bar{\mathbf{C}}$ models a language with exceptions, those must be excluded from $\bar{\mathbf{C}}$, for otherwise uniformity would again become unsound.

If we demand all morphisms to be tame, we will obtain a notion very close to that of Kleene-Kozen category (Definition 3).

▶ **Definition 14** ($*$-Idempotence)**.** *A KiC is* $*$-idempotent *if it satisfies* (2)*.*

▶ **Proposition 15.** *A category* $\mathbf{C}$ *is Kleene-Kozen iff* $(\mathbf{C}, \mathbf{C})$ *is a* $*$*-idempotent KiC.*

Proof. As shown previously [16], $\mathbf{C}$ is a Kleene-Kozen category iff
1. $\mathbf{C}$ is enriched over bounded join-semilattices and strict join-preserving morphisms;
2. there is an operator $(-)^* \colon \mathbf{C}(X, X) \to \mathbf{C}(X, X)$ such that
   a. $p^* = 1 + p \,;\, p^*$;
   b. $1^* = 1$;
   c. $p^* = (p + 1)^*$;
   d. $u \,;\, p = q \,;\, u$ implies $u \,;\, p^* = q^* \,;\, u$.

This yields sufficiency by noting that (1) states precisely that all morphisms in $\mathbf{C}$ are linear. To show necessity, it suffices to obtain (2.c) from the assumptions that $(\mathbf{C}, \mathbf{C})$ is a $*$-idempotent KiC and that all morphisms in $\mathbf{C}$ are linear. Indeed, we have $(p+1)^* = 1^* ; (p; 1^*)^* = p^*$. ◀

KiCs thus deviate from Kleene algebras precisely in four respects:
1. by generalizing from monoids to categories,
2. by allowing non-linear morphisms,
3. by dropping $*$-idempotence, and
4. by requiring binary coproducts.

▶ **Example 16.** The axiom $*$-**Sum**, included in Definition 13, is one of the classical *Conway identities*. The other one $(p; q)^* = 1+p; (q; p)^* ; q$ is derivable if $\mathbf{C} = \bar{\mathbf{C}}$, e.g. in Kleene-Kozen categories. Indeed, $q \,;\, p \,;\, q = q \,;\, p \,;\, q$ entails $q \,;\, (p \,;\, q)^* = (q \,;\, p)^* \,;\, q$ by $*$-**Uni**, and using $*$-**Fix**, $(p \,;\, q)^* = 1 + p \,;\, q \,;\, (p \,;\, q)^* = 1 + p \,;\, (q \,;\, p)^* \,;\, q$.

Clearly, this argument remains valid with only q being tame, but otherwise the requisite identity is not provable.

It may not be obvious why the requirement to support binary coproducts is part of Definition 13, given that the axioms of iteration do not involve them. The reason is that certain identities that also do not involve coproducts are only derivable in their presence.

▶ **Example 17.** The identity $p^* = (p \,;\, (1 + p))^*$ holds in any KiC.

A standard way to instantiate Definition 13 is to start with a category $\mathbf{V}$ with coproducts, and a monad $\mathbf{T}$ on it, and take $\mathbf{C} = \mathbf{V}_{\mathbf{T}}$, $\bar{\mathbf{C}} = \mathbf{V}$ or, possibly, $\bar{\mathbf{C}} = \mathbf{V}_{\mathbf{T}}$. The monad must support nondeterminism and Kleene iteration so that the axioms of KiC are satisfied. Consider a class of Kleene-Kozen categories that arise in this way.

▶ **Example 18.** Let Q be a unital quantale, and let $TX = Q^X$ for every set X. Then T extends to a monad on **Set** as follows: $\eta(x)(x) = 1$, $\eta(x)(y) = \bot$ if $x \neq y$, and

$$(p \colon X \to Q^Y)^\sharp(f \colon X \to Q)(y \in Y) = \bigvee\nolimits_{x \in X} p(x)(y) \cdot f(x).$$

We obtain a Kleene-Kozen structure in $\mathbf{Set}_{\mathbf{T}}$ as follows:
- $0 \colon X \to Q^Y$ sends x to $\lambda y.\, \bot$;
- $p + q \colon X \to Q^Y$ sends x to $\lambda y.\, p(x)(y) \vee q(x)(y)$;
- $p^* \colon X \to Q^X$ is the least fixpoint of the map $q \mapsto 1 + q \,;\, p$.

This construction restricts to $Q^X_{\omega_1} = \{f \colon X \to Q \mid |\mathsf{supp}\, f| \leqslant \omega\}$ where $\mathsf{supp}\, f$ is the set of those $x \in X$, for which $f(x) \neq 0$. Thus, e.g. the Kleisli categories of the powerset monad $\mathcal{P}$ and the countable powerset monad $\mathcal{P}_{\omega_1}$ are Kleene-Kozen.

For a contrast, consider a similar construction that yields a KiC, which is not Kleene-Kozen.

▶ **Example 19.** Let $Q = \{0, 1, \infty\}$ be the complete lattice under the ordering $0 < 1 < \infty$, and let us define commutative binary multiplication as follows: $0 \cdot x = 0$, $1 \cdot x = x$ and $\infty \cdot \infty = \infty$. This turns Q into a unital quantale, hence an idempotent semiring, whose binary summation $+$ is binary join. Next, define infinite summation with the formula

$$\sum_{i \in I} x_i = \begin{cases} \bigvee_{i \in I'} x_i, & \text{if } I' = \{i \in I \mid x_i > 0\} \text{ is finite} \\ \infty, & \text{otherwise} \end{cases}$$

This makes Q into a *complete semiring* [11]. Let us define the monad $\mathcal{R}$ and the idempotent grove structure on $\mathbf{Set}_{\mathcal{R}}$ like $Q^{(-)}_{\omega_1}$ in Example 18 (with $\sum$ instead of $\bigvee$). For every $f \colon X \to \mathcal{R}X$, let $p^* = \sum_{n \in \mathbb{N}} p^n \colon X \to \mathcal{R}X$ where, inductively, $p^0 = 1$ and $p^{n+1} = p\,;p^n$, and infinite sums are extended from Q to the Kleisli hom-sets pointwise.

It is easy to verify that $(\mathbf{Set}_{\mathcal{R}}, \mathbf{Set}_{\mathcal{R}})$ is a KiC, but $\mathbf{Set}_{\mathcal{R}}$ is not a Kleene-Kozen category, for $*$-idempotence fails: $\eta^* = \sum_{n \in \mathbb{N}} \eta^n = \sum_{n \in \mathbb{N}} \eta = \lambda x, y.\, \infty \neq \eta$. We will use a more convenient notation for the elements of $\mathcal{R}X$ as infinite formal sums $\sum_{i \in I} x_i$ ($x_i \in X$), modulo associativity, commutativity, idempotence (but without countable idempotence $\sum_{i \in I} x = x$!).

Below, we provide two results for constructing new KiCs from old: Theorem 20 and Theorem 23, which are also used prominently in our characterization result in Section 7.

▶ **Theorem 20.** *Let $(\mathbf{C}, \bar{\mathbf{C}})$ be a KiC and let $\mathbf{T}$ be a monad on $\mathbf{C}$ such that*
1. *$Tr^* = (Tr)^*$, for all $r \in \mathbf{C}(X, X)$;*
2. *the monad $\mathbf{T}$ restricts to a monad on $\bar{\mathbf{C}}$.*

Then $\bar{\mathbf{C}}_{\mathbf{T}}$ is a subcategory of $\bar{\mathbf{C}}$ and $(\mathbf{C}_{\mathbf{T}}, \bar{\mathbf{C}}_{\mathbf{T}})$ is a KiCT where 0 and $+$ are defined as in $\mathbf{C}$, and for any $p \colon X \to TX$, the corresponding Kleene iteration is computed as $\eta\,; (p^\sharp)^*$.

Let us illustrate the use of Theorem 20 by a simple example.

▶ **Example 21** (Finite Traces)**.** Consider the monad $\mathcal{P}(A^\star \times -)$ on $\mathbf{Set}$. Elements of $\mathcal{P}(A^\star \times X)$ are standardly used as (finite) trace semantics of programs. A trace is then a sequence of actions from A, followed by an end result in X. Of course, it can be verified directly that the Kleisli category of $\mathcal{P}(A^\star \times -)$ is Kleene-Kozen. Let us show how this follows from Theorem 20.

The Kleisli category of $\mathcal{P}$ is isomorphic to the category of relations, and is obviously Kleene-Kozen. For $\mathcal{P}$, like for any commutative monad, the Kleisli category $\mathbf{Set}_{\mathcal{P}}$ is symmetric monoidal: $X \otimes Y = X \times Y$ and, given $p \colon X \to \mathcal{P}Y$, $q \colon X' \to \mathcal{P}Y'$,

$$(p \otimes q)(x, x') = \{(y, y') \mid y \in p(x),\ y' \in q(x')\}.$$

The set $A^\star$ is a monoid in $\mathbf{Set}_{\mathcal{P}}$ w.r.t. this monoidal structure. This yields a writer monad $\mathbf{T}$ on $\mathbf{Set}_{\mathcal{P}}$ via $TX = A^\star \otimes X$ and $(Tp)(w, x) = \{(w, y) \mid y \in p(x)\}$. Its Kleisli category $(\mathbf{Set}_{\mathcal{P}})_{\mathbf{T}}$ is isomorphic to our original Kleisli category of interest. The assumptions (1) of Theorem 20 are thus satisfied in the obvious way. The assumption (2) is vacuous, as we chose all morphisms to be tame.

Finally, we establish robustness of KiCs under the *generalized coalgebraic resumption monad transformer* [32, 18], which is defined as follows.

▶ **Definition 22** (Coalgebraic Resumptions)**.** *Let $\mathbf{V}$ be a category with coproducts and let $\mathbf{T}$ be a monad on $\mathbf{V}$. Let $H \colon \mathbf{V} \to \mathbf{V}$ be some endofunctor and assume that all final coalgebras $\nu\gamma.\, T(X \oplus H\gamma)$ exist. The assignment $X \mapsto \nu\gamma.\, T(X \oplus H\gamma)$ yields a monad $\mathbf{T}_H$, called the (generalized) coalgebraic resumption monad transformer of $\mathbf{T}$.*

▶ **Theorem 23.** *Let* $\mathbf{T}_H$ *be as in Definition 22 and such that* $(\mathbf{V_T}, \overline{\mathbf{C}})$ *is a KiC for some choice of* $\overline{\mathbf{C}}$*. Then* $\mathbf{V_T}$ *is a wide subcategory of* $\mathbf{V}_{\mathbf{T}_H}$ *and* $(\mathbf{V}_{\mathbf{T}_H}, \overline{\mathbf{C}})$ *is a KiC w.r.t. the following structure:*

- *the bottom element in every* $\mathbf{V}(X, T_H Y)$ *is* $0 \,;\, \mathsf{out}^{-1}$*, and the join of* $p, q \in \mathbf{V}(X, T_H Y)$ *is* $(p \,;\, \mathsf{out} + q \,;\, \mathsf{out}) \,;\, \mathsf{out}^{-1}$*;*
- *given* $p \in \mathbf{V}(X, T_H X)$*,* $p^* \in \mathbf{V}(X, T_H X)$ *is the unique solution of the equation*

$$p^* \,;\, \mathsf{out} = \mathsf{in}_0 \,;\, [p \,;\, \mathsf{out}, 0]^* \,;\, T(1 \oplus Hp^*).$$

We defer the proof to Section 6 where we use reduction to the existing result [18], using the equivalence of Kleene and Elgot iterations, we establish in Section 6.

▶ **Example 24.** By taking $T = \mathcal{P}_{\omega_1}$ and $H = A \times \text{–}$ in Theorem 23, we obtain $T_H X = \nu\gamma.\, \mathcal{P}_{\omega_1}(X \oplus A \times \gamma)$ from Example 2. Let us illustrate the effect of Kleene iteration by example. Consider the system of equations

$$P = a.\, P + Q, \qquad Q = b.\, Q + P$$

for defining the behaviour of two processes P and Q. This system induces a function $p \colon \{P, Q\} \to T_H\{P, Q\}$, sending P to $a.\, P + Q$ and Q to $b.\, Q + P$. The expression $[p; \mathsf{out}, 0]^*$ calls the iteration operator of the powerset-monad, resulting in the function that sends both P and Q to $a.\, P + b.\, Q + Q + P$. Finally, p^* sends P to P' and Q to Q', where P' and Q' are the synchronization trees, obtained as unique solutions of the system:

$$P' = a.\, P' + b.\, Q' + Q + P, \qquad Q = a.\, P' + b.\, Q' + Q + P.$$

6 Elgot Iteration and While-Loops

In this section, we establish an equivalence between Kleene iteration, in the sense of KiC and *Elgot iteration*, as an operation with the following profile in a category $\mathbf{C}$ with coproducts:

$$(\text{–})^\dagger \colon \mathbf{C}(X, Y \oplus X) \to \mathbf{C}(X, Y). \tag{7}$$

This could be done directly, but we prove an equivalence between Elgot iteration and while-loops first, and then prove the equivalence of the latter and Kleene iteration. In this chain of equivalences, only while-loops need tests, and it will follow that a particular choice of tests is not relevant, once they are expressive. On the other hand, existence of expressive tests is guaranteed by Lemma 10. This explains why tests disappear in the resulting equivalence.

▶ **Definition 25** (Conway Iteration, Uniformity)**.** *An Elgot iteration operator* (7) *in a category* $\mathbf{C}$ *with coproducts is* Conway iteration *[13] if it satisfies the following principles:*

Naturality : $p^\dagger \,;\, q = (p \,;\, (q \oplus 1))^\dagger$ **Dinaturality** : $(p \,;\, [\mathsf{in}_0, q])^\dagger = p \,;\, [1, (q \,;\, [\mathsf{in}_0, p])^\dagger]$

Codiagonal : $(p \,;\, [1, \mathsf{in}_1])^\dagger = p^{\dagger\dagger}$

Moreover, given a subcategory $\mathbf{D}$ *of* $\mathbf{C}$*,* $(\text{–})^\dagger$ *is* uniform *w.r.t.* $\mathbf{D}$*, or* $\mathbf{D}$*-uniform, if it satisfies*

Uniformity : $\dfrac{u \,;\, q = p \,;\, (1 \oplus u)}{u \,;\, q^\dagger = p^\dagger}$ *(with* u *from* $\mathbf{D}$*)*

By taking $q = \mathsf{in}_1$ in **Dinaturality**, we derive

Fixpoint : $p \,;\, [1, p^\dagger] = p^\dagger$.

DW-Fix:	$\underline{\text{while}}\ d\ \underline{\text{do}}\ p = \underline{\text{if}}\,d\,\underline{\text{then}}\,p\,;(\underline{\text{while}}\ d\ \underline{\text{do}}\ p)\,\underline{\text{else}}\ 1$
DW-Or:	$\underline{\text{while}}\ (d \parallel e)\ \underline{\text{do}}\ p = (\underline{\text{while}}\ d\ \underline{\text{do}}\ p)\,;\underline{\text{while}}\ e\ \underline{\text{do}}\ (p\,;\underline{\text{while}}\ d\ \underline{\text{do}}\ p)$
DW-And:	$\underline{\text{while}}\ (d\ \&\&\ (e \parallel \mathsf{tt}))\ \underline{\text{do}}\ p = \underline{\text{while}}\ d\ \underline{\text{do}}\ (\underline{\text{if}}\,e\,\underline{\text{then}}\,p\,\underline{\text{else}}\ p)$
DW-Uni:	$\dfrac{u\,;\underline{\text{if}}\,d\,\underline{\text{then}}\,p\,;\mathsf{tt}\,\underline{\text{else}}\ \mathsf{ff} = \underline{\text{if}}\,e\,\underline{\text{then}}\,q\,;u\,;\mathsf{tt}\,\underline{\text{else}}\ v\,;\mathsf{ff}}{u\,;\underline{\text{while}}\ d\ \underline{\text{do}}\ p = (\underline{\text{while}}\ e\ \underline{\text{do}}\ q)\,;v}$

Figure 2 Uniform Conway iteration in terms of decisions.

▶ **Theorem 26.** *Let $\mathbf{C}$ be a category with coproducts, let $\mathbf{D}$ be its wide subcategory with coproducts, preserved by the inclusion, and let for every $X \in |\mathbf{C}|$, $\mathbf{C}^\diamond(X)$ be a set of decisions such that (i) $\mathsf{in}_0, \mathsf{in}_1 \in \mathbf{C}^\diamond(X)$, (ii) $\mathbf{C}^\diamond(X)$ is closed under* (5)*, (iii) $\mathsf{in}_0 \oplus \mathsf{in}_1 \in \mathbf{C}^\diamond(X)$ if $X = X_1 \oplus X_2$. Then, to give a $\mathbf{D}$-uniform Conway iteration on $\mathbf{C}$ is the same as to give an operator*

$$\frac{d \in \mathbf{C}^\diamond(X) \qquad p \in \mathbf{C}(X,X)}{\underline{\text{while}}\ d\ \underline{\text{do}}\ p \in \mathbf{C}(X,X)}$$

that satisfies the laws in Figure 2 with p, q ranging over $\mathbf{C}$, and with u, v ranging over $\mathbf{D}$.

Theorem 26 yields an equivalence between two styles of iteration: Elgot iteration and while-iteration. We next specialize it to idempotent grove categories with tests using Lemma 11.

Note that in any KiC $(\mathbf{C}, \overline{\mathbf{C}})$ with tests $\mathbf{C}^?$, in addition to the if-then-else (6), we have the while operator, defined in the standard way: given $b \in \mathbf{C}^?(X)$, $p \in \mathbf{C}(X,X)$,

$$\mathsf{while}\,b\,\mathsf{do}\,p = (b\,; p)^*\,; \bar{b}. \tag{8}$$

▶ **Proposition 27.** *Let $\mathbf{C}$ be an idempotent grove category, let $\mathbf{D}$ be a wide subcategory of $\mathbf{C}$ with coproducts, which are preserved by the inclusion to $\mathbf{C}$, and with expressive tests $\mathbf{C}^?$. Then $\mathbf{C}$ supports $\mathbf{D}$-uniform Conway iteration iff it supports a while-operator that satisfies the laws in Figure 3, where b and c come from $\mathbf{C}^?$, p and q come from $\mathbf{C}$ and u, v come from $\mathbf{D}$ and the if-then-else operator is defined as in* (8)*.*

Proof. For every $X \in |\mathbf{C}|$, let $\mathbf{C}^\diamond(X)$ be the image of $\mathbf{C}^?(X)$ under $\diamond$ from Lemma 11. As shown in the lemma, $\mathbf{C}^\diamond(X)$ inherits the Boolean algebra structure from $\mathbf{C}^?(X)$. Using the isomorphism between $\mathbf{C}^\diamond(X)$ and $\mathbf{C}^?(X)$ and Proposition 7, the laws from Figure 2 can be reformulated equivalently, resulting in **TW-Fix**, **TW-Or**, **TW-Uni**, and additionally

$$\mathsf{while}\,(b \wedge (c \vee \top))\,\mathsf{do}\,p = \ \mathsf{while}\,b\,\mathsf{do}\,(\mathsf{if}\,c\,\mathsf{then}\,p\,\mathsf{else}\,p)$$

which however holds trivially. ◀

Thus, in grove categories with expressive tests, Elgot iteration and while-loops are equivalent. We establish a similar equivalence between Kleene iteration and while-loops, which will entail an equivalence between Elgot iteration and Kleene iteration by transitivity.

▶ **Theorem 28.** *Let $\mathbf{C}$, $\overline{\mathbf{C}}$ and $\mathbf{C}^?$ be as follows.*

1. *$\mathbf{C}$ is an idempotent grove category with coproducts.*
2. *$\overline{\mathbf{C}}$ is a wide subcategory of $\mathbf{C}$ with coproducts, consisting of linear morphisms only and such that the inclusion of $\overline{\mathbf{C}}$ to $\mathbf{C}$ preserves coproducts.*
3. *$\mathbf{C}^?$ are expressive tests in $\mathbf{C}$.*

Then $(\mathbf{C}, \overline{\mathbf{C}})$ is a KiCT iff $\mathbf{C}$ supports a while-operator satisfying the laws in Figure 3.

TW-Fix:	$\mathsf{while}\, b\, \mathsf{do}\, p = \mathsf{if}\, b\, \mathsf{then}\, p\,;\,(\mathsf{while}\, b\, \mathsf{do}\, p)\, \mathsf{else}\, 1$
TW-Or:	$\mathsf{while}\,(b \vee c)\,\mathsf{do}\, p = (\mathsf{while}\, b\, \mathsf{do}\, p)\,;\,\mathsf{while}\, c\, \mathsf{do}\,(p\,;\,\mathsf{while}\, b\, \mathsf{do}\, p)$
TW-Uni:	$\dfrac{u\,;\,\bar{b} = \bar{c}\,;\,v \qquad u\,;\,b\,;\,p = c\,;\,q\,;\,u}{u\,;\,\mathsf{while}\, b\, \mathsf{do}\, p = (\mathsf{while}\, c\, \mathsf{do}\, q)\,;\,v}$

Figure 3 Uniform Conway iteration in terms of tests.

We can now characterize KiCs in terms of Elgot iteration.

▶ **Theorem 29.** *Let* $\mathbf{C}$ *be an idempotent grove category with coproducts, and let* $\bar{\mathbf{C}}$ *be a wide subcategory of* $\mathbf{C}$ *with coproducts, consisting of linear morphisms only and such that the inclusion of* $\bar{\mathbf{C}}$ *to* $\mathbf{C}$ *preserves coproducts.*

Then $(\mathbf{C}, \bar{\mathbf{C}})$ *is a KiC iff* $\mathbf{C}$ *supports* $\bar{\mathbf{C}}$*-uniform Conway iteration.*

Proof. Let us define $\mathbf{C}^?$ as in Definition 9. By Theorem 28, $(\mathbf{C}, \bar{\mathbf{C}})$ is a KiCT iff $\mathbf{C}$ supports a while-operator, satisfying the laws in Figure 3. By Proposition 27, the latter is the case iff $\mathbf{C}$ supports $\bar{\mathbf{C}}$-uniform Conway iteration. ◀

Now we can prove Theorem 23.

Proof Theorem 23 (Sketch). We need to check that $(\mathbf{V}_{\mathsf{T}_H}, \bar{\mathbf{C}})$ is a KiC. By Theorem 29, we equivalently prove that $\mathbf{V}_{\mathsf{T}_H}$ supports $\bar{\mathbf{C}}$-uniform Conway iteration. It is already known [18, Lemma 7.2] that if T supports Conway iteration, then so does T_H. By Theorem 29, we are left to check that T_H satisfies **Uniformity**, which is a matter of calculation. ◀

7 Free KiCTs and Completeness

In this section, we characterize a free KiCT with strict coproducts (i.e. those, for which coherence maps $X \oplus (Y \oplus Z) \cong (X \oplus Y) \oplus Z$ are identities) on a one-sorted signature. We achieve this by combining techniques from formal languages [7], category theory and the theory of Elgot iteration with coalgebraic reasoning [34], in particular proofs by coalgebraic bisimilarity. We claim that a more general characterization of a free KiCT on a multi-sorted signature can be achieved along the same lines, modulo a significant notation overhead and the necessity to form final coalgebras in the category of multisorted sets $\mathbf{Set}^{\mathcal{S}}$ where $\mathcal{S}$ is the set of sorts. We dispense with this option for the sake of brevity and readability. Let us fix

- a signatures of n-ary symbols Σ_n for each $n \in \mathbb{N}$, and let $\Sigma = \bigcup_n \Sigma_n$;
- a signature Γ of (unary) symbols, disjoint from Σ;
- a finite (!) signature Θ of (unary) symbols, disjoint from $\Sigma \cup \Gamma$.

Let $\widehat{\Theta}$ denote the set of finite subsets of Θ. We regard Θ as a signature for tests, Γ as a signature for tame morphisms and Σ as a signature for general morphisms; $\widehat{\Theta}$ is meant to capture finite conjunctions of the form $\mathsf{b}_1 \wedge \ldots \wedge \mathsf{b}_n \wedge \bar{\mathsf{b}}_{n+1} \wedge \ldots \wedge \bar{\mathsf{b}}_m$ as semantic correspondents of subsets $\{\mathsf{b}_1, \ldots, \mathsf{b}_n\} \in \widehat{\Theta}$, assuming an enumeration $\Theta = \{\mathsf{b}_1, \ldots, \mathsf{b}_m\}$. This is inspired by Kleene algebra with tests [24]. Furthermore, we accommodate *guarded strings* from *op. cit.*: let Γ^Θ be the set of strings $\langle b_1, \mathsf{u}_1, \ldots, b_n, \mathsf{u}_n, b_{n+1}\rangle$ with $\mathsf{u}_i \in \Gamma$, $b_i \in \widehat{\Theta}$.

7.1 Interpretations

An *interpretation* $[\![-]\!]$ of (Σ, Γ, Θ) over a KiCT $(\mathbf{C}, \bar{\mathbf{C}}, \mathbf{C}^?)$ is specified as follows:

$$[\![1]\!] \in |\mathbf{C}| \qquad\qquad [\![\mathsf{f}]\!] \in \mathbf{C}([\![1]\!], [\![n]\!]) \qquad (\mathsf{f} \in \Sigma_n)$$

$$[\![\mathsf{u}]\!] \in \bar{\mathbf{C}}([\![1]\!], [\![1]\!]) \qquad (\mathsf{u} \in \Gamma) \qquad\qquad [\![\mathsf{b}]\!] \in \mathbf{C}^?([\![1]\!], [\![1]\!]) \qquad (\mathsf{b} \in \Theta)$$

where $[\![n]\!]$ abbreviates the n-fold sum $[\![1]\!] \oplus \ldots \oplus [\![1]\!]$. The latter immediately extends to $\widehat{\Theta}$: $[\![\{\,\}]\!] = 1$, $[\![\{\mathsf{b}_1, \ldots, \mathsf{b}_n\}]\!] = \mathsf{b}_1 \,;\, \ldots \,;\, \mathsf{b}_n \,;\, \bar{\mathsf{b}}_{n+1} \,;\, \ldots \,;\, \bar{\mathsf{b}}_m$, assuming that $\Theta = \{\mathsf{b}_1, \ldots, \mathsf{b}_m\}$. Note that we interpret n-ary symbols over $\mathbf{C}([\![1]\!], [\![n]\!]) = \mathbf{C}^{\mathsf{op}}([\![1]\!]^n, [\![1]\!])$. This equation seems to suggest that it could be more natural to use categories with products as models, rather than categories with coproducts. Our present choice helps us to treat generic KiCTs on the same footing with the free KiCT, which is defined in terms of coproducts and not products.

▶ **Definition 30** (Free KiCT). *A* free KiCT *w.r.t.* (Σ, Γ, Θ) *is a KiCT* $(\mathfrak{F}_{\Sigma,\Gamma,\Theta}, \overline{\mathfrak{F}_{\Sigma,\Gamma,\Theta}}, \mathfrak{F}^?_{\Sigma,\Gamma,\Theta})$ *together with an interpretation of* (Σ, Γ, Θ) *in* $\mathfrak{F}_{\Sigma,\Gamma,\Theta}$*, such that for any other interpretation of* (Σ, Γ, Θ) *over a KiCT* $(\mathbf{C}, \bar{\mathbf{C}}, \mathbf{C}^?)$*, there is unique compatible functor from* $\mathfrak{F}_{\Sigma,\Gamma,\Theta}$ *to* $\mathbf{C}$*. More formally, for any interpretation* $[\![-]\!]$*, there is unique KiCT-functor* $[\![-]\!]_\uparrow \colon (\mathfrak{F}_{\Sigma,\Gamma,\Theta}, \overline{\mathfrak{F}_{\Sigma,\Gamma,\Theta}}, \mathfrak{F}^?_{\Sigma,\Gamma,\Theta}) \to (\mathbf{C}, \bar{\mathbf{C}}, \mathbf{C}^?)$ *such that the diagram*

$$\begin{array}{ccc} \mathfrak{F}_{\Sigma,\Gamma,\Theta} & \xrightarrow{[\![-]\!]_\uparrow} & \mathbf{C} \\ {\scriptstyle [\![-]\!]^{\mathfrak{F}}}\big\uparrow & \nearrow_{[\![-]\!]} & \\ (\Sigma, \Gamma, \Theta) & & \end{array} \tag{9}$$

commutes.

In what follows, we characterize $\mathfrak{F}_{\Sigma,\Gamma,\Theta}$ as a certain category of rational trees, i.e. trees with finitely many distinct subtrees. An alternative, equivalent formulation would be to view $\mathfrak{F}_{\Sigma,\Gamma,\Theta}$ as a free model of the (Lawvere) theory of KiCTs.

Like in the case of original Kozen's completeness result [23], a characterization of the free model immediately entails completeness of the corresponding axiomatization over it. Indeed, by generalities, a free KiCT is isomorphic to the free algebra of terms, quotiented by the provable equality relation. Hence, if an equality holds over the free model, it is provable.

7.2 A KiCT of Coalgebraic Resumptions

For any set X, define $TX = \mathcal{R}(\Gamma^\Theta \times X)$ and $T_\nu X = \nu\gamma.\, T(X \oplus \Sigma\gamma)$, in the category of sets $\mathbf{Set}$ where $\mathcal{R}$ is the monad from Example 19.

A stepping stone for constructing $\mathfrak{F}_{\Sigma,\Gamma,\Theta}$ is the observation that $(\mathbf{Set}_{\mathsf{T}_\nu}, \mathbf{Set}_{\mathsf{T}})$ forms a KiC. Indeed, $(\mathbf{Set}_{\mathcal{R}}, \mathbf{Set}_{\mathcal{R}})$ is a KiC and the monad $\mathcal{R}$ is commutative, hence symmetric monoidal. In $\mathbf{Set}_{\mathcal{R}}$, Γ^Θ is a monoid under the following operations:

$$\langle w_1, \ldots, w_{n+1}\rangle \cdot \langle u_1, \ldots, u_{m+1}\rangle = \begin{cases} \langle w_1, \ldots, w_n, u_2, \ldots, u_{m+1}\rangle & \text{if } w_n = u_1 \\ 0 & \text{otherwise} \end{cases}$$

This produces the monad T, whose Kleisli category is a KiC by Theorem 20, analogously to Example 21. Now, $(\mathbf{Set}_{\mathsf{T}_\nu}, \mathbf{Set}_{\mathsf{T}})$ is a KiC by Theorem 23. We will use the following representation for generic elements of $T_\nu X$:

$$t = \sum\nolimits_{i \in I} b_i.\, \mathsf{u}_i.\, t_i + \sum\nolimits_{i \in J} b_i.\, \mathsf{f}_i(t_{i,1}, \ldots, t_{i,n_i}) + \sum\nolimits_{i \in K} b_i.\, x_i \tag{10}$$

where I, J, K are mutually disjoint countable sets, b_i range over $\widehat{\Theta}$, u_i range over Γ, f_i range over Σ, $t_i, t_{i,j}$ range over $T_\nu X$ and x_i range over X.

▶ **Definition 31** (Derivatives). *For every $t \in T_\nu X$, as in* (10)*, define the following* derivative *operations:*

- $\partial_{b,\mathsf{u}}(t) = \sum_{i\in I, b=b_i, \mathsf{u}=\mathsf{u}_i} t_i$*, for* $b \in \widehat{\Theta}$, $\mathsf{u} \in \Gamma$*;*
- $\partial^k_{b,\mathsf{f}}(t) = \sum_{i\in J, b=b_i, \mathsf{f}=\mathsf{f}_i} t_{i,k}$*, for* $b \in \widehat{\Theta}$, $k \in \{1, \ldots, n_i\}$, $\mathsf{f} \in \Sigma_{n_i}$ *with* $n_i > 0$.

Additionally, let $o(t) = \sum_{i\in K} b_i . x_i$*. We extend these operations to arbitrary morphisms* $Y \to T_\nu X$ *pointwise. The* set of derivatives *of* $t \in T_\nu X$ *is the smallest set* $\mathfrak{D}(t)$ *that contains* t *and is closed under all* $\partial_{b,\mathsf{u}}$ *and* $\partial^k_{b,\mathsf{f}}$.

The following property is a direct consequence of these definitions:

▶ **Lemma 32.** *Let* $t \in T_\nu X$ *be as in* (10)*, and let* $s\colon X \to T_\nu Y$*. Then*

$$\partial_{b,\mathsf{u}}(t \,;\, s^\sharp) = \partial_{b,\mathsf{u}}(t) \,;\, s^\sharp + o(t) \,;\, (\partial_{b,\mathsf{u}}(s))^\sharp \qquad\qquad o(t \,;\, s^\sharp) = o(t) \,;\, (o(s))^\sharp$$
$$\partial^k_{b,\mathsf{f}}(t \,;\, s^\sharp) = \partial^k_{b,\mathsf{f}}(t) \,;\, s^\sharp + o(t) \,;\, (\partial^k_{b,\mathsf{f}}(s))^\sharp$$

▶ **Lemma 33.** *Given a set* X*, let* $\mathcal{B} \subseteq T_\nu X \times T_\nu X$ *be such a relation that whenever* $t \,\mathcal{B}\, s$*,*

1. $\partial_{b,\mathsf{u}}(t) \,\mathcal{B}\, \partial_{b,\mathsf{u}}(s)$ *for all* $b \in \widehat{\Theta}$, $\mathsf{u} \in \Gamma$*,*
2. $\partial^k_{b,\mathsf{f}}(t) \,\mathcal{B}\, \partial^k_{b,\mathsf{f}}(s)$ *for all* $b \in \widehat{\Theta}$, $\mathsf{f} \in \Sigma$*,*
3. $o(t) = o(s)$.

Then, $t = s$ *whenever* $t \,\mathcal{B}\, s$*.*

Proof Sketch. It suffices to show that $\mathcal{B}$ is a coalgebraic bisimulation. The claim is then a consequence of strong extensionality of the final coalgebra $T_\nu X$. Let us spell out what it means for $\mathcal{B}$ to be a coalgebraic bisimulation. Given t and t', such that $t \,\mathcal{B}\, t'$, and assuming representations

$$t = \sum\nolimits_{i\in I} g_i . \mathsf{f}_i(t_{i,1}, \ldots, t_{i,n_i}) + \sum\nolimits_{i\in J} g_i . x_i,$$
$$t' = \sum\nolimits_{i\in I'} g_i . \mathsf{f}_i(t_{i,1}, \ldots, t_{i,n_i}) + \sum\nolimits_{i\in J'} g_i . x_i$$

where the g_i range over Γ^Θ, the sums $\sum_{i\in J} g_i . x_i$ and $\sum_{i\in J'} g_i . x_i$ must be equal, and there must exist a set K and surjections $e\colon K \to I$, $e'\colon K \to I'$, such that for every $k \in K$, $g_{e(k)} = g_{e'(k)}$, $\mathsf{f}_{e(k)} = \mathsf{f}_{e'(k)}$ and $t_{e(k),1} \,\mathcal{B}\, t_{e'(k),1}, \ldots, t_{e(k),m} \,\mathcal{B}\, t_{e'(k),m}$ where m is the arity of $\mathsf{f}_{e(k)}$. This is indeed true for $\mathcal{B}$. The reason for it is that t and t' can be represented as

$$t = \sum\nolimits_{n\in\mathbb{N}} \sum\nolimits_{i\in I, |g_i|=n} g_i . \mathsf{f}_i(t_{i,1}, \ldots, t_{i,n_i}) + \sum\nolimits_{n\in\mathbb{N}} \sum\nolimits_{i\in J, |g_i|=n} g_i . x_i,$$
$$t' = \sum\nolimits_{n\in\mathbb{N}} \sum\nolimits_{i\in I', |g_i|=n} g_i . \mathsf{f}_i(t'_{i,1}, \ldots, t'_{i,n_i}) + \sum\nolimits_{n\in\mathbb{N}} \sum\nolimits_{i\in J', |g_i|=n} g_i . x_i$$

and we can derive the requisite properties for inner sums by induction on n from the assumptions. ◀

7.3 Rational Trees

In what follows, we identify every $n \in \mathbb{N}$ with the set $\{0, \ldots, n-1\}$, and select binary coproducts in **Set** so that $n \oplus m = \{0, \ldots, n-1\} \oplus \{0, \ldots, m-1\} = \{0, \ldots, n+m-1\} = n+m$. The inclusion of n to $m \geqslant n$ is then a coproduct injection, which we refer to as in^m_n.

▶ **Definition 34** (Prefinite, Flat, (Non-)Guarded, Rational, Definable)**.**

1. *The set of* prefinite *elements of $T_\nu X$ is defined by induction: $t \in T_\nu X$ of the form* (10) *is* prefinite *if the involved sums contain finitely many distinct elements and all the $t_i, t_{i,j} \in T_\nu X$ are prefinite.*
2. *A prefinite $t \in T_\nu X$ of the form* (10) *is* flat *if $t_i, t_{i,j} \in X$.*
3. *An element $t \in T_\nu X$ of the form* (10) *is* guarded *if $K = \emptyset$.*
4. *An element $t \in T_\nu X$ of the form* (10) *is* non-guarded *if $I \cup J = \emptyset$.*
5. *An element $t \in T_\nu X$ is* rational *if $\mathfrak{D}(t)$ is finite and t depends on a finite subset of $\Sigma \cup \Gamma$.*

A map $t\colon Y \to T_\nu X$ is prefinite/flat/guarded/non-guarded if correspondingly for every $x \in X$, every $t(x)$ is prefinite/flat/guarded/non-guarded. Finally:

6. *A map $t\colon k \to T_\nu n$ (with $k, n \in \mathbb{N}$) is* definable *if for some $m \geqslant k$ there is flat guarded $s\colon m \to T_\nu m$ and non-guarded $r\colon m \to T_\nu n$, such that $t = \mathsf{in}_k^m \,;\, s^* \,;\, r^\sharp$.*

Using Lemma 32, one can show

▶ **Lemma 35.** *Sum, composition and Kleene iteration of rational maps are again rational.*

The following property is a form of Kleene theorem, originally stating the equivalence of regular and recognizable languages [35]. In our setting it is proven with the help of Lemma 33.

▶ **Proposition 36.** *Given $n, k \in \mathbb{N}$, a map $n \to T_\nu k$ is rational iff it is definable.*

Let $\mathfrak{F} = \mathfrak{F}_{\Sigma,\Gamma,\Theta}$ be the (non-full) subcategory of $\mathbf{Set}_{\mathbf{T}_\nu}$, identified as follows:

- the objects of $\mathfrak{F}$ are positive natural numbers,
- the morphisms in $\mathfrak{F}(n, k)$ are rational maps $f\colon n \to T_\nu k$ (equivalently: (co)tuples $[t_0, \dots, t_{n-1}]$ of rational elements of $T_\nu k$).

Let the wide subcategory of tame morphisms $\overline{\mathfrak{F}}$ consist of such tuples $[t_0, \dots, t_{n-1}]$ that $t_i \in Tk$ for all i, and let $\mathfrak{F}^?$ consist of those maps in $\overline{\mathfrak{F}}$ that do not involve symbols from Γ. This defines a KiCT essentially due to the closure properties from Lemma 35.

Given an interpretation $[\![-]\!]\colon (\Sigma, \Gamma, \Theta) \to \mathbf{C}$, let us extend it to flat elements first via

$$[\![0]\!] = 0, \quad [\![t + s]\!] = [\![t]\!] + [\![s]\!], \quad [\![\eta^*]\!] = [\![1]\!]^*, \quad [\![t \,;\, s^\sharp]\!] = [\![t]\!] \,;\, [\![s]\!], \quad [\![k]\!] = \mathsf{in}_k \ (k \in n)$$

where η^* stands for a tuple of infinite sum $[\sum_{i\in\mathbb{N}}\{\}.0, \dots, \sum_{i\in\mathbb{N}}\{\}.(n-1)]$, and is the interpretation of $\eta^\star$ in $\mathfrak{F}$. The clause for η^* in necessary to cater for infinite sums that can occur in prefinite elements. Such sums can only contain finitely many distinct elements, and thus can be expressed via finite sums and composition with η^*. Next, define $[\![-]\!]_\uparrow\colon \mathfrak{F} \to \mathbf{C}$

- on objects via $[\![n]\!]_\uparrow = [\![1]\!] \oplus \ldots \oplus [\![1]\!]$ ($[\![1]\!]$ repeated n times),
- on morphisms, via $[\![t]\!]_\uparrow = \mathsf{in}_n^m \,;\, [\![s]\!]^* \,;\, [\![r]\!]$, where $t = \mathsf{in}_n^m \,;\, s^* \,;\, r^\sharp$, for a guarded flat $s\colon m \to T_\nu m$, and a non-guarded $r\colon m \to T_\nu k$, computed with Proposition 36.

▶ **Theorem 37.** *$\mathfrak{F}_{\Sigma,\Gamma,\Theta}$ is a free KiCT over Σ, Γ, Θ.*

The following property is instrumental for proving this result:

▶ **Lemma 38.** *Let $(\mathbf{C}, \overline{\mathbf{C}})$ and $(\mathbf{D}, \overline{\mathbf{D}})$ be two KiCs, and let F be the following map, acting on objects and on morphisms: $FX \in |\mathbf{D}|$ for every $X \in |\mathbf{C}|$, $Fp \in \mathbf{D}(FX, FY)$ for every $p \in \mathbf{C}(X, Y)$. Suppose that F preserves coproducts, $Fp \in \overline{\mathbf{D}}(FX, FY)$ for all $p \in \overline{\mathbf{C}}(X, Y)$. F is a KiC-functor if the following further preservation properties hold*

$$Fp^* = (Fp)^*, \quad F(p \,;\, \mathsf{in}_0) = Fp \,;\, \mathsf{in}_0, \quad F(p \,;\, \mathsf{in}_1) = Fp \,;\, \mathsf{in}_1, \quad F(p \,;\, [0,1]) = Fp \,;\, [0,1].$$

Let us briefly outline a potential application of Theorem 37 to may-diverge Kleene algebras, which we informally described in the introduction. Let us now define them formally:

▶ **Definition 39** (May-Diverge Kleene Algebra). *A* may-diverge Kleene algebra *is an idempotent semiring* $(S, 0, 1, +, \,;\,)$ *equipped with an iteration operator* $(-)^* : S \to S$ *satisfying the laws:*

$$p^* = 1 + p \,;\, p^* \qquad (p+q)^* = p^* \,;\, (q \,;\, p^*)^* \qquad \frac{r \,;\, p = q \,;\, r}{r \,;\, p^* = q^* \,;\, r}$$

Thus, may-diverge Kleene algebras are very close to KiCs of the form $(\mathbf{C}, \mathbf{C})$ with $|\mathbf{C}| = 1$, except that in our present treatment all KiCs come with binary coproducts as an additional structure. We conjecture though that any may-diverge Kleene algebra, viewed as a category, can be embedded to a KiC $(\mathbf{C}, \mathbf{C})$ with $|\mathbf{C}| = \{1, 2, \ldots\}$. Theorem 37 will then entail a characterization of the free may-diverge Kleene algebra on Γ as the full subcategory induced by the single object 1 of the Kleisli category of the monad $TX = \mathcal{R}(\Gamma^\star \times X)$. In other words, the free may-diverge Kleene algebra is carried (up-to-isomorphism) by rational elements of $\mathcal{R}(\Gamma^\star)$, similarly to that how the free Kleene algebra is carried by rational elements of $\mathcal{P}(\Gamma^\star)$.

8 Conclusions and Further Work

We developed a general and robust categorical notion of Kleene iteration – KiC(T) (=Kleene-iteration category (with tests)) – inspired by Kleene algebra (with tests) and its numerous cousins. We attested this notion with various yardsticks: stability under the generalized coalgebraic resumption monad transformer (hence under the exception transformer, as its degenerate case), equivalence to the classical notion of Conway iteration and to a suitably axiomatized theory of while-loops, but most remarkably, we established an explicit description of the ensuing free model, as a category of certain nondeterministic rational trees, playing the same role for our theory as the algebra of regular events for Kleene algebra. However, in our case, the free model is much more intricate and difficult to construct, as the iteration operator of it is neither a least fixpoint nor a unique fixpoint. A salient feature of our notion, mirrored in the structure of the free model, is that it can mediate between linear time and branching time semantics via corresponding specified classes of morphisms.

Given the abstract nature of our results, we expect them be be reusable for varying and enriching the core notion of Kleene iteration with other features. For example, our underlying notion of nondeterminism is that of idempotent grove category. General grove categories are a natural base for probabilistic or graded semantics, and we expect that most of our results, including completeness can be adapted to this case. Yet more generally, a relevant ingredient of our construction is monad $\mathcal{R}$, currently capturing the effect of nondeterminism, but which can potentially be varied to obtain other flavors of linear behavior.

An important open problem that remains for future work is that of defining KiCTs without coproducts, potentially providing a bridge to relevant algebraic structures as single-object categories. Now that the free KiCT with coproduct is identified, the free KiCT without coproducts is expected to be complete over the same model. Identifying such a notion is hard, because it would simultaneously encompass independent axiomatizations of iterative behavior, e.g. branching time and linear time. As of now, such axiomatizations are built on hard-to-reconcile approaches to iteration as either a least or a unique fixpoint.

References

1 Luca Aceto, Arnaud Carayol, Zoltán Ésik, and Anna Ingólfsdóttir. Algebraic synchronization trees and processes. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Proc. of 39th Int. Coll on Automata, Languages, and Programming, ICALP 2012, Part 2*, volume 7392 of *Lect. Notes in Comput. Sci.*, pages 30–41. Springer, 2012. `doi:10.1007/978-3-642-31585-5_7`.

2 Michael A. Arbib and Ernest G. Manes. Partially additive categories and flow-diagram semantics. *J. Algebra*, 62(1):203–227, 1980. doi:10.1016/0021-8693(80)90212-4.

3 Steve Awodey. *Category Theory*. Oxford University Press, 2nd edition, 2010.

4 David B. Benson and Jerzy Tiuryn. Fixed points in free process algebras, part I. *Theor. Comput. Sci.*, 63(3):275–294, 1989. doi:10.1016/0304-3975(89)90010-8.

5 S.L. Bloom, Z. Esik, and D. Taubner. Iteration theories of synchronization trees. *Information and Computation*, 102(1):1–55, 1993. doi:10.1006/inco.1993.1001.

6 Stephen L. Bloom and Zoltán Ésik. *Iteration Theories: The Equational Logic of Iterative Processes*. Springer, 1993. doi:10.1007/978-3-642-78034-9.

7 Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964. doi:10.1145/321239.321249.

8 Kenta Cho. Total and partial computation in categorical quantum foundations. In Chris Heunen, Peter Selinger, and Jamie Vicary, editors, *Proc. of 12th Int. Workshop on Quantum Physics and Logic, QPL 2015*, volume 195 of *Electron. Proc. in Theor. Comput. Sci.*, pages 116–135. Open Publishing Assoc., 2015. doi:10.4204/eptcs.195.9.

9 Robin Cockett. Itegories & PCAs, 2007. Slides from Fields Institute Meeting on Traces (Ottawa, 2007). URL: https://pages.cpsc.ucalgary.ca/~robin/talks/itegory.pdf.

10 Robin Cockett and Stephen Lack. Restriction categories III: colimits, partial limits and extensivity. *Math. Struct Comput. Sci.*, 17(4):775–817, 2007. doi:10.1017/s0960129507006056.

11 Manfred Droste and Werner Kuich. Semirings and formal power series. In *Handbook of Weighted Automata*, pages 3–28. Springer, 2009. doi:10.1007/978-3-642-01492-5_1.

12 Calvin Elgot. Monadic computation and iterative algebraic theories. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium 1973*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 175–230. Elsevier, 1975. doi:10.1016/s0049-237x(08)71949-9.

13 Zoltán Ésik. Equational properties of fixed-point operations in cartesian categories: An overview. *Math. Struct. Comput. Sci.*, 29(6):909–925, 2019. doi:10.1017/s0960129518000361.

14 Wan J. Fokkink and Hans Zantema. Basic process algebra with iteration: Completeness of its equational axioms. *Comput. J.*, 37(4):259–268, 1994. doi:10.1093/comjnl/37.4.259.

15 Leandro Gomes, Alexandre Madeira, and Luís S. Barbosa. On Kleene algebras for weighted computation. In Simone Cavalheiro and José Fiadeiro, editors, *Proc. of 20th Brazilian Symp. on Formal Methods, SBMF 2017*, volume 10623 of *Lect. Notes in Comput. Sci.*, pages 271–286. Springer, 2017. doi:10.1007/978-3-319-70848-5_17.

16 Sergey Goncharov. Shades of iteration: From Elgot to Kleene. In Alexandre Madeira and Manuel A. Martins, editors, *Revised Selected Papers from 26th IFIP WG 1.3 Int. Workshop on Recent Trends in Algebraic Development Techniques, WADT 2022*, volume 13710 of *Lect. Notes in Comput. Sci.*, pages 100–120. Springer, 2023. doi:10.1007/978-3-031-43345-0_5.

17 Sergey Goncharov, Lutz Schröder, and Till Mossakowski. Kleene monads: Handling iteration in a framework of generic effects. In Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki, editors, *Proc. of 3rd Int. Conf. Algebra and Coalgebra in Computer Science, CALCO 2009*, volume 5728 of *Lect. Notes in Comput. Sci.*, pages 18–33. Springer, 2009. doi:10.1007/978-3-642-03741-2_3.

18 Sergey Goncharov, Lutz Schröder, Christoph Rauch, and Julian Jakob. Unguarded recursion on coinductive resumptions. *Log. Methods in Comput. Sci.*, 14(3):10:1–10:47, 2018. doi:10.23638/lmcs-14(3:10)2018.

19 Clemens Grabmayer. Milner's proof system for regular expressions modulo bisimilarity is complete: Crystallization: Near-collapsing process graph interpretations of regular expressions. In *Proc. of 37th Ann. ACM/IEEE Symp. on Logic in Computer Science, LICS '22*, pages 34:1–34:13, New York, 2022. doi:10.1145/3531130.3532430.

20 Niels Grathwohl, Dexter Kozen, and Konstantinos Mamouras. KAT + B! In *Proc. of 23rd EACL Ann. Conf. on Computer Science Logic and 29th Ann. ACM/IEEE Symp. on Logic in Computer Science, CSL-LICS 2014*, pages 44:1–44:10. ACM, 2014. doi:10.1145/2603088.2603095.

21 Bart Jacobs. New directions in categorical logic, for classical, probabilistic and quantum logic. *Log. Methods Comput. Sci.*, 11(3):24:1–24:76, 2015. doi:10.2168/lmcs-11(3:24)2015.

22 S. C. Kleene. Representation of events in nerve nets and finite automata. In Claude Shannon and John McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, 1956.

23 Dexter Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Inf. Comput.*, 110(2):366–390, 1994. doi:10.1006/inco.1994.1037.

24 Dexter Kozen. Kleene algebra with tests. *ACM Trans. Program. Lang. Syst.*, 19(3):427–443, 1997. doi:10.1145/256167.256195.

25 Dexter Kozen and Konstantinos Mamouras. Kleene algebra with products and iteration theories. In Simona Ronchi Della Rocca, editor, *Proc. of 22nd EACSL Ann. Conf. on Computer Science Logic, CSL 2013*, volume 23 of *Leibniz Int. Proc. in Inform.*, pages 415–431. Dagstuhl Publishing, 2013. doi:10.4230/lipics.csl.2013.415.

26 Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer, 1971.

27 Annabelle McIver, Tahiry M. Rabehaja, and Georg Struth. On probabilistic Kleene algebras, automata and simulations. In Harrie de Swart, editor, *Proc. of 12th Int. Conf. on Relational and Algebraic Methods in Computer Science, RAMICS 2011*, volume 6663 of *Lect. Notes in Comput. Sci.*, pages 264–279. Springer, 2011. doi:10.1007/978-3-642-21070-9_20.

28 Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lect. Notes in Comput. Sci.* Springer, 1980. doi:10.1007/3-540-10235-3.

29 Robin Milner. A complete inference system for a class of regular behaviours. *J. Comput. Syst. Sci.*, 28(3):439–466, 1984. doi:10.1016/0022-0000(84)90023-0.

30 Eugenio Moggi. Notions of computation and monads. *Inf. Comput.*, 93(1):55–92, 1991. doi:10.1016/0890-5401(91)90052-4.

31 Bernhard Möller. Kleene getting lazy. *Sci. Comput. Program.*, 65(2):195–214, 2007. doi:10.1016/j.scico.2006.01.010.

32 Maciej Piróg and Jeremy Gibbons. Monads for behaviour. *Electron. Notes Theor. Comput. Sci.*, 298:309–324, 2013. doi:10.1016/j.entcs.2013.09.019.

33 Gordon D. Plotkin. A powerdomain construction. *SIAM J. Comput.*, 5(3):452–487, 1976. doi:10.1137/0205035.

34 J.J.M.M. Rutten. Behavioural differential equations: A coinductive calculus of streams, automata, and power series. *Theor. Comput. Sci.*, 308(1):1–53, 2003. doi:10.1016/S0304-3975(02)00895-2.

35 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

36 Alex Simpson and Gordon Plotkin. Complete axioms for categorical fixed-point operators. In *Proc. of 15th Ann. IEEE Symp. on Logic in Computer Science, LICS '00*, pages 30–41. IEEE, 2000. doi:10.1109/lics.2000.855753.

37 Steffen Smolka, Nate Foster, Justin Hsu, Tobias Kappé, Dexter Kozen, and Alexandra Silva. Guarded Kleene algebra with tests: Verification of uninterpreted programs in nearly linear time. *Proc. ACM Program. Lang.*, 4(POPL), 2020. doi:10.1145/3371129.

38 Daniele Turi and Jan Rutten. On the foundations of final coalgebra semantics: non-well-founded sets, partial orders, metric spaces. *Math. Struct. Comput. Sci.*, 8(5):481–540, 1998. doi:10.1017/s0960129598002588.

39 Tarmo Uustalu. Generalizing substitution. *Theor. Inform. Appl.*, 37(4):315–336, 2003. doi:10.1051/ita:2003022.

A Selected Proof Details

Proof of Proposition 8. The necessity is obvious. Let us show sufficiency. For every $b \in \mathbf{C}^?(X)$, let us fix some choice of $\bar{b} \in \mathbf{C}^?(X)$, for which the pair $(b, \bar{b})$ satisfies contradiction and excluded middle, and show that $\mathbf{C}^?(X)$ forms a Boolean algebra, i.e. $\mathbf{C}^?(X)$ is a complemented distributive lattice. Complementation amounts to the assumed identities, and we are left to show the laws of distributive lattices. Since complements are uniquely defined in Boolean algebras, it will follow that $\bar{b}$ is uniquely determined by b. Of course, this is not used in the subsequent proof.

It follows by definition that $(\mathbf{C}^?(X), 0, +)$ and $(\mathbf{C}^?(X), 1, ;)$ are monoids. Showing that they are idempotent and commutative (hence, are semilattices) amounts to showing that $b ; b = b$ and $b ; c = c ; b$ for all $b, c \in \mathbf{C}^?(X)$. The first identity is shown as follows, using linearity and the assumed identities:

$$b = b ; 1 = b ; (b + \bar{b}) = b ; b + b ; \bar{b} = b ; b + 0 = b ; b.$$

For the second one, note that $1 = b + \bar{b} = b + b + \bar{b} = b + 1$, and then

$$\begin{aligned} b ; c &= b ; c ; 1 = b ; c ; (b + 1) = b ; c ; b + b ; c \\ &= b ; c ; b + b ; c ; b ; c = b ; c ; b ; (1 + c) = b ; c ; b, \end{aligned}$$

where we used the instance of idempotence $b ; c = b ; c ; b ; c$ that we just established. Analogously, $c ; b = b ; c ; b$, and hence $b ; c = c ; b$.

Finally, distributivity amounts to $(a + b) ; c = a ; c + b ; c$ and $a + b ; c = (a + b) ; (a + c)$ for all $a, b, c \in \mathbf{C}^?(X)$. The first identity is an axiom of idempotent grove categories. The second one is obtained as follows:

$$\begin{aligned} (a + b) ; (a + c) &= a ; a + a ; c + b ; a + b ; c = a + a ; c + a ; b + b ; c \\ &= a ; (1 + c + b) + b ; c = a ; 1 + b ; c = a + b ; c \end{aligned}$$ ◀

Proof of Lemma 10. Negation is defined as follows:

$$\bar{0} = 1, \qquad \overline{[\mathsf{in}_0, 0]} = [0, \mathsf{in}_1], \qquad \overline{b + c} = \bar{b} ; \bar{c},$$
$$\bar{1} = 0, \qquad \overline{[0, \mathsf{in}_1]} = [\mathsf{in}_0, 0], \qquad \overline{b ; c} = \bar{b} + \bar{c}.$$

For every X, let $\mathbf{C}^?(X)$ be the smallest subset of $\mathbf{C}(X, X)$ that contains 0, 1, $[\mathsf{in}_0, 0]$ and $[0, \mathsf{in}_1]$, and closed under + and ;. By Proposition 8, we need to show that every $b \in \mathbf{C}^?(X)$ is linear and satisfies $b ; \bar{b} = 0$, $b + \bar{b} = 1$, which we do by induction. Let us strengthen the induction invariant by also adding $\bar{b} ; b = 0$. Note that the above equations do not uniquely define complement, e.g. $\overline{[\mathsf{in}_0, 0]} = [0, \mathsf{in}_1]$ refers to a particular decomposition of X as $X_1 \oplus X_2$, while another decomposition could theoretically produce a different result. Thus, more precisely, we use the fact that every element of $\mathbf{C}^?(X)$ has a representation in the free algebra of terms over 0, 1, $[\mathsf{in}_0, 0]$, $[0, \mathsf{in}_1]$, + and ;. The claim is then obtained by induction over this representation. ◀

Proof of Lemma 35. The dependency condition is obvious in all three cases. We will prove finiteness of sets of derivatives only.

Sum. Let $t, s \in T_\nu n$ be rational. Then $\mathfrak{D}(t + s) = \{t + s\} \cup \mathfrak{D}(t) \cup \mathfrak{D}(s)$, which is finite, since $\mathfrak{D}(t)$ and $\mathfrak{D}(s)$ are so.

Composition. It suffices to stick to the following instance: given $n, k \in \mathbb{N}$, a rational element $t \in T_\nu n$ and a rational map $s\colon n \to T_\nu k$, show that $t ; s^\sharp \in T_\nu k$ is rational. Consider the set P of sums of the form

$$t' ; s^\sharp + \sum\nolimits_{s' \in \mathfrak{D}(s)} r_{s'} ; (s')^\sharp$$

where t' ranges over $\mathfrak{D}(t)$ and $r_{s'}$ range over non-guarded elements of $T_\nu n$. Then P is finite. Moreover, P is closed under derivatives: using Lemma 32,

$$\begin{aligned}\partial_{b,\mathsf{u}}(t' \,;\, s^\sharp + \textstyle\sum_{s'\in\mathfrak{D}(s)} r_{s'} \,;\, (s')^\sharp)&\\ = \partial_{b,\mathsf{u}}(t') \,;\, s^\sharp + o(t') \,;\, (\partial_{b,\mathsf{u}}(s))^\sharp + \textstyle\sum_{s'\in\mathfrak{D}(s)} \partial_{b,\mathsf{u}}(r_{s'}) \,;\, (s')^\sharp&\\ + \textstyle\sum_{s'\in\mathfrak{D}(s)} o(r_{s'}) \,;\, (\partial_{b,\mathsf{u}}(s'))^\sharp&\\ = \partial_{b,\mathsf{u}}(t') \,;\, s^\sharp + o(t') \,;\, (\partial_{b,\mathsf{u}}(s))^\sharp + \textstyle\sum_{s'\in\mathfrak{D}(s)} o(r_{s'}) \,;\, (\partial_{b,\mathsf{u}}(s'))^\sharp&\\ = \partial_{b,\mathsf{u}}(t') \,;\, s^\sharp + (o(t') + o(r_s)) \,;\, (\partial_{b,\mathsf{u}}(s))^\sharp + \textstyle\sum_{s'\in\mathfrak{D}(s)\smallsetminus\{s\}} o(r_{s'}) \,;\, (\partial_{b,\mathsf{u}}(s'))^\sharp,&\end{aligned}$$

and analogously for $\partial^k_{b,\mathsf{f}}$. Note that $t \,;\, s^\sharp \in P$. Therefore $\mathfrak{D}(t \,;\, s^\sharp) \subseteq P$. Since P is finite, so is $\mathfrak{D}(t \,;\, s^\sharp)$.

Iteration. Let $t = [t_0, \ldots, t_{n-1}] \colon n \to T_\nu n$, and $\mathfrak{D}(t_i)$ be finite for $i = 0, \ldots, n-1$. Analogously to the previous clause, consider the set P of sums of the form

$$\textstyle\sum_{t'\in\mathfrak{D}(t)} r_{t'} \,;\, (t')^\sharp \,;\, (t^*)^\sharp + r$$

where t' ranges over $\mathfrak{D}(t)$ and $r_{s'}$, r range over those non-guarded elements of $T_\nu n$. In the same manner as in the previous clause: P is finite, contains t^* and is closed under derivatives, hence $\mathfrak{D}(t^*)$ is finite. ◀

Proof of Theorem 37. The key observation is that $[\![\mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp]\!]_\uparrow$ does not depend on the choice of s and r. This is argued as follows. Using the construction in Proposition 36, for a given $t = \mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp$, we obtain a canonical representation $t = \mathsf{in}^l_n \,;\, \hat{s}^* \,;\, \hat{r}^\sharp$, with $\hat{s} \colon l \to T_\nu l$, $\hat{r} \colon l \to T_\nu k$, and this representation only depends on t, hence, it suffices to show that

$$[\![\mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp]\!]_\uparrow = [\![\mathsf{in}^l_n \,;\, \hat{s}^* \,;\, \hat{r}^\sharp]\!]_\uparrow. \tag{11}$$

Because of the restrictions on s and r, there is an epimorphism $u \colon m \to l$, such that $s \,;\, T_\nu u = u \,;\, \hat{s}$ and $u \,;\, \hat{r} = r$. W.l.o.g. assume that $\mathsf{in}_{l,m}$ is a left inverse of u. Now, (11) is obtained as follows:

$$\begin{aligned}[\![\mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp]\!]_\uparrow &= \mathsf{in}^m_n \,;\, [\![s]\!]^* \,;\, [\![r]\!]\\ &= \mathsf{in}^m_n \,;\, [\![s]\!]^* \,;\, [\![u]\!] \,;\, [\![\hat{r}]\!]\\ &= \mathsf{in}^m_n \,;\, u \,;\, [\![\hat{s}]\!]^* \,;\, [\![\hat{r}]\!] && /\!/ \ast\text{-}\mathbf{Uni}\\ &= \mathsf{in}^l_n \,;\, \mathsf{in}_{l,m} \,;\, u \,;\, [\![\hat{s}]\!]^* \,;\, [\![\hat{r}]\!]\\ &= \mathsf{in}^l_n \,;\, [\![\hat{s}]\!]^* \,;\, [\![\hat{r}]\!]\\ &= [\![\mathsf{in}^l_n \,;\, \hat{s}^* \,;\, \hat{r}^\sharp]\!]_\uparrow\end{aligned}$$

The defined lifting $[\![-]\!]_\uparrow \colon \mathfrak{F} \to \mathbf{C}$ is easily seen to make (9) commute. Also, note that there is no more than one structure-preserving candidate for $[\![-]\!]_\uparrow$, to make (9) commute: indeed, since every morphism in $\mathfrak{F}$ is representable as $t = \mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp$, $[\![t]\!]_\uparrow$ must only be defined as $\mathsf{in}^m_n \,;\, [\![s]\!]^* \,;\, [\![r]\!]$.

We are left to check that $[\![-]\!]_\uparrow$ is a KiCT-functor, which is facilitated by Lemma 38. The only non-trivial clause is preservation of Kleene star. As an auxiliary step, we show that

$$[\![\eta + \mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp]\!]_\uparrow = [\![\eta]\!]_\uparrow + [\![\mathsf{in}^m_n \,;\, s^* \,;\, r^\sharp]\!]_\uparrow \tag{12}$$

for any guarded flat $s\colon m \to T_\nu m$, and a non-guarded $r\colon m \to T_\nu n$. In order to calculate the left-hand side of (12), we need to find a suitable representation for $1 + \mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp$. Concretely, we show that

$$\eta + \mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp = \mathsf{in}_n^{m+m} \mathbin{;} [\mathsf{in}_1 \mathbin{;} \eta, s \mathbin{;} T_\nu\, \mathsf{in}_1]^* \mathbin{;} [[\eta_n, 0], r]^\sharp$$

Indeed, using $*$-**Fix** and $*$-**Uni**,

$$\begin{aligned}
&\mathsf{in}_n^{m+m} \mathbin{;} [\mathsf{in}_1 \mathbin{;} \eta, s \mathbin{;} T_\nu\, \mathsf{in}_1]^* \mathbin{;} [[\eta_n, 0], r]^\sharp \\
&\quad= \mathsf{in}_n^{m+m} \mathbin{;} (\eta + [\mathsf{in}_1 \mathbin{;} \eta, s \mathbin{;} T_\nu\, \mathsf{in}_1] \mathbin{;} ([\mathsf{in}_1 \mathbin{;} \eta, s \mathbin{;} T_\nu\, \mathsf{in}_1]^*)^\sharp) \mathbin{;} [[\eta_n, 0], r]^\sharp \\
&\quad= \eta + \mathsf{in}_n^m \mathbin{;} \mathsf{in}_1 \mathbin{;} \eta \mathbin{;} ([\mathsf{in}_1 \mathbin{;} \eta, s \mathbin{;} T_\nu\, \mathsf{in}_1]^*)^\sharp \mathbin{;} [[\eta_n, 0], r]^\sharp \\
&\quad= \eta + \mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} T_\nu\, \mathsf{in}_1 \mathbin{;} [[\eta_n, 0], r]^\sharp \\
&\quad= \eta + \mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp.
\end{aligned}$$

Now, (12) turns into

$$\mathsf{in}_n^{m+m} \mathbin{;} [\mathsf{in}_1, [\![s]\!] \mathbin{;} \mathsf{in}_1]^* \mathbin{;} [[1_n, 0], [\![r]\!]] = 1 + \mathsf{in}_n^m \mathbin{;} [\![s]\!]^* \mathbin{;} [\![r]\!].$$

This equation is shown as above, since $*$-**Fix** and $*$-**Uni** are sound for **C**.

An analogous method is used to show that $[\![-]\!]_\uparrow$ preserves Kleene star. Let $s\colon m \to T_\nu m$ be guarded flat, and let $r\colon m \to T_\nu n$ be non-guarded, and prove that:

$$[\![(\mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp)^*]\!]_\uparrow = [\![\mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp]\!]_\uparrow^*. \tag{13}$$

The following equation is provable using the axioms of KiC

$$(\mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp)^* = \eta + \mathsf{in}_n^m \mathbin{;} ((r \mathbin{;} T_\nu\, \mathsf{in}_n^m)^* \mathbin{;} s^\sharp)^* \mathbin{;} ((r \mathbin{;} T_\nu\, \mathsf{in}_n^m)^* \mathbin{;} r^\sharp)^\sharp,$$

hence, the equation

$$(\mathsf{in}_n^m \mathbin{;} [\![s]\!]^* \mathbin{;} [\![r]\!]^\sharp)^* = 1 + \mathsf{in}_n^m \mathbin{;} (([\![r]\!] \mathbin{;} \mathsf{in}_n^m)^* \mathbin{;} [\![s]\!])^* \mathbin{;} ([\![r]\!] \mathbin{;} \mathsf{in}_n^m)^* \mathbin{;} [\![r]\!]$$

is provable as well. Now, the proof of (13) is as follows:

$$\begin{aligned}
[\![(\mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp)^*]\!]_\uparrow &= [\![\eta + \mathsf{in}_n^m \mathbin{;} ((r \mathbin{;} T_\nu\, \mathsf{in}_n^m)^* \mathbin{;} s^\sharp)^* \mathbin{;} ((r \mathbin{;} T_\nu\, \mathsf{in}_n^m)^* \mathbin{;} r^\sharp)^\sharp]\!]_\uparrow \\
&= 1 + \mathsf{in}_n^m \mathbin{;} (([\![r]\!] \mathbin{;} \mathsf{in}_n^m)^* \mathbin{;} [\![s]\!]^\sharp)^* \mathbin{;} ([\![r]\!] \mathbin{;} \mathsf{in}_n^m)^* \mathbin{;} [\![r]\!] && /\!/ \ (12) \\
&= (\mathsf{in}_n^m \mathbin{;} [\![s]\!]^* \mathbin{;} [\![r]\!]^\sharp)^* \\
&= [\![\mathsf{in}_n^m \mathbin{;} s^* \mathbin{;} r^\sharp]\!]_\uparrow^*.
\end{aligned}$$

◀