Kleene Algebra with Products and Iteration Theories

Dexter Kozen[1] and Konstantinos Mamouras[2]

1,2 Computer Science Department, Cornell University, Ithaca, NY 14853, USA
{kozen,mamouras}@cs.cornell.edu

Abstract

We develop a typed equational system that subsumes both iteration theories and typed Kleene algebra in a common framework. Our approach is based on cartesian categories endowed with commutative strong monads to handle nondeterminism.

1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs

Keywords and phrases Kleene algebra, typed Kleene algebra, iteration theories

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

In the realm of equational systems for reasoning about iteration, two chief complementary bodies of work stand out. One of these is *iteration theories* (IT), the subject of the extensive monograph of Bloom and Ésik [4] as well as many other authors (see the cited literature). The primary motivation for iteration theories is to capture in abstract form the equational properties of iteration on structures that arise in domain theory and program semantics, such as continuous functions on ordered sets. Of central interest is the dagger operation †, a kind of parameterized least fixpoint operator, that when applied to an object representing a simultaneous system of equations gives an object representing the least solution of those equations. Much of the work on iteration theories involves axiomatizing or otherwise characterizing the equational theory of iteration as captured by †. Complete axiomatizations have been provided [7, 9, 10] as well as other algebraic and categorical characterizations [2, 3].

Bloom and Ésik claim that "... the notion of an iteration theory seems to axiomatize the equational properties of all computationally interesting structures..." [6]. This is true to a certain extent, certainly if one is interested only in structures that arise in domain theory and programming language semantics. However, it is not the entire story.

Another approach to equational reasoning about iteration that has met with some success over the years is the notion of *Kleene algebra* (KA), the algebra of regular expressions. KA has a long history going back to the original paper of Kleene [12] and was further developed by Conway, who coined the name Kleene algebra in his 1971 monograph [8]. It has since been studied by many authors. KA relies on an iteration operator * that characterizes iteration in a different way from †. Its principal models are not those of domain theory, but rather basic algebraic objects such as sets of strings (in which * gives the Kleene asterate operation), binary relations (in which * gives reflexive transitive closure), and other structures with applications in shortest path algorithms on graphs and geometry of convex sets. Complete axiomatizations and complexity analyses have been given; the regular sets of strings over an alphabet A form the free KA on generators A in much the same way that the rational $\Sigma_\perp$-trees form the free IT on a signature Σ.


© John Q. Open and Joan R. Access;
licensed under Creative Commons License CC-BY
Conference title on which this volume is based on.
Editors: Billy Editor, Bill Editors; pp. 1–17

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Although the two systems fulfill many of the same objectives and are related at some level, there are many technical and stylistic differences. Whereas iteration theories are based on Lawvere theories, a categorical concept, Kleene algebra operates primarily at a level of abstraction one click down. For this reason, KA may be somewhat more accessible. KA has been shown to be useful in several nontrivial static analysis and verification tasks (see e.g. [16, 23]). Also, KA can model nondeterministic computation, whereas IT is primarily deterministic.

Nevertheless, both systems have claimed to capture the notion of iteration in a fundamental way, and it is interesting to ask whether they can somehow be reconciled. This is the investigation that we have undertaken in this paper. We start with the observation that ITs use the objects of a category to represent types. Technically the objects of interest in ITs are morphisms $f : n \to m$ in a category whose objects are natural numbers, and the morphism $f : n \to m$ is meant to model functions $f : A^m \to A^n$ (the arrows are reversed for technical reasons). Thus ITs might be captured by a version of KA with types. Although the primary version of KA is untyped, there is a notion of typed KA [21], although it only has types of the form $A \to B$, whereas to subsume IT it would need products as well. The presence of products allow ITs to capture parameterized fixpoints through the rule

$$\frac{f : n \to n + m}{f^\dagger : n \to m}$$

giving the parameterized least fixpoint $f^\dagger : A^m \to A^n$ of a parameterized function $f : A^m \times A^n \to A^n$. This would be possible to capture in KA if the typed version had products, which it does not. On the other hand, KA allows the modeling of nondeterministic computation, which IT does not, at least not in any obvious way. Thus to capture both systems, it would seem that we need to extend the type system of typed KA, or extend the categorical framework of IT to handle nondeterminism, or both.

The result of our investigation is a common categorical framework based on cartesian categories (categories with products) combined with a monadic treatment of nondeterminism. Types are represented by objects in the category, and we identify the appropriate axioms in the form of typed equations that allow equational reasoning on the morphisms. Our framework captures iteration as represented in ITs and KAs in a common language. We show how to define the KA operations as enrichments on the morphisms and how to define † in terms of *. Our main contributions are as follows.

- **Commutative strong monads.** To accommodate nondeterminism, we need to lift the computation to the Kleisli category of a monad representing nondeterminate values. However, the ordinary powerset monad does not suffice for this purpose, as it does not interact well with non-strictness. We axiomatize the relevant properties for an arbitrary *commutative strong monad*, where (i) the property of *commutativity* captures the idea that the computation of a pair can be done in either order, and (ii) *strength* refers to tensorial strength, which axiomatizes the interaction of pairs with the nondeterminism monad.
- **Lazy pairs**. We need to model non-strict (lazy) evaluation of programs in the presence of products. Ordinarily, a pair $\langle x, \bot\rangle$ would be $\bot$ by strictness. This requires the development of the concept of *lazy pairs* and its categorical axiomatization. Intuitively, in the case of *eager pairs*, the computation of a pair $\langle v, \bot\rangle$, where v is a value and $\bot$ denotes a diverging computation, would also be diverging, i.e., $\langle v, \bot\rangle = \bot$. This makes it impossible to recover the left component v of the pair: $\langle v, \bot\rangle; \pi_1 = \bot$.
- **Simplified axiomatization of commutative strong monads and lazy pairs**. We have given a simplified axiomatization of commutative monads with lazy pairs in terms of a certain operator ψ that captures the interaction of these concepts in a very concise form, much simpler than the axiomatizations of the two of them separately. This is an adaptation

of a construction that can be found in the work of Kock in the 1970s [13, 15]. We use this extensively in our development to simplify arguments.

- **Deterministic arrows**. Certain properties work only for deterministic computations. We show how to capture the necessary properties of determinism in the Kleisli category. A separate syntactic arrow provides a convenient notation for reasoning about deterministic computation in the underlying category when working in the Kleisli category, and we provide an axiomatization of the necessary properties.
- **Lifting the cartesian structure**. We show how the cartesian structure (pairing and projections) in the underlying category can be lifted in a smooth way to corresponding operations in the Kleisli category.
- **Capturing nondeterminism**. We give three equivalent ways of capturing (angelic) nondeterminism in the homsets of the Kleisli category of a monad. These characterizations make essential use of cartesian structure of the base category.
- **Capturing IT and KA**. We show how to enrich the homsets of the Kleisli category with the KA operations, including *, to obtain typed KA with products, and that all the axioms of KA (except the strictness axiom) are satisfied. Sequential composition is modeled by Kleisli composition. We also show that Park theories [9] are subsumed by KA with products. This is our main result.
- **Model theory**. Finally, we show that two particular monads, the *lowerset monad* and the *ideal completion monad*, provide natural concrete models in that they are commutative strong monads with lazy pairs. The ideal completion monad involves ideal completion in ω-complete partial orders (ω-CPOs) and models nondeterminism in those structures.

2 Commutative Strong Monads with Lazy Pairs

As a first step in the development of our category-theoretic framework, we define axiomatically the base cartesian category of "non-strict functions and values", where the notion of divergence is made explicit. We handle nondeterminism with a commutative strong monad, for which the formation of lazy pairs is implemented with just one very natural axiom that intuitively says: "forming a pair of a non-diverging computation with a diverging computation allows one to recover the non-diverging component". The Kleisli category of the monad, which we think of as a category of "non-strict programs and computations", is symmetric monoidal. In fact, it possesses more structure, which we will exploit by making the notion of deterministic arrow explicit.

In order to model non-strict (lazy) evaluation of programs and lazy pairs we consider our base category to be a *cartesian category $\mathcal{C}$ with bottom elements*. For an object X, we write the identity for X as $\mathrm{id}_X : X \to X$. The composition operation is written as ; and the operands are given in diagrammatic order: the composite of the two arrows $f : X \to Y$ and $g : Y \to Z$ is written $f; g : X \to Z$. For objects X and Y, we denote by $X \times Y$ their product with corresponding left and right projections $\pi_1^{XY} : X \times Y \to X$ and $\pi_2^{XY} : X \times Y \to Y$ respectively. The pairing operation $\langle \cdot, \cdot \rangle$ takes two arrows $f : X \to Y$ and $g : X \to Z$ with the same domain and produces an arrow $\langle f, g \rangle : X \to Y \times Z$. The terminal object is denoted by $\mathbb{1}$. We write $\bot_X : X \to \mathbb{1}$ for the unique arrow from X to $\mathbb{1}$. The equational axioms

$$\langle f, g \rangle; \pi_1 = f \qquad \langle f, g \rangle; \pi_2 = g \qquad \langle h; \pi_1, h; \pi_2 \rangle = h \qquad f = \bot_X$$

say that the operations $\times, \pi_1, \pi_2, \langle \cdot, \cdot \rangle, \mathbb{1}, \bot$ endow $\mathcal{C}$ with cartesian structure. For every object X, the *bottom element* of X is written as $\bot_{\mathbb{1}X} : \mathbb{1} \to X$. Define the *bottom morphism* $\bot_{XY}$ from X to Y by $\bot_{XY} := \bot_X; \bot_{\mathbb{1}Y}$. The bottom morphisms satisfy the axiom $f; \bot_{YZ} = \bot_{XZ} : X \to Z$. An arrow $f : X \to Y$ satisfying the equation $\bot_{\mathbb{1}X}; f = \bot_{\mathbb{1}Y}$ is called *strict*.

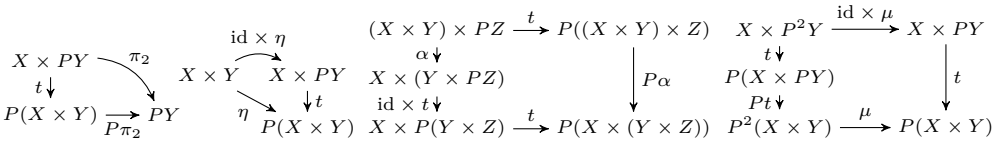

Figure 1 Axioms for tensorial strength $t : X \times PY \to P(X \times Y)$ of the monad (P, η, μ).

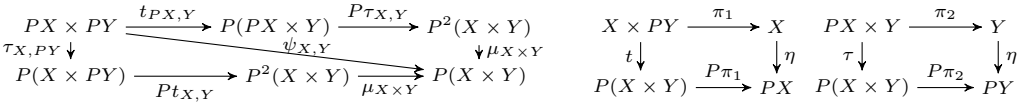

Figure 2 Commutativity axiom for the strong monad $(P, \eta, \mu), t$ (diagram on the left), and lazy pairs axiom in terms of t and equivalently in terms of τ (two diagrams on the right).

A *monad* (P, η, μ) over $\mathcal{C}$ consists of an endofunctor $P : \mathcal{C} \to \mathcal{C}$, and natural transformations $\eta_X : X \to PX$ and $\mu_X : P^2X \to PX$, called the *unit* and *multiplication* of the monad respectively, that satisfy: $P\mu_X; \mu_X = \mu_{PX}; \mu_X$, $\eta_{PX}; \mu_X = \mathrm{id}_{PX}$, and $P\eta_X; \mu_X = \mathrm{id}_{PX}$.

A monad (P, η, μ) is *strong* with *tensorial strength* $t_{X,Y} : X \times PY \to P(X \times Y)$ if t is a natural transformation satisfying the axioms of Figure 1. The arrow $\alpha_{X,Y,Z} : (X \times Y) \times Z \to X \times (Y \times Z)$ is the natural isomorphism defined as $\alpha := \langle \pi_1; \pi_1, \langle \pi_1; \pi_2, \pi_2 \rangle \rangle$. We define the *dual tensorial strength* $\tau_{X,Y} : PX \times Y \to P(X \times Y)$ as $\tau_{X,Y} = \mathsf{s}_{PX,Y}; t_{Y,X}; P\mathsf{s}_{Y,X}$, where $\mathsf{s}_{X,Y} : X \times Y \to Y \times X$ is the natural isomorphism given by $\mathsf{s}_{X,Y} = \langle \pi_2, \pi_1 \rangle$. The properties satisfied by t imply that τ is also a natural transformation and it satisfies the obvious dual axioms. For all objects X, Y define the morphism $\psi_{X,Y} : PX \times PY \to P(X \times Y)$ as $\psi_{X,Y} = t_{PX,Y}; P\tau_{X,Y}; \mu_{X \times Y}$. Using the fact that t and τ are natural transformations, it can be shown that ψ is also a natural transformation.

A strong monad $(P, \eta, \mu), t$ is *commutative* if the diagram on the left side of Fig. 2 commutes. Intuitively, the commutativity condition says that when computing a pair it does not matter in which order the components are computed. We are interested in strong monads that model *lazy pairs*. In the case of eager pairs, the computation of a pair $\langle v, \bot \rangle$, where v is a value and $\bot$ denotes a diverging computation, would also be diverging, i.e., $\langle v, \bot \rangle = \bot$. Therefore, it is not possible to recover the left component v of the pair: $\langle v, \bot \rangle; \pi_1 = \bot$. So, for lazy pairs we need an additional axiom, which can be given equivalently in terms of t or τ. The "lazy pairs" axiom says that the two diagrams on the right side of Fig. 2 commute.

▶ **Definition 1.** We say that $(P, \eta, \mu), t$ is a *commutative strong monad with lazy pairs* over $\mathcal{C}$ if (P, η, μ) is a monad with tensorial strength t so that the commutativity axiom of Figure 2 and the lazy pairs axiom of Figure 2 hold.

A commutative monad with lazy pairs can be equivalently given in terms of ψ. In Figure 3 we give properties that ψ satisfies, when it is defined in terms of t (see [13]). Conversely, we consider a monad (P, η, μ) over $\mathcal{C}$ together with a natural transformation $\psi : PX \times PY \to P(X \times Y)$ satisfying the axioms of Figure 3. Then, we can define t as $t_{X,Y} = (\eta_X \times \mathrm{id}_{PY}); \psi_{X,Y}$ and recover all the axioms we had given for $(P, \eta, \mu), t$ (see [15]). It has been proved by Anders Kock in [14] (Theorem 2.1) that for a commutative strong monad the axiom $\psi; \langle P\pi_1, P\pi_2 \rangle = \mathrm{id}$ is equivalent to $P\mathbb{1} \cong \mathbb{1}$, which says that P preserves final objects. We have opted for the former axiom in our definition, because it corresponds immediately to the intuition for the formation of lazy pairs.

The *Kleisli category* $\mathcal{C}_P$ has the same objects as $\mathcal{C}$. For all objects X, Y the homset $\mathcal{C}_P(X, Y)$ is equal to the homset $\mathcal{C}(X, PY)$. We use the notation $f : X \rightarrowtail Y$ for an arrow in $\mathcal{C}_P(X, Y)$. The composition operation in $\mathcal{C}_P$ is the *Kleisli composition* operation, denoted $\mathbf{;}$, and defined as $f \mathbf{;} g := f; Pg; \mu_Z : X \rightarrowtail Z$. For object X, the identity in $\mathcal{C}_P$ is $\eta_X : X \rightarrowtail X$.

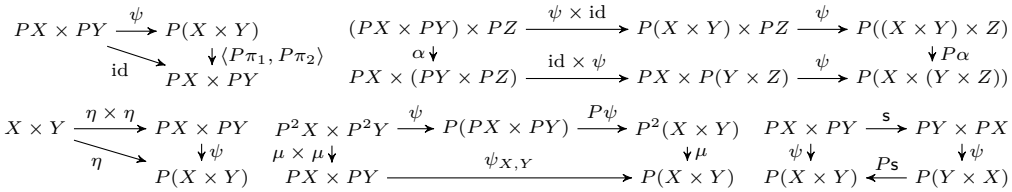

Figure 3 Commutative strong monad with lazy pairs (given in terms of ψ).

The equations $\eta_X ; f = f = f ; \eta_Y$ and $(f;g);h = f;(g;h)$ state that $\mathcal{C}_P$ is a category and they can be shown from the definitions and the monad axioms.

We define the map $H = (-;\eta)$ from the category $\mathcal{C}$ to the Kleisli category $\mathcal{C}_P$ by $HX := X$ and $Hf := f;\eta_Y : X \rightharpoonup Y$. We verify that H is a functor. First, note that it sends the identity $\mathrm{id}_X : X \to X$ of $\mathcal{C}$ to the identity $H\mathrm{id}_X = \eta_X : X \rightharpoonup X$ of $\mathcal{C}_P$. Moreover, the equation $H(f;g) = Hf ; Hg : X \rightharpoonup Z$ holds, which says that H commutes with composition. So, H is a functor from $\mathcal{C}$ to $\mathcal{C}_P$, which we will call here the *unit functor* of the monad. The functor H is the left adjoint of the Kleisli adjunction for the monad.

We say that an arrow $f : X \rightharpoonup Y$ of $\mathcal{C}_P$ is a *deterministic arrow* if there exists an arrow $f' : X \to Y$ of $\mathcal{C}$ such that $f = f';\eta_Y = Hf' : X \to PY$. So, the deterministic arrows of the Kleisli category are exactly the image of the arrows of $\mathcal{C}$ under the unit functor H. We indicate that f is a deterministic arrow of $\mathcal{C}_P$ by writing $f : X \to\!\!\!\!\bullet\, Y$. The Kleisli composite of two deterministic arrows is also deterministic. In our notation, if $f : X \to\!\!\!\!\bullet\, Y$ and $g : Y \to\!\!\!\!\bullet\, Z$ then $f;g : X \to\!\!\!\!\bullet\, Z$. The identity $\eta_X : X \to\!\!\!\!\bullet\, X$ is a deterministic arrow because $\eta_X = H\mathrm{id}_X$.

For the rest of this section, we assume that $(P,\eta,\mu),t$ is a commutative strong monad over $\mathcal{C}$ with lazy pairs. We will define "Kleisli versions" of projections, the pairing operation, and the product functor. We will prove useful properties that they satisfy. The notion of deterministic arrow turns out to be relevant.

We define the *Kleisli pairing* operation $\langle\!\langle \cdot,\cdot \rangle\!\rangle$ in $\mathcal{C}_P$ and the *Kleisli projections* (left and right) as follows: $\langle\!\langle f,g \rangle\!\rangle := \langle f,g\rangle ; \psi : X \rightharpoonup Y \times Z$ for $f : X \rightharpoonup Y$, $g : X \rightharpoonup Z$, and $\varpi_1 := H\pi_1 : X \times Y \to\!\!\!\!\bullet\, X$, $\varpi_2 := H\pi_2 : X \times Y \to\!\!\!\!\bullet\, Y$. The Kleisli projections are deterministic arrows. If $f : X \to\!\!\!\!\bullet\, Y$ and $g : X \to\!\!\!\!\bullet\, Z$ are deterministic arrows, then so is $\langle\!\langle f,g \rangle\!\rangle$. This is an immediate consequence of the equation $H\langle f,g\rangle = \langle\!\langle Hf, Hg \rangle\!\rangle : X \rightharpoonup Y \times Z$, which states that H commutes with the pairing operation.

▶ **Theorem 2.** *The following typed equations for Kleisli projections/pairing hold:*

$$\langle\!\langle f,g \rangle\!\rangle ; \varpi_1 = f : X \rightharpoonup Y \qquad \langle\!\langle h ; \varpi_1, h ; \varpi_2 \rangle\!\rangle = h : X \to\!\!\!\!\bullet\, Y \times Z$$
$$\langle\!\langle f,g \rangle\!\rangle ; \varpi_2 = g : X \rightharpoonup Z \qquad f ; \langle\!\langle g_1, g_2 \rangle\!\rangle = \langle\!\langle f;g_1, f;g_2 \rangle\!\rangle : X \rightharpoonup Z_1 \times Z_2$$

For the bottom-right equation, there is the extra hypothesis that $f : X \to\!\!\!\!\bullet\, Y$ is deterministic.

We define the operation $\otimes$ on $\mathcal{C}_P$, which we call *Kleisli product functor*, by $f_1 \otimes f_2 := (f_1 \times f_2);\psi : X_1 \times X_2 \rightharpoonup Y_1 \times Y_2$. Equivalently, we can define the Kleisli product as $f_1 \otimes f_2 = \langle\!\langle \varpi_1 ; f_1, \varpi_2 ; f_2 \rangle\!\rangle$. We observe that H commutes with the product functor, that is, $H(f \times g) = Hf \otimes Hg$. An easy consequence of the above results is that the Kleisli category $\mathcal{C}_P$ has symmetric monoidal structure given by $\otimes$ (tensor or monoidal product), $\mathbb{1}$ (identity object), $H\alpha : (X \otimes Y) \otimes Z \to\!\!\!\!\bullet\, X \otimes (Y \otimes Z)$ (associator), $\varpi_2 : \mathbb{1} \otimes X \to\!\!\!\!\bullet\, X$ (left unitor), $\varpi_1 : X \otimes \mathbb{1} \to\!\!\!\!\bullet\, X$ (right unitor), and $H\mathsf{s}_{X,Y} : X \otimes Y \to\!\!\!\!\bullet\, Y \otimes X$ (commutativity constraint).

3 Nondeterministic Monads

In this section we look at three equivalent ways of endowing with (angelic) nondeterministic structure the homsets of the Kleisli category of a monad. See also [11], where the similar

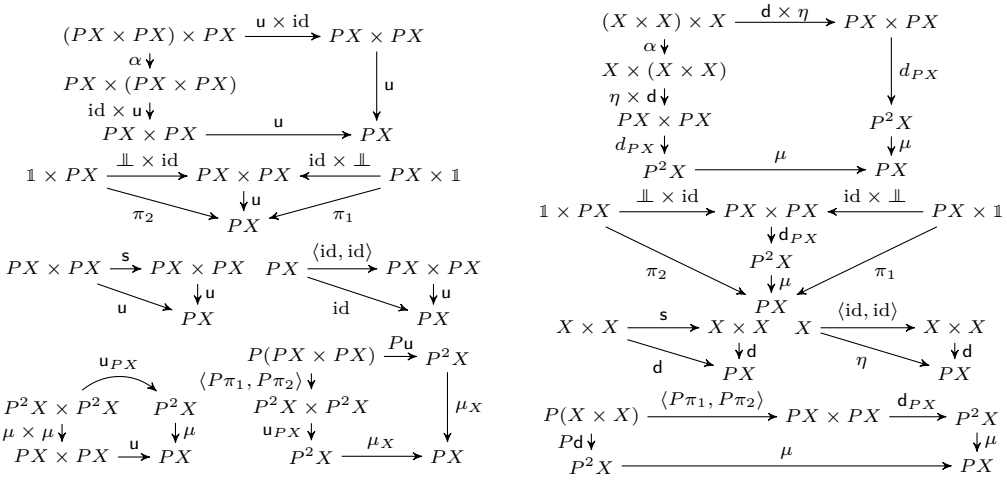

Figure 4 Axioms for $\mathsf{u}_X : PX \times PX \to PX$ (left side) and for $\mathsf{d}_X : X \times X \to PX$ (right side).

notion of additive monad is considered. The proofs of equivalence we present make essential use of the cartesian structure of the base category. The axiomatizations are simple and intuitive in that they correspond to familiar properties of the elementary operations of "binary union" and "formation of unordered pair", thus allowing us to easily identify models.

▶ **Definition 3.** Let (P, η, μ) be a monad over the category $\mathcal{C}$, let $\perp\!\!\!\perp_{XY} : X \to PY$ be a family of morphisms, and $+$ be a binary operation on $\mathcal{C}(X, PY)$ which we call *(nondeterministic) choice*. We say that $(P, \eta, \mu), +, \perp\!\!\!\perp$ is a *nondeterministic monad* if the axioms

$$(f+g)+h = f+(g+h) \qquad f+\perp\!\!\!\perp = f \qquad f;(g_1+g_2) = f;g_1 + f;g_2 \qquad f;\perp\!\!\!\perp = \perp\!\!\!\perp$$
$$f+g = g+f \qquad f+f = f \qquad (f_1+f_2);g = f_1;g + f_2;g$$

are satisfied. The above axioms state that every homset $\mathcal{C}(X, PY)$ is a commutative idempotent monoid w.r.t. $+$ and $\perp\!\!\!\perp$. Additionally, $+$ distributes over Kleisli composition, and $\perp\!\!\!\perp$ is a right annihilator for $;$.

Assuming the category $\mathcal{C}$ is cartesian, we will give an equivalent definition of the nondeterministic monad in terms of a natural transformation $\mathsf{u}_X : PX \times PX \to PX$, which we can intuitively think of as *binary union*. Then, we will also derive another equivalent definition of the nondeterministic monad in terms of a natural transformation $\mathsf{d}_X : X \times X \to PX$, which we think of as an operation that forms an *unordered pair* of two elements.

▶ **Theorem 4.** *Let $\mathcal{C}$ be a category with cartesian structure given by $\times, \pi_1, \pi_2, \langle\cdot,\cdot\rangle, \mathbb{1}, \bot$.*

- *Suppose $(P, \eta, \mu), +, \perp\!\!\!\perp$ is a nondeterministic monad. Define $\mathsf{u}_X := \pi_1 + \pi_2 : PX \times PX \to PX$. Then, u_X is natural and the diagrams on the left side of Figure 4 commute.*
- *Conversely, suppose that (P, η, μ) is a monad, $\mathsf{u}_X : PX \times PX \to PX$ is natural, and $\perp\!\!\!\perp_{\mathbb{1}X} : \mathbb{1} \to PX$ is a family of morphisms, so that the axioms on the left side of Figure 4 are satisfied. Define the operation $+$ on the homset $\mathcal{C}(X, PY)$ by $f + g := \langle f, g\rangle; \mathsf{u}_Y : X \to PY$, and $\perp\!\!\!\perp_{XY} := \bot_X; \perp\!\!\!\perp_{\mathbb{1}Y} : X \to PY$. Then, $(P, \eta, \mu), +, \perp\!\!\!\perp$ is a nondeterministic monad.*

It is easy to see that the two constructions described in Theorem 4 are mutually inverse. From the choice operation $+$ we obtain $\mathsf{u}_X = \pi_1 + \pi_2 : PX \times PX \to PX$ and then a new choice operation $\vee$ given by $f \vee g = \langle f, g\rangle; \mathsf{u}_X$. But we have that $f \vee g = \langle f, g\rangle; (\pi_1 + \pi_2) = \big(\langle f, g\rangle; \eta\big); (\pi_1 + \pi_2) = \langle f, g\rangle; \pi_1 + \langle f, g\rangle; \pi_2 = f + g$. For the other direction, we notice that from $\mathsf{u}_X : PX \times PX \to PX$ we obtain a choice operation $+$ and then a new binary union natural transformation $\hat{\mathsf{u}}_X = \pi_1 + \pi_2 = \langle \pi_1, \pi_2\rangle; \mathsf{u}_X = \mathrm{id}; \mathsf{u}_X = \mathsf{u}_X$.

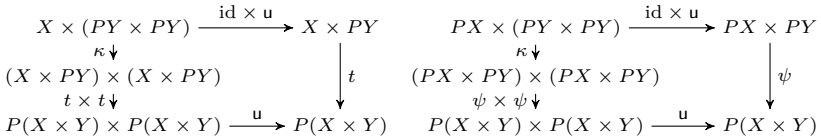

Figure 5 Axiom relating the binary union natural transformation $\mathsf{u}_X : PX \times PX \to PX$ with the tensorial strength $t : X \times PY \to P(X \times Y)$ or, equivalently, with $\psi : PX \times PY \to P(X \times Y)$.

▶ **Theorem 5.** *Let $\mathcal{C}$ be a category with cartesian structure given by $\times, \pi_1, \pi_2, \langle\cdot,\cdot\rangle, \mathbb{1}, \bot$. Let (P, η, μ) be a monad over $\mathcal{C}$, and $\mathbb{1}_{1X} : \mathbb{1} \to PX$ be a family of morphisms.*

- *Suppose that $\mathsf{u}_X : PX \times PX \to PX$ is natural and the diagrams on the left side of Figure 4 commute. Define $\mathsf{d}_X := (\eta_X \times \eta_X); \mathsf{u}_X : X \times X \to PX$. Then, d_X is natural and satisfies the axioms on the right side of Figure 4.*
- *Conversely, suppose that $\mathsf{d}_X : X \times X \to PX$ is natural and the diagrams on the right side of Figure 4 commute. Define $\mathsf{u}_X := \mathsf{d}_{PX}; \mu_X : PX \times PX \to PX$. Then, u is natural and satisfies the axioms on the left side of Figure 4.*

Again, the constructions described in Theorem 5 are mutually inverse. In one direction, we have $\mathsf{u}_X \mapsto \mathsf{d}_X = (\eta_X \times \eta_X); \mathsf{u}_X \mapsto \hat{\mathsf{u}}_X = \mathsf{d}_{PX}; \mu_X$, where $\hat{\mathsf{u}}_X = (\eta_{PX} \times \eta_{PX}); \mathsf{u}_{PX}; \mu_X = (\eta_{PX} \times \eta_{PX}); (\mu_X \times \mu_X); \mathsf{u}_X = \mathsf{u}_X$ using one of the axioms for u. In the other direction, we have $\mathsf{d}_X \mapsto \mathsf{u}_X = \mathsf{d}_{PX}; \mu_X \mapsto \hat{\mathsf{d}}_X = (\eta_X \times \eta_X); \mathsf{u}_X$ and therefore $\hat{\mathsf{d}}_X = (\eta_X \times \eta_X); \mathsf{d}_{PX}; \mu_X = \mathsf{d}_X; P\eta_X; \mu_X = \mathsf{d}_X$, since d is a natural transformation.

4 Nondeterministic Strong Monad with Iteration

In §3 we investigated the nondeterministic structure of the Kleisli category of a monad in isolation from products. This is not sufficient for our purposes, because we also want to capture the interaction between nondeterminism and products. For example, we would expect to be able to derive the property $\langle a + b, c\rangle = \langle a, c\rangle + \langle b, c\rangle$ for pairs with a nondeterminate component. So, we consider an additional axiom that relates the tensorial strength with the nondeterministic structure (Theorem 6). Then, we proceed to give our main definition of a nondeterministic strong monad with iteration, which puts together the axioms for all the symbols of our algebraic signature, including those for iteration (taken from [18]). The rest of the section is devoted to identifying models. We first consider the lowerset monad over pointed posets, which generalizes the familiar relational semantics of programs to the setting of lazy evaluation with lazy pairs. Then, we investigate the ideal completion monad over ω-CPOs, which is a kind of lower powerdomain construction. We derive several properties for this monad that will be essential for the main technical result of this paper: embedding the theory of † in KA with products.

▶ **Theorem 6.** *Let $\mathcal{C}$ be a category with cartesian structure given by $\times, \pi_1, \pi_2, \langle\cdot,\cdot\rangle, \mathbb{1}, \bot$. Let $(P, \eta, \mu), t$ be a commutative strong monad over $\mathcal{C}$ with tensorial strength $t_{X,Y} : X \times PY \to P(X \times Y)$. Assume additionally that (P, η, μ) is a nondeterministic monad together with $+$ and $\bot$, and also that the axiom $\langle \mathrm{id}, f + g\rangle; t = \langle \mathrm{id}, f\rangle; t + \langle \mathrm{id}, g\rangle; t$ holds. Then, the equations $\langle\!\langle f_1 + f_2, g\rangle\!\rangle = \langle\!\langle f_1, g\rangle\!\rangle + \langle\!\langle f_2, g\rangle\!\rangle$ and $\langle\!\langle f, g_1 + g_2\rangle\!\rangle = \langle\!\langle f, g_1\rangle\!\rangle + \langle\!\langle f, g_2\rangle\!\rangle$ hold.*

The stipulated axiom $\langle \mathrm{id}, f + g\rangle; t = \langle \mathrm{id}, f\rangle; t + \langle \mathrm{id}, g\rangle; t$ in Theorem 6 relates the nondeterministic structure, as given by the choice operation $+$, with the tensorial strength $t : X \times PY \to P(X \times Y)$. An equivalent characterization can be given in terms of the natural transformation $\mathsf{u}_X : PX \times PX \to PX$ as shown in Figure 5, where $\kappa =$

$\eta_X ; f = f$	$\langle f_1, f_2 \rangle ; \varpi_i = f_i$	$f = \perp\!\!\!\perp_{X\mathbb{1}}$
$f ; \eta_Y = f$	$\langle h ; \varpi_1, h ; \varpi_2 \rangle = h$ (h det.)	$f ; \perp\!\!\!\perp_{YZ} = \perp\!\!\!\perp_{XZ}$
$(f ; g) ; h = f ; (g ; h)$	f det.: $f ; \langle g_1, g_2 \rangle = \langle f ; g_1, f ; g_2 \rangle$	
	$(f_1 \otimes f_2) ; (g_1 \otimes g_2) = (f_1 ; g_1) \otimes (f_2 ; g_2)$	
$(f + g) + h = f + (g + h)$	$f ; (g_1 + g_2) = f ; g_1 + f ; g_2$	$\eta_X + f ; f^* \leq f^*$
$f + g = g + f$	$(f_1 + f_2) ; g = f_1 ; g + f_2 ; g$	$\eta_X + f^* ; f \leq f^*$
$f + \perp\!\!\!\perp = f$	$\langle h ; \varpi_1, h ; \varpi_2 \rangle \geq h$	$f ; g \leq g \Rightarrow f^* ; g \leq g$
$f + f = f$	$\langle f_1 + f_2, g \rangle = \langle f_1, g \rangle + \langle f_2, g \rangle$	$g ; f \leq g \Rightarrow g ; f^* \leq g$
	$\langle f, g_1 + g_2 \rangle = \langle f, g_1 \rangle + \langle f, g_2 \rangle$	

Table 1 Kleisli category of a nondeterministic strong monad with iteration.

$\langle \mathrm{id} \times \pi_1, \mathrm{id} \times \pi_2 \rangle : X \times (Y \times Z) \to (X \times Y) \times (X \times Z)$. In the set-theoretic models that we will consider, the equation corresponding to the right diagram of Figure 5 simply states that $A \times (B \cup C) = (A \times B) \cup (A \times C)$ for sets A, B, C. Similarly, we can also give an equivalent characterization in terms of $\mathsf{d}_X : X \times X \to PX$ according to the equation $(\mathrm{id}_X \times \mathsf{d}_Y) ; t_{X,Y} = \kappa ; \mathsf{d}_{X \times Y} : X \times (Y \times Y) \to P(X \times Y)$.

▶ **Definition 7.** Let $\mathcal{C}$ be a category with cartesian structure and bottom elements. We say that $(P, \eta, \mu), \psi, \mathsf{u}, {}^*$ is a *nondeterministic strong monad with iteration* if:

(i) $(P, \eta, \mu), \psi$ is a commutative strong monad with lazy pairs.

(ii) $(P, \eta, \mu), \mathsf{u}, \perp\!\!\!\perp$ is a nondeterministic monad, where $\perp\!\!\!\perp_{XY} = \perp_{XY} ; \eta_Y : X \to PY$.

(iii) For all $f, g : X \to PY$, the equation $\langle \mathrm{id}, f + g \rangle ; t = \langle \mathrm{id}, f \rangle ; t + \langle \mathrm{id}, g \rangle ; t$ holds, where t is the tensorial strength induced by ψ and $+$ is the choice operation induced by u.

(iv) The axiom $\mathrm{id}_{P(X \times Y)} \leq \langle P\pi_1, P\pi_2 \rangle ; \psi$ holds, where $\leq$ is the order induced by $+$.

(v) The *iteration* operation * sends $f : X \to PX$ to $f^* : X \to PX$. The axioms

$$\frac{f : X \rightharpoonup X}{\eta_X + f ; f^* \leq f^*} \qquad \frac{f : X \rightharpoonup X}{\eta_X + f^* ; f \leq f^*} \qquad \frac{f : X \rightharpoonup X \quad g : X \rightharpoonup Y}{f ; g \leq g \Rightarrow f^* ; g \leq g} \qquad \frac{g : X \rightharpoonup Y \quad f : Y \rightharpoonup Y}{g ; f \leq g \Rightarrow g ; f^* \leq g}$$

are satisfied. These are the axioms for the Kleene star operation introduced in [18, 19].

▶ **Theorem 8.** *Let $\mathcal{C}$ be a category with cartesian structure and bottom elements. Let $(P, \eta, \mu), \psi, \mathsf{u}, {}^*$ be a nondeterministic strong monad with iteration. The Kleisli category $\mathcal{C}_P$ with composition $;$ and identities η, together with the operations $\varpi_1, \varpi_2, \langle \cdot, \cdot \rangle, \perp\!\!\!\perp, +, {}^*$ (Kleisli projections, pairing, bottoms, choice, and iteration) satisfies the axioms of Table 1.*

A reasonable question to consider is whether there is any interaction between the iteration operation * and the tensorial strength. Even though it is not necessary for our purposes here, we note that two extra very natural axioms for the interaction between (non)determinism and Kleisli products together with the * axioms are sufficient to capture the interaction between * and ψ considered in [11]. We elaborate on this in the appendix.

4.1 The Lowerset Monad

In the usual relational interpretation of programs, a (strict) nondeterministic program $f : X \rightharpoonup Y$ is interpreted as a morphism in the category **Rel** of sets and binary relations. The category **Rel** is isomorphic to the Kleisli category $\mathbf{Set}_\wp$ of the powerset monad $\wp$ over the category **Set** of sets and total functions. In order to model lazy evaluation, the category $\mathbf{Set}_\wp$ (together with the operations of Kleisli composition and cartesian product) is not appropriate, since we need an explicit notion of divergence, non-strictness, and lazy pairs.

Instead, we consider the category **Pposet** of *pointed posets* (partial orders with a bottom element) and monotone functions, in which we can interpret non-strict deterministic programs that form lazy pairs. The bottom element $\perp_X$ of a pointed poset X denotes divergence. The arrows in **Pposet** can be partial, in the sense that they can send a non-bottom

element to bottom. For an object $(X, \leq)$ of **Pposet**, we understand the partial order $\leq$ as follows: $x \leq y$ intuitively means that x "has more diverging components" than y.

Pposet is a cartesian category with bottom elements. The product $X \times Y$ of two objects X, Y is the cartesian product together with the pointwise partial order. So, the bottom element of $X \times Y$ is $\bot_{X\times Y} = \langle \bot_X, \bot_Y \rangle$. The projections π_1 and π_2 are given by $\pi_i(x_1, x_2) = x_i$. The pairing operation $\langle \cdot, \cdot \rangle$ is defined as $\langle f, g \rangle = \lambda x \in X. \langle f(x), g(x) \rangle$ for $f : X \to Y$ and $g : X \to Z$. The terminal object is some singleton poset $\mathbb{1} = \{\bot_{\mathbb{1}}\}$. The bottom global element $\bot_{\mathbb{1}X} : \mathbb{1} \to X$ is the map that sends $\bot_{\mathbb{1}}$ to the bottom $\bot_X$ of X. So, $\bot_{XY} = \bot_X ; \bot_{\mathbb{1}Y}$ is the function that always diverges. We define the pointwise partial order $\leq$ on every homset $\mathbf{Pposet}(X, Y)$. Then, $\bot_{XY}$ is the bottom element of $\mathbf{Pposet}(X, Y)$.

▶ **Definition 9** (lowerset monad)**.** Let $(X, \leq)$ be a pointed poset, the bottom element of which is denoted $\bot_X$. For a subset $S \subseteq X$ define $\downarrow S = \{y \in X \mid y \leq x \text{ for some } x \in S\}$ to be the lowerset of X generated by S. For $x \in X$, we write $\downarrow x$ to mean $\downarrow\{x\}$. Define $\wp_\downarrow X$ to be the set of all non-empty lowersets of X. We observe that $\wp_\downarrow X$ is a complete lattice w.r.t. set inclusion. The top element is X and the bottom is $\{\bot_X\}$. The join is set-theoretic union and the meet is set-theoretic intersection. It follows that the homset $\mathbf{Pposet}(X, \wp_\downarrow Y)$ (w.r.t. the pointwise order induced by $\wp_\downarrow Y, \subseteq$) is also a complete lattice. We extend $\wp_\downarrow$ to an endofunctor $\mathbf{Pposet} \to \mathbf{Pposet}$ by putting $(\wp_\downarrow f)(S) := \downarrow\{f(x) \mid x \in S\}$ for every $S \in \wp_\downarrow X$. Together with the families of maps $\eta_X : x \in X \mapsto \downarrow x \in \wp_\downarrow X$ and $\mu_X : \mathcal{S} \in \wp^2_\downarrow X \mapsto \bigcup \mathcal{S} \in \wp_\downarrow X$ it forms a monad over **Pposet**. We call $(\wp_\downarrow, \eta, \mu)$ the *lowerset monad* over **Pposet**. Kleisli composition is given by $(f ; g)(x) = \bigcup_{y \in f(x)} g(y)$.

▶ **Theorem 10.** *Define $\psi_{X,Y} : \wp_\downarrow X \times \wp_\downarrow Y \to \wp_\downarrow(X \times Y)$ by $(S_1, S_2) \mapsto S_1 \times S_2$ and $\mathsf{u}_X : \wp_\downarrow X \times \wp_\downarrow X \to \wp_\downarrow X$ by $(S_1, S_2) \mapsto S_1 \cup S_2$. For $f : X \to \wp_\downarrow X$, define $f^i : X \to \wp_\downarrow X$ by induction: $f^0 = \eta_X$ and $f^{i+1} = f^i ; f$. Put $f^* := \bigvee_{i<\omega} f^i = \lambda x \in X. \bigcup_{i<\omega} f^i(x)$, where $\bigvee$ is the join of the complete lattice $\boldsymbol{Pposet}(X, \wp_\downarrow X)$. The lowerset monad $(\wp_\downarrow, \eta, \mu)$ over **Pposet**, together with $\psi, \mathsf{u}, ^*$, is a nondeterministic strong monad with iteration.*

4.2 The Ideal Completion Monad

An *ω-complete partial order* (ω-CPO) is a partially ordered set $(X, \leq)$ that has a least element $\bot_X$ and is *ω-complete* in the sense that every ω-chain (countable chain) $x_0 \leq x_1 \leq \cdots$ has a supremum $\sup_i x_i$. A function $f : X \to Y$ between ω-CPOs is called *ω-continuous* if it preserves suprema of ω-chains. That is, for every ω-chain $x_0 \leq x_1 \leq \cdots$ in X, $f(\sup_i x_i) = \sup_i f(x_i)$. An ω-continuous function is monotone. An ω-continuous function is *strict* if $f(\bot) = \bot$. If X, Y are ω-CPOs, then so is their cartesian product $X \times Y$ under the componentwise order: $(x_1, x_2) \leq (y_1, y_2)$ iff $x_1 \leq y_1 \wedge x_2 \leq y_2$. The least element is $\bot_{X\times Y} = (\bot_X, \bot_Y)$. For an ω-chain $(x_0, y_0) \leq (x_1, y_1) \leq \cdots$ in $X \times Y$, $\sup_i(x_i, y_i) = (\sup_i x_i, \sup_i y_i)$. We denote by $[X \to Y]$ the ω-CPO of all ω-continuous functions from X to Y ordered pointwise: $f \leq g$ iff $\forall x \in P.\ f(x) \leq g(x)$. The bottom element is $\bot_{XY} = \lambda x \in X. \bot_Y$. The supremum of a chain $f_0 \leq f_1 \leq \cdots$ in $[X \to Y]$ is $\sup_i f_i = \lambda x \in X. \sup_i f_i(x)$ and it is ω-continuous. The operations ; and $\langle \cdot, \cdot \rangle$ are monotone in all arguments. The ω-continuous functions on ω-CPOs are closed under well-typed composition and pairing and contain all identities and projections. Thus, ω-CPOs and ω-continuous functions form a cartesian category with bottom elements denoted **CPO**.

Let $(X, \leq)$ be an ω-CPO. A subset $I \subseteq X$ is called an *ideal* of X if it is a non-empty lower set and is closed under suprema of ω-chains. The set of all ideals of X is denoted $\mathfrak{I}X$. We denote by $\mathrm{cl}_X(S)$ the smallest ideal containing $S \subseteq X$. This is an operation of type $\mathrm{cl}_X : \wp X \to \mathfrak{I}X$. We also write $\mathrm{cl}(S)$ instead of $\mathrm{cl}_X(S)$ when no confusion arises. We say

$$\frac{f : X \to\!\!\!\!\bullet\; Y}{f : X \rightharpoonup Y} \qquad \frac{f : X \rightharpoonup Y \qquad g : Y \rightharpoonup Z}{f;g : X \rightharpoonup Z} \qquad \frac{f : X \rightharpoonup Y \qquad g : X \rightharpoonup Z}{\langle f, g\rangle : X \rightharpoonup Y \times Z} \qquad \frac{f, g : X \rightharpoonup Y}{f + g : X \rightharpoonup Y}$$

$$\frac{f : X \to\!\!\!\!\bullet\; Y \qquad g : Y \to\!\!\!\!\bullet\; Z}{f;g : X \to\!\!\!\!\bullet\; Z} \qquad \frac{f : X \to\!\!\!\!\bullet\; Y \qquad g : X \to\!\!\!\!\bullet\; Z}{\langle f, g\rangle : X \to\!\!\!\!\bullet\; Y \times Z} \qquad \frac{f : X \rightharpoonup X}{f^* : X \rightharpoonup X}$$

Table 2 Typing rules for KA with products.

that $\mathrm{cl}(S)$ is the *ideal generated by* S. The set $\downarrow x$ is an ideal and we call it the *principal ideal* generated by x. If $x_1 \leq x_2 \leq \ldots$ is an ω-chain in X, then $\mathrm{cl}[\{x_1, x_2, \ldots\}] = \downarrow \sup_i x_i$. The set $\mathfrak{I}X$ of all ideals of an ω-CPO X is a complete lattice w.r.t. set-theoretic inclusion. The meet $\bigwedge$ is set-theoretic intersection and the join $\bigvee$ is the ideal generated by the set-theoretic union. The bottom element is $\{\bot_X\}$ and the top element is X. If I, J are ideals of X, then so is $I \cup J$. In particular, $\mathrm{cl}(S_1 \cup S_2) = \mathrm{cl}(S_1) \cup \mathrm{cl}(S_2)$. Let X, Y be ω-CPOs. If I is an ideal of X and J is an ideal of Y, then $I \times J$ is an ideal of the ω-CPO $X \times Y$. If K is an ideal of $X \times Y$, then the left and right projections of K are ideals of X and Y respectively.

▶ **Definition 11** (the ideal completion monad)**.** We extend $\mathfrak{I}$ to an endofunctor $\mathbf{CPO} \to \mathbf{CPO}$ by putting $(\mathfrak{I}f)(S) := \mathrm{cl}_Y[f(S)] = \mathrm{cl}_Y\{f(x) \mid x \in S\}$, for every $S \in \mathfrak{I}X$, where $f : X \to Y$. We define the family of functions $\eta_X : X \to \mathfrak{I}X$ by $\eta_X(x) := \downarrow x$, and $\mu_X : \mathfrak{I}^2 X \to \mathfrak{I}X$ by $\mu_X(\mathcal{S}) := \bigvee \mathcal{S}$. Now, we claim that $(\mathfrak{I}, \eta, \mu)$ is a monad, called the *ideal completion monad* of $\mathbf{CPO}$. Kleisli composition can be given as $(f ; g)(x) = \bigvee_{y \in f(x)} g(y)$.

▶ **Theorem 12.** *Define* $\psi_{X,Y} : \mathfrak{I}X \times \mathfrak{I}Y \to \mathfrak{I}(X \times Y)$ *as* $(I, J) \mapsto I \times J$ *and* $\mathsf{u}_X : \mathfrak{I}X \times \mathfrak{I}X \to \mathfrak{I}X$ *as* $(I, J) \mapsto I \cup J$*. Also define the iteration operation by* $f^* := \bigvee_{i<\omega} f^i = \lambda x \in X.\ \bigvee_{i<\omega} f^i(x)$*. The ideal completion monad* $(\mathfrak{I}, \eta, \mu)$ *over the category* $\mathbf{CPO}$*, together with* $\psi, \mathsf{u}, ^*$*, is a nondeterministic strong monad with iteration.*

5 Typed Kleene Algebra with Products

Let Ω be a set of *atomic types*. Let $\mathbb{1} \notin \Omega$ be a special constant called the *unit type*. The set of *types over* Ω, denoted $\mathsf{Types}(\Omega)$, is the set of terms freely generated by Ω and $\mathbb{1}$ under the binary product type constructor $\times$. The terms of the language are typed. The types of terms are expressions of the form $X \rightharpoonup Y$, where $X, Y \in \mathsf{Types}(\Omega)$. We indicate the type of a term by writing $f : X \rightharpoonup Y$. Some of the terms will be designated as *deterministic* by writing $f : X \to\!\!\!\!\bullet\; Y$. We think of $X \to\!\!\!\!\bullet\; Y$ as a subtype of $X \rightharpoonup Y$ and hence we include the typing rule given in the first column of Table 2.

Let H be a set of *atomic arrows*, each endowed with a fixed type $h : X \rightharpoonup Y$. We write $h : X \to\!\!\!\!\bullet\; Y$ for a deterministic atomic arrow of H. Let id, π_1, π_2, $\bot$ be special deterministic polymorphic constants called *identities*, *left projections*, *right projections*, and *bottoms*, respectively. Where necessary, we use subscripts or superscripts to denote the specialization at a particular type; e.g., $\mathrm{id}_X : X \to\!\!\!\!\bullet\; X$, $\pi_1^{XY} : X \times Y \to\!\!\!\!\bullet\; X$, or $\bot_{XY} : X \to\!\!\!\!\bullet\; Y$. Let $;$ and $\langle \cdot, \cdot \rangle$ be polymorphic constructors called *composition* and *pairing*, respectively, satisfying the typing rules shown in Table 2. Note that compositions $f ; g$ are written in diagrammatic order. Composition and pairing preserve determinism. We add to the language the polymorphic constructors $+$ and *, called *(nondeterministic) choice* and *iteration* respectively, with the typing rules given in Table 2. Choice and iteration introduce nondeterminism.

▶ **Definition 13.** A *typed Kleene algebra with products* is a multi-sorted algebraic structure $\mathcal{K} = \big(\mathcal{K}_0, \times, \mathbb{1}, \mathcal{K}_1, \mathcal{K}_1^d, \mathsf{dom}, \mathsf{cod}, +, ;, ^*, \bot, \mathrm{id}, \langle \cdot, \cdot \rangle, \pi_1, \pi_2\big)$, where $\mathcal{K}_0$ is the set of *objects*, $\mathcal{K}_1$ is the set of *arrows* or *elements*, and $\mathcal{K}_1^d \subseteq \mathcal{K}_1$ is the set of *deterministic arrows*.

Park	KA
$f \le g \wedge g \le h \Rightarrow f \le h$ $f \le g \wedge g \le f \Rightarrow f = g$ $\bot \le f$ and $f \le f$	$f + (g + h) = (f + g) + h$ $f + g = g + f$ $f + \bot = f = f + f$
$f;(g;h) = (f;g);h$ $\mathrm{id}; f = f = f; \mathrm{id}$	$f;(g;h) = (f;g);h$ $f; \mathrm{id} = f = f; \mathrm{id}$
$g_1 \le g_2 \Rightarrow f;g_1 \le f;g_2$ $f_1 \le f_2 \Rightarrow f_1;g \le f_2;g$	$f;(g_1 + g_2) = f;g_1 + f;g_2$ $(f_1 + f_2);g = f_1;g + f_2;g$
	$f;\bot = \bot$
$\langle f^\dagger, \mathrm{id}_Y\rangle; f \le f^\dagger$ $\langle g, \mathrm{id}_Y\rangle; f \le g \Rightarrow f^\dagger \le g$ $g; f^\dagger \le [(\mathrm{id}_X \times g); f]^\dagger$	$\mathrm{id} + f; f^* \le f^* \quad f;g \le g \to f^*;g \le g$ $\mathrm{id} + f^*; f \le f^* \quad g;f \le g \to g;f^* \le g$
$f = \bot_X : X \to 1 \quad \langle h;\pi_1, h;\pi_2\rangle = h$ $\langle f_1, f_2\rangle;\pi_i = f_i$	$f = \bot_X : X \to 1 \quad \langle h;\pi_1, h;\pi_2\rangle = h$ (det. h) $\langle f_1, f_2\rangle;\pi_i = f_i$
	$(f_1 \times f_2);(g_1 \times g_2) = (f_1;g_1) \times (f_2;g_2)$ $h \le \langle h;\pi_1, h;\pi_2\rangle$ $f;\langle g_1, g_2\rangle = \langle f;g_1, f;g_2\rangle$ (det. f)
$f_1 \le f_2 \Rightarrow \langle f_1, g\rangle \le \langle f_2, g\rangle$ $g_1 \le g_2 \Rightarrow \langle f, g_1\rangle \le \langle f, g_2\rangle$	$\langle f_1 + f_2, g\rangle = \langle f_1, g\rangle + \langle f_2, g\rangle$ $\langle f, g_1 + g_2\rangle = \langle f, g_1\rangle + \langle f, g_2\rangle$

Table 3 Axioms of Park and KA.

- The operations dom and cod (called *domain* and *codomain*) map arrows to objects.
- The *type* of an element f of $\mathcal{K}$ is the expression $X \rightharpoonup Y$, where $X = \mathsf{dom} f$ and $Y = \mathsf{cod} f$. We write $f : X \rightharpoonup Y$ to denote this. If $f \in \mathcal{K}_1^d$, we write $f : X \to Y$.
- The distinguished *product* operation $\times$ is a function $\times : \mathcal{K}_0 \times \mathcal{K}_0 \to \mathcal{K}_0$. The object $\mathbb{1} \in \mathcal{K}_0$ is the distinguished *terminal object* of the structure.
- The polymorphic operations and constants $+$, $;$, *, $\bot$, id, $\langle\cdot,\cdot\rangle$, π_1, and π_2 satisfy the expected typing rules.
- Additionally, the structure is a model of the well-typed instances of the axioms given in the second column of Table 3. The partial order $\le$ is induced by $+$: $f \le g$ iff $f + g = g$. The product $\times$ is defined by: $f \times g = \langle \pi_1; f, \pi_2; g\rangle$.

We have included all the axioms of KA [18, 19] except for the *strictness* axiom $\bot; f = \bot$.

We will denote by KA the quasi-equational system of typed Kleene algebra with products. Notice that (up to a slight change in notation to make it less cumbersome), the axioms satisfied by a typed Kleene algebra with products are exactly those given in Table 1. So, any small subcategory of the Kleisli category of a nondeterministic strong monad with iteration that is closed under the appropriate operations is a typed Kleene algebra with products.

6 Iteration Theories

The language of iteration theories consists of *atomic typed actions* $h : n \to m$, where n, m are natural numbers, and polymorphic operation symbols ; (composition), id (identity), ι (injection), $[-,-,\cdots,-]$ (cotupling), $\bot$ (bottom), † (dagger). The typing rules for the language are the following: $\mathrm{id}_n : n \to n$, $\iota_i^n : 1 \to n$, $\bot_{nm} : n \to m$, and

$$\frac{f : n \to m \qquad g : m \to p}{f;g : n \to p} \qquad \frac{f_i : 1 \to m \qquad i = 1, \ldots, n}{[f_1, f_2, \ldots, f_n] : n \to m} \qquad \frac{f : n \to n + p}{f^\dagger : n \to p}$$

The *typed terms* $f : n \to m$ of the language are built from the atomic actions and the operation symbols according to the typing rules.

Consider now the category **CPO** of ω-CPOs and ω-continuous maps, which will provide the *standard interpretation* for the language of iteration theories. An interpretation $[\![\cdot]\!]$ in **CPO** consists of an ω-CPO A and a mapping of every atomic symbol $h : n \to m$ to

a morphism $[\![h]\!] : A^m \to A^n$ in **CPO**, where A^k denotes the k-fold associative cartesian product of A. The identity symbol $\mathrm{id}_n : n \to n$ is interpreted as the identity function $[\![\mathrm{id}_n]\!] : A^n \to A^n$. The injection symbol $\iota_i^n : 1 \to n$ is interpreted as the i-th projection $[\![\iota_i^n]\!] : A^n \to A$. The bottom symbol $\bot_{nm} : n \to m$ is interpreted as the least function $[\![\bot_{nm}]\!] : A^m \to A^n$ of $\mathbf{CPO}(A^m, A^n)$. Now, ; and $[-,-,\cdots,-]$ are interpreted as function composition and tupling respectively. For functions $f_i : 1 \to m$, $i = 1,\ldots,n$, the function denoted by $[f_1,\ldots,f_n] : n \to m$ is given by $[\![[f_1,\ldots,f_n]]\!](\bar{x}) := \langle [\![f_1]\!](\bar{x}),\ldots,[\![f_n]\!](\bar{x})\rangle$, for every $\bar{x} \in A^m$. Every ω-continuous function $\phi : X \to X$ in **CPO** has a least fixpoint, which is the supremum of the ω-chain $\bot \le \phi(\bot) \le \phi^2(\bot) \le \cdots \le \phi^n(\bot) \le \cdots$, where $\phi^0 = \mathrm{id}$ and $\phi^{n+1} = \phi^n; \phi$. We denote the least fixpoint of ϕ by $\mu(\phi) = \sup_i \phi^i(\bot)$. For an ω-continuous function $\phi : X \times Y \to X$, we define the function $\phi^\dagger : Y \to X$ by $\phi^\dagger(y) := \mu(\phi_y) = \sup_i \phi_y^i(\bot)$, where $\phi_y = \lambda x \in X.\phi(x,y) : X \to X$. The function $\phi^\dagger : Y \to X$ is also ω-continuous. We call † the *parametric fixpoint operation*. The operation † is monotone. We interpret the dagger symbol † of the language as parametric fixpoint †: for $f : n \to n+p$ we define $[\![f^\dagger]\!] = \big([\![f]\!] : A^n \times A^p \to A^n\big)^\dagger : A^p \to A^n$.

Every homset $\mathbf{CPO}(X,Y)$ is equipped with the pointwise partial order $\le$. For terms s,t we write $\mathbf{CPO} \models s \le t$ to mean that $[\![s]\!] \le [\![t]\!]$ for every interpretation $[\![\cdot]\!]$ of the language in **CPO**. Define $\mathsf{Th}(\mathbf{CPO})$ to be the set of all valid inequalities over **CPO**, that is, $\mathsf{Th}(\mathbf{CPO}) := \{s \le t \mid \mathbf{CPO} \models s \le t\}$, where s,t are terms of the language of iteration theories. $\mathsf{Th}(\mathbf{CPO})$ is the "(in)equational theory of iteration", in the words of Ésik [9].

▶ **Definition 14.** Define Park to be the universal Horn system (including equality) with: axioms for categories, axioms asserting that ι, $[-,-,\cdots,-]$, and $\bot$ give associative categorical coproducts, axioms stating that $\le$ is a partial order, that ; and $[-,-,\cdots,-]$ are motonone in all arguments w.r.t. $\le$, and that $\bot_{nm} : n \to m$ is the least element of $\mathrm{Hom}(n,m)$, and axioms for the dagger operation:

$$\frac{f : n \to n+p}{f;[f^\dagger,\mathrm{id}_p] \le f^\dagger : n \to p} \qquad \frac{f : n \to n+p \quad g : n \to p}{f;[g,\mathrm{id}_p] \le g \Rightarrow f^\dagger \le g} \qquad \frac{f : n \to n+p \quad g : p \to q}{f^\dagger; g \le [f;(\mathrm{id}_n \oplus g)]^\dagger : n \to q}$$

where the copairing operation $[-,-]$ is induced by the cotupling operation $[-,-,\ldots,-]$ in the obvious way. The three dagger axioms are called *pre-fixpoint inequality*, *least pre-fixpoint implication* or *Park induction rule*, and *parameter inequality* respectively.

▶ **Theorem 15** (Ésik [9]). *Park axiomatizes Th*(**CPO**)*, that is,* $\mathbf{CPO} \models t_1 \le t_2$ *iff* Park $\vdash t_1 \le t_2$*, for all terms* t_1, t_2 *in the language of iteration theories.*

The choice of a language with coproducts and copairing/injection symbols is confusing, because the standard models we are interested in here are models of functions where the symbols are interpreted as products and pairing/projections respectively. Moreover, there is no reason to collapse isomorphic products, such as $X \times (Y \times Z) \cong (X \times Y) \times Z$ or $1 \times X \cong X$. In fact, this only complicates the technical presentation of proofs.

So, we consider for the rest of the paper (as is also done in [6, 26]) that the language of iteration theories is instead as follows: For a set Ω of atomic types, let $\mathsf{Types}(\Omega)$ be the set freely generated by Ω, $\mathbb{1} \notin \Omega$, and the product constructor $\times$. The terms of the language are typed, e.g., $f : X \to Y$, where $X, Y \in \mathsf{Types}(\Omega)$. Each atomic arrow has a fixed type $h : X \to Y$. We have polymorphic constants π_1^{XY}, π_2^{XY}, id_X, $\bot_{XY}$ and polymorphic constructors ;, $\langle\cdot,\cdot\rangle$, †. The typing rules are as usual with the exception of the rule for †: for $f : X \times Y \to X$ we have $f^\dagger : Y \to X$. Now, a standard interpretation $[\![\cdot]\!]$ in **CPO** assigns an ω-CPO to each base type and an ω-continuous function to each atomic action. This extends in the obvious way to all terms of the language. In particular, the dagger symbol is interpreted as parametric fixpoint, e.g., $[\![f^\dagger]\!] = [\![f]\!]^\dagger : [\![Y]\!] \to [\![X]\!]$ for $f : X \times Y \to X$. See the first column of Table 3 for the axioms of Park in the language with products.

7 Embedding the Equational Theory of Iteration in KA

We augment the system of KA that we presented in Section 5 with an additional typing rule about the preservation of determinism:

$$\frac{g : X \to Y \qquad f : Y \to Y \qquad g \leq g; f : X \rightharpoonup Y}{g; f^* : X \to Y}. \tag{7}$$

We note that this rule is not valid in all Kleene algebras, but it is valid in the Kleisli category $\mathbf{CPO}_\mathfrak{I}$ of the ideal completion monad over **CPO**. For a term $f : X \times Y \rightharpoonup X$ of KA, we define the abbreviation $f^\ddagger := \langle \bot_{YX}, \mathrm{id}_Y \rangle; \langle f, \pi_2 \rangle^*; \pi_1 : Y \rightharpoonup X$. We call ‡ the *derived dagger operation* in KA. Using the rule (7) we can derive in KA the *dagger typing rule*: if $f : X \times Y \to X$ then $f^\ddagger : Y \to X$. The dagger typing rule states that the derived dagger operation preserves determinism.

▶ **Definition 16** (translation)**.** We define a *translation* $[\cdot]$ from the language of iteration theories to the language of KA with products. All atomic action symbols and atomic constants are translated as deterministic symbols of the same type. E.g., for $h : X \to Y$ we have $[h] = h : X \to Y$, and $[\pi_1] = \pi_1 : X \times Y \to X$. The dagger is translated as

$$[f^\dagger] := f^\ddagger = \langle \bot, \mathrm{id} \rangle; \langle [f], \pi_2 \rangle^*; \pi_1 : Y \to X$$

The translation function $[\cdot]$ commutes with the rest of the operation symbols.

▶ **Theorem 17** (Completeness and soundness)**.** *Let $t \leq t'$ be an inequality in the language of Park. Then,* $\mathbf{CPO} \models t \leq t'$ *iff* KA $\vdash [t] \leq [t']$.

Proof sketch. For completeness we use Theorem 15 and show how to obtain the Park axioms in KA. For soundness we exhibit an operation-preserving embedding of **CPO** in $\mathbf{CPO}_\mathfrak{I}$. ◀

Even though the above result was developed using Park theories, which have a universal Horn axiomatization, we also obtain a result for the equational theory of iteration theories. Recall the definition (see e.g. [7]): an *iteration theory* is a cartesian category with a dagger operation that satisfies the equations valid in **CPO**. So, the above theorem also says that the equational theory of iteration theories is embedded in KA.

8 Related Work

The work by Goncharov [11] is closely related to ours. He defines *additive (strong) monads* and *Kleene monads* axiomatically. Calculi for an extended metalanguage of effects are defined and completeness/incompleteness results are obtained. Our notion of a "nondeterministic strong monad with iteration" is different from that of a Kleene monad: we consider non-strict programs that form lazy pairs, and we axiomatize iteration quasi-equationally. The absence of the strictness axiom $\bot; f = \bot$ from our axiomatization and the use of lazy pairs are essential for our encoding of fixpoints. In particular, the axioms stipulated in [11] would force all the parametric fixpoints to be equal to $\bot$, because in that system $\langle \bot, \mathrm{id} \rangle; \langle f, \pi_2 \rangle^*; \pi_1 = \bot; \langle f, \pi_2 \rangle^*; \pi_1 = \bot$. So, the models we are investigating in the present work are crucially different from the models considered in [11].

The work on Hoare powerdomains [1, 25], which give models of angelic nondeterministic computations, is related. The (lower) Hoare powerdomain of a domain is formed by taking all the ideals of the domain, where by ideal we mean here the nonempty Scott-closed subsets of the domain. In the present work, we identify models of the axiomatically defined "nondeterministic monad with iteration," which are similar to and simpler than the construction of Hoare powerdomains over DCPOs. We first identify a simple model: the

lowerset monad over the category of posets with bottom elements. Then, we also prove that the ideal completion monad over the category of ω-CPOs is a model.

There is a long line of work, primarily by Bloom and Ésik, under the name of "iteration theories" or the "(in)equational theory of iteration" (see e.g. [4, 6, 7, 9, 26] and references therein), which is intimately related to the work on Kleene algebra [8, 16–23] in general and the present work in particular. The axioms of iteration theories capture the equational properties of fixpoints in several classes of structures relevant to computer science. For example, they capture the equational theory of ω-continuous functions between ω-CPOs, where the algebraic signature includes symbols for composition, pairing, and parametric fixpoints. Several different equational axiomatizations have been considered in the literature, all of which require substantial effort to parse and understand. By allowing quasi-equations, simpler axiomatizations can be found. Many examples of iteration theories involve functions on posets, so it is a natural question to look for complete axiomatizations of the valid inequalities over classes of structures that are of interest, e.g., structures of ω-continuous functions over ω-CPOs. One such universal Horn axiomatization is given in [9]. This axiomatization includes two inequalities and one implication for the $\dagger$ operation, which are both intuitive and easy to memorize. We note that in the work on iteration theories, the issue of how (non)determinism interacts with pairs, which is central in the present work, is not handled at all.

Of particular relevance to the relationship between iteration theories and Kleene algebra are the works on the so-called "matrix iteration theories" [4, 5, 7]. They are cartesian categories in which the homsets are commutative monoids with respect to an operation $+$, which distributes over composition. This also induces cocartesian structure and allows an easy translation between the dagger (parametric fixpoint) operation and Kleene star. However, this translation is not sound for the classes of structures we consider. In particular, the $+$ symbol cannot be interpreted as nondeterministic choice when $\langle \cdot, \cdot \rangle$ is interpreted as pairing: the axioms imply the property $\langle a, \bot \rangle + \langle \bot, b \rangle = \langle a, b \rangle$, which is not meaningful for programs. The translation of the dagger operation in the language of KA that we give here is crucially different, and is in fact sound for the class of matrix iteration theories as well.

There is a connection between our characterizations of nondeterministic structure and the work of Plotkin and Power on algebraic operations (see e.g. [24] for an overview). The theory of algebraic operations provides a uniform semantics of computational effects by considering primitive operations of type $f_X : (PX)^n \to PX$ that are the source of effects. The binary union operation $\mathsf{u}_X : PX \times PX \to PX$ that we consider is an algebraic operation in this sense. Our purpose here, however, is very different: we axiomatize u_X and establish equivalence with other axiomatizations of nondeterministic structure that do not fit in the framework of algebraic operations, e.g., with $\mathsf{d}_X : X \times X \to PX$. In particular, for the equivalence results that we develop here, there does not seem to be any known fact about algebraic operations that can be invoked to simplify our proofs.

Our work here builds directly upon the existing work on Kleene algebra [8, 17–20, 22]. The crucial axioms for the iteration operation * are taken from [18]. The system of KA we present is a typed Kleene algebra in the sense of [21] extended with products that satisfy weaker axioms than those of categorical products.

9 Conclusion and Future Work

In the present work we reconcile the notions of iteration captured by the star operation * of KA and the dagger operation $\dagger$ of IT. We present and investigate a system of typed KA with

products, in which the notion of a deterministic program turns out to be of importance. We work in the framework of cartesian categories combined with commutative strong monads to treat (angelic) nondeterminism. We have adapted an axiomatization of commutative strong monads from the work of Kock [13, 15] to our setting. We have described three equivalent ways, in the presence of cartesian structure, of capturing nondeterminism. We have identified two concrete monads, the monad of lowersets of pointed posets and the monad of ideals of ω-CPOs, as models. The main technical result of our paper is a translation of $\dagger$ in terms of * that gives an embedding of the (in)equational theory of $\dagger$ in KA.

The present work has been a first step in presenting a higher-order system of typed Kleene algebra. We would like to investigate what properties of recursion can be captured in such a higher-order system and how this would relate to the investigations of [6].

References

1 Samson Abramsky and Achim Jung. Domain theory, 1994.
2 J. Adámek, S. Milius, and J. Velebil. Elgot algebras. *Log. Meth. Comp. Sci.*, 2:1–31, 2006.
3 J. Adámek, S. Milius, and J. Velebil. Elgot theories: a new perspective of the equational properties of iteration. *Math. Structures Comput. Sci.*, 21(2):417–480, 2011.
4 S. L. Bloom and Z. Ésik. *Iteration Theories*. Springer, 1993.
5 S. L. Bloom and Z. Ésik. Matrix and matricial iteration theories, part I. *Journal of Computer and System Sciences*, 46(3):381 – 408, 1993.
6 S. L. Bloom and Z. Ésik. Fixed-point operations on ccc's. part I. *TCS*, 155(1):1–38, 1996.
7 S. L. Bloom and Z. Ésik. The equational logic of fixed points. *TCS*, 179(1–2):1–60, 1997.
8 John Horton Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.
9 Z. Ésik. Completeness of Park induction. *Theor. Comput. Sci.*, 177(1):217–283, 1997.
10 S. Ginali. Regular trees and the free iterative theory. *JCSS*, 18:228–242, 1979.
11 Sergey Goncharov. *Kleene Monads*. PhD thesis, Universität Bremen, 2010.
12 S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1956.
13 A. Kock. Monads on symmetric monoidal closed categories. *Ar. der Math.*, 21:1–10, 1970.
14 A. Kock. Bilinearity and cartesian closed monads. *Mathematica Scand.*, 29:161–174, 1971.
15 A. Kock. Strong functors and monoidal monads. *Archiv der Math.*, 23:113–120, 1972.
16 L. Kot and D. Kozen. Kleene algebra and bytecode verification. *ENTCS*, 141(1):221–236, 2005.
17 D. Kozen. On Kleene algebras and closed semirings. MFCS '90, pages 26–47, 1990.
18 D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Proceedings of 6th Annual IEEE Symp. on Logic in Comp. Sci.*, pages 214–225, 1991.
19 D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.
20 D. Kozen. Kleene algebra with tests. *ACM Trans. Prog. Lang. Syst.*, 19(3):427–443, 1997.
21 D. Kozen. Typed Kleene algebra. Technical report, Cornell University, 1998.
22 D. Kozen. On Hoare logic and Kleene algebra with tests. *TOCL*, 1(1):60–76, 2000.
23 D. Kozen and M.-C. Patron. Certification of compiler optimizations using Kleene algebra with tests. In *Proceed. of the First Intern. Conf. on Comput. Logic*, pages 568–582, 2000.
24 G. Plotkin and J. Power. Computational effects and operations. *ENTCS*, 73:149–163, 2004.
25 Gordon Plotkin. Domains, 1983. Pisa notes on domain theory.
26 A. Simpson and G. Plotkin. Complete axioms for categorical fixed-point operators. In *Proceedings of 15th Annual IEEE Symp. on Logic in Comp. Sci.*, pages 30 –41, 2000.

$$\frac{f : X \rightharpoonup Y \times Z \qquad f;\varpi_1 : X \to Y}{\langle\!\langle f;\varpi_1, f;\varpi_2\rangle\!\rangle = f : X \rightharpoonup Y \times Z} \qquad \frac{f : X \rightharpoonup Y \times Z \qquad f;\varpi_2 : X \to Z}{\langle\!\langle f;\varpi_1, f;\varpi_2\rangle\!\rangle = f : X \rightharpoonup Y \times Z}$$

Table 4 Additional rules for the interaction between (non)determinism and Kleisli products.

A Interaction between tensorial strength and iteration

In [11] an axiom is considered that relates the iteration operation μ with the tensorial strength of the nondeterministic monad:

$$\mu f.\big((\mathrm{id}_X \times p); t_{X,Y} + f; (\mathrm{id}_X \times q); t_{X,Y}\big) = \big(\mathrm{id}_X \times \mu f.(p + f; q)\big); t_{X,Y},$$

where $p : X \to PY$ and $q : Y \to PY$ (composition ; binds tighter than Kleisli composition ;, which in turn binds tighter than +). We rewrite the above axiom in terms of ψ using the property $t_{X,Y} = (\eta_X \times \mathrm{id}); \psi_{X,Y}$:

$$\mu f.\big((\eta_X \times p); \psi_{X,Y} + f; (\eta_X \times q); \psi_{X,Y}\big) = \big(\eta_X \times \mu f.(p + f; q)\big); \psi_{X,Y}.$$

Recall the definition of the Kleisli product functor: $f \otimes g = (f \times g); \psi$. So, we can rewrite the above equation as

$$\mu f.\big((\eta_X \otimes p) + f; (\eta_X \otimes q)\big) = \eta_X \otimes \mu f.(p + f; q).$$

The expression $\mu f.\phi(f)$ is meant to denote the least fixpoint of the map $f \mapsto \phi(f)$. So, we think intuitively of the expression $\mu f.(p + f; q)$ as denoting the supremum of the chain $p \leq p + p; q \leq p + p; q + p; q; q \leq \cdots$, which in our language would be represented by $p; q^*$. Using this correspondence, the axiom finally becomes

$$(\eta_X \otimes p); (\eta_X \otimes q)^* = \eta_X \otimes p; q^*$$

in the language of KA. We will identify two very fundamental axioms that together with the KA axioms allow us to prove the above equation.

Before we state the result, we give some intuition for the extra axioms we will assume. In the categorical models we have been investigating the uniqueness property $\langle\!\langle h;\varpi_1, h;\varpi_2\rangle\!\rangle = h$ holds whenever the arrow $h : X \rightharpoonup Y \times Z$ is deterministic, but does not hold for arbitrary h. The reason for this is illustrated by the following example:

$$S = \{(a, c), (b, d)\} : \mathbb{1} \to P(X \times Y) \qquad S;\varpi_1 = \{a, b\} : \mathbb{1} \to PX \qquad S;\varpi_2 = \{c, d\} : \mathbb{1} \to PY$$

and $\langle\!\langle S;\varpi_1, S;\varpi_2\rangle\!\rangle = \{(a, c), (a, d), (b, c), (b, d)\} \neq S$. However, if we change the example so that $S;\varpi_1$ is deterministic, then the uniqueness axiom is satisfied.

$$S = \{(a, c), (a, d)\} : \mathbb{1} \to P(X \times Y) \qquad S;\varpi_1 = \{a\} : \mathbb{1} \to PX \qquad S;\varpi_2 = \{c, d\} : \mathbb{1} \to PY$$

Notice that $S;\varpi_1 = \{a\}$ is deterministic and $\langle\!\langle S;\varpi_1, S;\varpi_2\rangle\!\rangle = S$, even though S is not deterministic. This example motivates the rules of Table 4.

▶ **Proposition 18.** *Every nondeterministic monad with iteration that additionally satisfies the two rules of Table 4 also satisfies the equation*

$$(\eta_X \otimes p); (\eta_X \otimes q)^* = \eta_X \otimes p; q^*.$$

Proof. Call L the left-hand side and R the right-hand side of the equation we have to prove. The inequality $L \leq R$ is easily seen to be provable from the axioms for *. It suffices to show that $\eta_X \otimes p \leq \eta_X \otimes p; q^*$, which holds because $p \leq p; q^*$, and also that

$$\begin{aligned}(\eta_X \otimes p; q^*); (\eta_X \otimes q) &= \eta_X; \eta_X \otimes p; q^*; q && [\otimes \text{ functor}]\\ &\leq \eta_X \otimes p; q^*. && [\text{star axiom}]\end{aligned}$$

We will see now that the inequality $R \leq L$ is also provable from the star axioms, by making use of the rules given in Table 4. First, we claim that

$$L; \varpi_1 = (\eta_X \otimes p); (\eta_X \otimes q)^*; \varpi_1 = \varpi_1.$$

Since $(\eta_X \otimes q); \varpi_1 = \varpi_1; \eta_X = \varpi_1$, we have that $(\eta_X \otimes q); \varpi_1 \leq \varpi_1$ and hence $(\eta_X \otimes q)^*; \varpi_1 \leq \varpi_1$. It follows that $L; \varpi_1 \leq (\eta_X \otimes p); \varpi_1 = \varpi_1$. Moreover,

$$\varpi_1 = (\eta_X \otimes p); \varpi_1 \leq (\eta_X \otimes p); (\eta_X \otimes q)^*; \varpi_1 = L; \varpi_1.$$

We have thus shown that $L; \varpi_1 = \varpi_1$ is deterministic. So, the first rule of Table 4 gives us that

$$(\eta_X \otimes p); (\eta_X \otimes q)^* = \langle\!\langle \varpi_1, (\eta_X \otimes p); (\eta_X \otimes q)^*; \varpi_2 \rangle\!\rangle.$$

We also have by definition of $\otimes$ that $\eta_X \otimes p; q^* = \langle\!\langle \varpi_1, \varpi_2; p; q^* \rangle\!\rangle$. It suffices to show that

$$\begin{aligned}\varpi_2; p; q^* \leq (\eta_X \otimes p); (\eta_X \otimes q)^*; \varpi_2 &\Longleftarrow (\eta_X \otimes p); \varpi_2; q^* \leq (\eta_X \otimes p); (\eta_X \otimes q)^*; \varpi_2\\ &\Longleftarrow \varpi_2; q^* \leq (\eta_X \otimes q)^*; \varpi_2,\end{aligned}$$

which is implied by $\varpi_2 \leq (\eta_X \otimes q)^*; \varpi_2$ (it clearly holds) and

$$(\eta_X \otimes q)^*; \varpi_2; q = (\eta_X \otimes q)^*; (\eta_X \otimes q); \varpi_2 \leq (\eta_X \otimes q)^*; \varpi_2. \qquad \blacktriangleleft$$

So, the above proposition says, somewhat informally, that the interaction between iteration and tensorial strength that is stipulated in [11] is a consequence of the properties of * and of two fundamental axioms that involve only determinism judgments and Kleisli products. The axioms of Table 4 are sound for the models we consider here, as well as for the powerset monad $\wp$ over **Set**, which is a model of the axioms considered in [11]. We have not included these axioms in the definition of a nondeterministic strong monad with iteration, because they are not necessary for our investigations.