Mathematical Structures in Computer Science (2022), **32**, pp. 472–510
doi:10.1017/S0960129522000330

CAMBRIDGE UNIVERSITY PRESS

PAPER

Weakest preconditions in fibrations

Alejandro Aguirre[1], Shin-ya Katsumata[2,*] and Satoshi Kura[3]

[1]Aarhus University, Åbogade 34, 8200 Aarhus N, Denmark, [2]National Institute of Informatics, Chiyoda City 100-8361, Japan and [3]JSPS Research Fellow and National Institute of Informatics, Chiyoda City 100-8361, Japan
[*]Corresponding author. Email: s-katsumata@nii.ac.jp

(Received 21 December 2020; revised 14 September 2022; accepted 14 September 2022;
first published online 28 October 2022)

Abstract

Weakest precondition transformers are useful tools in program verification. One of their key properties is composability, that is, the weakest precondition predicate transformer (wppt for short) associated to program $f; g$ should be equal to the composition of the wppts associated to f and g. In this paper, we study the categorical structure behind wppts from a fibrational point of view. We characterize the wppts that satisfy composability as the ones constructed from the Cartesian lifting of a monad. We moreover show that Cartesian liftings of monads along lax slice categories bijectively correspond to Eilenberg–Moore monotone algebras. We then instantiate our techniques by deriving wppts for commonplace effects such as the maybe monad, the nonempty powerset monad, the counter monad, or the distribution monad. We also show how to combine them to derive the wppts appearing in the literature of verification of probabilistic programs.

Keywords: Weakest precondition; strongest postcondition; Hoare logic; monad; fibration

1. Introduction

Dijkstra's *weakest precondition predicate transformer* (*wppt* for short) (Dijkstra 1975) computes, for an imperative program f and a predicate ψ on memory configurations, the *weakest* predicate $\mathrm{wp}(f, \psi)$ on memory configurations such that, for any memory configuration x satisfying $\mathrm{wp}(f, \psi)$, the execution of f from x yields a memory configuration satisfying ψ. When the imperative program f always terminates and only updates memory configurations deterministically, the behavior of f can be modeled as an endofunction $[\![f]\!]$ over the set M of memory configurations, and $\mathrm{wp}(f, \psi)$ is defined so that $[\![\mathrm{wp}(f, \psi)]\!]$ becomes the *inverse image* of $[\![\psi]\!] \subseteq M$ along $[\![f]\!]$. One of the key properties of predicate transformers that make them suitable for program verification is *composability*: the weakest precondition $\mathrm{wp}(f; g, \psi)$ of a composite program $f; g$ is equivalent to the composition of the weakest preconditions of its components, that is, $\mathrm{wp}(f, \mathrm{wp}(g, \psi))$. This allows us to inductively compute the weakest precondition predicate transformer of programs.

Following Dijkstra's seminal work, wppts and their variants have been applied to program verification in different manners, such as computing expected values over outputs of probabilistic programs McIver and Morgan (2005), estimating runtime Kaminski et al. (2016), or estimating tail bounds of rewards over control-flow graphs Kura et al. (2019). The motivation of this paper is to identify a mathematical structure behind these variations of wppt-like semantics. Toward this goal, in this paper we set out to study wppts and their composability using *fibrations*. Roughly speaking, a fibration is a functor $p : \mathbb{P} \to \mathbb{C}$ that exposes a relationship between a category $\mathbb{P}$ of predicates and an underlying category $\mathbb{C}$ modeling computation - when $pP = X$ holds, we regard $P \in \mathbb{P}$ as a predicate over $X \in \mathbb{C}$; this viewpoint is shared with the categorical study of *logical relations (Hermida 1993)* and *refinement types* (Melliès and Zeilberger 2015). Fibrations are especially suited to interpret predicate transformers and their composability thanks to the *Cartesian lifting property*, which allows us to take the "inverse image" of $Q \in \mathbb{P}$ along a morphism $f : X \to Y$ in $\mathbb{C}$, resulting an object $f^*Q \in \mathbb{P}$ such that $p(f^*Q) = X$. This is a categorical abstraction of the inverse image operation.

The main challenge we address in this paper is to develop a categorical theory of weakest precondition predicate transformer under the presence of computational effects modeled by monads.[1] The main technical vehicle of this development is the *lifting* of a monad T along a fibration $p : \mathbb{P} \to \mathbb{C}$ – it is a monad $\dot{\mathsf{T}}$ on $\mathbb{P}$ such that p strictly preserves the monad structure to T. We then regard a morphism $f : P \to \dot{\mathsf{T}}Q$ in $\mathbb{P}$ as a Hoare triple $P\{f\}Q$. This induces a natural wppt $\mathrm{wp}(f, P) \triangleq f^*(\dot{T}P)$ that characterizes the Hoare triple, but it does *not* satisfy composability in general. This raises the question of when monadic computations do induce compositional weakest precondition predicate transformers. In this paper, we answer to this question by introducing the property called *Cartesian-ness* on monad liftings, and contribute to its understanding as follows:

(1) We show that the wppt defined by a monad lifting is composable if and only if the monad lifting is Cartesian. This result establishes the tight connection between the composability of wppts in monadic setting and the Cartesian-ness of liftings. This result tells us what kind of formal structures are needed when giving compositional wppt-like semantics to imperative programming languages.

(2) We relate *strongest postcondition predicate transformers* (*sppt*s for short) as left adjoints to wppts, and discuss when they are available and composable.

(3) We study Cartesian liftings of monads along domain fibrations (see Section 4.2) from lax slice categories, which have as objects morphisms from an object of the base category to an object Ω of generalized truth values. For this class of fibrations, there is a bijective correspondence between Cartesian liftings of a monad T and Eilenberg-Moore monotone algebras of T; this is exhibited through a 2-categorical embedding of the 2-category of categories with ordered objects to the 2-category of fibrations. The correspondence result simplifies the task of exploring Cartesian liftings of monads along such fibrations and makes it possible to enumerate all Cartesian liftings of some monads. We show examples of Cartesian liftings of the following monads: the maybe monad, the nonempty powerset monad, the counting monad, the distribution monad, and the indexed distribution monad and some combinations of them. The wppts derived from Cartesian liftings of monads coincide with *weakest pre-expectation* (McIver and Morgan 2005) and the *higher moment transformer* (Kura et al. 2019).

(4) Computational effects are often modeled by monads composed via *distributive laws*. We extend the correspondence given in (3) to the one between Cartesian liftings of composite monads and pairs of Eilenberg-Moore monotone algebras satisfying an extra coherence law given in Beck (1969), Manes and Mulry (2007). This correspondence provides a modular method to compute Cartesian liftings of composite monads.

(5) To compute the wppts of the programs containing effectful commands (such as probabilistic choice and counting), we study the interaction between Plotkin and Power's *algebraic operations* (Plotkin and Power 2001), which are a categorical models of effectful commands, and wppts studied in (3).

(6) Apart from domain fibrations, we illustrate a few examples of Cartesian liftings of monads along relational fibrations. They are outside of the framework presented in Hasuo (2015) and Hino et al. (2016). These examples are derived from a general construction of the *change-of-base* of fibrations with (Cartesian) monad liftings.

(7) We demonstrate that categorical wppts can be used to *derive* predicate transformer semantics. The derivation takes two steps: (1) we compose a categorical wppt wp and the standard monadic semantics $[\![P]\!] : M \to TM$ of an imperative language and then (2) we represent the composite as an inductive definition over programs. The derived inductive definition gives a predicate-transformer semantics of the language. We illustrate this story by deriving Kaminski et al.'s *expected runtime transformer* (Kaminski et al. 2016) for the loop-free fragment of a probabilistic programming language.

Differences with respect to conference version. This paper is an extension of an MFPS conference paper of the same title (Aguirre and Katsumata 2020). The increments with respect to the conference paper are the following:

- We generalize our results from posetal fibrations to $\mathbb{K}$-fibrations, for an arbitrary subcategory $\mathbb{K}$ of the category of posets.
- We give a 2-categorical perspective of the lax slice construction in Section 4.2.
- We study liftings of strong monads and give sufficient and necessary conditions for the liftings to be strong.
- We identify a general principle behind our presentation of liftings along relational fibrations and show that (Cartesian) liftings of monads can be pulled back along a change-of-base with monad opfunctors (Street 1972, Section 4).
- We study continuity properties of weakest precondition transformers and apply it to the computation of the weakest precondition of while loops.
- We provide more extensive explanations and details of the proofs.

2. Preliminaries

Composition of functors and whiskering (Whiskering https://ncatlab.org/nlab/show/whiskering) of functors and natural transformations are denoted by juxtaposition. The vertical and horizontal compositions of natural transformations are denoted by $\star$ and $*$, respectively. We write **Pos** for the category of posets and monotone functions between them. The forgetful functor from **Pos** to **Set** is denoted by U. Subcategories of **Pos** are ranged over $\mathbb{K}$. A morphism $f : X \to Y$ in $\mathbb{K}$ is called a *right adjoint* if there is a monotone function $g \in \mathbf{Pos}(Y, X)$ such that $g(y) \le x \iff y \le f(x)$ holds for any $x \in X, y \in Y$.

2.1 Monads and Distributive Laws

We briefly recall here some definitions about monads. For a more detailed account see e.g. MacLane (1998, Section VI). Let $\mathbb{C}$ be a category. A *monad* is a triple (T, η, μ) of a functor $\mathsf{T} \colon \mathbb{C} \to \mathbb{C}$ and two natural transformations $\eta \colon \mathrm{Id} \Rightarrow \mathsf{T}$ (the *unit*) and $\mu \colon \mathsf{T}^2 \Rightarrow \mathsf{T}$ (the *multiplication*) such that $\mu \star \eta\mathsf{T} = \mathrm{id}_\mathsf{T} = \mu \star \mathsf{T}\eta$ and $\mu \star \mu\mathsf{T} = \mu \star \mathsf{T}\mu$.

The *Kleisli lifting* of $f : X \to \mathsf{T}Y$ is a morphism $f^\# \colon \mathsf{T}X \to \mathsf{T}Y$ defined as $f^\# \triangleq \mu_Y \circ \mathsf{T}f$. A monad may be equivalently given by a triple $(\mathsf{T}, \eta, (\_)^\#)$, replacing the multiplication with the Kleisli lifting $(\_)^\# : \mathbb{C}(X, \mathsf{T}Y) \to \mathbb{C}(\mathsf{T}X, \mathsf{T}Y)$; see Moggi (1991).

The Kleisli category $\mathbb{C}_\mathsf{T}$ of a monad (T, η, μ) is a category whose objects are the objects of $\mathbb{C}$ and whose homsets are defined by $\mathbb{C}_\mathsf{T}(X, Y) \triangleq \mathbb{C}(X, \mathsf{T}Y)$. The composition of morphisms in $\mathbb{C}_\mathsf{T}$ is denoted by $\bullet$ and is defined from the composition $\circ$ in $\mathbb{C}$ as $g \bullet f \triangleq g^\# \circ f$. Following Moggi (1991), we regard morphisms in $\mathbb{C}_\mathsf{T}$ as abstract representations of programs causing computational effects. There exists an adjunction $L \dashv R \colon \mathbb{C} \to \mathbb{C}_\mathsf{T}$ such that $R \circ L = T$, called the *Kleisli resolution* of T.

An Eilenberg–Moore (EM for short) algebra of a monad (T, η, μ) is a pair (X, o) of an object $X \in \mathbb{C}$ and a morphism $o\colon \mathsf{T}X \to X$ such that $o \circ \eta_X = \mathrm{id}_X$ and $o \circ To = o \circ \mu_X$.

Let $(\mathsf{T}, \eta^{\mathsf{T}}, \mu^{\mathsf{T}})$ and $(\mathsf{S}, \eta^{\mathsf{S}}, \mu^{\mathsf{S}})$ be monads on $\mathbb{C}$. A *distributive law* (Beck 1969) of S over T is a natural transformation $\alpha : \mathsf{ST} \to \mathsf{TS}$ such that

$$\eta^{\mathsf{T}}\mathsf{S} = \alpha \star \mathsf{S}\eta^{\mathsf{T}} \qquad \alpha \star \mathsf{S}\mu^{\mathsf{T}} = \mu^{\mathsf{T}}\mathsf{S} \star \mathsf{T}\alpha \star \alpha\mathsf{T} \qquad \mathsf{T}\eta^{\mathsf{S}} = \alpha \star \eta^{\mathsf{S}}\mathsf{T} \qquad \alpha \star \mu^{\mathsf{S}}\mathsf{T} = \mathsf{T}\mu^{\mathsf{S}} \star \alpha\mathsf{S} \star \mathsf{S}\alpha.$$

The distributive law yields the monad $\big(\mathsf{TS}, \eta^{\mathsf{T}} * \eta^{\mathsf{S}}, (\mu^{\mathsf{T}} * \mu^{\mathsf{S}}) \star (\mathsf{T}\alpha\mathsf{S})\big)$ over the composite functor. We denote this monad by $\mathsf{T} \circ_\alpha \mathsf{S}$.

2.2 Fibrations

Dijkstra's weakest precondition predicate transformer manipulates *predicates* over memory configurations. In the abstract study of predicate transformers, however, it is convenient to extend the concept of predicates so that they can be defined over arbitrary objects on a category. For this abstract and general treatment of predicates, we leverage the notions of *fibered category theory* (see e.g. Jacobs 1999). Before introducing these concepts, we give an informal account of them.

Given a category $\mathbb{C}$ (with objects ranging over X, Y, Z and morphisms ranging over f, g, h) we aim to define predicates over its objects by introducing a category $\mathbb{P}$ (with objects ranging over P, Q, R and morphisms ranging over dotted letters $\dot{f}, \dot{g}, \dot{h}$) and a functor $p : \mathbb{P} \to \mathbb{C}$. We understand this general situation as follows:

- An object in $\mathbb{P}$ is a predicate over some object in $\mathbb{C}$, which is recorded by the functor $p : \mathbb{P} \to \mathbb{C}$. That is, $pP = X$ means that $P \in \mathbb{P}$ is a predicate over $X \in \mathbb{C}$.
- A morphism $\dot{f} : P \to Q$ in $\mathbb{P}$ is a *witness* of the fact that the underlying morphism $p\dot{f} : pP \to pQ$ in $\mathbb{C}$ "preserves" these predicates, that is, $p\dot{f}$ maps elements satisfying the predicate P to those satisfying the predicate Q. In particular, if $\dot{f} : P \to Q$ satisfies $p\dot{f} = \mathrm{id}_X$, then $\dot{f}$ witnesses that P implies Q.

The inverse image of an object $X \in \mathbb{C}$ under the functor $p : \mathbb{P} \to \mathbb{C}$ forms a category denoted by $\mathbb{P}_X$, known as the *fiber category* above X. Formally, an object of $\mathbb{P}_X$ is a $\mathbb{P}$-object P such that $pP = X$, and a morphism from P to Q in $\mathbb{P}_X$ is a morphism $\dot{f} : P \to Q$ in $\mathbb{P}$ such that $p\dot{f} = \mathrm{id}_X$. Intuitively, objects of this category are predicates over X, and morphisms of this category represent implication relations between them.

When defining weakest precondition predicate transformers, the strength of predicates is compared by an order relation, so we focus our attention on *posetal fibrations*. Recall that the weakest precondition predicate transformer collects *all* the memory configurations that entail the postcondition. Set-theoretically, this operation is called *inverse image*, and *fibrations* offer a more general and flexible treatment of the inverse image operation. Roughly speaking, a fibration is a functor $p : \mathbb{P} \to \mathbb{C}$ such that for any morphism $f : X \to Y$ in $\mathbb{C}$ and $P \in \mathbb{P}_Y$, we can find the *inverse image* $f^*P \in \mathbb{P}_X$ of P along f. The formal definition of a posetal fibration follows.

- For objects $P, Q \in \mathbb{P}$ and a morphism $f : pP \to pQ$ in $\mathbb{C}$, we define the set $\mathbb{P}_f(P, Q)$ of morphisms in $\mathbb{P}$ above f by $\mathbb{P}_f(P, Q) \triangleq \{\dot{f} \in \mathbb{P}(P, Q) \mid p\dot{f} = f\}$.
- A morphism $\dot{f} : P \to Q$ in $\mathbb{P}$ is *Cartesian* if for any $R \in \mathbb{P}$ and morphism $h : pR \to pP$ in $\mathbb{C}$, the postcomposition of $\dot{f}$, regarded as a function of type $\mathbb{P}_h(R, P) \to \mathbb{P}_{p\dot{f} \circ h}(R, Q)$, is a bijection. This is the universal property of Cartesian morphisms. A Cartesian morphism $\dot{f} : P \to Q$ in $\mathbb{P}$ abstractly represents the situation that P is an inverse image of Q along $p\dot{f}$.
- A functor $p : \mathbb{P} \to \mathbb{C}$ is a *fibration* if for any morphism $f : X \to Y$ in $\mathbb{C}$ and $Q \in \mathbb{P}_Y$, there is an object $P \in \mathbb{P}_X$ and a Cartesian morphism $\dot{f} : P \to Q$ in $\mathbb{P}_f$ called the *Cartesian lifting* of f with Q.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

- A fibration $p : \mathbb{P} \to \mathbb{C}$ is *posetal* if each fiber category $\mathbb{P}_X$ is a poset. In any posetal fibration, the Cartesian lifting of a morphism $f : X \to Y$ in $\mathbb{C}$ with $Q \in \mathbb{P}_Y$ uniquely exists; we therefore denote it by $\overline{f}Q$ and its domain by f^*Q. We note that posetal fibrations are faithful. Therefore, a morphism $\dot{f} : P \to Q$ in $\mathbb{P}$ is Cartesian if and only if $P = (p\dot{f})^*Q$.
- In a posetal fibration $p : \mathbb{P} \to \mathbb{C}$, the assignment $Q \mapsto f^*Q \in \mathbb{P}_P$ for a morphism $f : X \to Y$ in $\mathbb{C}$ extends to a monotone function of type $\mathbb{P}_Y \to \mathbb{P}_X$, which we call the *reindexing functor* (we use the word functor for the compatibility with the existing terminology). The assignment $f \mapsto f^*$ furthermore satisfies $(\mathrm{id}_X)^* = \mathrm{id}_{\mathbb{P}_X}$ and $(g \circ f)^* = f^* g^*$.
- Let $\mathbb{K}$ be a subcategory of **Pos**. A functor $p : \mathbb{P} \to \mathbb{C}$ is a $\mathbb{K}$-*fibration* if each fiber category $\mathbb{P}_X$ belongs to $\mathbb{K}$, and each reindexing functor f^* belongs to $\mathbb{K}$ as a morphism.

A convenient way to obtain $\mathbb{K}$-fibrations is by the *Grothendieck construction* (in fact every $\mathbb{K}$-fibration is isomorphic to one obtained in this way). Let $F : \mathbb{C}^{op} \to \mathbb{K}$ be a functor. The *Grothendieck construction* applied to F yields a category $\int F$ defined by the data below:

- An object is a pair (X, x) of an object $X \in \mathbb{C}$ and an element $x \in FX$.
- A morphism from (X, x) to (Y, y) is a morphism $f : X \to Y$ in $\mathbb{C}$ such that $x \leq Ff(y)$ in the poset FX belonging to $\mathbb{K}$.

The forgetful functor $G_F : \int F \to \mathbb{C}$ defined by $G_F(X, x) \triangleq X$ and $G_F(f) \triangleq f$ is a $\mathbb{K}$-fibration; see e.g. Jacobs (1999).

Toward defining the 2-category of $\mathbb{K}$-fibrations, we introduce a few concepts. Let $p : \mathbb{P} \to \mathbb{C}$ and $q : \mathbb{Q} \to \mathbb{D}$ be $\mathbb{K}$-fibrations.

- A $\mathbb{K}$-*functor* is a pair of functors $(F : \mathbb{C} \to \mathbb{D}, \dot{F} : \mathbb{P} \to \mathbb{Q})$ such that $Fp = q\dot{F}$, and the restriction of $\dot{F}$ to each fiber, written $\dot{F}_X : \mathbb{P}_X \to \mathbb{Q}_{FX}$, belongs to $\mathbb{K}$. It is called *fibered* if $\dot{F}$ preserves Cartesian morphisms; this requirement is equivalent to requiring that the equality $\dot{F}(f^*P) = (Ff)^*(\dot{F}P)$ holds for any object $P \in \mathbb{P}$ and morphism $f : X \to pP$ in $\mathbb{C}$.
- Let $(F, \dot{F}), (G, \dot{G}) : p \to q$ be $\mathbb{K}$-functors. A *natural transformation* from the former to the latter is a pair of natural transformations $\alpha : F \to G$ and $\dot{\alpha} : \dot{F} \to \dot{G}$ such that $\alpha p = q\dot{\alpha}$. It is called *Cartesian* if $\dot{\alpha}_P$ is a Cartesian morphism above α_{pP} for each $P \in \mathbb{P}$.

Definition 1. *We define a chain of 2-categories $\mathbb{K}$-$\mathbf{Fib}_0 \supset \mathbb{K}$-$\mathbf{Fib} \supset \mathbb{K}$-$\mathbf{Fib}_c$ by the following table.*

2-Category	*0-Cell*	*1-Cell*	*2-Cell*
$\mathbb{K}$-$\mathbf{Fib}_0$	*$\mathbb{K}$-fibration*	*$\mathbb{K}$-functor*	*natural transformation*
$\mathbb{K}$-$\mathbf{Fib}$	*$\mathbb{K}$-fibration*	*fibered $\mathbb{K}$-functor*	*natural transformation*
$\mathbb{K}$-$\mathbf{Fib}_c$	*$\mathbb{K}$-fibration*	*fibered $\mathbb{K}$-functor*	*Cartesian natural transformation*

The following lemma characterizes (Cartesian) natural transformations between (fibered) $\mathbb{K}$-functors.

Lemma 2. *Let $p : \mathbb{P} \to \mathbb{C}$ and $q : \mathbb{Q} \to \mathbb{D}$ be $\mathbb{K}$-fibrations. Then the mapping $(\alpha, \dot{\alpha}) \mapsto \alpha$ gives the following bijections:*

$$\Xi : \mathbb{K}\text{-}\mathbf{Fib}_0(p, q)((F, \dot{F}), (G, \dot{G})) \cong \{\alpha : F \to G \mid \forall P \in \mathbb{P}\,.\, \dot{F}P \leq \alpha_{pP}^*(\dot{G}P)\}, \tag{1}$$

$$\Xi_c : \mathbb{K}\text{-}\mathbf{Fib}_c(p, q)((F, \dot{F}), (G, \dot{G})) \cong \{\alpha : F \to G \mid \forall P \in \mathbb{P}\,.\, \dot{F}P = \alpha_{pP}^*(\dot{G}P)\}. \tag{2}$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

We write Ξ, Ξ_c for the above bijections from left to right, respectively. We will use them later in Section 4.1.

Proof. We show (1); (2) is proved similarly. Let $\alpha : F \to G$ be a natural transformation such that $\dot{F}P \leq \alpha^*_{pP}(\dot{G}P)$. It suffices to exhibit a unique natural transformation $\dot{\alpha} : \dot{F} \to \dot{G}$ such that $\alpha p = q\dot{\alpha}$. Define $\dot{\alpha}_P$ by

$$\dot{\alpha}_P \triangleq \dot{F}P \xrightarrow{\quad\leq\quad} \alpha^*_{pP}(\dot{G}P) \xrightarrow{\overline{\alpha_{pP}}(\dot{G}P)} \dot{G}P\ .$$

It routine to check that $\dot{\alpha}$ is a natural transformation satisfying $\alpha p = q\dot{\alpha}$. Moreover, the faithfulness of q guarantees its uniqueness. $\square$

3. Dijkstra Structures and Weakest Precondition Predicate Transformers

A fibration $p \colon \mathbb{P} \to \mathbb{C}$ allows us to define an abstract form of inverse images for morphisms in $\mathbb{C}$. However, effectful computations are instead modeled as morphisms $f \colon X \to \mathsf{T}Y$ for some monad T, following Moggi (1991). To write logical specifications for such computations, we will lift T to a monad $\dot{\mathsf{T}}$ over $\mathbb{P}$, which will map predicates over Y to predicates over $\mathsf{T}Y$. We will call such a triple $(p \colon \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ a Dijkstra structure.

In this section, we present Dijkstra Structures and show how to use them to define an abstract notion of weakest precondition transformer for effectful computations (Section 3.1). We will then identify the subclass of Dijkstra Structures for which the transformer composes exactly (Section 3.2). We then study how Dijskstra Structures behave under change-of-base (Section 3.3). Then we define a variant of Dijkstra structures for strong monads (Section 3.4), which allows us to prove soundness of frame-like rules. Finally, we study the dual situation of strongest postconditions (Section 3.5).

3.1 Dijkstra Structures

Let $\mathbb{C}$ be a category and T a monad over it. We aim to model the statement that a computation represented as a morphism $f : X \to Y$ in the Kleisli category $\mathbb{C}_{\mathsf{T}}$ satisfies a specification given by a precondition and a postcondition, which are predicates over X and Y, respectively. Since the codomain of f is $\mathsf{T}Y$, we first need to construct a predicate over $\mathsf{T}Y$ from a predicate over Y. For this purpose, we define a lifting of T into a monad $\dot{\mathsf{T}}$ over $\mathbb{P}$.

Definition 3. *Let $p : \mathbb{P} \to \mathbb{C}$ be a $\mathbb{K}$-fibration and (T, η, μ) be a monad on $\mathbb{C}$. A monad $(\dot{\mathsf{T}}, \dot{\eta}, \dot{\mu})$ on $\mathbb{P}$ is called a* $\mathbb{K}$-lifting *of T (along p) if $(\mathsf{T}, \dot{\mathsf{T}})$ is a $\mathbb{K}$-functor from p to p, and $p\dot{\mathsf{T}} = \mathsf{T}p$, $p\dot{\eta} = \eta p$ and $p\dot{\mu} = \mu p$ holds. We say that the $\mathbb{K}$-lifting is* Cartesian *if*

- *$(\mathsf{T}, \dot{\mathsf{T}})$ is a fibered $\mathbb{K}$-functor from p to p, and*
- *$\dot{\eta}_P$ and $\dot{\mu}_P$ are respectively Cartesian above η_{pP} and μ_{pP} for any $P \in \mathbb{P}$. This is equivalent to having that $P = \eta^*_{pP}(\dot{\mathsf{T}}P)$ and $\dot{\mathsf{T}}\dot{\mathsf{T}}P = \mu^*_{pP}\dot{\mathsf{T}}P$ for any object $P \in \mathbb{P}$.*

The concept of monad lifting is not new; it appeared as a semantic counterpart of *logical relations for monads* (Filinski 1996, 2007; Goubault-Larrecq et al. 2008; Katsumata 2005; Katsumata et al. 2018). Hermida considered the comonadic case earlier than these works (Hermida 1993, Chapter 5). The definition of Cartesian lifting of monad makes sense when p is a non-posetal fibration. When $\mathbb{C}$ is a category with pullbacks, a monad T on $\mathbb{C}$ is Cartesian (see e.g. Leinster 2004, Section 4.1) if and only if the evident lifting $\mathsf{T}^{\to} : \mathbb{C}^{\to} \to \mathbb{C}^{\to}$ of T to the arrow category $\mathbb{C}^{\to}$ along the codomain fibration $\mathrm{cod} : \mathbb{C}^{\to} \to \mathbb{C}$ is Cartesian.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

The tuple consisting of a $\mathbb{K}$-fibration p, a monad T, and its $\mathbb{K}$-lifting $\dot{\mathsf{T}}$ along p provides us a setting in which we can define the Hoare triple and the *weakest precondition predicate transformer* (*wppt* for short) associated to it. In this paper, we package these data into one, and call it a *Dijkstra structure*.

Definition 4 (Dijkstra Structure)**.** *For a subcategory* $\mathbb{K}$ *of* **Pos***, a (resp. Cartesian)* $\mathbb{K}$-Dijkstra structure *is a tuple* $(p, \mathsf{T}, \dot{\mathsf{T}})$ *of a* $\mathbb{K}$*-fibration* $p : \mathbb{P} \to \mathbb{C}$*, a monad* T *on* $\mathbb{C}$ *and a (resp. Cartesian)* $\mathbb{K}$*-lifting* $\dot{\mathsf{T}}$ *of* T *along* p*. When* $\mathbb{K} = \mathbf{Pos}$*, we simply call it a Dijkstra structure.*

A (resp. Cartesian) $\mathbb{K}$-Dijkstra structure is precisely a monad in $\mathbb{K}$-$\mathbf{Fib}_0$ (resp. $\mathbb{K}$-$\mathbf{Fib}_c$).

Definition 5. *Let* $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ *be a* $\mathbb{K}$*-Dijkstra structure. Below* X, Y *range over* $\mathbb{C}$*-objects.*

(1) For a morphism $f : X \to \mathsf{T}Y$ *in* $\mathbb{C}$ *and objects* $P \in \mathbb{P}_X, Q \in \mathbb{P}_Y$*, we define the* Hoare triple $P\{f\}Q$ *by*

$$P\{f\}Q \triangleq \exists \dot{f} \in \mathbb{P}(P, \dot{\mathsf{T}}Q) \,.\, p\dot{f} = f. \tag{3}$$

Such $\dot{f}$ *is unique because* p *is faithful.*

(2) For a morphism $f : X \to \mathsf{T}Y$ *in* $\mathbb{C}$*, we define the* weakest precondition predicate transformer $\mathrm{wp}(f, -) \in \mathbb{K}(\mathbb{P}_Y, \mathbb{P}_X)$ *by*

$$\mathrm{wp}(f, Q) \triangleq f^*(\dot{\mathsf{T}}Q). \tag{4}$$

Since p is a fibration, Hoare triples and the wppt are linked by the following equivalence:

$$P \leq \mathrm{wp}(f, Q) \Longleftrightarrow P\{f\}Q.$$

We remark that for each morphism f, $\mathrm{wp}(f, \_)$ is a morphism in $\mathbb{K}$. Therefore, if morphisms in $\mathbb{K}$ are required to preserve order-theoretic structures, so does $\mathrm{wp}(f, \_)$. For instance, when $\mathbb{K}$ is the category $\mathbf{CLat}_\wedge$ of complete lattices and functions preserving all meets, the wppt of a $\mathbf{CLat}_\wedge$-Dijkstra structure preserves arbitrary meets in its second argument: $\mathrm{wp}(f, \bigwedge_i P_i) = \bigwedge_i \mathrm{wp}(f, P_i)$. In the context of program verification, such a wppt is called *conjunctive*.

3.2 Composability of the Weakest Precondition Predicate Transformers

The Hoare triple and the wppt in our categorical setting are more general than the standard ones since we can supply any $\mathbb{P}$-object and $\mathbb{C}_\mathsf{T}$-morphism to the wppt. This liberation allows us to relate the *composability* of the wppt and the *Cartesian-ness* of $\dot{\mathsf{T}}$. Since wp takes a Kleisli morphism as an argument, the composability should be discussed with respect to the Kleisli composition.

Theorem 6. *Let* $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ *be a* $\mathbb{K}$*-Dijkstra structure. We have inequalities*

$$\mathrm{wp}(\eta_{pP}, P) \geq P, \quad \mathrm{wp}(f \bullet g, P) \geq \mathrm{wp}(g, \mathrm{wp}(f, P)) \tag{5}$$

for any f, g, P *of appropriate type; here,* η *and* $\bullet$ *are the unit and the Kleisli composition of* T*, respectively. Moreover, inequalties in* (5) *become equalities if and only if the* $\mathbb{K}$*-Dijkstra structure is Cartesian.*

Proof. Let $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ be a $\mathbb{K}$-Dijkstra structure. Then the unit $\dot{\eta}_P : P \to \dot{\mathsf{T}}P$ of $\dot{\mathsf{T}}$ is above η_{pP}. Therefore

$$\mathrm{wp}(\eta_{pP}, P) = \eta^*_{pP}(\dot{\mathsf{T}}P) \geq P.$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

If $\dot\eta_P$ is Cartesian, this inequality becomes an equality. Next, the multiplication $\dot\mu_P : \dot{\mathsf{T}}\dot{\mathsf{T}}P \to \dot{\mathsf{T}}P$ of $\dot{\mathsf{T}}$ is above μ_{pP}. Therefore $\dot{\mathsf{T}}\dot{\mathsf{T}}P \le \mu^*_{pP}\dot{\mathsf{T}}P$ holds; this becomes an equality if $\dot\mu_P$ is Cartesian. Moreover, for any object $X \in \mathbb{C}, P \in \mathbb{P}$ and morphism $f : X \to pP$ in $\mathbb{C}$, we have $\dot{\mathsf{T}}(f^*P) \le (\mathsf{T}f)^*(\dot{\mathsf{T}}P)$; this becomes an equality if $\dot{\mathsf{T}}$ is fibered. From these, we obtain the following inequality, which becomes an equality if $\dot\mu_P$ is Cartesian and $\dot{\mathsf{T}}$ is fibered.

$$\begin{aligned}(f \bullet g)^*(\dot{\mathsf{T}}P) &= (\mu \circ \mathsf{T}f \circ g)^*(\dot{\mathsf{T}}P) = g^*((\mathsf{T}f)^*(\mu^*(\dot{\mathsf{T}}P))) \\ &\ge g^*((\mathsf{T}f)^*(\dot{\mathsf{T}}\dot{\mathsf{T}}P)) \ge g^*(\dot{\mathsf{T}}(f^*(\dot{\mathsf{T}}P)) = \mathrm{wp}(g, \mathrm{wp}(f, P)).\end{aligned}$$

We have thus proved inequalities (5), and if the assumed $\mathbb{K}$-Dijkstra structure is Cartesian, these inequalities become equalities.

We conversely assume

$$P = \mathrm{wp}(\eta_{pP}, P) = \eta^*_{pP}(\dot{\mathsf{T}}P), \tag{6}$$

$$g^*\dot{\mathsf{T}}(f^*\dot{\mathsf{T}}P) = \mathrm{wp}(g, \mathrm{wp}(f, P)) = \mathrm{wp}(f \bullet g, P) = g^*(Tf)^*\mu^*_{pP}\dot{\mathsf{T}}P. \tag{7}$$

Below let $X = pP$. (6) implies that $\dot\eta_P$ is Cartesian. By putting $g = \mathrm{id}_{TTX}$ and $f = \mathrm{id}_{TX}$ in (7), we obtain $\dot{\mathsf{T}}\dot{\mathsf{T}}P = \mu^*_X(\dot{\mathsf{T}}P)$, hence $\dot\mu_P$ is Cartesian. Let $h : Y \to X$ be a morphism in $\mathbb{C}$. By putting $g = \mathrm{id}_{TX}$ and $f = \eta_X \circ h : Y \to TX$ in (7), we obtain $\dot{\mathsf{T}}(h^*P) = \dot{\mathsf{T}}(h^*\eta^*_X\dot{\mathsf{T}}P) = (\mathsf{T}h)^*(\dot{\mathsf{T}}P)$, hence $\dot{\mathsf{T}}$ is fibered. □

Therefore, for a Cartesian $\mathbb{K}$-Dijkstra structure $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$, the following assignments of posets in $\mathbb{K}$ and monotone functions in $\mathbb{K}$ become a functor $\mathrm{Wp} : (\mathbb{C}_\mathsf{T})^{op} \to \mathbb{K}$:

$$\mathrm{Wp}(X) \triangleq \mathbb{P}_X, \qquad \mathrm{Wp}(f) \triangleq \mathrm{wp}(f, \_).$$

Then the Grothendieck construction G applied to Wp yields the $\mathbb{K}$-fibration that is isomorphic to the evident functor from $\mathbb{P}_{\dot{\mathsf{T}}}$ to $\mathbb{C}_\mathsf{T}$.

Theorem 7. *Let $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ be a Cartesian $\mathbb{K}$-Dijkstra Structure. We take Kleisli resolutions $(L \dashv R : \mathbb{C}_\mathsf{T} \to \mathbb{C}, \eta, \varepsilon)$ of T and $(\dot L \dashv \dot R : \mathbb{P}_{\dot{\mathsf{T}}} \to \mathbb{P}, \dot\eta, \dot\varepsilon)$ of $\dot{\mathsf{T}}$. We define the functor $p_{\mathsf{T},\dot{\mathsf{T}}} : \mathbb{P}_{\dot{\mathsf{T}}} \to \mathbb{C}_\mathsf{T}$ by*

$$p_{\mathsf{T},\dot{\mathsf{T}}}P \triangleq pP, \qquad p_{\mathsf{T},\dot{\mathsf{T}}}f \triangleq pf.$$

(1) We have an isomorphism $p_{\mathsf{T},\dot{\mathsf{T}}} \cong G_{\mathrm{Wp}}$ in $\mathbb{K}$-$\mathbf{Fib}_c$.
(2) We have an isomorphism $p \cong G_{\mathrm{Wp}\circ L}$ in $\mathbb{K}$-$\mathbf{Fib}_c$.
(3) $((L, \dot L), (R, \dot R), (\eta, \dot\eta), (\varepsilon, \dot\varepsilon))$ is an adjunction in $\mathbb{K}$-$\mathbf{Fib}_c$.

Proof. (1, 2) Routine. We note that the reindexing functor of the fibration $p_{\mathsf{T},\dot{\mathsf{T}}}$ is $\mathrm{wp}(f, \_)$. (3) We first show that $(L, \dot L), (R, \dot R)$ are 1-cells in $\mathbb{K}$-$\mathbf{Fib}_c$. A) Equations $p_{\mathsf{T},\dot{\mathsf{T}}}\dot L = Lp$ and $p\dot R = Rp_{\mathsf{T},\dot{\mathsf{T}}}$ easily follow from $\dot{\mathsf{T}}$ being a lifting of T along p. B) We show the fiberedness of $\dot L$ and $\dot R$, that is, $\dot L(f^*P) = \mathrm{wp}(Lf, \dot LP)$ and $\dot R(\mathrm{wp}(f, P)) = (Rf)^*(\dot RP)$. By the Cartesianness of $\dot{\mathsf{T}}$, we have

$$\begin{aligned}&\dot L(f^*P) = f^*P = f^*\eta^*(\dot{\mathsf{T}}P) = \mathrm{wp}(Lf, \dot LP), \\ &\dot R(\mathrm{wp}(f, P)) = \dot{\mathsf{T}}(f^*(\dot{\mathsf{T}}P)) = (Tf)^*(\dot{\mathsf{T}}\dot{\mathsf{T}}P) = (Tf)^*\mu^*(\dot{\mathsf{T}}P) = (\mu \circ Tf)^*(\dot{\mathsf{T}}P) = (Rf)^*(\dot RP).\end{aligned}$$

(C) We show that the restrictions of $\dot L, \dot R$ to fibers belong to $\mathbb{K}$. The restriction $\dot L_X : \mathbb{P}_X \to (\mathbb{P}_{\dot{\mathsf{T}}})_{LX}$ is the identity morphism, hence belongs to $\mathbb{K}$. The restriction $\dot R_Y : (\mathbb{P}_{\dot{\mathsf{T}}})_Y \to \mathbb{P}_{RY}$ coincides with $\dot T_Y$, which belongs to $\mathbb{K}$.

We next show that $(\eta, \dot\eta)$ and $(\varepsilon, \dot\varepsilon)$ are 2-cells in $\mathbb{K}$-$\mathbf{Fib}_c$. The proof proceeds as follows:

$$P = \eta^*_{pP}\dot TP = \eta^*_{pP}\dot R\dot LP \qquad \dot L\dot RQ = \dot TQ = \mathrm{id}^*_{\mathsf{T}p_{\mathsf{T},\dot{\mathsf{T}}}Q}\dot{\mathsf{T}}Q = wp(\varepsilon_{p_{\mathsf{T},\dot{\mathsf{T}}}Q}, Q).$$

It is routine to check that $((L, \dot L), (R, \dot R), (\eta, \dot\eta), (\varepsilon, \dot\varepsilon))$ forms an adjunction in $\mathbb{K}$-$\mathbf{Fib}_c$. □

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Recent work (Aguirre et al. 2021) presents a noncomposable predicate transformer to reason about relational pre-expectations of probabilistic programs. This noncomposability can be associated to the fact that the monad lifting used in the work is not Cartesian. Their relational pre-expectation operator coincides with our categorical wppt in the Dijkstra structure $(p : \mathbf{EPMet} \to \mathbf{Set}, \mathsf{D}_\omega, \mathsf{K})$, where p is the forgetful functor from the category of extended pseudometric spaces, which is a posetal fibration, D_ω is the countable probability distribution monad (see Example 40 for the finite case), and K is the Kantorovich metric construction:

$$\mathsf{K}(X, d) \triangleq (\mathsf{D}_\omega X, d^K), \qquad d^K(\mu_1, \mu_2) \triangleq \inf_{\mu \in \Gamma(\mu_1, \mu_2)} E_{(x,y)\sim\mu}[d(x, y)].$$

Here $\Gamma(\mu_1, \mu_2)$ is the set of probabilistic couplings between μ_1, μ_2, and E denotes the expectation; see Aguirre et al. (2021, Definition 2.2) for details. Since it fails to satisfy the composability, we conclude that the Dijkstra structure is *not* Cartesian.

3.3 Change-of-base of Dijkstra Structures

In fibrational category theory, the *change-of-base* is a convenient method to introduce a fibration over a category. This method takes a fibration $p : \mathbb{P} \to \mathbb{C}$ and a functor $F : \mathbb{D} \to \mathbb{C}$ as parameters and performs the following pullback in the (large) category **CAT** of categories and functors:

$$\begin{tikzcd}
F^*\mathbb{P} \arrow[r] \arrow[d, "F^*p"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{P} \arrow[d, "p"] \\
\mathbb{D} \arrow[r, "F"'] & \mathbb{C}
\end{tikzcd}$$

We explicitly specify the pullback category $F^*\mathbb{P}$ as follows: an object is a pair $(X \in \mathbb{D}, P \in \mathbb{P})$ of objects such that $FX = pP$, and a morphism from (X, P) to (Y, Q) is a pair of morphisms $(f, \dot{f}) \in \mathbb{D}(X, Y) \times \mathbb{P}(P, Q)$ such that $Ff = p\dot{f}$. The first projection functor F^*p is called the *change-of-base* of p along F and is again a fibration (Jacobs 1999, Lemma 1.5.1). The inverse image of $(Y, Q) \in F^*\mathbb{P}$ along a morphism $f : X \to Y$ in $\mathbb{D}$ is given by $(X, (Ff)^*Q)$. When $p : \mathbb{P} \to \mathbb{C}$ is a $\mathbb{K}$-fibration, so is $F^*p : F^*\mathbb{P} \to \mathbb{D}$, and each homset of $F^*\mathbb{P}$ satisfies

$$F^*\mathbb{P}((X, P), (Y, Q)) \cong \{f \in \mathbb{D}(X, Y) \mid P \leq (Ff)^*Q\}.$$

We therefore replace homsets of $F^*\mathbb{P}$ with the right-hand side when p is a $\mathbb{K}$-fibration.

Example 8. Let **Pred**(**Set**) be the category of predicates over **Set**. This has as objects pairs (P, X) of sets such that $P \subseteq X$, and as morphisms from (P, X) to (Q, Y) functions $f : X \to Y$ such that for every $x \in P$, $f(x) \in Q$. The evident forgetful functor p mapping (P, X) to X defines a posetal fibration known sometimes as the subobject fibration.

Now consider the functor $\times : \mathbf{Set}^2 \to \mathbf{Set}$. The change-of-base construction induced by p and $\times$ constructs a fibration with total category **BRel**(**Set**), defined by the data below:

- Objects are triples (P, X_1, X_2) of sets such that $P \subseteq X_1 \times X_2$.
- Morphisms from (P, X_1, X_2) to (Q, Y_1, Y_2) are pairs of morphisms $f_1 : X_1 \to Y_1, f_2 : X_2 \to Y_2$ such that for all $(x_1, x_2) \in P$, $(f_1(x_1), f_2(x_2)) \in Q$.

The fibration is given by the forgetful functor $(\times)^*p$ mapping (P, X_1, X_2) to (X_1, X_2).

In Section 10 we will see a generalization of this construction for a general base category $\mathbb{C}$ and a general truth-value object Ω.

We now apply the change-of-base method to Dijkstra structures. Since Dijkstra structures are fibrations with monads, we naturally expect that the functor, along which we take a pullback, interacts with the monad of a Dijkstra structure. We describe this interaction by the *monad opfunctor* axioms given by Street (Street 1972, Section 4).

Definition 9. *A* monad morphism *from a monad* $(\mathsf{S}, \eta^{\mathsf{S}}, \mu^{\mathsf{S}})$ *on* $\mathbb{D}$ *to a monad* $(\mathsf{T}, \eta^{\mathsf{T}}, \mu^{\mathsf{T}})$ *on* $\mathbb{C}$ *is a pair of a functor* $F : \mathbb{D} \to \mathbb{C}$ *and a natural transformation* $\alpha : F\mathsf{S} \to \mathsf{T}F$ *satisfying the following axioms:*

$$\alpha \star (F\eta^{\mathsf{S}}) = \eta^{\mathsf{T}} F, \qquad \alpha \star (F\mu^{\mathsf{S}}) = (\mu^{\mathsf{T}} F) \star (\mathsf{T}\alpha) \star (\alpha \mathsf{S}). \tag{8}$$

The following theorem extends the change-of-base method to Dijkstra structures along monad morphism. It preserves Cartesianness of Dijkstra structures.

Theorem 10. *Let* $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ *be a (resp. Cartesian)* $\mathbb{K}$*-Dijkstra structure,* S *be a monad on a category* $\mathbb{D}$ *and* $(F : \mathbb{D} \to \mathbb{C}, \alpha : F\mathsf{S} \to \mathsf{T}F)$ *be a monad morphism from* S *to* T*. We define a monad* $(\dot{\mathsf{S}}, \eta^{\dot{\mathsf{S}}}, \mu^{\dot{\mathsf{S}}})$ *on* $F^*\mathbb{P}$ *by*

$$\dot{\mathsf{S}}(X, P) \triangleq (\mathsf{S}X, \alpha_X^*(\dot{\mathsf{T}}P)), \qquad \dot{\mathsf{S}}f \triangleq \mathsf{S}f, \qquad \eta^{\dot{\mathsf{S}}}_{(X,P)} \triangleq \eta^{\mathsf{S}}_X, \qquad \mu^{\dot{\mathsf{S}}}_{(X,P)} \triangleq \mu^{\mathsf{S}}_X.$$

Then the tuple $(F^*p : F^*\mathbb{P} \to \mathbb{D}, \mathsf{S}, \dot{\mathsf{S}})$ *is a (resp. Cartesian)* $\mathbb{K}$*-Dijkstra structure.*

Proof. Let $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ be a $\mathbb{K}$-Dijkstra structure and $F^*p : F^*\mathbb{P} \to \mathbb{D}$ be the $\mathbb{K}$-fibration obtained by the change-of-base of p along F.

We first verify that $f \in F^*\mathbb{P}((X, P), (Y, Q))$ implies $\dot{S}f \in F^*\mathbb{P}(\dot{\mathsf{S}}(X, P), \dot{\mathsf{S}}(Y, Q))$.

$$\alpha_X^*(\dot{\mathsf{T}}P) \le \alpha_X^*(\dot{\mathsf{T}}((Ff)^*Q)) \le \alpha_X^*(\mathsf{T}Ff)^*(\dot{\mathsf{T}}Q) = (F\mathsf{S}f)^*\alpha_Y^*(\dot{\mathsf{T}}Q).$$

We next verify that $\eta^{\dot{\mathsf{S}}}_{(X,P)} \in F^*\mathbb{P}((X, P), \dot{\mathsf{S}}(X, P))$ and $\mu^{\dot{\mathsf{S}}}_{(X,P)} \in F^*\mathbb{P}(\dot{\mathsf{S}}(\dot{\mathsf{S}}(X, P)), \dot{\mathsf{S}}(X, P))$.

$$P \le (\eta^{\mathsf{T}}_{FX})^*(\dot{\mathsf{T}}P) = (F\eta^{\mathsf{S}}_X)^*\alpha_X^*(\dot{\mathsf{T}}P),$$
$$\alpha_{\mathsf{S}X}^*(\dot{\mathsf{T}}(\alpha_X^*(\dot{\mathsf{T}}P))) \le \alpha_{\mathsf{S}X}^*(\mathsf{T}\alpha_X)^*(\dot{\mathsf{T}}\dot{\mathsf{T}}P) \le \alpha_{\mathsf{S}X}^*(\mathsf{T}\alpha_X)^*(\mu^{\mathsf{T}}_{FX})^*(\dot{\mathsf{T}}P) = (F\mu^{\mathsf{S}}_X)^*\alpha_X^*(\dot{\mathsf{T}}P).$$

Now assume that the $\mathbb{K}$-Dijkstra structure $(p, \mathsf{T}, \dot{\mathsf{T}})$ is Cartesian. Then the last two inequalities become equalities. Moreover,

$$\begin{aligned} \dot{\mathsf{S}}(f^*(Y, Q)) &= \dot{\mathsf{S}}(X, (Ff)^*Q) = (\mathsf{S}X, \alpha_X^*(\dot{\mathsf{T}}((Ff)^*Q))) \\ &= (\mathsf{S}X, \alpha_X^*(\mathsf{T}Ff)^*(\dot{\mathsf{T}}Q)) = (\mathsf{S}X, (F\mathsf{S}f)^*\alpha_Y^*(\dot{\mathsf{T}}Q)) = (\mathsf{S}f)^*(\dot{\mathsf{S}}(Y, Q)). \end{aligned}$$

Therefore $\dot{\mathsf{S}}$ is fibered. The functor $\dot{\mathsf{S}}$ is a $\mathbb{K}$-lifting of S, because its restriction $\dot{\mathsf{S}}_X : F^*\mathbb{P}_X \to F^*\mathbb{P}_{\mathsf{S}X}$ to the fiber over $X \in \mathbb{D}$ is the composite of α_X^* and $\dot{\mathsf{T}}_X^*$, both of which belong to $\mathbb{K}$. This concludes that $(F^*p, \mathsf{S}, \dot{\mathsf{S}})$ is a Cartesian $\mathbb{K}$-Dijkstra structure. □

We will see examples of change-of-base of Dijkstra structures in Section 10.

3.4 Dijkstra Structures with Strengths

In Moggi's theory of monadic models of computational effects (Moggi 1991), *monad strengths* are an important ingredient, since they allow us to sequence effectful computations with a common context. Below we discuss an extension of Dijkstra structures with strengths.

Let $(\mathbb{C}, 1, \times)$ be a category with finite products and (T, η, μ) be a monad on it. We first recall the concept of *strength* on T (Moggi 1991, Definition 3.2). It is a natural transformation $\theta_{X,Y} : X \times \mathsf{T}Y \to \mathsf{T}(X \times Y)$ satisfying four axioms (below $\alpha_{X,Y,Z} : (X \times Y) \times Z \to X \times (Y \times Z)$ is the associator):

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

$$\mathsf{T}(\pi_2) \circ \theta_{1,X} = \pi_2 \qquad \mathsf{T}\alpha_{X,Y,Z} \circ \theta_{X\times Y,Z} = \theta_{X,Y\times Z} \circ (X \times \theta_{Y,Z}) \circ \alpha_{X,Y,\mathsf{T}Z}$$
$$\eta_{X\times Y} = \theta_{X,Y} \circ (X \times \eta_Y) \qquad \mu_{X\times Y} \circ \mathsf{T}\theta_{X,Y} \circ \theta_{X,\mathsf{T}Y} = \theta_{X,Y} \circ (X \times \mu_Y).$$

Let $f : X \to \mathsf{T}Y$ be a morphism in $\mathbb{C}$ and $Z \in \mathbb{C}$ be an object. The strength allows us to extend the domain and codomain of the effectful computation represented by f with Z. The extension is given as follows:

$$Z \cdot_\theta f \triangleq \theta_{Z,Y} \circ (\mathrm{id}_Z \times f) \in \mathbb{C}(Z \times X, \mathsf{T}(Z \times Y)).$$

Intuitively, the morphism $Z \cdot_\theta f$ takes an input (z, x) from the extended input space $Z \times X$, applies the effectful computation f to x, and returns the result (z, v), where v is the value part of the computation $f(x)$. The input z is directly forwarded to the output.

To extend the concept of strength to Dijkstra structures, we first set-up the concept of finite products on Dijkstra structures. We define *finite products* on a posetal fibration $p : \mathbb{P} \to \mathbb{C}$ to be the following data:

(1) a pair of terminal objects 1 in $\mathbb{C}$ and $\dot{1}$ in $\mathbb{P}$ such that $p\dot{1} = 1$, and
(2) a pair of binary product functors $(\times) : \mathbb{C}^2 \to \mathbb{C}$ and $(\dot{\times}) : \mathbb{P}^2 \to \mathbb{P}$ such that $(\dot{\times})$ preserves Cartesian morphisms and p strictly preserves binary products in the following sense:

$$(f \times g)^*(P \mathbin{\dot{\times}} Q) = f^*P \mathbin{\dot{\times}} g^*Q \qquad p(P \mathbin{\dot{\times}} Q) = pP \times pQ, \qquad p\pi_i^{P,Q} = \pi_i^{pP,pQ}.$$

A pair of a posetal fibration and finite products on it is denoted by $p : (\mathbb{P}, \dot{1}, \dot{\times}) \to (\mathbb{C}, 1, \times)$. We remark that every fiber of p has finite meets, and reindexing functors preserve these meets.[2] The top element of $\mathbb{P}_X$ is $!_X^*\dot{1} \in \mathbb{P}_X$, and the meet of $P, Q \in \mathbb{P}_X$ is given by $P \wedge Q \triangleq \langle \mathrm{id}_X, \mathrm{id}_X\rangle^* \circ (P \mathbin{\dot{\times}} Q)$, and the binary product $(\dot{\times})$ satisfies $P \mathbin{\dot{\times}} Q = \pi_1^*P \wedge \pi_2^*Q$. A logical analogy of the binary product $(\dot{\times})$ on $\mathbb{P}$ is to take the conjunction $\phi(i) \wedge \psi(j)$ of unary predicates $\phi(i)$ and $\psi(j)$ and regard it as a binary predicate.

We define a Dijkstra structure with finite products to be a tuple of a Dijkstra structure $(p, \mathsf{T}, \dot{\mathsf{T}})$ and a finite product $(1, \dot{1}, \times, \dot{\times})$ on p. We write $(p : (\mathbb{P}, \dot{1}, \dot{\times}) \to (\mathbb{C}, 1, \times), \mathsf{T}, \dot{\mathsf{T}})$ for such a tuple.

Definition 11. *A* strength *on a Dijkstra structure* $(p : (\mathbb{P}, \dot{1}, \dot{\times}) \to (\mathbb{C}, 1, \times), \mathsf{T}, \dot{\mathsf{T}})$ *with finite products is a pair* $(\theta, \dot{\theta})$ *of a strength* θ *on* T *and a strength* $\dot{\theta}$ *on* $\dot{\mathsf{T}}$ *such that* $p\dot{\theta}_{P,Q} = \theta_{pP,pQ}$ *holds for any* $P, Q \in \mathbb{P}$. *We say that the strength* $(\theta, \dot{\theta})$ *is* Cartesian *if each component of* $\dot{\theta}$ *is Cartesian.*

Again, Dijkstra structures with strengths are considered in the context of logical relations for strong monads (Filinski 1996, 2007; Goubault-Larrecq et al. 2008; Katsumata 2005; Katsumata et al. 2018). Since we are considering posetal fibrations, a natural transformation $\dot{\theta}$ is a strength on $\dot{\mathsf{T}}$ if it is above a strength on T. We formally state this property as follows.

Lemma 12. *Let* $(p : (\mathbb{P}, \dot{1}, \dot{\times}) \to (\mathbb{C}, 1, \times), \mathsf{T}, \dot{\mathsf{T}})$ *be a Dijkstra structure with finite products. Then A strength* $(\theta, \dot{\theta})$ *on* $(p, \mathsf{T}, \dot{\mathsf{T}})$ *bijectively corresponds to a strength* θ *on* T *satisfying the inequality* $P \mathbin{\dot{\times}} \dot{\mathsf{T}}Q \leq \theta_{pP,pQ}^* \dot{\mathsf{T}}(P \mathbin{\dot{\times}} Q)$. *Moreover, when the strength is Cartesian, the inequality becomes the equality.*

A strength on a Dijkstra structure with finite products is related to the reasoning principle behind the *frame rule*, which we informally explain below. Let f be an imperative program over a program variable context Γ, and suppose that it satisfies a Hoare triple $\Gamma \vdash \phi\{f\}\psi$; it is explicitly annotated with the context Γ for the program f and pre/post conditions ϕ, ψ. Let ρ be another assertion over a program variable context Δ that is disjoint from Γ. Then the *frame rule* allows us to derive a Hoare triple on the program f when executed in the extended program variable context Γ, Δ:

$$\frac{\Gamma \vdash \phi\{f\}\psi \quad \Delta \vdash \rho}{\Gamma, \Delta \vdash (\rho \wedge \phi)\{f\}(\rho \wedge \psi)}$$

The frame rule can be expressed in a general Dijkstra structure with finite products (see item (2) below) whose base monad has a strength. We show that the soundness of the frame rule is *equivalent* to the existence of a strength on the Dijkstra structure with finite products.

Proposition 13. *Let* $(p : (\mathbb{P}, \dot{1}, \dot{\times}) \to (\mathbb{C}, 1, \times), \mathsf{T}, \dot{\mathsf{T}})$ *be a Dijkstra structure with finite products and* θ *be a strength of* T. *The following are equivalent:*

(1) *There exists a strength* $\dot{\theta}$ *on* $\dot{\mathsf{T}}$ *such that* $(\theta, \dot{\theta})$ *is a strength of the Dijkstra structure.*
(2) $P\{f\}Q$ *implies* $(R \dot{\times} P)\{pR \cdot_\theta f\}(R \dot{\times} Q)$ *for any* $P, Q, R \in \mathbb{P}$ *and morphism* $f : pP \to \mathsf{T}pQ$ *in* $\mathbb{C}$.
(3) $R \dot{\times} \mathrm{wp}(f, Q) \leq \mathrm{wp}(pR \cdot_\theta f, (R \dot{\times} Q))$ *holds for any* $P, Q, R \in \mathbb{P}$ *and morphism* $f : pP \to \mathsf{T}pQ$ *in* $\mathbb{C}$.

Furthermore, $(\theta, \dot{\theta})$ *in* (1) *is Cartesian if and only if the inequality in* (3) *is an equality.*

Proof. Assume $f : X \to \mathsf{T}Y$. We first prove (1) implies (3). Let $Z = pR$. Note that $\dot{\theta} : R \dot{\times} \dot{\mathsf{T}}Q \to \dot{\mathsf{T}}(R \dot{\times} Q)$ is above $\theta_{Z,Y} : Z \times \mathsf{T}Y \to \mathsf{T}(Z \times Y)$. Therefore, $R \dot{\times} \dot{\mathsf{T}}Q \leq \theta^*_{Z,Y}(\dot{\mathsf{T}}(R \dot{\times} Q))$, and we have an equality if $\dot{\theta}$ is Cartesian. Then

$$\begin{aligned} R \dot{\times} \mathrm{wp}(f, Q) &= R \dot{\times} f^* \dot{\mathsf{T}}Q = (\mathrm{id}_Z \times f)^*(R \dot{\times} \dot{\mathsf{T}}Q) \leq (\mathrm{id}_Z \times f)^* \theta^*_{Z,Y} \dot{\mathsf{T}}(R \dot{\times} Q) \\ &= (\theta_{Z,Y} \circ (\mathrm{id}_Z \times f))^* \dot{\mathsf{T}}(R \dot{\times} Q) = (Z \cdot_\theta f)^* \dot{\mathsf{T}}(R \dot{\times} Q) = \mathrm{wp}(Z \cdot_\theta f, (R \dot{\times} Q)) \end{aligned}$$

We then show that (3) implies (1). By definition, $\mathrm{wp}(Z \cdot_\theta \mathrm{id}_{\mathsf{T}Y}, (R \dot{\times} Q)) = \theta^*_{Z,Y} \dot{\mathsf{T}}(R \dot{\times} Q)$ and by (3) we have an inclusion $R \dot{\times} \mathrm{wp}(\mathrm{id}_{\mathsf{T}Y}, Q) = R \dot{\times} \dot{\mathsf{T}}Q \leq \theta^*_{Z,Y} \dot{\mathsf{T}}(R \dot{\times} Q)$ in the fiber above $Z \times \mathsf{T}Y$. From Lemma 12, we obtain a strength $(\theta, \dot{\theta})$ for the Dijkstra structure. If $R \dot{\times} \dot{\mathsf{T}}Q = \theta^*_{Z,Y} \dot{\mathsf{T}}(R \dot{\times} Q)$, then the strength $(\theta, \dot{\theta})$ is Cartesian.

We now show that (3) implies (2). From (3), we get $(R \dot{\times} \mathrm{wp}(f, Q))\{Z \cdot_\theta f\}(R \dot{\times} Q)$ By assumption, also $P\{f\}Q$, so $P \leq \mathrm{wp}(f, Q)$. From all this, it follows that $(R \dot{\times} P)\{Z \cdot_\theta f\}(R \dot{\times} Q)$

Finally, we show that (2) implies (3). By definition,

$$\mathrm{wp}(f, Q)\{f\}Q \implies (R \dot{\times} \mathrm{wp}(f, Q))\{Z \cdot_\theta f\}(R \dot{\times} Q) \implies R \dot{\times} \mathrm{wp}(f, Q) \leq \mathrm{wp}(Z \cdot_\theta f, (R \dot{\times} Q)).$$

□

3.5 Strongest Postcondition Predicate Transformers

We turn our attention to *strongest postcondition predicate transformers*. In Hoare logic, the *strongest postcondition* with respect to a program f and a precondition ϕ is a formula ψ such that (1) $\phi\{f\}\psi$ holds, and (2) if $\phi\{f\}\xi$ holds, then ψ must entail ξ.

We formulate the concept of strongest postcondition in a $\mathbb{K}$-Dijkstra structure $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$. The strongest postcondition with respect to a morphism $f : X \to \mathsf{T}Y$ in $\mathbb{C}$ and an object $P \in \mathbb{P}_X$ is an object $S \in \mathbb{P}_Y$ such that for any $Q \in \mathbb{P}_Y$, $S \leq Q$ if and only if $P\{f\}Q$. If such S exists, it is uniquely determined by f and P, hence we write it by $\mathrm{sp}(f, P)$. To summarize, we obtain the three-way equivalence:

$$\mathrm{sp}(f, P) \leq Q \iff P\{f\}Q \iff P \leq \mathrm{wp}(f, Q). \tag{9}$$

This exhibits that the sppt corresponds to the *left adjoint* of the wppt, leading us to the following definition.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Definition 14. *We say that a* $\mathbb{K}$*-Dijkstra structure* $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ *admits the* strongest postcondition predicate transformer *(*sppt *for short) if for any morphism* $f : X \to \mathsf{T}Y$ *in* $\mathbb{C}$*,* $\mathrm{wp}(f, \_) \in \mathbb{K}(\mathbb{P}_Y, \mathbb{P}_X)$ *is a right adjoint. Its associated left adjoint is denoted by* $\mathrm{sp}(f, \_) \in \mathbf{Pos}(\mathbb{P}_X, \mathbb{P}_Y)$*.*

We note that the sppt $\mathrm{sp}(f, \_)$ does not need to belong to $\mathbb{K}$. From this definition, if $\mathbb{K}$ is a subcategory of **Pos** such that every morphism in $\mathbb{K}$ is a right adjoint, then any $\mathbb{K}$-Dijkstra structure admits the sppt. A typical example of such $\mathbb{K}$ is the subcategory $\mathbf{CLat}_\wedge$ of **Pos** consisting of complete lattices and meet-preserving functions. Since every morphism in $\mathbf{CLat}_\wedge$ is a right adjoint, any $\mathbf{CLat}_\wedge$-Dijkstra structure admits the sppt.

Like the case of wppts, sppts in general do not satisfy the composability. They do so if and only if the $\mathbb{K}$-Dijkstra structure is Cartesian:

Theorem 15. *Let* $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ *be a* $\mathbb{K}$*-Dijkstra structure admitting the sppt. We have inequalities*

$$\mathrm{sp}(\eta_X, P) \le P, \qquad \mathrm{sp}(f \bullet g, P) \le \mathrm{sp}(f, \mathrm{sp}(g, P)), \tag{10}$$

for any f, g, P *of appropriate type. Moreover, both inequalities in (10) become equalities if and only if the* $\mathbb{K}$*-Dijkstra structure is Cartesian.*

Proof. (10) easily follows by taking the adjoint mate of (5) in Theorem 6. Next, assume that inequalities in (10) are equalities. We show that inequalities in (5) become equalities. From the assumption, we obtain

$$\mathrm{wp}(\eta_X, P) = \mathrm{sp}(\eta_X, \mathrm{wp}(\eta_X, P)) \le P,$$

and

$$\mathrm{sp}(f, \mathrm{sp}(g, \mathrm{wp}(f \bullet g, P))) = \mathrm{sp}(f \bullet g, \mathrm{wp}(f \bullet g, P)) \le P,$$

which implies $\mathrm{wp}(f \bullet g, P) \le \mathrm{wp}(g, \mathrm{wp}(f, P))$. These are the other direction of (5), hence proved. The converse is proven similarly. $\square$

Therefore, for a Cartesian Dijkstra structure $(p : \mathbb{P} \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$ admitting sppts, the functor $p_{\mathsf{T},\dot{\mathsf{T}}} : \mathbb{P}_{\dot{\mathsf{T}}} \to \mathbb{C}_{\mathsf{T}}$ between Kleisli categories defined in Theorem 7 is a *posetal bifibration* (a posetal fibration such that each reindexing functor is a right adjoint; see Jacobs (1999, Definition 9.1.1) for the definition).

4. Dijkstra Structures on Lax Slice Categories

In the setting of an arbitrary posetal fibration, we are unaware of a general way of constructing Cartesian liftings of monads. However, in the specific case of a *domain fibration* from a *lax slice category* (introduced in Section 4.2), there is a recipe for constructing Cartesian liftings. We discuss this construction in this section, and visit examples in Sections 5–9.

A lax slice category is constructed from an object Ω in a category $\mathbb{C}$ and a partial order $\le_X$ given to each homset $\mathbb{C}(X, \Omega)$. We call the pair $(\Omega, \le)$ an *ordered object* in $\mathbb{C}$ (Definition 16). Then the lax slice category $\mathbb{C}/_\ell\Omega$ is defined by the following data: an object is a morphism into Ω, and a

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

morphism from f to g is a morphism $h : \mathrm{dom}(f) \to \mathrm{dom}(g)$ with the following inequality:

$$\begin{tikzcd}
X \arrow[rr, "h"] \arrow[dr, "f"'] & \arrow[d, phantom, "\leq_X"] & Y \arrow[dl, "g"] \\
& \Omega &
\end{tikzcd}$$

The *domain functor* extracting the domain part of this diagram, which is denoted by $d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}$, becomes a posetal fibration; we hence call it a *domain fibration*.[3] In Theorem 26, we show that a Cartesian lifting of a monad T on $\mathbb{C}$ along $d^{\mathbb{C},\Omega}$ bijectively corresponds to an Eilenberg–Moore monotone algebra on Ω. This correspondence significantly reduces the search space for Cartesian liftings. Specially, when $\mathbb{C} = \mathbf{Set}$ and Ω and $T\Omega$ are both finite, it is possible to enumerate all such algebras. We carry out this enumeration for some monads in Section 5.

We also seek for a principle behind the correspondence between structures on $d^{\mathbb{C},\Omega}$ and structures on Ω. We present it as a biequivalence result (Theorem 25) between the 2-category $\mathbf{CO}(\mathbb{K})$ of ordered objects (Definition 21) and the 2-category $\mathbb{K}$-$\mathbf{Fib}_g$ of $\mathbb{K}$-fibrations with split generic objects (Definition 24), the latter of which is a full sub-2-category of $\mathbb{K}$-$\mathbf{Fib}$ (the left of (11)). A similar result holds when 2-cells of $\mathbb{K}$-$\mathbf{Fib}_g$ and $\mathbb{K}$-$\mathbf{Fib}$ are restricted to Cartesian ones; 2-cells of $\mathbf{CO}(\mathbb{K})$ are also restricted accordingly (the right of (11)).

$$\mathbf{CO}(\mathbb{K}) \xrightarrow{\equiv} \mathbb{K}\text{-}\mathbf{Fib}_g \hookrightarrow \mathbb{K}\text{-}\mathbf{Fib} \qquad \mathbf{CO}_c(\mathbb{K}) \xrightarrow{\equiv} \mathbb{K}\text{-}\mathbf{Fib}_{c,g} \hookrightarrow \mathbb{K}\text{-}\mathbf{Fib}_c\,. \tag{11}$$

This biequivalence is working behind the proof of lifting-algebra correspondence (Theorem 25). The same proof strategy is employed for establishing a correspondence between (1) liftings of distributive laws and distributive laws satisfying certain axioms with respect to Eilenberg–Moore monotone algebra pairs (Theorem 27) and (2) liftings of monad strengths and strengths satisfying a certain inequality with respect to an Eilenberg–Moore monotone algebras (Theorem 29).

In Section 4.4, we discuss the interaction between wppts of a Cartesian Dijkstra structure of the form $(d^{\Omega,\mathbb{C}}, \mathsf{T}, \dot{\mathsf{T}})$ and *algebraic operations* of T in the sense of Plotkin and Power (2001).

4.1 Ordered Objects

With the goal of having fibers with a partial order structure, we consider slices over ordered objects, defined below:

Definition 16. *An* ordered object *in a category* $\mathbb{C}$ *is a pair of an object* Ω *and an assignment of a partial order* $\leq_X$ *to the homset* $\mathbb{C}(X, \Omega)$ *for each* $X \in \mathbb{C}$*. These partial orders should be such that, for any morphism* $f : Y \to X$ *in* $\mathbb{C}$*,* $i \leq_X j$ *implies* $i \circ f \leq_Y j \circ f$*.*

Equivalently, an ordered object is a pair of an object Ω and a functor $A : \mathbb{C}^{op} \to \mathbf{Pos}$ such that $U \circ A = \mathbb{C}(\_, \Omega)$, where $U : \mathbf{Pos} \to \mathbf{Set}$ is the forgetful functor. The object Ω corresponds to the set of truth values, and a morphism $i : X \to \Omega$ in $\mathbb{C}$ corresponds to an Ω-valued predicate on X. The order $\leq_X$ compares the strength of predicates: $i \leq_X j$ means that i implies j.

To impose further conditions on the partial order, we parameterize Definition 16 by a subcategory $\mathbb{K}$ of $\mathbf{Pos}$.

Definition 17. *We say that an ordered object* $(\Omega, \leq)$ *in a category* $\mathbb{C}$ belongs to $\mathbb{K}$ *if each poset* $(\mathbb{C}(X, \Omega), \leq_X)$ *belongs to* $\mathbb{K}$*, and for any morphism* $f : X \to Y$ *in* $\mathbb{C}$*, the function* $\_ \circ f$ *is a morphism from* $(\mathbb{C}(Y, \Omega), \leq_Y)$ *to* $(\mathbb{C}(X, \Omega), \leq_X))$ *in* $\mathbb{K}$*.*

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Example 18. A typical way to give an ordered object in **Set** is to take a poset $(\Omega, \leq)$ and define the partial order $\leq_X$ on the homset $\mathbf{Set}(X, \Omega)$ to be the pointwise order. Examples in Sections 5–8 all use domain fibrations arising from this kind of ordered object. Moreover, if the poset $(\Omega, \leq)$ belongs to a subcategory $\mathbb{K}$ of **Pos**, and $\mathbb{K}$ has small powers[4] of $(\Omega, \leq)$, then the ordered object $(\Omega, \leq_X)$ belongs to $\mathbb{K}$.

Example 19. Generalizing the previous example, let $\mathbb{C}$ be a category with finite limits. An *internal partial order* is a monomorphism $m : R \rightarrowtail \Omega \times \Omega$ satisfying reflexivity, transitivity, and antisymmetry expressed diagrammatically; see Lane and Moerdijk (1994, Section IV.8). For each $X \in \mathbb{C}$, the following binary relation $\leq_X$ over $\mathbb{C}(X, \Omega)$ becomes a partial order:

$$f \leq_X g \iff \exists h : X \to R \,.\, \langle f, g\rangle = m \circ h,$$

Then $(\Omega, \leq_X)$ becomes an ordered object in $\mathbb{C}$. To summarize, an internal partial order in $\mathbb{C}$ induces an ordered object in $\mathbb{C}$.

Example 20. Not every ordered object arises in the way given in Example 19. In categories without finite limits (such as Kleisli categories of monads), we cannot have an internal partial order, but we can still have ordered objects. For example, the Kleisli category of the finite probability distribution monad D (Example 40) over **Set** does not have finite products, but there is an ordered object $\{\bot, \top\}$ (a two-point set) whose order is defined by

$$f \leq_X g \iff \forall x \in X \,.\, f(x)(\top) \leq g(x)(\top).$$

When $\mathbb{C}$ comes with a **Pos**-enrichment, that is, an assignment of a partial order $\sqsubseteq_{X,Y}$ to each homset $\mathbb{C}(X, Y)$ making the composition monotone, any object $\Omega \in \mathbb{C}$ determines an ordered object $(\Omega, \sqsubseteq_{\_,\Omega})$. However, it is not desirable to restrict our attention to such ordered objects in **Pos**-enriched categories, because there can be cases where the order $\sqsubseteq_{X,\Omega}$ is not adequate for comparing strength of predicates on X. For instance, the category **Set** may be seen as a **Pos**-enriched category by considering the discrete order on each homset. However, the ordered object $(\Omega, =_{\_,\Omega})$ based on this enrichment only gives the discrete order on Ω-valued predicates, which is not very useful. Therefore we need freedom to give an ordered object independently from the **Pos**-enrichment on a category.

4.2 Lax Slice Construction

Let $\mathbb{C}$ be a category and $(\Omega, \leq)$ be an ordered object in $\mathbb{C}$ belonging to a subcategory $\mathbb{K}$ of **Pos**. We define the *lax slice category* $\mathbb{C}/_\ell\Omega$ by the following data:

- An object is a morphism in $\mathbb{C}$ whose codomain is Ω.
- A morphism from i to j is a morphism $h : \mathrm{dom}(i) \to \mathrm{dom}(j)$ in $\mathbb{C}$ such that $i \leq_{\mathrm{dom}(i)} j \circ h$.

The *domain functor* $d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}$ is defined by

$$d^{\mathbb{C},\Omega} i \triangleq \mathrm{dom}(i), \qquad\qquad d^{\mathbb{C},\Omega}(f) \triangleq f.$$

It is a $\mathbb{K}$-fibration, and the fiber category $(\mathbb{C}/_\ell\Omega)_X$ at $X \in \mathbb{C}$ is the poset $(\mathbb{C}(X, \Omega), \leq_X)$. The pullback of $i : X \to \Omega$ along $h : Y \to X$ is given by $h^* i = i \circ h$.

We further investigate the 2-categorical nature of the construction of lax slice categories. We introduce the following 2-categories $\mathbf{CO}(\mathbb{K})$ and $\mathbf{CO}_c(\mathbb{K})$ whose 0-cells are categories with ordered objects belonging to a subcategory $\mathbb{K}$ of **Pos**.

Definition 21. *Let $\mathbb{K}$ be a subcategory of* **Pos**. *The 2-category* $\mathbf{CO}(\mathbb{K})$ *is defined by the following data.*

- *A 0-cell is a tuple $(\mathbb{C}, \Omega, \leq)$ consisting of a category $\mathbb{C}$ and an ordered object $(\Omega, \leq)$ in $\mathbb{C}$ belonging to $\mathbb{K}$.*
- *A 1-cell from $(\mathbb{C}, \Omega, \leq)$ to $(\mathbb{D}, \Pi, \leq)$ is a pair of a functor $F : \mathbb{C} \to \mathbb{D}$ and a morphism $o : F\Omega \to \Pi$ in $\mathbb{D}$ such that $o \circ F\_$ is a monotone function belonging to $\mathbb{K}$ in the following sense:*

$$\forall X \in \mathbb{C} \,.\, o \circ F\_ \in \mathbb{K}((\mathbb{C}(X, \Omega), \leq_X), (\mathbb{D}(FX, \Pi), \leq_{FX})). \tag{12}$$

  The identity 1-cell and the composition of 1-cells are defined by

$$\mathrm{id}_{(\mathbb{C},\Omega,\trianglelefteq)} \triangleq (\mathrm{Id}_{\mathbb{C}}, \mathrm{id}_{\Omega}), \qquad (F, o) \circ (G, o') \triangleq (F \circ G, o \circ Fo').$$

- *For 1-cells $(F, o), (G, o') : (\mathbb{C}, \Omega, \leq) \to (\mathbb{D}, \Pi, \leq)$, a 2-cell from the former to the latter is a natural transformation $\alpha : F \to G$ such that $o \leq o' \circ \alpha_{\Omega}$.*

Definition 22. *We define a subcategory $\mathbf{CO}_c(\mathbb{K})$ of $\mathbf{CO}(\mathbb{K})$ by restricting 2-cells from (F, o) to (G, o') to natural transformations $\alpha : F \to G$ such that $o = o' \circ \alpha\Omega$.*

We extend the lax slice construction to 2-functors. We first mention a few facts:

- For a 1-cell $(F, o) \in \mathbf{CO}(\mathbb{K})((\mathbb{C}, \Omega, \leq), (\mathbb{D}, \Pi, \leq))$, let $F_o : \mathbb{C}/_{\ell}\Omega \to \mathbb{D}/_{\ell}\Pi$ be the functor defined by

$$F_o(i) \triangleq o \circ Fi, \qquad F_o(f) \triangleq Ff.$$

  From (12) it follows that (F, F_o) is a fibered $\mathbb{K}$-functor from $d^{\mathbb{C},\Omega}$ to $d^{\mathbb{D},\Pi}$.
- There is a bijective correspondence between 2-cells in $\mathbf{CO}(\mathbb{K})$ and $\mathbb{K}$-**Fib**:

$$\begin{aligned} &\mathbf{CO}(\mathbb{K})((\mathbb{C}, \Omega, \leq), (\mathbb{D}, \Pi, \leq))((F, o), (G, o')) && (13)\\ &= \{\alpha : F \to G \mid o \leq o' \circ \alpha_{\Pi}\} && (14)\\ &= \{\alpha : F \to G \mid \forall i \,.\, F_o i \leq \alpha^*_{\mathrm{dom}(i)}(G_{o'} i)\} && (15)\\ &\cong \mathbb{K}\text{-}\mathbf{Fib}(d^{\mathbb{C},\Omega}, d^{\mathbb{D},\Pi})((F, F_o), (G, G_o)). && (16) \end{aligned}$$

  The equality between (14) and (15) is by

$$\begin{aligned} (\forall i \,.\, F_o i \leq \alpha^*_{\mathrm{dom}(i)}(G_{o'} i)) &\iff (\forall i \,.\, o \circ Fi \leq o' \circ Gi \circ \alpha_{\mathrm{dom}(i)} = o' \circ \alpha_{\Pi} \circ Fi)\\ &\iff o \leq o' \circ \alpha_{\Pi}, \end{aligned}$$

  and the bijection from (15) to (16) is Ξ^{-1} given in (1). A similar bijective correspondence between 2-cells of $\mathbf{CO}_c(\mathbb{K})$ and $\mathbb{K}$-$\mathbf{Fib}_c$ also holds by using Ξ_c^{-1} instead of Ξ^{-1}:

$$\begin{aligned} &\mathbf{CO}_c(\mathbb{K})((\mathbb{C}, \Omega, \leq), (\mathbb{D}, \Pi, \leq))((F, o), (G, o'))\\ &\cong \mathbb{K}\text{-}\mathbf{Fib}_c(d^{\mathbb{C},\Omega}, d^{\mathbb{D},\Pi})((F, F_o), (G, G_o)). \end{aligned}$$

From these, we define 2-functors $L_{\mathbb{K}} : \mathbf{CO}(\mathbb{K}) \to \mathbb{K}\text{-}\mathbf{Fib}$ and $L_{\mathbb{K},c} : \mathbf{CO}_c(\mathbb{K}) \to \mathbb{K}\text{-}\mathbf{Fib}_c$ by

$$\begin{aligned} L_{\mathbb{K}}(\mathbb{C}, \Omega, \leq) &\triangleq d^{\mathbb{C},\Omega} : \mathbb{C}/_{\ell}\Omega \to \mathbb{C}, & L_{\mathbb{K}}(F, o) &\triangleq (F, F_o), & L_{\mathbb{K}}(\alpha) &\triangleq \Xi^{-1}(\alpha)\\ L_{\mathbb{K},c}(\mathbb{C}, \Omega, \leq) &\triangleq d^{\mathbb{C},\Omega} : \mathbb{C}/_{\ell}\Omega \to \mathbb{C}, & L_{\mathbb{K},c}(F, o) &\triangleq (F, F_o), & L_{\mathbb{K},c}(\alpha) &\triangleq \Xi_c^{-1}(\alpha). \end{aligned}$$

The verification of this being a 2-functor is routine, and omitted.

Theorem 23. *Let $\mathbb{K}$ be a subcategory of* **Pos**. *The 2-functors $L_{\mathbb{K}}$ and $L_{\mathbb{K},c}$ are local isomorphisms, that is, each hom-functor is an isomorphism.*

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Proof. We first prove the case of $L_{\mathbb{K}}$; the proof of the case $L_{\mathbb{K},c}$ is similar. It suffices to show that each hom-functor:

$$L_{\mathbb{K}} : \mathbf{CO}(\mathbb{K})((\mathbb{C}, \Omega, \leq), (\mathbb{D}, \Pi, \leq)) \to \mathbb{K}\text{-}\mathbf{Fib}(d^{\mathbb{C},\Omega}, d^{\mathbb{D},\Pi})$$

is bijective on objects, because full faithfulness is already guaranteed by (13)–(16). Let $(F, \dot{F}) : d^{\mathbb{C},\Omega} \to d^{\mathbb{D},\Pi}$ be a 1-cell in $\mathbb{K}$-**Fib**. We show that $(F, \dot{F}(\mathrm{id}_\Omega)) \in \mathbf{CO}(\mathbb{K})$ is the unique 1-cell yielding $(F, \dot{F})$. We first show the functor equality $F_{\dot{F}(\mathrm{id}_\Omega)} = \dot{F} : \mathbb{C}/_\ell\Omega \to \mathbb{D}/_\ell\Pi$.

$$F_{\dot{F}(\mathrm{id}_\Omega)} i = \dot{F}(\mathrm{id}_\Omega) \circ Fi = (Fi)^*(\dot{F}(\mathrm{id}_\Omega)) = \dot{F}(i^*(\mathrm{id}_\Omega)) = \dot{F}i,$$
$$F_{\dot{F}(\mathrm{id}_\Omega)} h = Fh = \dot{F}h.$$

Therefore, $(F, \dot{F}(\mathrm{id}_\Omega))$ is a 1-cell in $\mathbf{CO}(\mathbb{K})$. Next, let $(F, o) \in \mathbf{CO}(\mathbb{K})$ be a 1-cell such that $(F, F_o) = (F, \dot{F})$. Then $\dot{F}(\mathrm{id}_\Omega) = F_o(\mathrm{id}_\Omega) = o$, hence $(F, o) = (F, \dot{F}(\mathrm{id}_\Omega))$. This proves that $L_{\mathbb{K}}$ is bijective on objects. □

We characterize the image of $L_{\mathbb{K}}$ and $L_{\mathbb{K},c}$. Let $p : \mathbb{P} \to \mathbb{C}$ be a $\mathbb{K}$-fibration. A *split generic object* $\Omega \in \mathbb{P}$ is such that for any object $P \in \mathbb{P}$, there exists a unique morphism $\chi : pP \to p\Omega$ in $\mathbb{P}$ such that $P = \chi^*\Omega$ (Jacobs 1999, Definition 5.2.1). Note that split generic objects are mutually isomorphic.

Definition 24. *We write $\mathbb{K}$-$\mathbf{Fib}_g$ (resp. $\mathbb{K}$-$\mathbf{Fib}_{c,g}$) for the 2-category obtained by restricting 0-cells of $\mathbb{K}$-**Fib** (resp. $\mathbb{K}$-$\mathbf{Fib}_c$) to $\mathbb{K}$-fibrations having split generic objects.*

Theorem 25. *For any replete subcategory $\mathbb{K}$ of **Pos**, $\mathbf{CO}(\mathbb{K})$ (resp. $\mathbf{CO}_c(\mathbb{K})$) is biequivalent to $\mathbb{K}$-$\mathbf{Fib}_g$ (resp. $\mathbb{K}$-$\mathbf{Fib}_{c,g}$).*

Proof. We first show that $L_{\mathbb{K}}$ is essentially surjective on 0-cells. Let $p : \mathbb{P} \to \mathbb{C}$ be a $\mathbb{K}$-fibration with a split generic object $\Omega \in \mathbb{P}$. We first extract an ordered object in $\mathbb{C}$ belonging to $\mathbb{K}$. From the definition of split generic object, the function $(-)3^*\Omega : \mathbb{C}(X, p\Omega) \to \mathbf{Obj}(\mathbb{P}_X)$ is a bijection; we write Ψ_X for its inverse. We put a partial order on $\mathbb{C}(X, p\Omega)$ by: $f \leq_X g \iff f^*\Omega \leq g^*\Omega$. Then the bijection becomes an isomorphism between $(\mathbb{C}(X, p\Omega), \leq_X)$ and $\mathbb{P}_X$ in **Pos**, which further becomes an isomorphism in $\mathbb{K}$ because $\mathbb{K}$ is replete. Then for any morphism $f : X \to Y$ in $\mathbb{C}$, the composite of the following morphisms in $\mathbb{K}$ is equal to $\_ \circ f$:

$$(\mathbb{C}(Y, p\Omega), \leq_X) \xrightarrow{(\_)^*\Omega} \mathbb{P}_Y \xrightarrow{f^*} \mathbb{P}_X \xrightarrow{\Psi_X} (\mathbb{C}(X, p\Omega), \leq_Y)$$

Therefore, $p\Omega$ is an ordered object in $\mathbb{C}$ belonging to $\mathbb{K}$. It is then routine to show that $d^{\mathbb{C},p\Omega} : \mathbb{C}/_\ell p\Omega \to \mathbb{C}$ is isomorphic to $p : \mathbb{P} \to \mathbb{C}$ in $\mathbb{K}$-$\mathbf{Fib}_g$. The case of $L_{\mathbb{K},c}$ is proved similarly. □

4.3 Characterizing $\mathbb{K}$-Dijkstra Structures on Domain Fibrations

Recall that a Cartesian $\mathbb{K}$-Dijkstra structure is precisely a monad in $\mathbb{K}$-$\mathbf{Fib}_c$. The local isomorphism theorem (Theorem 23) implies the following characterization of Cartesian $\mathbb{K}$-Dijkstra structures on domain fibrations.

Theorem 26. *Let $(\mathbb{C}, \Omega, \leq) \in \mathbf{CO}_c(\mathbb{K})$ be a 0-cell. Then there is a bijective correspondence between*

(1) A Cartesian $\mathbb{K}$-Dijkstra structure whose $\mathbb{K}$-fibration is $d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}$.
*(2) A pair of a monad T on $\mathbb{C}$ and an EM monotone T-algebra (o, Ω) belonging to $\mathbb{K}$ (when $\mathbb{K} = $ **Pos**, we omit "belonging to $\mathbb{K}$"), that is, an EM algebra such that $o \circ \mathsf{T}\_$ satisfies (12).*

Proof. Let $(\mathbb{C}, \Omega, \leq) \in \mathbf{CO}_c(\mathbb{K})$ be a 0-cell and $(\mathsf{T}, \dot{\mathsf{T}}), (\mu, \dot{\mu}), (\eta, \dot{\eta})$ be a monad on $d^{\mathbb{C},\Omega}$ in the 2-category $\mathbb{K}$-$\mathbf{Fib}_c$. From Theorem 23, there exists a unique 1-cell (T, o) and 2-cells η, μ in $\mathbf{CO}_c(\mathbb{K})$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

whose images by $L_{\mathbb{K},c}$ coincide with $(\mathsf{T}, \dot{\mathsf{T}}), (\mu, \dot{\mu}), (\eta, \dot{\eta})$, respectively. It is then routine to check that $((\mathsf{T}, o), \eta, \mu)$ is a monad on $(\mathbb{C}, \Omega, \leq)$ in $\mathbf{CO}_c(\mathbb{K})$. By unraveling the definition, the monad consists of an ordinary monad (T, η, μ) on $\mathbb{C}$ and an EM T-algebra $o : \mathsf{T}\Omega \to \Omega$ such that $o \circ \mathsf{T}\_$ belongs to $\mathbb{K}$ in the sense of (12). □

In many situations, monads for modeling computational effects are composites of simpler monads through distributive laws. We extend our lifting theory of monads to composite monads. Let T, S be monads on $\mathbb{C}$ and $\alpha : \mathsf{ST} \to \mathsf{TS}$ be a distributive law (see Section 2). We are interested in EM monotone $\mathsf{T} \circ_\alpha \mathsf{S}$-algebras, which give Cartesian liftings of $\mathsf{T} \circ_\alpha \mathsf{S}$. EM algebras of composite monads are studied in Beck (1969, Section 2), Manes and Mulry (2007, Theorem 2.4.3). Below we mildly extend these results.

Theorem 27. *Let $(\mathbb{C}, \Omega, \leq) \in \mathbf{CO}_c(\mathbb{K})$ be a 0-cell, T, S be monads on $\mathbb{C}$ and $\alpha : \mathsf{ST} \to \mathsf{TS}$ be a distributive law. There is a bijective correspondence between each two of:*

(1) An EM monotone $\mathsf{T} \circ_\alpha \mathsf{S}$-algebra (o, Ω) belonging to $\mathbb{K}$.
(2) A pair of an EM monotone T-algebra (t, Ω) belonging to $\mathbb{K}$ and an EM monotone S-algebra (s, Ω) belonging to $\mathbb{K}$ satisfying

$$s \circ \mathsf{S}t = t \circ \mathsf{T}s \circ \alpha_\Omega. \tag{17}$$

(3) A triple $(\dot{\mathsf{T}}, \dot{\mathsf{S}}, \dot{\alpha})$ where $\dot{\mathsf{T}}, \dot{\mathsf{S}}$ are, respectively, Cartesian liftings of T, S along $d^{\mathbb{C},\Omega}$, and $\dot{\alpha} : \dot{\mathsf{S}}\dot{\mathsf{T}} \to \dot{\mathsf{T}}\dot{\mathsf{S}}$ is a distributive law above α whose components are Cartesian.

Proof. The bijection between (2) and (3) is proved in the same way as Theorem 26. We show a bijective correspondence between (1) and (2). Let (o, Ω) be an EM monotone $\mathsf{T} \circ_\alpha \mathsf{S}$-algebra belonging to $\mathbb{K}$. Then it is routine to show that $t \triangleq o \circ T\eta^{\mathsf{S}}_{\Omega} : \mathsf{T}\Omega \to \Omega$ and $s \triangleq o \circ \eta^{\mathsf{T}}_{\mathsf{S}\Omega} : \mathsf{S}\Omega \to \Omega$ are both EM algebras. To show that t, s are monotone and belong to $\mathbb{K}$, observe that functions on the left-hand side below are equal to composite monotone functions on the right-hand side belonging to $\mathbb{K}$:

$$t \circ \mathsf{T}\_ = \ (\mathbb{C}(X, \Omega), \leq_X) \xrightarrow{o \circ \mathsf{TS}\_} (\mathbb{C}(\mathsf{TS}X, \Omega), \leq_{\mathsf{TS}X}) \xrightarrow{\_ \circ \mathsf{T}\eta^{\mathsf{S}}_X} (\mathbb{C}(\mathsf{T}X, \Omega), \leq_{\mathsf{T}X})$$

$$s \circ \mathsf{S}\_ = \ (\mathbb{C}(X, \Omega), \leq_X) \xrightarrow{o \circ \mathsf{TS}\_} (\mathbb{C}(\mathsf{TS}X, \Omega), \leq_{\mathsf{TS}X}) \xrightarrow{\_ \circ \eta^{\mathsf{S}}_{\mathsf{T}X}} (\mathbb{C}(\mathsf{S}X, \Omega), \leq_{\mathsf{S}X})$$

Conversely, let $t : \mathsf{T}\Omega \to \Omega, s : \mathsf{S}\Omega \to \Omega$ be EM monotone algebras belonging to $\mathbb{K}$. Then it is routine to show that $o \triangleq t \circ \mathsf{T}s$ is an EM $\mathsf{T} \circ_\alpha \mathsf{S}$-algebra, provided that t, s satisfies (17). Moreover, o is a monotone and belongs to $\mathbb{K}$ because the function on the left-hand side below is the composite monotone function on the right-hand side, which belongs to $\mathbb{K}$:

$$o \circ \mathsf{TS}\_ = \ (\mathbb{C}(X, \Omega), \leq_X) \xrightarrow{s \circ \mathsf{S}\_} (\mathbb{C}(\mathsf{S}X, \Omega), \leq_{\mathsf{S}X}) \xrightarrow{t \circ \mathsf{T}\_} (\mathbb{C}(\mathsf{TS}X, \Omega), \leq_{\mathsf{TS}X}).$$

□

We note that the condition (17) appears in Beck (1969, Section 2). This theorem says that to identify a lifting of $\mathsf{T} \circ_\alpha \mathsf{S}$, it suffices to identify a pair of EM monotone T- and S-algebras satisfying (17).

We next apply the local isomorphism theorem (Theorem 23) to characterize strengths on Cartesian Dijkstra structures whose fibrations are domain fibrations. We first identify the situation when domain fibrations have finite products.

Proposition 28. *Let* $(\mathbb{C}, \Omega, \leq) \in \mathbf{CO}_c(\mathbf{Pos})$ *be a 0-cell. The following are equivalent:*

(1) The domain fibration $d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}$ *has finite products* $(1, \dot{1}, \times, \dot{\times})$.
(2) $\mathbb{C}$ *has finite products* $(1, \times)$, *and there are morphisms* $\top : 1 \to \Omega$, $m : \Omega^2 \to \Omega$ *such that* $\top \circ !_X$ *and* $m \circ \langle\_, \_\rangle$ *give finite meets in the poset* $(\mathbb{C}(X, \Omega), \leq_X)$.

Theorem 29. *Let* $(\mathbb{C}, \Omega, \leq) \in \mathbf{CO}_c(\mathbf{Pos})$ *be a 0-cell, and suppose that* $d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}$ *has finite products* $(1, \dot{1}, \times, \dot{\times})$. *We write* $m : \Omega^2 \to \Omega$ *for the meet morphism that exists by Proposition 28. Let* $(d^{\mathbb{C},\Omega}, \mathsf{T}, \dot{\mathsf{T}})$ *be a Cartesian Dijkstra structure and let* $o : \mathsf{T}\Omega \to \Omega$ *be the EM monotone* T*-algebra corresponding to* $\dot{\mathsf{T}}$. *There is a bijective correspondence between:*

(1) a strength on the Cartesian Dijkstra structure, and
(2) a strength θ *on* T *such that* $m \circ (\mathrm{id} \times o) \leq o \circ \mathsf{T}m \circ \theta_{\Omega,\Omega}$.

Proof. Let $(\theta, \dot{\theta})$ be a strength on a Cartesian Dijkstra structure. From Lemma 12, and the assumption that both $\dot{\times}$ and T are fibered, the strength bijectively corresponds to the 2-cell $(\theta, \dot{\theta})$ in **Pos**-**Fib** such that θ is a strength on T.

$$
\begin{tikzcd}
d^{\mathbb{C},\Omega} \times d^{\mathbb{C},\Omega} \arrow[r, "(\mathrm{Id}\times\mathsf{T},\mathrm{Id}\times\dot{\mathsf{T}})"] \arrow[d, "(\times,\dot{\times})"'] \arrow[dr, phantom, "\Downarrow(\theta,\dot{\theta})"] & d^{\mathbb{C},\Omega} \times d^{\mathbb{C},\Omega} \arrow[d, "(\times,\dot{\times})"] \\
d^{\mathbb{C},\Omega} \arrow[r, "(\mathsf{T},\dot{\mathsf{T}})"'] & d^{\mathbb{C},\Omega}
\end{tikzcd}
\qquad
\begin{tikzcd}
(\mathbb{C}^2, (\Omega, \Omega), \leq^2) \arrow[r, "(\mathrm{Id}\times\mathsf{T},(\mathrm{id},o))"] \arrow[d, "(\times,m)"'] \arrow[dr, phantom, "\Downarrow\theta"] & (\mathbb{C}^2, (\Omega, \Omega), \leq^2) \arrow[d, "(\times,m)"] \\
(\mathbb{C}, \Omega, \leq) \arrow[r, "(\mathsf{T},o)"'] & (\mathbb{C}, \Omega, \leq)
\end{tikzcd}
$$

By the local isomorphism theorem, such a 2-cell bijectively corresponds to a 2-cell in **CO**(**Pos**) (see the right diagram above) where θ is a strength of T. This is equivalent to that θ is a strength on T satisfying $m \circ (\mathrm{id} \times o) \leq o \circ \mathsf{T}m \circ \theta$. □

4.4 Algebraic Operations and Weakest Precondition Predicate Transformer

Various program constructs that cause computational effects can be naturally modeled by Plotkin and Power's *algebraic operations* (Plotkin and Power 2001). They frequently occur in the monadic semantics of effectful programs. In this section, we give a wppt semantics of algebraic operations in the Cartesian Dijkstra structures whose fibrations are domain fibrations.

In this section we fix a Cartesian category $(\mathbb{C}, 1, \times)$, an ordered object $(\Omega, \leq)$ in $\mathbb{C}$ belonging to a subcategory $\mathbb{K}$ of **Pos**, and a strong monad $(\mathsf{T}, \eta, \mu, \theta)$.

Definition 30. *A morphism* $g : A \to \mathsf{T}B$ *in* $\mathbb{C}$ *is called* generic effect *(Plotkin and Power 2001). For an EM* T*-algebra* $o : \mathsf{T}X \to X$, *we define the following operation* $\mathrm{op}(o, g) : \mathbb{C}(Y, A) \times \mathbb{C}(Y \times B, X) \to \mathbb{C}(Y, X)$ *by*

$$\mathrm{op}(o, g)(a, b) \triangleq o \circ \mathsf{T}(b) \circ \theta_{Y,B} \circ \langle \mathrm{id}_Y, g \circ a\rangle.$$

We specifically write $\alpha(g)_X$ *for* $\mathrm{op}(\mu_X, g)$ *and call it the* algebraic operation *corresponding to* g.

The first and second arguments of $\mathrm{op}(o, g)$ are, respectively, called a *parameter* and an *argument*. The above definition of algebraic operation is different from the original one presented in Plotkin and Power (2001). Originally, an algebraic operation corresponding to a generic effect $g : A \to \mathsf{T}B$ is a certain natural transformation $\alpha_X : A \times (B \Rightarrow \mathsf{T}X) \to \mathsf{T}X$. In this paper, we present it as an operation on morphisms in a Cartesian category.

Proposition 31. *Let* $o : \mathsf{T}\Omega \to \Omega$ *be an EM monotone* T*-algebra belonging to* $\mathbb{K}$*. Then for any generic effect* $g : A \to \mathsf{T}B$ *and morphism* $a : Y \to A$*, we have*

$$\mathrm{op}(o, g)(a, \_) \in \mathbb{K}((\mathbb{C}(Y \times B, \Omega), \leq_{Y\times B}), (\mathbb{C}(Y, \Omega), \leq_Y)).$$

Proof. Notice that $\mathrm{op}(o, g)(a, \_)$ is the composite of the following morphisms in $\mathbb{K}$:

$$(\mathbb{C}(Y \times B, \Omega), \leq_{Y\times B}) \xrightarrow{o\circ\mathsf{T}\_} (\mathbb{C}(\mathsf{T}(Y \times B), \Omega), \leq_{\mathsf{T}(Y\times B)}) \xrightarrow{\_\circ\langle \mathrm{id}_Y, g\circ a\rangle} (\mathbb{C}(Y, \Omega), \leq_Y).$$

□

The following theorem helps us (in Section 7) to compute the wppt of programs involving algebraic operations:

Theorem 32. *Let* $o : \mathsf{T}\Omega \to \Omega$ *be an EM monotone* T*-algebra belonging to* $\mathbb{K}$ *and* $(d^{\mathbb{C},\Omega}, \mathsf{T}, \mathsf{T}_o)$ *be the corresponding Cartesian* $\mathbb{K}$*-Dijkstra structure. Then the following equality holds for any generic effect* $g : A \to \mathsf{T}B$ *and morphism* $a : Y \to A$, $b : Y \times B \to \mathsf{T}X$, $i : X \to \Omega$*:*

$$\mathrm{wp}(\alpha(g)_X(a, b), i) = \mathrm{op}(o, g)(a, \mathrm{wp}(b, i)).$$

Proof. The equation easily follows from the EM axiom $o \circ \mu_\Omega = o \circ \mathsf{T}o$:

$$\begin{aligned}
\mathrm{wp}(\alpha(g)_X(a, b), i) &= o \circ \mathsf{T}i \circ \mu_X \circ \mathsf{T}(b) \circ \theta \circ \langle \mathrm{id}, g \circ a\rangle \\
&= o \circ \mu_\Omega \circ \mathsf{T}^2 i \circ \mathsf{T}(b) \circ \theta \circ \langle \mathrm{id}, g \circ a\rangle \\
&= o \circ \mathsf{T}o \circ \mathsf{T}^2 i \circ \mathsf{T}(b) \circ \theta \circ \langle \mathrm{id}, g \circ a\rangle \\
&= \mathrm{op}(o, g)(a, \mathrm{wp}(b, i)).
\end{aligned}$$

□

When $\mathbb{C}$ has distributive finite coproducts, we can make algebraic operations to take tuples of morphisms as arguments. Below we look at the case $n = 2$. We write 2 for the coproduct $1 + 1$. For the setting above, it is useful to define an operator $\beta : \mathbb{C}(X, Y) \times \mathbb{C}(X, Y) \to \mathbb{C}(X \times 2, Y)$:

$$\beta(b_1, b_2) \triangleq [b_1 \circ r_X, b_2 \circ r_X] \circ \mathrm{dist}_{Y,1,1} \tag{18}$$

where $r_X : X \times 1 \to X$ is the right unitor and $\mathrm{dist}_{X,Y,Z} \circ X \times (Y + Z) \to X \times Y + X \times Z$ is the *distributive law* of the product over the coproduct. Then we can show:

Corollary 33. *Let* $o : \mathsf{T}\Omega \to \Omega$ *be an EM monotone* T*-algebra belonging to* $\mathbb{K}$ *and* $(d^{\mathbb{C},\Omega}, \mathsf{T}, \mathsf{T}_o)$ *be the corresponding Cartesian* $\mathbb{K}$*-Dijkstra structure. For any generic effect* $g : A \to \mathsf{T}2$ *and morphism* $a : Y \to A$, $b_1, b_2 : Y \to \mathsf{T}X$, $i : X \to \Omega$*, the following equality holds:*

$$\mathrm{wp}(\alpha(g)_X(a, \beta(b_1, b_2)), i) = op(o, g)(a, \beta(\mathrm{wp}(b_1, i), \mathrm{wp}(b_2, i))).$$

Proof. The key step is checking that $\mathrm{wp}(\beta(b_1, b_2), i) = \beta(\mathrm{wp}(b_1, i), \mathrm{wp}(b_2, i))$. Indeed,

$$\begin{aligned}
\mathrm{wp}(\beta(b_1, b_2), i) &= o \circ \mathsf{T}i \circ [b_1 \circ r, b_2 \circ r] \circ \mathrm{dist} \\
&= [o \circ \mathsf{T}i \circ b_1 \circ r, o \circ \mathsf{T}i \circ b_2 \circ r] \circ \mathrm{dist} \\
&= \beta(\mathrm{wp}(b_1, i), \mathrm{wp}(b_2, i)).
\end{aligned}$$

□

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

5. Dijkstra Structures on $d^{\mathbf{Set},\mathbf{2}}$

We apply the techniques developed in the previous sections to concrete settings. We first consider the lax slice category with the poset $\mathbf{2} = \{\bot \leq \top\}$, viewed as an ordered object in **Set** (Example 18). We identify an object $i : X \to \mathbf{2}$ in $\mathbf{Set}/_\ell \mathbf{2}$ as a pair (X, P) of sets such that $P \subseteq X$. The fibration $d^{\mathbf{Set},\mathbf{2}} : \mathbf{Set}/_\ell \mathbf{2} \to \mathbf{Set}$ is isomorphic to the subobject fibration (see e.g. Jacobs 1999, Chapter 0) of **Set**.

Example 34. (Hasuo 2015, Section 2.3) As a warm-up exercise, we consider Cartesian liftings of the *maybe monad* (a.k.a. partiality monad) M along $d^{\mathbf{Set},\mathbf{2}}$. Its functor part is defined by $\mathsf{M}X \triangleq X + \{*\}$, where $\{*\}$ denotes the singleton set. In this example, we leave implicit the coproduct injection $\iota_i : X_i \to X_1 + X_2$ to improve readability.

There are exactly two EM monotone M-algebras tot, par over $\mathbf{2}$:

$$\mathrm{tot}(x) = \top \iff x = \top$$
$$\mathrm{par}(x) = \top \iff (x = \top \vee x = *).$$

They induce Cartesian liftings $\mathsf{M}_{\mathrm{tot}}, \mathsf{M}_{\mathrm{par}}$ of M along $d^{\mathbf{Set},\mathbf{2}}$. The following table describes how the liftings are defined on objects and how the Hoare triple statements are interpreted in the corresponding Cartesian Dijkstra structures using these liftings:

o	Lifting M_o, object part	$(X, P)\{f\}(Y, Q)$ in $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{M}, \mathsf{M}_o)$
tot	$\mathsf{M}_{\mathrm{tot}}(X, P) = (\mathsf{M}X, P)$	$\forall x \in X \,.\, x \in P \implies f(x) \neq * \wedge f(x) \in Q$
par	$\mathsf{M}_{\mathrm{par}}(X, P) = (\mathsf{M}X, P \cup \{*\})$	$\forall x \in X \,.\, x \in P \wedge f(x) \neq * \implies f(x) \in Q$

The Cartesian Dijkstra structures associated to these Cartesian liftings offer the *total* and *partial* correctness interpretations of Hoare triples.

Example 35. Generalizing Example 34, we consider Cartesian liftings of the *exception monad* E along $d^{\mathbf{Set},\mathbf{2}}$. Its functor part is defined by $\mathsf{E}X \triangleq X + E$, where E is a fixed set. We can easily verify that an EM monotone E-algebras over $\mathbf{2}$ bijectively corresponds to a subset of E. For a subset $A \subseteq E$, the corresponding algebra $\mathrm{exc}_A : \mathsf{E}\mathbf{2} \to \mathbf{2}$ is

$$\mathrm{exc}_A(x) = \top \iff x = \iota_1(\top) \vee \exists e \in A \,.\, x = \iota_2(e).$$

Let us write $\mathsf{E}_{\mathrm{exc}_A}$ for the associated Cartesian lifting of E. Its object part is given by

$$\mathsf{E}_{\mathrm{exc}_A}(X, P) = (\mathsf{E}X, \{\iota_1(x) \mid x \in P\} \cup \{\iota_2(e) \mid e \in A\}),$$

and the Hoare triple $(X, P)\{f\}(Y, Q)$ in $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{E}, \mathsf{E}_{\mathrm{exc}_A})$ is equivalent to

$$\forall x \in X \,.\, x \in P \implies (\exists y \in Q \,.\, f(x) = \iota_1(y)) \vee (\exists e \in A \,.\, f(x) = \iota_2(e)).$$

This statement says that f returns a value satisfying Y, or raise an exception belonging to A.

Example 36. (Hino et al. 2016, Example 3.3) We next consider the Cartesian liftings of the *nonempty powerset monad* P^+ along $d^{\mathbf{Set},\mathbf{2}}$. Its functor part is defined by $\mathsf{P}^+X \triangleq \{U \subseteq X \mid U \neq \emptyset\}$. There are exactly two EM monotone P^+-algebras may, must over $\mathbf{2}$:

$$\mathrm{may}(U) = \top \iff \top \in U,$$
$$\mathrm{must}(U) = \top \iff \bot \notin U$$

They induce two Cartesian liftings $\mathsf{P}^+{}_{\mathrm{may}}, \mathsf{P}^+{}_{\mathrm{must}}$ of P^+ along $d^{\mathbf{Set},\mathbf{2}}$. The Dijkstra structures with these Cartesian liftings of P^+ along $d^{\mathbf{Set},\mathbf{2}}$ offer the *may* and *must* correctness interpretations of Hoare triples.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

o	Lifting $\mathsf{P}^+{}_o$, object part	$(X,P)\{f\}(Y,Q)$ in $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{P}^+, \mathsf{P}^+{}_o)$
may	$\mathsf{P}^+{}_{\mathrm{may}}(X,P) = (\mathsf{P}^+X, \{U \mid U \cap P \neq \emptyset\})$	$\forall x \in X . x \in P \implies \exists u \in f(x) . u \in Q$
must	$\mathsf{P}^+{}_{\mathrm{must}}(X,P) = (\mathsf{P}^+X, \{U \mid U \subseteq P\})$	$\forall x \in X . x \in P \implies \forall u \in f(x) . u \in Q$

Example 37. We consider again the monad P^+ of the previous example, but now we regard $d^{\mathbf{Set},\mathbf{2}}$ as a $\mathbf{CLat}_\wedge$-fibration. Then must above is the only EM monotone P^+-algebra over $\mathbf{2}$ belonging to $\mathbf{CLat}_\wedge$. Therefore the *must* correctness interpretation of the Hoare triple has the composable strongest postcondition predicate transformer (see Section 3.5).

Example 38. We consider the state monad S, whose functor part is defined by $\mathsf{S}\,X = S \Rightarrow (X \times S)$; here S is the set of states. Recall that a computation $f\colon X \to \mathsf{S}Y$ can be seen as a procedure from X to Y that has access to an external memory device with states in S. However, the two-point set $\mathbf{2}$ is too small to remember stored values when $|S| \geq 2$. Indeed,

Theorem 39. *When $|S| \geq 2$, there is no EM monotone S-algebra over $\mathbf{2}$.*

Proof. Plotkin and Power (2001) observed that algebras of the set monad correspond bijectively to mnemoids, sets that satisfy the algebraic theory of update-lookup. For the carrier 2, these operations have the types:

$$lookup : \mathbf{2}^S \to \mathbf{2} \qquad\qquad update : S \times \mathbf{2} \to \mathbf{2}$$

and satisfy the equations below:

$$\forall s \in S. \forall b \in \mathbf{2}.\ lookup(\lambda s.update(s,b)) = b \tag{19}$$

$$\forall s_1, s_2 \in S. \forall b \in \mathbf{2}.\ update(s_1, update(s_2, b)) = update(s_2, b) \tag{20}$$

$$\forall s \in S. \forall f \in \mathbf{2}^S.\ update(s, lookup(f)) = f(s) \tag{21}$$

The algebraic theory implies that if $update(s,b) = 1$ for any s and b, then $update(t,1) = 1$ for all t, since

$$update(t,1) = update(t, update(s,b)) = update(s,b) = 1$$

by Equation 20. An analogous reasoning can be done when $update(s,b) = 0$. This leaves two cases, either $update(t,0) = 0$ and $update(t,1) = 1$ for all t, or $update(t,0) = update(t,1)$ for all t.

In the second case, we derive a contradiction easily:

$$1 = lookup(\lambda t.update(t,1)) = lookup(\lambda t.update(t,0)) = 0$$

where the first and last equalities are given by Equation (19).

For the first case, we need a further subcase distinction. Let s be a state and $f_s \in 2^S$ the function that returns 1 on s and 0 otherwise. Either $lookup(f_s) = 0$ or $lookup(f_s) = 1$. If it is 0, then

$$1 = update(s,1) = update(s, lookup(f_s)) = update(s,0) = 0$$

Here, the second equality comes from applying Equation (21).

The subcase $lookup(f_s) = 1$ is analogous. This completes the proof. □

When we replace the slicing object $\mathbf{2}$ with $S \Rightarrow \mathbf{2}$ (with the pointwise order), we find that it carries an EM monotone S-algebra $o(k) \triangleq \lambda s . \pi_1(k\,s)(\pi_2(k\,s))$; see also Pitts (1991, Example 3.4.2) and Maillard et al. (2019, Section 4.4). The corresponding Cartesian lifting of S along $d^{\mathbf{Set},(S\Rightarrow\mathbf{2})}$: $\mathbf{Set}/_\ell(S \Rightarrow \mathbf{2}) \to \mathbf{Set}$ and the derived wppt satisfies

$$\mathsf{S}_o(i)(c) = \lambda s . i(\pi_1(cs))(\pi_2(cs)), \quad \mathrm{wp}(f,i)(x) = \lambda s . i(\pi_1(f(x)(s)))(\pi_2(f(x)(s))).$$

By using the uncurrying i' and f' of i and f, respectively, wppt can be simplified: $\mathrm{wp}(f, i)(x) = \lambda s\ .\ i'(f'(x, s))$. The Hoare triple $(X, i)\{f\}(Y, j)$ is valid iff for all $x \in X$ and $s \in S$, $i(x, s)$ implies $j(f(x, s))$.

In this setting, the strength cannot be lifted to the slice category. The following counterexample (Kura 2022, Example 3.4.11) shows that the conditions of Theorem 29 are not satisfied. Let s_1, s_2 be distinct states, $x \triangleq \lambda s.(s = s_1) : S \Rightarrow \mathbf{2}$ and $t \triangleq \lambda s.\langle(\lambda s'.(s' = s_2)), s_2\rangle : \mathsf{S}(S \Rightarrow \mathbf{2})$. Then, $x \wedge o(t) = \lambda s.(s = s_1) \not\leq \lambda s.\bot = (o \circ \mathsf{S}(\wedge) \circ \theta^{\mathsf{S}})(x, t)$.

Example 40. Finally, we consider the finite probability distribution monad D. A *finite probability distribution* on a set X is a function $\mu : X \to [0, 1]$ such that μ is nonzero at finitely many elements in X, and $\sum_{i \in X} \mu(i) = 1$. The functor part of the monad D is defined by

$$\mathsf{D}X \triangleq \{\mu \text{ is a finite probability distribution on } X\}.$$

The unit $\eta^{\mathsf{D}}(x)$ is the Dirac distribution at x, which has value 1 at x and 0 everywhere else, and the Kleisli lifting $f^{\#}$ for $f : X \to \mathsf{D}Y$ is given by $f^{\#}(\mu)(y) \triangleq \sum_{x \in X} f(x)(y)\mu(x)$. Given $x_1, x_2 \in X$ and $\rho \in (0, 1)$, the convex combination $(x_1 \oplus_\rho x_2)$ is the probability distribution with value ρ on x_1, $(1 - \rho)$ on x_2 and 0 everywhere else.

Theorem 41. *There are exactly two EM monotone* D*-algebras over* $\mathbf{2}$*, given by*

$$\mathrm{pmay}(\mu) = \top \iff \mu(\top) > 0,$$
$$\mathrm{pmust}(\mu) = \top \iff \mu(\top) = 1$$

Proof. Let $h : \mathsf{D}2 \to 2$ be a D-algebra. Then it must satisfy:

$$h(\eta(0)) = 0 \qquad h(\eta(1)) = 1 \qquad h(p \oplus_\alpha q) = h(h(p) \oplus_\alpha h(q))$$

Assume $h(p) = 1$ for some $p \in (0, 1)$. Then, by the third and second conditions we have, for every $\alpha \in [0, 1]$,

$$h(p \oplus_\alpha 1) = h(h(p) \oplus_\alpha h(1)) = h(1) = 1$$

In other words, for every $q \in [p, 1], f(q) = 1$.

On the other hand, also by the third and first conditions

$$h(p \oplus_p 0) = h(h(p) \oplus_p h(0) = h(p^2) = 1.$$

With these two results combined, we get that $f(q) = 1$ for every $q \in [0, 1]$. □

They induce two Cartesian liftings $\mathsf{D}_{\mathrm{pmay}}, \mathsf{D}_{\mathrm{pmust}}$ of D along $d^{\mathbf{Set},\mathbf{2}}$. The predicate transformers associated to them generalize the *may* and *must* correctness of nondeterministic programs to the probabilistic setting.

o	Lifting D_o, object part	$(X, P)\{f\}(Y, Q)$ in $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{D}, \mathsf{D}_o)$
pmay	$\mathsf{D}_{\mathrm{pmay}}(X, P) = (\mathsf{D}X, \{\mu \mid \exists x \in P\ .\ \mu(x) > 0\})$,	$\forall x \in X\ .\ x \in P \implies \mathsf{Pr}_{y \sim f(x)}[y \in Q] > 0$
pmust	$\mathsf{D}_{\mathrm{pmust}}(X, P) = (\mathsf{D}X, \{\mu \mid \forall x \in P\ .\ \mu(x) > 0\})$	$\forall x \in X\ .\ x \in P \implies \mathsf{Pr}_{y \sim f(x)}[y \in Q] = 1$

Here the notation $\mathsf{Pr}_{y \sim f(x)}[y \in Q]$ denotes the probability of $y \in Q$ when sampling y from $f(x)$. Explicitly, this is defined as

$$\mathsf{Pr}_{y \sim f(x)}[y \in Q] \triangleq \sum_{y \in Y} \mathbf{1}_Q(y) \cdot f(x)(y)$$

where $\mathbf{1}_Q$ is the characteristic function of Q.

We note that Cartesian liftings are rather rare, compared to arbitrary liftings. We look at the case of the powerset monad P along $d^{\mathbf{Set},\mathbf{2}}$. For each regular cardinal λ, $\mathsf{P}_\lambda(P \subseteq X) \triangleq \{U \subseteq P \mid |U| < \lambda\}$ gives a lifting of the powerset monad P along $d^{\mathbf{Set},\mathbf{2}}$. On the other hand (a modification of), Example 36 shows that there are only two Cartesian liftings of P along $d^{\mathbf{Set},\mathbf{2}}$. Example 38 also shows that there is *no* Cartesian lifting of the nontrivial state monad along $d^{\mathbf{Set},\mathbf{2}}$.

We end this section with a discussion about strengths on Dijkstra structures of the form $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{T}, \dot{\mathsf{T}})$. First, the fibration $d^{\mathbf{Set},\mathbf{2}} : \mathbf{Set}/_\ell\mathbf{2} \to \mathbf{Set}$ has finite products $(1, \dot{1}, \times, \dot{\times})$. Second, every monad (T, η, μ) on $\mathbf{Set}$ has a unique strength θ given by

$$\theta_{X,Y}(x, c) = (\lambda y \in Y.\eta_{X\times Y}(x, y))^{\#}(c); \tag{22}$$

see Moggi (1991). We then obtain the following:

Proposition 42. *Every Dijkstra structure* $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{T}, \dot{\mathsf{T}})$ *has a unique strength* $(\theta, \dot{\theta})$.

Proof. From Theorem 29, it suffices to show $x \wedge o(c) \leq o \circ T(\lambda y.x \wedge y)(c)$ for any $x \in \mathbf{2}$ and $c \in \mathsf{T}\mathbf{2}$. It suffices to check whether this holds for $x = \bot$ and $x = \top$; both cases trivially hold. □

6. Dijkstra Structures on $d^{\mathbf{Set},[0,\infty]}$

We next replace the poset $\mathbf{2} = \{\bot \leq \top\}$ with the poset $[0, \infty]$ of nonnegative extended reals, with the usual numerical order. It is a completely distributive lattice, and the domain fibration $d^{\mathbf{Set},[0,\infty]} : \mathbf{Set}/_\ell[0, \infty] \to \mathbf{Set}$ has a rich structure (in fact it is a tripos Pitts 1999, Example 2.2). An object of $\mathbf{Set}/_\ell[0, \infty]$ is a function $i : X \to [0, \infty]$.

Example 43. We consider the *counting monad* C over $\mathbf{Set}$, whose functor part is given by $\mathsf{C}X \triangleq \mathbb{N} \times X$. This is identical to the writer monad over a single alphabet ($\mathbb{N} \cong 1^*$). Each monotone function $f : [0, \infty] \to [0, \infty]$ (bijectively) determines an EM monotone C-algebra $o(f)(n, x) \triangleq f^n(x)$, and the corresponding Cartesian lifting $\mathsf{C}_{o(f)}$ of C along $d^{\mathbf{Set},[0,\infty]}$ is given by $\mathsf{C}_{o(f)}(i)(n, x) = f^n(i(x))$. The standard choice of $f : [0, \infty] \to [0, \infty]$ is the successor function $s(x) = x + 1$. In this case, the lifting becomes $\mathsf{C}_{o(s)}i = \lambda(n, x) \,.\, n + i(x)$.

Example 44. We consider Cartesian liftings of the nonempty powerset monad P^+ along $d^{\mathbf{Set},[0,\infty]}$. There are at least two EM monotone P^+-algebras $\sup, \inf : \mathsf{P}^+[0, \infty] \to [0, \infty]$. They induce two Cartesian liftings:

$$\mathsf{P}^+{}_{\sup}(i) = \lambda U \,.\, \sup_{x \in U} i(x) \qquad (i : X \to [0, \infty])$$
$$\mathsf{P}^+{}_{\inf}(i) = \lambda U \,.\, \inf_{x \in U} i(x) \qquad (i : X \to [0, \infty])$$

These Cartesian liftings can be seen as quantitative generalizations of the may and must liftings from Example 36.

Example 45. We can view objects in $\mathbf{Set}/_\ell[0, \infty]$ as random variables. The expected value of a random variable $i : X \to [0, \infty]$ over a finite probability distribution $\mu \in \mathsf{D}X$, denoted by $E_{x\sim\mu}[i(x)]$, is computed as $\mathrm{E} \circ \mathsf{D}i(\mu)$, where $\mathrm{E} : \mathsf{D}[0, \infty] \to [0, \infty]$ is the expected value function on $[0, \infty]$. It is easy to check that E is an EM monotone D-algebra over $[0, \infty]$. Therefore, the expected value *is* a Cartesian lifting of D along $d^{\mathbf{Set},[0,\infty]}$:

$$\mathsf{D}_{\mathrm{E}}(i) \triangleq \lambda\mu \,.\, E_{x\sim\mu}[i(x)] = \lambda\mu \,.\, \sum i(x) \cdot \mu(x). \quad (i : X \to [0, \infty])$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

The wppt of the Cartesian Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, \mathsf{D}, \mathsf{D}_{\mathrm{E}})$ computes the *weakest pre-expectation* (McIver and Morgan 2005):

$$\mathrm{wp}(f,i) = \lambda j\,.\,E_{x\sim f(j)}[i(x)]. \quad (f: Y \to \mathsf{D}X, i: X \to [0,\infty])$$

We point out that the Cartesian Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, \mathsf{D}, \mathsf{D}_{\mathrm{E}})$ does not have a strength, since the strength for the finite probability distribution monad does not satisfy the conditions of Theorem 29. The strength $\theta^{\mathsf{D}}_{X,Y}$ is defined as

$$\theta^{\mathsf{D}}_{X,Y}(x,\nu)(x',y) = \begin{cases} \nu(y) & x' = x \\ 0 & \text{otherwise.} \end{cases}$$

The conditions of the Theorem require that for every $x \in [0,\infty]$ and every $\nu \in \mathsf{D}([0,\infty])$, $\min(x, \mathrm{E}(\nu)) \le \mathrm{E}_{y\sim\nu}[\min(x,y)]$. However, we can find a counterexample (see also Kura 2022, Example 3.4.10) by setting $x = 1$ and ν such that $\nu(0) = \nu(2) = \frac{1}{2}$. Then, we have $\min(1, \mathrm{E}(\nu)) = 1 \not\le \frac{1}{2} = \mathrm{E}_{y\sim\nu}[\min(1,y)]$.

Example 46. When modeling a programming language with both nondeterministic choice and probabilistic choice, we might want to combine P^+ and D via a distributive law $\mathsf{D} \circ \mathsf{P}^+ \to \mathsf{P}^+ \circ \mathsf{D}$. However, Plotkin showed that there is *no* such distributive law (Varacca and Winskel 2006). To remedy this, Varacca and Winskel introduced the *indexed valuation monad* (Varacca and Winskel 2006), which does have a distributive law with P^+.

In this paper, we employ its finite variant called *finite indexed distribution monad* (Sato 2011). A *finite indexed distribution* over a set X is a finite multiset of pairs (p,x) where $p \in (0,1]$ and $x \in X$, and moreover, the sum of ps in the multiset is 1. We define

$$\mathsf{I}(X) \triangleq \{\mu : \text{finite indexed distribution over } X\}.$$

The probabilistic sum of finite indexed distributions is defined as follows (below @ denotes multiset union):

$$[\cdots(p_i,x_i)\cdots] \oplus_p [\cdots(q_j,y_j)\cdots] \triangleq \begin{cases} [\cdots(q_j,y_j)\cdots] & (p=0) \\ [\cdots(p\cdot p_i,x_i)\cdots]@[\cdots((1-p)q_j,y_j)\cdots] & p\in(0,1) \\ [\cdots(p_i,x_i)\cdots] & (p=1) \end{cases}$$

Every finite probability distribution may be regarded as a finite indexed distribution; we write the inclusion map by $\iota_X : \mathsf{D}(X) \to \mathsf{I}(X)$. We can extend I to a monad on **Set** called the *finite indexed distribution monad.* The concept of expectation can also be extended to finite indexed distributions. First, the function $\mathrm{IE} : \mathsf{I}([0,\infty]) \to [0,\infty]$ defined below gives an EM monotone I-algebra over $[0,\infty]$:

$$\mathrm{IE}[(p_1,x_1),\cdots,(p_n,x_n)] \triangleq \sum_{i=1}^{n} p_i \cdot x_i.$$

We write $IE_{x\sim\nu}[i(x)]$ to mean $\mathrm{IE}\circ(\mathsf{I}(i)(\nu))$. We note that $\mathrm{E} = \mathrm{IE}\circ\iota$. The corresponding Cartesian lifting of I along $d^{\mathbf{Set},[0,\infty]}$, and the wppt within the derived Cartesian Dijkstra structure is

$$\mathsf{I}_{\mathrm{IE}}(i)(\nu) \triangleq IE_{x\sim\nu}[i(x)] = \sum_{(p,x)\in\nu} p\cdot i(x), \quad \mathrm{wp}(f,i)(x) = \sum_{(p,y)\in f(x)} p\cdot i(y).$$

6.1 Dijkstra Structures for Composite Monads

We next visit examples of liftings of composite monads via distributive laws.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Example 47. We have seen all the Cartesian liftings of the Maybe monad M (Example 34) and the nonempty powerset monad P^+ (Example 36) along $d^{\mathbf{Set},2}$. We next derive Cartesian liftings of the composite monad $\mathsf{P}^+ \circ_\beta \mathsf{M}$ via the distributive law $\beta : \mathsf{M} \circ \mathsf{P}^+ \to \mathsf{P}^+ \circ \mathsf{M}$ given by $\beta(*) = \{*\}$ and $\beta(U) = U$. The composite monad can model the computational effects of both nondeterministic choice and diverging computations. To give Cartesian liftings of $\mathsf{P}^+ \circ_\beta \mathsf{M}$ along $d^{\mathbf{Set},2}$, it suffices to show (17) for all the four combinations of EM monotone P^+- and M-algebras. They all satisfy (17).

Theorem 48. *There are exactly four Cartesian liftings of* $\mathsf{P}^+ \circ_\beta \mathsf{M}$ *along* $d^{\mathbf{Set},2}$.

Let us denote these liftings by $\mathsf{P}^+{}_a \circ_\beta \mathsf{M}_b$, where (a, b) range over the four-point set $\{\mathrm{may}, \mathrm{must}\} \times \{\mathrm{tot}, \mathrm{par}\}$. Each lifting is explicitly described as follows:

$$\mathsf{P}^+{}_a \circ_\beta \mathsf{M}_b(I, X) = \left(\mathsf{P}^+(\mathsf{M}(I)), \begin{cases} \{U \mid U \cap X \neq \emptyset\} & (a = \mathrm{may}, b = \mathrm{tot}) \\ \{U \mid U \subseteq X\} & (a = \mathrm{must}, a = \mathrm{tot}) \\ \{U \mid U \cap (X + \{*\}) \neq \emptyset\} & (a = \mathrm{may}, b = \mathrm{par}) \\ \{U \mid U \subseteq X + \{*\}\} & (a = \mathrm{must}, b = \mathrm{par}) \end{cases} \right)$$

Next, we look at examples of composite monads with the counting monad C in Example 43. The counting monad is useful for measuring various intensional quantities of program executions. The distributive law of any **Set**-monad T over the counting monad C exists and is provided by the unique strength $\theta_{X,Y} : X \times \mathsf{T}Y \to \mathsf{T}(X \times Y)$ of T given as (22). Here we discuss giving a Cartesian lifting of $\mathsf{T} \circ_\theta \mathsf{C}$.

Theorem 49. *Let* $o : \mathsf{T}[0, \infty] \to [0, \infty]$ *be an EM monotone* T*-algebra and* $f : [0, \infty] \to [0, \infty]$ *be a monotone function. Then* o *and* $o(f)$ *(the EM monotone* C*-algebra associated to* f*; see Example 43) satisfy (17) if and only if* $o \circ \mathsf{T}f = f \circ o$*, that is,* f *is a* T*-algebra endomorphism over* o*.*

Example 50. We consider the instance of Theorem 49 where T is the finite probability distribution monad D. The composite monad $\mathsf{D} \circ_\theta \mathsf{C}$ is suitable for modeling probabilistic programs that report cost counting during their executions. We will see this in Section 7. The EM monotone D-algebra $\mathrm{E} : \mathsf{D}[0, \infty] \to [0, \infty]$ and the successor function $s : [0, \infty] \to [0, \infty]$ evidently satisfies the equality condition of Theorem 49:

$$\mathrm{E} \circ \mathsf{D}s = \lambda\mu \,.\, E_{x\sim\mu}[1 + x] = \lambda\mu \,.\, 1 + E_{x\sim\mu}[x] = s \circ \mathrm{E}.$$

Therefore, E and $o(s)$ satisfy (17) by Theorem 49. The composite $\mathsf{D}_{\mathrm{E}} \circ \mathsf{C}_{o(s)}$ is a Cartesian lifting of $\mathsf{D} \circ_\theta \mathsf{C}$ along $d^{\mathbf{Set},[0,\infty]}$ and is computed as $\mathsf{D}_{\mathrm{E}} \circ \mathsf{C}_{o(s)}(i) = \lambda\mu \,.\, E_{(c,x)\sim\mu}[c + i(x)]$. The wppt in the Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, \mathsf{D} \circ_\theta \mathsf{C}, \mathsf{D}_{\mathrm{E}} \circ \mathsf{C}_{o(s)})$ is

$$\mathrm{wp}(f, j)(x) = E_{(c,y)\sim f(x)}[c + j(y)] \quad (f : X \to \mathsf{D} \circ_\theta \mathsf{C}(Y), j : Y \to [0, \infty]).$$

Example 51. The nonempty powerset monad P^+ distributes over the finite indexed distribution monad I. The distributive law $\gamma : \mathsf{I} \circ \mathsf{P}^+ \to \mathsf{P}^+ \circ \mathsf{I}$ is given by

$$\gamma_X[(p_1, U_1), \cdots, (p_n, U_n)] \triangleq \{[(p_1, x_1), \cdots, (p_n, x_n)] \mid x_1 \in U_1, \cdots, x_n \in U_n\}.$$

It is easy to check that the expectation function $\mathrm{IE} : \mathsf{I}[0, \infty] \to [0, \infty]$ (Example 46) distributes over $\sup : \mathsf{P}^+[0, \infty] \to [0, \infty]$ (Example 44). Therefore, the wppt in the Cartesian Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, \mathsf{P}^+ \circ_\gamma \mathsf{I}, \mathsf{P}^+{}_{\sup} \circ \mathsf{I}_{\mathrm{IE}})$ is

$$\mathrm{wp}(f, i)(j) = \sup_{\nu \in f(j)} IE_{x\sim\nu}[i(x)] \quad (f : X \to \mathsf{P}^+ \circ_\gamma \mathsf{I}(Y), i : Y \to [0, \infty], j \in X).$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

This wppt performs *angelic choice*. We can also derive the one performing *demonic choice* by replacing sup with inf in this argument. Finally, we compose it with the counting monad C. This yields the Cartesian Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, (\mathsf{P}^+ \circ_\gamma \mathsf{I}) \circ_\theta \mathsf{C}, \mathsf{P}^+{}_{\sup} \circ \mathsf{I}_{\mathrm{IE}} \circ \mathsf{C}_{o(s)})$, whose wppt is:

$$\mathrm{wp}(f, i)(j) = \sup_{\nu \in f(j)} IE_{(c,x)\sim\nu}[c + i(x)] \quad (f : X \to \mathsf{P}^+ \circ_\gamma \mathsf{I}(Y), i : Y \to [0, \infty], j \in X).$$

Recently some new approaches to monadic models of nondeterministic and probabilistic branching have been proposed. Moggi et al. construct a monad structure on the composite functor $\mathsf{P}^+ \circ \mathsf{T}$ for any monad T over **Set** (Moggi et al. 2020). This is not given by distributive law. Goy and Petrisan employ a weaker notion of distributive law to combine the powerset monad and the probability distribution monad (Goy and Petrisan 2020). We leave it to a future work to study wppts for the monads arising from these approaches.

7. Expected Runtime as Weakest Precondition

So far we have been studying abstract wppts formulated in fibered category theory. One might wonder how they are useful for giving a concrete wppt semantics of an actual imperative programming language. An intended use of the fibrational wppt is the following.

Suppose that an imperative programming language L and its monadic denotational semantics $[\![\_]\!]$ are given. The semantics $[\![\_]\!]$ interprets a program C as a memory transformer $[\![C]\!] : M \to \mathsf{T}M$. To derive a wppt semantics of L, we prepare a Cartesian Dijkstra structure $(d^{\mathbb{C},\Omega} : \mathbb{C}/_{\ell}\Omega \to \mathbb{C}, \mathsf{T}, \dot{\mathsf{T}})$, then define the wppt semantics $W(C)$ of a program C as

$$W(C) \triangleq \mathrm{wp}([\![C]\!], \_) \in \mathbf{Pos}((\mathbb{C}(M, \Omega), \le), (\mathbb{C}(M, \Omega), \le)).$$

When the semantics $[\![\_]\!]$ consists of the standard monadic constructs, such as the Kleisli composition and algebraic operations, we may derive an inductive definition of W by using the composability of wp and the commutativity of wp with algebraic operations (Proposition 31).

We illustrate this story by deriving Kaminski et al's *expected runtime transformer* (ert for short) (Kaminski et al. 2016). Kaminski et al. showed that the expected runtime of probabilistic programs can be computed using a predicate transformer-like operator. We apply our theory of Dijkstra structures to derive their ert operator from a fibrational wppt and a monadic denotational semantics with cost counting program transformation (Theorem 52).

The language we consider here is a loop-free fragment of Kaminski et al.'s language (we will discuss languages with loops in Section 9) . Fix sets Var, Exp, Bool of variables, probabilistic expressions and probabilistic Boolean expressions, respectively. The syntax of the language is defined by the following BNF grammar (below $x \in \mathrm{Var}$, $e \in \mathrm{Exp}$, $b \in \mathrm{Bool}$):

$$\mathrm{Prog} \ni C := \mathrm{empty} \mid C; C \mid x :\sim e \mid \mathrm{tick} \mid \{C\}\Box\{C\} \mid \mathrm{if}\,(b)\{C\}\{C\}.$$

The command $x :\sim e$ samples a value from a probabilistic expression e. The box operator $\{C\}\Box\{C'\}$ nondeterministically chooses C or C' and executes the chosen program. The conditional expression if $(b)\{C\}\{C'\}$, samples a bit from a Bernoulli distribution with bias given by an expression b, then runs C if the sample is 1; otherwise runs C'. This conditional expression thus combines both probabilistic choice and deterministic conditionals.

Before introducing the expected runtime transformer, we prepare some interpretations of primitives. We fix a set V of values and let $M = V^{\mathrm{Var}}$ be the set of memory configurations. We give interpretations of $e \in \mathrm{Exp}$ and $b \in \mathrm{Bool}$ by functions $[\![e]\!] : M \to \mathsf{D}V$ and $[\![b]\!] : M \to [0, 1]$, respectively. We also assume that the memory configuration update function $\mathrm{upd}_x : M \times V \to M$ is suitably defined for each variable $x \in \mathrm{Var}$. The expected runtime transformer ert : $\mathrm{Prog} \times \mathbf{Set}(M, [0, \infty]) \to \mathbf{Set}(M, [0, \infty])$ is inductively defined as follows (below $i : M \to [0, \infty]$ and $\rho \in M$):

$$
\begin{aligned}
\mathrm{ert}(\mathrm{empty}, i)\rho &= i(\rho) \\
\mathrm{ert}(\mathrm{tick}, i)\rho &= 1 + i(\rho) \\
\mathrm{ert}(C_1; C_2, i)\rho &= \mathrm{ert}(C_1, \mathrm{ert}(C_2, i))\rho \\
\mathrm{ert}(\{C_1\}\square\{C_2\}, i)\rho &= \max(\mathrm{ert}(C_1, i)\rho, \mathrm{ert}(C_2, i)\rho) \\
\mathrm{ert}(x :\sim e, i)\rho &= 1 + E_{v \sim [\![e]\!]\rho}[i(\mathrm{upd}_x(\rho, v))] \\
\mathrm{ert}(\mathrm{if}\,(b)\{C_1\}\{C_2\}, i)\rho &= 1 + \mathrm{ert}(C_1, i)\rho \oplus_{[\![b]\!]\rho} \mathrm{ert}(C_2, i)\rho
\end{aligned}
$$

To derive the ert in our framework, we first give a monadic semantics of the language using the composite monad $\mathsf{T} \triangleq (\mathsf{P}^+ \circ_\gamma \mathsf{I}) \circ_\theta \mathsf{C}$ in Example 51. We interpret effectful commands and conditional commands using the following generic effects (Section 4.4) t, u, c for *counter tick*, *nondeterministic choice* and *probabilistic choice*, respectively.

$$
\begin{aligned}
t&\colon 1 \to \mathsf{T}1 & t(*) &= \{[(1, (1, *))]\} \\
u&\colon 1 \to \mathsf{T}2 & u(*) &= \{[(1, (0, \top))], [(1, (0, \bot))]\} \\
c&\colon [0, 1] \to \mathsf{T}2 & c(p) &= \{[(p, (0, \top)), (1 - p, (0, \bot))]\}
\end{aligned}
$$

They induce the following algebraic operations:

$$
\begin{aligned}
\alpha(t)_X(!_Y, f) &= \lambda y \in Y\,.\,\{\mathsf{I}(\lambda x\,.\,x + 1)(v) | v \in f(y, *)\} \\
\alpha(u)_X(!_Y, \beta(f_1, f_2)) &= \lambda y \in Y\,.\,f_1(y) \cup f_2(y) \\
\alpha(c)_X(p, \beta(f_1, f_2)) &= \lambda y \in Y\,.\,\{v \oplus_{p(y)} \mu | v \in f_1(y), \mu \in f_2(y)\}
\end{aligned}
$$

Using these algebraic operations, we define our denotational semantics:

$$
\begin{aligned}
[\![\mathrm{empty}]\!] &= \eta^{\mathsf{T}}_M \\
[\![\mathrm{tick}]\!] &= \alpha(t)_M(!_M, \eta^{\mathsf{T}}_M \circ r_M) \quad (r_M : M \times 1 \to M) \\
[\![C_1; C_2]\!] &= [\![C_2]\!] \bullet [\![C_1]\!] \\
[\![\{C_1\}\square\{C_2\}]\!] &= \alpha(u)_M(!_M, \beta([\![C_1]\!], [\![C_2]\!])) \\
[\![x :\sim e]\!] &= \mathsf{T}(\mathrm{upd}_x) \circ \theta_{M,V} \circ \langle \mathrm{id}_M, \iota_M \circ [\![e]\!]\rangle \\
[\![\mathrm{if}\,(b)\{C_1\}\{C_2\}]\!] &= \alpha(c)_M([\![b]\!], \beta([\![C_1]\!], [\![C_2]\!]))
\end{aligned}
$$

Here $\iota_M : \mathsf{D}M \to \mathsf{I}M$ is the inclusion (Example 46).

Recall that in Example 51, we obtained a Cartesian lifting $\dot{\mathsf{T}} \triangleq \mathsf{P}^+{}_{\mathrm{sup}} \circ \mathsf{I}_{\mathrm{IE}} \circ \mathsf{C}_{o(s)}$ of T along $d^{\mathbf{Set},[0,\infty]}$. In the Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, \mathsf{T}, \dot{\mathsf{T}})$, wp $([\![C]\!], i)$ satisfies

$$
\begin{aligned}
\mathrm{wp}([\![\mathrm{empty}]\!], i)\rho &= i(\rho) \\
\mathrm{wp}([\![\mathrm{tick}]\!], i)\rho &= 1 + i(\rho) \\
\mathrm{wp}([\![C_1; C_2]\!], i)\rho &= \mathrm{wp}([\![C_1]\!], \mathrm{wp}([\![C_2]\!], i))\rho \\
\mathrm{wp}([\![\{C_1\}\square\{C_2\}]\!], i)\rho &= \max(\mathrm{wp}([\![C_1]\!], i)\rho, \mathrm{wp}\,([\![C_2]\!], i)\rho) \\
\mathrm{wp}([\![x :\sim e]\!], i)\rho &= E_{v \sim [\![e]\!]\rho}[i(\mathrm{upd}_x\,(\rho, v))] \\
\mathrm{wp}([\![\mathrm{if}\,(b)\{C_1\}\{C_2\}]\!], i)\rho &= \mathrm{wp}([\![C_1]\!], i)\rho \oplus_{[\![b]\!]\rho} \mathrm{wp}([\![C_2]\!], i)\rho
\end{aligned}
$$

This is already very close to Kaminski et al's expected runtime transformer. The major difference between wp and ert is that the latter adds 1 to reflect the time consumption by probabilistic assignments and probabilistic conditionals. This difference can be resolved by an *accounting transformation* A defined below. It inserts a tick command before time-consuming instructions:

$$
\begin{aligned}
&A(\mathrm{empty}) = \mathrm{empty}, && A(x :\sim e) = \mathrm{tick}\,; x :\sim e && A(\{C\}\square\{C'\}) = \{A(C)\}\square\{A(C')\} \\
&A(C; C) = A(C); A(C') && A(\,\mathrm{tick}\,) = \mathrm{tick} && A(\,\mathrm{if}\,(b)\{C\}\{C'\}) = \mathrm{tick}; \mathrm{if}\,(b)\{C\}\{C'\}
\end{aligned}
$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Theorem 52. *Under the Cartesian Dijkstra structure* $(d^{\mathbf{Set},[0,\infty]}, \mathsf{T}, \dot{\mathsf{T}})$, *we have*

$$\mathrm{ert}(C, i) = \mathrm{wp}([\![A(C)]\!], i).$$

8. Dijkstra Structures on $d^{\mathbf{Set},[0,\infty]^n}$

As our final example, we consider an example of predicates that take values over $[0, \infty]^n$, based on Kura et al. (2019). They study the problem of computing bounds on tail probabilities to answer questions of the form *"what is the probability that a program takes more than n steps before terminating?"*. Their technique is based on using concentration bounds, for which one needs to compute higher moments of the form $E[T^n]$, where T is a random variable expressing the runtime of the execution of a program. To compute these, they carry a vector $(E[T], E[T^2], \ldots, E[T^n])$ containing the current estimations for the higher order moments. When the program takes a step, this vector can be used to compute the updated $(E[T+1], E[(T+1)^2], \ldots, E[(T+1)^n])$. For instance, for the second moment, we have $E[(T+1)^2] = E[T^2] + 2E[T] + 1$.

We derive this computation as a wppt in a suitable Cartesian Dijkstra structure on a domain fibration (unlike Kura et al. 2019 here we do not consider the nondeterministic computations). Consider again the composite $\mathsf{D} \circ_\theta \mathsf{C}$ of the counter monad and the finite probability distribution monad considered in Example 50. We first introduce EM monotone algebras for C and D as follows:

- We define an algebraic operation $\mathrm{tick}_n \colon [0, \infty]^n \to [0, \infty]^n$ that simulates letting one unit of time pass:

$$\mathrm{tick}_n(r_1, \ldots, r_n) = (\mathrm{tick}_n^1(r_1, \ldots, r_n), \ldots, \mathrm{tick}_n^n(r_1, \ldots, r_n))$$
$$\text{where } \mathrm{tick}_n^k(r_1, \ldots, r_n) = 1 + \sum_{i=1}^{k} \binom{k}{i} r_i \qquad (1 \le k \le n)$$

  Here the inputs are taken to be the moments of T, where r_i represents $E[T^i]$, and tick_n^i computes $E[(T+1)^i]$ from them. This coincides with the *elapse function* defined in Kura et al. (2019). This operation induces an EM monotone C-algebra $o(\mathrm{tick}_n) \colon \mathsf{C}[0, \infty]^n \to [0, \infty]^n$ for each n.
- For D we take the product of n-fold copies of the expected value $\mathrm{E} \colon \mathsf{D}[0, \infty] \to [0, \infty]$ which is an EM monotone D-algebra as shown in Example 45. This product is itself an EM monotone D-algebra $\mathrm{E}^n \colon \mathsf{D}[0, \infty]^n \to [0, \infty]^n$.

It follows from affinity of tick_n and linearity of the expected value that tick_n is a D-algebra endomorphism over E^n. From Theorem 49, we obtain the Cartesian lifting $\mathsf{D}_{\mathrm{E}^n} \circ \mathsf{C}_{o(\mathrm{tick}_n)}$ of $\mathsf{D} \circ_\theta \mathsf{C}$ along $d^{\mathbf{Set},[0,\infty]^n} : \mathbf{Set}/_\ell[0, \infty]^n \to \mathbf{Set}$. In the Dijkstra structure $(d^{\mathbf{Set},[0,\infty]^n}, \mathsf{D} \circ_\theta \mathsf{C}, \mathsf{D}_{\mathrm{E}^n} \circ \mathsf{C}_{o(\mathrm{tick}_n)})$, the wppt satisfies the definition below:

$$\mathrm{wp}(f, p)(j) \triangleq (\mathrm{wp}_1(f, p)(j), \ldots, \mathrm{wp}_n(f, p)(j)), \qquad (f : X \to \mathsf{D} \circ_\theta \mathsf{C}(Y), p : Y \to [0, \infty]^n)$$
$$\text{where} \quad \mathrm{wp}_k(f, p)(j) \triangleq \mathrm{E}_{(c,i) \sim f(j)} \left[c^k + \sum_{l=1}^{k} \binom{k}{l} c^{k-l} p_l(i) \right]$$

Here f represents an effectful computation, and p is a function that maps final states of f to the vector of moments for the runtime of some continuation of f. The wp operator then computes the vector of moments for the runtime of f followed by its continuation. In particular, this coincides with the F_n operator from Kura et al. (2019).

9. Continuity and Fixed Points

For a $\mathbb{K}$-Dijkstra structure $(p, \mathsf{T}, \dot{\mathsf{T}})$, its associated wppt $\mathrm{wp}(f, \_)$ is a monotone function belonging to $\mathbb{K}$ (Definition 5-(2)); this means that it obeys the constraint imposed by the subcategory $\mathbb{K}$ of **Pos**. For example, the wppt of an ω**CPO**-Dijkstra structure is continuous on its second argument. For practical applications, it is also interesting to consider continuity of wppts on their first argument. This is useful for defining the wppt of an unbounded loop program in closed form via a fixpoint equation, such as it is done in other calculi (Kaminski et al. 2016; McIver and Morgan 2005). In this section we introduce the concept of enrichment on Kleisli categories and discuss the continuity of wppts on their first argument. We then show that under a certain condition, the wppt of a while loop program can be computed by the least fixpoint of the wppt of the loop body.

Throughout this section, we fix a category $\mathbb{C}$ and a monad (T, η, μ) on $\mathbb{C}$. Recall that an ω-complete partial order (ωCPO) is a CPO $(X, \sqsubseteq_X)$ in which every ω-chain $x_0 \sqsubseteq_X x_1 \sqsubseteq_X \ldots$ has a least upper bound $\sqcup_{i\in\omega} x_i$. We define ω**CPO** to be the category of ωCPOs (which may not have the least element) and continuous functions (i.e. functions that preserve least upper bounds) between them.

9.1 Continuity of Weakest Precondition Predicate Transformers on Morphisms

To discuss the continuity of wppts in their first argument, we first introduce the concept of enrichment of Kleisli categories by the Cartesian category $(\omega\mathbf{CPO}, 1, \times)$.

Proposition 53. *Let $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ be a family such that each $\sqsubseteq_{X,Y}$ is an ωCPO on the homset $\mathbb{C}(X, \mathsf{T}Y)$. Then the following are equivalent (below f_n, g_n are ω-chains of morphisms of appropriate type):*

(1) The Kleisli composition $(\bullet)$ is monotone and continuous in the following sense:

$$(\bullet) \in \omega\mathbf{CPO}((\mathbb{C}(Y, \mathsf{T}Z), \sqsubseteq_{Y,Z}) \times (\mathbb{C}(X, \mathsf{T}Y), \sqsubseteq_{X,Y}), (\mathbb{C}(X, \mathsf{T}Z), \sqsubseteq_{X,Z})).$$

(2) The Kleisli composition $(\bullet)$ is monotone and continuous in each argument.

(3) Precomposition $\_ \circ g$, the Kleisli lifting operation $(-)^{\#}$, and the postcomposition $f \bullet \_$ are all monotone and continuous.

The proof of this proposition is easy and omitted.

Definition 54. *An* ω**CPO**-enrichment *of the Kleisli category $\mathbb{C}_\mathsf{T}$ is a family $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ such that each $\sqsubseteq_{X,Y}$ is an ωCPO on the homset $\mathbb{C}(X, \mathsf{T}Y)$, and moreover one of the equivalent conditions in Proposition 53 hold.*

We now seek for the condition for wppts to be continuous in their first argument.

Proposition 55. *Let $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ be an ω**CPO**-enrichment of $\mathbb{C}_\mathsf{T}$, $(\Omega, \leq)$ be an ordered object in $\mathbb{C}$ belonging to ω**CPO**, and $o : \mathsf{T}\Omega \to \Omega$ be an EM monotone algebra. Consider the wppt* wp *of the Dijkstra structure $(d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}, \mathsf{T}, \mathsf{T}_o)$. Then the following are equivalent:*

(1) $o \circ \_ \in \omega\mathbf{CPO}((\mathbb{C}(X, \mathsf{T}\Omega), \sqsubseteq_{X,\Omega}), (\mathbb{C}(X, \Omega), \leq_X))$ for every $X \in \mathbb{C}$.
(2) $\mathrm{wp}(\_, i) \in \omega\mathbf{CPO}((\mathbb{C}(X, \mathsf{T}Y), \sqsubseteq_{X,Y}), (\mathbb{C}(X, \Omega), \leq_X))$ for every $X, Y \in \mathbb{C}$ and $i : Y \to \Omega$.

Proof. (1) $\Longrightarrow$ (2) Since $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ is an ω**CPO**-enrichment, the postcomposition $Ti \circ \_ = (\eta \circ i)^{\#} \circ -$ is monotone and continuous by Proposition 53. Thus, we obtain the following diagram in ω**CPO**, and the composition coincides with $\mathrm{wp}(\_, i) = o \circ Ti \circ \_$. This concludes the monotonicity and continuity of $\mathrm{wp}(\_, i)$.

$$wp(\_, i) = \ (\mathbb{C}(X, \mathsf{T}Y), \sqsubseteq_{X,Y}) \xrightarrow{Ti\circ -} (\mathbb{C}(X, \mathsf{T}\Omega), \sqsubseteq_{X,\Omega}) \xrightarrow{o\circ -} (\mathbb{C}(X, \Omega), \leq_X).$$

(2) $\Longrightarrow$ (1) Immediate from $o \circ \_ = \mathrm{wp}(\_, \mathrm{id}_\Omega)$. $\square$

Definition 56. *Let $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ be an ω**CPO**-enrichment of $\mathbb{C}_\mathsf{T}$, $(\Omega, \leq)$ be an ordered object in $\mathbb{C}$ belonging to ω**CPO**, and $o : \mathsf{T}\Omega \to \Omega$ be an EM monotone algebra. We say that o is* compatible *with the ω**CPO**-enrichment if one of the equivalent conditions in Proposition 55 holds.*

In any Kleisli category $\mathbb{C}_\mathsf{T}$ with an ω**CPO**-enrichment, each free algebra is monotone, belonging to ω**CPO**, and is compatible with the enrichment.

Proposition 57. *Let $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ be an ω**CPO**-enrichment on $\mathbb{C}_\mathsf{T}$ and $\Omega \in \mathbb{C}$ be an object. Then $(\mathsf{T}\Omega, \sqsubseteq_{\_,\Omega})$ is an ordered object in $\mathbb{C}$ belonging to ω**CPO**, and $\mu_\Omega : \mathsf{T}\mathsf{T}\Omega \to \mathsf{T}\Omega$ is an EM monotone T-algebra belonging to ω**CPO** that is compatible with the ω**CPO**-enrichment.*

Example 58. In the category **Set** with a monad T, it is sometimes enough to equip $\mathsf{T}Y$ with an ωCPO $\sqsubseteq_Y$ for every set Y, then extend it to an ωCPO on the homset $\mathbf{Set}(X, \mathsf{T}Y)$ by the pointwise order. This is the case of:

- The maybe monad M in Example 34, where the ωCPO $\sqsubseteq_Y$ on $\mathsf{M}Y$ is the flat order making $\{*\}$ the least element. We then extend this order to the pointwise order $\sqsubseteq_{X,Y}$ on $\mathbf{Set}(X, \mathsf{M}Y)$. This is an ω**CPO**-enrichment on $\mathbf{Set}_\mathsf{M}$. Among EM monotone M-algebras tot, par in Example 34, only tot belongs to ω**CPO** and is compatible with the ω**CPO**-enrichment.
- The nonempty powerset monad $\mathsf{P}^+X \triangleq \{U \subseteq X \mid U \neq \emptyset\}$ in Example 36, where the ωCPO $\sqsubseteq_Y$ on P^+Y is given by set inclusion. We then extend this order to the pointwise order $\sqsubseteq_{X,Y}$ on $\mathbf{Set}(X, \mathsf{P}^+Y)$. This is an ω**CPO**-enrichment on $\mathbf{Set}_{\mathsf{P}^+}$. Among EM monotone P^+-algebras may, must in Example 36, only may belongs to ω**CPO** and is compatible with the ω**CPO**-enrichment.
- The subdistribution monad SD, which assigns to $X \in \mathbf{Set}$ the set of probability subdistributions over X, i.e., functions $\nu : X \to [0, 1]$ such that ν is nonzero at countably many points and $\sum_{x\in X} \nu(x) \leq 1$. This is a variant of the finite probability distribution monad D in Example 40. Given $\mu_1, \mu_2 \in \mathsf{SD}(X)$, we define the order $\mu_1 \sqsubseteq \mu_2$ by $\mu_1(x) \leq \mu_2(x)$ for every $x \in X$. We extend this order to the pointwise order $\sqsubseteq_{X,Y}$ on $\mathbf{Set}(X, \mathsf{SD}Y)$. This is an ω**CPO**-enrichment on SD. The expectation function $\mathsf{E} : \mathsf{SD}[0, \infty] \to [0, \infty]$ given by $\mathsf{E}(\mu) = \sum_{x\in X} x\mu(x)$ is the EM monotone SD-algebra belonging to ω**CPO** that is compatible with the ω**CPO**-enrichment.

We end this section by updating Theorem 32 with an ω**CPO** enrichment. Let $(1, \times)$ be finite products on $\mathbb{C}$ and θ be a strength for T, $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ be an ω**CPO**-enrichment on $\mathbb{C}_\mathsf{T}$, $(\Omega, \leq)$ be an ordered object in $\mathbb{C}$ belonging to ω**CPO**, and $o : \mathsf{T}\Omega \to \Omega$ be an EM monotone T-algebra belonging to ω**CPO** and compatible with the ω**CPO**-enrichment. Then for any generic effect $g : A \to \mathsf{T}B$ and $X, Y \in \mathbb{C}$, we obtain the following commutative diagram in ω**CPO**:

$$\begin{tikzcd}
(\mathbb{C}(Y \times B, \mathsf{T}X), \sqsubseteq_{Y\times B,X}) \arrow[r, "\mathrm{wp}(-,i)"] \arrow[d, "\alpha(g)_X(a,_)"'] & (\mathbb{C}(Y \times B, \Omega), \leq_{Y\times B}) \arrow[d, "op(o,g)(a,_)"] \\
(\mathbb{C}(Y, \mathsf{T}X), \sqsubseteq_{Y,X}) \arrow[r, "\mathrm{wp}(-,i)"'] & (\mathbb{C}(Y, \Omega), \leq_Y).
\end{tikzcd}$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Theorem 32 guarantees the commutativity of the diagram. It remains to verify that morphisms in the above diagram are inhabitants of ω**CPO**:

(1) From Proposition 57, $(\mathsf{T}X, \sqsubseteq_{\_,X})$ is an ordered object in $\mathbb{C}$ belonging to ω**CPO**, and μ_X is an EM monotone T-algebra belonging to ω**CPO**. Therefore, the operation $\alpha(g)_X(a, \_)$ belongs to ω**CPO** by Proposition 31.
(2) Since $(\Omega, \leq)$ and $o : \mathsf{T}\Omega \to \Omega$ belongs to ω**CPO**, $\mathrm{op}(o, g)(a, \_)$ belongs to ω**CPO** by Proposition 31.
(3) Since o is compatible with the ω**CPO**-enrichment, $\mathrm{wp}(\_, i)$ belongs to ω**CPO** by Proposition 55.

9.2 Application: Semantics of While Loops

We use the theory developed in the previous section to study the weakest precondition of while loops in imperative languages. This is carried out in the Dijkstra structure $(d^{\mathbb{C},\Omega}, \mathsf{T}, \mathsf{T}_o)$ specified by the following data:

- $\mathbb{C}$ is a category with finite products $(1, \times)$ and finite coproducts $(0, +)$, with a distributive law (Section 4.4).
- $(\mathsf{T}, \eta, \mu, \theta)$ is a strong monad on $\mathbb{C}$, and $\{\sqsubseteq_{X,Y}\}_{X,Y\in\mathbb{C}}$ is an ω**CPO**-enrichment on $\mathbb{C}_\mathsf{T}$. We assume that each order $\sqsubseteq_{X,Y}$ admits the least element $\perp_{X,Y}$, and the cotupling $[-,-]$ of the coproduct is an isomorphism in ω**CPO**:
$$[-,-] : (\mathbb{C}(X, \mathsf{T}Z), \sqsubseteq_{X,Z}) \times (\mathbb{C}(Y, \mathsf{T}Z), \sqsubseteq_{Y,Z}) \cong (\mathbb{C}(X+Y, \mathsf{T}Z), \sqsubseteq_{X+Y,Z}). \tag{23}$$
- $(\Omega, \leq)$ is an ordered object in $\mathbb{C}$ belonging to ω**CPO**. We assume that each order $\leq_X$ admits the least element $\perp_X$, and the cotupling of the coproduct is an isomorphism in ω**CPO**:
$$[-,-] : (\mathbb{C}(X, \Omega), \leq_X) \times (\mathbb{C}(Y, \Omega), \leq_Y) \cong (\mathbb{C}(X+Y, \Omega), \leq_{X+Y}). \tag{24}$$
- $o : \mathsf{T}\Omega \to \Omega$ is an EM monotone T-algebra belonging to ω**CPO** that is compatible with the ω**CPO**-enrichment. Moreover, $o \circ \mathsf{T}i \circ \perp_{X,Y} = \perp_X$ holds for any object $X, Y \in \mathbb{C}$ and morphism $i : Y \to \Omega$. This guarantees that the wppt wp of the Dijkstra structure $(d^{\mathbf{Set},\Omega}, \mathsf{T}, \mathsf{T}_o)$ is not only continuous but also *strict* in its first argument.

We assume that a program C is interpreted as a memory transformer $[\![C]\!] : M \to \mathsf{T}M$, where M is the object representing the collection of memory configurations. We also assume that a Boolean expression b is interpreted as a function $[\![b]\!] : M \to 2$, where 2 is the coproduct $1+1$. We represent true and false by the left and right injections, respectively. To interpret a while loop of an imperative language as a morphism $[\![\mathrm{while}\,(b)\{C\}]\!] : M \to \mathsf{T}M$, we first define the following family of partial iterations:

$$\begin{aligned} W_{b,C,0} &\triangleq \perp_{M,M} \\ W_{b,C,n+1} &\triangleq \alpha(\eta_2)_M([\![b]\!], \beta(W_{b,C,n} \bullet [\![C]\!], \eta_M)) \\ &= [(W_{b,C,n} \bullet [\![C]\!]) \circ r_M, \eta_M \circ r_M] \circ \mathrm{dist}_{M,1,1} \circ \langle \mathrm{id}_M, [\![b]\!] \rangle \end{aligned}$$

where $W_{b,C,n}$ executes n iterations of the loop and returns the bottom element in the traces that have not terminated. For the definition of β, see (18). The right-hand side of $W_{b,C,n+1}$ is the denotation of the conditional command $\mathrm{if}\,(b)\{C; W_{b,C,n}\}$. We can see that $W_{b,C,n} = \Phi^n_{b,C}(\perp_{M,M})$ where $\Phi_{b,C} : \mathbb{C}(M, \mathsf{T}M) \to \mathbb{C}(M, \mathsf{T}M)$ is an operator defined as

$$\Phi_{b,C}(f) \triangleq \alpha(\eta_2)_M([\![b]\!], \beta(f \bullet [\![C]\!], \eta_M)).$$

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

It is a continuous function with respect to the ωCPO given by the enrichment:

Lemma 59. $\Phi_{b,C} \in \omega\mathbf{CPO}((\mathbb{C}(M, \mathsf{T}M), \sqsubseteq_{M,M}), (\mathbb{C}(M, \mathsf{T}M), \sqsubseteq_{M,M}))$.

Proof. This is immediate because $\alpha(\eta_2)_M([\![b]\!], \_)$ is continuous by Proposition 31, and the mapping $f \mapsto \beta(f \bullet [\![C]\!], \eta_M) = [(f \bullet [\![C]\!]) \circ r, \eta_M \circ r] \circ \mathrm{dist}$ is continuous from the property of $\omega\mathbf{CPO}$-enrichment and the assumption (23) on cotupling. □

We then take the least fixpoint of $\Phi_{b,C}$ and employ it as the denotation of the while program while $(b)\{C\}$:

$$[\![\mathrm{while}\,(b)\{C\}]\!] \triangleq \bigsqcup_n \Phi^n_{b,C}(\bot_{M,M}) = \mathsf{lfp}\ \Phi_{b,C}.$$

We want to compute the weakest precondition $\mathrm{wp}([\![\ \mathrm{while}\,(b)\{C\}]\!], j)$ of the while loop at a morphism $j : M \to \Omega$. By Corollary 33, for all n:

$$\mathrm{wp}(W_{b,C,n+1}, j) = \mathrm{op}(o, \eta_2)([\![b]\!], \beta(\mathrm{wp}([\![C]\!], \mathrm{wp}\ (W_{b,C,n}, j)), j)).$$

We thus let $\Psi_{b,C,j} : \mathbb{C}(M, \Omega) \to \mathbb{C}(M, \Omega)$ be another operator

$$\Psi_{b,C,j}(i) \triangleq \mathrm{op}(o, \eta_2)([\![b]\!], \beta(\mathrm{wp}([\![C]\!], i), j)).$$

This operation is also continuous:

Lemma 60. *For any morphism $j : M \to \Omega$ in $\mathbb{C}$, we have*

$$\Psi_{b,C,j} \in \omega\mathbf{CPO}((\mathbb{C}(M, \Omega), \leq_M), (\mathbb{C}(M, \Omega), \leq_M)).$$

Proof. Let $j : M \to \Omega$ be a morphism in $\mathbb{C}$. Notice that $\Psi_{b,C,j}$ can be factored as

$$\Psi_{b,C,j} = \mathrm{op}(o, \eta_2)([\![b]\!], \_) \circ \beta(\_, j) \circ \mathrm{wp}([\![C]\!], \_).$$

Then $\mathrm{op}(o, \eta_2)([\![b]\!], \_)$ and $\mathrm{wp}([\![C]\!], \_)$ are continuous because o belongs to $\omega\mathbf{CPO}$ by Proposition 31. Also, $\beta(\_, j) = [\_ \circ r, j \circ r] \circ \mathrm{dist}$ is continuous because o is an ordered object and cotupling is continuous by (24). □

Proposition 61. *For any morphism $j : M \to \Omega$ in $\mathbb{C}$, we have the following commutative diagram in $\omega\mathbf{CPO}$:*

$$\begin{tikzcd}
(\mathbb{C}(M, \mathsf{T}M), \sqsubseteq_{M,M}) \arrow[r, "\mathrm{wp}(-,j)"] \arrow[d, "\Phi_{b,C}"'] & (\mathbb{C}(M, \Omega), \leq) \arrow[d, "\Psi_{b,C,j}"] \\
(\mathbb{C}(M, \mathsf{T}M), \sqsubseteq_{M,M}) \arrow[r, "\mathrm{wp}(-,j)"'] & (\mathbb{C}(M, \Omega), \leq)
\end{tikzcd}$$

Proof. Since the EM monotone T-algebra o is compatible with the $\omega\mathbf{CPO}$-enrichment, $\mathrm{wp}(\_, j)$ belongs to $\omega\mathbf{CPO}$ by Proposition 55. We next show the commutativity. Let $f : M \to \mathsf{T}M$ and $j : M \to \Omega$ be morphisms in $\mathbb{C}$. We have

$$\begin{aligned}
\mathrm{wp}(\Phi_{b,C}(f), j) &= \mathrm{wp}(\alpha(\eta_2)_M([\![b]\!], \beta(f \bullet [\![C]\!], \eta_M)), j) && \text{(definition)}\\
&= \mathrm{op}(o, \eta_2)([\![b]\!], \beta(\mathrm{wp}([\![C]\!], \mathrm{wp}(f, j)), j)) && \text{(Corollary 33)}\\
&= \Psi_{b,C,j}(\mathrm{wp}(f, j)). && \text{(definition)}
\end{aligned}$$

□

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Theorem 62. *For any morphism $j : M \to \Omega$ in $\mathbb{C}$, $\Phi_{b,C}$ and $\Psi_{b,C,j}$ defined as above satisfy*

$$\mathrm{wp}(\mathsf{lfp}\ \Phi_{b,C}, j) = \mathrm{wp}([\![\mathrm{while}\ (b)\{C\}]\!], j) = \mathsf{lfp}(\Psi_{b,C,j}).$$

Proof. From the strictness assumption and continuity of $\mathrm{wp}(\_, j)$, plus Proposition 61 and Scott induction, we conclude the equation. □

10. Cartesian Dijkstra Structures Arising from Change-of-Base

In Section 3.3, we have seen the change-of-base construction of Dijkstra structures. In this section, we present some examples of this construction to derive Cartesian Dijkstra structures over relational fibrations.

We first present an example of the change-of-base to obtain fibrations whose total category contains binary relations as objects (Bonchi et al. 2018; Jacobs 1999).

Example 63. Let $(\mathbb{C}, \mathbf{I}, \otimes)$ be a monoidal category and $(\Omega, \leq)$ be an ordered object in $\mathbb{C}$. We take the change-of-base of $d^{\mathbb{C},\Omega} : \mathbb{C}/_\ell\Omega \to \mathbb{C}$ along the tensor product functor $(\otimes) : \mathbb{C}^2 \to \mathbb{C}$. We write the resulting fibration $r^{\mathbb{C},\Omega} : \mathbf{BRel}(\mathbb{C}, \Omega) \to \mathbb{C}^2$.

$$\begin{tikzcd}
\mathbf{BRel}(\mathbb{C}, \Omega) \arrow[r] \arrow[d, "r^{\mathbb{C},\Omega}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{C}/_\ell\Omega \arrow[d, "d^{\mathbb{C},\Omega}"] \\
\mathbb{C}^2 \arrow[r, "(\otimes)"'] & \mathbb{C}
\end{tikzcd}$$

The concrete description of $\mathbf{BRel}(\mathbb{C}, \Omega)$ is the following: objects are triples $(X, Y, i : X \otimes Y \to \Omega)$ of two objects and a morphism in $\mathbb{C}$, and a morphism from (X, Y, i) to (X', Y', i') is a pair of $\mathbb{C}$-morphisms $f : X \to X'$ and $g : Y \to Y'$ such that $i \leq_{X\otimes Y} i' \circ (f \otimes g)$. Intuitively, $\mathbf{BRel}(\mathbb{C}, \Omega)$ is the category of Ω-valued binary relations between $\mathbb{C}$-objects. In the fibration $r^{\mathbb{C},\Omega} : \mathbf{BRel}(\mathbb{C}, \Omega) \to \mathbb{C}^2$, the pullback is given by $(f, g)^*(X', Y', i') = i' \circ (f \otimes g)$. Example 8 is an instance of this construction where $\mathbb{C} = \mathbf{Set}$ and $\Omega = \mathbf{2}$ and the tensor product functor $(\otimes)$ is the binary product functor $(\times)$.

Example 64. Let $(\mathbb{C}, \mathbf{I}, \otimes)$ be a monoidal category and (T, η, μ) be a monad on $\mathbb{C}$. Then $(\otimes, m)$ is a monad opfunctor from T^2 (the product monad on $\mathbb{C}^2$) to T if and only if $m_{X,Y} : \mathsf{T}X \otimes \mathsf{T}Y \to \mathsf{T}(X \otimes Y)$ is a natural transformation satisfying

$$\begin{tikzcd}
X \otimes Y \arrow[d, "\eta_X \otimes \eta_Y"'] \arrow[dr, "\eta_{X\otimes Y}"] & \\
\mathsf{T}X \otimes \mathsf{T}Y \arrow[r, "m_{X,Y}"'] & \mathsf{T}(X \otimes Y)
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{T}^2X \otimes \mathsf{T}^2Y \arrow[r, "m_{\mathsf{T}X,\mathsf{T}Y}"] \arrow[d, "\mu_X \otimes \mu_Y"'] & \mathsf{T}(\mathsf{T}X \otimes \mathsf{T}Y) \arrow[r, "\mathsf{T}m_{X,Y}"] & \mathsf{T}^2(X \otimes Y) \arrow[d, "\mu_{X\otimes Y}"] \\
\mathsf{T}X \otimes \mathsf{T}Y \arrow[rr, "m_{X,Y}"'] & & \mathsf{T}(X \otimes Y).
\end{tikzcd}
\tag{25}$$

This is exactly the tensor product part of the data to make T a *monoidal functor*.

Specially, for any commutative monad $(\mathsf{T}, \eta, \mu, \theta_{X,Y})$ on a symmetric monoidal category $(\mathbb{C}, \mathbf{I}, \otimes)$, put $\theta'_{X,Y}$ the symmetric version of the strength θ. Then the following *double strength* (Jacobs 1994, Definition 3.4) satisfies (25).

$$\mathrm{dst}^{\mathsf{T}}_{X,Y} :\ \mathsf{T}X \otimes \mathsf{T}Y \xrightarrow{\theta_{\mathsf{T}X,Y}} \mathsf{T}(\mathsf{T}X \otimes Y) \xrightarrow{\mathsf{T}(\theta'_{X,Y})} \mathsf{T}^2(X \otimes Y) \xrightarrow{\mu_{X\otimes Y}} \mathsf{T}(X \otimes Y)$$

This is shown in Theorem 2.3 in Kock (1970). Therefore $((\otimes), \mathrm{dst}^{\mathsf{T}})$ is a monad opfunctor from T^2 to T.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Example 65. The nonempty powerset monad P^+ is commutative over the Cartesian category $(\mathbf{Set}, 1, \times)$. Its double strength $\mathrm{dst}^{\mathsf{P}^+}_{X,Y} : \mathsf{P}^+X \times \mathsf{P}^+Y \to \mathsf{P}^+(X \times Y)$ satisfies

$$\mathrm{dst}^{\mathsf{P}^+}_{X,Y}(U, V) = \{(u, v) \mid u \in U, v \in V\}.$$

From Example 64, the pair $((\times), \mathrm{dst}^{\mathsf{P}^+})$ becomes a monad opfunctor from $(\mathsf{P}^+)^2$ to P^+.

We use this monad opfunctor to take the change-of-base of two Cartesian Dijkstra structures $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{P}^+, \mathsf{P}^+{}_{\mathrm{may}})$ and $(d^{\mathbf{Set},\mathbf{2}}, \mathsf{P}^+, \mathsf{P}^+{}_{\mathrm{must}})$ in Example 36. The resulting Cartesian Dijkstra structures are denoted by $(r^{\mathbf{Set},\mathbf{2}}, (\mathsf{P}^+)^2, \mathsf{T}_{\mathrm{may}})$ and $(r^{\mathbf{Set},\mathbf{2}}, (\mathsf{P}^+)^2, \mathsf{T}_{\mathrm{must}})$, respectively. The behavior of the liftings $\mathsf{T}_{\mathrm{may}}, \mathsf{T}_{\mathrm{must}}$ is described as follows. We first represent each object in $\mathbf{BRel}(\mathbf{Set}, \mathbf{2})$ by a triple of sets (X, Y, R) such that $R \subseteq X \times Y$. Then liftings $\mathsf{T}_{\mathrm{may}}, \mathsf{T}_{\mathrm{must}}$ act on $\mathbf{BRel}(\mathbf{Set}, \mathbf{2})$-objects as follows:

$$\mathsf{T}_{\mathrm{may}}(X, Y, R) = (\mathsf{P}^+X, \mathsf{P}^+Y, \{(U, V) \mid \exists u \in U . \exists v \in V . (u, v) \in R\})$$
$$\mathsf{T}_{\mathrm{must}}(X, Y, R) = (\mathsf{P}^+X, \mathsf{P}^+Y, \{(U, V) \mid \forall u \in U . \forall v \in V . (u, v) \in R\}).$$

Example 66. The finite probability distribution monad D is also commutative over the Cartesian category $(\mathbf{Set}, 1, \times)$; its double strength $\mathrm{dst}^{\mathsf{D}}_{X,Y} : \mathsf{D}X \times \mathsf{D}Y \to \mathsf{D}(X \times Y)$ takes the *product* of two probability distributions:

$$\mathrm{dst}^{\mathsf{D}}_{X,Y}(\mu, \nu) = \lambda(x, y) . \mu(x) \times \nu(y) \triangleq \mu \otimes \nu.$$

From Example 64, the pair $((\times), \mathrm{dst}^{\mathsf{D}})$ becomes a monad opfunctor from D^2 to D.

We use this monad opfunctor to take the change-of-base of the Cartesian Dijkstra structure $(d^{\mathbf{Set},[0,\infty]}, \mathsf{D}, \mathsf{D}_{\mathrm{E}})$ in Example 45. The resulting Cartesian Dijkstra structure $(r^{\mathbf{Set},[0,\infty]}, \mathsf{D}^2, \mathsf{T}_{\mathrm{E}})$ has the lifting whose object part behaves as

$$\mathsf{T}_{\mathrm{E}}(X, Y, i) = (DX, DY, \lambda(\mu, \nu) . E_{(x,y)\sim\mu\otimes\nu}[i(x, y)]).$$

11. Related Work

One of the earliest studies on a logic for a monadic programming language is Pitts' *evaluation logic* (Pitts 1991). The logic has an evaluation modality $[x \Leftarrow E]\phi(x)$, internalizing the wppt in the predicate logic. The semantics of the evaluation modality is given by a *T-modality* $\Box_{X,Y}$ (Pitts 1991, Definition 3.3.1), which is an extension of monad lifting with an extra parameter, but axiomatized differently from (Cartesian) monad liftings. Later, Moggi gave a semantics of evaluation logic in the framework of *dominance* (Moggi 1995). Some examples of T-modalities are also given in Pitts (1991), Moggi (1995), overlapping with examples in Section 5. A precise relationship between T-modalities and Cartesian liftings is yet to be studied. We currently know that in a Dijkstra structure $(p, \mathsf{T}, \dot{\mathsf{T}})$ with a strength θ for T, the operator $\Box_{X,Y}(P) \triangleq \theta^*_{X,Y}\dot{\mathsf{T}}(P)$ satisfies the T-modality axioms. Conversely, any T-modality induces a Cartesian lifting of the underlying monad in a suitable fibration.

The interpretation of Hoare logic has also been pursued using other structures than fibrations. Abramsky et al. introduce *specification structures* as a loose framework for general Hoare logic (Abramsky et al. 1996). It is easy to see that a Dijkstra structure in our setting determines a specification structure. Specification structures themselves do not provide wppts, and computational effects are not explicitly modeled.

Martin et al. (2006) give a categorical framework for Hoare logic using certain functors of type $H : S \to \mathbf{PreOrd}$. They correspond to *opfibrations*, hence they offer *strongest postconditions predicate transformers* instead. Computational effects are not explicitly modeled. One unique feature of Martin et al. (2006) is that their Hoare logic supports *trace operators* in the sense of Joyal et al. (1996).

In Goncharov and Schröder (2013), Goncharov et al. constructed semantics of Hoare logic from a pair $(\mathscr{P}, \mathscr{T})$ of an order-enriched monad $\mathscr{T}$, and an *innocent submonad* $\mathscr{P}$ of $\mathscr{T}$. They showed that $\mathscr{P}1$ is a frame, and used this fact to interpret Hoare logic predicates by morphisms of type $X \to \mathscr{P}1$. They introduced the wppt using the join of $\mathscr{P}1$ in Goncharov and Schröder (2013), and called its composability *sequential compatibility*. Later they showed the wppt induced from the pair $(\mathscr{P}, \mathscr{T})$ is sequentially compatible if and only if the canonical morphism $\mathscr{T}\mathscr{P}1 \to \mathscr{P}1$ is an EM algebra (Rauch et al. 2016, Remark 11). This fact bridges the work (Goncharov and Schröder 2013; Rauch et al. 2016) and the following approach studied by Hasuo.

In Hasuo (2015), Hasuo introduced *PT situations* to construct composable wppts. A PT situation is a tuple of an order-enriched monad $(\mathsf{T}, \preceq)$ (in a different sense from Goncharov and Schröder 2013), an object $\Omega \in \mathbb{C}$ and an EM T-algebra τ on $\mathsf{T}\Omega$, satisfying certain conditions. Each PT situation induces a functor $\mathbb{P}^{Kl}(\tau) : (\mathbb{C}_\mathsf{T})^{op} \to \mathbf{Pos}$ embodying a wppt. The construction of $\mathbb{P}^{Kl}$ can be accounted for by the recipe given in Section 4. Let $(\mathsf{T}, \preceq, \Omega, \tau)$ be a PT situation. It determines an ordered object $(\mathsf{T}\Omega, \leqslant)$ in $\mathbb{C}$, and τ is monotone (Hasuo 2015, Definition 2.4). Using the recipe in Section 4, it determines a Cartesian lifting T_τ of T along $d^{\mathbb{C},\mathsf{T}\Omega} : \mathbb{C}/\mathsf{T}\Omega \to \mathbb{C}$. Then the wppt wp in the Cartesian Dijkstra structure $(d^{\mathbb{C},\mathsf{T}\Omega}, \mathsf{T}, \mathsf{T}_\tau)$ coincides with $\mathbb{P}^{Kl}(\tau)$. This demonstrates that the recipe in Section 4 generalizes the construction of $\mathbb{P}^{Kl}(\tau)$ given in Hasuo (2015). Especially, our recipe does not demand order enrichment on T and allows us to take arbitrary ordered object (not limited to the form $\mathsf{T}\Omega$) as the basis of the lax slice construction. Examples 36, 38, 40 are benefited from this generalization.

Hino et al. (2016) introduced another construction of wppts using *relative algebras*. A relative algebra is a monad morphism of type $\mathsf{T} \to \mathbb{D}(\Omega^-, \Omega)$, where T is a **Set**-monad and $\mathbb{D}$ is a category with powers Ω^I (MacLane 1998, Section X.4 (4)) of a fixed $\mathbb{D}$-object Ω. They construct wppts by composing the comparison functor $\mathbf{Set}_\mathsf{T} \to \mathbf{Set}^\mathsf{T}$ (MacLane 1998, Theorem VI.3) and the functor $\mathbf{Set}^\mathsf{T} \to \mathbb{D}^{\mathrm{op}}$ induced by a relative algebra (Hino et al. 2016, Theorem 4.8). When $\mathbb{D} = \mathbf{Pos}$, their wppts coincide with those of Cartesian liftings of T along the posetal fibration $d^{\mathbf{Set},\Omega}$ in Section 4. Therefore examples in Section 5–8 are also covered by the relative-algebra based wppts. On the other hand, the interaction between generic effects and wppts discussed in Section 4.4 is new compared to Hino et al. (2016). It is the key in the semantic analysis of Kaminski's ert in Section 7.

Example 47 is inspired by the four qualification modes introduced by Unno et al. (2018). They consider a functional programming language supporting nondeterministic choice and recursion and design a refinement-type system to analyze the behavior of programs. Their refinement-type system introduces *qualified types* for various modal properties of nondeterministic and diverging computations. The qualified type is annotated with a *mode index* $Q_1Q_2 \in \{\forall, \exists\}^2$. The index Q_1 specifies whether the return value of a program always ($Q_1 = \forall$) or sometimes ($Q_1 = \exists$) satisfy the refinement. The index Q_2 is about the partial ($Q_2 = \forall$) or total ($Q_2 = \exists$) correctness of a program execution. We find a striking similarity with qualified types and the above four liftings of ${\mathsf{P}^+}_a \circ_\beta \mathsf{M}_b$. Applying these liftings to a sound semantics of Unno et. al.'s refinement-type system would be an interesting future work.

The concept of EM monotone algebra appears in Voorneveld's program logic for a call-by-push-value language (Voorneveld 2019). The logic can have multiple modalities to make assertions about effectful programs, and these modalities are interpreted by endofunctor algebras of the monad T_Σ of possibly infinite and partial Σ-terms. He introduces two conditions on such endofunctor algebras: *leaf-monotonicity* (Voorneveld 2019, Definition 4.5) and *sequentiality* (Voorneveld 2019, Definition 4.10). They are, respectively, equivalent to monotonicity (12) and the EM algebra axiom.

Recent work by Wolter et al. (2020) study the fibrational structure of Hoare logic and predicate transformers in the setting of imperative languages, for partial correctness. Their approach starts by defining indexed categories of programs, syntactic predicates, and semantic predicates and then use Grothendieck constructors to obtain fibrations from them. The relations between these fibrations are used to discuss notions such as soundness and completeness of Hoare logic. Similarly to us, they identify Hoare triples with Cartesian arrows. However, their framework does not cover other side-effects.

While preparing this paper, Batz et al. published a paper about a weakest preweighting semantics of an imperative programming language with a weighting effect and branching effect (Batz et al. 2022). They point out that their preweighting semantics can be an instance of the fibrational wppt semantics of this paper.

Our categorical framework is designed to provide an underlying semantic structure for the Hoare logic where formulas can only examine memory configurations. On the other hand, several extensions of program logics are introduced for the languages with exception (Leino and van de Snepscheut 1994; Maillard et al. 2019; Sekerinski 2012). In these extensions, formulas in the logic can also examine exceptions raised by programs – for instance, in Sekerinski (2012), the Hoare logic is extended so that we can specify two postconditions for normal and exceptional termination. It is an interesting challenge to understand these extended Hoare logic within our fibrational framework.

Turning our eyes to functional programming languages, a series of recent papers (Ahman et al. 2017; Maillard et al. 2019, 2020) develop *Dijkstra monads* for the verification of effectful programs in dependent-type theories. The main task of a Dijkstra monad is to take a type A and a specification $w \in \mathsf{W}A$ written in the language of a *specification monad* W and collect computations that satisfy w. In Maillard et al. (2019, Section 5), they establish an equivalence between Dijkstra monads and monad morphisms into ordered monads. Dijkstra monads intersect with our categorical wppts when the former arise from monad morphisms of type $\mathsf{T} \to \mathsf{W}^{\mathrm{pure}}$, where $\mathsf{W}^{\mathrm{pure}}$ is an ordered continuation monad (see Maillard et al. 2019, Section 4.1 for detail). In this situation, such a monad morphism bijectively corresponds to an EM monotone T-algebra o over the return type of $\mathsf{W}^{\mathrm{pure}}$ (Maillard et al. 2019, Section 4.4). The corresponding Dijkstra monad then collects all the computations $c \in \mathsf{T}A$ such that the predicate transformer $\mathrm{wp}(-, c)$ induced from o is below the specification $w \in \mathsf{W}^{\mathrm{pure}}A$. This is the only relationship we know between Dijkstra monads and our fibrational theory of wppts, and we leave exploring further relationships between them as future work.

In this paper, we only considered posetal fibrations, because when verifying program properties, we do not care very much about proof terms. However, in principle, it is also possible to consider proof terms during the verification of program properties. Non-posetal fibrations will then be used for modeling such proof-relevant Hoare logic and wppts. It is an interesting future work to see how much of this paper's results extend to non-posetal fibrations.

12. Conclusion

We have presented a fibrational and monadic semantics of Hoare triples and weakest precondition predicate transformers. The key fact is that the composability of wppts is equivalent to the Cartesianness of the monad lifting. We next studied the Cartesian liftings of monads along domain fibrations; they bijectively correspond to EM monotone algebras, giving a fibrational view of Hasuo (2015). Despite examples presented here being within the framework of Hasuo (2015), Hino et al. (2016), we studied new examples, in which we have revealed the monads behind wppt-like operators used to compute expected run-time (Kaminski et al. 2016) and higher moments (Kura et al. 2019).

Acknowledgements. This paper was first submitted while the first author was affiliated to the Imdea Software Institute. The authors are grateful to Mayuko Kori for spotting a mistake in the example in Section 8.

Financial support. This research was supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST. This research was conducted during the first author's visit to the National Institute of Informatics.

Conflicts of interest. The authors declare none.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Notes

1 Here side effects refer to those that *cannot* be modeled by memory update functions of type $M \to M$, such as input, output, nondeterministic choice, probabilistic choice, and manipulating *an external memory device*. The last one may be modeled by the state monad $S \Rightarrow (- \times S)$ employing the set S of states taken independently from M.
2 That is, p is a **MSLat**-fibration, where **MSLat** is the subcategory of **Pos** consisting of posets with finite meets and finite-meet preserving monotone functions.
3 In Jacobs (1999), the word "domain fibration" refers to the functor from a (strict) slice category.
4 A category $\mathbb{C}$ has small powers of $\Omega \in \mathbb{C}$ if for any set I, $\mathbb{C}$ has a product of I-many copies of Ω; see MacLane (1998, Section III.4).

References

Abramsky, S., Gay, S. and Nagarajan, R. (1996). Specification structures and propositions-as-types for concurrency. In *Logics for Concurrency: Structure versus Automata*, Berlin, Heidelberg, Springer Berlin Heidelberg, 5–40.
Aguirre, A., Barthe, G., Hsu, J., Kaminski, B. L., Katoen, J.-P. and Matheja, C. (2021). A pre-expectation calculus for probabilistic sensitivity. *Proceedings of the ACM on Programming Languages* **5** (POPL) 1–28.
Aguirre, A. and Katsumata, S. (2020). Weakest preconditions in fibrations. *Electronic Notes in Theoretical Computer Science* **352** 5–27. The 36th Mathematical Foundations of Programming Semantics Conference, 2020.
Ahman, D., Hritcu, C., Maillard, K., Martínez, G., Plotkin, G. D., Protzenko, J., Rastogi, A. and Swamy, N. (2017). Dijkstra monads for free. In: Castagna, G. and Gordon, A. D. (eds.) *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18–20, 2017*, ACM, 515–529.
Batz, K., Gallus, A., Kaminski, B. L., Katoen, J.-P. and Winkler, T. (2022). Weighted programming: A programming paradigm for specifying mathematical models. *Proceedings of the ACM on Programming Languages* **6** (OOPSLA1) 1–30.
Beck, J. (1969). Distributive laws. In: Eckmann, B. (ed.) *Seminar on Triples and Categorical Homology Theory*, Berlin, Heidelberg, Springer Berlin Heidelberg, 119–140.
Bonchi, F., König, B. and Petrisan, D. (2018). Up-to techniques for behavioural metrics via fibrations. In: Schewe, S. and Zhang, L. (eds.) *29th International Conference on Concurrency Theory, CONCUR 2018, September 4–7, 2018, Beijing, China*, LIPIcs, vol. 118, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 17:1–17:17.
Dijkstra, E. W. (1975). Guarded commands, nondeterminacy and formal derivation of programs. *Communications of the ACM* **18** (8) 453–457.
Filinski, A. (1996). *Controlling Effects*. Phd thesis, Carnegie Mellon University.
Filinski, A. (2007). On the relations between monadic semantics. *Theoretical Computer Science* **375** (1–3) 41–75.
Goncharov, S. and Schröder, L. (2013). A relatively complete generic hoare logic for order-enriched effects. In: *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25–28, 2013*, IEEE Computer Society, 273–282.
Goubault-Larrecq, J., Lasota, S. and Nowak, D. (2008). Logical relations for monadic types. *Mathematical Structures in Computer Science* **18** (6) 1169–1217.
Goy, A. and Petrisan, D. (2020). Combining probabilistic and non-deterministic choice via weak distributive laws. In: Hermanns, H., Zhang, L., Kobayashi, N. and Miller, D. (eds.) *LICS'20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8–11, 2020*, ACM, 454–464.
Hasuo, I. (2015). Generic weakest precondition semantics from monads enriched with order. *Theoretical Computer Science* **604** 2–29. Coalgebraic Methods in Computer Science.
Hermida, C. (1993). *Fibrations, Logical Predicates and Indeterminants*. Phd thesis, University of Edinburgh.
Hino, W., Kobayashi, H., Hasuo, I. and Jacobs, B. (2016). Healthiness from duality. In: Grohe, M., Koskinen, E. and Shankar, N. (eds.) *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'16, New York, NY, USA, July 5–8, 2016*, ACM, 682–691.
Jacobs, B. (1994). Semantics of weakening and contraction. *Annals of Pure and Applied Logic* **69** (1) 73–106.
Jacobs, B. (1999). *Categorical Logic and Type Theory*, Studies in Logic and the Foundations of Mathematics, vol. 141, Amsterdam, North Holland.
Joyal, A., Street, R. and Verity, D. (1996). Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society* **119** (3) 447–468.
Kaminski, B. L., Katoen, J.-P., Matheja, C. and Olmedo, F. (2016). Weakest precondition reasoning for expected run-times of probabilistic programs. In: Thiemann, P. (ed.) *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2–8, 2016, Proceedings*, Lecture Notes in Computer Science, vol. 9632, Springer, 364–389.
Katsumata, S. (2005). A semantic formulation of $\top\top$-lifting and logical predicates for computational metalanguage. In: *Proceedings of CSL'05*, LNCS, vol. 3634, Springer, 87–102.
Katsumata, S., Sato, T. and Uustalu, T. (2018). Codensity lifting of monads and its dual. *Logical Methods in Computer Science* **14** (4).
Kock, A. (1970). Strong functors and monoidal monads. *Archiv der Mathematik* **23** 113–120.

https://doi.org/10.1017/S0960129522000330 Published online by Cambridge University Press

Kura, S. (2022). *Semantic Refinements for Program Verification*. Phd thesis, The Graduate University for Advanced Studies, SOKENDAI.

Kura, S., Urabe, N. and Hasuo, I. (2019). Tail probabilities for randomized program runtimes via martingales for higher moments. In: Vojnar, T. and Zhang, L. (eds.) *Tools and Algorithms for the Construction and Analysis of Systems*, Cham, Springer International Publishing, 135–153.

Lane, S. M. and Moerdijk, I. (1994). *Sheaves in Geometry and Logic*, Universitext, New York, NY, Springer.

Leino, K. R. M. and van de Snepscheut, J. L. A. (1994). Semantics of exceptions. In: Olderog, E.-R. (ed.) *Programming Concepts, Methods and Calculi, Proceedings of the IFIP TC2/WG2.1/WG2.2/WG2.3 Working Conference on Programming Concepts, Methods and Calculi (PROCOMET'94) San Miniato, Italy, 6–10 June, 1994*, IFIP Transactions, vol. A-56, North-Holland, 447–466.

Leinster, T. (2004). *Higher Operads, Higher Categories*, London Mathematical Society Lecture Note Series, Cambridge, Cambridge University Press.

MacLane, S. (1998). *Categories for the Working Mathematician*, 2nd ed., Graduate Texts in Mathematics, vol. 5, New York, NY, Springer.

Maillard, K., Ahman, D., Atkey, R., Martínez, G., Hritcu, C., Rivas, E. and Tanter, É. (2019). Dijkstra monads for all. *Proceedings of the ACM on Programming Languages* **3** (ICFP) 104:1–104:29.

Maillard, K., Hritcu, C., Rivas, E. and Van Muylder, A. (2020). The next 700 relational program logics. *Proceedings of the ACM on Programming Languages* **4** (POPL) 4:1–4:33.

Manes, E. and Mulry, P. (2007). Monad compositions I: general constructions and recursive distributive laws. *Theory and Applications of Categories* **18** 172–208.

Martin, U., Mathiesen, E. A. and Oliva, P. (2006). Hoare logic in the abstract. In: Ésik, Z. (ed.) *Computer Science Logic*, Berlin, Heidelberg, Springer Berlin Heidelberg, 501–515.

McIver, A. and Morgan, C. (2005). *Abstraction, Refinement and Proof for Probabilistic Systems*, New York, NY, Springer.

Melliès, P.-A. and Zeilberger, N. (2015). Functors are type refinement systems. In: Rajamani, S. K. and Walker, D. (eds.) *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15–17, 2015*, ACM, 3–16.

Moggi, E. (1991). Notions of computation and monads. *Information and Computation* **93** (1) 55–92.

Moggi, E. (1995). A semantics for evaluation logic. *Fundamenta Informaticae* **22** (1/2) 117–152.

Moggi, E., Taha, W. and Thunberg, J. (2020). Sound over-approximation of probabilities. *Acta Cybernetica* **24** (3) 269–285.

Pitts, A. M. (1991). Evaluation Logic. In: van Rijsbergen, C. J. and Birtwistle, G. (eds.) *IV Higher Order Workshop, Banff 1990*, Workshops in Computing, London, Springer London, 162–189.

Pitts, A. M. (1999). Tripos theory in retrospect. *Electronic Notes in Theoretical Computer Science* **23** (1) 111–127. Tutorial Workshop on Realizability Semantics and Applications (associated to FLoC'99, the 1999 Federated Logic Conference).

Plotkin, G. and Power, J. (2001). Semantics for algebraic operations. In: Brooks, S. and Mislove, M. (eds.) *Proceedings of the Seventeenth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2001, Aarhus, Denmark, May 23–26, 2001*, Electronic Notes in Theoretical Computer Science, vol. 45, Elsevier, 332–345.

Rauch, C., Goncharov, S. and Schröder, L. (2016). Generic hoare logic for order-enriched effects with exceptions. In: James, P. and Roggenbach, M. (eds.) *Recent Trends in Algebraic Development Techniques - 23rd IFIP WG 1.3 International Workshop, WADT 2016, Gregynog, UK, September 21–24, 2016, Revised Selected Papers*, Lecture Notes in Computer Science, vol. 10644, Springer, 208–222.

Sato, T. (2011). A probabilistic monad with nondeterminism for coalgebraic trace semantics. tetsuya sato. In: *CALCO Young Researchers Workshop CALCO-jnr 2011 Selected Papers*, University of Southampton.

Sekerinski, E. (2012). Exceptions for dependability. In: *Dependability and Computer Engineering: Concepts for Software-Intensive Systems*, IGI Global, 11–35.

Street, R. (1972). The formal theory of monads. *Journal of Pure and Applied Algebra* **2** (2) 149–168.

Unno, H., Satake, Y. and Terauchi, T. (2018). Relatively complete refinement type system for verification of higher-order non-deterministic programs. *PACMPL* **2** (POPL) 12:1–12:29.

Varacca, D. and Winskel, G. (2006). Distributing probability over non-determinism. *Mathematical Structures in Computer Science* **16** (1) 87–113.

Voorneveld, N. F. W. (2019). Quantitative logics for equivalence of effectful programs. In: König, B. (ed.) *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4–7, 2019*, Electronic Notes in Theoretical Computer Science, vol. 347, Elsevier, 281–301.

Whiskering. `https://ncatlab.org/nlab/show/whiskering`.

Wolter, U. E., Martini, A. R. and Häusler, E. H. (2020). Indexed and fibred structures for Hoare logic. *Electronic Notes in Theoretical Computer Science* **348** 125–145.

Cite this article: Aguirre A, Katsumata S and Kura S (2022). Weakest preconditions in fibrations. *Mathematical Structures in Computer Science* **32**, 472–510. https://doi.org/10.1017/S0960129522000330