

Healthiness Conditions for Predicate Transformers

Klaus Keimel^{1,2}

*Fachbereich Mathematik
Technische Universität Darmstadt
64289 Darmstadt, Germany*

Abstract

The behavior of a program can be modeled by describing how it transforms input states to output states, the *state transformer semantics*. Alternatively, for verification purposes one is interested in a 'predicate transformer semantics' which, for every condition on the output, yields the weakest precondition on the input that guarantees the desired property for the output.

In the presence of computational effects like nondeterministic or probabilistic choice, a computation will be modeled by a map $t: X \rightarrow \mathcal{T}Y$, where \mathcal{T} is an appropriate computational monad. The corresponding predicate transformer assigns predicates on Y to predicates on X . One looks for necessary and, if possible, sufficient conditions (healthiness conditions) on predicate transformers that correspond to state transformers $t: X \rightarrow \mathcal{T}Y$.

In this paper we propose a framework for establishing healthiness conditions for predicate transformers. As far as the author knows, it fits to almost all situations in which healthiness conditions for predicate transformers have been worked out. It may serve as a guideline for finding new results; but it also shows quite narrow limitations.

Keywords: predicate transformers, healthiness conditions, continuation monad, commuting operations, entropic algebras

1 Introduction: An example

In denotational semantics we distinguish two complementary approaches that we shortly call state transformer semantics and predicate transformer semantics. Let us begin with the well-known example of angelic nondeterminism to explain our intentions. As semantic domains we will use directed complete partially ordered sets (dcpos), maps will be Scott-continuous, that is, they preserve the order and suprema of directed subsets.

In the presence of nondeterministic choice, running a program for an input x belonging to a domain X will lead to a set $t(x)$ of possible outputs in a domain Y . In

¹ Supported by Deutsche Forschungsgemeinschaft

² Email: keimel@mathematik.tu-darmstadt.de

the angelic interpretation of nondeterminism, $t(x)$ will be a non-empty Scott-closed subset of the dcpo Y and t will be a Scott-continuous map from the dcpo X to the *Hoare powerdomain* $\mathcal{H}Y$ of all nonempty Scott-closed subsets of Y . The binary choice operator is interpreted by union on the Hoare powerdomain. Thus, a program will be interpreted by a *state transformer*, a Scott-continuous map $t: X \rightarrow \mathcal{H}Y$.

Observable predicates on Y are Scott-open subsets U of Y (see, e.g., [25]). Thus, the complete lattice $\mathcal{O}Y$ of all Scott-open subsets of Y represents the dcpo of predicates on Y . A Scott-continuous map $p: \mathcal{O}Y \rightarrow \mathcal{O}X$ transforming predicates on Y to predicates on X will be a *predicate transformer*.

To a state transformer $t: X \rightarrow \mathcal{H}Y$ we associate the predicate transformer $p: \mathcal{O}Y \rightarrow \mathcal{O}X$ defined by:

$$p(U) = \{x \in X \mid t(x) \cap U \neq \emptyset\}$$

the set of all points in X that lead to at least one output with the desired property U (the angelic point of view). The state transformer t can be recovered from the associated predicate transformer p by

$$t(x) = \bigcap \{Y \setminus U \mid x \notin p(U)\}$$

We are concerned with the problem to find properties (healthiness conditions) that characterize those predicate transformers $p: \mathcal{O}Y \rightarrow \mathcal{O}X$ that correspond to state transformers $t: X \rightarrow \mathcal{H}Y$. The answer in this case is:

The predicate transformers $p: \mathcal{O}Y \rightarrow \mathcal{O}X$ that correspond to state transformers $t: X \rightarrow \mathcal{H}Y$ are characterized by the properties:

$$p(\emptyset) = \emptyset, \quad p(U \cup U') = p(U) \cup p(U')$$

Equivalently, these are the maps p preserving arbitrary unions.

The above considerations become more elegant, but less intuitive, by passing to a functional setting. We use that the category **DCPO** of dcpos and Scott-continuous maps is Cartesian closed. The exponential of two dcpos X and Y ,

$$\text{denoted by } Y^X \text{ and equally by } [X \rightarrow Y]$$

is the dcpo of all Scott-continuous maps $u: X \rightarrow Y$ with the pointwise defined order (one may consult [3] for background on dcpos).

We endow the two element domain $\mathbf{2} = \{0 < 1\}$ with the structure of a unital semilattice with $x \vee y = \max(x, y)$ and the constant (unit) 0. A predicate (a Scott-open subset U) of a dcpo Y is identified with the Scott-continuous map $f_U: Y \rightarrow \mathbf{2}$ with value 1 iff $x \in U$. Thus the dcpo $\mathcal{O}Y$ of predicates is identified with the function space $\mathbf{2}^Y$. This function space is also a unital semilattice when equipped with the pointwise defined operation \vee and the constant function 0. The Scott-continuous unital semilattice homomorphisms $\varphi: \mathbf{2}^Y \rightarrow \mathbf{2}$ form a dcpo $[\mathbf{2}^Y \xrightarrow{\vee, 0} \mathbf{2}]$ which is also a unital semilattice for the pointwise defined semilattice operations. We will use that it is isomorphic to the Hoare powerdomain $\mathcal{H}Y$; indeed, these homomorphisms φ correspond to the Scott-closed subsets of Y by assigning to φ the Scott-closed set $C = Y \setminus \bigcup \{U \in \mathcal{O}Y \mid \varphi(f_U) = 0\}$.

Now a state transformer will be a Scott-continuous map $t: X \rightarrow [\mathbf{2}^Y \xrightarrow{\vee, 0} \mathbf{2}]$ and a

predicate transformer a Scott-continuous map $p: \mathbf{2}^Y \rightarrow \mathbf{2}^X$. For a state transformer t the corresponding predicate transformer p is given by

$$p(g)(x) = t(x)(g) \text{ for all } g \in [\mathbf{2}^Y \xrightarrow{\vee, 0} \mathbf{2}], x \in X$$

We can recover t from p by reading this equation from right to left. For a state transformer $t: X \rightarrow [\mathbf{2}^Y \xrightarrow{\vee, 0} \mathbf{2}]$ the corresponding predicate transformer p is a unital \vee -homomorphism. Indeed, using that $t(x)$ is a unital \vee -homomorphism for every $x \in X$, we have $p(0)(x) = t(x)(0) = 0$ and $p(g \vee g')(x) = t(x)(g \vee g') = t(x)(g) \vee t(x)(g') = p(g)(x) \vee p(g')(x) = (p(g) \vee p(g'))(x)$ for all $x \in X$, whence $p(0) = 0$ and $p(g \vee g') = p(g) \vee p(g')$. Conversely, a similar calculation shows that, for a Scott-continuous unital \vee -semilattice homomorphism $p: \mathbf{2}^Y \rightarrow \mathbf{2}^X$, the map $t(x)$ defined by

$$t(x)(g) = p(g)(x) \text{ for all } g \in \mathbf{2}^Y$$

is indeed a unital \vee -homomorphism for every $x \in X$ and $t: X \rightarrow [\mathbf{2}^Y \xrightarrow{\vee, 0} \mathbf{2}]$ is a state transformer. Thus:

The predicate transformers $p: \mathbf{2}^Y \rightarrow \mathbf{2}^X$ corresponding to state transformers $t: X \rightarrow [\mathbf{2}^Y \xrightarrow{\vee, 0} \mathbf{2}]$ are characterized by the properties

$$p(0) = 0, \quad p(g \vee g') = p(g) \vee p(g')$$

This functional approach is easily generalized to other situations. As above, in most applications an essential step will be the translation of the given situation into a functional setting.

2 The problem

We work in the category DCPO of dcpos and Scott-continuous functions, a category which is commonly used in semantics. We agree on some assumptions that will be tacitly assumed throughout the paper.

Convention 2.1 *All maps between dcpos will tacitly supposed to be Scott-continuous. All definitions of functions are expressible in the language of typed λ -calculus so that these functions are automatically Scott-continuous, since we are in a Cartesian closed category (see, for example, [19, Part I]). For this reason, we will never verify continuity of functions explicitly.*

R will be a fixed dcpo, called the dcpo of observations;

X and Y denote arbitrary dcpos;

x, y and r denote elements of X, Y and R , respectively;

u denotes Scott-continuous maps $u: X \rightarrow Y$;

f and g denote Scott-continuous maps $f: X \rightarrow R$ and $g: Y \rightarrow R$;

φ denotes Scott-continuous maps $\varphi: R^X \rightarrow R$.

Predicates on a dcpo Y will be R -valued³, that is, Scott-continuous functions $g: Y \rightarrow$

³ Since R can be very different from the two-element set of truth values, this notion of a predicate is very wide, and one instead uses terms like ‘prevision’ [4], ‘expectation’ [21], ‘random variable’ [18] instead, depending on the concrete situation.

R. The function space (exponential) for which we will use two notations in parallel, $R^Y = [Y \rightarrow R]$, will be the domain of predicates on Y .

The contravariant functor $R^{(-)}$ assigns to every dcpo X the exponential R^X and to every map $u: X \rightarrow Y$ of dcpos the map $R^u: R^Y \rightarrow R^X$ defined by $R^u(g) = g \circ u$ for all $g \in R^Y$. Applying the contravariant functor $R^{(-)}$ twice yields a covariant functor $R^{R^{(-)}} = [R^{(-)} \rightarrow R]$, the *continuation monad*⁴ over R . The unit of the monad represents X as a sub-dcpo of R^{R^X} and is given by the map $\delta: X \rightarrow R^{R^X}$ that assigns to every $x \in X$ the *projection* or *evaluation map* $\delta_x: R^X \rightarrow R$ defined by

$$(1) \quad \delta_x(f) = f(x)$$

For a map $t: X \rightarrow [R^Y \rightarrow R]$ its Kleisli lifting $t^\dagger: [R^X \rightarrow R] \rightarrow [R^Y \rightarrow R]$ is given by

$$(2) \quad t^\dagger(\varphi)(g) = \varphi(\lambda x. t(x)(g)).$$

According to our setting, a program will be interpreted by a *state transformer*, a Scott continuous map $t: X \rightarrow R^{R^Y}$. A predicate transformer will be a Scott-continuous map $p: R^Y \rightarrow R^X$. To every state transformer corresponds a predicate transformer that assigns the weakest precondition to every postcondition:

Lemma 2.2 *The dcpos of state and predicate transformers are canonically isomorphic:*

$$(3) \quad [R^Y \rightarrow R]^X \cong [R^Y \rightarrow R^X]$$

A state transformer $t: X \rightarrow [R^Y \rightarrow R]$ and a predicate transformer $p: R^Y \rightarrow R^X$ correspond to one another under this isomorphism if and only if

$$(4) \quad t(x)(g) = p(g)(x) \quad \text{for } x \in X, g \in R^Y.$$

Proof. For a state transformer $t: X \rightarrow [R^Y \rightarrow R]$ let $p = P(t): R^Y \rightarrow R^X$ be defined by $P(t) = \lambda g. \lambda x. t(x)(g)$. As we are in a Cartesian closed category and $P(t)$ is defined by a λ -calculus term, $P(t)$ is Scott-continuous. Similarly if, for a predicate transformer $p: R^Y \rightarrow R^X$, we define $t = T(p) = \lambda x. \lambda g. p(g)(x)$, then $T(p)(x): R^Y \rightarrow R$ is in fact Scott-continuous for every $x \in X$ and $T(p)$ is a Scott-continuous map from X to $[R^Y \rightarrow R]$. Moreover, $P(T(p)) = p$ for every predicate transformer p , since $P(T(p))(g)(x) = T(p)(x)(g) = p(g)(x)$ for every $x \in X$ and every $g \in R^Y$. Similarly, $T(P(t)) = t$ for every state transformer t . One may notice that equation (3) is an isomorphism of dcpos, since P and T are λ -definable hence Scott-continuous mutually inverse maps. \square

Our domain R of observations will carry additional structure, it will be a $d\text{-}\Omega$ -algebra: a dcpo with an algebraic structure of signature Ω which interprets the constructors in the programming language (see Section 3). The exponentials R^X and R^{R^X} become $d\text{-}\Omega$ -algebras, too, with pointwise defined operations. A program will be interpreted by a state transformer $t: X \rightarrow \mathcal{F}_R Y$, where $\mathcal{F}_R Y$ is the $d\text{-}\Omega$ -subalgebra of R^{R^Y} generated by the projections δ_y , $y \in Y$.

⁴ See [20], for example, for background on monads.

The assignment $X \mapsto \mathcal{F}_R X$ gives rise to a monad that we call *subordinate* to the continuation monad. This paper deals with the

Problem 2.3 *Find conditions (healthiness conditions) that characterize the predicate transformers $p: R^Y \rightarrow R^X$ that correspond to the state transformers $t: X \rightarrow \mathcal{F}_R Y$.*

We cannot offer a complete answer to this question. But we exhibit a framework which always yields necessary conditions that the predicate transformers corresponding to state transformers $t: X \rightarrow \mathcal{F}_R Y$ must satisfy. And we give a criterion for these necessary conditions to be also sufficient. This criterion has to be checked separately in each special situation.

Let us make precise what we mean by a *monad \mathcal{F} subordinate to the continuation monad*: Suppose that \mathcal{F} assigns to every dcpo X a sub-dcpo $\mathcal{F}X$ of $[R^X \rightarrow R]$ in such a way that the following properties are satisfied:

$$\begin{aligned} \delta_x &\in \mathcal{F}X \quad \text{for all } x \in X; \\ t^\dagger(\mathcal{F}X) &\subseteq \mathcal{F}Y \quad \text{for every } t: X \rightarrow [R^Y \rightarrow R]. \end{aligned}$$

Then R^{R^u} maps $\mathcal{F}X$ into $\mathcal{F}Y$ for every map $u: X \rightarrow Y$; indeed, R^{R^u} is the Kleisli lifting of $\delta \circ u: X \rightarrow [R^Y \rightarrow R]$. Thus R^{R^u} induces a map $\mathcal{F}u: \mathcal{F}X \rightarrow \mathcal{F}Y$ in such a way that \mathcal{F} becomes a functor, and even a monad with (the corestriction of) δ as unit and the (restriction-corestriction of the) Kleisli lifting $t^\dagger|_{\mathcal{F}X}: \mathcal{F}X \rightarrow \mathcal{F}Y$ for $t: X \rightarrow \mathcal{F}Y$.

Our methods can be applied to quite some examples in the literature, in particular for nondeterminism and probability [4,5,9,13,14,15,16,18,21,25]. There, the monads are usually presented in the form of powerdomains. For applying our results one has to find functional representations for these powerdomains of the type $\mathcal{F}_R X$, as we have seen in the Introduction. This paper is based on previous work by the author [11]. There, several examples are worked out explicitly which is not possible in this paper because of space restrictions. The reader is invited to consult that source for examples.

Acknowledgments

The author would like to thank Gordon Plotkin for many insights. Thomas Streicher has listened to many questions and patiently discussed possible answers. The referees' suggestions were quite helpful.

3 Algebraic structure

We recall a few concepts from universal algebra adapted to the category DCPO.

An *operation of arity* $n \in \mathbb{N}$ on a dcpo A is a map $\omega: A^n \rightarrow A$. If A and B both carry an n -ary operation ω , a map $h: A \rightarrow B$ is an ω -*homomorphism* if, for all $(a_1, \dots, a_n) \in A^n$, we have:

$$(5) \quad h(\omega(a_1, \dots, a_n)) = \omega(h(a_1), \dots, h(a_n))$$

Definition 3.1 A *d-signature* Ω is a sequence of dcpos $\Omega_n, n \in \mathbb{N}$. The elements $\omega \in \Omega_n$ are the *operation symbols of arity n* .

Definition 3.2 A *d-algebra of d-signature Ω* (a *d- Ω -algebra*, for short) consists of a dcpo A together with operations $\omega^A: A^n \rightarrow A$, one for each $\omega \in \Omega_n$, such that

$$(\omega, a_1, \dots, a_n) \mapsto \omega^A(a_1, \dots, a_n): \Omega_n \times A^n \rightarrow A$$

is Scott-continuous for every n . A map $u: A \rightarrow B$ of d- Ω -algebras is an *Ω -homomorphism*, if it is an ω -homomorphism for every $\omega \in \Omega$.

We stress that the value $\omega^A(a_1, \dots, a_n)$ depends continuously not only on the arguments a_i but also on $\omega \in \Omega_n$. By choosing the Ω_n to be (unordered) sets we recover the usual notion of a signature Ω in universal algebra.

Convention 3.3 We will omit the superscript when denoting operations ω^A on a d- Ω -algebra A and simply write ω instead of ω^A .

In proofs, we will use a binary operation, denoted by $+$, instead of an arbitrary n -ary operation ω . In this way, proofs become easier to read. Of course, we will not use any special property like commutativity that one usually associates with an operation $+$. This does not affect the general validity of our proofs; one just has to replace $x_1 + x_2$ by $\omega(x_1, \dots, x_n)$ in order to obtain the general proof.

We fix a d-signature Ω and a d- Ω -algebra R . For every dcpo X , the function space R^X also becomes a d- Ω -algebra. For $\omega \in \Omega_n$ the operation ω on the function space R^X is defined pointwise: For all $f_1, \dots, f_n \in R^X$ and all $x \in X$,

$$(6) \quad \omega(f_1, \dots, f_n)(x) = \omega(f_1(x), \dots, f_n(x)).$$

For every map $u: X \rightarrow Y$, the map $R^u: R^Y \rightarrow R^X$ is an Ω -homomorphism. Thus, we may view $R^{(-)}$ to be contravariant functor from the category DCPO to the category of d- Ω -algebras and Ω -homomorphisms.

In the same way, the operations ω can be extended to the function space $R^{R^X} = [R^X \rightarrow R]$ so that the latter becomes a d- Ω -algebra, too, and the maps R^{R^u} are Ω -homomorphisms.

Lemma 3.4 The Kleisli lifting $t^\dagger: [R^X \rightarrow R] \rightarrow [R^Y \rightarrow R]$ is an Ω -homomorphism for every $t: X \rightarrow [R^Y \rightarrow R]$.

Proof. We check that, for every binary operation $+$ in Ω_2 and all φ_1, φ_2 , we have $t^\dagger(\varphi_1 + \varphi_2) = t^\dagger(\varphi_1) + t^\dagger(\varphi_2)$. For every $g \in R^Y$ we have indeed: $t^\dagger(\varphi_1 + \varphi_2)(g) = (\varphi_1 + \varphi_2)(\lambda x. t(x)(g)) = \varphi_1(\lambda x. t(x)(g)) + \varphi_2(\lambda x. t(x)(g)) = t^\dagger(\varphi_1)(g) + t^\dagger(\varphi_2)(g) = (t^\dagger(\varphi_1) + t^\dagger(\varphi_2))(g)$. \square

4 R -Free algebras

We are interested in the monad that represents the free objects over a dcpo X relative to our d- Ω -algebra R of observations that we keep fixed throughout this section.

A subalgebra of a $d\text{-}\Omega$ -algebra A which is a sub-dcpo, too, is called a *d- Ω -subalgebra*. The intersection of any family of $d\text{-}\Omega$ -subalgebras is again a $d\text{-}\Omega$ -subalgebra. Thus every subset of A generates a $d\text{-}\Omega$ -subalgebra, the intersection of all $d\text{-}\Omega$ -subalgebras containing the subset.

Definition 4.1 The $d\text{-}\Omega$ -subalgebra $\mathcal{F}_R X$ of $[R^X \rightarrow R]$ generated by the projections δ_x , $x \in X$, is called the *free d- Ω -algebra over X with respect to R* or simply the *R -free d- Ω -algebra over X* .

For a map $t: X \rightarrow \mathcal{F}_R Y$, the Kleisli lifting $t^\dagger: [R^X \rightarrow R] \rightarrow [R^Y \rightarrow R]$ maps $\mathcal{F}_R X$ into $\mathcal{F}_R Y$, since t^\dagger is an Ω -homomorphism by Lemma 3.4. This shows:

Proposition 4.2 $(\mathcal{F}_R, \delta, \dagger)$ is a monad over the category DCPO subordinate to the continuation monad in the sense made precise at the end of Section 2, the Kleisli lifting of a map $t: X \rightarrow \mathcal{F}_R Y$ being the restriction and corestriction of the Kleisli lifting t^\dagger for the continuation monad $R^{R^{(-)}}$.

Since we have a monad, the $d\text{-}\Omega$ -algebras $\mathcal{F}_R X$ are free for the class of its Eilenberg-Moore algebras. It is a challenge to determine these Eilenberg-Moore algebras concretely. A natural conjecture would be that $\mathcal{F}_R X$ is free over X for the class of $d\text{-}\Omega$ -algebras determined by the (in)equational theory of the $d\text{-}\Omega$ -algebra R . This conjecture is supported by a theorem due to G. Birkhoff (see [2]) which tells us that, in the category SET, $\mathcal{F}_R X$ is free over the set X in the class of all Ω -algebras that satisfy the equational laws that hold in R . Such a strong statement will not hold in the dcpo-setting, in general, although it holds in many examples.

The following proposition (that we state without proof) shows that $\mathcal{F}_R X$ is free in the class of $d\text{-}\Omega$ -algebras that are embeddable in some power of R . This class is sometimes called the quasi-variety generated by R . The algebras in this class satisfy not only all equational and inequational laws that hold in R , but also all implications between two such laws (Horn formulas) that hold in R .

Proposition 4.3 Let u be a map from a dcpo X to a $d\text{-}\Omega$ -algebra A that is embeddable in some R^Y as a $d\text{-}\Omega$ -subalgebra. Then there is a unique Ω -homomorphism $\hat{u}: \mathcal{F}_R X \rightarrow A$ extending u along δ .

5 Homomorphism monads

We continue with a fixed $d\text{-}\Omega$ -algebra R of d -signature Ω . In order to find properties characterizing the predicate transformers $p: R^Y \rightarrow R^X$ that correspond to the state transformers $t: X \rightarrow \mathcal{F}_R Y$, we need a second monad subordinate to the continuation monad.

For two $d\text{-}\Omega$ -algebras A and B , we denote by

$$[A \xrightarrow{\Omega} B]$$

the set of all Ω -homomorphisms $u: A \rightarrow B$. The pointwise supremum of a directed family of Ω -homomorphisms is again an Ω -homomorphism. Thus, $[A \xrightarrow{\Omega} B]$ be-

comes a dcpo, a sub-dcpo of the dcpo $[A \rightarrow B]$ of all Scott-continuous maps from A to B .

Proposition 5.1 *For a $d\text{-}\Omega$ -algebra R , the assignment*

$$X \mapsto [R^X \xrightarrow{\Omega} R]$$

yields a monad subordinate to the continuation monad. The unit is (the corestriction of) δ and the Kleisli lifting of a map $t: X \rightarrow [R^Y \xrightarrow{\Omega} R]$ is (the restriction-corestriction) $t^\dagger: [R^X \xrightarrow{\Omega} R] \rightarrow [R^Y \xrightarrow{\Omega} R]$.

Proof. We show that we are in a situation as described at the end of Section 2.

(a) Clearly, the projections $\delta_x: R^X \rightarrow R$ are Ω -homomorphisms for every $x \in X$.

(b) For every state transformer $t: X \rightarrow [R^Y \xrightarrow{\Omega} R]$, the Kleisli lifting t^\dagger maps $[R^X \xrightarrow{\Omega} R]$ to $[R^Y \xrightarrow{\Omega} R]$. Indeed, let $\varphi: R^X \rightarrow R$ be an Ω -homomorphism. For a binary operation $+$ in Ω and $g_1, g_2 \in R^Y$, we have:

$$\begin{aligned} & t^\dagger(\varphi)(g_1 + g_2) \\ &= \varphi(\lambda x. t(x)(g_1 + g_2)) && \text{by the definition of } t^\dagger \\ &= \varphi(\lambda x. (t(x)(g_1) + t(x)(g_2))) && \text{since } t(x) \text{ is a homomorphism} \\ &= \varphi(\lambda x. t(x)(g_1) + \lambda x. t(x)(g_2)) && \text{since } + \text{ is defined pointwise} \\ &= \varphi(\lambda x. t(x)(g_1)) + \varphi(\lambda x. t(x)(g_2)) && \text{since } \varphi \text{ is a homomorphism} \\ &= t^\dagger(\varphi(g_1)) + t^\dagger(\varphi(g_2)) && \text{by the definition of } t^\dagger. \end{aligned}$$

□

The ‘homomorphism monad’ $([R^{(-)} \xrightarrow{\Omega} R], \delta, \dagger)$ exhibited in the previous proposition behaves well with respect to the one-to-one correspondence between state and predicate transformers:

Proposition 5.2 *Let R be $d\text{-}\Omega$ -algebra. Under the one-to-one correspondence between state transformers and predicate transformers in Lemma 2.2 the predicate transformers $p: R^Y \rightarrow R^X$ corresponding to the state transformers $t: X \rightarrow [R^Y \xrightarrow{\Omega} R]$ are characterized by the property of being Ω -homomorphisms: $[R^Y \xrightarrow{\Omega} R]^X \cong [R^Y \xrightarrow{\Omega} R^X]$*

Proof. Let $t: X \rightarrow [R^Y \rightarrow R]$ be a state transformer and $p: R^Y \rightarrow R^X$ the corresponding predicate transformer according to Lemma 2.2. We show that, for a binary operation $+$ in Ω , $t(x)$ is a $+$ -homomorphism for every $x \in X$ if, and only if, p is a $+$ -homomorphism.

If $t(x)$ is a $+$ -homomorphism for every $x \in X$ then, for all $g_1, g_2 \in R^Y$, $p(g_1 + g_2)(x) = t(x)(g_1 + g_2) = t(x)(g_1) + t(x)(g_2) = p(g_1)(x) + p(g_2)(x) = (p(g_1) + p(g_2))(x)$, whence $p(g_1 + g_2) = p(g_1) + p(g_2)$, that is, p is a $+$ -homomorphism. If conversely p is a $+$ -homomorphism, then $t(x)(g_1 + g_2) = p(g_1 + g_2)(x) = (p(g_1) + p(g_2))(x) = p(g_1)(x) + p(g_2)(x) = t(x)(g_1) + t(x)(g_2)$ which shows that $t(x)$ is a $+$ -homomorphism for all $x \in X$. □

One may notice that the proof above is identical to the only proof that we gave in the Introduction for that special situation.

6 Commuting operations

We come back the monad \mathcal{F}_R of section 4 for a given d- Ω -algebra R . We want to consider state transformers $t: X \rightarrow \mathcal{F}_R Y$ and the corresponding predicate transformers $p: R^Y \rightarrow R^X$ according to Lemma 2.2. In order to apply the results obtained in the previous section with the homomorphism monad we have to introduce a new framework.

Definition 6.1 Given two operations σ of arity m and ω of arity n on a dcpo A , we say that σ and ω *commute* if for all 'matrices' $(x_{ij})_{i=1,\dots,m, j=1,\dots,n}$ of elements in A , we have:

$$\begin{aligned} & \omega(\sigma(x_{11}, \dots, x_{m1}), \dots, \sigma(x_{1n}, \dots, x_{mn})) \\ &= \sigma(\omega(x_{11}, \dots, x_{1n}), \dots, \omega(x_{m1}, \dots, x_{mn})) \end{aligned}$$

This is equivalent to the statement that $\sigma: A^m \rightarrow A$ is an ω -homomorphism, equivalently, that $\omega: A^n \rightarrow A$ is a σ -homomorphism.

Example 6.2 A constant c commutes with an n -ary operation ω if and only if $\omega(c, \dots, c) = c$. Two commuting constants have to agree. Two unary operations ρ and σ commute if they commute as functions: $\rho \circ \sigma = \sigma \circ \rho$. A unary operation ρ commutes with a binary operation $+$ if and only if

$$(7) \quad \rho(x + y) = \rho(x) + \rho(y)$$

Two binary operation $+$ and $*$ commute if

$$(8) \quad (x_1 * x_2) + (x_3 * x_4) = (x_1 + x_3) * (x_2 + x_4)$$

In particular, a binary relation $*$ commutes with itself if

$$(9) \quad (x_1 * x_2) * (x_3 * x_4) = (x_1 * x_3) * (x_2 * x_4)$$

Thus, every commutative, associative binary operation commutes with itself.

Now let Ω be a d-signature and R a d- Ω -algebra. Let

$$\Sigma_n = [R^n \xrightarrow{\Omega} R]$$

be the dcpo of all Ω -homomorphisms $\sigma: R^n \rightarrow R$, that is, Σ_n consists of all operations of arity n on R that commute with all $\omega \in \Omega$. The Σ_n form a second d-signature Σ and R is a d- Σ -algebra, too. The fact that, on R , the operations in Σ commute with those in Ω is given by equational laws of the form in Definition 6.1. These equational laws are inherited by exponentials R^X and R^{R^X} considered as d- $(\Omega \cup \Sigma)$ -algebras (with pointwise defined operations) so that the operations $\omega \in \Omega$ commute with all the operations $\sigma \in \Sigma$ on all exponentials of R .

At this point it becomes clear, why we wanted to choose signatures which are dcpos and not simply (unordered) sets; indeed, our signature Σ is a dcpo in a natural way.

The homomorphisms between two Ω -algebras do not form an Ω -algebra, in general. The following observation was a surprise to me. But if you think about it, you might find that you always have known it:

Lemma 6.3 *Suppose that R is a d - Ω -algebra and Σ a d -signature of operations σ on R that commute with all $\omega \in \Omega$. Then the set $[R^X \xrightarrow{\Sigma} R]$ of all Σ -homomorphisms $\varphi: R^X \rightarrow R$ is a d - Ω -subalgebra of $[R^X \rightarrow R]$ containing the R -free algebra $\mathcal{F}_R X$.*

Proof. If $\varphi_1, \dots, \varphi_n: R^X \rightarrow R$ are Σ -homomorphisms, then $\omega(\varphi_1, \dots, \varphi_n)$ is also a Σ -homomorphism for every $\omega \in \Omega_n$. Indeed, if ω is a binary operation $+$ then, for every binary operation $*$ in Σ_2 , hence commuting with $+$, we have:

$$\begin{aligned} (\varphi_1 + \varphi_2)(f_1 * f_2) &= \varphi_1(f_1 * f_2) + \varphi_2(f_1 * f_2) \\ &= \varphi_1(f_1) * \varphi_1(f_2) + \varphi_2(f_1) * \varphi_2(f_2) \\ &= (\varphi_1(f_1) + \varphi_2(f_1)) * (\varphi_1(f_2) + \varphi_2(f_2)) \\ &= (\varphi_1 + \varphi_2)(f_1) * (\varphi_1 + \varphi_2)(f_2) \end{aligned}$$

Thus the Σ -homomorphisms $\varphi: R^X \rightarrow R$ form an Ω -subalgebra $[R^X \xrightarrow{\Sigma} R]$. Clearly all the projections δ_x are Σ -homomorphisms. Hence, $[R^X \xrightarrow{\Sigma} R]$ contains $\mathcal{F}_R X$, the d - Ω -subalgebra of $[R^X \rightarrow R]$ generated by the projections. \square

From the previous lemma and Proposition 5.2 we immediately deduce our main result on healthiness conditions:

Theorem 6.4 *Suppose that R is a d - Ω -algebra and Σ a d -signature of operations on R that commute with all $\omega \in \Omega$. Then the predicate transformers $p: R^Y \rightarrow R^X$ corresponding to the state transformers $t: X \rightarrow \mathcal{F}_R Y$ are necessarily Σ -homomorphisms.*

If $\mathcal{F}_R Y = [R^Y \xrightarrow{\Sigma} R]$, then the Σ -homomorphisms $p: R^Y \rightarrow R^X$ are precisely the predicate transformers corresponding to state transformers $t: X \rightarrow \mathcal{F}_R Y$:

$$(\mathcal{F}_R Y)^X \cong [R^Y \xrightarrow{\Sigma} R^X].$$

For applying this Theorem, the challenge is to find operations on R that commute with those in Ω . This then yields necessary healthiness conditions for the predicate transformers. It depends very much on the special situation whether these healthiness conditions are also sufficient: one has to show that the d - Ω -algebra $[R^Y \xrightarrow{\Sigma} R]$ is indeed generated by the projections δ_y , hence equal to the R -free d - Ω -algebra $\mathcal{F}_R Y$. The classical example of observable predicates can be treated in this way, and also the example of convex sets and effect modules as viewed by B. Jacobs [8]. In the first case role of R is taken by the two element dcpo $\mathbf{2} = \{0 < 1\}$ without any algebraic structure, in the second case by the unit interval.

We now look at the special situation where the operations of the d - Ω -algebra commute with one another (see also [23]) :

Definition 6.5 A d - Ω -algebra is called *entropic* if any two operations $\sigma, \omega \in \Omega$ commute.

We note that the entropic d - Ω -algebras are the algebras of a commutative monad over the category **DCPO** in the sense of A. Kock [17].

As a particular case of Lemma 6.3 and Theorem 6.4 with $\Omega = \Sigma$ we have:

Corollary 6.6 *If R is an entropic d - Ω -algebra, the Ω -homomorphisms $\varphi: R^X \rightarrow R$ form a d - Ω -algebra $[R^X \xrightarrow{\Omega} R]$ containing the R -free algebra $\mathcal{F}_R X$ as a d - Ω -subalgebra.*

The predicate transformers $p: R^Y \rightarrow R^X$ corresponding to state transformers $t: X \rightarrow \mathcal{F}_R Y$ are Ω -homomorphisms. If $\mathcal{F}_R Y = [R^Y \xrightarrow{\Omega} R]$ these predicate transformers are precisely the Ω -homomorphisms.

Entropicity is quite a special property. Using Example 6.2 we obtain examples of entropic algebras: commutative semigroups, commutative monoids, commutative groups, modules over commutative rings, semimodules over commutative semirings, semilattices and unital semilattices.

Corollary 6.6 can be used for deriving the healthiness criteria for angelic nondeterminism in the Introduction (Section 1). The only specific property to be proved is that every unital semilattice homomorphism $\varphi: \mathbf{2}^Y \rightarrow \mathbf{2}$ is the supremum of the projections δ_x with $\delta_x \leq \varphi$ which is equivalent to the property that every nonempty Scott-closed subset of a dcpo is the union of the principal ideals $\downarrow x$, $x \in X$. In the same way this Corollary can be used for deriving healthiness criteria for predicate transformers in the case of demonic and erratic (the combination of angelic and demonic) nondeterminism as well as for probabilistic nondeterminism as in [9,10,25].

But the known results for predicate transformers in the presence of both nondeterministic and probabilistic choice do not fit into the framework developed above. The reason is that, for example, on the nonnegative reals, the operation of addition and the semilattice operation \max and \min do not commute. We therefore propose a relaxed framework.

7 Relaxed morphisms and relaxed entropic algebras

We relax the previous framework by replacing equalities by inequalities (compare Definition 6.1):

Definition 7.1 Let ω be an operation of arity n defined on dcpo's A and B . A map $h: A \rightarrow B$ is called an ω -submorphism⁵ if

$$h(\omega(x_1, \dots, x_n)) \leq \omega(h(x_1), \dots, h(x_n)) \quad \text{for all } x_1, \dots, x_n \in A.$$

An ω -supermorphism is defined in the same way replacing the inequality \leq by its opposite \geq .

For d -algebras of d -signature Ω , we want to distinguish some operations $\omega \in \Omega$ for which we would like to consider relaxed morphisms. For this, we suppose that

⁵ For the terminology we have been guided by a common terminology in analysis. A function on a vector space is subadditive if $h(x+y) \leq h(x) + h(y)$ and superadditive if the reverse inequality holds.

the d-signature Ω is the union of two d-sub-signatures Ω^{\leq} and Ω^{\geq} which need not be disjoint.

Definition 7.2 A map $h: A \rightarrow B$ between d-algebras of d-signature $\Omega = \Omega^{\leq} \cup \Omega^{\geq}$ is said to be a *relaxed Ω -morphism* if h is an ω -submorphism for all $\omega \in \Omega^{\leq}$, but an ω -supermorphism for $\omega \in \Omega^{\geq}$. (For ω in both Ω^{\leq} and Ω^{\geq} , h will be an ω -homomorphism.)

The pointwise supremum of a directed family of relaxed Ω -morphisms is again a relaxed Ω -morphism. Thus the set $[A \xrightarrow{r\Omega} B]$ of relaxed Ω -morphisms from A to B is a sub-dcpo of the function space $[A \rightarrow B]$. As in Proposition 5.1 and 5.2 we have:

Proposition 7.3 Let R be a d- Ω -algebra of d-signature $\Omega = \Omega^{\leq} \cup \Omega^{\geq}$.

(a) For every state transformer $t: X \rightarrow [R^Y \xrightarrow{r\Omega} R]$, the Kleisli lifting $t^\dagger: [R^X \rightarrow R] \rightarrow [R^Y \rightarrow R]$ maps relaxed Ω -morphisms to relaxed Ω -morphisms, so that our continuation monad $([R^- \rightarrow R], \delta, \dagger)$ restricts to a monad $([R^{(-)} \xrightarrow{r\Omega} R], \delta, \dagger)$.

(b) Under the bijective correspondence of Lemma 2.2, the predicate transformers $p: R^Y \rightarrow R^X$ corresponding to state transformers $t: X \rightarrow [R^Y \xrightarrow{r\Omega} R]$ are the relaxed Ω -morphisms:

$$[R^Y \xrightarrow{r\Omega} R]^X \cong [R^Y \xrightarrow{r\Omega} R^X]$$

The proofs are the same as for the corresponding claims in 5.1 and 5.2. We just have to replace the equality sign by the appropriate inequality (\leq in case $\omega \in \Omega^{\leq}$ and \geq in case $\omega \in \Omega^{\geq}$) every time that we have used the homomorphism property there.

We now turn to the question under what circumstances the relaxed Ω -morphisms form a subalgebra of $[R^X \rightarrow R]$.

Definition 7.4 We will say that an operation σ of arity m on a dcpo R *subcommutes* with an operation ω of arity n (equivalently, ω *supercommutes* with σ) if, for all $x_{ij} \in R, i = 1, \dots, m, j = 1, \dots, n$:

$$\begin{aligned} & \sigma(\omega(x_{11}, \dots, x_{1n}), \dots, \omega(x_{m1}, \dots, x_{mn})) \\ & \leq \omega(\sigma(x_{11}, \dots, x_{m1}), \dots, \sigma(x_{1n}, \dots, x_{mn})) \end{aligned}$$

This is equivalent to the statement that $\sigma: R^m \rightarrow R$ is an ω -submorphism, and also equivalent to the statement that $\omega: R^n \rightarrow R$ is σ -supermorphism. Whenever this inequational law holds in R , it also holds in R^X and in R^{R^X} .

We now let R be a d- Ω -algebra. For every natural number m , we denote by Σ_m^{\geq} and Σ_m^{\leq} the dcpos of all operations $\sigma: R^m \rightarrow R$ that subcommute, resp., supercommute, with all $\omega \in \Omega$. These give rise to d-signatures Σ^{\leq} , Σ^{\geq} , and $\Sigma = \Sigma^{\leq} \cup \Sigma^{\geq}$. As in Lemma 6.3 we have:

Lemma 7.5 The relaxed Σ -morphisms $\varphi: R^X \rightarrow R$ form a d- Ω -subalgebra $[R^X \xrightarrow{r\Sigma} R]$ of $[R^X \rightarrow R]$.

We now are ready for our main theorem corresponding to Theorem 6.4:

Theorem 7.6 Consider a d -signature $\Omega = \Omega^{\leq} \cup \Omega^{\geq}$ and a d - Ω -algebra R . Let $\Sigma = \Sigma^{\leq} \cup \Sigma^{\geq}$ be a d -signature of operations that subcommute, resp. supercommute with all $\omega \in \Omega$. Then the d - Ω -algebra $\mathcal{F}_R Y$ generated by the projections is a d - Ω -subalgebra of $[R^Y \xrightarrow{r\Sigma} R]$.

The predicate transformers $p: R^Y \rightarrow R^X$ corresponding to state transformers $t: X \rightarrow \mathcal{F}_R Y$ are relaxed Σ -morphisms. If $\mathcal{F}_R Y = [R^Y \xrightarrow{r\Sigma} R]$, then these predicate transformers are precisely the relaxed Σ -morphisms.

We are mainly interested in the following situation where we can choose $\Omega = \Sigma$:

Definition 7.7 A d - Ω -algebra R is said to be *relaxed entropic*, if every $\sigma \in \Omega$ either subcommutes with every $\omega \in \Omega$ or supercommutes with every $\omega \in \Omega$.

From the preceding Theorem we deduce:

Corollary 7.8 Let R be a relaxed entropic d - Ω -algebra. The set $[R^X \xrightarrow{r\Omega} R]$ of all relaxed Ω -morphisms $\varphi: R^X \rightarrow R$ is a d - Ω -subalgebra of $[R^X \rightarrow R]$.

The d - Ω -subalgebra $\mathcal{F}_\Omega X$ of $[R^X \rightarrow R]$ generated by the projections δ_x , $x \in X$, is a d - Ω -subalgebra of $[R^X \xrightarrow{r\Omega} R]$.

The predicate transformers $p: R^Y \rightarrow R^X$ corresponding to state transformers $t: X \rightarrow \mathcal{F}_\Omega Y$ are relaxed Ω -morphisms. If $\mathcal{F}_\Omega Y = [R^Y \xrightarrow{r\Omega} R]$, then these predicate transformers are precisely the relaxed Ω -morphisms.

Whether we have equality $\mathcal{F}_R Y = [R^Y \xrightarrow{r\Omega} R]$, has to be decided separately in each special case.

On the nonnegative real line addition subcommutes with the semilattice operation $x \vee y = \max(x, y)$ and it supercommutes with $x \wedge y = \min(x, y)$. One has indeed for arbitrary nonnegative real numbers:

$$(10) (x_1 + x_2) \vee (x_3 + x_4) \leq (x_1 \vee x_3) + (x_2 \vee x_4)$$

$$(11) (x_1 + x_2) \wedge (x_3 + x_4) \geq (x_1 \wedge x_3) + (x_2 \wedge x_4)$$

These simple facts allow us to use our relaxed setting for deriving healthiness conditions for predicate transformers in the presence of mixed nondeterministic and probabilistic choice as in [4,5,13,14,15,16,21].

8 Concluding remarks

The framework for deriving healthiness conditions for predicate transformers developed in this paper looks quite narrow, although it applies to almost all situations known to the author. (An exception is [7], where one meets a quite different notion of predicate.) There is some evidence that it may not be possible to characterize predicate transformers in situations that do not fit under this framework.

Nevertheless, our methods allow quite some extensions. We have not carried them out in this paper in order to keep it at a technically simple level.

1. Firstly we may allow infinite arities for signatures and consider operations $\omega: R^I \rightarrow R$ for infinite sets I . We may also allow arities to be dcpos; that is, a signature may contain operation symbols ω of arity P , where P is a dcpo; then ω will be interpreted as a map $\omega: R^P \rightarrow R$. For example, we may choose P to be the two element dcpo $\mathbf{2} = \{0 < 1\}$; an operation $\omega: R^{\mathbf{2}} \rightarrow R$ of arity $\mathbf{2}$ will be defined on the graph of the order of R and not on all of $R \times R$.

2. We have worked in the category DCPO of directed complete partially ordered sets and Scott-continuous functions. The results apply in particular to the subcategory SET of sets. One can use the same arguments in other Cartesian closed categories as, for example, the category of qcb-spaces (quotients of countably based topological spaces [1]) and the category POSET of partially ordered sets and order preserving functions. The relaxed setting will need poset enriched categories, of course.

3. I. Hasuo [5] deals with predicate transformers for monads enriched with order in a very general way. The results in his main examples on weakest precondition semantics for two player games [5, Sections 4 and 6] can be recovered by our methods if one transfers them to the category POSET which of course contains the category SET.

4. We can apply our methods also in situations where the ambient category is no longer Cartesian closed. Since we are working with exponentials of a fixed object R and certain subobjects thereof, we have to ensure that these exponentials exist and yield a model of simply typed λ -calculus. For this, an appropriate setting is provided by Hofmann and Streicher [6]. A category \mathbf{C} they call *category with continuations* if \mathbf{C} has finite products and a subclass \mathbf{T} of objects with a distinguished object $R \in \mathbf{T}$ of *responses* such that every $A \in \mathbf{T}$ has an exponential $R^A \in \mathbf{T}$, with the property that $R^A \times B \in \mathbf{T}$ for any $B \in \mathbf{T}$. A simple example for this situation is the category of continuous dcpos and Scott-continuous maps provided that R is a continuous lattice. Another example is the category of topological spaces and continuous maps: For \mathbf{T} one may take the class of exponentiable spaces and for R a continuous lattice with the Scott topology.

5. For an equationally defined class of entropic algebras, the monad given by the free algebras is commutative in the sense of A. Kock [17]. Thus, there should be a category theoretical extension our results.

The referees would have liked to see a new striking example, where our methods can be applied. But our Theorems 6.4 and 7.6 clearly indicate quite narrow limitations to the use of predicate transformer semantics. It is quite a rare phenomenon that operations commute or subcommute. One may not be able to go far beyond the known examples. And if the operations do not commute, one has to find a manageable collection of operations that commute with or subcommute with the given ones, a task that I have no idea how to be attacked except for some very simple cases.

References

- [1] I. Battenfeld, M. Schröder and A. Simpson, A convenient category of domains. *Electronic Notes in Theoretical Computer Science* **172** (2007), pp. 69–99.
- [2] G. Birkhoff, *Lattice Theory*. American Mathematical Society Colloquium Publications vol. XXV, 3rd edition, (1967).
- [3] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove and D. S. Scott, *Continuous Lattices and Domains*. Encyclopedia of Mathematics and its Applications, Vol. 93, Cambridge University Press, 2003.
- [4] J. Goubault-Larrecq, Prevision Domains and Convex Powercones. In: FoSSaCS'08, *Lecture Notes in Computer Science* **4962** (2008), pp. 318–333. Springer-Verlag.
- [5] I. Hasuo, Generic weakest precondition semantics from monads enriched with order. *Theoretical Computer Science* (2015) (In Press) <http://dx.doi.org/10.1016/j.tcs.2015.03.047>
- [6] M. Hofmann and Th. Streicher, Completeness of continuation models for λ_μ -calculus. *Information and Computation* **179** (2002), pp. 332–355.
- [7] E. D'Hondt and P. Panangaden, Quantum weakest preconditions. *Mathematical Structures in Computer Science* **16**(3) (2006), pp. 429–451.
- [8] B. Jacobs, New directions in categorical logic, for classical, probabilistic and quantum logic. *Logical Methods in Computer Science* (to appear), arXiv:1205.3940v3 (2014).
- [9] C. Jones, *Probabilistic non-determinism*. Ph.D. Thesis, University of Edinburgh, Report ECS-LFCS-90-105, 1990.
- [10] C. Jones and G. D. Plotkin, A probabilistic powerdomain of evaluations. *Proc. LICS '89*, pp. 186–195, IEEE Press, 1989.
- [11] K. Keimel, On the equivalence of state transformer semantics and predicate transformer semantics. Proceedings of the Workshop Informatics and Information Technologies in Education: Theory, Practice, Didactics, Novosibirsk, vol. 1 (2012), pp. 78–104. (See also arXiv:1410.7930.)
- [12] K. Keimel and J. D. Lawson, Extending algebraic operations to D-completions. *Theoretical Computer Science* **430** (2012), pp. 73–87.
- [13] K. Keimel and G. D. Plotkin, Predicate transformers for extended probability and non-determinism. *Mathematical Structures in Computer Science* **19** (2009), pp. 501–539.
- [14] K. Keimel and G. D. Plotkin, Mixed powerdomains for probability and nondeterminism. Submitted.
- [15] K. Keimel, A. Rosenbusch and Th. Streicher, A Minkowski type duality mediating between state and predicate transformer semantics for a probabilistic nondeterministic language. *Annals of Pure and Applied Logic* **159** (2009), pp. 307–317.
- [16] K. Keimel, A. Rosenbusch and Th. Streicher, Relating direct and predicate transformer partial correctness semantics for an imperative probabilistic-nondeterministic language. *Theoretical Computer Science* **412** (2011), pp. 2701–2713.
- [17] A. Kock, Commutative monads as a theory of distributions. *Theory Appl. Categ.* **26**(4) (2012), pp. 97–131.
- [18] D. Kozen, Semantics of probabilistic programs. *Journal of Computation and System Sciences*, **22** (1981), pp. 328–350.
- [19] J. Lambek and P. J. Scott, *Introduction to higher order categorical logic*. Cambridge Studies in Advanced Mathematics **7**, Cambridge University Press, 1986.
- [20] S. Mac Lane, *Categories for the Working Mathematician*. Graduate Texts in Mathematics, 2nd ed., Springer Verlag, 1998.
- [21] A. McIver and C. Morgan, *Abstraction, Refinement and Proof for Probabilistic Systems*. Springer Verlag, 2005.
- [22] E. Moggi, (1991) Notions of computation and monads. *Information and Computation* **93**(1) (1991), pp. 55–92.
- [23] A. B. Romanowska and J. D. H. Smith, *Modes*. World Scientific, 2002.

- [24] M. Schröder and A. Simpson, Probabilistic observations and valuations (extended abstract). *Electronic Notes in Theoretical Computer Science* **155** (2006), pp. 605–615.
- [25] M. B. Smyth, Power domains and predicate transformers: A topological point of view. In: *Automata, Languages and Programming*, Lecture Notes in Computer Science **154** (1983), pp. 662–675.
- [26] R. Tix, K. Keimel and G. Plotkin, Semantic domains for combining probability and non-determinism. *Electronic Notes in Theoretical Computer Science* **222** (2009), pp. 1–99.