Semantics of Binary Choice Constructs

Dr. Reinhold Heckmann

FB 14 – Informatik, Prof. Wilhelm
Universität des Saarlandes, Postfach 151150
D-66041 Saarbrücken, Germany

e-mail: heckmann@cs.uni-sb.de

Abstract

This paper is a summary of the following six publications:
(1) Stable Power Domains [Hec94d]
(2) Product Operations in Strong Monads [Hec93b]
(3) Power Domains Supporting Recursion and Failure [Hec92]
(4) Lower Bag Domains [Hec94a]
(5) Probabilistic Domains [Hec94b]
(6) Probabilistic Power Domains, Information Systems, and Locales [Hec94c]
After a general introduction in Section 0, the main results of these six publications are summarized in Sections 1 through 6.

0 Introduction

In this section, we provide a common framework for the summarized papers. In Subsection 0.1, Moggi's approach to specify denotational semantics by means of strong monads is introduced. In Subsection 0.2, we specialize this approach to languages with a binary choice construct. Strong monads can be obtained in at least two ways: as free constructions w.r.t. algebraic theories (Subsection 0.3), and by using second order functions (Subsection 0.4). Finally, formal definitions of those concepts which are used in all or many of the summarized papers are collected in Subsection 0.5.

0.1 Values and Computations

Assume a typed functional language with certain non-functional features such as states and assignments, or non-deterministic choice. In denotational semantics, every type τ is mapped into a semantic domain τ^* of *values* of type τ. In a call-by-value language, every variable of type τ is bound to a value of type τ, i.e., a member of the semantic domain τ^*. The semantics of an *expression* e of type τ, however, is not just a value of type τ. In a language with states and assignments, for instance, the meaning of e is a function which, depending on the current state, computes some value along with a new state. In general, a semantic domain $\tau^{\#}$ of *computations* of type τ is derived from τ^*, and the meaning of an expression of type τ is a member of $\tau^{\#}$.

The construction which derives a domain of computations from a domain of values depends on the actual features of the language. This construction will be called M.

Let us first consider a language whose only non-functional concept is recursion. Though every reasonable functional language provides some means for recursive definitions, recursion should be considered as a non-functional concept since it is not contained in simply typed λ-calculus and has to be imported into a functional language by special constants (fixed point combinator) or special syntax (μ-expressions or `letrec`). The evaluation of an expression in a functional language with recursion either terminates and yields a value, or it does not terminate and yields nothing. Thus, $\tau^{\#}$ is τ^* plus a special element $\perp$ which denotes non-termination. Hence, we have $MX = X_\perp$ in this case, where $X_\perp$ is the domain X augmented by $\perp$.

In a language with state and assignment, however, we have $MX = [\mathit{State} \to X \times \mathit{State}]$ because a computation depends on the state and computes a value together with a new state. This only holds for terminating computations; in presence of recursion, we have $MX = [\mathit{State} \to X \times \mathit{State}]_\perp$ instead.

Some expressions, e.g., constants and variables in a call-by-value language, immediately *result* in a value when they are evaluated. To model the semantics of such expressions, functions $\mathsf{r} = \mathsf{r}_X : X \to MX$ are needed for every domain X which may occur as a domain of values. In case of $MX = X_\perp$, r is just the inclusion of X into $X_\perp$. In the state transformer example, $\mathsf{r}x$ yields x without looking at the state or changing it: $\mathsf{r}x = \lambda s.\,(x, s)$.

A function of type $\sigma \to \tau$ in a call-by-value language denotes a semantic function from σ-values to τ-computations, i.e., of semantic type $\sigma^* \to M\tau^*$. When such a function is applied to an argument expression, then a semantic function of type $M\sigma^* \to M\tau^*$ is needed since the semantics of the argument expression is in $M\sigma^*$. To perform this *extension* from values to computations, we need a semantic function $\mathsf{E} : [X \to MY] \to [MX \to MY]$.

The approach to denotational semantics which is sketched in this subsection was developed by Moggi [Mog89, Mog91]. He showed that the denotational semantics of a typed functional language with computations can be schematically written down, once a category of semantic domains and semantic functions, the semantic domains β^* of values of the base types β, and a triple $(M, \mathsf{r}, \mathsf{E})$ as described above are fixed. The category of domains should be *cartesain closed*, i.e., admit the construction of product and function domains with certain properties, and the triple $(M, \mathsf{r}, \mathsf{E})$ should be a *strong monad.*

A strong monad in a cartesian closed category $\mathcal{C}$ of domains and morphisms is a triple $(M, \mathsf{r}, \mathsf{E})$ where

- M maps domains into domains,
- r is a family of morphisms $\mathsf{r}_X : X \to MX$ for every domain X,
- E is a family of morphisms $\mathsf{E}_{XY} : [X \to MY] \to [MX \to MY]$ for every two domains X and Y

which satisfies certain axioms (which are omitted for the sake of brevity). In this paper, we usually drop the type subscripts of r_X and E_{XY}, and often write M for $(M, \mathsf{r}, \mathsf{E})$.

Defining $\mathsf{M}f = \mathsf{E}(\mathsf{r}_Y \circ f)$ for $f : X \to Y$ yields a functor (M, M); r becomes a natural transformation.

0.2 Languages with Choice

Suppose now our functional language has a binary choice construct $x \mid y$, which informally means evaluate x or evaluate y. In denotational semantics, this choice construct is modeled by a binary operation '$\oplus$' which has to exist in the domains of computations for all types τ. This can be guaranteed by using a strong monad M such that for every domain X, there is a morphism $\oplus : MX \times MX \to MX$.

Monads satisfying this requirement exist in great variety, as will be exhibited soon. The choice of a particular one of them depends on

(1) which informations on the overall behavior of a program with choice should be modeled in the semantic domains, and

(2) which other non-functional concepts exist in the language and have to be built into the monad.

Some possible questions about the overall behavior of a choice program are:

(1) what are some of the possible results of the program?

(2) which properties are shared by all possible results?

(3) how often do particular results appear if all possible traces through the program are considered?

(4) what is the probability of some particular result if every choice is performed by tossing an unbiased coin?

Questions (1) and (2) are dealt with by the lower and upper power domain constructions, respectively; and both together by the Plotkin power construction. So far, these constructions were defined as monads in the category of dcpo's (see Subsection 0.5) or certain subcategories thereof. In [Hec94d], we studied analoga of these constructions in the category of *stable dcpo's* and *stable functions* (see Section 1).

To answer question (3), the monad should construct some kind of multiset or *bag domains*. Such bag domains are studied in [Hec94a] (see Section 4).

When one is interested in probability, the monad should be the *probabilistic power construction* of Jones and Plotkin [Jon90, JP89], or a variant thereof. In [Hec94b], we studied the algebraic properties of probabilistic choice (see Section 5), and in [Hec94c], we developed an effective presentation of the probabilistic power construction in terms of Vickers information systems, and a logical description in terms of locales (see Section 6).

Another non-functional concept which often occurs alongside with binary choice is *failure*, i.e., the possibility to terminate without any result. Many authors [Plo76, HP79, Smy78] study languages with choice but no failure, but most process languages [Abr91, page 172] and Prolog [CC84] support failure. In some respect, it is the zero case of choice, which belongs to binary choice as zero belongs to addition, or the empty set belongs to union. The classical power constructions exist in two closely related versions, one with and one without the empty set.

Every reasonable functional language provides some means for recursive definitions. Semantically, recursion can be modeled using the fixed point theorem, provided that the semantic domains are dcpo's with a least element $\perp$ (bottom). A monad M used to describe a language with recursion should thus have the property that every domain MX has a least element.

The Plotkin power domains lack a least element if a failure element (empty set) is included. This problem can be fixed by some modifications in the power domains, which are presented in [Hec92] (see Section 3).

In Moggi's approach to denotational semantics, the semantics of a language is fixed, once a strong monad in a suitable category of 'domains' is found and the semantic domains of values for the base types of the language are fixed. In particular, this kind of standard semantics evaluates pairs, or more generally tuples, in a sequential left-to-right manner. In [Hec93b], we proposed to extend the monad triples $(M, \mathsf{r}, \mathsf{E})$ to quadruples $(M, \mathsf{r}, \mathsf{E}, \bar{\times})$, where $\bar{\times} : MX \times MY \to M(X \times Y)$ specifies how a pair of computations (of the components of a pair expression) is combined into a computation of the pair. We studied which properties such a product should have, and which products exist for a given monad $(M, \mathsf{r}, \mathsf{E})$ (see Section 2).

0.3 Algebraic Theories and Free Constructions

In this subsection and the next one, we present two ways to obtain strong monads. One way is to set up a suitable algebraic theory and then take the free construction w.r.t. this theory as the strong monad. This approach has a long-standing tradition. It was first suggested in [HP79], then extended in [Mai85], studied more closely in [Hoo87] and later [Hec90, Hec91], and is used in several of the summarized papers. The most complete description is the one in [JA94].

Usually, an algebraic theory consists of a set of operators with given arity and a set of axioms in the form of equations $L = R$ over these operators. Since we want to consider algebraic theories in $\mathcal{DCPO}$ we also allow inequations $L \sqsubseteq R$ as axioms. Also, we allow arities of the form $D \times X^n \to X$, where D is a fixed domain, and X is the algebra. In $\mathcal{SET}$, this would give no further generalization, since an operation $f : D \times X^n \to X$ may be replaced by a family $f_d : X^n \to X$ with d in D. In $\mathcal{DCPO}$, this reduction is not possible, because $f : D \times X^n \to X$ stipulates continuity of f in its first argument.

To obtain a strong monad with choice, the algebraic theory should contain a binary operator '$\oplus$' in its signature. The axioms of the theory are chosen as to reflect the properties of the choice construct w.r.t. the informations we want to model. If we are only interested in the set of possible results, then choice is commutative, associative, and idempotent. If we are interested in multiplicities as well, then idempotence has to be dropped. Probabilistic choice (by tossing an unbiased coin) is commutative and idempotent, but not associative; in $A \oplus (B \oplus C)$, A is chosen with probability $1/2$, whereas in $(A \oplus B) \oplus C$, it merely has probability $1/4$. (For more properties of probabilistic choice, see Subsection 2.2 and Section 5).

If failure and/or recursion should be included in the semantic model, then constants $\diamond$ for failure and $\perp$ for bottom can be added to the signature, with the axiom $\perp \sqsubseteq x$ to guarantee that $\perp$ will be the least element, and appropriate axioms for $\diamond$.

Usually, models of a theory T (or T-algebras) are sets together with operations modeling the operators of T and satisfying the axioms of T. Here, we consider dcpo models, where the carrier is a dcpo and the operations have to be continuous. A T-algebra homomorphism, or shortly *T-morphism*, is a *continuous* function between T-algebras preserving all operations in the signature of T.

Definition 0.1 (Free constructions) A *free construction* $(\mathcal{T}, \mathsf{r})$ for an algebraic theory T maps every dcpo X into a T-model $\mathcal{T}X$ with a morphism $\mathsf{r} : X \to \mathcal{T}X$, such that for every T-model A and every morphism $f : X \to A$, there is a unique T-morphism $\overline{f} : \mathcal{T}X \to A$ with $\overline{f} \circ \mathsf{r} = f$.

Categorically speaking, free constructions for T are left adjoint to the forgetful functor from the category of T-models and T-morphisms to $\mathcal{DCPO}$.

Theorem 0.2 In $\mathcal{DCPO}$, free constructions exist for all algebraic theories. □

A free construction $(\mathcal{T}, \mathsf{r})$ for theory T implies in a fairly obvious manner a strong monad $(\mathcal{T}, \mathsf{r}, \mathsf{E})$ where by construction every $\mathcal{T}X$ is a T-algebra.

0.4 Second Order Functions

The second approach to obtain strong monads is less common. It works in arbitrary cartesian closed categories. Originally, it was used to obtain a strong monad for functional languages with explicit continuations. In [Hec90, Hec93a] the approach was chosen to characterize strong monads with binary choice (power domain constructions). A similar, but different approach was chosen in [Jon90, JP89] to define the probabilistic power domains. The idea is to fix a dcpo R and to define a strong monad as follows:

$$\begin{aligned} MX &= [[X \to R] \to R] \\ \mathsf{r}x &= \lambda p.\, p\, x \\ \mathsf{E} f A &= \lambda p.\, A(\lambda a.\, f a p) \end{aligned}$$

Often, it is useful not to consider the full function space MX, but only a certain subspace consisting of functions satisfying special properties.

To obtain domains MX with a choice operation '$\oplus$', take a domain R with some binary operation '$\oplus$'. The domains MX inherit '$\oplus$' from R by defining $A \oplus B = \lambda p.\, Ap \oplus Bp$. Moreover, they inherit the basic algebraic properties of '$\oplus$'; if '$\oplus$' is commutative / associative / idempotent in R, then it is so in MX for all X.

The two approaches have different merits. The first yields domains which are not explicitly given, but implicitly described by the universal property to be free algebras. In [Hec93b] and [Hec92], such universal properties are exploited without caring about explicit descriptions (see Sections 2 and 3). The second approach yields explicitly given domains without a powerful universal property. Often, the quite complex description in terms of second order functions can be simplified so far that a universal property of MX can be proved at least in the case of continuous domains X, i.e., one can prove that the two approaches actually define the same thing. In [Hec90, Hec93a], this exercise was performed for the classical power constructions. The treatment of the probabilistic power construction in [Jon90] is similar, but starts from a different representation. In [Hec94d], the stable power domains are handled, and in [Hec94a] the bag domains (see Sections 1 and 4).

0.5 Dcpo's and Topological Spaces

In this subsection, we introduce those concepts which occur in all or many of the summarized papers.

We use the standard definitions: a poset $(X, \sqsubseteq)$ is a set X together with a reflexive, anti-symmetric, and transitive relation '$\sqsubseteq$'. For a subset A of a poset X, we define $\downarrow A = \{x \in X \mid \exists a \in A : x \sqsubseteq a\}$ and $\uparrow A = \{x \in X \mid \exists a \in A : a \sqsubseteq x\}$. We shall often abbreviate $\downarrow\{a\}$ by $\downarrow a$ and $\uparrow\{a\}$ by $\uparrow a$. The set A is *lower* if $A = \downarrow A$, and *upper* if $A = \uparrow A$. The *least upper bound* or *join* of A is denoted by $\bigsqcup A$ (if it exists).

A subset D of a poset is *directed* if it is non-empty, and for all x, y in D, there is z in D with $x, y \sqsubseteq z$. A *dcpo* is a poset where every directed set has a least upper bound. We sometimes call dcpo's *domains*, especially if they are continuous (see below). A function $f : \mathbf{X} \to \mathbf{Y}$ between dcpo's $\mathbf{X}$ and $\mathbf{Y}$ is *continuous* if for all directed subsets D of $\mathbf{X}$, $f(\bigsqcup D) = \bigsqcup f[D]$ holds. The category of dcpo's and continuous functions is called $\mathcal{DCPO}$. This category is small complete and cartesian closed. The function space $[\mathbf{X} \to \mathbf{Y}]$ for instance consists of the continuous functions from $\mathbf{X}$ to $\mathbf{Y}$ ordered by $f \sqsubseteq g$ iff $fx \sqsubseteq gx$ for all x in $\mathbf{X}$.

A *topological space* is a set X together with a set ΩX of subsets of X which is closed under finite intersections and arbitrary unions. The sets in ΩX are called *open*, and their complements are called *closed*. A function $f : X \to Y$ between two topological spaces X and Y is *continuous* iff $f^{-1}[V]$ is open in X for every open set V of Y.

A subset O of a dcpo $\mathbf{X}$ is *Scott open* (or briefly *open*) if it is upper, and for all directed sets D, $\bigsqcup D \in O$ implies $D \cap O \neq \emptyset$. With this definition, every dcpo becomes a topological space. A function $f : \mathbf{X} \to \mathbf{Y}$ between dcpo's is dcpo continuous iff it is topologically continuous. Thus, $\mathcal{DCPO}$ can be considered as a full subcategory of the category of topological spaces.

Every topological space can be preordered by defining $x \sqsubseteq x'$ iff every open set which contains x also contains x'. This is called the *specialization preorder* of the space. Every continuous function is in particular monotonic. A space is a $\mathcal{T}_0$-*space* iff this preorder is a partial order, i.e., anti-symmetric. The specialization preorder of a dcpo with its Scott topology is the original order of the dcpo. Hence, every dcpo is a $\mathcal{T}_0$-space.

For two elements x and y of a dcpo $\mathbf{X}$, x is *way-below* y ($x \ll y$) iff for all directed $D \subseteq \mathbf{X}$, $y \sqsubseteq \bigsqcup D$ implies $x \in \downarrow D$. The dcpo $\mathbf{X}$ is *continuous* iff for all x in $\mathbf{X}$, the set $\{a \in \mathbf{X} \mid a \ll x\}$ is directed with join x. A subset B of $\mathbf{X}$ is a *basis* of $\mathbf{X}$ iff for all x in $\mathbf{X}$, the set $\{b \in B \mid b \ll x\}$ is directed with join x. By definition, every continuous domain $\mathbf{X}$ has a basis, namely $\mathbf{X}$ itself. In general, there are many other bases. $\mathbf{X}$ is ω-*continuous* iff it has a countable basis. In a continuous domain $\mathbf{X}$, the set $\Uparrow A = \{x \in \mathbf{X} \mid \exists a \in A : a \ll x\}$ is Scott open for every $A \subseteq \mathbf{X}$.

An element a of a dcpo is *compact* iff $a \ll a$. The set of compact elements of $\mathbf{X}$ is denoted by $\mathsf{K}\mathbf{X}$. A continuous domain $\mathbf{X}$ is *algebraic*, iff there is a basis $B \subseteq \mathsf{K}\mathbf{X}$ of $\mathbf{X}$, iff $\mathsf{K}\mathbf{X}$ itself is a basis of $\mathbf{X}$. We call this basis the *canonical* basis of $\mathbf{X}$.

1 Stable Power Domains [Hec94d]

Classically, all power domain constructions were defined in the category of dcpo's and continuous functions. This was still the case in the thesis [Hec90] and the papers derived from it. Also, all the publications summarized in this paper deal with dcpo's — except for the one summarized in this section.

The subsections of this section correspond to the sections of the summarized paper.

1.1 Introduction

In [Ber78], Berry introduced the notion of *stability* to ban certain parallel functions like the 'parallel or' from the semantic domains of sequential languages. Although stability does not exclude all parallel functions, it can be understood as an approximation to the — not yet semantically describable — notion of sequentiality.

Berry introduced a special kind of domains — the *dI domains* — as the objects of his category of stable mappings. Like the classical category of Scott domains and continuous functions, the category of dI domains suffers from the fact that it is not small complete: equalizers of parallel pairs of morphisms do not exist in general. It is well known that the category of Scott domains can be embedded into the much larger category of all dcpo's that is both small complete and cartesian closed. In [Ama91], Amadio presented a category of *stable dcpo's* and 'stable' mappings that is small complete and cartesian closed, and contains the dI domains as a full subcategory.

In the previous paragraph, we put the word stable into quotes, because the stable mappings of Amadio are called *conditionally multiplicative* (*cm*) by Berry. Being *cm* is an approximation of the mathematically complex notion of stability in [Ber78]. Whereas on dI domains, stability and *cm* are equivalent notions, this is not true for more general kinds of domains. Nevertheless, we shall adopt Amadio's habit to call the *cm* functions stable.

1.2 Scpo's and Stable Functions

Two points a and b of a poset are *compatible*, $a \uparrow b$, if they have a common upper bound. Stability is the requirement to respect meets of pairs of compatible points.

Definition 1.1 A dcpo X is *stable* iff every pair of compatible points $a \uparrow b$ of X has a meet $a \sqcap b$, and the meet operation is continuous for compatible points: if $D \subseteq X$ is directed and $x \uparrow \bigsqcup D$, then $x \sqcap \bigsqcup D = \bigsqcup_{d \in D}\ x \sqcap d$. We abbreviate the wording stable dcpo to *scpo*.

A function $f : X \to Y$ between two scpo's is *stable* iff it is continuous and respects compatible meets: $a \uparrow b$ implies $f(a \sqcap b) = fa \sqcap fb$. The category of scpo's and stable functions is called $\mathcal{SCPO}$.

Theorem 1.2
The category $\mathcal{SCPO}$ is small complete and cartesian closed. The exponential $[X \to Y]$ consists of the stable functions from X to Y ordered by the *stable order*:

$$f \sqsubseteq g \ \text{ iff } \ \forall x, x' \in X : (x \sqsubseteq x' \text{ implies } fx = fx' \sqcap gx)$$

The stable order implies the pointwise order: $f \sqsubseteq g$ implies $fx \sqsubseteq gx$ for all x in X. □

Both scpo's and stable maps are defined using the compatibility relation '$\uparrow$'. However, this relation has the drawback that it is not 'continuous' in general, i.e., it is not closed under directed joins. This problem can be overcome by introducing *weak compatibility.* With $\Delta = \Delta_X = \{(x, x) \mid x \in X\}$, it is easily seen that $u \uparrow v$ iff $(u, v) \in {\downarrow}\Delta$. This leads to the definition that u and v are weakly compatible, $u \overline{\uparrow} v$, iff $(u, v) \in \mathsf{cl}\,\Delta$, where '$\mathsf{cl}$' denotes Scott closure. This relation is closed under directed joins. The definition of stability could be based on weak compatibility as well since meets of weakly compatible points exist in scpo's and are preserved by stable maps.

1.3 Free Constructions on Stable Dcpo's

Following the approach of Subsection 0.3, stable power constructions are introduced as free constructions w.r.t. certain algebraic theories. Of course, we consider here models of these theories in $\mathcal{SCPO}$ instead of $\mathcal{DCPO}$. This means that the carriers of the algebras must be scpo's, and the operations and homomorphisms have to be stable.

Theorem 1.3 In $\mathcal{SCPO}$, free constructions exist for all algebraic theories. □

Free constructions in $\mathcal{SCPO}$ have similar properties as those in $\mathcal{DCPO}$. In the formulations of these properties, however, additional hypotheses appear which are tautologies in $\mathcal{DCPO}$ and hence not visible there. One such property is separability.

Definition 1.4 An scpo X is called *separable* iff for all points a and b of X with $a \not\sqsubseteq b$, there is a stable map $\sigma : X \to \mathbf{2} = \{\bot \sqsubset \top\}$ with $\sigma a \not\sqsubseteq \sigma b$, i.e., $\sigma a = \top$ and $\sigma b = \bot$.

All dcpo's are separable by continuous maps. In contrast, there are separable scpo's as well as non-separable ones. Products and sub-scpo's of separable scpo's are separable; and if Y is separable, then $[X \to Y]$ is separable as well. Also, every continuous scpo is separable. Here, continuity of an scpo means continuity in the usual sense of the underlying dcpo.

An scpo is non-discrete iff there are points a and b such that $a \sqsubset b$, i.e., $a \sqsubseteq b$ and $a \neq b$.

Theorem 1.5 Let T be an algebraic theory with free construction $\mathcal{T}$. If there is at least one non-discrete T-algebra, then $\mathsf{r} : X \to \mathcal{T}X$ is an order embedding for all separable scpo's X. Otherwise for all non-discrete X, r is not injective. □

Another such property is exponentiability of theories.

Definition 1.6 An algebraic theory T is *exponentiable* iff for every scpo X and T-algebra A, the function space $[X \to A]$ becomes a T-algebra, if the operations on A are raised to operations on functions.

In $\mathcal{DCPO}$, all theories are exponentiable: since both equality and order of functions are defined pointwise, the validity of the axioms on A implies the validity on $[X \to A]$. In $\mathcal{SCPO}$, the situation becomes more complex because functions are not ordered pointwise. Certain inequational axioms cannot be lifted to the function space, whereas others can be.

Theorem 1.7 Free constructions for exponentiable theories have stable extension. □

In $\mathcal{DCPO}$, the extension E induced by a free construction is always continuous. In $\mathcal{SCPO}$ however, extension is not always stable. Since the definition of a strong monad includes the requirement that E is a morphism of the underlying category, free constructions in $\mathcal{SCPO}$ do not always induce strong monads.

1.4 Power Theories and Free Power Constructions

Stable power constructions are introduced as free constructions w.r.t. certain algebraic theories. We consider only set-like theories, where choice is commutative, associative, and idempotent. (The set of results of a choice program is analyzed, not multiplicities or probabilities). Since the power domains are thought of as consisting of formal sets, we rename the function $\mathsf{r} : X \to MX$ into s (singleton function).

The plain power theory $\mathcal{P}$ has just one binary operation '$\oplus$' that is commutative, associative, and idempotent:

$$\oplus : A \times A \to A \qquad a \oplus b = b \oplus a \qquad a \oplus (b \oplus c) = (a \oplus b) \oplus c \qquad a \oplus a = a$$

$\mathcal{P}$-algebras are also called *semilattices*. The *lower* theory $\mathcal{L}$ has the additional inequational axiom $a \sqsubseteq a \oplus b$, whereas the *upper* theory $\mathcal{U}$ has the dual axiom $a \oplus b \sqsubseteq b$.

There are variants with a failure element 0 with axiom $a \oplus 0 = a$ ($\mathcal{P}^0$, $\mathcal{L}^0$, $\mathcal{U}^0$), variants with a bottom element $\perp$ with axiom $\perp \sqsubseteq a$ ($\mathcal{P}_\perp$ etc.), and variants with both ($\mathcal{P}^0_\perp$ etc.).

The requirement of stability puts severe restrictions on '$\oplus$'.

Lemma 1.8 Let X be an scpo, and $\oplus : X \times X \to X$ a stable operation that is commutative and idempotent. Then $a \mathbin{\overline{\uparrow}} b$ in X implies $a \oplus b = a \sqcap b$. □

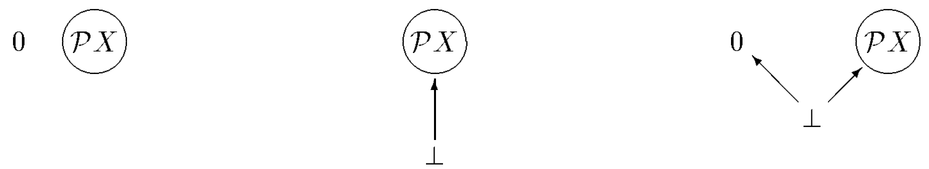
This lemma implies that '$\oplus$' is strict, and that non-empty $\mathcal{L}$-algebras have singleton carriers. Hence, lower power domains $\mathcal{L}X$, $\mathcal{L}^0X$ etc. in $\mathcal{SCPO}$ are degenerated, since they have at most one element, no matter how large X is. This does not happen to the other power domains:

Proposition 1.9 For all separable X, the singleton map $\mathsf{s} : X \to \mathcal{C}X$ is an order embedding, if $\mathcal{C}$ is any of the upper and plain constructions $\mathcal{P}$ through $\mathcal{U}^0_\perp$. □

The plain theories are exponentiable, whereas the upper theories are not. The extensions E are stable for all plain power domains, whereas they are in general not even monotonic for the upper power domains.

Thus, only the plain power constructions are worth for further study. The cases with 0 and/or $\perp$ can be reduced to the basic case:

Theorem 1.10 The power domains $\mathcal{P}^0X$, $\mathcal{P}_\perp X$, and $\mathcal{P}^0_\perp X$ look as follows:



In the classical case, the power domains with 0 and/or $\perp$ also look as depicted above — except for $\mathcal{P}^0_\perp X$ that has many more elements since $a \oplus \perp$ is different from $\perp$ in general. The structure of the classical $\mathcal{P}^0_\perp X$ is analyzed in [Hec92] (see Section 3).

1.5 Stable Topology

In the remainder of [Hec94d], an explicit description of the plain power domains $\mathcal{P}X$ is developed using topological methods.

In $\mathcal{DCPO}$, the function space $[X \to \mathbf{2}]$ with $\mathbf{2} = \{\bot \sqsubset \top\}$ is isomorphic to ΩX, the dcpo of Scott open subsets of X ordered by inclusion. In $\mathcal{SCPO}$, $[X \to \mathbf{2}]$ (with the stable order) is isomorphic to the scpo $\Omega_{\mathrm{s}} X$ of *stable opens* of X ordered by '$\sqsubseteq_{\mathrm{o}}$'.

Here, a subset O of an scpo X is *stable open* iff it is Scott open and closed under compatible meets. For two stable opens U and V, $U \sqsubseteq_{\mathrm{o}} V$ holds iff $U \subseteq V$ and $V \cap {\downarrow} U \subseteq U$.

Finite intersections, $\subseteq$-directed unions, and disjoint unions of stable open sets are stable open. Binary unions of stable open sets are not stable open in general. Thus, the stable open sets do not form a proper topology on X.

Definition 1.11

The *saturated hull* of a subset A of an scpo X is $\mathsf{sat}\, A = \bigcap \{O \mid O \supseteq A,\ O \text{ is stable open}\}$. The set A is *saturated* iff $\mathsf{sat}\, A = A$.

In the classical case, $\mathsf{sat}\, A$ would be identical to ${\uparrow} A$, and saturated sets would be just upper sets. In the stable case however, $\mathsf{sat}\, A$ is in general much larger than ${\uparrow} A$.

1.6 Second Order Predicates

We now use the approach of Subsection 0.4 to define a strong monad in terms of second order functions. After some modifications, it will turn out that it yields the free stable semilattice at least for continuous scpo's.

Let $\mathcal{P}^2$ be defined as $\mathcal{P}^2 X = [[X \to \mathbf{2}] \to \mathbf{2}]$ where $\mathbf{2} = \{\bot \sqsubset \top\}$. This defines a strong monad in $\mathcal{SCPO}$. The scpo's $\mathcal{P}^2 X$ are separable stable semilattices if '$\oplus$' is defined by $P \oplus Q = \lambda p.\, P p \sqcap Q p$.

The full set $\mathcal{P}^2 X$ of second order predicates contains much junk that cannot be reached by the operations s, '$\oplus$', and directed joins. The amount of junk can be reduced by imposing the following conditions on the elements P of $[[X \to \mathbf{2}] \to \mathbf{2}]$:

(1) $P\,(p \sqcap q) = P p \sqcap P q$

(2) $P\,(\lambda x.\, \top) = \top$

(3) $P\,(\lambda x.\, \bot) = \bot$

(4) If $(p_i)_{i \in I}$ is directed in the pointwise order, then $P\,(\bigsqcup_{i \in I} p_i) = \bigsqcup_{i \in I} (P\, p_i)$

Let $\mathcal{P}_{\mathrm{f}} X$ be the scpo of all elements of $\mathcal{P}^2 X$ which satisfy the four conditions above. Then $\mathcal{P}_{\mathrm{f}}$ is still a strong monad which produces separable stable semilattices.

An alternative isomorphic description of $\mathcal{P}^2$ is possible in terms of stable open sets by $\mathcal{P}^2 X \cong \Omega_{\mathrm{s}}(\Omega_{\mathrm{s}} X)$. By translating the four conditions into the language of stable open sets, one obtains a description of $\mathcal{P}_{\mathrm{f}} X$ as a certain sub-scpo of $\Omega_{\mathrm{s}}(\Omega_{\mathrm{s}} X)$. This description of $\mathcal{P}_{\mathrm{f}} X$ is called $\mathcal{P}_\Phi X$. It is still of second order because it uses certain sets of stable open sets.

We also looked for a first order description. For subsets A of an scpo X, let φA be the set of all stable open supersets of A. This set φA is a member of $\mathcal{P}_\Phi X$ iff A is *stably compact* and not empty.

Definition 1.12

A subset A of an scpo X is *stably compact* iff for all $\subseteq$-directed families $(O_i)_{i \in I}$ of stable open sets with $A \subseteq \bigcup_{i \in I} O_i$, there is some k in I with $A \subseteq O_k$.

For every scpo X, let $\mathcal{P}_k X$ be the set of all non-empty saturated stably compact subsets of X ordered by $A \sqsubseteq_k B$ iff $A \supseteq B$ and $A \subseteq U \Rightarrow A \subseteq \uparrow(U \cap \downarrow B)$ for all stable opens U.

The restriction to saturated sets is needed for injectivity of φ; $\varphi A = \varphi B$ holds iff $\mathsf{sat}\, A = \mathsf{sat}\, B$. The seemingly strange 'compact order' $\sqsubseteq_k$ was chosen so that φ becomes an order embedding.

Proposition 1.13 For every scpo X, the mapping $\varphi : \mathcal{P}_k X \to \mathcal{P}_\Phi X$ is an order embedding: $A \sqsubseteq_k B$ iff $\varphi A \sqsubseteq_o \varphi B$. □

1.7 The Continuous Case

If X is a continuous scpo (which means that the underlying dcpo is continuous in the usual sense), then the free stable semilattice $\mathcal{P}X$, the first order description $\mathcal{P}_k X$, and the second order descriptions $\mathcal{P}_\Phi X$ / $\mathcal{P}_f X$ are isomorphic and continuous again.

The proof of this main theorem requires a number of technical lemmas. They are used to prove $\mathcal{P}_k X \cong \mathcal{P}_\Phi X \cong \mathcal{P}_f X$ for continuous scpo's X. Also, a saturated subset of a continuous scpo is stably compact iff it is Scott compact.

To prove continuity and freeness of $\mathcal{P}_k X$, yet another explicit description is developed. This description is in terms of finite strongly independent sets.

Definition 1.14 A subset A of an scpo is *strongly independent* iff its points are pairwise weakly incompatible, i.e., if u, v in A, then $u \mathrel{\overline{\uparrow}} v$ implies $u = v$.

Finite strongly independent sets have a host of interesting properties:

Proposition 1.15 Let X be a continuous scpo.

(1) For every finite strongly independent subset A of X, $\mathsf{sat}\, A$ equals $\uparrow A$.

(2) For every non-empty finite set F, there is a *unique* non-empty finite strongly independent set A with $\mathsf{sat}\, F = \mathsf{sat}\, A$.

(3) The mapping $A \mapsto \mathsf{sat}\, A$ is an order embedding from the set of non-empty finite strongly independent subsets of X with the Egli-Milner ordering to $\mathcal{P}_k X$, the set of non-empty saturated stably compact subsets of X ordered by '$\sqsubseteq_k$'. □

If X is a continuous scpo with basis $\mathcal{B}$, let $\mathcal{B}^*$ be the set of all $\uparrow E$ where E is a non-empty finite strongly independent subset of $\mathcal{B}$.

Theorem 1.16 If X is an (ω-)continuous scpo with basis $\mathcal{B}$, then $\mathcal{P}_k X$ is an (ω-)continuous scpo with basis $\mathcal{B}^*$.

If $\mathcal{B}$ is closed under weakly compatible meets, then the basis $\mathcal{B}^*$ of $\mathcal{P}_k X$ is closed under addition '$\oplus$'. □

Knowing bases of $\mathcal{P}_k X$, one can prove:

Theorem 1.17 Let X be a continuous scpo.

(1) $\mathcal{P}_k X$ cannot be restricted further: all elements can be reached from singletons by finite addition and directed join.

(2) $\mathcal{P}_k X$ is the free stable semilattice over X. □

1.8 The Algebraic Case

Theorem 1.18 If X is an (ω-)algebraic scpo, then $\mathcal{P}_k X$ is (ω-)algebraic with canonical basis $(\mathsf{K}X)^*$ (which is the collection of all sets $\uparrow E$ where E is a non-empty finite independent subset of $\mathsf{K}X$). $\square$

There is another isomorphic description of the canonical basis:

Theorem 1.19 Let X be an algebraic scpo. The canonical basis of $\mathcal{P}_k X$ is isomorphic to the poset of all finite non-empty independent subsets of the canonical basis of X, ordered by Egli-Milner. $\square$

Surprisingly, our stable power domains turn out to be identical with the so-called *lossless power domains* of [Bun88, Jag89], which were proposed to model relations in data bases. So far, lossless power domains were never related with stable functions, nor were they shown to be semilattices.

Definition 1.20 An scpo X is *stably algebraic* iff it is algebraic and its canonical basis is closed w.r.t. compatible meets.

Theorem 1.21 If X is stably algebraic, then so is $\mathcal{P}X$, and its canonical basis is closed w.r.t. addition. $\square$

1.9 Domain-theoretic Properties of $\mathcal{P}$

Knowing several explicit descriptions of $\mathcal{P}X$, we can check whether various subclasses of $\mathcal{SCPO}$ are closed under $\mathcal{P}$.

(1) $\mathcal{P}$ preserves continuity, algebraicity, and stable algebraicity, and also the ω-versions of these classes.

(2) If X is an algebraic scpo with property I, then so is $\mathcal{P}X$.

(3) If X is finite, then so is $\mathcal{P}X$. It consists of the non-empty independent subsets of X.

(4) If S is a discrete scpo, then so is $\mathcal{P}S$. In this case, $\mathcal{P}S$ consists of the finite non-empty subsets of S, and $\mathsf{s}x = \{x\}$ and $A \oplus B = A \cup B$ hold.

(5) If $S_\perp$ is a flat scpo, then so is $\mathcal{P}(S_\perp)$, and $\mathcal{P}(S_\perp) \cong (\mathcal{P}S)_\perp$ holds.

(6) If X is an scpo with least element $\perp$, then $\mathcal{P}X$ has least element $\mathsf{s}\perp$.

(7) If X is an algebraic scpo with property L, then so is $\mathcal{P}X$.

(8) $\mathcal{P}$ does not preserve bounded completeness, nor the properties dL or dI.

(9) If X is stable (ω-)bifinite [Ama91], then so is $\mathcal{P}X$.

2 Product Operations in Strong Monads [Hec93b]

If a strong monad M is used to define the denotational semantics of a functional language with computations, a product operation $\bar{\times} : MX \times MY \to M(X \times Y)$ is needed to define the semantics of pairing. Every strong monad is equipped with two standard products, which correspond to left-to-right and right-to-left evaluation. In [Hec93b], we studied the algebraic properties of these standard products in general. Then we defined alternative products with similar properties for strict and parallel evaluation in the special case of strong monads in $\mathcal{DCPO}$ which are obtained as free constructions w.r.t. various theories of non-deterministic computation, including both classical and probabilistic theories.

The subsections of this section do *not* correspond to the sections of the summarized paper.

2.1 The Two Standard Products of a Strong Monad

After putting it into our notation, Moggi's choice for '$\bar{\times}$' becomes

$$A \overrightarrow{\times} B \;=\; \mathsf{E}(\lambda a.\, \mathsf{E}(\lambda b.\, \mathsf{r}(a,b))\, B)\, A,$$

which corresponds to a left-to-right evaluation of pairs. Equally well, one could define

$$A \overleftarrow{\times} B \;=\; \mathsf{E}(\lambda b.\, \mathsf{E}(\lambda a.\, \mathsf{r}(a,b))\, A)\, B,$$

which corresponds to right-to-left evaluation. Strong monads with $\overrightarrow{\times} = \overleftarrow{\times}$ are *symmetric*.

Theorem 2.1

With every strong monad in a CCC, for both $\bar{\times} = \overrightarrow{\times}$ and $\bar{\times} = \overleftarrow{\times}$, the following holds:

(P1) For all a in X and B in MY, $\mathsf{r}a \,\bar{\times}\, B = \mathsf{M}\sigma_a B$, where $\sigma_a b = (a,b)$.
For all A in MX and b in Y, $A \,\bar{\times}\, \mathsf{r}b = \mathsf{M}\sigma^b A$, where $\sigma^b a = (a,b)$.

(P2) (Naturality) For all A in MX, B in MY, $f : X \to X'$, and $g : Y \to Y'$:
$\mathsf{M}fA \,\bar{\times}\, \mathsf{M}gB = \mathsf{M}(f \times g)(A \,\bar{\times}\, B)$, where $(f \times g)(a,b) = (fa, gb)$.

(P3) (Associativity) For A in MX, B in MY, and C in MZ:
$A \,\bar{\times}\, (B \,\bar{\times}\, C) = \mathsf{M}\alpha\, ((A \,\bar{\times}\, B) \,\bar{\times}\, C)$, where $\alpha((a,b),c) = (a,(b,c))$.

If the strong monad is symmetric, then $\bar{\times} = \overrightarrow{\times} = \overleftarrow{\times}$ additionally satisfies

(P4) (Commutativity) For A in MX and B in MY:
$B \,\bar{\times}\, A = \mathsf{M}\gamma\, (A \,\bar{\times}\, B)$, where $\gamma(a,b) = (b,a)$. □

If we want to augment a given strong monad $(M, \mathsf{r}, \mathsf{E})$ with a product operation '$\bar{\times}$' of our choice, then '$\bar{\times}$' should at least satisfy (P1) through (P3) to be as good as the standard products.

2.2 Power Domain Constructions from Non-Deterministic Theories

We now assume that the functional language with computations, whose semantics shall be described by a strong monad, is a non-deterministic language with a binary choice operator $\mathsf{|}$, a special expression `fail`, and syntactic means for recursion.

Using the method of obtaining strong monads from free constructions (Section 0.3), we set up several algebraic theories with a binary operator $\oplus$ for choice and two constants $\diamond$ for failure and $\perp$ for non-termination, containing at least the axiom $\perp \sqsubseteq x$. Such theories are called *non-deterministic theories* or briefly *ND-theories*. A function f between the models of two possibly different ND-theories is called *additive* if $f(a \oplus b) = fa \oplus fb$, *strict* if $f\perp = \perp$, and $\diamond$-*strict* if $f\diamond = \diamond$.

If we are interested in the *set* of values a non-deterministic expression may produce, then we should give '$\oplus$' the properties of set union: associativity $(A \oplus B) \oplus C = A \oplus (B \oplus C)$, commutativity $A \oplus B = B \oplus A$, and idempotence $A \oplus A = A$. A failing expression produces no values at all; thus, $\diamond$ is the neutral element of '$\oplus$': $A \oplus \diamond = A$. A non-terminating expression produces no values, too, whence $\perp = \diamond$. We call this the *lower set-like theory* LS. It is well-known that the free LS-algebras are the *lower* or *Hoare power domains* [HP79].

If we want to retain the information whether a computation is possibly non-terminating, we have to distinguish between $\perp$ and $\diamond$. This is done simply by dropping the axiom $\perp = \diamond$. The element $\diamond$ is still the neutral element of '$\oplus$', since the expressions `fail | ` A and A may possibly produce the same values and have the same termination behaviour. We call this theory the *set-like theory* S. Its free construction is studied in [Hec92] (see Section 3).

If we want to retain the information, how often a particular value may occur as the result of an expression, then we are interested in the bag or multiset of possible result values, and we have to drop the axiom of idempotence $A \oplus A = A$. According to whether we keep $\perp = \diamond$ or not, the resulting theory is the *lower multiset-like theory* LM or the *multiset-like theory* M. The free construction for LM is studied in [Hec94a] (see Section 4). Although general idempotence does not hold in LM and M, we still have $\diamond \oplus \diamond = \diamond$ by neutrality, and also $\perp \oplus \perp = \perp$, since $\perp \sqsubseteq \perp \oplus \perp \sqsubseteq \diamond \oplus \perp = \perp$.

Quite a different theory results from a probabilistic interpretation of non-determinism, where $A \oplus B$ means that A and B are chosen with equal probability 1/2. Then '$\oplus$' is commutative and idempotent, but not associative, since in $A \oplus (B \oplus C)$, A is chosen with probability 1/2, whereas in $(A \oplus B) \oplus C$, it has probability 1/4. On the other hand, there is still something left of associativity: the *rectangle axiom* $(A \oplus B) \oplus (C \oplus D) = (A \oplus C) \oplus (B \oplus D)$, which holds since all four components are chosen with probability 1/4. The four operands A, B, C, and D may be thought as arranged at the corners of a rectangle; the axiom then says that they may be combined first horizontally, then vertically, as well as first vertically, then horizontally; hence the name. The rectangle law also holds in the theories S and M as a consequence of commutativity and associativity.

There is another, less obvious axiom for the probabilistic interpretation. In a recursive definition such as `x = 1 | x`, the result 1 is chosen with probability 1/2 in the first iteration, with probability 1/4 in the second, and so on. Altogether, it is chosen with probability 1, and the probability of non-termination is 0. Generalizing, this leads to the axiom that the fixed point iteration for $X = A \oplus X$ yields A, for every A. This fixed-point axiom $\mu X.\, A \oplus X = A$ is equivalent to $A \oplus A = A$, which is idempotence, together with the conditional axiom $A \oplus B = B \Rightarrow A \sqsubseteq B$. In [Hec94b], we present alternative descriptions of this theory avoiding conditional axioms (see Section 5).

Under the probabilistic interpretation, $\diamond$ is by no means a neutral element. Thus, we obtain as *probabilistic theory* P the theory with commutativity, idempotence, rectangle axiom, and fixed point axiom. The theory with additionally $\perp = \diamond$ is called the *lower probabilistic theory* LP. It is equivalent to the theory of probabilistic algebras in [Jon90, JP89]. The proof of equivalence is contained in [Hec94b] (see Section 5).

The basic properties that are shared by all theories presented above are the rectangle law, $\diamond \oplus \diamond = \diamond$, and $\perp \oplus \perp = \perp$. ND-theories with these properties will be called *decent*. The significance of this notion is indicated by the following proposition:

Proposition 2.2 Let T and T$'$ be ND-theories such that T$'$ is decent. If X is a T-model and Y a T$'$-model, then the (pointwise defined) sum of additive (strict, $\diamond$-strict) functions from X to Y is additive (strict, $\diamond$-strict) again. The constant functions $\lambda x.\perp$ and $\lambda x.\diamond$ are additive. □

2.3 The Standard Products in Free Constructions

Proposition 2.3 Let T be an ND-theory and $\mathcal{P}$ its free construction.

(1) The left product '$\vec{\times}$' is strict, $\diamond$-strict, and additive in its left argument:
$\perp \vec{\times} B = \perp \qquad \diamond \vec{\times} B = \diamond \qquad (A_1 \oplus A_2) \vec{\times} B = (A_1 \vec{\times} B) \oplus (A_2 \vec{\times} B)$.

(2) The right product '$\overleftarrow{\times}$' is strict, $\diamond$-strict, and additive in its right argument.

(3) If T is decent, then both '$\vec{\times}$' and '$\overleftarrow{\times}$' are additive in both arguments (*biadditive*). □

Next, we consider symmetry.

Proposition 2.4 Let T be an ND-theory with free construction $\mathcal{P}$.

(1) If $\mathcal{P}$ is symmetric, then $\perp = \diamond$ holds in all $\mathcal{P}$-powerdomains.

(2) If T is decent with $\perp = \diamond$, then $\mathcal{P}$ is symmetric. □

Thus, in particular the power constructions $\mathcal{P}_{LS}$, $\mathcal{P}_{LM}$, and $\mathcal{P}_{LP}$ are symmetric, whereas $\mathcal{P}_S$, $\mathcal{P}_M$, and $\mathcal{P}_P$ are not symmetric. To speak precisely, the non-symmetry statement is only proved after it is shown that the axioms of the respective theories do not imply $\perp = \diamond$. This can be done by presenting models with $\perp \neq \diamond$. The dcpo $\{\perp \sqsubset \diamond\}$ is an M- and S-model with '$\oplus$' being join, and a P-model with '$\oplus$' being meet.

2.4 The Choice Product

Now, we try the most obvious non-standard product in a power domain construction: the product $A*B = (A\vec{\times}B)\oplus(A\overleftarrow{\times}B)$, where we choose the evaluation order non-deterministically. To avoid bothering about two different versions of '$*$', we assume that '$\oplus$' is commutative. We shall soon see that idempotence is also important, and finally we assume the rectangle law to make the results of the previous section available. Thus, the following results cover the theories S and P, but not M.

Theorem 2.5 Let T be an ND-theory, where '$\oplus$' is commutative, idempotent, and the rectangular law holds. Then '$*$' has property (P1), is natural (P2), commutative (P4), and biadditive, and the equations $\diamond * \diamond = \diamond$ and $\perp * \perp = \perp$ hold. □

Surprisingly, associativity (P3) does not hold in general. It *does* hold in the set-like theory S, but fails in the probabilistic theory P.

2.5 Decent Products

After the attempt to construct an alternative product by an explicit formula has failed to yield an associative product, we try an axiomatic approach. We call a binary operation $\bar{\times} : \mathcal{P}X \times \mathcal{P}Y \to \mathcal{P}(X \times Y)$ a *decent product* iff it is biadditive, satisfies (P1), $\diamond \bar{\times} \diamond = \diamond$, and $\perp \bar{\times} \perp = \perp$.

Proposition 2.6 Let '$\bar{\times}_1$' and '$\bar{\times}_2$' be two decent products. Then $\bar{\times}_1 \sqsubseteq \bar{\times}_2$ iff $\diamond \,\bar{\times}_1\, \bot \sqsubseteq \diamond \,\bar{\times}_2\, \bot$ and $\bot \,\bar{\times}_1\, \diamond \sqsubseteq \bot \,\bar{\times}_2\, \diamond$. The same holds with '$\sqsubseteq$' replaced by '$=$'. $\square$

Thus, a decent product is uniquely determined by its values on $(\bot, \diamond)$ and $(\diamond, \bot)$. We are particularly interested in the following four possibilities:

- Left product $\times_L$: $\quad \bot \times_L \diamond = \bot \qquad \diamond \times_L \bot = \diamond$
- Right product $\times_R$: $\quad \bot \times_R \diamond = \diamond \qquad \diamond \times_R \bot = \bot$
- Strict product $\check{\times}$: $\quad \bot \,\check{\times}\, \diamond = \bot \qquad \diamond \,\check{\times}\, \bot = \bot$
- Parallel product $\hat{\times}$: $\quad \bot \,\hat{\times}\, \diamond = \diamond \qquad \diamond \,\hat{\times}\, \bot = \diamond$

Proposition 2.7 For a decent theory, the left and right products exist and are given by the standard products '$\overrightarrow{\times}$' and '$\overleftarrow{\times}$'. For the set-like theory S, the strict product exists and is given by the choice product. $\square$

The properties of left and right product are thus given by the properties of the standard products. The properties of strict and parallel product are as follows:

Theorem 2.8 For every ND-theory T, the strict product $\check{\times}$ and the parallel product $\hat{\times}$ — provided they exist — are biadditive, satisfy (P1), are natural (P2), associative (P3), and commutative (P4).

(1) If $\bot \oplus \bot = \bot$ holds in T and '$\check{\times}$' exists, then $A \,\check{\times}\, \bot = \bot \,\check{\times}\, B = \bot$ for all A and B.

(2) If $\diamond \oplus \diamond = \diamond$ holds in T and '$\hat{\times}$' exists, then $A \,\hat{\times}\, \diamond = \diamond \,\hat{\times}\, B = \diamond$ for all A and B. $\square$

2.6 Existence of Parallel Products

Parallel products exist for the ND-theories S, M, and P. This is shown by a universal construction, which in principle works for all decent ND-theories. An analogous construction may be used to show the existence of strict products for these theories.

In the sequel, let T be a decent ND-theory, and $\mathcal{P}$ the corresponding free construction. For dcpo's X and Y, let $Z = [\mathcal{P}Y \overset{\oplus\diamond}{\to} \mathcal{P}(X \times Y)]$ be the dcpo of *additive $\diamond$-strict* morphisms from $\mathcal{P}Y$ to $\mathcal{P}(X \times Y)$. On Z we define '$\oplus$' and $\diamond$ pointwise as $f \oplus g = \lambda B.\, fB \oplus gB$ and $\diamond = \lambda B.\, \diamond$, and additionally $\bot = \lambda B.\, \bot \,\overrightarrow{\times}\, B$.

We have to show that Z is a T-model. We cannot do this in general, because the theory T may contain exotic axioms that do not carry over to the function space. On the other hand, all axioms that occur in this paper do carry over.

Once Z is shown to be a T-model, we introduce the function $p = \lambda a.\, \mathsf{M}\sigma_a$, which maps from X to Z, since $\mathsf{M}\sigma_a = \mathsf{E}(\mathsf{r} \circ \sigma_a)$ is additive and $\diamond$-strict as an extended function. Extending p yields $P : \mathcal{P}X \to Z$, which by uncurrying induces a function $\hat{\times} : \mathcal{P}X \times \mathcal{P}Y \to \mathcal{P}(X \times Y)$. This function is shown to be the parallel product for $\mathcal{P}$.

Summarizing, we have shown the existence of parallel products for the theories S, M, and P. If T is another decent theory, then our method also works if the axioms of T can be proved for the function space Z.

3 Power Domains Supporting Recursion and Failure [Hec92]

In this paper, versions of the classical power domains in $\mathcal{DCPO}$ are studied which not only model non-deterministic choice, but also failure and / or non-termination caused by recursion. These extended power domains are obtained as free dcpo-algebras w.r.t. theories whose signature contains a failure constant and / or an explicit least element besides the choice operator. In the category $\mathcal{SCPO}$ of stable dcpo's, the relationship between the extended power domains and the basic ones is as depicted in Theorem 1.10. In $\mathcal{DCPO}$, the situation is more complex; the extended power domains differ from the basic ones by more additional elements than just the one or two constants.

The subsections of this section do *not* correspond to the sections of the summarized paper.

3.1 The Monads for Recursion and Non-determinism

Non-termination can be modeled by the *lift monad* $\mathcal{B}$. It is the free construction for the $\perp$-theory which has a constant $\perp$ with just one axiom $\perp \sqsubseteq x$ for all x. $\perp$-algebras are dcpo's with least element, and $\perp$-morphisms are strict continuous functions. The lift monad works by adding a new least element $\perp$.

Since we are interested in the set of all results, the algebraic theory for choice plus failure is the theory of commutative idempotent monoids, *CI monoids* for short. Without failure, the closely related theory of *CI semigroups* can be employed which has no neutral element. The free construction $\mathcal{P}^0$ of CI monoids can be built by composing the free construction $\mathcal{P}$ of CI semigroups and the free construction $\mathcal{N}$ from CI semigroups to CI monoids.

Proposition 3.1 The free construction $\mathcal{N}$ from CI semigroups to CI monoids is given by

Elements :	$\mathcal{N}Y = \{\mathsf{n}y \mid y \in Y\} \cup \{0\}$	
Order :	$\mathsf{n}y \sqsubseteq \mathsf{n}y'$ if $y \sqsubseteq y'$	$0 \sqsubseteq 0$
Operations :	$\mathsf{n}a \oplus \mathsf{n}b = \mathsf{n}(a \oplus b)$	$z \oplus 0 = 0 \oplus z = z$

Thus, for all dcpo's X, $\mathcal{P}^0X$ results from $\mathcal{P}X$ by adding a new element 0 incomparable to every other element. □

Non-termination and non-determinism together can be modeled by using the theory of *CI$\perp$-monoids*. A CI$\perp$-monoid is a CI monoid with an additional constant $\perp$ satisfying the axiom $\perp \sqsubseteq x$. In the similar theory of *CI$\perp$-semigroups*, the neutral element 0 of '$\oplus$' is missing. The free construction of CI$\perp$-monoids is called $\mathcal{P}^0_\perp$. In the sequel, more explicit descriptions of $\mathcal{P}^0_\perp$ are presented.

3.2 Decomposition into Three Free Constructions

Theorem 3.2 The construction $\mathcal{P}^0_\perp$ can be decomposed into three free constructions to be applied sequentially:

(1) The free construction $\mathcal{B}$ from the empty theory to the $\perp$-theory (lifting),

(2) The free construction $\mathcal{P}_\perp$ from the $\perp$-theory to the theory of CI$\perp$-semigroups, which coincides with $\mathcal{P}$;

(3) The free construction $\mathcal{N}_\perp$ from CI$\perp$-semigroups to CI$\perp$-monoids, which is given by:

$$
\begin{array}{lll}
\textbf{Elements:} & \mathcal{N}_\perp Y \;=\; \{\mathsf{n}_\perp y \mid y \in Y\} \;\;\cup\;\; \{0\} & \\
\textbf{Order:} & \mathsf{n}_\perp y \sqsubseteq \mathsf{n}_\perp y' \text{ if } y \sqsubseteq y' & \mathsf{n}_\perp \perp \sqsubseteq 0 \qquad 0 \sqsubseteq 0 \\
\textbf{Operations:} & \mathsf{n}_\perp a \oplus \mathsf{n}_\perp b = \mathsf{n}_\perp (a \oplus b) & z \oplus 0 = 0 \oplus z = z
\end{array}
$$
□

The theorem tells that $\mathcal{P}^0_\perp$ is *not* the composition of the lift monad $\mathcal{B}$ and the monad $\mathcal{P}^0 = \mathcal{N} \circ \mathcal{P}$ modeling choice and failure. Instead of $\mathcal{N} \circ \mathcal{P} \circ \mathcal{B}$, it is $\mathcal{N}_\perp \circ \mathcal{P} \circ \mathcal{B}$ where $\mathcal{N}_\perp$ differs from $\mathcal{N}$ by the additional relationship $\mathsf{n}_\perp \perp \sqsubseteq 0$.

Corollary 3.3 For every dcpo X, the power domain $\mathcal{P}^0_\perp X$ results from adjoining a new element 0 to $\mathcal{P}(X_\perp)$ just above its least element. □

3.3 Vertical Decomposition

The decomposition of $\mathcal{P}^0_\perp$ in Theorem 3.2 suggests that the failure element 0 be something special. It seems to be added artificially by the third component of the construction. The power domains can also be decomposed 'vertically' into terminating sets, which can be found in the upper regions of the power domain, and non-terminating sets, which are located in the lower regions. In this decomposition, 0 does not play a special role.

Lower CI monoids are CI monoids with the additional axiom $0 \sqsubseteq x$. The free construction $(\mathcal{L}^0, \mathsf{l}^0)$ of lower CI monoids is the well-known lower or Hoare power domain construction.

The mapping $\mathsf{l}^0 : X \to \mathcal{L}^0 X$ is continuous, and $\mathcal{L}^0 X$ is a CI monoid. Thus, there is a unique continuous monoid homomorphism $\ell : \mathcal{P}^0 X \to \mathcal{L}^0 X$ with $\ell \circ \mathsf{p}^0 = \mathsf{l}^0$. We use this canonical mapping to compare members of $\mathcal{P}^0 X$ and $\mathcal{L}^0 X$ in the decomposition theorem.

The decomposition theorem states that $\mathcal{P}^0_\perp X$ consists of an instance of $\mathcal{P}^0 X$ containing the terminating sets, and an instance of $\mathcal{L}^0 X$ containing the non-terminating sets. Two formal symbols, P and L, are used to label the members of the two components.

Theorem 3.4 For every dcpo X, the power domain $\mathcal{P}^0_\perp X$ is given by

$$
\begin{array}{lll}
\textbf{Elements:} & \mathcal{P}^0_\perp X \;=\; \{\mathsf{L}A \mid A \in \mathcal{L}^0 X\} \;\;\cup & \{\mathsf{P}A \mid A \in \mathcal{P}^0 X\} \\
\textbf{Order:} & \mathsf{L}A \sqsubseteq \mathsf{L}B \text{ iff } A \sqsubseteq B & \mathsf{P}A \sqsubseteq \mathsf{P}B \text{ iff } A \sqsubseteq B \\
 & \mathsf{L}A \sqsubseteq \mathsf{P}B \text{ iff } A \sqsubseteq \ell B & \mathsf{P}A \sqsubseteq \mathsf{L}B \text{ never holds} \\
\textbf{Constants:} & \perp = \mathsf{L}0 & 0 = \mathsf{P}0 \\
\textbf{Addition:} & \mathsf{L}A \oplus \mathsf{L}B = \mathsf{L}(A \oplus B) & \mathsf{P}A \oplus \mathsf{P}B = \mathsf{P}(A \oplus B) \\
 & \mathsf{L}A \oplus \mathsf{P}B = \mathsf{L}(A \oplus \ell B) & \mathsf{P}A \oplus \mathsf{L}B = \mathsf{L}(\ell A \oplus B) \\
\textbf{Generators:} & \mathsf{p}^0_\perp x = \mathsf{P}(\mathsf{p}^0 x) &
\end{array}
$$
□

A dcpo is *discrete* iff its order is equality: $a \sqsubseteq b$ iff $a = b$. It is easy to see that the discrete dcpo's form a full sub-category of $\mathcal{DCPO}$ which is isomorphic to $\mathcal{SET}$.

With Theorem 3.4 and known representations of the power domains $\mathcal{L}^0 X$ and $\mathcal{P}^0 X$ for discrete X, we obtain:

Theorem 3.5 For every discrete dcpo X, the power domain $\mathcal{P}^0_\perp X$ is given by

Elements :	$\mathcal{P}^0_\perp X \;=\; \{A \subseteq X \cup \{\perp\} \mid \perp \in A$ or A is finite$\}$
Order :	Case $\perp \in A$: $\;A \sqsubseteq B$ iff $A \setminus \{\perp\} \subseteq B$
	Case $\perp \notin A$: $\;A \sqsubseteq B$ iff $A = B$
Constants :	$\perp = \{\perp_X\} \qquad 0 = \emptyset$
Addition :	union
Generators :	$\mathsf{p}^0_\perp x = \{x\}$

□

The condition that infinite sets have to contain $\perp$ reflects the fact that processes that produce an infinite number of results cannot terminate after finite time. The order given in the theorem above can also be interpreted intuitively: as long there is $\perp$ in the set A, the corresponding process did not yet terminate and may produce further results; once $\perp$ disappears from A, the process has terminated and the set of results cannot increase further.

3.4 The Basis in the Algebraic Case

Theorem 3.6 If X is algebraic, then $\mathcal{P}^0_\perp X$ is algebraic with the following canonical basis:

Elements :	$\{(\downarrow E,\, \uparrow E) \mid E \subseteq_{fin} \mathsf{K}X\} \;\cup\; \{(\downarrow E,\, \perp) \mid E \subseteq_{fin} \mathsf{K}X\}$
Order :	$(U,\, V) \sqsubseteq (U',\, V')$ iff $U \subseteq U'$ and ($V \supseteq V'$ or $V = \perp$)
Constants :	$0 = (\emptyset,\, \emptyset) \qquad \perp = (\emptyset,\, \perp)$
Addition :	$(U,\, V) \oplus (U,\, V') = (U \cup U',\, V \cup V')$
Generators :	$\mathsf{p}^0_\perp x = (\downarrow\{x\},\, \uparrow\{x\})$

$\perp$ is a new object that is not a set. Union is defined for $\perp$ by $V \cup \perp = \perp \cup V' = \perp$. □

The explicit representation of the canonical basis of $\mathcal{P}^0_\perp X$ allows the derivation of some domain-theoretic properties of $\mathcal{P}^0_\perp$.

Theorem 3.7 $\mathcal{P}^0_\perp$ preserves algebraicity, ω-algebraicity, continuity, finiteness, and bifiniteness. Bounded completeness and the property to be a Scott domain are not preserved. □

4 Lower Bag Domains [Hec94a]

If we are interested in multiplicities of results, we have to drop idempotence of the choice operation. Applying the two methods of Subsections 0.3 and 0.4, we consider the 'initial construction' $\mathcal{L}^*_{\mathrm{i}}$ of free 'lower monoids', and a construction $\mathcal{L}^*_{\mathrm{f}}$ which is defined explicitly in terms of second order functions.

For sober dcpo's $\mathbf{X}$, the elements of $\mathcal{L}^*_{\mathrm{f}}\mathbf{X}$ can be described concretely as bags. For continuous domains $\mathbf{X}$, $\mathcal{L}^*_{\mathrm{f}}\mathbf{X}$ coincides with $\mathcal{L}^*_{\mathrm{i}}\mathbf{X}$. In this case, the domain $\mathcal{L}^*\mathbf{X}$ ($= \mathcal{L}^*_{\mathrm{f}}\mathbf{X} = \mathcal{L}^*_{\mathrm{i}}\mathbf{X}$) is continuous again, and can be described via a basis which is constructed from a basis of $\mathbf{X}$.

The subsections of this section correspond to the sections of the summarized paper, except that the first two sections of the paper are not represented. Hence, the subsections are numbered from 3.

4.3 Lower Monoids

A *lower monoid* is a commutative monoid $(M, +, 0)$ (in the category $\mathcal{DCPO}$) with the additional property $0 \sqsubseteq m$ for all m in M, or equivalently $a \sqsubseteq a + b$ for all a, b in M.

By definition, every lower monoid has a least element, namely 0. A lower monoid M has a largest element, too, since M is directed by $a, b \sqsubseteq a + b$, and directed sets have joins.

Every lower monoid M admits infinite summation. For an indexed family $(m_i)_{i \in I}$ of members of M, we define

$$\sum_{i \in I} m_i = \bigsqcup \{\sum_{i \in F} m_i \mid F \subseteq_{fin} I\}.$$

The join exists since the set is directed because of the property $a \sqsubseteq a+b$. Additive continuous functions preserve such infinite sums: $f(\sum_{i \in I} m_i) = \sum_{i \in I} f(m_i)$.

Every complete lattice L implies a lower monoid $(L, \sqcup, \bot)$. Another example is $(\mathbb{N}_0^\infty, +, 0)$ where $\mathbb{N}_0^\infty = \{0 \sqsubset 1 \sqsubset 2 \sqsubset \cdots \sqsubset \infty\}$ and '+' is the usual addition extended to ∞ by $n + \infty = \infty + n = n$ for all n in $\mathbb{N}_0^\infty$. This is the free lower monoid in $\mathcal{DCPO}$ with one generator, namely 1.

The 'final' lower bag domains $\mathcal{L}_{\mathrm{f}}^* \mathbf{X}$ are given by $\mathcal{L}_{\mathrm{f}}^* \mathbf{X} = [[\mathbf{X} \to \mathbb{N}_0^\infty] \overset{add}{\to} \mathbb{N}_0^\infty]$, i.e., the final lower bag domain over $\mathbf{X}$ consists of the continuous additive functions from the lower monoid $[\mathbf{X} \to \mathbb{N}_0^\infty]$ (with pointwise operations) to the lower monoid $\mathbb{N}_0^\infty$.

Apart from existence and universal property, the general theory does not provide an explicit description of $\mathcal{L}_{\mathrm{i}}^* \mathbf{X}$ except for $\mathcal{L}_{\mathrm{i}}^* \mathbf{1} = \mathbb{N}_0^\infty$. In the summarized paper, we make $\mathcal{L}_{\mathrm{i}}^* \mathbf{X}$ explicit by comparing it with $\mathcal{L}_{\mathrm{f}}^* \mathbf{X}$; we show $\mathcal{L}_{\mathrm{i}}^* \mathbf{X} \cong \mathcal{L}_{\mathrm{f}}^* \mathbf{X}$ for continuous domains $\mathbf{X}$. To this end, we study $\mathcal{L}_{\mathrm{f}}^* \mathbf{X}$ in greater detail in the following sections.

Many properties of $\mathcal{L}_{\mathrm{f}}^* \mathbf{X}$ do not depend on the fact that $\mathbf{X}$ is a dcpo; they hold for more general classes of topological spaces. Thus, we generalize $\mathcal{L}_{\mathrm{f}}^*$ with $\mathcal{L}_{\mathrm{f}}^* X = [[X \to \mathbb{N}_0^\infty] \overset{add}{\to} \mathbb{N}_0^\infty]$ to a construction mapping a topological space X into a dcpo $\mathcal{L}_{\mathrm{f}}^* X$, where $[X \to \mathbb{N}_0^\infty]$ is understood as the dcpo of continuous functions from X to $\mathbb{N}_0^\infty$ ordered pointwise.

4.4 The Final Construction in terms of Modular Functions

Now, we present a simpler description of the final lower bag domains. It is inspired by [Jon90, JP89], where a *probabilistic power construction* $\mathcal{P}$ is defined by $\mathcal{P}X = [\Omega X \overset{mod}{\to} \mathbb{I}]$, the dcpo of continuous modular functions from ΩX, the lattice of open sets of X, to $\mathbb{I}$, the unit interval of the reals with the Scott topology. We define an analogous construction $\mathcal{L}_{\mathrm{m}}^* X = [\Omega X \overset{mod}{\to} \mathbb{N}_0^\infty]$, and show $\mathcal{L}_{\mathrm{m}}^* X \cong \mathcal{L}_{\mathrm{f}}^* X$ for all topological spaces X.

A function μ from ΩX to $\mathbb{N}_0^\infty$ is *modular* if it is strict ($\mu\,\emptyset = 0$) and satisfies the *modular law*

$$\mu(U \cap V) + \mu(U \cup V) = \mu U + \mu V.$$

This property makes μ similar to an (integer-valued) measure, but it is defined on the opens of X and not on the Borel sets. The dcpo $\mathcal{L}_{\mathrm{m}}^* X$ consists of the Scott continuous modular functions from the dcpo ΩX to the dcpo $\mathbb{N}_0^\infty$. In analogy to [Jon90, JP89], we call the members of $\mathcal{L}_{\mathrm{m}}^* X$ *evaluations*. (The exact name would be integer-valued continuous evaluations.)

Obviously, the function $\lambda U.\, 0$ is continuous and modular, and the pointwise sum of two continuous and modular functions is again so. With these operations, $\mathcal{L}_{\mathrm{m}}^* X$ becomes a lower monoid. Additionally, for every x in X, we get an evaluation $\mathsf{s}_\mathrm{m} x = \lambda U.\, `x \in U'$, where '$x \in U$' is 1 if x is in U, and 0 otherwise. The 'singleton' evaluations $\mathsf{s}_\mathrm{m} x$ will be called *point evaluations*. Note that continuity of the function s_m itself is only guaranteed if X is a dcpo, but this does not matter in the following. We can make $\mathcal{L}^*_\mathrm{m}$ into a functor by defining $\mathcal{L}^*_\mathrm{m}\, g\, \mu = \lambda U.\, \mu\,(g^{-1}[U])$.

Theorem 4.1

For every topological space X, the two lower monoids $\mathcal{L}^*_\mathrm{f} X = [[X \to \mathbb{N}_0^\infty] \stackrel{add}{\to} \mathbb{N}_0^\infty]$ and $\mathcal{L}^*_\mathrm{m} X = [\Omega X \stackrel{mod}{\to} \mathbb{N}_0^\infty]$ are isomorphic. The isomorphism φ also respects the 'singleton functions' (i.e., $\varphi \circ \mathsf{s}_\mathrm{f} = \mathsf{s}_\mathrm{m}$) and the functor structure (i.e., $\varphi \circ \mathcal{L}^*_\mathrm{f}\, g = \mathcal{L}^*_\mathrm{m}\, g \circ \varphi$). □

The isomorphism $\varphi : \mathcal{L}^*_\mathrm{f} X \to \mathcal{L}^*_\mathrm{m} X$ is easily defined. For A in $\mathcal{L}^*_\mathrm{f} X$, let $\varphi A = \lambda U.\, A\langle U\rangle$, where $\langle U\rangle : X \to \mathbb{N}_0^\infty$ is defined by $\langle U\rangle x = `x \in U'$.

To define the inverse isomorphism $\psi : \mathcal{L}^*_\mathrm{m} X \to \mathcal{L}^*_\mathrm{f} X$, we have to compute a number $\psi(\mu)(f)$ for every continuous $\mu : \Omega X \stackrel{mod}{\to} \mathbb{N}_0^\infty$ and $f : X \to \mathbb{N}_0^\infty$. This is done by *integration*: $\psi(\mu)(f) = \int f \mathsf{d}\mu$.

Informally, the integral $\int f\, \mathsf{d}\mu$ is defined by $\sum_{n \in \mathbb{N}_0^\infty} n \cdot \mu\{x \mid fx = n\}$. The formal definition is

$$\int f\, \mathsf{d}\mu \;=\; \bigsqcup_{k \in \mathbb{N}} \textstyle\int_k f_k\, \mathsf{d}\mu \quad \text{where } f_k = \lambda x.\, k \sqcap fx \quad \text{and}$$

$$\textstyle\int_k g\, \mathsf{d}\mu \;=\; \sum_{n=0}^{k} n \cdot (\mu\{x \mid gx \geq n\} - \mu\{x \mid gx > n\}).$$

An equivalent definition is $\int f\, \mathsf{d}\mu = \sum_{n \in \mathbb{N}} \mu\{x \mid fx \geq n\}$. Note that this sum does not range over the cases $n = 0$ or $n = \infty$.

4.5 From Bags to Evaluations

A *bag* on a set X is a function $A : X \to \mathbb{N}_0^\infty$. For every x in X, Ax tells the number of occurrences of x in A. Note that this number is at most countably infinite. A bag is finite if Ax is finite for all x in X and the carrier set $\mathsf{set}\, A = \{x \in X \mid Ax > 0\}$ is finite. A bag A is a *subbag* of a bag B — $A \subseteq B$ — iff for all x in X, $Ax \leq Bx$ holds. The *empty bag* $\emptyset$ is the function $\lambda x.0$. The *sum* $A+B$ of two bags A and B is defined pointwise: $(A+B)x = Ax+Bx$. A bag A on a topological space X is just a bag $A : X \to \mathbb{N}_0^\infty$ on the underlying set X; the function A need not be continuous or even monotonic.

Let X be an arbitrary topological space. Given a bag A on X and a subset S of X, we define $A[S] = \sum_{x \in S} Ax \in \mathbb{N}_0^\infty$. Every bag A on X induces an evaluation A^* in $\mathcal{L}^*_\mathrm{m} X$ by defining $A^*(O) = A[O]$ for O in ΩX. With this definition, $(A+B)^* = A^* + B^*$ holds.

The subbag inclusion $A \subseteq B$ implies $A^* \sqsubseteq B^*$ in $\mathcal{L}^*_\mathrm{m} X$. The inverse implication does not hold in general. Later, criteria for $A^* \sqsubseteq B^*$ and $A^* = B^*$ will be presented.

Definition 4.2 Let A be a bag in a topological space X.

- The *infinite part* $\mathcal{I}(A)$ of A is the set of all points x such that $A^*(O) = \infty$ for every open O containing x. The *finite part* $\mathcal{F}(A)$ is the complement of $\mathcal{I}(A)$.
- From A, we define a bag $\overline{A}$ with $\overline{A}x = \infty$ for all x in $\mathcal{I}(A)$, and $\overline{A}x = Ax$ for x in $\mathcal{F}(A)$.
- The bag A is called ∞-*closed* iff $Ax = \infty$ for every x in $\mathcal{I}(A)$.

The map $A \mapsto \overline{A}$ is a closure operator which belongs to the notion of being ∞-closed, and $\overline{A} = \overline{B}$ implies $A^* = B^*$. The opposite implication holds in T0-spaces.

Theorem 4.3

For two bags A and B in a $\mathcal{T}_0$-space, the following statements are equivalent:

(1) $\overline{A} = \overline{B}$.

(2) $\mathcal{F}(A) = \mathcal{F}(B)$, and for all x in $\mathcal{F}(A)$, $Ax = Bx$ holds.

(3) $A^* = B^*$. □

Finite bags are ∞-closed. Hence:

Corollary 4.4 For finite bags A and B in a $\mathcal{T}_0$-space, $A^* = B^*$ implies $A = B$. □

Applying some auxiliary lemmas, a criterion for $A^* \sqsubseteq B^*$ is found.

Theorem 4.5 For two bags A and B in a topological space X, where B is ∞-closed, the following statements are equivalent:

(1) $A^* \sqsubseteq B^*$.

(2) $A[O] \le B[O]$ holds for all open sets O.

(3) $A[S] \le B[S]$ holds for all upper sets S.

(4) $A[F] \le B[\uparrow F]$ holds for all finite sets F.

(5) $A[F] \le B[\uparrow F]$ holds for all finite sets $F \subseteq \mathsf{set}\, A$.

(6) For every finite subbag A' of A, there is a finite subbag B' of B such that $A'^* \sqsubseteq B'^*$.□

4.6 Finite Bags and Evaluations

Above, arbitrary bags A were represented as functions $A : X \to \mathbb{N}_0^\infty$. For finite bags, we also use another kind of representation. A (finite) *bag enumeration* is a function $\alpha : I \to X$ where $I = \mathsf{dom}\, \alpha$ is a finite index set. We often write α_i for αi. A bag enumeration α *enumerates* the finite bag $\widehat{\alpha}$ with $\widehat{\alpha} x = \#\{i \in I \mid \alpha_i = x\}$, where $\#S$ is the cardinality of S. Two enumerations $\alpha : I \to X$ and $\beta : J \to X$ enumerate the same bag iff there is a bijection $\varphi : I \to J$ with $\beta \circ \varphi = \alpha$; we write $\alpha \cong \beta$ in this situation. Every finite bag A can be enumerated; let e.g., $I = \{(n, x) \in \mathbb{N} \times X \mid n \le Ax\}$ and $\alpha(n, x) = x$.

The action of an enumerated bag $\widehat{\alpha}$ on a set S can be described in terms of α:

Proposition 4.6 Let $\alpha : I \to X$ be a finite bag enumeration.
Then $\widehat{\alpha}[S] = \#\{i \in I \mid \alpha_i \in S\}$ holds for all subsets S of X. □

The evaluation induced by an enumerated bag $\widehat{\alpha}$ should be written as $\widehat{\alpha}^*$. We simplify this notation to α^*.

Theorem 4.7 Let X be an arbitrary topological space. For two bag enumerations $\alpha : I \to X$ and $\beta : J \to X$, the following statements are equivalent:

(1) $\alpha^* \sqsubseteq \beta^*$;

(2) for all U in ΩX: $\#\{i \in I \mid \alpha_i \in U\} \le \#\{j \in J \mid \beta_j \in U\}$;

(3) as (2), but even for all upper sets U of X;

(4) for all $T \subseteq I$: $\#T \le \#\{j \in J \mid \beta_j \in \uparrow\{\alpha_i \mid i \in T\}\}$;

(5) there is an injective function $\varphi : I \to J$ with $\alpha_i \sqsubseteq \beta(\varphi i)$, or shortly $\alpha \sqsubseteq \beta \circ \varphi$;

(6) there is a finite bag enumeration $\gamma : K \to X$ with $\alpha \cong \gamma$, $K \subseteq J$, and $\gamma_k \sqsubseteq \beta_k$ for all k in K. □

The pre-order on finite bags which is described by Theorem 4.7 is the same as the order introduced in [LW93]. Our Cor. 4.4 proves that it is anti-symmetric if the underlying space is a $\mathcal{T}_0$-space. This anti-symmetry is already proved in [LW93] by other means. Proposition 5.2 of [LW93] states that $\alpha^* \sqsubseteq \beta^*$ can be checked in time $O(n^{5/2} + n^2 \cdot k)$, where n is the sum of the sizes of the domains of α and β, and k is the time for comparisons in the argument space X.

4.7 From Evaluations to Bags

So far, we proved that the mapping $A \mapsto A^*$ is injective when restricted to ∞-closed bags. Now, we investigate its surjectivity: given an evaluation μ in $\mathcal{L}^*_{\mathrm{m}}X$, can we find a bag A in X such that $\mu = A^*$? Once we know that $\mu = A^*$ for some bag A, we also know that μ can be obtained as a (possibly infinite) sum of singletons $\sum_{i \in I} \mathsf{s}\alpha_i$, where $\alpha : I \to X$ is an enumeration of the (possibly infinite) bag A.

Let $\Im\mu = \{\mu O \mid O \in \Omega X\}$ denote the image of μ.

Definition 4.8 An evaluation $\mu : \Omega X \stackrel{mod}{\to} \mathbb{N}_0^\infty$ is

- *idempotent* if $\mu + \mu = \mu$ (or equivalently $\Im\mu \subseteq \{0, \infty\}$),
- *simple* if $\Im\mu$ is finite and does not contain ∞ (or equivalently $\mu X < \infty$),
- *primitive* if $\Im\mu = \{0, 1\}$.

With these notions, we show:

(1) Every evaluation can be written as the sum of an idempotent evaluation and a directed join of simple evaluations.

(2) Every idempotent evaluation is an infinite sum of point evaluations.

(3) Every simple evaluation is a finite sum of primitive evaluations.

(4) Every point evaluation is primitive. The converse holds iff X is sober.

A (possibly infinite) sum of point evaluations can be written as A^* for some bag A. Thus, we obtain:

Theorem 4.9 For a $\mathcal{T}_0$-space X, the following statements are equivalent:

(1) X is sober.

(2) Every evaluation in $\mathcal{L}^*_{\mathrm{m}}X$ can be written as a sum of point evaluations.

(3) For every evaluation μ in $\mathcal{L}^*_{\mathrm{m}}X$, there is a bag A such that $\mu = A^*$. □

4.8 The Continuous Case

The main theorems about continuous domains (dcpo's) are as follows:

Theorem 4.10 Let $\mathbf{X}$ be an (ω-)continuous domain with a basis B. Then $\mathcal{L}^*_{\mathrm{m}}\mathbf{X}$ is (ω-)continuous with basis $B^* = \{\mathsf{s}x_1 + \cdots + \mathsf{s}x_n \in \mathcal{L}^*_{\mathrm{m}}\mathbf{X} \mid n \in \mathbb{N}_0,\ x_i \in B\}$. □

Theorem 4.11 For continuous $\mathbf{X}$, $\mathcal{L}_{\mathrm{m}}^*\mathbf{X}$ is a free lower monoid over $\mathbf{X}$ in $\mathcal{DCPO}$. $\square$

Now that we know $\mathcal{L}_{\mathrm{i}}^*\mathbf{X} \cong \mathcal{L}_{\mathrm{f}}^*\mathbf{X} \cong \mathcal{L}_{\mathrm{m}}^*\mathbf{X}$ for continuous domains $\mathbf{X}$, we may use the simpler notation $\mathcal{L}^*\mathbf{X}$ without index.

4.9 Preservation of Domain Classes

We already know that the lower bag domain construction $\mathcal{L}^*$ preserves continuity of domains. Now we consider other domain classes.

Theorem 4.12

(1) If $\mathbf{X}$ is an (ω-)algebraic domain, then so is $\mathcal{L}^*\mathbf{X}$. Its canonical basis is $(\mathsf{K}\mathbf{X})^* = \{\mathsf{s}x_1 + \cdots + \mathsf{s}x_n \in \mathcal{L}^*\mathbf{X} \mid n \in \mathbb{N}_0,\ x_i \in \mathsf{K}\mathbf{X}\}$.

(2) If $\mathbf{X}$ is an I-domain, then so is $\mathcal{L}^*\mathbf{X}$.

(3) If $\mathbf{X}$ is an M-domain, then so is $\mathcal{L}^*\mathbf{X}$. $\square$

In contrast to the positive results above, the three classes of Scott domains, bifinite domains, and L-domains are not preserved under $\mathcal{L}^*$. This is particularly bad since these three classes are cartesian closed categories of algebraic domains, whereas the classes considered above are not.

Proposition 4.13 Let $\mathbf{X}$ be an M-domain which contains three compact elements a, b, and c, where a and b are incomparable, and $c = a \sqcup b$ holds. Then $\mathcal{L}^*\mathbf{X}$ is neither bifinite nor an L-domain. $\square$

The simplest Scott domain which satisfies the hypothesis of Prop. 4.13 is $\mathbf{X} = \{\perp, a, b, c\}$, where c is the largest element, and a and b are incomparable. It is even a finite complete distributive lattice. The least element $\perp$ is not required; the dcpo $\{a, b, c\}$ would do the job as well.

Prop. 4.13 has the unfortunate consequence that there is no cartesian closed category of algebraic domains which would be closed under $\mathcal{L}^*$ as well: Starting with the one-point domain $\mathbf{1}$, we may form $\mathbf{X} = \mathcal{L}^*\mathbf{1} \times \mathcal{L}^*\mathbf{1} \cong \mathbb{N}_0^\infty \times \mathbb{N}_0^\infty$. The domain $\mathbf{X}$ satisfies the hypothesis of Prop. 4.13 with $a = (1, 0)$, $b = (0, 1)$, and $c = (1, 1)$. Thus, $\mathbf{Y} = \mathcal{L}^*\mathbf{X}$ is neither bifinite nor an L-domain. By Cor. 3.7 in [Jun90], the function space $[\mathbf{Y} \to \mathbf{Y}]$ is not algebraic.

4.10 Related Work

We already mentioned Libkin and Wong [LW93] who investigate the expressibility of various query languages for data bases. To compare informations with respect to their partiality, they introduce an order on bags that is identical to our order (the one characterized in Theorem 4.7). Apart from defining the order, they do not introduce a bag domain construction as a monad producing monoids, nor do they study the preservation of domain classes.

A different order is used by Broy [Bro82]. Remember that we have $\alpha^* \sqsubseteq \beta^*$ iff there is an injective function $\varphi : \mathsf{dom}\,\alpha \to \mathsf{dom}\,\beta$ with $\alpha \sqsubseteq \beta \circ \varphi$, which means that bags may become bigger by increasing their elements or by acquiring new elements. Broy defines $\alpha^* \sqsubseteq \beta^*$ iff there is a surjective function $\psi : \mathsf{dom}\,\beta \to \mathsf{dom}\,\alpha$ with $\alpha \circ \psi \sqsubseteq \beta$, which means that bags may become bigger by increasing their elements or by duplicating existing elements. Broy's order always is a subset of our order. In general, it is a proper subset, but for bags containing at least one occurrence of $\perp$, both orders are equivalent. Whereas we derived our order from general considerations, Broy introduces his order in an ad-hoc manner. Nevertheless, he also obtains a continuous singleton function s, extension functional E, and bag addition '+'. Broy uses his construction to specify the denotational semantics of a programming language. He does not give a universal property of his bag domains, nor a description in terms of second order functions. Concerning the preservation of domain classes, Broy's construction suffers from the same problem as ours. This can be shown at the argument domain $\{\perp, a, b, c\}$.

Quite a different kind of 'lower bag domain' is introduced in [Vic92], which is discussed in a more general context in [Joh92]. Vickers's notion of domain is far more general than ours; his bag domains are small categories, which are not just posets. Given an algebraic domain (= dcpo) $\mathbf{X}$, he constructs a category whose objects are the enumerations of finite bags of compact elements of $\mathbf{X}$. The morphisms from a bag enumeration α to a bag enumeration β are all functions $\varphi : \mathsf{dom}\,\alpha \to \mathsf{dom}\,\beta$ with $\alpha \sqsubseteq \beta \circ \varphi$. The bag 'domain' is the categorical 'ideal completion' of this category. In contrast to our approach, Vickers considers *all* functions between the index sets of the bag enumerations, not just the *injective* functions as we do. Thus, a bag increases in the sense of Vickers if its elements become bigger, new elements appear, *or different occurrences of the same element are identified.* This order does not take into account the multiplicities of elements of a bag. In fact, the poset reflection of Vickers's bag 'domain' is not our bag domain, but the lower power domain.

5 Probabilistic Domains [Hec94b]

We show the equivalence of several different axiomatizations of the notion of (abstract) *probabilistic domain* in the category $\mathcal{DCPO}$. The axiomatization with the richest set of operations provides probabilistic selection among a finite number of possibilities with arbitrary probabilities, whereas the poorest one has binary choice with equal probabilities as the only operation. It is the theory mentioned in Subsection 2.2. The remaining theories lie in between; one of them is the theory of binary choice by Graham [Gra88].

The subsections of this section correspond to the sections of the summarized paper.

5.1 Introduction

A probabilistic programming language could contain different kinds of language constructs to express probabilistic choice. In a rather poor language, there might be a construct $x \oplus y$, whose semantics is a choice between the two possibilities x and y with equal probabilities 1/2. A quite rich language could contain a construct $[p_1 : x_1, \ldots, p_n : x_n]$, where p_i are real numbers between 0 and 1 whose sum is 1. The semantics would be to select one of the possibilities x_i with probability p_i.

Graham [Gra88], Jones [Jon90], and Jones / Plotkin [JP89] consider an intermediate language with a construct $x \xrightarrow{p} y$, where p is a real number between 0 and 1. The semantics of this construct is to select x with probability p and y with probability $1 - p$. The notion of an (abstract) *probabilistic domain* was introduced by Graham [Gra88] and further elaborated by

Jones [Jon90, JP89] to describe the denotational semantics of a probabilistic language with the construct $x \xrightarrow{p} y$. A probabilistic domain is a dcpo together with a continuous operation satisfying several axioms, which is used to model the choice construct semantically. For reasons of simplicity, we denote the results of this semantic operation by $x \xrightarrow{p} y$ as well, where x and y are no longer language constructs, but members of the underlying dcpo.

In the summarized paper, we introduce other kinds of probabilistic domains which describe a whole range of language constructs from the binary choice $x \oplus y$ over $x \xrightarrow{p} y$ till $[p_1 : x_1, \ldots, p_n : x_n]$. Every kind of probabilistic domain is described as model of some algebraic theory in the category $\mathcal{DCPO}$. Then, we prove that *all these theories are equivalent* in the sense that their categories of models are equivalent. This has several consequences. First, one sees that in the category $\mathcal{DCPO}$, all the probabilistic operations mentioned above are in fact equally expressive; even the multiple choice $[p_1 : x_1, \ldots, p_n : x_n]$ can be defined in terms of binary choice $x \oplus y$ (with the help of fixed point iteration). Next, the probabilistic power construction of [Jon90, JP89] not only produces the free probabilistic domain with $x \xrightarrow{p} y$, but free probabilistic domains with all other operations as well because all the categories of probabilistic domains are equivalent. Thus, one power construction can be used to describe the semantics of a whole range of probabilistic languages.

5.2 Mathematical Background

The *unit interval* $\mathbb{I}$ is the set of real numbers r with $0 \le r \le 1$. We can either consider it as a dcpo ordered by '$\le$' and equip it with the Scott topology, or topologize it by the standard Hausdorff topology.

5.3 Multiple Choice with Divergence (MCD)

The basic operation of MCD is choice between n possibilities $x_1, \ldots, x_n$ with probabilities $p_1, \ldots, p_n$, whose sum is at most 1.

In the following table as well as in all subsequent ones, we assume that the probabilistic domain to be described is a *dcpo* called X.

Operations: Continuous functions $\nabla_n \times X^n \to X$ for every $n \ge 0$, where

$$\nabla_n = \{(p_1, \ldots, p_n) \in \mathbb{I}^n \mid p_1 + \cdots + p_n \le 1\}$$

are dcpo's, i.e., equipped with the Scott topology. The result of applying the operation of degree n to $(p_1, \ldots, p_n)$ in ∇_n and $(x_1, \ldots, x_n)$ in X^n is written $[p_1 : x_1, \ldots, p_n : x_n]$.

Axioms: The axioms are those of linear combinations $\sum_{i=1}^{n} p_i \cdot x_i$, namely:

Permutation P: For every permutation π of $\{1, \ldots, n\}$,

$$[p_{\pi 1} : x_{\pi 1}, \ldots, p_{\pi n} : x_{\pi n}] \;=\; [p_1 : x_1, \ldots, p_n : x_n]$$

1 law: $[1 : x] \;=\; x$

0 law: $[0 : x, p_1 : y_1, \ldots, p_n : y_n] \;=\; [p_1 : y_1, \ldots, p_n : y_n]$

Addition $+$: $[p : x, q : x, r_1 : y_1, \ldots, r_n : y_n] \;=\; [p + q : x, r_1 : y_1, \ldots, r_n : y_n]$

Substitution S: $[p : [q_1 : x_1, \ldots, q_k : x_k], r_1 : y_1, \ldots, r_n : y_n]$
$$= [p\, q_1 : x_1, \ldots, p\, q_k : x_k, r_1 : y_1, \ldots, r_n : y_n]$$

5.4 Binary Choice without Divergence (BC)

Theory BC comes up in four versions with different sets of axioms. The binary choice $x \overset{p}{-} y$ introduced here corresponds to $[p : x,\ 1 - p : y]$ from above. Whereas MCD refers to ∇_n with the Scott topology, the BC theories refer to $\mathbb{I}$ with the Hausdorff topology.

Operations: A constant $0 : X$ and an operation $\beta : \mathbb{I} \times X \times X \to X$, which is continuous in each argument separately, if $\mathbb{I}$ is equipped with the standard Hausdorff topology. We write $x \overset{p}{-} y$ instead of $\beta(p,\ x,\ y)$.

Axioms, version L (large axiom set):

$\perp$	Least element:	$0 \sqsubseteq x$ for all x in X.
C	Commutativity:	$x \overset{p}{-} y = y \overset{1-p}{-} x$.
0	Zero law:	$x \overset{0}{-} y = y$.
1	One law:	$x \overset{1}{-} y = x$.
I	Idempotence:	$x \overset{p}{-} x = x$.
D	Distributivity:	$x \overset{p}{-} (y \overset{q}{-} z) = (x \overset{p}{-} y) \overset{q}{-} (x \overset{p}{-} z)$.
P	Product law:	$(x \overset{p}{-} y) \overset{q}{-} y = x \overset{pq}{-} y$.
L	Linear combination:	$(x \overset{p}{-} y) \overset{r}{-} (x \overset{q}{-} y) = x \overset{s}{-} y$ where $s = rp + (1 - r)q$.
R	Rectangle law:	$(x \overset{q}{-} y) \overset{p}{-} (u \overset{q}{-} v) = (x \overset{p}{-} u) \overset{q}{-} (y \overset{p}{-} v)$.
A	Associativity:	If $pq \neq 1$, then $(x \overset{p}{-} y) \overset{q}{-} z = x \overset{pq}{-} (y \overset{r}{-} z)$, where $r = \frac{q(1-p)}{1-pq}$.

Axioms, version A (with associativity): Axioms $\perp$, C, 1, I, and A.

Axioms, version R (with rectangle law): Axioms $\perp$, 0, 1, L, and R.

Axioms, version P (with product law): Axioms $\perp$, 0, 1, I, P, $L\frac{1}{2}$, and $R\frac{1}{2}$, where $L\frac{1}{2}$ is the instance of L with $r = 1/2$, and $R\frac{1}{2}$ is the instance of R with $p = q = 1/2$.

We presented theory BC in four versions because the equivalence of the respective sets of axioms is by no means obvious. Theory BC-L comprises all interesting equalities we know of, BC-A is the theory of Graham and Jones, BC-R is the nicest theory in our view since it consists of simple unconditional axioms only, and BC-P consists of just the properties we need to derive IM-S (which in turn consists of just the properties we need to derive MV).

In the summarized paper, BC-L is derived from MCD, the derivation of BC-A and BC-R from BC-L is trivial, and BC-P is derived from BC-R as well as from BC-A.

5.5 $\mathbb{I}$-Modules

Now, we introduce the theory IM of $\mathbb{I}$-modules. Its 'addition' is binary choice with equal probabilities 1/2, and its 'multiplication' is choice between a point and 0. The axioms are very much like those of a module, hence the name.

Theory IM comes up in two versions: IM-L has a large set of axioms which includes all useful properties we know of, and IM-S has a small subset thereof, just enough to derive theory MV in the next section.

Operations: A constant $0 : X$, a continuous operation $\oplus : X \times X \to X$, and a continuous operation $\cdot : \mathbb{I} \times X \to X$, where $\mathbb{I}$ is considered as a dcpo (Scott topology).

Axioms, version L (large axiom set):

$\perp$	Least element:	$0 \sqsubseteq x$ for all x in X.
C	Commutativity:	$x \oplus y = y \oplus x$.
I	Idempotence:	$x \oplus x = x$.
R	Rectangle law:	$(x \oplus y) \oplus (u \oplus v) = (x \oplus u) \oplus (y \oplus v)$.
$R0$	Right zero:	$p \cdot 0 = 0$.
RD	Right distributivity:	$p \cdot (x \oplus y) = p \cdot x \oplus p \cdot y$.
$L0$	Left zero:	$0 \cdot x = 0$.
LD	Left distributivity:	$(p \oplus q) \cdot x = p \cdot x \oplus q \cdot x$, where $p \oplus q = (p+q)/2$ in $\mathbb{I}$.
1	One law:	$1 \cdot x = x$.
PA	Product associativity:	$p \cdot (q \cdot x) = (p \cdot q) \cdot x$.

Axioms, version S (small axiom set): Axioms R, $L0$, LD, and 1.

Theory IM-S is derived from BC-P, and IM-L from BC-L.

5.6 Mean Values

The probabilistic theory with the weakest operations is MV, the theory of mean values. It results from IM by dropping multiplication. Thus, MV does not mention real numbers explicitly.

Operations: A constant $0 : X$, and a continuous operation $\oplus : X \times X \to X$.

Axioms: Commutativity: $x \oplus y = y \oplus x$,

Rectangle law: $(x \oplus y) \oplus (u \oplus v) = (x \oplus u) \oplus (y \oplus v)$,

Least element: $0 \sqsubseteq x$,

Fixed point axiom: The least fixed point of $\lambda x.\, a \oplus x$ is a.

For the fixed point axiom, remember that the carrier X is a dcpo. The fixed point axiom may alternatively be written as idempotence $a \oplus a = a$, which means that a is a fixed point of $\lambda x.\, a \oplus x$, plus the conditional axiom $a \oplus b = b \Rightarrow a \sqsubseteq b$, which means that a is the least fixed point. Yet another formulation follows from making the fixed point iteration explicit: if $a_0 = 0$ and $a_{n+1} = a \oplus a_n$, then $\bigsqcup_{n \geq 0} a_n = a$. Notice that MV is identical to the algebraic theory LP of Subsection 2.2.

In the summarized paper, theory MV is derived from IM-S.

5.7 The Big Step: From MV to MCD

To show equivalence of the theories, we assume an MV algebra X as given, and have to prove that it is an MCD algebra as well. To this end, we have to reconstruct multiple choice with arbitrary probabilities from binary choice with probabilities 1/2. The first step is to introduce multiple binary choice.

Let X be an MV algebra. For every n, we define an operator $\bigoplus_n : X^{2^n} \to X$ which takes 2^n arguments from X.

- $\bigoplus_0(x) = x$.
- $\bigoplus_{n+1}(x_i \mid i = 1 \,..\, 2^{n+1}) \;=\; \bigoplus_n(x_i \mid i = 1 \,..\, 2^n) \oplus \bigoplus_n(x_i \mid i = 2^n + 1 \,..\, 2^{n+1})$.

Thus, $\bigoplus_n(x_1, \ldots, x_{2^n})$ is obtained by evaluating a complete binary tree of depth n, whose inner nodes are marked by '$\oplus$' and whose leaves are x_1 through x_{2^n}.

From commutativity and the rectangle law, we can conclude:

Proposition 5.1 The operands of $\bigoplus_n$ can be arbitrarily permuted. □

In the sequel, we shall use an abbreviation: if an argument of $\bigoplus_n$ occurs k times, we shall write $k \cdot x$, e.g., $\bigoplus_2(3 \cdot x, 1 \cdot y, 0 \cdot z) = \bigoplus_2(x, x, x, y)$.

Now, we construct values corresponding to multiple choice $[p_1 : x_1, \ldots, p_r : x_r]$ with $p_i = k_i/2^n$ where k_i and n are non-negative integers. For simplicity, the exponent of the denominator is written as an index, i.e., we define values

$$\langle k_1 : x_1, \ldots, k_r : x_r\rangle_n$$

where $r \geq 0$, x_i in X, n and k_i are non-negative integers with $k_1 + \cdots + k_r \leq 2^n$.

The definition of these values is in terms of $\bigoplus_n$, where the necessary number of arguments is obtained by filling up with 0.

$$\langle k_1 : x_1, \ldots, k_r : x_r\rangle_n \;=\; \bigoplus_n(k_1 \cdot x_1, \ldots, k_r \cdot x_r, (2^n - \sum_{i=1}^{r} k_i) \cdot 0)$$

These dyadic choice expressions satisfy close analogues of the axioms of MCD. They are used to define choice expressions with arbitrary real numbers as coefficients.

For every p in $\mathbb{I}$, we define $p^{(n)}$ to be the greatest non-negative integer k with $k/2^n < p$ (setting $p^{(n)} = 0$ if $p = 0$). With this notation, arbitrary choice is defined as follows:

$$[p_1 : x_1, \ldots, p_r : x_r] = \bigsqcup_{n \geq 0} \langle p_1^{(n)} : x_1, \ldots, p_r^{(n)} : x_r\rangle_n$$

The join is well-defined because it is directed. The MCD-expressions are continuous in all arguments x_i and p_i. A multiple choice with coefficients which happen to be dyadic rationals coincides with the dyadic choice introduced earlier:

$$[k_1/2^n : x_1, \ldots, k_r/2^n : x_r] \;=\; \langle k_1 : x_1, \ldots, k_r : x_r\rangle_n$$

Finally, the MCD-expressions are shown to satisfy all the axioms of MCD listed in Section 5.3.

5.8 Going Back and Forth

The derivations of MV from MCD and vice versa are inverse to each other. If we start with an MV algebra and construct multiple choice, then $[1/2 : x, 1/2 : y]$ equals $x \oplus y$. This means that we get back the original MV algebra by restriction.

Conversely, if we start with a multiple choice operator, restrict it to the special case $x \oplus y = [1/2 : x, 1/2 : y]$, and then reconstruct multiple choice following the lines of this section, then we obtain the original multiple choice back.

If homomorphisms for theory T are introduced as continuous functions commuting with the operations of T, then one can prove that all the theories of probabilistic domains have the same homomorphisms. Thus, the categories of models of all these theories in $\mathcal{DCPO}$ are isomorphic to each other.

6 Probabilistic Power Domains, Information Systems, and Locales [Hec94c]

The probabilistic power domain construction $\mathcal{P}_D$ of Jones and Plotkin [Jon90, JP89] is defined by a construction on dcpo's. In the summarized paper, we present alternative definitions in terms of information systems à la Vickers [Vic93], and in terms of locales. On continuous domains, all three definitions coincide.

The subsections of this section correspond to the sections of the summarized paper.

6.1 Introduction

In [Jon90, JP89], it is shown that $\mathcal{P}_D$ preserves (ω-)continuity of dcpo's. On the other hand, it does not preserve algebraicity: even the power domain of the one-point domain is not algebraic. Thus, $\mathcal{P}_D$ cannot be described by the more conventional information systems, which are only suited for certain classes of algebraic domains, as for instance the information systems of [Sco82] for the class of bounded complete algebraic domains (Scott Domains). The information systems of [Hoo90, Hoo92] present bounded complete continuous domains, but this is still not sufficient, since $\mathcal{P}_D$ does not preserve bounded completeness as shown in [Jon90, Section 4.5].

In [Vic93], Vickers introduced a kind of information systems (*infosyses*) suitable to cover all continuous domains. In the summarized paper, we show that $\mathcal{P}_D$ can be described in terms of these infosyses. This was already conjectured by Vickers. We also looked for a localic description of the power construction. Starting from the frame of opens of a continuous base domain, we show how the frame of opens of the power domain may be constructed in terms of generators and relations.

6.2 The Background

We assume the reader to be familiar with the frames and locales of [Joh82] or [Vic89]. We denote the least element of a frame by 0 or F standing for 'false', and the greatest element by T for 'true', never 1.

Topological Systems

In [Vic89], Vickers introduced *topological systems* as a common generalization of topological spaces and locales. In analogy to his abbreviation 'infosys' for 'information system', we shall abbreviate 'topological system' by 'topsys'. A *topsys* is a pair $X = (\mathsf{pt}\, X, \Omega X)$ of a set of 'points' $\mathsf{pt}\, X$ and a frame of 'opens' ΩX together with a relation '$\models$' between points and opens, which respects the frame operations.

A topological space X induces a topsys, where $\mathsf{pt}\, X$ is the underlying set of X, ΩX is the frame of open sets of X, and $x \models u$ iff $x \in u$. This in particular applies to dcpo's with their Scott topology.

A locale X is defined by a frame ΩX. It becomes a topsys by taking $\mathsf{pt}\, X$ as the set of frame homomorphisms from ΩX to $\mathbf{2} = \{\mathsf{F}, \mathsf{T}\}$, with $x \models u$ iff $xu = \mathsf{T}$.

A continuous function $f : X \to Y$ between topsyses has two components, $\mathsf{pt}\, f : \mathsf{pt}\, X \to \mathsf{pt}\, Y$ and $\Omega f : \Omega Y \to \Omega X$, where Ωf is a frame homomorphism and $\mathsf{pt}\, f x \vDash v$ iff $x \vDash \Omega f v$ for all x in $\mathsf{pt}\, X$ and v in ΩY. In case of topological spaces, these are the usual continuous functions, and f is determined by $\mathsf{pt}\, f$. In case of locales, these are frame homomorphisms in the opposite direction, and f is determined by Ωf.

Vickers Information Systems (Infosyses)

An *infosys* is a pair $(D, <)$ of a set of *tokens* D and a binary relation '$<$' on D which is *transitive* — $a < b$ and $b < c$ implies $a < c$ — and *interpolative* — if $a < c$, then there is a token b with $a < b$ and $b < c$. In contrast to the more conventional preorders, reflexivity is not required.

We need several notions and notations: for a set $A \subseteq D$, $\uparrow A = \{b \in D \mid \exists a \in A : a < b\}$ is the upper set of A. We use the abbreviation $\sharp a$ for $\uparrow\{a\}$. The set of upper bounds of a token set A is $\mathsf{ub}\, A = \{b \in D \mid \forall a \in A : a < b\}$.

A *point* of an infosys is a subset x of D with properties analogous to those of ideals: if $a < b$ and $b \in x$, then $a \in x$; and for $a_1, \ldots, a_n$ in x, there is b in x with $a_1, \ldots, a_n < b$. For the second condition, cases $n = 0$ (x is not empty) and $n = 2$ are enough. The points of an infosys D ordered by inclusion form a continuous dcpo $\mathsf{pt}\, D$. This is the continuous domain represented by the infosys D.

Conversely, for every continuous dcpo X, there is an infosys I with $\mathsf{pt}\, I \cong X$, namely $I = (X, \ll)$, where '$\ll$' is the way-below relation on X. In general, there are many other infosyses I with $\mathsf{pt}\, I \cong X$.

An *open* of an infosys is a subset u of D with $\uparrow u = u$. Ordered by inclusion, the opens of D form a frame ΩD. The joins of this frame are simply given by union, whereas the meets are *not* given by intersection: $u \wedge v$ is $\uparrow(u \cap v)$. The mapping $\omega : \Omega D \to 2^{\mathsf{pt}\, D}$ with $\omega u = \{x \in \mathsf{pt}\, D \mid x \cap u \neq \emptyset\}$ provides an isomorphism between the frame ΩD and the frame $\Omega(\mathsf{pt}\, D)$ of Scott open sets of the continuous domain $\mathsf{pt}\, D$. Thus, the topsys induced by the dcpo $\mathsf{pt}\, D$ is isomorphic to $(\mathsf{pt}\, D, \Omega D)$ with $p \vDash u$ iff $p \cap u \neq \emptyset$. We call the latter topsys $[D]$. Two infosyses D and E are *equivalent* iff $[D] \cong [E]$.

The frame ΩD of an infosys D can be presented by generators and relations: Generators are $\sharp a$ for a in D, and the relations are $\bigwedge_{a \in S} \sharp a = \bigvee_{b \in \mathsf{ub}\, S} \sharp b$ for every finite subset S of D. This one relation can be equivalently replaced by three:

(1) Monotonicity: If $a < b$, then $\sharp a \geq \sharp b$;

(2) All tokens: $\bigvee_{a \in D} \sharp a = \mathsf{T}$;

(3) Meets: $\sharp a \wedge \sharp b \leq \bigvee_{c > a,b} \sharp c$.

The Unit Interval

We denote the usual order on real numbers by '$\sqsubseteq$' instead of '$\leq$', and accordingly its strict variant ($x \sqsubseteq y$ and $x \neq y$) by '$\sqsubset$' instead of '$<$'. This is done to avoid confusion with the infosys order introduced below.

By these conventions, the unit interval is $\mathbb{I} = \{r \in \mathbb{R} \mid 0 \sqsubseteq r \sqsubseteq 1\}$ ordered by '$\sqsubseteq$'. It forms a continuous domain, which can be described by the infosys $\mathbb{QI}$ with tokens $\{q \in \mathbb{Q} \mid 0 \sqsubseteq q \sqsubset 1\}$ and order $x < y$ iff $x \sqsubset y$ or $x = y = 0$.

The frame $\Omega\mathbb{QI} \cong \Omega\mathbb{I}$ can be presented by generators $\sharp q$ for q in $\mathbb{QI}$ with relations

(1) Rational zero: $\sharp 0 = \top$;

(2) Rational continuity: $\bigvee_{r>q} \sharp r = \sharp q$.

Probabilistic Power Domains

For given dcpo X, the *probabilistic power domain* $\mathcal{P}_D X$ is defined in [Jon90, JP89] as $\mathcal{P}_D X = [\Omega X \stackrel{mod}{\to} \mathbb{I}]$, the dcpo of *continuous evaluations* from ΩX to $\mathbb{I}$ with the pointwise order. A function μ from a frame to $\mathbb{I}$ is an evaluation if it satisfies the zero law $\mu(0) = 0$ and the modular law $\mu(a \vee b) + \mu(a \wedge b) = \mu a + \mu b$.

Since the definition of $\mathcal{P}_D X$ only refers to ΩX, it can be applied to any topsys X producing a dcpo and thus a topsys. Similarly, the probabilistic power locale construction $\mathcal{P}_L$ soon to be introduced only relies on ΩX and thus can be applied to any topsys. We shall prove (Theorem 6.1):

(1) For every topsys X, $\mathsf{pt}\,\mathcal{P}_D X$ and $\mathsf{pt}\,\mathcal{P}_L X$ are isomorphic posets (in fact dcpo's).

In Subsection 6.4, we shall introduce the probabilistic power infosys construction $\mathcal{P}_I$ mapping infosyses to infosyses. We shall prove (Theorem 6.10):

(2) For every infosys D, $\Omega \mathcal{P}_I D$ and $\Omega \mathcal{P}_L [D]$ are isomorphic frames.

Since the topsyses described by infosyses are continuous domains and thus localic (sober), (2) suffices to conclude:

(3) For every infosys D, $[\mathcal{P}_I D]$ and $\mathcal{P}_L [D]$ are isomorphic topsyses.

First, we can conclude from this that $\mathcal{P}_I$ preserves equivalence of infosyses, which would not be obvious a priori. Second, we can conclude that for every continuous domain X, $\mathcal{P}_L X$ is a continuous domain again, and thus a dcpo. The topsyses induced by dcpo's are spatial, whence (1) suffices to conclude

(4) For every continuous domain X, $\mathcal{P}_D X$ and $\mathcal{P}_L X$ are isomorphic topsyses (in fact continuous domains).

This is our main result: on continuous domains, $\mathcal{P}_D$, $\mathcal{P}_L$, and $\mathcal{P}_I$ coincide.

6.3 A Probabilistic Power Locale

Now, we introduce the probabilistic power locale construction $\mathcal{P}_L$ and show that its power locales have the same points as Jones's power domains.

The locale $\mathcal{P}_L X$ for a topsys X should be close to the dcpo $\mathcal{P}_D X = [\Omega X \stackrel{mod}{\to} \mathbb{I}]$. Ideally, $\Omega \mathcal{P}_L X$ should be the frame of Scott open sets of $\mathcal{P}_D X$, but this frame is difficult to axiomatize. Instead, we shall consider the pointwise topology on the function space $[\Omega X \to \mathbb{I}]$. Its subbasic opens are $F(u, v)$ for members u of ΩX and opens v in $\Omega \mathbb{I}$, where $F(u, v) = \{\mu : \Omega X \to \mathbb{I} \mid \mu u \in v\}$. Instead of all opens in $\Omega \mathbb{I}$, it suffices to consider the basic opens $\sharp q$ for q in $\mathbb{Q}\mathbb{I}$. Thus, the frame $\Omega \mathcal{P}_L X$ will have generators $\langle u, q \rangle$ with u in ΩX and q in $\mathbb{Q}\mathbb{I}$, where the intended meaning of $\langle u, q \rangle$ is $\{\mu \mid \mu u > q\}$.

In the summarized paper, the relations on the frame $\Omega \mathcal{P}_L X$ are derived from the properties of the functions in $[\Omega X \to \mathbb{I}]$, namely mapping to $\mathbb{I}$ and being continuous, strict, and modular. The results are summarized in the following theorem:

Theorem 6.1 For every topsys X, let $\mathcal{P}_D X$ be the dcpo $[\Omega X \stackrel{mod}{\to} \mathbb{I}]$, and $\mathcal{P}_L X$ be the locale, whose frame of opens $\Omega \mathcal{P}_L X$ is presented by:

Generators: $\langle u, q\rangle$ with u in ΩX and q in $\mathbb{QI}$,

Relations: Rational zero: $\langle u, 0\rangle = \top$ for all u in ΩX;

Rational continuity: $\bigvee_{r>q}\langle u, r\rangle = \langle u, q\rangle$ for all u in ΩX and q in $\mathbb{QI}$;

Continuity: If $\mathcal{U} \subseteq \Omega X$ is directed, then $\langle \bigvee \mathcal{U}, q\rangle = \bigvee_{u\in\mathcal{U}}\langle u, q\rangle$;

Zero law: If $q \neq 0$, then $\langle 0, q\rangle = 0$;

Modularity: For all rational q with $0 \sqsubseteq q \sqsubset 2$, and all u, v in ΩX,

$$\bigvee\{\langle u, r\rangle \wedge \langle v, s\rangle \mid r, s \in \mathbb{QI}, r + s = q\} =$$
$$\bigvee\{\langle u \vee v, r\rangle \wedge \langle u \wedge v, s\rangle \mid r, s \in \mathbb{QI}, r + s = q\}.$$

Then there is a continuous function $f : \mathcal{P}_D X \to \mathcal{P}_L X$ defined by

$$\Omega f\langle u, q\rangle = \{\mu \in [\Omega X \overset{mod}{\to} \mathbb{I}] \mid \mu u > q\}$$

such that $\mathsf{pt}\, f : \mathsf{pt}\, \mathcal{P}_D X \to \mathsf{pt}\, \mathcal{P}_L X$ is a poset isomorphism. □

6.4 The Power Construction on Infosyses

Here, we define the probabilistic construction $\mathcal{P}_I$ on infosyses. The power tokens (tokens of $\mathcal{P}_I D$) may be thought of as formal convex combinations $\sum_{i=1}^{n} r_i \cdot a_i$ with a_i in D and r_i in $\mathbb{QI}$ such that $\sum_{i=1}^{n} r_i < 1$. This is formalized in the following definition, which was already conjectured in [Vic93].

Definition 6.2 The tokens of $\mathcal{P}_I D$ are finite bags A of pairs (r, a) from $\mathbb{QI} \times D$ with $\sum\{\!|r \mid (r, a) \in A|\!\} < 1$.

Bags or *multi-sets* are similar to sets, but elements may occur more than once in them. Notationally, we use $\{\!|.|\!\}$ for bags in contrast to $\{.\}$ for sets, but keep on using '$\emptyset$', '$\in$', and '$\subseteq$' as for sets. The bag union of two bags is denoted by '+', e.g., $\{\!|1|\!\} + \{\!|1, 2|\!\} = \{\!|1, 1, 2|\!\}$. Bags will be manipulated by means of *bag abstractions*, whose meaning is analogous to that of set abstractions, but keeping different occurrences of the same element apart. Thus, for instance $\sum\{\!|r \mid (r, a) \in \{\!|(0.2, b), (0.2, b), (0.2, c)|\!\}|\!\} = \sum\{\!|0.2, 0.2, 0.2|\!\} = 0.6$.

In the sequel, we need some notions and notations concerning power tokens.

Definition 6.3 For power tokens A in $\mathcal{P}_I D$, let $\mathsf{set}\, A = \{a \mid (r, a) \in A\}$ be the *set* of D-tokens occurring in A, and $\mathsf{num}\, A = \{\!|r \mid (r, a) \in A|\!\}$ be the *bag* of numbers in A.

For power tokens A in $\mathcal{P}_I D$ and opens u in ΩD, let $A_p u = \sum\{\!|r \mid (r, a) \in A,\ a \in u|\!\}$ and $A_o u = \sum\{\!|r \mid (r, a) \in A,\ \sharp a \subseteq u|\!\}$.

We use the indices p and o to indicate that in the first case, the token a is considered as a *point* of the open u, whereas in the latter case, it codes for the *open* $\sharp a$.

Proposition 6.4 For all A in $\mathcal{P}_I D$:

(1) For all u in ΩD: $0 < A_p u \sqsubseteq A_o u \sqsubseteq \sum \mathsf{num}\, A < 1$.

(2) A_p is continuous, and A_o is monotonic.

(3) A_p satisfies the zero law: $A_p \emptyset = 0$.

(4) For all u in ΩD: $\emptyset_p u = \emptyset_o u = 0$. □

Neither A_p nor A_u are modular in general, but they satisfy some kind of modularity in co-operation.

Proposition 6.5
For all A in $\mathcal{P}_I D$ and u, v in ΩD, $A_p u + A_p v \sqsubseteq A_p(u \vee v) + A_o(u \wedge v) \sqsubseteq A_o u + A_o v$. $\square$

Next, we define the order on the power tokens.

Definition 6.6 For two functions f and g from ΩD to $\mathbb{I}$, we define $f < g$ iff for all u in ΩD, $fu < gu$ holds. For two tokens A and B in $\mathcal{P}_I D$, we define $A < B$ iff $A_o < B_p$ holds.

Proposition 6.7 For a token A in $\mathcal{P}_I D$ and a monotonic function $\mu : \Omega D \to \mathbb{I}$, the following three statements are equivalent:

(1) $A_o < \mu$, i.e., for all u in ΩD, $A_o u < \mu u$ holds.

(2) $A_o u < \mu u$ holds for all u in $\mathcal{U}(A) = \{\uparrow \mathsf{set}\, S \mid S \subseteq A\}$.

(3) For all subbags S of A, $\sum \mathsf{num}\, S < \mu(\uparrow \mathsf{set}\, S)$ holds.

In (3), one may also quantify 'for all non-empty subbags'. $\square$

Notice how the universal quantification of (1) is reduced to the finite quantifications in (2) and (3).

Corollary 6.8
For two tokens A and B in $\mathcal{P}_I D$, $B > A$ iff for all $S \subseteq A$, $B_p(\uparrow \mathsf{set}\, S) > \sum \mathsf{num}\, S$. $\square$

The intuitions of Vickers in [Vic93] suggest a quite different definition of the order on the tokens. A token A is below a token B iff every element (r, a) can be split into some (r_i, a) with $\sum r_i = r$, then grown by enlarging both the rational number and the ground token, then recombined again to obtain some (not necessarily all) of the elements of B.

Theorem 6.9 Let $A = \{\!|(r_1, a_1), \ldots, (r_n, a_n)|\!\}$ and $B = \{\!|(s_1, b_1), \ldots, (s_m, b_m)|\!\}$ be two power tokens with a fixed order to enumerate their elements, and let $I = \{1, \ldots, n\}$ and $J = \{1, \ldots, m\}$. Then the following are equivalent:

(1) $A < B$, i.e., for all opens u, $A_o u < B_p u$;

(2) there are numbers t_{ij} in $\mathbb{I}$ for every i in I and j in J, such that $t_{ij} \neq 0$ implies $a_i < b_j$, $\sum_{j \in J} t_{ij} = r_i$ for every i in I, and $\sum_{i \in I} t_{ij} < s_j$ for every j in J.

(3) as (2), but the t_{ij} are in $\mathbb{Q}\mathbb{I}$, and $\sum_{j \in J} t_{ij} > r_i$. $\square$

In the proof, the Max-Flow Min-Cut Theorem 5.1 of [FF62] is used.

6.5 Comparing Power Infosys and Power Locale

Now, we show — as announced in the beginning — that the frames of opens of the power infosys and the power locale are isomorphic.

Let D be some fixed infosys. The frame F_1 is $\Omega \mathcal{P}_I D$, and the frame F_2 is $\Omega \mathcal{P}_L[D]$. It is defined in terms of $\Omega[D]$, which is isomorphic to ΩD. Thus, we arrive at the following presentations:

- Generators for F_1: $\sharp A$ for A in $\mathcal{P}_I D$.
- Relations:
  (1) Monotonicity: If $A < B$, then $\sharp A \geq \sharp B$;
  (2) All tokens: $\bigvee_{A \in \mathcal{P}_I D} \sharp A = \mathsf{T}$;

(3) Meets: $\sharp A \wedge \sharp B \leq \bigvee_{C > A,B} \sharp C$.

- Generators for F_2: $\langle u, q\rangle$ with u in ΩD and q in $\mathbb{QI}$,
- Relations:

(1) $\langle u, 0\rangle = \mathsf{T}$;

(2) $\bigvee_{r>q} \langle u, r\rangle = \langle u, q\rangle$;

(3) If $\mathcal{U} \subseteq \Omega X$ is directed, then $\langle \bigvee \mathcal{U}, q\rangle = \bigvee_{u \in \mathcal{U}} \langle u, q\rangle$;

(4) If $q \neq 0$, then $\langle 0, q\rangle = 0$;

(5) For all rational q with $0 \sqsubseteq q \sqsubset 2$, $\bigvee\{\langle u, r\rangle \wedge \langle v, s\rangle \mid r, s \in \mathbb{QI}, r + s = q\} = \bigvee\{\langle u \vee v, r\rangle \wedge \langle u \wedge v, s\rangle \mid r, s \in \mathbb{QI}, r + s = q\}$.

In view of the 'intended meaning' of the generators $\langle u, q\rangle$, we define $\varphi : F_2 \to F_1$ by

$$\varphi\langle u, q\rangle \;=\; \bigvee\{\sharp A \mid A_p u > q\}$$

In the summarized paper, we show that this definition preserves the relations of F_2. Thus, it extends to a frame homomorphism $\varphi : F_2 \to F_1$.

To establish a frame homomorphism $\psi : F_1 \to F_2$, we have to specify $\psi(\sharp A)$ for A in $\mathcal{P}_I D$. Here, we refer to Prop. 6.7, which says $A_o < \mu$ iff for all subbags S of A, $\sum \mathsf{num}\, S < \mu(\uparrow\!\mathsf{set}\, S)$. This motivates the following definition:

$$\psi(\sharp A) \;=\; \bigwedge_{S \subseteq A} \langle \uparrow\!\mathsf{set}\, S, \textstyle\sum \mathsf{num}\, S\rangle,$$

which involves a finite meet by finiteness of A. We have to show that ψ preserves the relations of F_1. Proving the preservation of the meets relation turns out to be difficult and is postponed until later. For the moment, we assume that it has been shown, so that $\psi : F_1 \to F_2$ is a well defined frame homomorphism.

The next step is to prove that φ and ψ are inverse to each other. Again, there is a difficult part in this proof which is postponed. Except for the two postponed statements, the proof of $F_1 \cong F_2$ is now completed.

Theorem 6.10 For every infosys D, the frames $\Omega\mathcal{P}_I D$ and $\Omega\mathcal{P}_L[D]$ are isomorphic. $\square$

6.6 Proof of the Critical Lemma

The two facts whose proofs were postponed reveal a common structure. Both may be derived from a more general *critical lemma*:

Lemma 6.11 For every infosys D, for every finite index set I and all families $(u_i)_{i \in I}$ in ΩD and $(q_i)_{i \in I}$ in $\mathbb{QI}$,

$$\bigwedge_{i \in I} \langle u_i, q_i\rangle \;\leq\; \bigvee\{\bigwedge_{S \subseteq C} \langle \uparrow\!\mathsf{set}\, S, \textstyle\sum \mathsf{num}\, S\rangle \mid C \in \mathcal{P}_I D,\, C_p u_i > q_i \;\forall i \in I\}$$

holds in $F_2 = \Omega\mathcal{P}_L[D]$. $\square$

Our proof of the critical lemma is quite complex. The very basic idea is taken from the proof of Lemma 5.3 in [Jon90] or Lemma 8.3 in [JP89], which state a vaguely similar property involving points and concrete open sets. The added difficulty in our proof is due to the fact that we have to work pointless, because the spatiality of the locale of F_2 is a priori unknown. (A posteriori, it follows from Theorem 6.10.)

The above mentioned proof of Jones employs so-called *crescents*, which are set differences of two concrete open sets. To mimic this, we set up a new frame F_3 with generators $\langle u, v, q\rangle$, whose intended spatial meaning is the set of all μ with $\mu(u \setminus v) > q$, or $\mu u - \mu(u \wedge v) > q$. With an appropriate choice of relations for F_3, the obvious assignment $\langle u, q\rangle \mapsto \langle u, 0, q\rangle$ becomes a frame homomorphism $\alpha : F_2 \to F_3$.

Let the critical lemma be $L \leq R$ in F_2. The added possibilities of F_3 allow proving $\alpha L \leq \alpha R$ in F_3 in vague analogy to Jones's proof of the Lemmas mentioned above. After having managed this, we only have to show that α is an order embedding; then $\alpha L \leq \alpha R$ in F_3 implies $L \leq R$ in F_2.

By the Corollary in paragraph II, 2.6 of [Joh82, page 53], F_2 can be embedded by a frame homomorphism η into a complete Boolean algebra G. Given η and G, we are able to define a (non-trivial) frame homomorphism $\beta : F_3 \to G$ with $\beta \circ \alpha = \eta$. Then α is an embedding, since η is an embedding.

The proof outlined above fills about 8 pages in the summarized paper. Its complexity could only be mastered by introducing some auxiliary notation, which is interesting on its own. From the generators $\langle u, q\rangle$ of F_2, we abstract the rational number q, writing $[u]$ for $\lambda q.\, \langle u, q\rangle$. With $0 = \lambda q.\, 0$, $1 = \lambda q.\, 1$, and

$$f + g \;=\; \lambda q.\, \Big(\bigvee_{r,s:\, r+s=q} fr \wedge gs\Big),$$

the relations of F_2 can be written as

- Bounded by one: $[u] \leq 1$;
- Continuity: for directed families $(u_i)_{i\in I}$, $[\bigvee_{i\in I} u_i] = \bigvee_{i\in I}[u_i]$;
- Zero law: $[0] = 0$;
- Modularity: $[u \vee v] + [u \wedge v] = [u] + [v]$.

Comparing with the description at the beginning of Subsection 6.5, this is a really notational benefit. Furthermore, the auxiliary notation admits the application of familiar algebraic laws; '+' is commutative and associative with neutral element 0.

The frame F_3 can be presented in a similar way by using abstracted generators $[u, v] = \lambda q.\, \langle u, v, q\rangle$ for every u, v in ΩD, whose intuitive meaning is the size or area of the 'region' $u \setminus v$, and relations

Bounded by one: $[u, v] \leq 1$;
Continuity: for directed families $(u_i)_{i\in I}$ and v with $v \leq u_i$ for all i, $[\bigvee_{i\in I} u_i, v] = \bigvee_{i\in I}[u_i, v]$;
Zero law: $[0, 0] = 0$;
Meet law: $[u, v] = [u, u \wedge v]$;
Join law: $[u, v] = [u \vee v, v]$;
Split law: if $u \geq v \geq w$, then $[u, w] = [u, v] + [v, w]$.

In the final part of the outlined proof, where the complete Boolean algebra G comes into play, the auxiliary notation is augmented by subtraction

Definition 6.12 For f, g with $f \geq g$, let

$$f - g \;=\; \lambda q. \begin{cases} \mathsf{T} & \text{if } q = 0 \\ \bigvee_{s\in\mathbb{Q}_0^+} f(q+s) \wedge \neg gs & \text{if } q \neq 0 \end{cases}$$

using negation which is available in G. This subtraction has familiar algebraic properties such as $(f + g) - g = f$. It is used to define the frame homomorphism from F_3 to G by $\beta[u, v] = [u] - [u \wedge v]$ where the embedding $\eta : F_2 \to G$ is notationally suppressed.

References

[Abr91] S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92:161–218, 1991.

[Ama91] R. Amadio. Bifinite domains: Stable case. In D.H. Pitt, P.-L. Curien, S. Abramsky, A.M. Pitts, A. Poigné, and D.E. Rydeheard, editors, *Category Theory and Computer Science*, pages 16–33. LNCS 530, Springer-Verlag, 1991.

[Ber78] G. Berry. Stable models of typed lambda-calculi. In *Proc. 5th Coll. on Automata, Languages, and Programming*, pages 72–89. LNCS 62, Springer-Verlag, 1978.

[Bro82] M. Broy. A fixed point approach to applicative multiprogramming. In M. Broy and G. Schmidt, editors, *Theoretical Foundations of Programming Methodology.* Reidel, 1982.

[Bun88] P. Buneman. Lossless powerdomains. Oral presentations and private communications, 1988.

[CC84] W.F. Clocksin and C.S.Mellish. *Programming in Prolog.* Springer-Verlag, 1984.

[FF62] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[Gra88] S.K. Graham. Closure properties of a probabilistic domain construction. In Michael G. Main, A. Melton, Michael Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Language Semantics (MFPLS '87)*, pages 213–233. LNCS 298, Springer-Verlag, 1988.

[Hec90] R. Heckmann. *Power Domain Constructions.* PhD thesis, Universität des Saarlandes, 1990.

[Hec91] R. Heckmann. Power domain constructions. *Science of Computer Programming*, 17:77–117, 1991.

[Hec92] R. Heckmann. Power domains supporting recursion and failure. In J.-C. Raoult, editor, *CAAP '92*, pages 165–181. LNCS 581, Springer-Verlag, 1992.

[Hec93a] R. Heckmann. Power domains and second order predicates. *Theoretical Computer Science*, 111:59–88, 1993.

[Hec93b] R. Heckmann. Product operations in strong monads. In G.L. Burn, S.J. Gay, and M.D. Ryan, editors, *Proceedings of the First Imperial College, Department of Computing, Workshop on Theory and Formal Methods*, Workshops in Computing, pages 159–170. Springer-Verlag, 1993.

[Hec94a] R. Heckmann. Lower bag domains. *Fundamenta Informaticae*, ?, 1994. Accepted for publication.

[Hec94b] R. Heckmann. Probabilistic domains. In S. Tison, editor, *CAAP '94*, pages 142–156. LNCS 787, Springer-Verlag, 1994.

[Hec94c] R. Heckmann. Probabilistic power domains, information systems, and locales. In S.D. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *MFPS '93.* LNCS ?, Springer-Verlag, 1994. To appear.

[Hec94d] R. Heckmann. Stable power domains. *Theoretical Computer Science*, ?, 1994. Accepted for publication.

[Hoo87] R. Hoofman. Powerdomains. Technical Report RUU-CS-87-23, Rijksuniversiteit Utrecht, November 1987.

[Hoo90] R. Hoofman. Continuous information systems. Technical Report RUU-CS-90-25, Rijksuniversiteit Utrecht, July 1990. To appear in Information and Computation.

[Hoo92] R. Hoofman. *Non-Stable Models of Linear Logic.* PhD thesis, Rijksuniversiteit Utrecht, 1992.

[HP79] M.C.B Hennessy and G.D. Plotkin. Full abstraction for a simple parallel programming language. In J. Becvar, editor, *Foundations of Computer Science*, pages 108–120. LNCS 74, Springer-Verlag, 1979.

[JA94] A. Jung and S. Abramsky. Domain theory. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, Vol. III.* Oxford University Press, 1994. To appear.

[Jag89] R. Jagadeesan. L-domains and lossless powerdomains. In Michael G. Main, A. Melton, Michael Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Semantics (MFPS '89)*, pages 364–372. LNCS 442, Springer-Verlag, 1989.

[Joh82] P. Johnstone. *Stone Spaces.* Cambridge University Press, 1982.

[Joh92] P. Johnstone. Partial products and bagdomains. In M. Fourman, P. Johnstone, and A. Pitts, editors, *Application of Categories in Computer Science*, pages 315–339. London Mathematical Society Lecture Notes 177, Cambridge University Press, 1992.

[Jon90] C.J. Jones. *Probabilistic Non-Determinism.* PhD thesis, University of Edinburgh, 1990.

[JP89] C.J. Jones and G.D. Plotkin. A probabilistic powerdomain of evaluations. In *LICS '89*, pages 186–195. IEEE Computer Society Press, 1989.

[Jun90] A. Jung. Cartesian closed categories of algebraic cpo's. *Theoretical Computer Science*, 70:233–250, 1990.

[LW93] L. Libkin and L. Wong. Query languages for bags. Internal Report MS-CIS-93-36, Logic & Computation 59, University of Pennsylvania, March 1993.

[Mai85] M.G. Main. Free constructions of powerdomains. In A. Melton, editor, *Mathematical Foundations of Programming Semantics*, pages 162–183. LNCS 239, Springer-Verlag, 1985.

[Mog89] E. Moggi. Computational lambda-calculus and monads. In *4th LICS Conference*, pages 14–23. IEEE, 1989.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[Plo76] G.D. Plotkin. A powerdomain construction. *SIAM Journal on Computing*, 5(3):452–487, 1976.

[Sco82] D. Scott. Domains for denotational semantics. In *Proc. 9th Coll. on Automata, Languages, and Programming*, pages 577–613. LNCS 140, Springer-Verlag, 1982.

[Smy78] M.B. Smyth. Power domains. *Journal of Computer and System Sciences*, 16:23–36, 1978.

[Vic89] S. Vickers. *Topology via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1989.

[Vic92] S. Vickers. Geometric theories and databases. In M. Fourman, P. Johnstone, and A. Pitts, editors, *Application of Categories in Computer Science*, pages 288–314. London Mathematical Society Lecture Notes 177, Cambridge University Press, 1992.

[Vic93] S. Vickers. Information systems for continuous posets. *Theoretical Computer Science*, 116(2):201–229, 1993.