POWERDOMAINS

R. Hoofman

RUU-CS-87-23
November 1987

Rijksuniversiteit Utrecht

Vakgroep informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

POWERDOMAINS

R. Hoofman

Technical Report RUU-CS-87-23
November 1987

Department of Computer Science
University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht
The Netherlands

Abstract

This paper gives a constructive introduction to powerdomains. Powerdomains are some sort of mathematical structures used to describe the semantics of computer programs. They arise out of the combination of order-theoretic semantics and nondeterminism. In chapter 1 these and other concepts are clarified.

The next chapter gives the construction of some restricted class of powerdomains. The powerdomains are defined in an abstract way in chapter 3, and with the help of some category-theory their existence is proved. Chapter 4 states some results about the form of powerdomains (for example, it turns out that each powerdomain is a certain completion of a free nondeterministic poset), using the new approach of the previous chapter.

We only consider bounded powerdomains, except for chapter 5, where we give a countable, lower powerdomain.

Contents

1 **2**

1.1 Order-theoretic semantics 2

1.2 Nondeterminism 5

1.3 Category-theory 8

2 **12**

2.1 The Smyth-approach 12

2.2 Some properties of powerdomains 17

3 **23**

3.1 Powerdomains are continuous algebras 23

4 **29**

4.1 Powerdomains over restricted classes of dcpo's 29

4.2 Powerdomains over arbitrary dcpo's 34

5 **42**

5.1 A (lower) powerdomain for countable nondeterminism 42

Chapter 1

1.1 Order-theoretic semantics

Order-theoretic semantics interpret syntactic (formal) objects as elements of a set with some ordering structure. First we review some of the basic concepts.

Definition 1.1.1 *A* partially ordered set (poset) *is a pair* $(S, \leq)$*, where* S *is a set and* $\leq$ *is a binary relation on the elements of* S*, such that* $\forall x, y, z \in S$*:*

1. $x \leq x$
2. $x \leq y \wedge y \leq x \Rightarrow x = y$
3. $x \leq y \wedge y \leq z \Rightarrow x \leq z$

Definition 1.1.2 *Let* $(S, \leq)$ *be a poset,* $S' \subseteq S$ *and* $x \in S$*, then*

1. x *is a* minimal (maximal) *element of* S' *iff* $x \in S'$ *and for each* $y \in S'$*:* $y \leq x$ $(x \leq y)$ *implies* $x = y$*.*
2. x *is a* lower (upper) bound *of* S' *iff* $x \leq y$ $(y \leq x)$ *for all* $y \in S'$*.*
3. x *is a* least (greatest) *element iff* $x \in S'$ *and* x *is a lower (upper) bound of* S'*.*
4. x *is the* greatest lower (least upper) bound *of* S' *iff* x *is the greatest (least) element of the set of all lower (upper) bounds of* S'*.*

The least upper bound (lub) of a set S is denoted by $\sqcup S$, and the greatest lower bound (glb) by $\sqcap S$.

There is a special kind of functions which preserves the order structure of posets.

Definition 1.1.3 *A function* f*:* $(S, \leq) \to (R, \leq')$ *between posets is* monotone *iff for all* $x, y \in S$*:*$x \leq y$ *implies that* $f(x) \leq' f(y)$*.*

The behaviour of a computer program can be represented by a sequence of partial results. We can take the set of all partial results as our domain of interpretation, and define an ordering $\leq$ on this set such that sequences describing the behaviour of programs are increasing with respect to $\leq$.

Example 1.1.1 *Let P_1:="For i:=1 to n do print('$\star$')" be a program, then we have a sequence of partial results $[\star, \star\star, \star\star\star, \ldots, \star\ldots\star]$, where the last string is the final result and consists of n stars. Take $S_1 = \{\overline{x} | \overline{x}_i = \star \wedge |\overline{x}| < \infty\}$ the set of all finite strings of $\star$, and order them by length.*

Example 1.1.2 *Let P_2:="While true do print('$\star$')" be a program. Now take $S_1 = \{\overline{x} | \overline{x}_i = \star \wedge |\overline{x}| \leq \infty\}$ ordered by length. We can say that the infinite string $\star\star\ldots$ is the end result of P_2.*

We see that end results are lubs of chains consisting of partial results. It is therefore natural to take those posets as semantic domains which have lubs of all chains.

Definition 1.1.4 *A* chain *C in a poset $(S, \leq)$ is a countable subset $C \subseteq S$, such that there is a denumeration $c_0, c_1, c_2, \ldots$ of the elements of C, which satifies $\forall i : c_i \leq c_{i+1}$.*

Definition 1.1.5 *A* complete partially ordered set (cpo) *is a poset which has lubs of all its chains.*

The poset S_2 of example 2 is, and the poset S_1 of example 1 is nòt a cpo. Often instead of cpo's so-called *dcpo's* are used, because these structures are more easy to handle mathematically.

Definition 1.1.6 *Let $(S, \leq)$ be a poset, then $S' \subseteq S$ is* directed *iff it is not empty, and $\forall x, y \in S' \exists r \in S'$ such that $x \leq r \wedge y \leq r$.*

Equivalently we could say that every finite subset of a directed set S has an upperbound in S.

It is easy to see that every chain is a directed set. In fact the concept of *directed set* is a kind of generalization of the concept *chain*. We can generalize cpo's to dcpo's in the same manner.

Definition 1.1.7 *A* directed complete partially ordered set (dcpo) *is a poset which has lubs of all its directed subsets.*

Clearly every dcpo is a cpo. The following theorem shows that the difference between cpo's and dcpo's is essentially one of cardinality.

Theorem 1.1.1 *A poset is a cpo if and only if it has all lubs of countable directed sets.*

Proof: See [21].

□

Definition 1.1.8 *A function f:$(S, \leq) \rightarrow (R, \leq)$ between posets is* continuous *iff it is monotone and it preserves lubs of directed subsets. That is, whenever $S' \subseteq S$ is directed and $\sqcup S'$ exists, it follows that $f(\sqcup S') = \sqcup\{f(x) | x \in S'\}$.*

Note that functions between arbitrary posets can be continuous, this in contrast with the more usual definition of continuous functions between dcpo's.

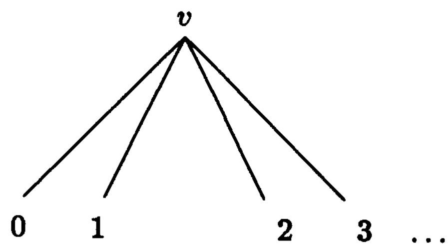
An important application of order-theoretic semantics is the attaching of meaning to recursive specifications of functions([17]). For this sort of applications we need dcpo's which have a least element $\perp$. Another important issue are the so-called domain-equations, where dcpo's are specified in terms of the result of operators applied to other dcpo's([21]).

1.2 Nondeterminism

Nondeterminism is an ever recurring concept in computer science. For example in the area of formal automaton theory and concurrency.

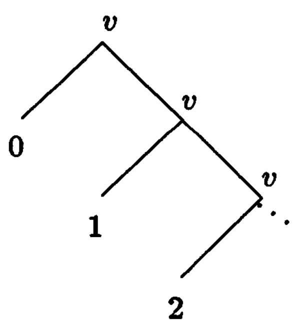
In general a distinction is made between *bounded* and *unbounded* nondeterminism. In the former case there is at every moment (point) a choice between only a finite number of alternatives, whereas in the latter case there may be an infinite number of alternatives. Unbounded nondeterminism can be further distinguished in *countable* and *uncountable* nondeterminism.

Example 1.2.1 *Let P_1:="x:=?;print(x)" be a nondeterministic program, where "x:=?" is a nondeterministic assignement which assigns to x an arbitrary integer. Clearly P_1 is* unbounded *nondeterministic, and may be represented by the following diagram:*



where v is a nondeterministic choice node.

Example 1.2.2 *Let P_2:="x:=-1; Repeat x:=x+1; s:=(true or false) until s; print(x)" be a nondeterministc program, where* or *is a nondeterministic choice operator which yields one of its arguments. Although the result of P_2 can be any integer (just as with P_1) , P_2 is a* bounded *nondeterministic program:*



So bounded or unbounded nondeterminism is strongly related to the level at which one chooses his primitives.

We can give a function semantics to programs, i.e. we interpret each program P as a function $g_P : S \to R$, where S is a set of input values and R is a set of output values. For a value $s \in S$ g_P yields the result of P when given input s.

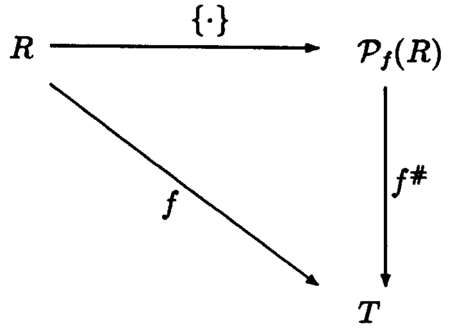
In the same manner nondeterministic programs can be represented as functions $h_P : S \to \mathcal{P}(R)$, i.e. functions from the set of input values to the powerset of the set of output values. Here h_P yields the set of all possible results for a certain input $s \in S$. When we consider bounded nondeterminism only, we restrict our attention to functions $h_P : S \to \mathcal{P}_f(R)$, with $\mathcal{P}_f(R)$ the set of *finite* subsets of R.

Nondeterminism arises out of the existence of nondeterministic choice in programs, which we model by a choice operator *or*. This operator takes a set as argument and yields nondeterministically one of its elements. In the case of bounded nondeterminism this set has to be finite, so we might as well take a binary *or*-operator which yields one of its arguments.

We interpret the nondeterministic choice operator *or* in the semantic domain $\mathcal{P}_f(R)$ as set-union ($\cup$), so for example if h_Q is the function semantics of the nondeterministic program Q and h_P that of P, then $\lambda x.h_P(x) \cup h_Q(x)$ is the function semantics of the program P *or* Q.

$(\mathcal{P}_f(R), \cup)$ is in fact a very special structure, for it is the free (bounded) nondeterministic set generated by R. A (bounded) nondeterministic set $(T, +)$ is a set together with a binary, commutative, associative and idempotent operator $+$. This $+$ is the interpretation of *or* when we take T as a semantic domain. For example P *or* Q" would be interpreted as $\lambda x.h_P(x) + h_Q(x)$. It is easy to see that $(\mathcal{P}_f(R), \cup)$ is a nondeterministic set.

$\mathcal{P}_f(R)$ is *free* for it has the following property: Given a function $f : R \to T$, with T a nondeterministic set, there is a unique $f^\# : \mathcal{P}_f(R) \to T$ such that the following diagram commutes:



with $\{\cdot\} : x \mapsto \{x\}$.

So in a certain sense $\mathcal{P}_f(R)$ is the unique way of making R in a nondeterministic set.

In this paper we will try to do the same with dcpo's, i.e. to find the free nondeterministic dcpo generated by a certain dcpo. These free nondeterministic dcpo's are called *powerdomains* and they are a combination of order-theoretic semantics and nondeterminism.

Example 1.2.3 *Let $S_\perp$ be the* flat poset *over the set S, i.e. $S_\perp = S \cup \{\perp\}$ and for $x, y \in S_\perp$ we have $x \leq y$ implies $x = y$ or $x = \perp$. Clearly $S_\perp$ is a dcpo. The free nondeterministic dcpo generated by $S_\perp$ is the set $\{S' | S' \subseteq S \wedge (\perp \in S' \vee S$ finite$\}$ together with an ordering: $S'_1 \leq S'_2$ implies $S'_1 = S'_2$ if $\perp \notin S'_1$ and $S'_1 - \{\perp\} \subseteq S'_2$*

if $\perp \in S_2'$, *and with set-union as* $+$*-operator.*

Powerdomains are used in a lot of different areas, such as semantics of concurrent programming languages ([4,7,9,10,11]), denotational semantics of non-deterministic program schemes([5,17,18]), and in connection with so-called *port automatons*([14]).

1.3 Category-theory

In this section we give a short introduction to some concepts of category-theory we will need later on. Proofs of theorems will be omitted, but may be found for example in [13].

To avoid some set theoretical paradoxes (such as the set of all sets not members of themselves) we will work inside a universe $\mathcal{V}$, i.e. a set of sets which is closed under certain standard operators such as powerset, cartesian product, etc. A set which is a member of $\mathcal{V}$ is called *small*. In general all sets we use in this paper will be small, so this predicate will often be omitted.

The first definition is that of a category.

Definition 1.3.1 *A* category *C is given by data (1),(2),(3) subject to axioms (a),(b),(c) as follows:*

1. *A class $Ob(C)$ of C-objects $X, Y, Z, \ldots$*
2. *For each ordered pair of objects (X, Y) a set $C(X, Y)$ of C-arrows from X to Y.*

a. *If $C(X,Y) \cap C(\bar{X}, \bar{Y}) \neq \emptyset$, then $X = \bar{X}$ and $Y = \bar{Y}$.*

3. *A composition operator $\circ$ assigning to each $f \in C(X,Y)$ and $g \in C(Y,Z)$ a third arrow $f \circ g \in C(X,Z)$.*

b. *Composition is assiociative.*

c. *For each object X there exists an identity arrow $id_X \in C(X,X)$ such that for each $f \in C(Y,X) : id_X \circ f = f$, and for each $g \in C(X,Z) : g \circ id_X = g$.*

Example 1.3.1 *Set is the category with as objects all (small) sets, and as arrows all functions between these sets. Set^+ is the category with as objects all (small) nondeterministic sets (i.e. sets with a binary, associative, commutative and idempotent operator $+$), and as arrows all linear functions (i.e. functions f such that $f(x + y) = f(x) + f(y)$).*

We can define morphisms between categories as follows.

Definition 1.3.2 *A* functor *$F : C \to D$ between categories C and D, assigns to each object X in C an object FX in D, and to each arrow $f \in C(X,Y)$ an arrow $Ff \in D(FX, FY)$, in such a way that $F(id_X) = id_{FX}$ and $F(g \circ f) = Fg \circ Ff$.*

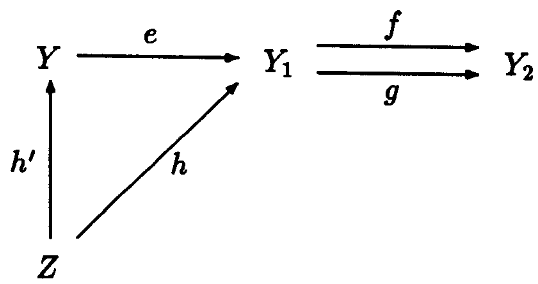
A *forgetful* functor is a functor which simply "forgets" some or all of the structure of an algebraic object.

Example 1.3.2 *Define a functor $U : Set^+ \to Set$ on objects as $U(< S, + >) = S$, and on arrows as $Ff = f$, then U forgets the +-operator and is therefore a forgetful functor.*

There are some constructions on objects and arrows in a category.

Definition 1.3.3 *The* product *of objects* Y_1, Y_2 *in a category* C *consists of an object* $Y_1 \prod Y_2$ *and arrows* $\pi_1 \in C(Y_1 \prod Y_2, Y_1), \pi_2 \in C(Y_1 \prod Y_2, Y_2)$*, such that for every other object* Z *and arrows* $f_i \in C(Z, Y_i)$*,* $i \in \{1,2\}$*, there is a unique* $f \in C(Z, Y_1 \prod Y_2)$ *with* $\pi_i \circ f = f_i$*.*

Definition 1.3.4 *An* equalizer *of two arrows* $f, g \in C(Y_1, Y_2)$ *is an object* Y *of* C *and an arrow* $e \in C(Y, Y_1)$ *such that* $f \circ e = g \circ e$*, and for every other object* Z *and arrow* $h \in C(Z, Y_1)$ *with* $f \circ h = g \circ h$ *there is a unique* $h' \in C(Z, Y)$ *such that* $e \circ h' = h$*.*

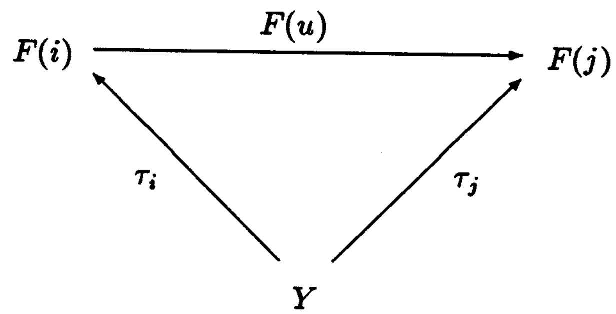


Example 1.3.3 *In Set the product of two sets is the usual cartesian product, while the equalizer of two functions* $f, g : Y_1 \to Y_2$ *is the object* $Y = \{a \in Y_1 | f(a) = g(a)\}$ *together with the inclusion arrow* $e : Y \to Y_1$*.*

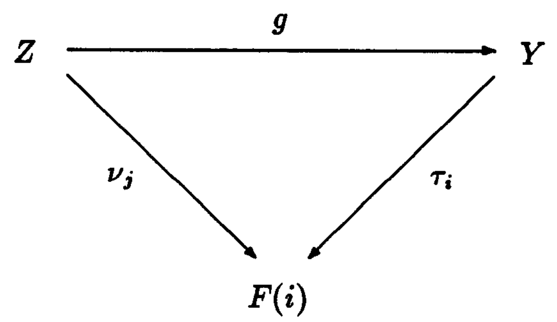
Definition 1.3.3 can easily be generalized to the product of any (small) set of objects.

Products and equalizers are instances of the more general concept of a *limit.*

Definition 1.3.5 *Let* $F : J \to C$ *be a functor. A* cone *from an object* Y *in* C *to* F *is a Ob(J)-indexed family* τ *of arrows* $\tau_i \in C(Y, Fi)$*, such that for every arrow* $u \in J(i, j)$ *the following diagram commutes:*



A limit *of* F *is a cone* (Y, τ) *such that for every other cone* (Z, ν) *there is a unique arrow* $g \in C(Z, Y)$ *such that the following diagram commutes for every object* i *in* J*.*



If we take $\cdot \rightrightarrows \cdot$ for J then limits of functors $F : J \to C$ are equalizers in C. Reversing all arrows in the previous definition gives a definition of the so-called *colimits.*

A category is small if both the set of its objects and the set containing all its arrows are small.

Definition 1.3.6 *A category C is called* small-complete *if every functor $F : J \to C$ from a small category J to a category C has a limit.*

Theorem 1.3.1 *If C has all small products (= product of a small set of objects) and equalizers of all pairs of arrows, then C is small-complete.*

Example 1.3.4 *By example 1.3.3 and theorem 1.3.1 it follows that Set is small-complete.*

Definition 1.3.7 *A functor $H : C \to D$ is said to* preserve the limit *of a functor $F : J \to C$ when every limit (Y, τ) of F in C yields by composition with H a limit $(HY, H\tau)$ of $HF : J \to D$ in D.*

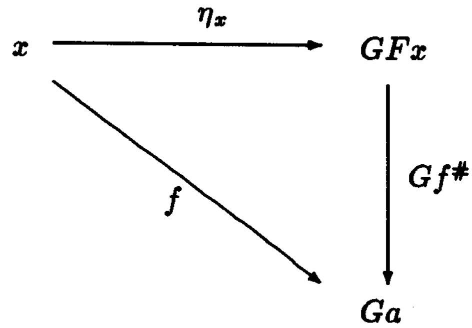
Theorem 1.3.2 *If C is a small-complete category, and $H : C \to D$ preserves all small products and all equalizers, then H preserves all small limits.*

We now give the following important definition of an *adjunction.*

Definition 1.3.8 *An* adjunction $< G, F, \eta >: X \rightharpoonup A$ *is given by the following data:*

1. *A functor $G : A \to X$.*
2. *For every object $x \in X$ an object $Fx \in A$.*
3. *An X-indexed family of arrows η in X, such that $\eta_x \in X(x, GFx)$.*

subject to the requirement that for each $f \in X(x, Ga)$, with a an object of A, there exists a unique $f^\# \in A(Fx, a)$ such that $Gf^\# \circ \eta_x = f$. So the following diagram *commutes:*



The map F of an adjunction can be extended to a functor $X \to A$ (denoted by the same symbol F), if we define $Ff = (\eta_y \circ f)^{\#}$, for arrows $f \in X(x,y)$. This functor F is called the *left-adjoint* of G, while G is the *right-adjoint* of F. The set η is the *unit* of the adjunction. The subscript x of η_x will often be omitted.

In this paper adjunctions will be used to characterize free structures. In general *a left-adjoint to a forgetful functor maps objects to free structures.*

Example 1.3.5 *Define a map $\mathcal{P}_f : Ob(Set) \to Ob(Set^+)$ by $\mathcal{P}_f(S) := < \{S' | S' \subseteq S \wedge S \text{ finite}\}, \cup >$, i.e. $\mathcal{P}_f$ maps each set to the set of all its finite subsets, while the +-operator is set-union($\cup$). Now $< U, \mathcal{P}_f, \{\cdot\} >: Set \rightharpoonup Set^+$ is an adjunction, with $\{\cdot\} : t \mapsto \{t\}$, and U the functor of example 1.3.2. So the functor $\mathcal{P}_f$ (defined on arrows as $\mathcal{P}_f(g)(S') = \{g(x) | x \in S'\}$, for $g : S \to R$ and $S' \subseteq S$) is a left-adjoint of U.*

A few results about adjunctions:

Theorem 1.3.3 *Any two left-adjoints F and F' of a functor $G : A \to X$ are isomorphic, i.e. $Fx \cong F'x$ in A.*

Theorem 1.3.4 *Given two adjunction $< F, G, \eta >: X \rightharpoonup A$ and $< \bar{F}, \bar{G}, \bar{\eta} >: X \rightharpoonup A$ the composite functors $\bar{F}F$ and $G\bar{G}$ yield an adjunction.*

The following theorem states when a given functor has a left-adjoint.

Theorem 1.3.5 (Freyd Adjoint Functor Theorem) *Given a small-complete category A, a functor $G : X \to A$ has a left-adjoint if and only if it preserves all small limits and it satisfies the following*
Solution Set Condition*: For each object $x \in X$ there is a small set I and an I-indexed family of arrows $F_i : x \to Ga_i$ such that every arrow $h : x \to Ga$ can be written as a composite $h = Gt \circ f_i$ for some index i and some $t : a_i \to a$.*

Chapter 2

2.1 The Smyth-approach

In [20] Smyth gave a construction of powerdomains over *ω-algebraic* dcpo's with least elements. In fact he defined two kinds of powerdomains: the *upper* powerdomain $P_U(D)$ and the *convex* powerdomain $P_C(D)$ over a dcpo D. More recently a third powerdomain is introduced (for example in [22]), the *lower* powerdomain $P_L(D)$.

In this section we state some of Smyth's results and give proofs for the lower powerdomain. At the end we will have two different (but isomorphic) representations of the powerdomains over ω-algebraic dcpo's.

Definition 2.1.1 *An element d of a dcpo (cpo) D is* compact *provided that, for every directed $S \subseteq D$ (chain S), if $d \leq \sqcup S$, then $d \leq s$ for some $s \in S$.*

Let D_0 be the set of compact elements of D.

Definition 2.1.2 *A dcpo (cpo) D is* algebraic *iff each element $d \in D$ is the lub of a directed subset of D_0 (of a chain in D_0).*

So every element of an algebraic dcpo D can be "approximated" by a set of compact elements.

Definition 2.1.3 *A dcpo (cpo) D is* ω-algebraic *iff it is algebraic and D_0 is a countable set.*

In this chapter dcpo's (cpo's) will be ω-algebraic and contain a least element.

Just as deterministic programs can be represented by an increasing chain of intermediate results, and the final result is the lub of this chain, so the behaviour of a bounded nondeterministic program could be depicted by a finitary tree, where the nodes are marked with intermediate results. The branching at a node stands for a nondeterministic choice, and every path in the tree is an increasing chain. So we introduce the concept of a generating tree.

Definition 2.1.4 *Let D be a dcpo. A* generating tree *over D is a node-labeled finitary tree T satisfying:*

1. for each node t the label $l(t) \in D$,

2. if t' is a descendant of t in T, then $l(t) \leq l(t')$,

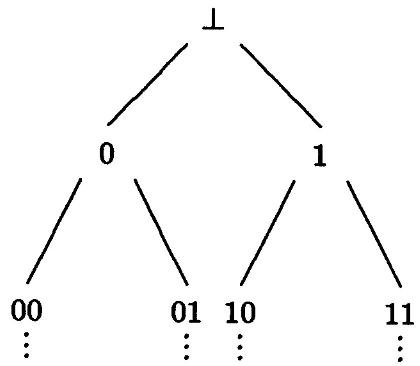
3. T has no terminating branches.

A *path* in a tree T is a sequence of nodes beginning in the root, and such that if t' follows t, then t' is a descendant of t in T.

Definition 2.1.5 *A set S* is generated *by a tree T iff* $S = \{\sqcup\pi | \pi$ *is a path in* $T\}$. *A set* $S \subseteq D$ *is* finitely generable (f.g.) *iff there is a generating tree T, which generates S.*

An f.g. set is the final result of a computation represented by a generating tree.

Example 2.1.1 *Let P:="While true do print(0 or 1)" be a program, then T(P), the generating tree representing P, is as follows:*



The set S generated by T(P) will contain for example 1^ω, *the string consisting of infinitely many one's (this is the lub of the rightmost path in T(P)).*

A generating tree is called *strict* if all the node labels are in D_0. The class of f.g. sets generated by *all* generating trees is the same as the class of f.g. sets generated by the all strict generating trees (see [20]).

Let F(D) be the set of all sets generated by (strict) generating trees, i.e. the set of all end results of nondeterministic computations. We will define three different orderings on F(D). It will turn out that the the set of equivalence classes of the equivalence relation induced by such an order is a powerdomain.

First we define three relations $\mathcal{P}_f(D_0) \times \mathcal{F}(D)$.

Definition 2.1.6 *Let* $A \in \mathcal{P}_f(D_0)$ *and* $S \in \mathcal{F}(D)$, *then*

$$A \sqsubseteq_L S := \forall a \in A \exists s \in S : a \leq s$$

$$A \sqsubseteq_U S := \forall s \in S \exists a \in A : a \leq s$$

$$A \sqsubseteq_C S := A \sqsubseteq_L S \wedge A \sqsubseteq_U S$$

The relation $\sqsubseteq_C$ is sometimes called the *Egli-Milner ordering.*

With the help of these relations we define an order on $\mathcal{F}(D)$.

Definition 2.1.7 *Let $S, S' \in \mathcal{F}(D)$ then*

$$S \preceq_L S' := \forall A \in \mathcal{P}_f(D_0), A \sqsubseteq_L S \Rightarrow A \sqsubseteq_L S'$$
$$S \preceq_U S' := \forall A \in \mathcal{P}_f(D_0), A \sqsubseteq_U S \Rightarrow A \sqsubseteq_U S'$$
$$S \preceq_C S' := \forall A \in \mathcal{P}_f(D_0), A \sqsubseteq_C S \Rightarrow A \sqsubseteq_C S'$$

The elements of $\mathcal{P}_f(D_0)$ are cross-sections of generating trees over D. So they are in a certain sense approximations of end results of programs, i.e. of f.g. sets. Definition 2.1.7 states that for $S, S' \in \mathcal{F}(D)$ S' is greater as S if every approximant of S approximates S' too. The three different kinds of ordering reflects a difference in the philosophy about what information the elements of $\mathcal{P}_f(D_0)$ should give us about the final result of a computation.

Definition 2.1.8 *A* pre-cpo *is a structure $(S, \leq)$ wich satisfies all the requirements of a cpo, except that the relation $\leq$ is not anti-symmetric.*

We are going to prove that the structures $(\mathcal{F}(D), \preceq)$ are ω-algebraic pre-cpo's (if a statement is true of all three orderings we will omit the subscript). Because the orderings $\preceq$ are not anti-symmetric we have pre-cpo's instead of cpo's. Quotienting the preorder $(\mathcal{F}(D), \preceq)$ by the equivalence $\preceq \cap \preceq^{-1}$ will give us a cpo.

The following lemmas and theorems are true for all three orderings, but we wil give proofs only for $\preceq_L$.For $\preceq_U$ and $\preceq_C$ proofs may be found in [20].

Lemma 2.1.1 *Suppose S is generated by a tree T,$A \in \mathcal{P}_f(D_0)$ and $A \sqsubseteq S$, then $A \sqsubseteq T_m$ for some cross-section T_m of T.*

Proof: It is known that $A \sqsubseteq_L S$, so $\forall a \in A \exists s \in S : a \leq s$. Now the elements of A are compact, and those of S are lubs of paths in T. So for all $a \in A$ there are nodes t_a in T such that $a \leq l(t_a)$. Let each such t_a lie in cross-section T_a, then there is a cross-section T_m which is lower in T then all T_a (for A is finite). T_m contains for each t_a an t such that $l(t_a) \leq l(t)$ because $T_a \sqsubseteq_L T_m$. So $a \leq l(t_a) \leq l(t)$ and $A \sqsubseteq_L T_m$.

□

Lemma 2.1.2 *If S is generated by tree T then S is the lub of the set of cross-sections of T with respect to the order $\preceq$.*

Proof: $T_n \preceq S$ is trivial. Suppose $\forall n : T_n \preceq Y$. We have to prove $\forall A \in \mathcal{P}_f(D_0)$: $A \sqsubseteq S \Rightarrow A \sqsubseteq Y$. Suppose $A \sqsubseteq S$ then by the previous lemma $A \sqsubseteq T_m$, so $A \sqsubseteq Y$.

□

Lemma 2.1.3 *Every increasing chain with elements of $\mathcal{P}_f(D_0)$ has a f.g. set as lub.*

Proof: Given the chain $(A_i|i \in N)$ ordered by $\sqsubseteq_L$ construct a tree T as follows: Label the root with $\perp$. If v is a node at depth n labeled with $b \in D$, then take as successors of v (if any) one node for each $c \in A_{n+1}$ with $b \leq c$. Add a node at level n with label $\perp$, and take as successors of this node one node for each label in A_{n+1} which is not smaller then any label in A_n. This node $\perp$ will also has as successor the node with label $\perp$ which is just in the same manner added to A_{n+1}. Now T is a finitary tree with infinite paths, while the cross-sections are the sets $A_i \cup \{\perp\}$.

We will prove that $\sqcup(A_i) = S$, with S the set generated by T:

- S is an upperbound of (A_i).
  For $A_i \sqsubseteq_L A_i \cup \{\perp\} = T_i \sqsubseteq_L S$.
- S is the least upperbound of (A_i).
  Suppose $\forall A_i : A_i \sqsubseteq_L Y$. Then if $A \sqsubseteq_L S$, $A \sqsubseteq_L T_i \sqsubseteq_L A_i \sqsubseteq_L Y$. So $S \sqsubseteq_L Y$.

□

Lemma 2.1.4 *Let $A \in \mathcal{P}_f(D_0)$ and $(A_i|i \in N)$ a chain in $\mathcal{P}_f(D_0)$. If $A \sqsubseteq \sqcup(A_i|i \in N)$ then $A \sqsubseteq A_n$ for a certain $n \in N$.*

Proof: There is a f.g. set S such that $S = \cup(A_i|i \in N)$. Suppose T is the tree which has S as lub (built as in the previous lemma), then with lemma 2.1.1 $A \sqsubseteq_L T_i = A_i \cup \{\perp\}$, so $A \sqsubseteq_L A_i$.

□

Theorem 2.1.1 *$(\mathcal{F}(D), \preceq)$ is an ω-algebraic pre-cpo with $\mathcal{P}_f(D_0)$ as set of compact elements.*

Proof: For $x \in \mathcal{F}(D)$ define $B(x) := \{f \in \mathcal{P}_f(D_0)|f \sqsubseteq x\}$. Now:

(a) The least element is $\{\perp\}$.

(b) Suppose (A_i) is a chain in $\mathcal{F}(D)$, then $A = \cup\{B(A_i)|i \in N\}$ is a countable directed set with the same lub as (A_i). Because of the countability and by lemma 2.1.3 A has a lub in $\mathcal{F}(D)$.

(c) Suppose (A_i) is a chain in $\mathcal{F}(D)$, and $f \sqsubseteq \sqcup(A_i)$ with $f \in \mathcal{P}_f(D_0)$. Then $f \sqsubseteq \sqcup A$ by the previous point. So with lemma 2.1.4 we have $\exists f' \in A$: $f \sqsubseteq f'$. Now $\exists A_n \in (A_i)$ such that $f' \in B(A_n)$, so $f \sqsubseteq f' \sqsubseteq A_n$.

(d) $\mathcal{P}_f(D_0)$ is countable.

(e) By lemma 2.1.2 it follows that each element of $\mathcal{F}(D)$ is the lub of a chain with elements of $\mathcal{P}_f(D_0)$.

□

Corollary 2.1.1 *$(\mathcal{F}(D)/\preceq, \preceq / \preceq)$ is an ω-algebraic cpo.*

Every ω-algebraic cpo is an ω-algebraic dcpo (see [21]).

Corollary 2.1.2 *$(\mathcal{F}(D)/\preceq, \preceq / \preceq)$ is an ω-algebraic dcpo.*

Although the powerdomain construction given above is very plausible intuitively , we will give an other equivalent construction which is more easily to handle mathematically. The elements of this powerdomain representation will be *left-closed, directed subsets of $\mathcal{P}_f(D_0)$*, while the ordering is just ordinary set-inclusion. First we will introduce some terminology.

Definition 2.1.9 *Let $S \subseteq P$, with P a poset. S is* left-closed *if $a \leq b \in S$ implies $a \in S$.*

Definition 2.1.10 *An* ideal *over a poset P is a left-closed, directed subset of P. For $a \in P$ the* principal ideal (generated by a) *I_a is the set $\{a' \in P | a' \leq a\}$.*

We represent each element S of $\mathcal{F}(D)$ by the set containing those elements of $\mathcal{P}_f(D_0)$ which are smaller than S with respect to the ordering $\sqsubseteq$. In fact every such a set is an ideal over $\mathcal{P}_f(D_0)$,and every ideal over $\mathcal{P}_f(D_0)$ is of the form $\{A \in \mathcal{P}_f(D_0) | A \sqsubseteq S\}$ for some $S \in \mathcal{F}(D)$.

Theorem 2.1.2 *Let D be an ω-algebraic dcpo. Let $I(D, \sqsubseteq)$ be the set of ideals over D with respect to the ordering $\sqsubseteq$. Then:*

1. *the lower powerdomain $P_L(D)$ is isomorphic to $< I(\mathcal{P}_f(D_0), \sqsubseteq_L), \subseteq >$.*
2. *the upper powerdomain $P_U(D)$ is isomorphic to $< I(\mathcal{P}_f(D_0), \sqsubseteq_U), \subseteq >$.*
3. *the convex powerdomain $P_C(D)$ is isomorphic to $< I(\mathcal{P}_f(D_0), \sqsubseteq_C), \subseteq >$.*

Proof: It is easily proved that the "generating tree construction" is equivalent to the "ideal construction" of this theorem. For take an arbitrary set of the form $\{A \in \mathcal{P}_f(D_0) | A \sqsubseteq S\}$ with $S \in \mathcal{F}(D)$, then it is trivial that this set is an ideal. The other way round take an ideal I, then because $\mathcal{P}_f(D_0) \subseteq \mathcal{F}(D)$ and every directed subset of $\mathcal{F}(D)$ has a lub, I has a lub too. Moreover if $A \in \mathcal{P}_f(D_0)$ and $A \sqsubseteq \sqcup I$ then because A is compact it follows that $\exists A' \in I : A \sqsubseteq A'$, so by left-closedness of I we have $A \in I$. Therefore every ideal is of the form above mentioned.

The order $\preceq$ reduces to $\subseteq$, because every element of $\mathcal{F}(D)$ is represented by the set of its approximations, i.e. the elements of $\mathcal{P}_f(D_0)$ which are smaller.

□

2.2 Some properties of powerdomains

In this section we will use the ideal representation of powerdomains. First we state a lemma which we will make frequent use of in this section.

Lemma 2.2.1 *Let $S \subseteq \mathcal{P}_f(D_0)$ be a finite subset, then*

1. $\cup S = \sqcup S$ *with respect to* $\sqsubseteq_L$
2. $\cup S = \sqcap S$ *with respect to* $\sqsubseteq_U$

Proof: Trivial.

□

We will look at the forms which the *or*-operator assumes in the different powerdomains.

Definition 2.2.1 *Let $I_1, I_2 \in P(D)$, then define $I_1 \mathbin{\bar{\cup}} I_2 = \{A_1 \cup A_2 | A_i \in I_i\}$.*

It will turn out that $\bar{\cup}$ is the *or*-operation in $P_U(D)$ and $P_C(D)$. We show that $\bar{\cup}$ is well-defined, i.e. $I_1 \mathbin{\bar{\cup}} I_2$ is an ideal.

Lemma 2.2.2 *$\bar{\cup}$ is well-defined in $P_U(D)$ and $P_C(D)$.*

Proof: (a) $I_1 \mathbin{\bar{\cup}} I_2$ is left-closed.

- $\sqsubseteq_U$

  Let $A \in I_1 \mathbin{\bar{\cup}} I_2$ and $B \sqsubseteq_U A$. There are $A_i \in I_i$ such that $A_1 \cup A_2 = A$. If $B \sqsubseteq_U A_1 \cup A_2$ then by the properties of $\sqsubseteq_U$ it follows that $B \sqsubseteq_U A_i$. So by left-closedness of I_i we have $B \in I_i$. Therefore $B = B \cup B \in I_1 \mathbin{\bar{\cup}} I_2$.

- $\sqsubseteq_C$

  Suppose $B \sqsubseteq_C A = A_1 \cup A_2 \in I_1 \mathbin{\bar{\cup}} I_2$ as before. By writing out the properties of $\sqsubseteq_C$ we have the following two statements:

  1. $\forall a \in A \exists b \in B : b \leq a$
  2. $\forall b \in B \exists a \in A : b \leq a$

  Now define $B_1 := \{b \in B | \exists a \in A_1 : b \leq a\}$ and $B_2 := \{b \in B | \exists a \in A_2 : b \leq a\}$. By 2. it follows that $B = B_1 \cup B_2$ and by 1. it follows that $B_i \neq \emptyset$.

  It is easy to verify that $B_i \sqsubseteq_C A_i$, so $B_i \in I_i$ and $B = B_1 \cup B_2 \in I_1 \mathbin{\bar{\cup}} I_2$.

(b) $I_1 \mathbin{\bar{\cup}} I_2$ is directed.

- $\sqsubseteq_U$

  Suppose $A, B \in I_1 \mathbin{\bar{\cup}} I_2$ and $A = A_1 \cup A_2$, $B = B_1 \cup B_2$, $A_i, B_i \in I_i$, then there are $R_i \in I_i$ such that $A_i \sqsubseteq_U R_i \wedge B_i \sqsubseteq_U R_i$. Now $B_1 \cup B_2 \sqsubseteq_U R_1 \cup R_2$, $A_1 \cup A_2 \sqsubseteq_U R_1 \cup R_2$ and $R_1 \cup R_2 \in I_1 \mathbin{\bar{\cup}} I_2$.

- $\sqsubseteq_C$
  Analogous to the $\sqsubseteq_U$-order.

□

Contrary to [22] $\sqcup$ is not well-defined in $P_L(D)$.

Example 2.2.1 *Take D the flat poset generated by the set $\{a,b\}$, i.e. $D = (\{\bot,a,b\},\leq)$ with $x,y \in D : x \leq y$ implies $x = \bot \vee x = y$. Now define the following two ideals over D:*
$I_1 := \{\{\bot\},\{a\},\{\bot,a\}\}$
$I_2 := \{\{\bot\},\{b\},\{\bot,b\}\}$
then $I_1 \sqcup I_2 = \{\{\bot\},\{\bot,a\},\{\bot,b\},\{a,b\},\{\bot,a,b\}\}$. This is not left-closed, for $\{a\}\sqsubseteq_L\{a,b\}$, but $\{a\} \notin I_1 \sqcup I_2$.

Therefore the *or*-operation will be defined in $P_L(D)$ as as $I_1 \cup I_2 \cup (I_1 \sqcup I_2)$.

Lemma 2.2.3 *If $I_1, I_2 \in P_L(D)$, then $I_1 \cup I_2 \cup (I_1 \sqcup I_2) \in P_L(D)$.*

Proof: (a) $I_1 \cup I_2 \cup (I_1 \sqcup I_2)$ is left-closed.
If $B\sqsubseteq_L A \in I_i$, then $B \in I_i$. Now suppose $B\sqsubseteq_L A \in I_1 \sqcup I_2$. Then $A = A_1 \cup A_2$, with $A_i \in I_i$. Define $B_i := \{b \in B | \exists a \in A_i : b \leq a\}$ for $i = 1,2$. By the properties of $\sqsubseteq_L$ it follows that $B = B_1 \cup B_2$.
If $B_1 \neq \emptyset$ then $B = B_2 \sqsubseteq_L A_2$ so $B \in I_2$. The case that $B_2 = \emptyset$ is analogous. If both $B_1, B_2 \neq \emptyset$ we have $B_i \sqsubseteq_L A_i$, so $B_i \in I_i$ and $B = B_1 \cup B_2 \in I_1 \sqcup I_2$.

(b) $I_1 \cup I_2 \cup (I_1 \sqcup I_2)$ is directed
According to lemma 2.2.1 below $I_1 \sqcup I_2$ contains all the lubs of sets $\{A_1, A_2\}$ with $A_i \in I_i$. So by the first part of lemma 2.2.5 $I_1 \cup I_2 \cup (I_1 \sqcup I_2)$ is directed.

□

Now we can state the following theorem.

Theorem 2.2.1 *Let I_1, I_2 be ideals, then*

I_1 or $I_2 := I_1 \sqcup I_2$ in $P_U(D)$.

I_1 or $I_2 := I_1 \sqcup I_2$ in $P_C(D)$.

I_1 or $I_2 := I_1 \cup I_2 \cup (I_1 \sqcup I_2)$ in $P_L(D)$.

Proof: By the previous two lemmas the *or*-operations are well-defined. We check that they are commutative, associative and idempotent. The first two properties are trivial, so we concentrate on the last one.

- $\sqsubseteq_U$

  We have to prove that $I = I \,\dot\cup\, I$. Trivially $I \subseteq I \,\dot\cup\, I$. Now take an element $A \cup A'$ of $I \,\dot\cup\, I$,then $A \cup A' \sqsubseteq_U A$ so by left-closedness $A \cup A' \in I$.

- $\sqsubseteq_C$

  $I \subseteq I \,\dot\cup\, I$. Take $A \cup A' \in I \,\dot\cup\, I$. I is directed so $\{A, A'\}$ has an upperbound U in I. It is easy to verify that $A \cup A' \sqsubseteq_C U$, so $A \cup A' \in I$.

- $\sqsubseteq_L$

  Analogous to the case $\sqsubseteq_C$.

□

The powerdomains are dcpo's so they have lubs of all directed sets. In the rest of this section we will give some results about the existence of glb's and lubs of other subsets of $P(D)$ (a statement without subscript is true of all three powerdomains).

Lemma 2.2.4 *Let $I \in P(D)$,then $\{\perp\} \in I$.*

Proof: Trivial.

□

Theorem 2.2.2 *For every subset $S \subseteq P_L(D)$ $\sqcap S$ exists.*

Proof: We prove that $\cap S$ is an ideal, then it follows by the properties of set-inclusion that $\sqcap S = \cap S$.

(a) $\cap S \neq \emptyset$

By the previous lemma.

(b) $\cap S$ is left-closed

Suppose $A \in \cap S$, then $\forall I \in S : A \in I$. If $B \sqsubseteq_L A$ then $\forall I \in S : B \in I$, so $B \in \cap S$.

(c) $\cap S$ is directed

Suppose $A, B \in \cap S$. Then $\forall I \in S : A, B \in I$, so there is a $C_I \in I$ such that $A \sqsubseteq_L C_I \wedge B \sqsubseteq_L C_I$. But by lemma 2.2.1 $A \cup B$ is the lub of $\{A, B\}$, so $A \cup B \sqsubseteq_L C_I$ and $A \cup B \in I$.

Therefore $A \cup B \in \cap S$,and $\{A, B\}$ has an upperbound in $\cap S$.

□

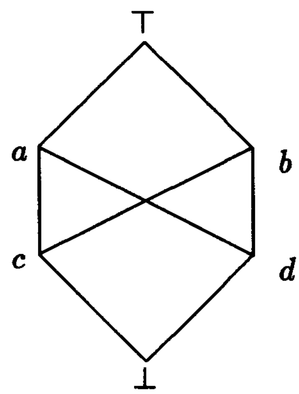
Theorem 2.2.3 *For every finite subset $S \subseteq P_U(D)$ $\sqcap S$ exists.*

Proof: We prove that $\cap S$ is an ideal, then $\sqcap S = \cap S$.
Non-emptiness and left-closedness are as in the proof of the previous theorem. Now take $A, B \in \cap S$ then $\forall I \in S : A, B \in I$, so there is a $C_I \in I$ which is an upperbound of the set $\{A, B\}$. With lemma 2.2.1 it follows that $\cup\{C_I | I \in S\} = \sqcap\{C_I | I \in S\}$, so $\forall I \in S : \cup\{C_I | I \in S\} \sqsubseteq_U C_I$, therefore $\forall I \in S : \cup\{C_I | I \in S\} \in I$. Trivially we have $A \sqsubseteq_U \cup \{C_I | I \in S\} \wedge B \sqsubseteq_U \cup \{C_I | I \in S\}$, and so has $\{A, B\}$ an upperbound in $\cap S$.

$\square$

$P_C(D)$ does not have a glb for every subset $S \subseteq P_C(D)$.

Example 2.2.2 *Take for D the following ω-algebraic dcpo:*



Every element of D is compact. Now define the following two ideals:
$I_1 := \{A \in \mathcal{P}_f(D) | A \sqsubseteq_C \{a\}\}$
$I_2 := \{A \in \mathcal{P}_f(D) | A \sqsubseteq_C \{b\}\}$
Now $I_1 \cap I_2$ is not an ideal because $\{c\}, \{d\} \in I_1 \cap I_2$ do not have an upperbound in this set. Moreover
$J_1 := \{A \in \mathcal{P}_f(D) | A \sqsubseteq_C \{c\}\}$
$J_1 := \{A \in \mathcal{P}_f(D) | A \sqsubseteq_C \{d\}\}$
are two different maximal lower bounds of $\{I_1, I_2\}$.

We now give some results about the existence of lubs.

Lemma 2.2.5 *Let $S = \{I_j | j \in J\}$ be a finite set of ideals over a poset $(D, \leq)$, and let $S_f := \{f(j) | j \in J\}$ for a function $f : J \to \cup\{I_j | j \in J\}$ wich satifies $f(j) \in I_j$. If $\sqcup S_f$ exists for all functions f, then $\sqcup S$ exists in the ideal completion of D.*

Proof: Define $L := \{\sqcup S_f | f \textit{ is a function as defined in lemma}\}$. We will prove that $I := \{d \in D | \exists d' \in \cup(S) \cup L : d \leq d'\}$ is the lub of S.

(a) I is directed.
We prove that $\cup(S) \cup L$ is directed, then it follows that I is directed. Take $a, b \in \cup(S) \cup L$ then if

- $a, b \in I_j$, then they have an upperbound in I_j for this set is directed.

- $a \in I_j, b \in I_k, i \neq k$, then there is a function f with $f(j) = a, f(k) = b$, and $\sqcup S_f \in L \subseteq \cup(S) \cup L$.
- $a \in I_j, b \in L$, then $b = \sqcup S_f$ for a certain function f. Now a and $f(j)$ do have an upperbound a' in I_j. Take a function f' which is equal to f, except that $f'(j) = a'$. We have that $\sqcup S_{f'} \in L$ and $\sqcup S_{f'}$ is an upperbound of S_f. So $b \leq \sqcup S_{f'}$. It is trivial that $a \leq a' \leq \sqcup S_{f'}$, so $\{a, b\}$ has an upperbound.
- $a, b \in L$, then there are functions f, g such that $\sqcup S_f = a$ and $\sqcup S_g = b$. Take a function h such that $\forall j \in J : f(j) \leq h(j) \wedge g(j) \leq h(j)$. Then $\sqcup S_h \in L$ is an upperbound of S_f and S_g, so $a = \sqcup S_f \leq \sqcup S_h$ and $b = \sqcup S_g \leq \sqcup S_h$.

(b) I is left-closed.
Trivial.

(c) I is an ideal.
By point (a) and (b).

(d) $I = \sqcup S$.
$\cup S \subseteq I$ is trivial.
Suppose $\cup S \subseteq U$, with U an ideal. Take $a \in I$, then if $a \in I_j$ for some $j \in J$ it follows $a \in U$. If $a \in L$ then $a = \sqcup S_f$ for some function f. Now $S_f \subseteq U$ and S_f is finite, so this set has an upperbound in U. But then by leftclosedness of U it follows that $a = \sqcup S_f \in U$.

□

Theorem 2.2.4 *Every finite $S \subseteq P_L(D)$ has a lub.*

Proof: Every finite subset $R \subseteq \mathcal{P}_f(D_0)$ has a lub with respect to the ordering $\sqsubseteq_L$, for $\cup R = \sqcup R$ (lemma 2.2.1). So by the previous lemma $\sqcup S = \{A \in \mathcal{P}_f(D_0) | \exists A' \in \cup(S) \cup L : A \leq A'\}$, with $L = \bar{\cup} S$.

□

The lub of the set $\{I_1, I_2\}$ is given by $\sqcup S = \{A \in \mathcal{P}_f(D_0) | \exists A' \in I_1 \cup I_2 \cup (I_1 \bar{\cup} I_2) : A \leq A'\}$. According to the first part of the proof of lemma 2.2.3 $I_1 \cup I_2 \cup (I_1 \bar{\cup} I_2)$ is left-closed, so in fact $\sqcup\{I_1, I_2\} = I_1 \cup I_2 \cup (I_1 \bar{\cup} I_2)$. So it turns out that in the lower powerdomain the *or*-operation yields the lub.

Definition 2.2.2 *A poset P is* finite bounded complete *if for every subset $S \subseteq P$ which has finite many but not zero upperbounds in P, $\sqcup S$ exists.*

Lemma 2.2.6 *Let P be a poset, then:*
for every finite $S \subseteq P$ $\sqcap S$ exists $\Rightarrow$ P is finite bounded complete.

Proof: Take $S \subseteq P$ and suppose S has finite many but not zero upperbounds. Define $\mathrm{Up}(S) := \{r | r \textit{ is an upperbound of } S\}$. Now S has upperbounds so $\mathrm{Up}(S) \neq \emptyset$. $\sqcap \mathrm{Up}(S)$ exists (for $\mathrm{Up}(S)$ is finite) and it is easy to verify that $\sqcup S = \sqcap \mathrm{Up}(S)$.

□

Theorem 2.2.5 *$P_U(D)$ is finite bounded complete.*

Proof: This follows by theorem 2.2.3 and lemma 2.2.6.

□

Chapter 3

3.1 Powerdomains are continuous algebras

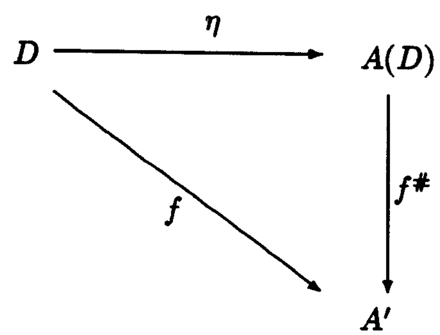
From now on dcpo's need not have a least element.
In a paper by Hennessey and Plotkin ([11]) powerdomains are characterized as free continuous algebras. In the rest of the paper we take this more abstract approach to powerdomains.

Definition 3.1.1 *An* algebra *is a set (the* carrier *of the algebra) together with an indexed family of operations (functions) defined on (cartesian powers of) that set.*

Definition 3.1.2 *An* ordered algebra *is an algebra with a poset as carrier and monotone operations.*

Definition 3.1.3 *A* continuous algebra *is an algebra with a dcpo as carrier and continuous operations.*

A free algebra is just an instance of the more general concept of a free structure. The free continuous algebra $A(D)$ generated by a dcpo D satisfies the requirements made visible in the following diagram:



Thus there is a continuous function $\eta : D \to A(D)$ such that given a continuous $f : D \to A'$, with A' an algebra of the same signature as $A(D)$ (i.e. it has the same operations),there is a unique continuous operator-preserving function $f^\# : A(D) \to A'$ such that $f^\# \circ \eta = f$.

Definition 3.1.4 *A function $f : A \to A'$ between algebras of the same signature is* operator-preserving *iff $f(\omega(a_1, \ldots, a_n)) = \omega(f(a_1), \ldots, f(a_n))$ for all operations $\omega : A^n \to A$, $n \in N$ and elements $a_i \in A$.*

Note that in the above we have confused algebras with their carriers, a harmless practice which will be carried on in the rest of this paper.

As all free structures, free algebras can be understood as elements in the range of a left-adjoint to a forgetful functor, in this case the forgetful functor which maps algebras into the category of their carriers. In this section we will prove the existence of a left-adjoint which constructs powerdomains. In the next section we will see that this abstract approach can be used to find representations of powerdomains over some restricted classes of dcpo's.

Definition 3.1.5 *Let $+$ be a binary, commutative, associative and idempotent operator.*
A (bounded) nondeterministic set *is an algebra with a $+$-operator.*
A (bounded) nondeterministic poset *is an ordered algebra with a $+$-operator. A* (bounded) nondeterministic dcpo *is a continuous algebra with a $+$-operator.*

Now define the following categories:
Dcpo : dcpo's and continuous functions.
$Dcpo^+$: nondeterministic dcpo's and linear, continuous functions.
The functions in $Dcpo^+$ are operator-preserving (here *linear*), i.e. $f(x+y) = f(x) + f(y)$.

There is a forgetful functor $U : Dcpo^+ \rightarrow Dcpo$ which maps nondeterministic dcpo's to their carriers, i.e. it "forgets" simply the $+$-operation. The left-adjoint of U makes every object of *Dcpo* into a free nondeterministic dcpo (a powerdomain).

Theorem 3.1.1 *The functor $U : Dcpo^+ \rightarrow Dcpo$ has a left-adjoint P.*

Before we prove this theorem we give some of the results of [11]. For each algebraic element D of *Dcpo* the functor P constructs the convex powerdomain, so in this case $P(D) \cong P_C(D)$. Left-adjoints which give the lower and upper powerdomain when applied to an algebraic dcpo can be obtained in the same manner if we put some additional requirements on the elements of $Dcpo^+$. If we take those objects of $Dcpo^+$ which satisfy $x \leq x+y$, and we take the left-adjoint of the forgetful functor from *Dcpo* to this subcategory, then we have the lower powerdomain constructor P_L. In the same manner the equation $x + y \leq x$ gives the upper powerdomain constructor P_U. We will concentrate on the convex powerdomain in this chapter.

Further we need not worry about the existence of a least element for according to [11] P maps dcpo's with a least element to powerdomains with a least element.

Now we are going to prove theorem 3.1.1 by the Freyd Adjoint Functor Theorem (FAFT). To apply the FAFT we have to verify the following items:

- $Dcpo^+$ is small-complete.

- $U : Dcpo^+ \rightarrow Dcpo$ preserves all limits.

- U satisfies the solution set condition.

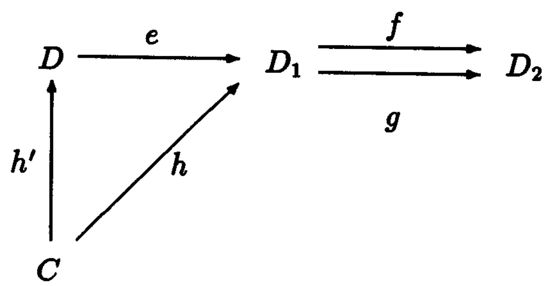
Theorem 3.1.2 *$Dcpo^+$ has all small products.*

Proof: Given a small index set I we have $\prod(D_i|i \in I) = \{a|a : I \to \cup D_i \textit{ and } a(i) \in D_i\}$. For a, b elements of the product we have
$a \leq b := \forall i \in I : a(i) \leq b(i)$
$(a + b)(i) := a(i) + b(i)$.
Let S be a directed subset of the product then
$\sqcup S = a$, with $a(i) := \sqcup\{b(i)|b \in S\}$.
Projections: $pr_j : \prod(D_i|i \in I) \to D_j : a \mapsto a(j)$, for $j \in I$.

□

Theorem 3.1.3 *$Dcpo^+$ has all equalizers.*

Proof: Let $f, g : D_1 \to D_2$ be two arrows in $Dcpo^+$. We need an object D and arrow $e : D \to D_1$ such that the following diagram commutes:



It is easy to verify that if we take $D := \{x \in D_1|f(x) = g(x)\}$ and e the inclusion $D \to D_1$ then these requirements are satisfied.

□

Theorem 3.1.4 *$Dcpo^+$ is small-complete.*

Proof: By theorem 3.1.2, 3.1.3 and 1.3.1.

□

Small products and equalizers in $Dcpo$ are the same as in $Dcpo^+$ but without the +-operation. So the functor $U : Dcpo^+ \to Dcpo$ triviallly preserves small products and equalizers. By theorem 1.3.2 and the previous theorem it follows therefore that

Theorem 3.1.5 *$U : Dcpo^+ \to Dcpo$ preserves all small limits.*

Before we can prove that the solution set condition holds for U we need some theory about closed sets.

Definition 3.1.6 *Let S be a subset of a set D. S is* closed *for a certain property p iff $p(S)$ is true. The* closure *of S in D for property p is a closed set $S' \subseteq D$ such that (i) $S \subseteq S'$ and (ii) if $S \subseteq R$ and R is closed for p then $S' \subseteq R$.*

The closure of a set S for a property p always exists if V consists of for p closed sets implies that $\cap V$ is closed for p, and if p(D) is true. In this case the closure of S is equal to $\cap V$ with $V := \{S' | S \subseteq S' \text{ and } S' \text{ is closed for } p\}$. $V \neq \emptyset$ because $D \in V$.

In this paper we will use the following types of closed sets and corresponding closures.

Definition 3.1.7 *Let D be a nondeterministic poset and $S \subseteq D$, then S is:*
$\sqcup$-closed iff $S' \subseteq S$ directed and $\sqcup S' \in D$ implies $\sqcup S' \in S$,
$\downarrow$-closed (=left-closed) iff $x \leq y \in S$ implies $x \in S$,
$+$-closed iff $x, y \in S$ implies $x + y \in S$,
$\sqcup\downarrow$-closed iff S is $\sqcup$-closed and $\downarrow$-closed,
$\sqcup+$-closed iff S is $\sqcup$-closed and $+$-closed,
$\sqcup\downarrow+$-closed iff S is $\sqcup$-closed, $\downarrow$-closed and $+$-closed.

We will now concentrate on the $\sqcup$-closure of a set S in a poset D, denoted by $\overline{S}$. In the following $f(S) = \{f(a) | a \in S\}$ for a function f and a set S.

Theorem 3.1.6 *Let $S, S_1, S_2 \subseteq D \in Pos^+$, then*

1. $\overline{\emptyset} = \emptyset$
2. $\overline{\overline{S}} = \overline{S}$
3. $S \subseteq \overline{S}$
4. $S_1 \subseteq S_2$ *implies* $\overline{S}_1 \subseteq \overline{S}_2$
5. *If D in Dcpo, then $\overline{S}$ in Dcpo.*

Proof: Trivial.

□

Theorem 3.1.7 *If $f : D \to E$ is continuous then $f(\overline{S}) \subseteq \overline{f(S)}$.*

Proof: Define $R := \{x \in \overline{S} | f(x) \in \overline{f(S)}\}$ then clearly $S \subseteq R \subseteq \overline{S}$. We now show that R is $\sqcup$-closed.
Take $R' \subseteq R$, R' directed and suppose that $\sqcup R'$ exists in D. Then $\sqcup R' \in \overline{S}$ and $f(\sqcup R') = \sqcup f(R') \in \overline{f(S)}$, so $\sqcup R' \in R$. R is $\sqcup$-closed, so $\overline{S} \subseteq R$, so $\overline{S} = R$ and $f(\overline{S}) = \overline{f(S)}$.

□

Theorem 3.1.8 *If $f : D \to D$ continuous, then $f(S) \subseteq S$ implies $f(\overline{S}) \subseteq \overline{S}$.*

Proof: By the previous theorem we have $f(\overline{S}) \subseteq \overline{f(S)}$. Now $f(S) \subseteq S$, so $\overline{f(S)} \subseteq \overline{S}$ by theorem 3.1.6.

□

Theorem 3.1.9 $\overline{S} \times \overline{S} = \overline{S \times S}$

Proof: It is trivial that $\overline{S \times S} \subseteq \overline{S} \times \overline{S}$, for $\overline{S} \times \overline{S}$ is ⊔-closed and $S \times S \subseteq \overline{S} \times \overline{S}$. We now prove $\overline{S} \times \overline{S} \subseteq \overline{S \times S}$ in two steps:

(a) $S \times \overline{S} \subseteq \overline{S \times S}$
For $a \in D$ define $f_a : D \to D \times D : y \mapsto < a, y >$. This function is continuous, so $f_a(\overline{S}) \subseteq \overline{f_a(S)}$ by theorem 3.1.7. Now $S \times \overline{S} = \cup\{f_a(\overline{S}) | a \in S\} \subseteq \cup\{\overline{f_a(S)} | a \in S\}$. Clearly $f_a(S) \subseteq S \times S$ for $a \in S$, so $\overline{f_a(S)} \subseteq \overline{S \times S}$, and $\cup\{\overline{f_a(S)} | a \in S\} \subseteq \overline{S \times S}$.

(b) $\overline{S} \times \overline{S} \subseteq \overline{S \times \overline{S}}$ For $a \in D$ define $g_a : D \to D \times D : y \mapsto < y, a >$. This function is continuous, so $g_a(\overline{S}) \subseteq \overline{g_a(S)}$ by theorem 3.1.7. Now $\overline{S} \times \overline{S} = \cup\{g_a(\overline{S}) | a \in \overline{S}\} \subseteq \cup\{\overline{g_a(S)} | a \in \overline{S}\}$. Clearly $g_a(S) \subseteq S \times \overline{S}$ for $a \in \overline{S}$, so $\overline{g_a(S)} \subseteq \overline{S \times \overline{S}}$, and $\cup\{\overline{g_a(S)} | a \in \overline{S}\} \subseteq \overline{S \times \overline{S}}$.

By (a) $S \times \overline{S} \subseteq \overline{S \times S}$, so $\overline{S \times \overline{S}} \subseteq \overline{S \times S}$. So $\overline{S} \times \overline{S} \subseteq \overline{S \times \overline{S}} \subseteq \overline{S \times S}$.

□

Theorem 3.1.10 *$f(S \times S) \subseteq S$ implies $f(\overline{S} \times \overline{S}) \subseteq \overline{S}$, with $f : D \times D \to D$ continuous.*

Proof: By theorem 3.1.9 we have $f(\overline{S} \times \overline{S}) = f(\overline{S \times S})$. By theorem 3.1.7 we have $f(\overline{S \times S}) \subseteq \overline{f(S \times S)}$. Finally $f(S \times S) \subseteq S$ so by theorem 3.1.6 (4) it follows that $\overline{f(S \times S)} \subseteq \overline{S}$.

□

Corollary 3.1.1 *If S is +-closed, then $\overline{S}$ is ⊔+-closed.*

Proof: Trivially $\overline{S}$ is ⊔-closed. S is +-closed, so for every $x, y \in S : x + y \in S$. But then by the previous theorem $x, y \in \overline{S} \Rightarrow x + y \in \overline{S}$, so $\overline{S}$ is +-closed.

□

Corollary 3.1.2 ⊔+-*closure(S)* $= \overline{+ - closure(S)}$

Proof: Define S_1 = ⊔+-closure(S), $S_2 = \overline{+ - closure(S)}$. Now S_2 is ⊔-closed and +-closed (by the previous corollary) and $S \subseteq S_2$, so $S_1 \subseteq S_2$.
S_1 is +-closed and $S \subseteq S_1$, so +-closure(S)$\subseteq S_1$. Further S_1 is ⊔-closed and with +-closure(S) $\subseteq S$ we have that $S_2 \subseteq S_1$.

□

It is easy to see that the cardinality of the $\sqcup$-closure and the $+$-closure of a set S is bounded by the cardinality of S (in fact $card(\overline{S}) \leq 2^{card(S)}$, see [3]). So by the previous corollary this is true of the $\sqcup+$-closure of S.
Now we can give

Proof of theorem 2.2.1: We have already verified that $Dcpo^+$ is small-complete and that $U : Dcpo^+ \rightarrow Dcpo$ preserves all small limits. It remains to find a solution set for each $D \in Dcpo$.
Consider any function $f : D \rightarrow UE$ with $E \in Dcpo^+$. Let E' be the $\sqcup+$-closure of $f(D)$ in E, then the cardinal number of E' is bounded given D. Taking one copy of each isomorphism class of such nondeterministic dcpo's E' then gives a small set of nondeterministic dcpo's (see [13]), and the set of all functions $D \rightarrow UE'$ is then a solution set (see [13]).

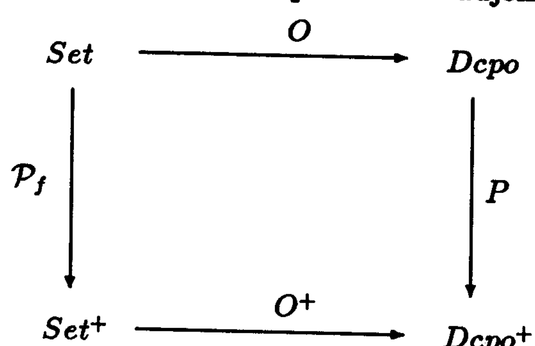
$\square$

The existence of arbitrary free continuous algebras with finitary operations (i.e. every operation is of the form $\omega : A^n \rightarrow A, n \in N$) can be proved by a slight extension of the theory of this section.

Chapter 4

4.1 Powerdomains over restricted classes of dcpo's

Existence of powerdomains (free nondeterministic dcpo's) has been proven in the previous section. In this section we try to find a representation for powerdomains by categorical means.
First we introduce some new categories:
Set : Sets and functions between sets,
Set^+ : Nondeterministic sets and linear functions,
Pos : Posets and monotone functions,
Pos^+ : Nondeterministic posets and monotone, linear functions.
We have forgetful functors $Dcpo^+ \to Set^+$ and $Dcpo \to Set$ which forget the ordening. There is also a forgetful functor $Set^+ \to Set$, which maps algebras with a +-operator to their carriers. All these functors have left-adjoints. Consider the following diagram where the arrows represent left-adjoints.



The functor O maps sets to discrete posets, i.e. the order is such that $x \leq y \Leftrightarrow x = y$. Discrete posets are trivially dcpo's because the only directed subsets are the one element sets.
$\mathcal{P}_f$ is the finite powerset constructor, which we have already met in the first chapter. More formally:

Theorem 4.1.1 *The forgetful functor Dcpo → Set has a left-adjoint $O : Set \to Dcpo$.*

Proof: For $S \in Set$ define $O(S) = S$, and for $x, y \in S : x \leq y := (x = y)$. The components of the unit are the identity functions.

□

Theorem 4.1.2 *The forgetful functor $Dcpo^+ \to Set^+$ has a left-adjoint $O^+ : Set^+ \to Dcpo^+$.*

Proof: For $S \in Set^+$ define $O^+(S) = S$. Order and unit are as with O. The +-operator in $O^+(S)$ is that of S.

□

Theorem 4.1.3 *The forgetful functor $Set^+ \to Set$ has a left-adjoint $\mathcal{P}_f : Set \to Set^+$.*

Proof: For $S \in Set$ we have $\mathcal{P}_f(S) = \{x \subseteq S | x \neq \emptyset \wedge x\ finite\}$ and $x + y := x \cup y$, for $x, y \in \mathcal{P}_f$. The unit of the adjunction consists of the one-element set constructor: $\{\cdot\} : S \to U\mathcal{P}_f(S) : a \mapsto \{a\}$.

□

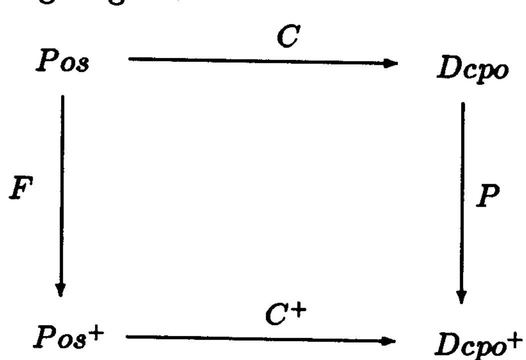
According to theorem 1.3.4 (the composition of two left-adjoints is a left-adjoint) the functors PO and $O^+\mathcal{P}_f$ are both left-adjoints of the forgetful functor $Dcpo^+ \to Set$. It follows with theorem 1.3.3 that they are isomorphic. So the diagram above can be said to commute in a weak sense, i.e. $PO(S) \cong O^+\mathcal{P}_f(S)$ for a set S. Any discrete dcpo $D = (S, \leq)$ can be written as $O(S)$, with S the set of all its elements. So

Theorem 4.1.4 *For any discrete poset $D = (S, \leq)$, $P(D) \cong O^+\mathcal{P}_f(S)$.*

Proof: $P(D) = PO(S) \cong O^+\mathcal{P}_f(S)$.

□

We have found a representation for powerdomains over discrete dcpo's, because $O : Set \to Dcpo$ identified this subcategory of Dcpo. The idea is now to take another functor to $Dcpo$ with a larger image, and to apply the same method. Consider the following diagram:



The arrows are left-adjoints to the appropiate forgetful functors. Again the square (weakly) commutes.

Theorem 4.1.5 *The forgetful functor $Pos^+ \to Pos$ has a left-adjoint $F : Pos \to Pos^+$.*

Proof: Given a subset $S \subseteq D$, with $D \in Pos$, define $Minmax(S) := Min(S) \cup Max(S)$, with $Min(S)$ the set containing the minima of S, and $Max(S)$ the set containing the maxima of S.
Now for $D \in Pos$ define
$F(D) := \{Minmax(x) | x \subseteq D \wedge x \neq \emptyset \wedge x \text{ finite}\}$,
$x \leq y := x \sqsubseteq_C y$, for $x, y \in F(D)$,
$x + y := Minmax(x \cup y)$, for $x, y \in F(D)$.
The unit $\eta_D : D \to UF(D)$ is given by $\eta_D(a) = \{a\}$.

(a) $(F(D), \sqsubseteq_C)$ is a poset.
Reflectivity and transitivity are trivial. Now suppose $x \sqsubseteq_C y \wedge y \sqsubseteq_C x$, $x, y \in F(D)$, and take $a \in x$, then there are $b, b' \in y$ such that $b' \leq a \leq b$. If a is a minimum in x, then take $a_1 \in x$ such that $a_1 \leq b' \leq a$, so $a_1 \leq a$ and $a = a_1$, but then $b = a$ and $a \in y$. If a is a maximum in x then take $a_2 \in y$ such that $a \leq b \leq a_2$, thus $a = a_2$ and $a = b \in y$. Therefore $x \subseteq y$.
Analogous $y \subseteq x$ and so $x = y$.

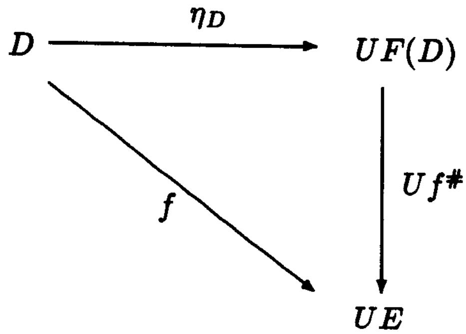
(b) $+$ is well-defined.
Trivial.

(c) $+$ is monotone.
Suppose $x \sqsubseteq_C x'$. We have to show that $x + y \sqsubseteq_C x' + y$ for all $y \in D$. Take $a \in x + y$. If $a \in x$ then $\exists a' \in x'$ such that $a \leq a'$, so there is an element greater than a in $x' + y$. If $a \in y$ then $a \in x' + y$ or there is an other element greater than a in $x' + y$.
Take $a' \in x' + y$ then there is a smaller element in $x + y$ analogously. So $x + y \sqsubseteq_C x' + y$.

(d) $+$ is commutative, associative and idempotent.
Trivial.

(e) η_D is well-defined and monotone.
Trivial.

(f) Given a monotone function $f : D \to UE$, with $E \in Pos^+$, define $f^\# : F(D) \to E$ as $f^\#(\{y_1, \ldots, y_n\}) = f(y_1) + \cdots + f(y_n)$. This function $f^\#$ is well-defined.



(g) $f^\#$ is linear.
Suppose $y, y' \in F(D)$ with $y = \{y_1, \ldots, y_n\}$, $y' = \{y'_1, \ldots, y'_m\}$. We have to show that $f^\#(y + y') = f^\#(y) + f^\#(y')$. Now $f^\#(y + y') = f^\#(Minmax(y \cup y')) = f^\#(Minmax\{y_1, \ldots, y_n, y'_1, \ldots, y'_m\})$. Say $Minmax\{y'_1, \ldots,$ $\{x_1, \ldots, x_p\}$. It is easy to see that
$f^\#(x_1) + \cdots + f^\#(x_p) \leq f^\#(y_1) + \cdots + f^\#(y'_m)$
and
$f^\#(y_1) + \cdots + f^\#(y'_m) \leq f^\#(x_1) + \cdots + f^\#(x_p)$ in E (because + is monotone in E). So $f^\#(y + y') = f^\#(y) + f^\#(y')$.

(h) $f^\#$ is monotone.
Trivial (because f and + are monotone).

(i) $Uf^\# \circ \eta_D = f$
$Uf^\#(\eta_D(a)) = Uf^\#(\{a\}) = Uf(a) = f(a)$.

(j) $f^\#$ is unique.
Suppose g is an arrow in Pos^+ and $Ug(\eta_D(a)) = f(a)$. Then for $y \in F(D)$ we have $Ug(y) = Ug(\{y_1\} \cup \cdots \cup \{y_n\}) = Ug(\{y_1\}) + \cdots + Ug(\{y_n\}) = f(y_1) + \cdots + f(y_n) = Uf^\#(y)$. So $g = f^\#$.

□

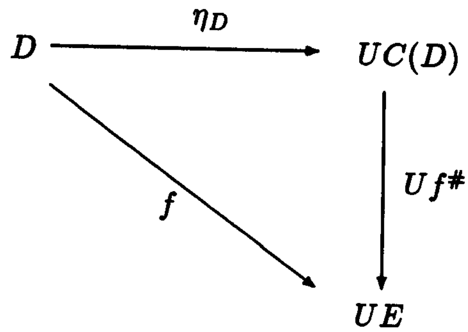
Just as with $\mathcal{P}_f$ the elements of the free ordered algebras are *finite* sets. This is because we consider bounded nondeterminism only, and so we have finite sums.

The left-adjoints $C : Pos \to Dcpo$ is the so-called *completion by ideals* of a poset D.

Theorem 4.1.6 *The forgetful functor Dcpo → Pos has a left-adjoint C : Pos → Dcpo.*

Proof: We will just describe the adjoint and leave the full proof, which is just a (boring) verification of properties, to the reader.
Define $C(D) := \{I | I \text{ is an ideal over } D\}$. For $I_1, I_2 \in C(D) : I_1 \leq I_2 := I_1 \subseteq I_2$ and given a directed subset $S \subseteq C(D)$ we have $\sqcup S = \cup S$. The unit maps each element to the principal ideal generated by that element

$\eta_D : D \to UC(D) : a \mapsto I_a$. Given a monotone function $f : D \to UE$, with $E \in Dcpo$ define $f^\# : C(D) \to E$ as $f^\#(I) = \sqcup\{f(a)|a \in I\}$. This $f^\#$ is the unique continuous function which makes the following diagram commute.



□

The functor $C^+ : Pos^+ \to Dcpo^+$ is the same as C, except that a +-operation is defined for the elements of C(D).

Theorem 4.1.7 *The forgetful functor $Dcpo^+ \to Pos^+$ has a left-adjoint $C^+ : Pos^+ \to Dcpo^+$.*

Proof: Let $C^+(D) := C(D)$, with the same order, lubs and unit. Let $\downarrow S$ be the $\downarrow$-closure of a set S. For $I_1, I_2 \in C^+(D)$ define $I_1 + I_2 :=\downarrow \{x_1 + x_2 | x_1 \in I_1 \wedge x_2 \in I_2\}$.

□

Take $D \in Pos$ then we have $PC(D) \cong C^+F(D)$. It is of some interest to characterize the subcategory Image(C) of *Dcpo*.
It contains all (not necessarily countable) algebraic dcpo's. For suppose D an algebraic dcpo and D_0 the set of its compact elements, then $D \cong C(D_0)$. The other way round take $D \in Pos$, then $C(D)$ is algebraic and has the set of principal ideals as the set of its compact elements.
So we have the following result:

Theorem 4.1.8 *If D is an algebraic dcpo, then $P(D) \cong C^+F(D_0)$.*

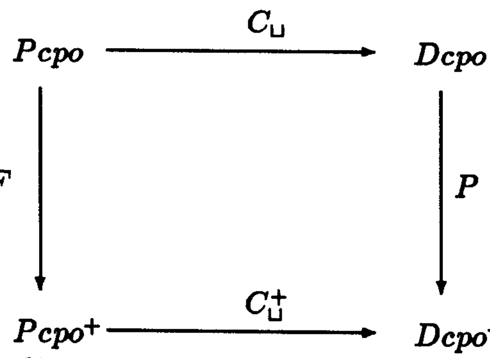
Proof: $P(D) \cong P(C(D_0) \cong C^+F(D_0)$.

□

This is the same we have proven in chapter 2 but here it follows in a much more elegant way.

4.2 Powerdomains over arbitrary dcpo's

In this section we will find that the powerdomain over a dcpo D is isomorphic to a certain completion of $F(D)$, with F the left-adjoint $F : Pos \to Pos^+$.
First we define two new categories:
$Pcpo$: posets and continuous functions,
$Pcpo^+$: nondeterministic posets and linear, continuous functions.
Clearly $Dcpo$ is a subcategory of $Pcpo$,and $Pcpo$ is a subcategory of Pos ($Pcpo$ has the same objects as Pos). Consider the following diagram:



The arrows are left-adjoints to the appropriate forgetful functors. All these left-adjoints will be shown to exist. Because the diagram (weakly) commutes we have that $PC_\sqcup(D) \cong C_\sqcup^+ F(D)$, for $D \in Pcpo$.

In contrast with the left-adjoint $C : Pos \to Dcpo$ the functor $C_\sqcup : Pcpo \to Dcpo$ preserves all existing lubs of directed subsets, because the unit of $C_\sqcup$ is continuous. So if $D \in Pcpo$ and D is directed complete then $C_\sqcup(D) \cong D$, so $P(D) \cong PC_\sqcup(D) \cong C_\sqcup^+ F(D)$. Thus the construction of P follows from that of the left-adjoint $Pcpo \to Dcpo^+$.
First we have:

Theorem 4.2.1 *The forgetful functor Dcpo $\to$ Pcpo has a left-adjoint $C_\sqcup : Pcpo \to Dcpo$.*

Proof: Left to the reader, or see [3].

□

For every $D \in Pcpo$ we take a special $Z_D \in Dcpo$:

Definition 4.2.1 *Let $D \in Dcpo$, then $Z_D := \{x \subseteq D | x \text{ is } \sqcup \downarrow\text{-closed}\}$.*

Theorem 4.2.2 *$(Z_D, \subseteq)$ with $\subseteq$ the subset order is an object of Dcpo.*

Proof: In fact $(Z_D, \subseteq)$ has lubs of all subsets $S \subseteq Z_D$. For define $\sqcup S := \pi(\cup S)$, with $\pi(R) = \sqcup \downarrow$-closure($R$) for a set R.
Clearly $\forall x \in S : x \subseteq \pi(\cup S)$. Suppose $\forall x \in S : x \subseteq L \in Z_D$, then $\cup S \subseteq L$, so $\pi(\cup S) \subseteq L$, for L is $\sqcup \downarrow$-closed and $\pi(\cup S)$ is the $\sqcup \downarrow$-closure of $\cup S$ in D.

□

We can define a continuous function $\eta : D \rightarrow Z_D$ by $\eta(a) := \{b \in D | b \leq a\}$. Although we can not prove it we think that $C_\sqcup(D)$ is isomorphic to the $\sqcup$-closure of $\eta(D)$ in Z_D (we only need to prove that given a continuous $f : D \rightarrow E$ there is a unique $f^\# : \sqcup\text{-closure}(\eta(D)) \rightarrow E$ such that $f = f^\# \circ \eta$).

Anyhow the mere existence of Z_D gives us some information about the structure of $C_\sqcup(D)$, for the unique function $g^\# : C_\sqcup(D) \rightarrow Z_D$ induced by a function $g : D \rightarrow Z_D$ has to be continuous.

Now *Set*,*Set*$^+$,*Pos*,*Pos*$^+$,*Pcpo*,*Pcpo*$^+$,*Dcpo* and *Dcpo*$^+$ are all small-complete and do have the same products and equalizers (possible with additional order, lubs, etc.). Forgetful functors between these categories preserve small products and equalizers, and therefore preserve all small limits. In particular this is the case with the forgetful functor *Dcpo*$^+ \rightarrow$ *Pcpo*$^+$.

Theorem 4.2.3 *The forgetful functor Dcpo*$^+ \rightarrow$ *Pcpo*$^+$ *has a left-adjoint* $C_\sqcup^+$: *Pcpo*$^+ \rightarrow$ *Dcpo*$^+$.

Proof: By the above *Dcpo*$^+$ is small-complete and U preserves all small limits. We have to find a solution set for each $D \in$ *Dcpo*$^+$. So given $f : D \rightarrow UE$, $D \in$ *Pcpo*$^+$, $E \in$ *Dcpo*$^+$ consider $\overline{f(D)}$, the $\sqcup$-closure of $f(D)$ in E. $f(D)$ is +-closed for take $y_1, y_2 \in f(D)$,then $y_i = f(x_i)$, so $y_1 + y_2 = f(x_1) + f(x_2) = f(x_1 + x_2) \in f(D)$, for f is an arrow in *Pcpo*$^+$ and is therefore linear. By corollary 2.1.1 we have that $\overline{f(D)}$ is $\sqcup$+-closed, so $\overline{f(D)} \in$ *Dcpo*$^+$. The cardinal number of $\overline{f(D)}$ is bounded by that of D. Taking one copy of each isomorphism class of such nondeterministic dcpo's $\overline{f(D)}$ then gives a small set of nondeterministic dcpo's (see [13]), and the set of all functions $D \rightarrow U\overline{f(D)}$ is then a solution set (see [13]).

□

For $D \in$ *Pcpo*$^+$ we define $W_D \in$ *Dcpo*$^+$.

Definition 4.2.2 *Let* $D \in$ *Pcpo*$^+$, *then* $W_D := \{x \subseteq D | x$ *is* $\sqcup \downarrow$ +- *closed*$\}$.

Theorem 4.2.4 $(W_D, \subseteq, +) \in$ *Dcpo*$^+$ *with* $\subseteq$ *the subset order, and* $x + y := \phi\{a + b | a \in x, b \in y\}$ *with* $\phi(R) = \sqcup \downarrow$ +- *closure*(R) *for a set R, and* $x, y \in W_D$.

Proof: Define $R_{x,y} := \{a + b | a \in x, b \in y\}$ for sets x and y. We prove that the +-operator is continuous in W_D, i.e. if S is a directed subset of W_D, then for all $y \in D$ $\sqcup(S) + y = \sqcup\{x + y | x \in S\}$. In fact this is true for all subsets S. First we prove that $\sqcup\{x + y | x \in S\} \subseteq \sqcup(S) + y$. Take $a \in x$, then $a \in \cup S$, so $a \in \phi(\cup S) = \sqcup S$. It follows that $R_{x,y} \subseteq R_{\sqcup S,y}$. So $x + y = \phi(R_{x,y}) \subseteq \phi(R_{\sqcup S,y}) = \sqcup(S) + y$.
The other way round we prove that $\sqcup(S) + y \subseteq \sqcup\{x + y | x \in S\}$. So we have to prove that $\phi\{a + b | a \in \phi(\cup S), b \in y\} \subseteq \phi \cup_{x \in S} \phi\{a + b | a \in x, b \in y\}$. If we can show that $\{a + b | a \in \phi(\cup S), b \in y\} \subseteq \phi \cup_{x \in S} \phi\{a + b | a \in x, b \in y\}$, then this follows trivially. We can further reduce this with the following fact:

- $\phi \bigcup_{p \in V} \phi(p) = \phi \bigcup (V)$, for a set V containing sets.

So we have to show that $\{a + b | a \in \phi(\bigcup S), b \in y\} \subseteq \phi\{a + b | a \in \bigcup S, b \in y\}$. Now we need the following:

- $f\phi(V) \subseteq \phi f(V)$, for a linear, continuous $f : G \to H$ between nondeterministic posets G and H.

This can proved easily, in analogy with the proof of theorem 3.1.7. Now define a function $f_b : D \to D : a \mapsto a + b$, then clearly this function is continuous and linear. So $f_b(\phi(\bigcup S)) \subseteq \phi f_b(\bigcup S)$ for an element $b \in y$. Therefore $\{a + b | a \in \phi(\bigcup S), b \in y\} = \bigcup\{f_b(\phi(\bigcup S)) | b \in y\} \subseteq \{\phi f_b(\bigcup S) | b \in y\} \subseteq \phi\{a + b | a \in \bigcup S, b \in y\}$.

□

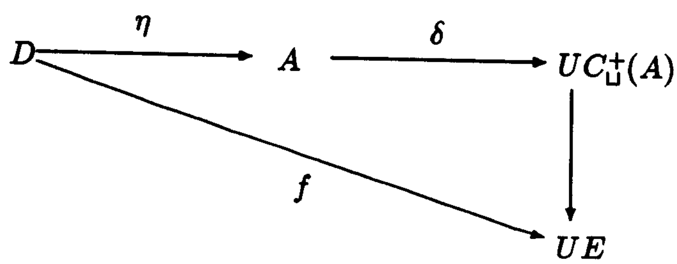
Define $\eta' : D \to W_D$ by $\eta'(a) := \{b \in D | b \leq a\}$, then η' is continuous and linear. Perhaps $C^+_\sqcup$ is isomorphic to the $\sqcup$+-closure of $\eta'(D)$ in W_D. Because $Dcpo \to Pcpo$ and $Dcpo^+ \to Dcpo$ do have left-adjoints it follows that a left-adjoint of $Dcpo^+ \to Pcpo$ exists.

Theorem 4.2.5 *The forgetful functor $Dcpo^+ \to Pcpo$ has a left-adjoint $K : Pcpo \to Dcpo^+$.*

As already showed we have that if $D \in Dcpo$ then $K(D) \cong P(D)$.

Theorem 4.2.6 *Let $D \in Pcpo$, η the unit of K and A the +-closure of $\eta(D)$ in $K(D)$, then $K(D) \cong C^+_\sqcup(A)$.*

Proof: Consider the following diagram:



where $E \in Dcpo^+$, δ the unit of $C^+_\sqcup$ and U the forgetful functor $Dcpo^+ \to Pcpo$. We show first that everything in the diagram is well-defined.
η is a function $D \to UK(D)$, so we can restrict its image to A, because $\eta(D) \subseteq A$. $A \in Dcpo^+$, because the +-operator is continuous in A, so $C^+_\sqcup(A)$ is well-defined.
We now prove that $\delta \circ \eta$ is universal. First $\delta \circ \eta$ is an arrow in $Pcpo$, because both η and δ are continuous and so is their composition. Given a continuous

$f : D \to UE$ there is a unique continuous, linear $f_1 : K(D) \to E$ such that $f = Uf_1 \circ \eta$. This function cuts down to a function $f_1 : A \to E$. So there is a unique continuous, linear $f_2 : C^+_{\sqcup}(A) \to E$ such that $f_1 = f_2 \circ \delta$. So we have $f = Uf_2 \circ \delta \circ \eta$. Thus there is a unique function which makes the diagram above commute. It follows that $\delta \circ \eta$ is a universal arrow. But then $C^+_{\sqcup}(A) \cong K(D)$.

□

It will turn out that $A \cong F(D)$, with F the left-adjoint of $Pos^+ \to Pos$. But, before we can give a proof of this, we need some theory about *order-creating* functions.

Definition 4.2.3 *A function $f : A \to B$ between posets is* order-creating (o.c.) *iff $x \leq y \Leftrightarrow f(x) \leq f(y)$.*

Note that an order-creating function is order-preserving (monotone).

Theorem 4.2.7 *If a function $f : A \to B$ between posets is o.c., then f is injective.*

Proof: Trivial.

□

Theorem 4.2.8 *If a function $f : A \to B$ between posets is o.c. and surjective, then f is continuous.*

Proof: Suppose $f(\sqcup S) \neq \sqcup\{f(x)|x \in S\}$, then there is an y in the range of f such that y is an upperbound of $f(S)$ and $f(\sqcup S) \not\leq y$. Because f is surjective there is an $x \in S : f(x) = y$. So $\sqcup S \not\leq x$, and x is an upperbound of S. This gives a contradiction with the fact that $\sqcup S$ is the lub of S.

□

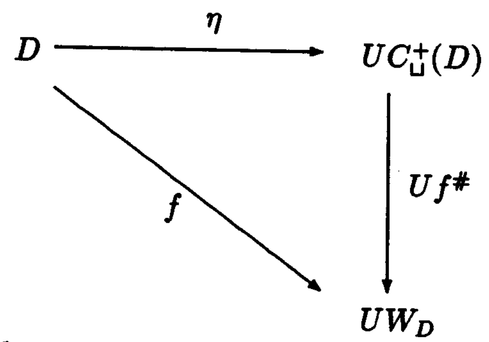
Theorem 4.2.9 *If a function $f : A \to B$ between posets is o.c. and surjective, then f has a continuous inverse.*

Proof: By theorem 4.2.7 f is injective and so it is bijective. Therefore f has an inverse f^{-1}. It is easy to verify that f^{-1} is o.c. too. So by surjectivity of f^{-1} and by theorem 4.2.8 it follows that f^{-1} is continuous.

□

Theorem 4.2.10 *The unit of $C^+_{\sqcup}$ is o.c.*

Proof: Let η be the unit of $C_{\sqcup}^{+}$. It is trivial that η is monotone, for f is continuous. Suppose $x \not\leq y$ and $\eta(x) \leq \eta(y)$. Consider the following diagram:

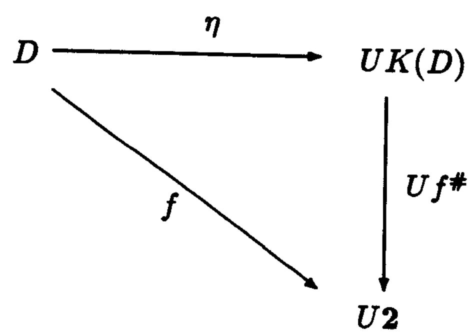


where W_D is the nondeterministic dcpo of theorem 4.2.4, and $f(a) = \{b \in D | b \leq a\}$. The order on W_D is inclusion. So if $x \not\leq y$ we have $f(x) \not\subseteq f(y)$. Now there does not exist a unique continuous, linear function $f^{\#}$ which make the diagram above commute, for $\eta(x) \leq \eta(y)$ and $f^{\#}(\eta(x)) = f(x) \not\subseteq f(y) = f^{\#}(\eta(y))$, so $f^{\#}$ can not be monotone. This gives a contradiction so $f(x) \leq f(y) \Rightarrow x \leq y$.

□

Theorem 4.2.11 *The unit of K is o.c.*

Proof: Let $\mathbf{2} = \{\bot, \top\}$, $\bot \leq \top$ and $\bot + \top = \top$ then $\mathbf{2} \in Dcpo^{+}$. Let η be the unit of K, $D \in Pcpo$.
Suppose $x, y \in D$, $x \not\leq y$ and $\eta(x) \leq \eta(y)$. Let $f : D \to U\mathbf{2}$ be the function defined by $f(z) = \bot$ if $z \leq y$, and $f(z) = \top$ otherwise. for $z \in D$. This function f is continuous, so there is a unique continuous, linear function $f^{\#} : K(D) \to \mathbf{2}$ which makes the following diagram commute.



However this function $f^{\#}$ can not be monotone, because $Uf^{\#}(\eta(x)) = f(x) = \top \geq \bot = f(y) = Uf^{\#}(\eta(y))$, but $\eta(x) \leq \eta(y)$. So we have a contradicton, and it follows that $f(x) \leq f(y)$ implies $x \leq y$.
The reverse is trivial for η is continuous.

□

It would be nice if we could show that $F : Pos \to Pos^+$ extends to a left-adjoint of the forgetful functor $Pcpo^+ \to Pcpo$. The following two theorems seem to lead into this direction.

Theorem 4.2.12 *The unit of F is continuous.*

Proof: Suppose $S \subseteq D$ directed, and $\sqcup S$ exists for $D \in Pos$. Let $U \in F(D)$ be an upperbound of $\eta(S)$, with η the unit of F. Then for each $x \in S, a \in U : x \leq a$, so U contains upperbounds of S. Now we have that $\forall a \in U : \sqcup S \leq a$, so $\eta(\sqcup S) = \{\sqcup S\} \leq U$.

□

Theorem 4.2.13 *For $D \in Pos$, the +-operator is continuous in F(D).*

Proof: Suppose $S \subseteq D$ directed, and $\sqcup S$ exists. We prove that $\sqcup(S) + y$ is the lub of $S' := \{x + y | x \in S\}$, with $y \in F(D)$.
Let u be an upperbound of S'. We show that there is a $v \subseteq u$, such that v is an upperbound of S.
Suppose it is not so, then $\forall v' \subseteq u \exists x \in S$ such that $x \not\sqsubseteq_C v$. Define a set $R := \{x \in S | \forall v' \subseteq u \exists x \in R : x \not\sqsubseteq_C v'\}$. Then $R \subseteq S$ and R is finite, because u is finite. So there is an upperbound x' of R in S (for S is directed), and $x' + y \sqsubseteq_C u$, so by the properties of $\sqsubseteq_C$ we have $\exists v \subseteq u$ such that $x \sqsubseteq_C x' \sqsubseteq_C v$. Contradicton.
Now take $v := \cup\{v \subseteq u | v$ is an upperbound of S$\}$. Then $\sqcup S \sqsubseteq_C v$, so $\sqcup(S) + y \sqsubseteq_C v + y \sqsubseteq_C u$.

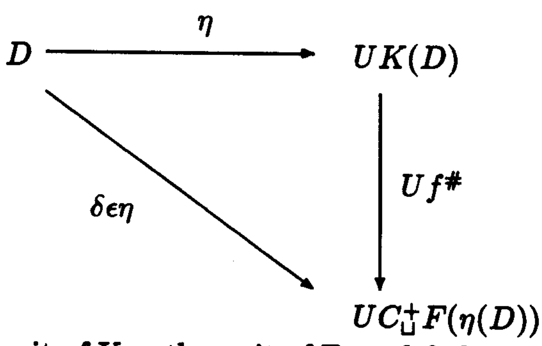
□

However it is difficult to show directly that the unique function $f^\# : F(D) \to E$ induced by a continuous $f : D \to E$ is continuous. We already know that $K(D) \cong C^+_\sqcup(A)$ (theorem 4.2.6), and now we will prove that $F(D) \cong A$. It then easily follows that F is a left-adjoint of the forgetful functor $Pcpo^+ \to Pcpo$.

Definition 4.2.4 *Let $D \in Pcpo$ and η the unit of K, then define a function $\alpha : F(\eta(D)) \to A$ as $\alpha(\{\eta(d_1), \ldots, \eta(d_n)\}) = \eta(d_1) + \cdots + \eta(d_n)$.*

Lemma 4.2.1 *α is o.c.*

Proof: It is easy to see that α is monotone.
Now suppose $x = \{\eta(d_1), \ldots, \eta(d_n)\} \not\sqsubseteq_C \{\eta(d'_1), \ldots, \eta(d'_m)\} = x'$ and $y = \alpha(x) \leq \alpha(x') = y'$. Consider the following diagram:



where η is the unit of K, ϵ the unit of F, and δ the unit of $C^+_\sqcup$. According to theorem 4.2.13 the +-operation is continuous in F(D), $D \in Pos$. $\eta(D) \in Pos$ so $C^+_\sqcup F(\eta(D))$ is well-defined. η, δ are continuous because they are arrows in *Dcpo* and *Pcpo*$^+$ resp. , and ϵ is continuous by theorem 4.2.12. So the composition $\delta \circ \epsilon \circ \eta$ is continuous as well.
Now there exists a unique continuous, linear $f^\#$ which makes commute the diagram above. We have $f^\#(y) = f^\#(\eta(d_1) + \cdots + \eta(d_n)) = f^\#(\eta(d_1)) + \cdots + f^\#(\eta(d_n)) = \delta \circ \epsilon \circ \eta(d_1) + \cdots + \delta \circ \epsilon \circ \eta(d_n) = \delta(\{\eta(d_1)\}) + \cdots + \delta(\{\eta(d_n)\})$, because $\epsilon = \{\cdot\}$. Because of the fact that δ is linear the last expression is equal to $\delta(\{\eta(d_1)\} + \cdots + \{\eta(d_n)\} = \delta(\{\eta(d_1), \ldots, \eta(d_n)\})$. Analogous we have $f^\#(y') = \delta(\{\eta(d'_1), \ldots, \eta(d'_m)\})$. Now $y \leq y'$, so since $f^\#$ is monotone we must have $f^\#(y) \leq f^\#(y')$. But $\{\eta(d_1), \ldots, \eta(d_n)\} \not\sqsubseteq_C \{\eta(d'_1), \ldots, \eta(d'_m)\}$, so since δ is o.c. (theorem 4.2.10) it follows that $f^\#(y) = \delta(\{\eta(d_1), \ldots, \eta(d_n)\}) \not\leq \delta(\{\eta(d'_1), \ldots, \eta(d'_m)\})$. Contradiction. So $\alpha(x) \leq \alpha(x')$ implies $x \leq x'$.

□

Lemma 4.2.2 *α is surjective, continuous and linear.*

Proof: Take $y \in A$, then $y = \eta(d_1) + \cdots + \eta(d_n)$. Now let $x = Minmax\{\eta(d_1), \cdots, \eta(d_n)\}$ then $x \in F(\eta(D))$ and $\alpha(x) = y$. So α is surjective.
By theorem 4.2.8 and the fact that α is surjective and o.c. it follows that α is continuous.
Finally it is easy to verify that α is linear.

□

Theorem 4.2.14 *Let η be the unit of K, then $F(\eta(D)) \cong A$.*

Proof: α is an arrow in *Pcpo*$^+$ by the previous lemma. By theorem 4.2.9 and lemma 4.2.2 α has a continuous linear inverse α^{-1}, and it is easy to verify that α^{-1} is linear.

□

Definition 4.2.5 *Let $D \in Pcpo$ and η the unit of K, then define a function $\beta : F(D) \to F(\eta(D))$ as follows: $\beta(\{d_1, \ldots, d_n\}) = \{\eta(d_1), \ldots, \eta(d_n)\}$.*

Lemma 4.2.3 *β is o.c.*

Proof: Trivially β is monotone (because η is monotone).
Suppose $x = \{d_1, \ldots, d_n\} \not\sqsubseteq_C \{d'_1, \ldots, d'_m\} = x'$ and $\beta(x) \sqsubseteq_C \beta(x')$. Because $x \sqsubseteq_C x'$ there is a $d'_i \in x'$ such that $\forall d'_j \in x' : d_i \not\leq d'_j$, or there is a $d'_i \in x'$ such that $\forall d_j \in x : d_j \not\leq d'_i$. In both cases $f(x) \not\sqsubseteq_C f(x')$ for η is o.c. (theorem 4.2.11).

□

Lemma 4.2.4 *β is surjective, continuous and linear.*

Proof: Left to the reader.

□

Lemma 4.2.5 *Let $D \in Pcpo$ and η the unit of K, then $F(D) \cong F(\eta(D))$.*

Proof: Left to the reader.

□

Corollary 4.2.1 *$K(D) \cong C^+_\sqcup F(D)$, for $D \in Pos$.*

Corollary 4.2.2 *The forgetful functor $Pcpo \to Pcpo^+$ has as left-adjoint F.*

So what we in fact have done was to extend *Dcpo* to a category *Pcpo* and find left-adjoints $F : Pcpo \to Pcpo^+$ and $C^+_\sqcup : Pcpo^+ \to Dcpo^+$. The composition of these two functors is isomorphic to the left-adjoint $K : Pcpo \to Dcpo^+$, and if K is restricted to *Dcpo* it is isomorphic to the left-adjoint $P : Dcpo \to Dcpo^+$.

So the solution of the problem lies not in the considering of subcategories of *Dcpo*, but on the contrary in the extension of *Dcpo*. Also note that it was essential that we knew that a left-adjoint of $Dcpo^+ \to Pcpo$ existed.

The result of this section is that the powerdomain P(D) over a dcpo D is in essence isomorfic to F(D). The only thing we have to do to make F(D) completely isomorphic to P(D), is to add in a certain manner lubs of directed subsets to F(D).

Chapter 5

5.1 A (lower) powerdomain for countable nondeterminism

Powerdomains with an *or*-operation which can take an infinite number of arguments, are much more difficult to find than bounded nondeterministic powerdomains. In [2] a construction is given for upper and convex countable powerdomains generated by flat posets. In this section we will see that lower bounded powerdomains are easily extended to lower countable powerdomains over arbitrary dcpo's.

First define $Dcpo_L^+$ as the subcategory of $Dcpo^+$ such that for every object D of it the following holds: $\forall x, y \in D : x \leq x + y$.

Theorem 5.1.1 *The forgetful functor* $Dcpo_L^+ \to Dcpo$ *(which "forgets" the +-operation) has a left-adjoint* $P_L : Dcpo \to Dcpo_L^+$.

Proof: See [11].

□

The functor P_L constructs for every algebraic dcpo the lower powerdomain of chapter 2.

Now for countable nondeterminism we need an operation $\Sigma : D^\omega \to D$ which is commutative, associative and idempotent. Because of the idempotency Σ can be used for finite nondeterminism too, for example we can define $x + y := \Sigma xyyyy\ldots$.

Define a new category $Dcpo_L^\Sigma$ as follows: the objects of $Dcpo_L^\Sigma$ are dcpo's D with an operator $\Sigma : D^\omega \to D$ which is continuous, and such that $\{\overline{x}_i | i \in N\} \subseteq \{\overline{y}_i | i \in N\}$ implies $\Sigma\overline{x} \leq \Sigma\overline{y}$, for $\overline{x}, \overline{y} \in D^\omega$. The arrows in $Dcpo_L^\Sigma$ are all linear an continuous functions between its objects. Clearly $Dcpo_L^\Sigma$ is an extened version of $Dcpo_L^+$.

For every $\overline{x} \in D^\omega \in Dcpo_L^\Sigma$ define a set $S_{\overline{x}}$ as follows: $S_{\overline{x}} := \{\Sigma\overline{y} | \overline{y} \in D^\omega\ and\ \forall\overline{y}\exists n \in N : \forall i \leq n : \overline{x}_i = \overline{y}_i and \forall i > n : \overline{y}_i = \overline{x}_n\}$. It is easy to verify that $S_{\overline{x}}$ is directed and that $\Sigma\overline{x} = \sqcup S_{\overline{x}}$.

Consider the forgetful functor $Dcpo_L^\Sigma \to Dcpo_L^+$ which forgets the sums of infinite different elements.

Theorem 5.1.2 *The forgetful functor $Dcpo_L^{\Sigma} \to Dcpo_L^{+}$ has a left-adjoint P_L^{Σ} : $Dcpo_L^{+} \to Dcpo_L^{\Sigma}$.*

Proof: Let D be an object of $Dcpo_L^{+}$, then $P_L^{\Sigma}(D) = D$. The order and lubs of $P_L^{\Sigma}(D)$ are the same as those of D.
Define an operator $\Sigma : (P_L^{\Sigma}(D))^{\omega} \to P_L^{\Sigma}(D)$ as follows: $\Sigma\overline{x} = \sqcup\{y_1 + \cdots + y_n | n \in N \textit{ and } \forall y_j \exists i : y_j = \overline{x}_i\}$.

□

Theorem 5.1.3 *The forgetful functor $Dcpo_L^{\Sigma} \to Dcpo$ has a left-adjoint.*

Proof: We have the left-adjoint $P_L : Dcpo \to Dcpo_L^{+}$ and $P_L^{\Sigma} : Dcpo_L^{+} \to Dcpo_L^{\Sigma}$. By theorem 1.3.4 $P_L^{\Sigma} P_L$ is a left-adjoint of this theorem.

□

So for countable nondeterminism we can use the same lower powerdomain construction as for bounded nondeterminism, except that we have an *or*-operation which is defined for countable many arguments.

Acknowledgement : I thank Jan van Leeuwen for suggesting powerdomains as a fruitful subject for study and for useful comments on a draft version of this paper.

Bibliography

[1] Abramsky S.
On semantic foundations for applicative multiprogramming,
in: J. Diaz (ed), Automata, Languages and Programming, 10th Colloquium, Lect. Notes in Comp. Sc., vol 154, pp 1-14, Springer-Verlag, Heidelberg,1983 Proc. ICALP '83, LNCS 154

[2] Apt K.R., Plotkin G.D.
A Cook's tour of countable nondeterminism,
in: S. Even, O. Kariv (ed), Automata, Languages and Programming, 8th Colloquium, Lect. Notes in Comput. Sci., vol 115, pp 479-494, Springer-Verlag, Heidelberg,1981

[3] Banaschewski B., Nelson E.
Completions of partially ordered sets,
SIAM J. Comput. 11 (1982), 521-528

[4] Broy M.
A fixed point approach to applicative multiprogramming,
Theoretical Foundations of Programming Methodology, D. Reidel Publishing Company, Dordrecht, 1982, pp 565-622

[5] Broy M.
On the Herbrand-Kleene universe for nondeterministic computations,
Theor. Comput. Sci. 36 (1985), 1-19

[6] Escrig D.F.
A characterization of Plotkin's order in powerdomains, and some of its properties,
Theor. Comput. Sci. 31 (1984), 73-82

[7] Francez N. et al.
Semantics of Nondeterminism, Concurrency, and Communication,
J. Comput. Syst. Sci. 19 (1979), 290-308

[8] Goguen J.A., Thatcher J.W., Wagner E.G., Wright J.B.
Initial algebra semantics and continuous algebras,
J. Assoc. Comput. Mach. 24 (1977), 68-95

[9] Hennessy M.C.B.
Powerdomains and nondeterministic recursive definitions,
in: M. Derani-Ciancaglini, U. Montanari (ed), International Symposium on programming, 5th Colloquium, Lect. Notes in Comp. Sci., vol 137, pp 178-193, Springer-Verlag, Heidelberg, 1982

[10] Hennessey M.C.B., Ashcroft E.A.
A mathematical semantics for a nondeterministic typed λ-calculus,
Theor. Comput. Sci. 11 (1980), 227-245

[11] Hennessey M.C.B., Plotkin G.D.
Full abstraction for a simple parallel programming language,
in: J. Becvar (ed), Foundations of Computer Science, Lect. Notes in Comp. Sci., vol 74, pp 108-120, Springer-Verlag, Heidelberg, 1979

[12] Lehmann D.J.
Categories for fixpoint semantics,
in: Proceedings 17th IEEE Symposium on Foundations of Computer Science (1976), pp 122-126

[13] Maclane S.
Categories for the Working Mathematician,
Springer-Verlag, New York, 1971

[14] Milne G., Milner R.
Concurrent processes and their syntax,
J. Assoc. Comput. Mach. 26 (1979), 302-321

[15] Nelson E.
Recent results on continuous ordered algebras,
in: L. Budach (ed), Fundamentals of Computation Theory, Lect. Notes in Comp. Sci., vol 199, pp 320-330, Springer-Verlag, Heidelberg, 1985

[16] Plotkin G.D.
A powerdomain for countable nondeterminism,
in: Automata, Languages and Programming, 9th Colloquium, Lect. Notes in Comput. Sci., vol 140, pp 418-428 Springer-Verlag, Heidelberg, 1982

[17] Poigne A.
On effective computations of nondeterministic schemes,
in: M. Derani-Ciancaglini, U. Montanari (ed), International Symposium on programming, 5th Colloquium, Lect. Notes in Comp. Sci., vol 137, pp 323-336 Springer-Verlag, Heidelberg, 1982

[18] Poigne A.
On algebras of computation sequences and proofs of equivalence of operational and denotational semantics,