Set Domains

Reinhold Heckmann

FB 14 – Informatik
Universität des Saarlandes
6600 Saarbrücken
Bundesrepublik Deutschland

email: heckmann@cs.uni-sb.de

January 30, 1990

Abstract

Set domains are intended to give semantics to a data type of sets together with a wide range of useful set operations. The classical power domain constructions are shown to be inappropriate for this purpose. Lower and upper domain do not support quantification, whereas Plotkin's domain does not contain the empty set. This is an immense defect, since the empty set is not only interesting in its own, but is also needed to define operations such as filtering a set through a predicate. Two constructions, the *big* and the *small set domain*, are proposed that support the desired set operations. The big domain is bounded complete, whereas the small one only respects Plotkin's SFP-property. Both constructions are free with respect to suitable algebraic theories.

1 Introduction

Following the Scott-Strachey approach [12] to the semantics of programming languages, the various types occurring in a functional programming language are associated with corresponding domains of semantic entities. For instance, tuple or record types and functional types correspond to product and exponentiation of domains respectively. Originally, there was no domain construction corresponding to set types over a given ground type.

In 1976, Plotkin [8] proposed a power domain construction, analogous to power set forming, to describe the semantics of non-deterministic programming languages. Because his construction goes beyond the category of bounded complete (Scott) domains, Plotkin proposed the larger category of *SFP-domains* that is closed under his construction. A short time later, Smyth [9] introduced a simpler construction, the upper or Smyth power domain, that respects bounded completeness. In [10], a third power domain construction occurs, the lower power domain, that completes the triumvirate of classical power domain constructions.

For a lazy functional programming language, we want to introduce an abstract data type of sets with some primitive operations that allow to derive a wide range of useful other operations, including some of higher order. As outlined in section 3, the set data type should support union, filtering a set through a predicate, and checking whether a predicate holds for some or all elements of a set (quantification). The semantics of the set data type should be given by means of a suitable 'set domain' construction.

In section 4, we check whether some of the classical power domains could serve as set domain, i.e. satisfy the specification given in section 3. All three power domains have an important defect: they do not contain a representation of the empty set. The empty set is not only interesting in its own. Its lack also makes important operations, such as filtering a set through a predicate, impossible to define. Adjoining the empty set to lower and upper power domain is simple, but both of them support quantification in a half-hearted manner: they only say 'yes', never 'no', or vice versa. Plotkin's power domain better supports quantification, but there is no simple way to extend it by the empty set.

Thus, we look for new constructions. The fact that lower and upper domain support opposite halves of quantification suggests to combine them to their product after extension by the empty set. Certain conditions make stand out two subsets of this product, the small and the big set domain which support all operations we want to have. The small domain is a subset of the big one, and it still contains isomorphic images of the three classical power domains. The big set domain introduced in section 5 is bounded complete, hence it is useful in the context of Scott domains. Similar to Plotkin's domain, the construction of the small set domain described in section 6 leads out of the scope of Scott domains, but respects Plotkin's SFP-property.

All members of the base of the small set domain may be built up by finite applications of our primitive set operations. This is not true for the big domain. An equality on sets may be derived from the primitive set operations and a given equality on the ground domain. If the given equality is the greatest continuous equality on the ground domain, then the set equality is again the greatest continuous one on the small set domain (see section 7). This again is not true for the big one.

Starting from problems in data base theory, Buneman et al. [1] also proposed to combine lower and upper power domain to a so-called sandwich power domain. In contrast to me, they left off the empty set and did not investigate their construction in detail.

Gunter investigated the logic of the classical power domains [2]. By extending the logic of Plotkin's domain in a natural way, he developed a so-called mixed power domain. It is a subset from our small set domain, but inherits from Plotkin's domain the lack of the empty set. In the last few months, Gunter further developed the theory of the mixed power domain [3, 4]. In particular, he added the empty set and arrived exactly at what we call the small set domain. In parallel independent work, Gunter and I developed the notion of a mix algebra and showed that the small set domain is a free construction w.r.t. the mix theory (see section 8). I found the additional result that the big set domain is also a free construction w.r.t. an algebraic theory which however is less natural than the mix theory (see section 9).

2 Domain-theoretic background

We start off with the necessary domain-theoretic background. We only give definitions and facts. For proofs, we refer to the literature or to an elaborated version of this paper [5].

A *poset* (partially ordered set) $(P, \leq)$ is a set P together with a reflexive, antisymmetric, and transitive relation '$\leq$'. When there is no risk of confusion, we identify the poset $\mathbf{P} = (P, \leq)$ with its carrier P. We refer to the standard notions of minimal and maximal elements, upper and lower bounds, bounded subsets, least upper bound (lub) denoted by '$\bigsqcup$', greatest lower bound (glb), directed set, directed complete poset (dcpo), monotonic and continuous function.[1] Two points x and y are consistent – $x \uparrow y$ – if they have a common upper bound.

A set $A \subseteq \mathbf{P}$ is a *lower* set iff with any point a in A, all points *below* a belong to A. It is an *upper* set iff with any point a in A, all points *above* a belong to A. It is *convex*, iff with any points a and c in A, all points *between* a and c belong to A. For $A \subseteq \mathbf{P}$, let $\downarrow A = \{y \in \mathbf{P} \mid \exists x \in A : x \geq y\}$ be the lower closure of A, and $\uparrow A = \{y \in \mathbf{P} \mid \exists x \in A : x \leq y\}$ the upper closure. $\downarrow A$ is the least lower superset of A, and $\uparrow A$ the least upper one. We use the abbreviations $\downarrow x = \downarrow\{x\}$ etc.

A point a in a directed complete poset $\mathbf{P}$ is *isolated* (or: *finite*), iff for all directed sets $D \subseteq \mathbf{P}$ with $a \leq \bigsqcup D$, there is an element d in D such that $a \leq d$. The set of all isolated points of $\mathbf{P}$ is called $\mathbf{P}^0$. We use the name 'isolated' in order to avoid the name conflict with finite sets. Nevertheless, finite sets of isolated (finite) elements are called *bifinite*. A directed complete poset $\mathbf{P}$ is *algebraic*, iff every point of $\mathbf{P}$ is the lub of a directed set of isolated points. The set $\mathbf{P}^0$ of all isolated points of $\mathbf{P}$ is called the *base*. A *domain* is an algebraic poset whose base is countable, and which contains a least element $\perp$. The *iso-lower closure* $\Downarrow A$ of a subset A of an algebraic dcpo $\mathbf{P}$ is the set of all isolated points below some member of A, i.e. $\Downarrow A = \downarrow A \cap \mathbf{P}^0$. If $\Downarrow A$ equals A, i.e. if A is a lower subset of the base, then A is called *iso-lower* set.

A domain has property M [6, 8], if for any bifinite set E, there is a finite set F of upper bounds of E with the property, that there is a member of F below every upper bound of E. Generally, there exist different finite sets F with this property for a given bifinite set E. Among those, there is a least one w.r.t. set inclusion. It is bifinite and consists of the *minimal upper bounds (mubs)* of E. A domain has property SFP [8], if it has property M, and every bifinite set E has a bifinite superset G that is closed w.r.t. mubs of subsets. A thorough discussion of these properties may be found in [6]. A domain is bounded complete if every bounded subset has a lub, or equivalently if every two bounded isolated points have a lub. It is complete if every subset has a lub.

Thus, a hierarchy of domains is introduced. Complete domains are bounded complete, the latter have property SFP, and SFP implies M. Membership in these domain classes is preserved by sum and product. Exponentiation, i.e. forming the space of continuous functions, only preserves (bounded) completeness and SFP, but neither M nor the property to be a plain domain [11].

The Lawson topology

In this section, we introduce the Lawson topology [7, 8] for a fixed underlying M-domain $\mathbf{D}$. The closed sets of this topology have many pleasing mathematical properties. They may be used to define the classical power domains.

Following Plotkin [8], we define the Lawson topology[2] by its sub-basis: the collection of all cones $\uparrow a$ for isolated a and of their complements. The topological closure of a set A is denoted by $\overline{A}$. It is the least closed superset of A. All finite sets are closed. The Lawson topology is homeomorphic to a compact [7] metric [5] space.

The set $\mathcal{C}\mathbf{D}$ of Lawson closed sets of a given M-domain $\mathbf{D}$ forms a complete domain when ordered by '$\supseteq$' [5]. Its isolated points are the sets being both closed and open. All lubs exist, lub is intersection, glb is set union followed by closure, the least element is $\mathbf{D}$, and the greatest one is $\emptyset$. The useful mathematical properties of closed sets are summarized in the following theorem [5]:

[1] w.r.t. directed sets, not ascending sequences.

[2] called Cantor topology by Plotkin

Theorem 2.1 (Operations in the Lawson domain)
For domains $\mathbf{D}$, $\mathbf{X}$, and $\mathbf{Y}$, the following operations are continuous:

Union:	$\cup : \mathcal{C}\mathbf{D} \times \mathcal{C}\mathbf{D} \to \mathcal{C}\mathbf{D}$	
Intersection:	$\cap : \mathcal{C}\mathbf{D} \times \mathcal{C}\mathbf{D} \to \mathcal{C}\mathbf{D}$	
Upper cone:	$\uparrow : \mathbf{D} \to \mathcal{C}\mathbf{D}$	
Upper closure:	$\uparrow : \mathcal{C}\mathbf{D} \to \mathcal{C}\mathbf{D}$	
Lower closure:	$\downarrow : \mathcal{C}\mathbf{D} \to \mathcal{C}\mathbf{D}$	
Product:	$\times : \mathcal{C}\mathbf{X} \times \mathcal{C}\mathbf{Y} \to \mathcal{C}(\mathbf{X} \times \mathbf{Y})$	
Extension:	$ext : (\mathbf{X} \to \mathcal{C}\mathbf{Y}) \to (\mathcal{C}\mathbf{X} \to \mathcal{C}\mathbf{Y})$	with $ext\, f\, A = \bigcup_{a \in A} fa$

Note that a lot of information is contained in this theorem, e.g. 'if A is closed, then $\downarrow A$ and $\uparrow A$ are closed', '$\uparrow(\bigcap_{A \in D} A) = \bigcap_{A \in D} \uparrow A$ for all directed collections D', 'if A is closed and f is continuous and produces only closed sets, then $\bigcup_{a \in A} fa$ is closed' etc.

3 Specification of set domains

3.1 Introduction

In this section, we state the properties we expect from a set domain construction. It is a trial to formalize some intuitive requirements, and hence, it might seem a bit vague. However, in sections 5 and 6, we rigorously construct the big and the small set domain and show later that they both meet the specification. This also implies that the given specification is not exact enough to describe a unique domain construction. Besides small and big set domain, there might be further constructions which meet the specification.

Assume an M-domain $\mathbf{D}$, the ground domain, is given. We want to specify a domain $\mathcal{S}\mathbf{D}$ of sets over $\mathbf{D}$ that supports a broad collection of set operations. The set domains over different ground domains $\mathbf{X}$ and $\mathbf{Y}$ are not independent, because some set operations involve both $\mathcal{S}\mathbf{X}$ and $\mathcal{S}\mathbf{Y}$. Thus, we should speak of a set domain construction $\mathbf{D} \mapsto \mathcal{S}\mathbf{D}$ instead of speaking of a single set domain. For the sake of brevity, we often omit the word 'construction' hoping that the reader nevertheless understands what we mean. The symbol '$\mathcal{S}$' is to be understood as generic; we denote the small set domain (construction) by 'σ' and the big one by 'Σ' in the later parts of this article.

To have more freedom in their construction, we do not require that set domains really consist of sets. We call the elements of a set domain *formal* sets in contrast to the actual, mathematical sets. Naturally, there is some relation between formal and actual sets as outlined below.

The operations on formal sets usually differ from the corresponding mathematical operations on actual sets. Thus, a distinction is needed. We adopt the convention to mark the formal set operations by means of additional horizontal or vertical bars.

3.2 Empty set and finite union

As a first requirement, we want the set domain $\mathcal{S}\mathbf{D}$ to contain a formal empty set and a formal set union.

Definition 3.1 (CAIN domains and linear maps)
A *CAIN poset* $(\mathbf{P}, \bar{\cup}, \bar{\emptyset})$ is a poset $\mathbf{P}$ together with a monotonic map $\bar{\cup} : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$ being

Commutative, Associative, and Idempotent (i.e. $A \bar{\cup} A$ for all $A \in \mathbf{P}$), and an element $\bar{\emptyset}$ of $\mathbf{P}$ which is the Neutral element of ' $\bar{\cup}$ '.

A *CAIN domain* $(\mathbf{P}, \bar{\cup}, \bar{\emptyset})$ is a CAIN poset where $\mathbf{P}$ is a domain and $\bar{\cup}$ is continuous.

A map $f : X \to Y$ between CAIN posets is *linear* iff it is a homomorphism w.r.t. the CAIN theory, i.e. $f\bar{\emptyset}_X = \bar{\emptyset}_Y$ and $f(A \bar{\cup} B) = fA \bar{\cup} fB$ hold. □

As usual in algebra, we identify CAIN posets with their carrier. If $\mathbf{X}$ is an arbitrary domain and $\mathbf{Y}$ is a CAIN domain, then the function space $X \to Y$ becomes a CAIN domain in a natural way by defining $\bar{\emptyset} = \lambda x.\, \bar{\emptyset}_Y$ and $f \bar{\cup} g = \lambda x.\, fx \bar{\cup} gx$.

Returning to set domains, our first requirement for the set domain is that it be a CAIN domain.

3.3 Singleton sets

As a next requirement, there should be a function which maps elements into singleton sets. We require this function $\{\!|.|\!\} : \mathbf{D} \to \mathcal{S}\mathbf{D}$, $x \mapsto \{\!|x|\!\}$ not only to be continuous, but also to be injective, i.e. $x \neq y$ has to imply $\{\!|x|\!\} \neq \{\!|y|\!\}$. An additional condition is that the empty set is not a singleton, i.e. $\{\!|x|\!\} \neq \bar{\emptyset}$ for all $x \in \mathbf{D}$.

As we shall see, the singleton map for the big and also the small set domain are even embeddings, i.e. $x \leq y$ iff $\{\!|x|\!\} \leq \{\!|y|\!\}$ holds. However, we do not require this for general set domains explicitly.

By means of the operations $\bar{\emptyset}$ and $\bar{\cup}$, we may extend $\{\!|.|\!\}$ to finite subsets of the ground domain thus relating actual with formal sets:

$$\{\!|x_1, \ldots, x_n|\!\} = \begin{cases} \{\!|x_1|\!\} \bar{\cup} \cdots \bar{\cup} \{\!|x_n|\!\} & \text{if } n > 0 \\ \bar{\emptyset} & \text{if } n = 0 \end{cases}$$

3.4 Function extension

Next, we require that functions defined on elements of the ground domain may be raised to functions defined on formal sets. More precise, we want a continuous second order functional $ext : (\mathbf{X} \to \mathcal{S}\mathbf{Y}) \to (\mathcal{S}\mathbf{X} \to \mathcal{S}\mathbf{Y})$ for every two domains $\mathbf{X}$ and $\mathbf{Y}$. For every continuous map $f : \mathbf{X} \to \mathcal{S}\mathbf{Y}$, the extended map $ext\, f$ should be linear and extend f properly to sets, i.e. $ext\, f\, \{\!|x|\!\} = fx$ has to hold. These axioms imply $ext\, f\, \{\!|x_1, \ldots, x_n|\!\} = fx_1 \bar{\cup} \cdots \bar{\cup} fx_n$ for $n > 0$. Furthermore, we want ext to be linear in its functional argument, i.e. $ext\, (\lambda x.\, fx \bar{\cup} g\, x)\, S = (ext\, f\, S) \bar{\cup} (ext\, g\, S)$.

Given the ext functional as a primitive operator of a set data type allows us to derive many other useful operations. Some examples are given now:

- $map : (\mathbf{X} \to \mathbf{Y}) \to (\mathcal{S}\mathbf{X} \to \mathcal{S}\mathbf{Y})$
  defined by $map\, f\, A = ext\, (\lambda x.\, \{\!|fx|\!\})\, A$ or equivalently $map\, f = ext\, (\{\!|.|\!\} \circ f)$
  $map\, f$ is a linear map between $\mathcal{S}\mathbf{X}$ and $\mathcal{S}\mathbf{Y}$ with $map\, f\, \{\!|x_1, \ldots, x_n|\!\} = \{\!|fx_1, \ldots, fx_n|\!\}$.

- $union : \mathcal{S}(\mathcal{S}\mathbf{D}) \to \mathcal{S}\mathbf{D}$ defined by $union = ext\, (\lambda x.\, x)$
  It is a linear map with the property $union\, \{\!|S_1, \ldots, S_n|\!\} = S_1 \bar{\cup} \cdots \bar{\cup} S_n$ for $n > 0$, i.e. it is a formal 'big union'.

- $\_ \bar{\times} \_ : \mathcal{S}\mathbf{X} \times \mathcal{S}\mathbf{Y} \to \mathcal{S}(\mathbf{X} \times \mathbf{Y})$
  defined by using ext twice: $A \bar{\times} B = ext\, (\lambda a.\, ext\, (\lambda b.\, \{\!|(a, b)|\!\})\, B)\, A$
  Using the two-fold linearity of ext in its functional and its set argument, one obtains $\{\!|x_1, \ldots, x_n|\!\} \bar{\times} \{\!|y_1, \ldots, y_m|\!\} = \{\!|(x_i, y_j) \mid 1 \leq i \leq n, 1 \leq j \leq m|\!\}$ using an obvious notation analogous to ZF notation for actual sets. Hence, ' $\bar{\times}$ ' denotes a formal Cartesian product.

3.5 Existential quantification

As further operations, we want some set theoretic predicates. In order to derive them, formal sets must somehow be related with Booleans. In doing so, we introduce the Boolean poset $\mathbf{B} = \{\perp, \mathsf{T}, \mathsf{F}\}$ ordered by $\perp \leq \mathsf{T}$ and $\perp \leq \mathsf{F}$. We turn it into a CAIN domain by letting $\bar{\emptyset}$ be F. The CAIN axioms then enforce $\mathsf{F} \mathbin{\bar{\cup}} \mathsf{F} = \mathsf{F}$ and $\mathsf{F} \mathbin{\bar{\cup}} \mathsf{T} = \mathsf{T} \mathbin{\bar{\cup}} \mathsf{F} = \mathsf{T} \mathbin{\bar{\cup}} \mathsf{T} = \mathsf{T}$, i.e. formal union is disjunction. Concerning $\perp$, the equations $\mathsf{F} \mathbin{\bar{\cup}} \perp = \perp \mathbin{\bar{\cup}} \mathsf{F} = \perp \mathbin{\bar{\cup}} \perp = \perp$ have to hold. However, the CAIN axioms don't fix the values of $\mathsf{T} \mathbin{\bar{\cup}} \perp$ and $\perp \mathbin{\bar{\cup}} \mathsf{T}$ except they have to be equal and cannot be F. Thus, there are two choices: the strict disjunction where the result is $\perp$, and parallel disjunction where the result is T. We arbitrarily choose the latter because it is more defined. Henceforth, $\mathbf{B}$ is always understood as CAIN domain equipped with 'parallel or' denoted by '$\vee$'. Sometimes, we also use 'parallel and' denoted by '$\wedge$'. As indicated above, the space of predicates $\mathbf{D} \to \mathbf{B}$ also becomes a CAIN domain by raising disjunction to predicates, i.e. with operations $p \vee q = \lambda x.\, p\, x \vee q\, x$ and $\underline{\mathsf{F}} = \lambda x.\, \mathsf{F}$.

After these preliminaries, we can now state our next requirement for the generic set domain. There has to be a functional for existential quantification which, given a predicate, tells whether some element of a given formal set satisfies it. Quantification is specified analogously to functional extension.[3]

For every domain $\mathbf{D}$, there should be a continuous second order functional $\mathit{exists} : (\mathbf{D} \to \mathbf{B}) \to (\mathcal{S}\mathbf{D} \to \mathbf{B})$ such that for every continuous predicate $p : \mathbf{D} \to \mathbf{B}$, there is a linear predicate $\mathit{exists}\ p$ extending p properly to sets, i.e. $\mathit{exists}\ p\ \{\!|x|\!\} = p\, x$ has to hold. These axioms imply $\mathit{exists}\ p\ \{\!|x_1, \ldots, x_n|\!\} = p\, x_1 \vee \cdots \vee p\, x_n$ for $n > 0$. In addition, we require exists to be also linear in its functional argument.

Given the *exists* functional as a primitive operator of a set data type allows us to derive many other useful predicates, e.g.

- $\mathit{forall} : (\mathbf{D} \to \mathbf{B}) \to (\mathcal{S}\mathbf{D} \to \mathbf{B})$ defined by $\mathit{forall}\ p\ S = \neg(\mathit{exists}\ (\lambda x.\, \neg(p\, x))\ S)$
  This implies $\mathit{forall}\ p\ \{\!|x_1, \ldots, x_n|\!\} = p\, x_1 \wedge \cdots \wedge p\, x_n$ for $n > 0$.
- $\mathit{empty} : \mathcal{S}\mathbf{D} \to \mathbf{B}$ defined by $\mathit{empty}\ S = \neg(\mathit{exists}\ (\lambda x.\, \mathsf{T})\ S) = \mathit{forall}\ (\lambda x.\, \mathsf{F})\ S$
  Hence, we obtain $\mathit{empty}\ \{\!|x_1, \ldots, x_n|\!\} = \begin{cases} \mathsf{T} & \text{if } n = 0 \\ \mathsf{F} & \text{if } n > 0 \end{cases}$

For the tests of containment, inclusion, and set equality, we need a given continuous equality '==' on the ground domain.

- $\_\ \mathit{in}\ \_ :\quad \mathbf{D} \times \mathcal{S}\mathbf{D} \to \mathbf{B} \qquad x\ \mathit{in}\ B = \mathit{exists}\ (\lambda b.\, x == b)\ B$
- $\_ \mathbin{\bar{\subseteq}} \_ :\quad \mathcal{S}\mathbf{D} \times \mathcal{S}\mathbf{D} \to \mathbf{B} \qquad A \mathbin{\bar{\subseteq}} B = \mathit{forall}\ (\lambda a.\, a\ \mathit{in}\ B)\ A$
- $\_ == \_ :\quad \mathcal{S}\mathbf{D} \times \mathcal{S}\mathbf{D} \to \mathbf{B} \qquad (A == B) = (A \mathbin{\bar{\subseteq}} B) \wedge (B \mathbin{\bar{\subseteq}} A)$

We consider this equality check a bit closer in section 7.

3.6 Filtering a set through a predicate

We finally want to provide an operation that filters a set through a given predicate. The operation $\mathit{filter} : (\mathbf{D} \to \mathbf{B}) \to (\mathcal{S}\mathbf{D} \to \mathcal{S}\mathbf{D})$ should be linear in its set argument as well as in its

[3] Later, we shall see that for the two set domains, quantification is a special instance of extension.

predicate argument and operate on singletons as one expects.

$$\textit{filter}\ p\,(A \mathbin{\bar\cup} B) = (\textit{filter}\ p\ A) \mathbin{\bar\cup} (\textit{filter}\ p\ B) \qquad \textit{filter}\ (p \vee q)\ A = (\textit{filter}\ p\ A) \mathbin{\bar\cup} (\textit{filter}\ q\ A)$$

$$\textit{filter}\ p\ \bar\emptyset = \textit{filter}\ \underline{\mathsf{F}}\ S = \bar\emptyset \qquad \textit{filter}\ p\ \{\!|x|\!\} = \begin{cases} \{\!|x|\!\} & \text{if } p\,x = \mathsf{T} \\ \bar\emptyset & \text{if } p\,x = \mathsf{F} \end{cases}$$

Instead of requiring the existence of an operation satisfying these axioms, we derive it from a simpler one using *ext*: $\quad \textit{filter}\ p\ S = \textit{ext}\ (\lambda x.\, x\ \textit{when}\ (p\,x))\ S$

where $\_\ \textit{when}\ \_ : \mathbf{D} \times \mathbf{B} \to \mathcal{S}\mathbf{D}$ is an operation which models the effect of *filter* on singletons. Hence, $x\ \textit{when}\ \mathsf{T} = \{\!|x|\!\}$ and $x\ \textit{when}\ \mathsf{F} = \bar\emptyset$ must hold. With these axioms for '*when*', one obtains linearity of *filter* in the set argument and its correct behavior for singletons. For linearity of *filter* in the predicate, one needs the additional axiom $x\ \textit{when}\ (a \vee b) = (x\ \textit{when}\ a) \mathbin{\bar\cup} (x\ \textit{when}\ b)$, i.e. linearity of '*when*' in its Boolean argument.

Instead of requiring now the existence of an operation '*when*' with these axioms, we go one step further and define it by means of an even simpler operation. In order to define '*when*', only $x\ \textit{when}\ \bot$ has to be fixed, because the values for T and F are already given by the *when*-axioms above. We define

$$x\ \textit{when}\ b = \begin{cases} \{\!|x|\!\} & \text{if } b = \mathsf{T} \\ \{\!|x?|\!\} & \text{if } b = \bot \\ \{\!||\!\} & \text{if } b = \mathsf{F} \end{cases}$$

using a new operation $\{\!|.?|\!\} : \mathbf{D} \to \mathcal{S}\mathbf{D}, x \mapsto \{\!|x?|\!\}$. For monotonicity of '*when*', the new operation must satisfy the laws $\{\!|x?|\!\} \le \{\!|x|\!\}$ and $\{\!|x?|\!\} \le \{\!||\!\} = \bar\emptyset$. For linearity of '*when*', one additionally needs $\{\!|x|\!\} \mathbin{\bar\cup} \{\!|x?|\!\} = \{\!|x|\!\}$.

The question mark was chosen as symbol, because $\{\!|x?|\!\}$ intuitively denotes a set where the membership of x in it is questionable. As the computation proceeds, the membership might either be validated ($\{\!|x?|\!\} \le \{\!|x|\!\}$) or refuted ($\{\!|x?|\!\} \le \{\!||\!\}$). Extending our formal set notation in an obvious way, one obtains for instance the formal set $\{\!|1?,\, 2,\, 3|\!\} = \{\!|1?|\!\} \mathbin{\bar\cup} \{\!|2|\!\} \mathbin{\bar\cup} \{\!|3|\!\}$ which surely contains 2 and 3 and might also contain 1, but no other number. Hence, formal sets differ from actual sets of mathematics in that they may contain doubtful members which are neither proved to be inside nor outside. The last axiom of the previous paragraph reads $\{\!|x?,\, x|\!\} = \{\!|x|\!\}$, i.e. sure membership overrides doubtful membership.

3.7 Summary

Our list of requirements for a general set domain construction is now complete. We list the required operations and their axioms again for later reference:

- $\_ \mathbin{\bar\cup} \_ : \mathcal{S}\mathbf{D} \times \mathcal{S}\mathbf{D} \to \mathcal{S}\mathbf{D}$ commutative, associative, and idempotent with unit $\bar\emptyset$;
- $\{\!|.|\!\} : \mathbf{D} \to \mathcal{S}\mathbf{D}$ injective and never returning $\bar\emptyset$;
- $\textit{ext} : (\mathbf{X} \to \mathcal{S}\mathbf{Y}) \to (\mathcal{S}\mathbf{X} \to \mathcal{S}\mathbf{Y})$ linear in both arguments and satisfying $\textit{ext}\ f\ \{\!|x|\!\} = f\,x$;
- $\textit{exists} : (\mathbf{X} \to \mathbf{B}) \to (\mathcal{S}\mathbf{X} \to \mathbf{B})$ linear in both arguments and satisfying $\textit{exists}\ p\ \{\!|x|\!\} = p\,x$;
- $\{\!|.?|\!\} : \mathbf{D} \to \mathcal{S}\mathbf{D}$ satisfying the axioms $\{\!|x?|\!\} \le \{\!|x|\!\}$ and $\{\!|x?|\!\} \le \bar\emptyset$ and $\{\!|x|\!\} \mathbin{\bar\cup} \{\!|x?|\!\} = \{\!|x|\!\}$.

One might additionally want to have an operation creating a 'universal set' over any ground domain, i.e. a set containing all members of the ground domain. We think that this operation would have a higher level of conceptual complexity than the operations listed above that only allow for constructing sets from given sets or creating small sets (empty set and singletons) from scratch. For the same reasons, there is no operation of complement – the complement of the empty set would be the universal set – and no big intersection – the intersection over the empty set is the universal set.

For lists, there is another useful higher order operation called *fold* or *reduce* that takes a binary operation and combines the list items by this operation. Since the items of a set do not appear in a fixed order, the binary operation must be commutative and associative in order to make its *fold* be well defined. Therefore, we don't propose a general *fold* operation, but only some special instance of it: big union is a folded union.

4 The classical power domains

In this section, we show that the three classical power domain constructions fail to meet the complete specification above. In the section thereafter, we then present our set domains.

4.1 The lower power domain

We start with the lower or Hoare power domain $\mathcal{L}\mathbf{D}$. It may be defined as the set of all iso-lower subsets of $\mathbf{D}$ ordered by inclusion '$\subseteq$'. An isomorphic representation would be the set of Lawson closed lower subsets[4] of $\mathbf{D}$ ordered by inclusion, but the approach using iso-lower sets is technically simpler for our purposes.

Usually, the empty set is omitted from the domain, although it is a perfect iso-lower resp. closed lower set. Adding it poses no difficulty; it becomes the least element of the power domain.

$\mathcal{L}\mathbf{D}$ is a complete domain. Lubs and glbs are given by set union and intersection respectively. The least element is the empty set, and the greatest one is the set of all isolated points $\mathbf{D}^0$. The isolated sets in $\mathcal{L}\mathbf{D}$ are the sets $\Downarrow E$ where E is bifinite.

We now browse through the various topics of the specification and investigate to what extent they are satisfied.

- $\bar{\emptyset} = \emptyset$ and $A \mathbin{\bar{\cup}} B = A \cup B$ These operations obviously satisfy the CAIN axioms.
- $\{\!|x|\!\} = \Downarrow x$ is a continuous embedding. Because of $x \in \Downarrow x$, the result is never empty.

Combining these operations, one obtains $\{\!|x_1, \ldots, x_n|\!\} = \Downarrow\{x_1, \ldots, x_n\}$.

- $ext\, f\, A = \bigcup_{a \in A} fa$ One easily verifies that it is continuous and linear in both arguments and that $\bigcup_{a \in \Downarrow x} fa = fx$ holds.

Now, we derive the *map* functional and the big union from *ext*.

- $map\, f\, A = ext\, (\lambda x.\, \{\!|fx|\!\})\, A = \bigcup_{a \in A} \Downarrow(fa) = \Downarrow(fA)$
- $union\, A = ext\, (\lambda x.\, x)\, A = \bigcup_{a \in A} a$

[4]These are the closed sets of the Scott topology

- $$\begin{aligned} A \bar{\times} B &= ext\ (\lambda a.\ ext\ (\lambda b.\ \{\!|(a, b)|\!\})\ B)\ A \\ &= \bigcup_{a \in A} \bigcup_{b \in B} \Downarrow \{(a, b)\} = \bigcup_{c \in A \times B} \Downarrow \{c\} = \Downarrow (A \times B) = A \times B \end{aligned}$$

  The very last equation holds, because the product of iso-lower sets is again an iso-lower set.

Next, we consider the operation of doubtful membership. Since $\{\!|x?|\!\} \leq \bar{\emptyset} = \emptyset$ has to hold, and $\emptyset$ is the least element of the domain, there is no other choice than

- $\{\!|x?|\!\} = \emptyset$
- $x\ when\ b = \begin{cases} \Downarrow x & \text{if } b = \mathsf{T} \\ \emptyset & \text{if } b = \perp \text{ or } = \mathsf{F} \end{cases}$
- $$\begin{aligned} filter\ p\ A &= ext\ (\lambda a.\ a\ when\ (p\, a))\ A \ = \ \bigcup_{a \in A} (a\ when\ (p\, a)) \\ &= \bigcup \{\Downarrow a \mid a \in P,\ p\, a = \mathsf{T}\} \ = \ \Downarrow \{a \in P \mid p\, a = \mathsf{T}\} \end{aligned}$$

In contrast to the operations above, there is no existential quantification which meets the specification. To show this, we consider the lower power domain $\mathcal{L}\mathbf{1}$ over the one-point-domain $\mathbf{1} = \{\bullet\}$. It consists of two points: $\emptyset$ and $\{\bullet\}$, with $\emptyset \leq \{\bullet\}$. Now let p be the predicate assigning T to $\bullet$. Then $exists\ p\ \emptyset = \mathsf{F}$ and $exists\ p\ \{\bullet\} = \mathsf{T}$ should hold in contradiction to monotonicity.

Summarizing, we see that the lower power domain is able to give quite naturally meaning to the operations '$\bar{\emptyset}$', ' $\bar{\cup}$ ', '*ext*', '*union*', and ' $\bar{\times}$ '. But all operations using predicates can only use one part of the predicate, and operations returning Booleans are only able to return one Boolean or $\perp$. Thus, the lower power domain in some sense only gives half of the necessary semantic information.

In the course of the computation of a set, the sets in the lower power domain always increase. No element can disappear during a computation, but there might appear new elements at any time, even if the set was empty before. Thus, the sets of this power domain seem to give a lower bound on the final result of computations. In fact, a formal set L approximates an actual set S, iff $L \leq \Downarrow S$, i.e. L contains all elements which are known to be below some member of the actual set to be approximated.

This interpretation lets the *exists* predicate only telling T or $\perp$, never F. Once an element with a required property was detected in a set, it cannot disappear later on, and we know there is a member of the actual set above it which must also satisfy the property due to monotonicity. Thus, the answer is T. Conversely, if no element was found satisfying a predicate, only $\perp$ can be returned, since such an element might appear later.

For the same reasons, $x\ when\ \perp\ = \{\!|x?|\!\}$ is empty. As long as the condition is not evaluated, we cannot be sure whether there is any element in the outcoming set.

4.2 The upper or Smyth power domain

Now we turn to the upper power domain first proposed by Smyth [9]. The upper power domain $\mathcal{U}\mathbf{D}$ over a domain $\mathbf{D}$ consists of the Lawson closed upper sets of $\mathbf{D}$ ordered by containment '$\supseteq$'. This poset is a complete domain, even if the ground domain is not; the ground domain only needs to be an M-domain. Its isolated sets are the sets $\uparrow E$, where E is bifinite. The lub coincides with set intersection[5]. Its least element is the whole domain $\mathbf{D}$, and its greatest one is $\emptyset$.

[5] but glb is not identical with set union.

Smyth originally did not include the empty set in his power domain, although it is a perfect closed upper set. Furthermore, it is isolated, because it is the upper closure of the bifinite set $\emptyset$.

Now, we investigate to what extent the upper power domain satisfies the specification of the big set domain.

- $\bar{\emptyset} = \emptyset$ and $P \mathbin{\bar{\cup}} Q = P \cup Q$ obviously satisfy the specification.
- $\{\!|x|\!\} = \uparrow x$ is continuous and an injective embedding. It is never empty.

Combining these operations, one obtains $\{\!|x_1, \ldots, x_n|\!\} = \uparrow\{x_1, \ldots, x_n\}$.

- $ext\, f\, A = \bigcup_{a \in A} f a$ is well defined and continuous by theorem 2.1. It is not difficult to show that this operation satisfies its specification.

Now, we derive some set theoretic functions from *ext*.

- $map\, f\, A = ext\, (\lambda x.\, \{\!|f x|\!\})\, A = \bigcup_{a \in A} \uparrow (f a) = \uparrow (f A)$
- $union\, A = ext\, (\lambda x.\, x)\, A = \bigcup_{a \in A} a$
- $$\begin{aligned} A \mathbin{\bar{\times}} B &= ext\, (\lambda a.\, ext\, (\lambda b.\, \{\!|(a,\, b)|\!\})\, B)\, A \\ &= \textstyle\bigcup_{a \in A} \bigcup_{b \in B} \uparrow \{(a,\, b)\} = \bigcup_{p \in A \times B} \uparrow \{p\} = \uparrow (A \times B) = A \times B \end{aligned}$$

  The very last equation holds, because the product of upper sets is again an upper set.

Next, we consider the doubtful singletons. $\{\!|x?|\!\} \leq \bar{\emptyset} = \emptyset$ holds automatically, because $\emptyset$ is the greatest element. Then, $\{\!|x?|\!\} \leq \{\!|x|\!\}$ has to hold which is satisfied by the simplest possible choice

- $\{\!|x?|\!\} = \{\!|x|\!\} = \uparrow x$
- $x\ when\ b = \begin{cases} \{\!|x|\!\} & \text{if } b = \mathsf{T} \text{ or } = \bot \\ \emptyset & \text{if } b = \mathsf{F} \end{cases}$
- $$\begin{aligned} filter\ p\ A &= ext\, (\lambda a.\, a\ when\ (p\, a))\, A &= \textstyle\bigcup_{a \in A} (a\ when\ (p\, a)) \\ &= \textstyle\bigcup \{\uparrow a \mid a \in A,\, p\, a \neq \mathsf{F}\} &= \uparrow \{a \in A \mid p\, a \neq \mathsf{F}\} \end{aligned}$$

As with the lower power domain, there is no existential quantification satisfying the requirements. Consider the domain $\mathcal{U}\mathbf{1}$. It consists of two points: $\emptyset$ and $\{\bullet\}$, with $\{\bullet\} \leq \emptyset$. If p is a predicate mapping $\bullet$ to T, then $exists\ p\ \{\bullet\} = \mathsf{T}$ and $exists\ p\ \emptyset = \mathsf{F}$ should hold contradicting monotonicity.

During a computation, the sets in the Smyth power domain always shrink. Every element might disappear, even until the empty set, but it is impossible that new elements appear. Thus, the sets of this power domain give an upper bound on the final result of computations; they contain elements that are suspected to be in the final result. A member U of the upper power domain approximates an actual set S iff $S \subseteq U$.

According to this interpretation, $x\ when\ \bot$ equals $\{\!|x|\!\}$. Before the condition is not evaluated, x might be in the final result and cannot yet be dropped. For the same reasons, the *exists* predicate can only return F or $\bot$, never T. If there is no element in the Smyth set satisfying the predicate, *exists* may safely return F, since there is no possibility that such an element appears somehow. On the other hand, the answer must be $\perp$, if an element satisfying the predicate was found, because this element might disappear later on.

Summarizing, we see that the Smyth power domain gives only one half of the desired semantic information. Fortunately, its semantic properties are dual to those of the lower power domain, such that we may hope to get the full information by combining the two power domains. We try this in the next section after having briefly considered the remaining construction.

4.3 The convex or Plotkin power domain

Given a domain $\mathbf{D}$, sets of points of $\mathbf{D}$ might be ordered by the so-called Egli-Milner ordering. $A \leq B$ holds, if for all a in A, there is b in B with $a \leq b$, and also for all b in B, there is a in A with $a \leq b$. This ordering is reflexive and transitive, but not antisymmetric. Restriction to convex sets makes the Egli-Milner ordering antisymmetric. Thus, the Lawson closed convex sets form a poset when ordered by Egli-Milner. We call it extended Plotkin domain $\mathcal{P}'\mathbf{D}$ since it contains the empty set in contrast to Plotkin's original construction. It would be an M-domain, if it had a least element. It does not have a least element because of the empty set which is totally unrelated with any other set.

The Plotkin power domain $\mathcal{P}\mathbf{D}$ over $\mathbf{D}$ consists of the *non-empty* closed convex subsets of $\mathbf{D}$ ordered by the Egli-Milner ordering. This 'power domain' is in fact a domain (in the sense of Plotkin). The singleton $\{\perp\}$ is the least element.

The Plotkin power domain over the one-point-domain $\mathbf{1}$ has exactly one member: the set $\{\bullet\}$. On the other hand, there should be at least two formal sets – $\bar{\emptyset}$ and $\{\!|\bullet|\!\}$. This makes Plotkin's construction inappropriate for our purposes. Using the extended Plotkin domain instead, one might choose $\bar{\emptyset} = \emptyset$ and $\{\!|\bullet|\!\} = \{\bullet\}$, but then it is impossible to define $\{\!|\bullet?|\!\}$ properly. A solution could be to put the empty set artificially above $\{\perp\}$, or to add a new artificial least element. We did not follow these approaches further because the algebraic properties of the pure Plotkin domain are messed up by such artificial supplements.

5 The big set domain

In this section, we present the big set domain as combination of lower and upper power domain that is well suited for our purposes.

A member A of the big set domain is represented as pair (A^L, A^U)[6] consisting of an iso-lower set A^L and a Smyth set A^U. According to our interpretation of the lower and upper power domain, A approximates an actual set S, if all members a of A^L may surely be extended to an element s of S above a, and there are no members of S outside A^U. Formally, this is written $A^L \subseteq \Downarrow S$ and $S \subseteq A^U$. This implies that $A^L \subseteq \Downarrow A^U$ should hold. Pairs satisfying this condition are called *legal*. Legal pairs are also called *sandwiches* following Buneman [1], because they approximate an actual set S from both sides.

Definition 5.1 The big set domain $\Sigma\mathbf{D}$ over an M-domain $\mathbf{D}$ has carrier

$$\{(A^L, A^U) \mid \begin{array}{ll} (1) & A^L \text{ is an iso-lower set of } \mathbf{D} \\ (2) & A^U \text{ is an upper closed set, i.e. a Smyth set} \\ (3) & A^L \subseteq \Downarrow A^U \end{array} \}$$

[6] Buneman and also Gunter write the pairs the other way round, i.e. the upper set to the left.

The elements are ordered by

$$(A^L, A^U) \le (B^L, B^U) \text{ iff } A^L \subseteq B^L \text{ and } A^U \supseteq B^U$$

i.e. the order is inherited from $\mathcal{L}\mathbf{D} \times \mathcal{U}\mathbf{D}$. $\square$

Theorem 5.2
$\Sigma\mathbf{D}$ is a Scott domain. $(\emptyset, \mathbf{D})$ is its least element. If it exists, the lub of a family $(A_i)_{i\in I}$ is $(\bigcup_{i\in I} A_i^L, \bigcap_{i\in I} A_i^U)$. Isolated points are all pairs $(\Downarrow E, \uparrow F)$ where E and F are bifinite and $E \subseteq \Downarrow\uparrow F$ holds. The projections $\pi^L : \Sigma\mathbf{D} \to \mathcal{L}\mathbf{D}$ and $\pi^U : \Sigma\mathbf{D} \to \mathcal{U}\mathbf{D}$ are both continuous. $\square$

Note that $\Sigma\mathbf{D}$ is bounded complete, even if the ground domain $\mathbf{D}$ is not.

The big set domain is a sub-domain of the product $\mathcal{L}\mathbf{D}\times\mathcal{U}\mathbf{D}$ of lower and upper power domain. Because all pairs below a legal pair are also legal, the highest parts of the product are cut off by the condition of legality. Especially, this removes the top element of the product and causes the existence of many maximal elements.

When we consider the set operations, we can use our knowledge of the operations in the lower and upper domain. Most operations are simple combinations, only in case of inclusion and equality, a synergetic effect occurs.

In the sequel, we adopt some conventions. The operations of the lower power domain are indexed by L. For instance, ext_L denotes its extension. Similarly, the upper operations are indexed by U. If A is a sandwich, then $A = (A^L, A^U)$, and similarly, $B = (B^L, B^U)$ etc. If f is a function from somewhere to $\Sigma\mathbf{D}$, then $f^L = \pi^L \circ f : \ldots \to \mathcal{L}\mathbf{D}$, and analogously $f^U = \pi^U \circ f : \ldots \to \mathcal{U}\mathbf{D}$.

Continuity and correctness of the combined functions follow from those of the participants. For the basic operations, legality of the results must be shown. Naturally, this is not necessary for the derived operations.

- $\bar{\emptyset} = (\bar{\emptyset}^L, \bar{\emptyset}^U) = (\emptyset, \emptyset)$ This is a sandwich, since $\emptyset \subseteq \Downarrow\emptyset$.
- $P \mathbin{\bar{\cup}} Q = (P^L \mathbin{\bar{\cup}_L} Q^L, P^U \mathbin{\bar{\cup}_U} Q^U) = (P^L \cup Q^L, P^U \cup Q^U)$
  The result is legal, since $P^L \cup Q^L \subseteq \Downarrow P^U \cup \Downarrow Q^U = \Downarrow(P^U \cup Q^U)$.
- $\{\!|x|\!\} = (\{\!|x|\!\}_L, \{\!|x|\!\}_U) = (\Downarrow x, \uparrow x)$ For all x, $\Downarrow x \subseteq \Downarrow\uparrow x$ holds as required.
- $ext\ f\ A = (ext_L\ f^L\ A^L, ext_U\ f^U\ A^U) = (\bigcup_{a\in A^L} (fa)^L, \bigcup_{a\in A^U} (fa)^U)$
  To prove that the result is legal, let $y \in \bigcup_{a\in A^L} (fa)^L$. Then there is $x \in A^L$ such that $y \in (fx)^L$. Since A is legal, $x \in A^L \subseteq \Downarrow A^U$ holds, i.e. there is $x' \in A^U$ with $x \le x'$. Then

$$\begin{array}{rcll} y \in (fx)^L & \subseteq & (fx')^L & \text{by monotonicity of } f \\ & \subseteq & \Downarrow (fx')^U & \text{since } fx' \text{ is legal} \\ & \subseteq & \Downarrow \bigcup_{a\in A^U} (fa)^U & \end{array}$$

The equations for the derived operations *map*, *union*, and formal product may be computed from the definition of *ext*. They turn out to be free combinations of the corresponding lower and upper operations, for instance $P \mathbin{\bar{\times}} Q = (P^L \times Q^L, P^U \times Q^U)$.

Existential quantification may be derived from functional extension. The big set domain $\Sigma\mathbf{1}$ over the one-point-domain consists of the legal pairs of iso-lower and Smyth sets of $\mathbf{1}$. There are four pairs: $(\emptyset, \emptyset)$, $(\emptyset, \{\bullet\})$, $(\{\bullet\}, \emptyset)$, and $(\{\bullet\}, \{\bullet\})$. The third in this enumeration is illegal. Hence, three sandwiches remain. Their order is depicted in the following diagram:

$$\begin{tikzcd}
\bar{\emptyset} = (\emptyset, \emptyset) & & (\{\bullet\}, \{\bullet\}) = \{\!|\bullet|\!\} \\
& (\emptyset, \{\bullet\}) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Thus, this domain is isomorphic to the Boolean domain. By relating $\bar{\emptyset}$ with F, and $\{\!|\bullet|\!\}$ with T we obtain the two mappings *mkbool* and *mkset*. They are even linear, i.e. CAIN isomorphisms between $(\Sigma\mathbf{1},\ \bar{\cup}\ ,\ \bar{\emptyset})$ and $(\mathbf{B}, \vee, \mathsf{F})$.

- The predicate *exists* may be built using *mkbool* and *mkset*:

  $$\mathit{exists}\ p = \mathit{mkbool} \circ (\mathit{ext}\ (\mathit{mkset} \circ p))$$

  This function obviously meets its specification. Some formal manipulation gives the result

  $$\mathit{exists}\ p\ A = \begin{cases} \mathsf{T} & \text{if } \exists\, u \in A^L : p\, u = \mathsf{T} \\ \mathsf{F} & \text{if } \forall\, v \in A^U : p\, v = \mathsf{F} \\ \bot & \text{otherwise} \end{cases}$$

  i.e. this predicate may return every Boolean, and it looks at both parts of the argument predicate.

- $$\mathit{forall}\ p\ A = \neg(\mathit{exists}\ (\lambda a.\, \neg(p\, a))\ A) = \begin{cases} \mathsf{F} & \text{if } \exists\, u \in A^L : p\, u = \mathsf{F} \\ \mathsf{T} & \text{if } \forall\, v \in A^U : p\, v = \mathsf{T} \\ \bot & \text{otherwise} \end{cases}$$

- $$\mathit{empty}\ A = \mathit{forall}\ (\lambda a.\, \mathsf{F})\ A = \begin{cases} \mathsf{F} & \text{if } A^L \neq \emptyset \\ \mathsf{T} & \text{if } A^U = \emptyset \text{ (then } A^L \text{ must be empty, too)} \\ \bot & \text{if } A^L = \emptyset \text{ and } A^U \neq \emptyset \end{cases}$$

The remaining predicates are considered more closely in section 7. We finally consider the operations leading to *filter*.

- $$\{\!|x?|\!\} = (\{\!|x?|\!\}_L,\ \{\!|x?|\!\}_U) = (\emptyset,\ \uparrow x)$$

  The result is always legal, and the operation satisfies its axioms.

- $$\begin{aligned} \mathit{filter}\ p\ A &= \mathit{ext}\ (\lambda a.\, a\ \mathit{when}\ (p\, a))\ A \ = \ (\mathit{filter}_L\ p\ A^L, \mathit{filter}_U\ p\ A^U) \\ &= (\Downarrow \{a \in A^L \mid p\, a = \mathsf{T}\}, \uparrow \{a \in A^U \mid p\, a \neq \mathsf{F}\}) \end{aligned}$$

As the operations show, we were successful in finding a set domain that is an almost free combination of the lower and upper power domain, and thus, can answer both 'yes' and 'no'. But we are not able to construct all isolated sandwiches by means of the basic operations, because there are too much. In the next section, we strengthen the condition of legality in order to exclude some sandwiches.

6 The small set domain

Now, we identify a subset of the big set domain that exactly comprises those sandwiches that are obtained by the set operations and lubs of directed collections.

The big set domain as defined in the previous section consists of pairs $A = (A^L, A^U)$ where A^L is an iso-lower set, A^U is a Smyth set, and $A^L \subseteq \Downarrow A^U$ holds.

The singleton operation $\{\!|x|\!\} = (\Downarrow x, \uparrow x)$ produces sandwiches, where the lower and the upper part touch each other in the point x. Intuitively, such touching points are preserved by union '$\bar{\cup}$', and also by *map f* due to the continuity of f. Thus, it seems that all points of A^L may be raised into a touching point, or more formally, $A^L \subseteq \Downarrow$*(touch set)*.

The set of all touching points between A^L and A^U is the intersection of A^U with the closure of A^L. Thus, we finally reach at the condition $A^L \subseteq \Downarrow(\overline{A^L} \cap A^U)$. Here, the Lawson closure may be replaced by the directed closure, i.e. the set of all lubs of directed sets in A^L. Since $\Downarrow(\overline{A^L} \cap A^U)$ is a subset of $\Downarrow A^U$, the condition is indeed stronger than the condition of legality.

The condition is a bit unhandy, since it involves a closure and two occurrences of A^L. Fortunately, there is a much simpler, but equivalent condition which is used in the following definition:

Definition 6.1 The small set domain $\sigma\mathbf{D}$ over an M-domain $\mathbf{D}$ has carrier
$\{(A^L, A^U) \mid A^U$ is a Smyth set and there is a subset A^M of A^U with $A^L = \Downarrow A^M\}$
Its order is inherited from $\Sigma\mathbf{D}$. The sandwiches in $\sigma\mathbf{D}$ are called *tight*. □

The small set domain is isomorphic to the mixed power domain with empty set defined by Gunter [3, 4]. The restriction of tightness makes the sandwiches less dependent on very low points like $\perp$, since A^L is uniquely determined by a subset of A^U. Informally, one can say that only those parts of A^L lying within A^U are important, whereas the lower parts of A^L don't play any role.

The small set domain is still big enough for our purposes:

Theorem 6.2
All results of the operations $\bar{\emptyset}$, $\{\!|.|\!\}$, $\{\!|.?|\!\}$, and *mkset* are tight. If A and B are tight, then $A \bar{\cup} B$ is, too. If f generates tight sandwiches for all arguments, and A is tight, then *ext f A* is tight. The least sandwich $(\emptyset, \mathbf{D})$ is tight. Lubs of directed collections of tight sandwiches are tight. □

The theorem shows that $\sigma\mathbf{D}$ is big enough to allow all set operations we wanted to have. Additionally, it claims that the bottom element of $\Sigma\mathbf{D}$ is also in $\sigma\mathbf{D}$ and that directed lubs in $\Sigma\mathbf{D}$ remain valid in $\sigma\mathbf{D}$. Thus, the poset $\sigma\mathbf{D}$ is directed complete and has a least element. In the following, we consider further mathematical properties of the small set domain.

Lower and upper power domain are easily embedded by means of the mappings $L \mapsto (L, \mathbf{D})$ and $U \mapsto (\emptyset, U)$. Plotkin's power domain, even with the extension by the empty set, is isomorphically embedded into the small set domain by the function φ with $\varphi S = (\Downarrow S, \uparrow S)$. Its image is characterized by the additional condition $A^U \subseteq \uparrow(\overline{A^L} \cap A^U)$.

For every general sandwich S, there is a member of the extended Plotkin power domain P such that $\varphi P \geq S$. Since φP is a tight sandwich, the maximal points of the two set domains and of the extended Plotkin power domain embedded by φ are all the same. The isolated maximal points are exactly the images (by φ) of finite sets of isolated maximal elements.

The small set domain $\sigma\mathbf{D}$ is not necessarily bounded complete, even in the case of bounded complete ground domain $\mathbf{D}$. An example may be obtained from Plotkin's example for his power domain over pairs of Booleans by means of the embedding φ. The sets of Plotkin's example are consistent, but have no lub in his domain. One has to show that they don't have a lub in the greater setting of the small set domain after embedding. Naturally, they do have a lub in the big set domain because it is bounded complete. However, this lub is not tight.

Big and small set domain over the one-point domain coincide. Over the Booleans, the big set domain has 17 elements, whereas the small one only has 14. The three sandwiches not being tight are $(\{\perp\}, \{\mathsf{T}\})$, $(\{\perp\}, \{\mathsf{F}\})$, and $(\{\perp\}, \{\mathsf{T}, \mathsf{F}\})$. Plotkin's domain has 7 elements in this case.

Now, we state that the small set 'domain' is in fact a domain in the sense of Plotkin:

Theorem 6.3
The poset $\sigma\mathbf{D}$ is an M-domain.[7] Its isolated points are the sandwiches $(\Downarrow E, \uparrow F)$ where E and F are bifinite and E is a subset of $\uparrow F$. If $\mathbf{D}$ is an SFP-domain, then $\sigma\mathbf{D}$ is an SFP-domain, too. $\square$

This theorem may be proved by explicitly computing the mubs of bifinite sets.

We already showed that all set operations are possible in the small set domain $\sigma\mathbf{D}$. We now demonstrate that they are enough to construct all domain elements.

Theorem 6.4
All isolated points of the small set domain may be constructed by a finite number of applications of the operations $\bar{\emptyset}$, $\bar{\cup}$, $\{\!|.|\!\}$, and $\{\!|.?|\!\}$ applied to isolated points of the ground domain. $\square$

Since the small power domain is algebraic, an arbitrary sandwich in it may be obtained by the same operations plus directed lubs.

Proof: Let $P = (\Downarrow E, \uparrow F)$ be an isolated point of the small set domain. This means E and F are bifinite, and $E \subseteq \uparrow F$. Let $E = \{e_1, \ldots, e_r\}$. Since $E \subseteq \uparrow F$, for each point e_k in E, there is a point f_k in F below. Let $F' = \{f_1, \ldots, f_r\}$, and $G = F \setminus F'$ the remainder of F, $G = \{g_1, \ldots, g_s\}$. With these notations, we claim

$$P = \{\!|e_1, \ldots, e_r|\!\} \mathbin{\bar{\cup}} \{\!|f_1, \ldots, f_r|\!\} \mathbin{\bar{\cup}} \{\!|g_1?, \ldots, g_s?|\!\}$$

By executing the singleton operations and some of the unions, the big expression above reduces to $(\Downarrow E, \uparrow E) \mathbin{\bar{\cup}} (\Downarrow F', \uparrow F') \mathbin{\bar{\cup}} (\emptyset, \uparrow G) = (\Downarrow E \cup \Downarrow F', \uparrow E \cup \uparrow F' \cup \uparrow G)$. We may further compute $\uparrow F' \cup \uparrow G = \uparrow (F' \cup G) = \uparrow F$. $E \subseteq \uparrow F$ implies $\uparrow E \subseteq \uparrow F$, whence the second component of the pair simplifies to $\uparrow F$. Remember for any f_k in F', there is a point e_k in E above, whence $\Downarrow F' \subseteq \Downarrow E$. Therefore, the left component becomes $\Downarrow E$. Thus, we finally obtain $(\Downarrow E, \uparrow F) = P$. $\square$

7 Set equality

A continuous equality for a given domain $\mathbf{D}$ is a continuous mapping $Eq : \mathbf{D} \times \mathbf{D} \to \mathbf{B}$ such that $Eq\,(x, y) = \mathsf{T}$ implies $x = y$, and $Eq\,(x, y) = \mathsf{F}$ implies $x \neq y$. The mathematical equality '=' in $\mathbf{D}$ is not a continuous equality since it is not even monotonic.

There is a continuous equality for every domain, namely $\lambda(x, y).\perp$. Among all the continuous equalities, there is a greatest one, that is the function defined by

$$(x == y) \; = \; \begin{cases} \mathsf{T} & \text{if } x \text{ and } y \text{ are identical, maximal, isolated points} \\ \mathsf{F} & \text{if } x \not\uparrow y \text{, i.e. } x \text{ and } y \text{ are not consistent} \\ \perp & \text{otherwise} \end{cases}$$

From the basic set operations and a continuous equality in the ground domain, we derived an equality operation for formal sets in section 3. If the greatest continuous equality of the ground domain is used, the derived equality in the small set domain is again the greatest continuous equality, i.e. it can be used once more to construct the greatest continuous equality for sets of sets

[7]Remember that $\sigma\mathbf{D}$ was only defined for M-domains $\mathbf{D}$.

etc. For the big set domain, the derived equality is *not* the greatest continuous one. Its 'positive' part is as required: it returns T, iff the argument sets are identical maximal isolated points. Its 'negative' part however is too small: there are inconsistent sandwiches, where the constructed equality returns $\perp$ instead of F.

For instance, consider the big set domain over the Booleans. The sandwiches $A = (\{\perp\}, \{\mathsf{T}\})$ and $B = (\{\perp\}, \{\mathsf{F}\})$ are inconsistent, because their formal lub in the free product of lower and upper power domain, namely $(\{\perp\}, \emptyset)$, is not legal. Intuitively, A describes non-empty sets that at most contain T, and B describes non-empty sets that at most contain F. Using the set equality derived in section 3, $(A == B)$ evaluates to $\perp$. Both A and B are not tight, such that this example does not matter for the small set domain.

8 The small set domain as a free construction

For given algebraic theory $\mathcal{T}$, a $\mathcal{T}$-algebra $\mathbf{F}$ is free over a generating object $\mathbf{X}$, if there is an embedding $\iota : \mathbf{X} \hookrightarrow \mathbf{F}$ such that for every $\mathcal{T}$-algebra $\mathbf{Y}$ and every map $f : \mathbf{X} \to \mathbf{Y}$ there is a unique $\mathcal{T}$-homomorphism $\bar{f} : \mathbf{F} \to \mathbf{Y}$ which extends f, i.e. $\bar{f} \circ \iota = f$. A free construction $\mathcal{F}$ w.r.t. $\mathcal{T}$ is a map that attaches to every generator $\mathbf{X}$ a free $\mathcal{T}$-algebra $\mathcal{F}\mathbf{X}$ over $\mathbf{X}$.

In the context of domain theory, all objects and algebras have to be (SFP)-domains, and all mappings have to be continuous. Since every monotonic mapping from a domain base $\mathbf{X}^0$ to a domain $\mathbf{Y}$ may be uniquely extended to a continuous mapping from $\mathbf{X}$ to $\mathbf{Y}$, one needs to consider posets and monotonic functions only. Such algebras are called monotonic algebras.

The classical power domains are free construction w.r.t. suitable algebraic theories: Plotkin's power domain extended by the empty set is free for the CAIN theory, the lower one for the CAIN theory with additional axiom $A \leq A \mathbin{\bar{\cup}} B$, and the upper one for the CAIN theory with additionally $A \mathbin{\bar{\cup}} B \leq B$.

The small set domain is a free construction for the *mix theory* as Gunter [3, 4] and I independently found out. The mix theory is a CAIN theory enriched by an additional unary operation '?'.[8] This operation is the extension of the operation of doubtful singleton, $\{\!|.?|\!\}$, from elements to formal sets. $A?$ is built from A by discarding all secure memberships and only retaining the doubtful ones, e.g. $\{\!|1?, 2|\!\}? = \{\!|1?, 2?|\!\}$. Its axioms are derived from what we required for $\{\!|.?|\!\}$ in section 3 together with linearity. In the following definition, we give – in contrast to Gunter – a minimal set of axioms, i.e. for each of the four axioms, there is a monotonic algebra satisfying all axioms except the given one.

Definition 8.1 (Mix algebras)
A (monotonic) mix algebra $(\mathbf{P}, \bar{\cup}, \bar{\emptyset}, \_?)$ is a CAIN poset $(\mathbf{P}, \bar{\cup}, \bar{\emptyset})$ with an additional monotonic operation $\_? : \mathbf{P} \to \mathbf{P}$ satisfying the following 4 axioms[9]

$$
\begin{array}{llll}
(A1) & A? \leq \bar{\emptyset} & (A2) & A? \leq A \\
(A3) & A \mathbin{\bar{\cup}} A? \geq A & (A4) & (A \mathbin{\bar{\cup}} B)? \leq A? \mathbin{\bar{\cup}} B?
\end{array}
$$

A mapping f between two mix algebras is a mix homomorphism iff it is monotonic and linear and additionally satisfies $f(A?) = (fA)?$. For a mix domain, the underlying poset has to be a domain and all operations must be continuous. □

[8] denoted by □ by Gunter

[9] Here and in the sequel, all axioms and theorems are implicitly universally quantified over $\mathbf{P}$.

One may easily check out that small as well as big set domain become mix domains by defining $(A^L, A^U)? = (\emptyset, A^U)$. The Boolean domain is a mix domain isomorphic to $\sigma\mathbf{1} = \Sigma\mathbf{1}$ with $\bar{\emptyset} = \mathsf{F}$, $\bar{\cup}$ being parallel disjunction, and $\mathsf{F}? = \mathsf{F}$ and $\mathsf{T}? = \bot? = \bot$.

The mix theory as defined above allows for deriving some theorems which hold in all mix algebras. Among those, there is (A3) and (A4) with equality. We now present the most important of these theorems with their proofs which end up in a characterization of mix homomorphisms. Gunter did not do so in his publications as far as I know them.

(T1) $A \mathbin{\bar{\cup}} B? \le A$ since $A \mathbin{\bar{\cup}} B? \stackrel{A1}{\le} A \mathbin{\bar{\cup}} \bar{\emptyset} \stackrel{N}{=} A$

(T2) $A \mathbin{\bar{\cup}} A? = A$ by (A3) and (T1)

(T3) $\bar{\emptyset}? = \bar{\emptyset}$ since $\bar{\emptyset} \stackrel{T2}{=} \bar{\emptyset} \mathbin{\bar{\cup}} \bar{\emptyset}? \stackrel{N}{=} \bar{\emptyset}?$

(T4) $A?? = A?$ since $A?? \stackrel{A2}{\le} A? \stackrel{T2}{=} A? \mathbin{\bar{\cup}} A?? \stackrel{T1}{\le} A??$

(T5) $A? = A$ iff $A \le \bar{\emptyset}$
Proof: '$\Rightarrow$' $A \stackrel{lhs}{=} A? \stackrel{A1}{\le} \bar{\emptyset}$ '$\Leftarrow$' $A? \stackrel{A2}{\le} A \stackrel{T2}{=} A \mathbin{\bar{\cup}} A? \stackrel{rhs}{\le} \bar{\emptyset} \mathbin{\bar{\cup}} A? \stackrel{N}{=} A?$

(T6) $X \le \bar{\emptyset}$ and $X \le A$ iff $X \le A?$ i.e. $A?$ is the greatest lower bound of $\bar{\emptyset}$ and A.
Proof: '$\Rightarrow$' $X \le \bar{\emptyset}$ implies $X = X?$ by (T5). $X \le A$ implies $X? \le A?$ by monotonicity of '?'. Together, $X \le A?$ follows. '$\Leftarrow$' by (A1) and (A2).

(T7) $(A \mathbin{\bar{\cup}} B)? = A? \mathbin{\bar{\cup}} B?$
Proof: '$\le$' is (A4). '$\ge$' is deduced by (T6) from $A? \mathbin{\bar{\cup}} B? \le \bar{\emptyset}$ (by (A1) and (N)) and $A? \mathbin{\bar{\cup}} B? \le A \mathbin{\bar{\cup}} B$ (by (A2)).

(T8) The three statements $A \le A \mathbin{\bar{\cup}} B$ and $A? \le B?$ and $A? \le B$ are equivalent.
Proof: (1) $\Rightarrow$ (2): $A? \stackrel{1}{\le} (A \mathbin{\bar{\cup}} B)? \stackrel{T7}{=} A? \mathbin{\bar{\cup}} B? \stackrel{T1}{\le} B?$
(2) $\Rightarrow$ (3): $A? \stackrel{2}{\le} B? \stackrel{A2}{\le} B$
(3) $\Rightarrow$ (1): $A \stackrel{T2}{=} A \mathbin{\bar{\cup}} A? \stackrel{3}{\le} A \mathbin{\bar{\cup}} B$

(T9) $X \le \bar{\emptyset}$ and $X \le A$ and $A \mathbin{\bar{\cup}} X \ge A$ iff $X = A?$
Proof: '$\Leftarrow$' is immediate by (A1), (A2), and (A3).
'$\Rightarrow$': $X \le \bar{\emptyset}$ and $X \le A$ imply $X \le A?$ by (T6). $A \mathbin{\bar{\cup}} X \ge A$ implies $A? \le X$ by (T8).

Gunter defined mix algebras by an axiom system consisting of (T7), (T4), (T2), (A2), and (T1). Because (T1) implies (A1) by choosing $A = \bar{\emptyset}$ and (T2) implies (A3) and (T7) implies (A4), his mix theory is equivalent with ours.

(T9) is a particularly interesting theorem. It implies that the operation '?' is uniquely determined in a given mix algebra, i.e. for given CAIN poset, there is at most one choice for the operation '?' to turn it into a mix algebra. Another important consequence is the following theorem which cannot be found in Gunter's publications [2, 3, 4].

Theorem 8.2
A monotonic linear mapping between two mix algebras is automatically a mix homomorphism. □

Proof: Let $f : \mathbf{X} \to \mathbf{Y}$ be a monotonic linear map between the two mix algebras $\mathbf{X}$ and $\mathbf{Y}$. Then for all $A \in \mathbf{X}$, $A? \le \bar{\emptyset}$ and $A? \le A$ and $A \mathbin{\bar{\cup}} A? \ge A$ imply $f(A?) \le \bar{\emptyset}$ and $f(A?) \le fA$ and $fA \mathbin{\bar{\cup}} f(A?) \ge fA$ respectively. By (T9), $f(A?) = (fA)?$ follows. □

Finally, one can show that the small set domain is free for the mix algebra:

Theorem 8.3 For every domain $\mathbf{D}$, the small set domain $\sigma\mathbf{D}$ is a mix domain. For every arbitrary mix domain $\mathbf{Y}$ and every continuous map $f : \mathbf{D} \to \mathbf{Y}$, there is a unique linear continuous map $\bar{f} : \sigma\mathbf{D} \to \mathbf{Y}$ with $\bar{f}\{\!|x|\!\} = fx$. $\bar{f}$ is a mix homomorphism. □

A proof of a similar theorem may be found in [3]. Our theorem follows from it by applying Theorem 8.2.

Corollary 8.4
For every continuous map $f : \mathbf{X} \to \sigma\mathbf{Y}$, there is a unique linear continuous extension $\bar{f} : \sigma\mathbf{X} \to \sigma\mathbf{Y}$ with $\bar{f} \circ \{\!|.|\!\} = f$. It is explicitly given by $\bar{f} = \mathit{ext}\, f$ where *ext* is defined as in section 5. □

Theorem 8.3 and (T9) imply together that the functions *ext*, *exists*, and '$\{\!|.?|\!\}$' as defined in section 5 are the only ones that satisfy the respective specifications in case of the small set domain.

9 The big set domain as a free construction

In Gunter's papers, the freedom of the big set domain is an open question. I found out that it is also a free construction, but w.r.t. an algebraic theory that is less natural than the mix theory since it involves a partial operation with a strange meaning. The problem is that all reasonable operations on sandwiches seem to preserve tightness.

The solution is to use an operation that marries sandwiches to each other i.e. $(A^L, A^U) \otimes (B^L, B^U) = (A^L, B^U)$. A problem is that the partners may not harmonize such that the marriage is not durable. This means that '$\otimes$' is a partial operation only. The exact definition for the big set domain is

$$(A^L, A^U) \otimes (B^L, B^U) = \begin{cases} (A^L, B^U) & \text{if } A^L \subseteq \Downarrow B^U \\ \mathit{undefined} & \text{otherwise} \end{cases}$$

'$\otimes$' becomes a total continuous operation if '*undefined*' is added to the big set domain as an artificial top element. This is done in the following definition which enumerates the axioms of a sandwich algebra.

Definition 9.1 (Sandwich algebras)
A (monotonic) sandwich algebra $(\mathbf{P}, \bar{\cup}, \bar{\emptyset}, \otimes)$ is a CAIN poset $(\mathbf{P}, \bar{\cup}, \bar{\emptyset})$ with an additional monotonic operation $\_ \otimes \_ : \mathbf{P}\times\mathbf{P} \to \mathbf{P}\cup\{\mathit{undefined}\}$ where $x \le \mathit{undefined}$ for all $x \in \mathbf{P}$ satisfying the following 4 axioms

$(A1)$ $A \otimes A = A$ $(A2)$ $A \otimes B \le (A \cup A') \otimes B$ $(A3)$ $A \otimes B \ge A \otimes (B \cup B')$

$(A4)$ If $A \otimes B$ and $A' \otimes B'$ are defined, then $(A \cup A') \otimes (B \cup B') = (A \otimes B) \cup (A' \otimes B')$

A mapping f between two sandwich algebras is a sandwich homomorphism iff it is monotonic and linear and additionally satisfies $f(A \otimes B) = (fA) \otimes (fB)$ whenever $A \otimes B$ is defined, i.e. in $\mathbf{P}$. For a sandwich domain, the underlying poset has to be a domain and all operations must be continuous. □

One may easily verify that the big set domain becomes a sandwich domain by defining '$\otimes$' as indicated above. The Boolean domain is a sandwich domain with $\bar{\emptyset} = \mathsf{F}$ and $\bar{\cup}$ being parallel disjunction since it is isomorphic to $\Sigma\mathbf{1}$.

As for the mix theory, some theorems may be derived from the axioms.

(T1) If $A \otimes B$ and $A \otimes B'$ are defined, then $A \otimes (B \cup B') = (A \otimes B) \cup (A \otimes B')$
Proof: let $A' = A$ in (A4) and apply (I).

(T2) If $A \circledcirc B$ and $A' \circledcirc B$ are defined, then $(A \cup A') \circledcirc B = (A \circledcirc B) \cup (A' \circledcirc B)$
Proof: let $B' = B$ in (A4) and apply (I).

(T3) If $A \circledcirc B$ is defined, then $(A \cup B) \circledcirc B = (A \circledcirc B) \cup B$ and $A \circledcirc (A \cup B) = A \cup (A \circledcirc B)$.
Proof: the first equation by (T2) and (A1), the second one by (T1) and (A1).

(T4) $A \circledcirc B$ and B are both $\leq (A \cup B) \circledcirc B$.
Proof: $B = B \circledcirc B$ by (A1), then apply (A2) in both cases.

(T5) $A \circledcirc (A \cup B) \leq$ both of A and $A \circledcirc B$.
Proof: $A = A \circledcirc A$ by (A1), then apply (A3) in both cases.

(T6) $\bar{\emptyset} \circledcirc B \leq B$ since $\bar{\emptyset} \circledcirc B \overset{T4}{\leq} (\bar{\emptyset} \cup B) \circledcirc B \overset{N}{=} B \circledcirc B \overset{A1}{=} B$

(T7) $\bar{\emptyset} \circledcirc B \leq \bar{\emptyset}$ since $\bar{\emptyset} \circledcirc B \overset{N}{=} \bar{\emptyset} \circledcirc (\bar{\emptyset} \bar{\cup} B) \overset{T5}{\leq} \bar{\emptyset}$

(T8) Every sandwich algebra is a mix algebra by the definition $B? = \bar{\emptyset} \circledcirc B$.
Proof: (T6) means $B? \leq B$ and (T7) means $B? \leq \bar{\emptyset}$. These inequations also show that $B?$ is defined for all B. (T3) implies $B? \cup B = (\bar{\emptyset} \circledcirc B) \cup B = (\bar{\emptyset} \cup B) \circledcirc B = B$ by (N) and (A1). (T1) gives $(A \cup B)? = A? \cup B?$.

(T9) $A \leq A \cup B$ iff $\bar{\emptyset} \circledcirc A \leq \bar{\emptyset} \circledcirc B$ iff for all C: $C \circledcirc A \leq C \circledcirc B$.
Proof: '(1) $\Rightarrow$ (3)': $C \circledcirc A \overset{1}{\leq} C \circledcirc (A \cup B) \overset{A3}{\leq} C \circledcirc B$
(3) obviously implies (2), and (2) implies (1) by (T8) and theorem (T8) of the mix theory.

(T10) $A \cup B \leq B$ iff for all C: $A \circledcirc C \leq B \circledcirc C$.
Proof: '$\Rightarrow$': $A \circledcirc C \overset{A2}{\leq} (A \cup B) \circledcirc C \overset{lhs}{\leq} B \circledcirc C$
'$\Leftarrow$': $A \cup B \overset{A1}{=} (A \cup B) \circledcirc (A \cup B) \overset{T2}{=} (A \circledcirc (A \cup B)) \cup (B \circledcirc (A \cup B))$. (T2) is applicable since both marriages to the right are defined due to (T5) viz. $A \circledcirc (A \cup B) \leq A$ and $B \circledcirc (A \cup B) \leq B$. The precondition with $C = A \cup B$ may be applied to the last term resulting in $(B \circledcirc (A \cup B)) \cup (B \circledcirc (A \cup B)) \overset{I}{=} B \circledcirc (A \cup B) \overset{T5}{\leq} B$.

(T11) $A \leq A \cup B \geq B$ iff for all C: $C \circledcirc A = C \circledcirc B$.
This is deduced directly from (T9). We call A and B in this case upper equivalent and write $A \approx_U B$. This relation obviously is an equivalence relation. For the big set domain, one can show $A \approx_U B$ iff $A^U = B^U$, hence the name.

(T12) $A \geq A \cup B \leq B$ iff for all C: $A \circledcirc C = B \circledcirc C$.
This is deduced directly from (T10). We call A and B in this case lower equivalent and write $A \approx_L B$. For the big set domain, one can show $A \approx_L B$ iff $A^L = B^L$, hence the name.

(T13) If $A \circledcirc B$ is defined, then it is in the intersection of the lower class of A and the upper class of B, i.e. $A \approx_L A \circledcirc B \approx_U B$.
Proof: The lower equivalence by (T3) and (T5), and the upper one by (T3) and (T4).

(T14) For given A and B, there is an X such that $A \approx_L X \approx_U B$ iff $A \circledcirc B$ is defined. In this case, $A \circledcirc B = X$ holds. $A \circledcirc B$ is undefined iff there is no such X.
Proof: If $A \circledcirc B$ is defined, there is such an X by (T13). Conversely, assume such an X. Note that (T11) and (T12) hold no matter whether the married couples involving 'C' are defined or not. Hence, $A \circledcirc B \overset{T12}{=} X \circledcirc B \overset{T11}{=} X \circledcirc X \overset{A1}{=} X$, i.e. $A \circledcirc B$ is defined and equals X.

Similar to mix algebras, (T14) implies that the operation '$\circledcirc$' is uniquely determined in a given sandwich algebra, i.e. for given CAIN poset, there is at most one choice for the operation '$\circledcirc$' to turn it into a sandwich algebra. Another important consequence is the following theorem:

Theorem 9.2 A monotonic linear mapping between two sandwich algebras is automatically a sandwich homomorphism. $\square$

Proof: Let $f : \mathbf{X} \to \mathbf{Y}$ be a monotonic linear map between the two sandwich algebras $\mathbf{X}$ and $\mathbf{Y}$. Let A and B be members of $\mathbf{X}$ such that $A \circledcirc B$ is defined. Then $A \approx_L A \circledcirc B \approx_U B$ holds by (T13). Using the equivalence of $S \approx_L T$ with $S \geq S \cup T \leq T$ and of $S \approx_U T$ with $S \leq S \cup T \geq T$, one obtains by monotonicity and linearity of f the relations $fA \approx_L f(A \circledcirc B) \approx_U fB$. (T14) then implies $fA \circledcirc fB = f(A \circledcirc B)$. $\square$

Finally, one can show that the big set domain is free for the sandwich algebra:

Theorem 9.3 For every domain $\mathbf{D}$, the big set domain $\Sigma\mathbf{D}$ is a sandwich domain. For every arbitrary sandwich domain $\mathbf{Y}$ and every continuous map $f : \mathbf{D} \to \mathbf{Y}$, there is a unique linear continuous map $\bar{f} : \Sigma\mathbf{D} \to \mathbf{Y}$ with $\bar{f}\{\!|x|\!\} = fx$. $\bar{f}$ is a sandwich homomorphism. $\square$

Sketch of proof: For a bifinite actual set $E \subseteq \mathbf{D}$, we define $f'E = \overline{\bigcup}_{e \in E} fe$. The isolated sandwiches are given by $(\Downarrow E, \uparrow F)$ with $E \subseteq \Downarrow\uparrow F$. We define for them $\bar{f}(\Downarrow E, \uparrow F) = f'E \circledcirc f'F$. Then one has to show that this mapping is well defined, i.e. independent from the actual choice of E and F, and that the married couple on the right hand side is always defined. Next $\bar{f}$ is shown to be monotonic and linear. Then it may be extended to the whole domain $\Sigma\mathbf{D}$ to a continuous linear map. Uniqueness is shown by using the fact $(\Downarrow E, \uparrow F) = (\Downarrow E, \uparrow E) \circledcirc (\Downarrow F, \uparrow F) = \{\!|e \mid e \in E|\!\} \circledcirc \{\!|y \mid y \in F|\!\}$ and theorem 9.2. $\square$

Corollary 9.4
For every continuous map $f : \mathbf{X} \to \Sigma\mathbf{Y}$, there is a unique linear continuous extension $\bar{f} : \Sigma\mathbf{X} \to \Sigma\mathbf{Y}$ with $\bar{f} \circ \{\!|.|\!\} = f$. It is explicitly given by $\bar{f} = \mathit{ext}\, f$ where *ext* is defined as in section 5. $\square$

It is particularly nice because it neither mentions sandwich theory nor marrying, i.e. it is independent from this partial and semantically dubious operation.

The corollary, (T8) and (T14) imply together that the functions *ext*, *exists*, and '$\{\!|.?|\!\}$' as defined in section 5 are the only ones that satisfy the respective specifications in case of the big set domain.

10 Conclusion

In order to describe the semantics of a set type with given operations, we proposed two set domains whose construction is identical with the sandwich [1] and the mixed power domain with empty set [3, 4]. The big set domain is useful in the context of bounded complete (Scott) domains, whereas the small one is suitable in the setting of SFP (Plotkin) domains. Besides not being bounded complete, the mathematical properties of the small domain are superior to those of the big one, because all its isolated points may be constructed by means of the supported set operations, and the greatest continuous equality of the ground domain easily carries over to it.

Both constructions produce free domains with respect to suitable algebraic theories where the theory of the small set domain is more natural than that of the big one. The freedom results allow for deducing that our primitive operations *ext*, *exists*, and doubtful singleton are uniquely determined by their specification if the underlying set domain with empty set, union, and singleton is given.

The discussion of the set operations they support sheds some light upon the informational content of the various domains. We worked out a more formal characterization of the domains by the behavior of existential quantification. It induces an isomorphism between the big set domain $\Sigma\mathbf{D}$ and the space of linear continuous second order predicates $(\mathbf{D} \to \mathbf{B}) \overset{lin}{\to} \mathbf{B}$ – linear w.r.t. parallel disjunction. Since all other power domains are subdomains of the big set domain, they all correspond to sets of linear second order predicates which may be characterized by additional logical constraints.[10]

Acknowledgements

I am most grateful to Fritz Müller for his hints to the literature, many fruitful discussions, and the suggestion to investigate existential quantification on the set domains. Helmut Seidl also was always ready for discussions and read some drafts, and Reinhard Wilhelm made valuable proposals to improve the quality of a first draft. I also wish to thank the ESOP referee who suggested to investigate the question of freedom of the two set domains.

References

[1] Buneman, P., Davidson, S.B., Watters, A.: A semantics for Complex Objects and Approximate Queries: Extended Abstract, Internal Report MS-CIS-87-99, University of Pennsylvania, (Oct. 1988). Also in 7th ACM Principles of Database Systems.

[2] Gunter, C.A.: A Logical Interpretation of Powerdomains, Internal Report without number and date, University of Pennsylvania, before Sept. 1989

[3] Gunter, C.A.: The Mixed Powerdomain, Internal Report MS-CIS-89-77, Logic & Computation 18, University of Pennsylvania, (Dec. 1989)

[4] Gunter, C.A.: Relating Total and Partial Correctness Interpretations of Non-Deterministic Programs, In: P. Hudak, ed.: Principles of Programming Languages (POPL '90), ACM, (1990), 306-319

[5] Heckmann, R.: Set Domains, Ph.D. Thesis, Universität des Saarlandes, to appear

[6] Jung, A.: Cartesian Closed Categories of Domains, Ph.D. Thesis, FB Mathematik, Technische Hochschule Darmstadt, (1988)

[7] Lawson, J.D.: The Versatile Continuous Order, in: Main, M., Melton, A., Mislove, M., Schmidt, D., eds: Mathematical foundations of programming language semantics, 1987, Lecture Notes in Computer Science 298, (1988)

[8] Plotkin, G.D.: A powerdomain Construction, SIAM J. Comput., Vol. 5, No. 3, (1976), 452-487

[9] Smyth, M.B.: Power Domains, Journal of Computer and System Sciences 16, (1978), 23-36

[10] Smyth, M.B.: Power Domains and Predicate Transformers: A Topological View, ICALP 83, in: J. Diaz, ed.: Lecture Notes in Computer Science 154, Springer, (1983), 662-676

[11] Smyth, M.B.: The Largest Cartesian Closed Category of Domains, Theoretical Computer Science 27, (1983), 109-119

[12] Stoy, J.E.: Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory, M.I.T. Press, (1977)

[10] These results were submitted to LICS'90, but there was no notification of (non-)acceptance yet.