July 1993

Algebraic Characterization of Edible Powerdomains

Leonid Libkin
University of Pennsylvania

Follow this and additional works at: http://repository.upenn.edu/cis_reports

Recommended Citation
Abstract
Powerdomains like mixes, sandwiches, snacks and scones are typically used to provide semantics of collections of descriptions of partial data. In particular, they were used to give semantics of databases with partial information. In this paper we argue that to be able to put these constructions into the context of a programming languages it is necessary to characterize them as free (ordered) algebras. Two characterizations — for mixes and snacks — are already known, and in the first part of the paper we give characterizations for scones and sandwiches and provide an alternative characterization of snacks. The algebras involved have binary and unary operations and relatively simple equational theories. We then define a new construction, which is in essence all others put together (hence called salad) and give its algebraic characterization. It is also shown how all algebras considered in the paper are related in a natural way, that is, in a way that corresponds to embeddings of their powerdomains. We also discuss some semantic issues such as relationship between the orderings and the semantics and justification for choosing the orderings. Finally, we outline prospects for further research.

Comments
Algebraic Characterization of Edible Powerdomains

MS-CIS-93-70
LOGIC & COMPUTATION 71

Leonid Libkin

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

July 1993
Algebraic Characterization of Edible Powerdomains

Leonid Libkin*

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA

Abstract

Powerdomains like mixes, sandwiches, snacks and scones are typically used to provide semantics of collections of descriptions of partial data. In particular, they were used to give semantics of databases with partial information. In this paper we argue that to be able to put these constructions into the context of a programming languages it is necessary to characterize them as free (ordered) algebras. Two characterizations – for mixes and snacks – are already known, and in the first part of the paper we give characterizations for scones and sandwiches and provide an alternative characterization of snacks. The algebras involved have binary and unary operations and relatively simple equational theories. We then define a new construction, which is in essence all others put together (hence called salad) and give its algebraic characterization. It is also shown how all algebras considered in the paper are related in a natural way, that is, in a way that corresponds to embeddings of their powerdomains. We also discuss some semantic issues such as relationship between the orderings and the semantics and justification for choosing the orderings. Finally, we outline prospects for further research.

1 Introduction

It has become a tradition to give food names to domains used in approximations. It started when Peter Buneman and Susan Davidson invented sandwiches which consist of lower and upper approximations and denote precisely what is in between, hence the name. A slight generalization in which one of the approximations consists of several sets, was perceived

---

*This work started when the author was visiting Fachbereich Mathematik, Technische Hochschule Darmstadt.
as “many sandwiches” and thus given the name of the snack powerdomain. Carl Gunter invented mixes, which combine lower and upper powerdomains. Perhaps it was not his intention to follow the “edible” tradition, but the name he chose fit in rather nicely with the others. Another generalization was due to Achim Jung, and it is basically to snacks what sandwiches are to mixes. It was probably not this observation but the desire to follow tradition that motivated him to look for a food name. At that time a graduate student in Darmstadt baked some scones and brought them to the department, and the name of the scone powerdomain appeared. Finally, in this paper we use a new powerdomain which is just all others put together. Rejecting a nice idea of Peter Buneman to call it the “kitchen sink” powerdomain, I shall use the salad powerdomain instead.

Let me first motivate the use of the edible powerdomains. Since late 70s, many researchers tried to understand partial information in databases: [1, 14, 9, 23, 6, 5, 17] name just a few of the many books and papers in this area. In [5, 6] Buneman and others proposed to recast the main principles of relational databases in domain theory. The ordering on objects was interpreted as partiality, i.e. \( x \leq y \) means that \( x \) is more partial than \( y \), or \( y \) is more informative than \( x \). For example,

\[
[\text{Name} \to 'Joe', \text{Age} \to \bot] \leq [\text{Name} \to 'Joe', \text{Age} \to '25']
\]

This works very well for ordering records and natural extension of the concept of scheme and the main principles of the relational design theory can be worked out, see Buneman et al. [6] and Libkin [17]. However, we face certain problems trying to generalize this approach to nested relations. With nesting, there must be a mechanism of ordering subsets of ordered sets. Unfortunately, domain theory falls short in providing us with a universal way to do it. There are various ways of making a domain out of subsets of a domain, the best known of them being the lower, the upper and the convex powerdomains. In [6] the upper powerdomain was used. While it was very convenient to use to obtain the natural join operation “for free”, it led to a number of counterintuitive observations. In Libkin [17] the lower powerdomain was used, but no justifications were given. Later, Libkin and Wong [20] gave an “update” semantics for the ordering whose meaning is being more informative. The lower powerdomain corresponded to the natural ordering of sets, while the upper powerdomain led to the ordering of so-called or-sets which in essence are sets of disjunctive possibilities (cf. [15, 29]).

However, in some cases it is desirable to retain information conveyed by both upper and lower powerdomains. This happens if the set of partial descriptions is not given explicitly, but rather approximated from below and above. Then one of the approximations, that correspond to the definite information, behaves like the lower powerdomain, but the other approximation corresponding to the possible part behaves like the upper powerdomain. First such construction was called sandwich [5].

The lower, upper and convex powerdomains can be understood as functors from the category of domains \( \text{Dom} \) to the category of domains with some additional structure. Moreover, they
are left adjoints to forgetful functors and thus they give rise to monads in \textbf{Dom}. It was shown recently that the monad structure is a very powerful programming tool \cite{30, 3}. The monads associated with lower and upper powerdomains give us a polynomial language to work with nested sets and \textit{or}-sets \cite{20}. When this language is endowed with a new primitive which is essentially an isomorphism between iterated powerdomains \cite{18}, it becomes sufficiently rich to express practically all queries on sets and \textit{or}-sets.

Since monads arise from adjunctions, above them we have yet another powerful programming tool which is the structural recursion. We do not discuss it here but refer the reader to \cite{2, 4, 3, 21} for discussion on advantage and problems of using the structural recursion.

Thus, it is the freeness property of a construction that admits an easy way of being incorporated into the syntax of a programming language. Therefore, if we want to program with approximations, we should look for their representation as free algebras.

Two such representations are already known. Gunter \cite{11} proved it for the mixed powerdomain and Puhlmann \cite{25} proved it for the snack powerdomain. Gunter's characterization uses one binary operation and one unary modal operation \(\Box\) in the spirit of Winskel \cite{31}. Puhlmann’s characterization uses two binary operations.

In this paper I will show how sandwiches and scones can be seen as free constructions. I then define the salad powerdomain and prove that it also arises as a free algebra. Then I demonstrate how all these powerdomains are related, in fact, how all of them can be represented in the salad powerdomain in a way that preserves their equational theories.

The paper is organized as follows. All necessary definitions are given in the next section. The rest is divided into three parts. In the “bottom–up” part, we start with the simplest of approximations – mixes – and go up proving algebraic characterizations for sandwiches, snacks, scones and salads. In the “top–down” part we show how to represent simple edible powerdomains in the more complicated ones, i.e. we start with salads and find out how scones are represented in salads, snacks in scones etc. Then we discuss the semantics of sets of partially defined objects given by the edible powerdomains. Conclusion and outline of further research are then given in Section 6.

### 2 Definitions

A domain in this paper is an algebraic cpo with bottom. Given a domain \(D\), \(\leq\) denotes its order and \(\mathbf{KD}\) is the set of its compact elements. Given \(X, Y \subseteq D\), lower and upper powerdomain orderings are given by

\[
X \sqsubseteq Y \iff \forall x \in X \exists y \in Y : x \leq y
\]
A subset of an ordered set is called an antichain if no two elements in it are comparable. If \((A, \preceq)\) is an ordered set and \(X \subseteq A\), then \(\max_X X\) and \(\min_X X\) are sets of maximal and minimal elements of \(X\). We will use just \(\max X\) and \(\min X\) if the ordering is understood. \(\mathcal{A}_{ne}(A)\) stands for the set of all finite antichains of \(A\).

Let \(\text{Idl}(\cdot)\) denote the ideal completion. Then the lower and upper powerdomains are defined as
\[
\text{Idl}(\langle \mathcal{A}_{ne}(KD), \preceq^\ast \rangle) \quad \text{and} \quad \text{Idl}(\langle \mathcal{A}_{ne}(KD), \preceq^\sharp \rangle)
\]
respectively. They are denoted by \(\varphi_\uparrow(D)\) and \(\varphi_\downarrow(D)\). It follows from the properties of the ideal completion that \(\langle \mathcal{A}_{ne}(KD), \preceq^\ast \rangle\) and \(\langle \mathcal{A}_{ne}(KD), \preceq^\sharp \rangle\) are posets of compact elements of \(\varphi_\uparrow(D)\) and \(\varphi_\downarrow(D)\). We reserve the notation \(\mathcal{P}_\uparrow(KD)\) and \(\mathcal{P}_\downarrow(KD)\) for them.

**Remark:** A traditional definition of the powerdomain construction is the ideal completion of \(P_{ne}(KD)\), the set of all finite subsets of \(KD\). The two can be easily shown to be equivalent. We often prefer to work with antichains because \(\preceq^\ast\) and \(\preceq^\sharp\) are partial orders on \(\mathcal{A}_{ne}(KD)\) but only preorders on \(P_{ne}(KD)\).

\(\varphi_\uparrow(D)\) is always an algebraic lattice. \(\varphi_\downarrow\) can be seen to be a functor from \(\text{Dom}\), the category of domains and continuous maps, to \(\text{AlgLat}\), the category of algebraic lattices and continuous \(\forall\)-homomorphisms. Moreover, it turns out to be left adjoint to the forgetful functor \(U : \text{AlgLat} \rightarrow \text{Dom}\). It is not hard to show that this remains true if the morphisms in both categories are restricted to those preserving compactness.

For the case of the upper powerdomain, we restrict our attention to Scott domains only. That is, the domains that happen to be complete meet-semilattices. Let \(\text{Dom}^k_A\) be the subcategory of Scott domains in which morphisms are required to preserve the meet operation and compactness. The upper powerdomain construction can be seen now as a functor \(\varphi_\downarrow : \text{Dom} \rightarrow \text{Dom}^k_A\). Then \(\varphi_\downarrow\) is left adjoint to the forgetful functor from \(\text{Dom}^k_A\) to \(\text{Dom}\).

**Remark:** This adjunction works in the greater generality. But the monad associated with the one given above is the monad used in [20] to design a language to work with or-sets. In fact, all adjunctions in this paper will work when maps are required to preserve compactness.

For an arbitrary poset \(A\), its ordering is denoted by \(\preceq\). The notation \(\preceq\) is reserved for partially ordered algebras and domains. Two elements \(x, y \in A\) are called consistent if there is \(z \in A\) such that \(x, y \preceq z\), which is denoted by \(x \uparrow y\). To make notation easier, we often write \(x\) instead of singleton set \(\{x\}\). In particular, given a pair whose first component is a singleton \(\{x\}\) and whose second component is a family of sets that happen to contain only one singleton \(\{y\}\) (that is, \(\{\{y\}\}\)), we will write \((x, \{y\})\) to denote such a pair.
3 Powerdomains as free constructions

In this section we give algebraic characterizations of edible powerdomains, starting from the simplest one and going up to the more sophisticated constructions.

Since all powerdomains are defined in terms of ideal completion of a certain partial order constructed out of its compact elements, it is enough to give an characterization of this partial order only. Since ideal completion is left adjoint to the forgetful functor from CPO to the category of posets, then, by standard technique, all results can be lifted from posets of compact elements to the powerdomains themselves. See [12] for the general technique and [11] for the mixed powerdomain.

3.1 The mixed powerdomain

The characterization of the mixed powerdomain is known [11]. For the sake of completeness, we recall it here. Let \( (A, \preceq) \) be a poset. A mix on \( A \) is a pair of finite antichains \( U \) and \( L \) such that \( \uparrow L \subseteq \uparrow U \). The mix order \( \sqsubseteq^{\text{mix}} \) is defined as follows:

\[
(U, L) \sqsubseteq^{\text{mix}} (V, M) \iff U \subseteq V \quad \text{and} \quad L \subseteq M
\]

Let \( \mathcal{P}^{\text{mix}}(A) \) denote \( A_{\text{sa}}(A) \times A_{\text{sa}}(A) \) ordered by \( \sqsubseteq^{\text{mix}} \). \( \mathcal{P}^{\text{mix}}(A) \) is a poset. The mixed powerdomain is defined as \( \mathcal{P}^{\text{mix}}(D) = \text{Idl}(\mathcal{P}^{\text{mix}}(KD)) \).

Definition. A mix algebra \( (M, +, \square, e) \) has partially ordered carrier \( M \), one monotone binary operation \( + \) and one monotone unary operation \( \square \). \( (M, +, e) \) is a semilattice with identity \( e \), and in addition the following equations must hold:

1) \( \square(x + y) = \square x + \square y \),
2) \( \square \square x = \square x \),
3) \( \square x \leq x \),
4) \( x + \square x = x \),
5) \( x + \square y \leq x \).

A mix homomorphism of two mix algebras \( (M_1, +_1, \square_1, e_1) \) and \( (M_2, +_2, \square_2, e_2) \) is a monotone map \( f : M_1 \to M_2 \) such that \( f(x +_1 y) = f(x) +_2 f(y), f(\square_1 x) = \square_2 f(x) \) and \( f(e_1) = f(e_2) \). That is, in addition to being homomorphism in the usual sense, \( f \) must be monotone as well.

\( \mathcal{P}^{\text{mix}}(A) \) can be given the structure of a mix algebra by taking the ordering \( \sqsubseteq^{\text{mix}} \) and defining \( (U, L) + (V, M) = (\min(U \cup V), \max(L \cup M)) \) and \( \square(U, L) = (U, \emptyset) \).

Theorem 1 ([11]) \( \mathcal{P}^{\text{mix}}(A) \) is the free mix algebra generated by \( A \). That is, if we define \( \eta : A \to \mathcal{P}^{\text{mix}}(A) \) by \( \eta(x) = (x, x) \), then for any mix algebra \( M \) and a monotone map \( f : A \to N \) there exists unique \( f^+ : \mathcal{P}^{\text{mix}}(A) \to M \) such that \( f^+ \circ \eta = f \). \( \square \)
3.2 The sandwich powerdomain

Let \((A, \preceq)\) be a poset. A sandwich on \(A\) is a pair of finite antichains \(U\) and \(L\) such that there exists a set \(W\) with \(U \subseteq W\) and \(L \sqsubseteq W\). Clearly, for every \(l \in L\) there exists \(u \in U\) such that \(u \parallel l\). The sandwich order \(\sqsubseteq^\ast\) is defined exactly as the mix order.

Let \(\mathcal{P}^\ast(A)\) denote \(\mathcal{A}_{\text{na}}(A) \times \mathcal{A}_{\text{na}}(A)\) ordered by \(\sqsubseteq^\ast\). Then \(\mathcal{P}^\ast(A)\) is a poset. The sandwich powerdomain is defined as \(\varphi^\ast(D) = \text{ldl}(\mathcal{P}^\ast(KD))\).

We would like to define sandwiches as a free construction over \(A\). Suppose we start with the same function \(\eta : A \rightarrow \mathcal{P}^\ast(A)\) given by \(\eta(x) = (x, x)\). For any pair \(x, y \in A\) such that \(x \parallel y\) there is a sandwich \((x, y)\) over \(A\). Thus, if we view \(\mathcal{P}^\ast(A)\) as a free algebra in a certain signature, there must be a way to construct \((x, y)\) out of pairs with identical components. But this way must use information that \(x \parallel y\) and therefore can not be “universal”.

Therefore, the information about consistency in \(A\) must be conveyed by the generating poset. We now introduce the consistent closure of \(A\) as

\[
A \uparrow A = \{(a, b) \mid a \in A, b \in A, a \parallel b\}
\]

The surprising result now says that sandwiches over \(A\) are the free mix algebra generated by the consistent closure of \(A\!\).

However, before formulating the result, let us observe that in giving the universality property it is no longer enough to require that the map from \(A \uparrow A\) to a mix algebra \(M\) be just monotone. Since we imposed additional structure on the generating poset, this structure must be preserved.

Definition Let \(M\) be a mix algebra. A monotone map \(f : A \uparrow A \rightarrow M\) is called admissible if \(f(x, y) + f(z, y) \leq f(x, y)\) and \(\square f(x, y) = \square f(x, z)\).

Define \(\eta^\uparrow : A \uparrow A \rightarrow \mathcal{P}^\ast(A)\) by \(\eta^\uparrow((x, y)) = (x, y)\).

Theorem 2 Given a poset \(A\), \(\mathcal{P}^\ast(A)\) is the free mix algebra generated by \(A \uparrow A\). That is, given a mix algebra \(M\) and an admissible map \(f : A \uparrow A \rightarrow M\), there exists a unique mix homomorphism \(f^+ : \mathcal{P}^\ast(A) \rightarrow M\) such that \(f^+ \circ \eta^\uparrow = f\).

Proof. We omit an easy verification that \(\mathcal{P}^\ast(A)\) is a mix algebra.

Let us first establish a number of useful properties of admissible maps. In what follows, \(f\) is always an admissible map from \(A \uparrow A\) to \(M\).

1) Assume \(v \preceq u\) and \(u \parallel l\). Then \(f(u, l) + f(v, l) = f(v, l)\).
First. \( f(u, l) \geq f(v, l) \). By monotonicity of \( + \), \( f(v, l) = f(v, l) + f(v, l) \leq f(v, l) + f(u, l) \). But since \( f \) is admissible, \( f(u, l) + f(v, l) \leq f(v, l) \). Hence, 1) holds.

2) Assume \( p \geq l \), \( v \not\in A \) and \( q \not\in A \). Then \( f(v, l) + f(q, p) = \square f(v, v) + f(q, p) \).

First show \( f(q, p) + f(q, l) = f(q, p) \). By monotonicity, \( f(q, p) + f(q, l) \leq f(q, p) + f(q, p) = f(q, p) \). On the other hand, \( f(q, p) + f(q, l) \geq f(q, p) + f(q, p) = f(q, p) \) which proves the equation. Since \( \square f(v, v) = \square f(v, l) \leq f(v, l) \), the \( \geq \) inequation for 2) holds. Conversely, \( f(v, l) + f(q, p) = f(v, l) + f(q, l) + f(q, p) = \square f(v, l) + f(q, l) + f(q, p) \leq \square f(v, l) + f(q, l) + f(q, p) \leq f(v, l) + f(q, l) + f(q, p) \) which shows the reverse inequation. 2) is proved.

3) If \( l \leq m \), then \( f(v, l) + f(q, m) = \square f(v, v) + f(q, m) \).

The \( \geq \) inequation is obvious. As in the proof of 2), we obtain \( f(v, l) + f(q, m) = f(v, l) + f(q, l) + f(q, m) = \square f(v, l) + f(q, l) + f(q, m) \leq \square f(v, l) + f(q, l) + f(q, m) \leq \square f(v, l) + f(q, m) = \square f(v, v) + f(q, m) \).

4) Assume \( v \leq u \). Then \( f(v, l) = f(u, l) + \square f(v, v) \).

First, \( f(u, l) + f(v, l) \leq f(u, l) = f(v, l) + f(v, l) \leq f(u, l) + f(v, l) \); hence \( f(u, l) + f(v, l) = f(v, l) \). Now we have: \( f(v, l) = f(v, l) + f(u, l) \geq f(u, l) + f(v, l) = f(u, l) + f(v, v) \). On the other hand, \( f(v, l) = f(v, l) + f(v, l) \geq f(u, l) + f(v, v) \), proving 4).

5) If \( v \geq u \), then \( \square f(u, u) + \square f(v, v) = f(v, v) \).

According to the proof of 4), \( f(u, v) + f(v, v) = f(v, v) \) and from this 5) follows immediately.

6) Assume \( u \not\in A \) and \( v \not\in A \). Then \( f(v, l) + \square f(u, u) = f(v, l) + \square f(u, u) + f(u, l) \).

Since \( \square f(u, u) = f(u, u) \), the \( \leq \) inequality holds. Since \( f(v, l) + f(u, l) \leq f(v, l) \), we obtain the reverse inequality.

Now let us come back to the statement of the theorem. Let \( S = (U, L) \) be a sandwich over \( A \) with \( U = \{u_1, \ldots, u_n\} \) and \( L = \{l_1, \ldots, l_k\} \). Since \( S \) is a sandwich, for every \( l_j \in L \) there exists \( u_i \in U \) such that \( l_j \uparrow u_i \). Let \( I \subseteq \{1, \ldots, n\} \times \{1, \ldots, k\} \) be the set of pairs of indices such that \( (i, j) \in I \Rightarrow u_i \uparrow l_j \). Then

\[
S = \sum_{(i, j) \in I} (u_i, l_j) + \square \sum_{i=1}^{n} (u_i, u_i)
\]

Using representation (1), define \( f^+ \) for an admissible \( f : A \uparrow A \rightarrow M \) as follows:

\[
f^+(S) = \sum_{(i, j) \in I} f(u_i, l_j) + \square \sum_{i=1}^{n} f(u_i, u_i)
\]
Let us show that $f^+$ is a homomorphism. Prove that $f^+$ is monotone first. Let $\mathcal{S}_1 = (U, L)$ and $\mathcal{S}_2 = (V, M)$ be two sandwiches such that $\mathcal{S}_1 \subseteq \mathcal{S}_2$, that is, $U \subseteq V$ and $L \subseteq M$. Let $\mathcal{S} = (U, M)$. Observe that $\mathcal{S}$ is a sandwich. Therefore, the proof of $f^+(\mathcal{S}_1) \leq f^+(\mathcal{S}_2)$ is contained in the following two claims.

**Claim 1:** $f^+(\mathcal{S}_1) \leq f^+(\mathcal{S})$.

**Proof of claim 1:** Since $L \subseteq M$, there is a sequence of sets $L_0 = L, L_1, \ldots, L_n = M$ such that each $L_i \subseteq L \cup M$ and either $L_{i+1} = \max(L_i \cup l)$ or $L_{i+1} = \max((L_i - L') \cup l)$ where $l' \leq l$ for all $l' \in L'$, see [20]. Then each $(U, L_i)$ is a sandwich. We must show $f^+(U, L_i) \leq f^+(U, L_{i+1})$.

Consider the first case, i.e. $L_{i+1} = \max(L_i \cup l)$. To verify $f^+(U, L_i) \leq f^+(U, L_{i+1})$ in this case, it is enough to show $\Box f(u, u') + f(u, l) \geq \Box f(u, u')$ if $u \vdash l$ and, if there is an element $l' \in L$ such that $l' \leq l$, then $f(u', l') + f(u, l) + \Box f(u, u') \geq f(u', l') + \Box f(u, u')$ if $u \vdash l'$. The first one is easy: $\Box f(u, u') + f(u, l) = \Box f(u, u') + f(u, l) = f(u, l) \geq \Box f(u, u')$. The second one follows from monotonicity of $+$: $f(u, l) + \Box f(u, u') \geq \Box f(u, l) = \Box f(u, u')$.

Consider the second case, i.e. $L_{i+1} = \max((L_i - L') \cup l)$. Assume $u \vdash l$. Then $u \vdash l'$ for any $l' \in L'$. Therefore, any summand $f(u, l)$ in (2) for $(U, L_{i+1})$ is bigger than $f(u, l')$ in (2) for $(U, L_i)$. Now suppose there is $l' \in L'$ such that $u' \vdash l'$ but $u'$ is not consistent with $l$. If $l$ is consistent with some $u \in U$, then $u \vdash l'$. Therefore, to finish the proof of claim 1, we must show that $f(u', l') + f(u, l') \leq f(u, l)$. But this follows from admissibility of $f$: $f(u', l') + f(u, l') \leq f(u, l)$. Claim 1 is proved.

**Claim 2:** $f^+(\mathcal{S}) \leq f^+(\mathcal{S}_2)$.

**Proof of claim 2:** We start with proving the following. Given a sandwich $(W, N)$ and $n \in N$, let $w_n$ be arbitrarily chosen element of $W$ such that $w_n \vdash n$. Then, given an admissible function $f$, $f^+(W, N)$ defined by (2) equals $\Sigma_{n \in N} f(w_n, n) + \Box \Sigma_{w \in W} f(w, w)$. To prove this, assume that there are two elements $w_1$ and $w_2$ in $W$ consistent with $n \in N$. Then we must show $f(w_1, n) + f(w_2, n) + \Box f(w_1, w_1) + \Box f(w_2, w_2) = f(w_1, n) + \Box f(w_1, w_1) + \Box f(w_2, w_2)$. That the left hand side is less than the right hand side follows from admissibility. On the other hand, $f(w_1, n) + \Box f(w_1, w_1) + \Box f(w_2, w_2) = f(w_1, n) + \Box f(w_2, w_2) \leq f(w_1, n) + \Box f(w_1, w_1) + \Box f(w_2, w_2)$ which proves our claim.

Now, to prove claim 2, consider $\mathcal{S}_2 = (V, M)$ and let $v_m$ be an element of $V$ consistent with $m \in M$. Since $U \subseteq V$, let $u_m$ be an element of $U$ under $v_m$. Then $u_m \uparrow m$. Also, let $u^v$ be an element of $U$ under $v \in V$. Then $\Box \Sigma_{u \in U} f(u, u) = \Box \Sigma_{u \in V} f(u^v, u^v) + \Box \Sigma_{u \neq u^v} f(u, u) \leq \Box \Sigma_{w \in W} f(w, w) \leq \Box \Sigma_{v \in V} f(v, v)$. Now, by the claim proved above, $f^+(\mathcal{S}) = \Sigma_{m \in M} f(u_m, m) + \Box \Sigma_{u \in U} f(u, u) \leq \Sigma_{m \in M} f(v_m, m) + \Box \Sigma_{v \in V} f(v, v) = f^+(\mathcal{S}_2)$ which finishes the proof of claim 2 and monotonicity of $f^+$.

Now we demonstrate that $f^+$ preserves the operations of the signature of the mix algebras. Since $\Box$ distributes over $+$, $\Box f^+(\mathcal{S}) = \Sigma_{(i,j) \in I} \Box f(u_i, l_j) + \Sigma_i \Box f(u_i, u_i)$. Since $\Box f(u_i, l_j) + \Box f(u_i, u_i) = \Box f(u_i, u_i)$, we obtain $\Box f^+(\mathcal{S}) = \Sigma_{i=1}^n \Box f(u_i, u_i) = f^+(\Box \mathcal{S})$. 

8
Let \( S_1 = (U, L) \), \( S_2 = (V, M) \). Let \( S = S_1 + S_2 = (W, N) \). Consider a pair \((u_i, l_j)\) with \((i, j) \in I\). There are three cases: this pair is either present in the representation \((1)\) of \( S \) or \( u_i \geq v_k \) for some \( v_k \in V \cap \min(U \cup V) \) or \( l_j \geq m_k \in M \cap \max(L \cup M) \).

Consider the second case. We have \( v_k \uparrow l_j \). Assume \( l_j \preceq p \) and \( p \in N \). We know that \( p \uparrow q \) for some \( q \in W \). Since \( f(v_k, l_j) + f(q, p) + \Box f(v_k, v_kk) = f(q, p) + \Box f(v, v) \) by \( 2 \), we obtain \( f^+(S) = f^+(S) + f(v_k, l_j) \). Furthermore, since \( \Box f(v_k, v_kk) + f(u_i, l_j) + f(v_k, l_j) = \Box f(v_k, v_kk) + f(v_k, l_j) \) by \( 1 \), we have \( f^+(S) = f^+(S) + f(v_k, l_j) + f(u_i, l_j) \).

Consider the third case. Assume \( u_i \) is greater or equal than some \( v \in W \) and \( m_k \uparrow q \) for \( q \in W \). Then \( f(v, l_j) + f(q, m_k) = \Box f(v, v) + f(q, m_k) \) by \( 3 \), and hence \( f^+(S) = f^+(S) + f(v, l_j) \).

Since \( f(v, l_j) = f(u, l_j) + \Box f(v, v) \) by \( 4 \), we obtain \( f^+(S) = f^+(S) + f(u_i, l_j) \).

Assume that \( u \preceq v \). Since \( \Box f(u, u) + \Box f(v, v) = \Box f(v, v) \) by \( 5 \), we obtain \( f^+(S) = f^+(S) + \Box f(u_i, u_i) \) for any \( u_i \).

All this shows that \( f^+(S) \) can be rewritten as \( f^+(S_1) + f^+(S_2) + X \) where \( X \) is a sum of some elements of form \( f(u_i, m_j) \) or \( f(v_i, l_j) \). Consider a pair \((u_i, m_j)\) such that \( u_i \uparrow m_j \). There exists \( v_k \) such that \( v_k \uparrow m_j \). Since \( f(v_k, m_j) + \Box f(u_i, u_i) = f(v_k, m_j) + \Box f(u_i, u_i) + f(u_i, m_j) \) by \( 6 \), the summand \( f(u_i, m_j) \) can be safely removed from \( X \). Thus, any summand can be removed from \( X \) and \( f^+(S) = f^+(S_1) + f^+(S_2) \). Therefore, \( f^+ \) is a homomorphism.

The uniqueness of \( f^+ \) follows from \((1)\). Since \( f^+((\eta^i(x, x))) = f(x, x) + \Box f(x, x) = f(x, x) \), we have \( f^+ \circ \eta^i = f \). The theorem is proved. \( \square \)

### 3.3 The snack powerdomain

Snacks were introduced by Peter Buneman. They were studied by Teow-Hin Ngair in his dissertation [22] and characterized by Puhlmann [25] as free distributive bisemilattices [8, 24]. Since Puhlmann's proof is not widely available, and since it is not very complicated, for the sake of completeness I shall give it here. The presentation is slightly different from the one in [25]. In particular, in [25] a slightly different equational theory is used.

**Definition.** A snack over a poset \( A \) is a pair \((U, \mathcal{L})\) where \( U \) is antichain, and \( \mathcal{L} = \{L_1, \ldots, L_k\} \) is a family of antichains satisfying the consistency condition: \( \uparrow L_i \subseteq \uparrow U \) for all \( i \). Moreover, \( \mathcal{L} \) itself is required to be an antichain with respect to \( \subseteq^+ \).

The idea is that now the lower approximation has a Smyth-type behavior as well. It is no longer true that the lower approximation is just a set of elements, each approximating an element, but rather a set of sets such that each set approximates an element. Formally, semantics of a snack is defined as

\[
\models [U, \mathcal{L}] = \{X \in \mathcal{P}_{\text{na}}(A) \mid U \subseteq^+ X \text{ and } \forall i : \uparrow L_i \cap X \neq \emptyset \}
\]
The ordering on snacks is similar to that on mixes and sandwiches. Given two snacks, \((U, \mathcal{L})\) and \((V, \mathcal{M})\),

\[
(U, \mathcal{L}) \sqsubseteq (V, \mathcal{M}) \iff U \sqsubseteq V \quad \text{and} \quad \forall L \in \mathcal{L}, \exists M \in \mathcal{M} : L \sqsubseteq M
\]

Compactly, \(\sqsubseteq = \mathbb{P} \times (\mathbb{P})^3\). The family of snacks over \(A\) is denoted by \(\mathcal{P}^\circ(A)\). The ordering \(\sqsubseteq\) gives \(\mathcal{P}^\circ(A)\) the structure of a meet-semilattice where

\[
(U, \mathcal{L}) \wedge (V, \mathcal{M}) = (\min(U \cup V), \max^2\{\min(L \cup M) \mid L \in \mathcal{L}, M \in \mathcal{M}\})
\]

**Definition** (see [8, 24]). A bisemilattice is an algebra \((B, +, \cdot)\) such that + and \(\cdot\) are semilattice operations. A bisemilattice \(B\) is called distributive if both distributive laws hold, that is: \(x(y + z) = xy + xz\) and \(x + yz = (x + y)(x + z)\). (For convenience, we often omit \(\cdot\) in formulas and equations.)

When we speak of the ordering on a bisemilattice \(B\), we mean the ordering associated with \(\cdot\), that is, \(x \leq y\) iff \(xy = x\).

\(\mathcal{P}^\circ(A)\) can be given the structure of distributive bisemilattice by making \(\cdot\) to be the inf operation above by defining + as

\[
(U, \mathcal{L}) + (V, \mathcal{M}) = (\min(U \cup V), \max^2(\mathcal{L} \cup \mathcal{M}))
\]

Observe that the empty snack \((\emptyset, \emptyset)\) is the identity for +.

**Definition.** A snack algebra is a distributive bisemilattice in which + has identity \(e\).

A homomorphism of snack algebras is a homomorphism in the usual algebraic sense. In other words, there is no need to require monotonicity as we did for mixes, because it is implied: if \(x \leq y\), then \(h(x) \cdot h(y) = h(x \cdot y) = h(x)\) and \(h(x) \leq h(y)\).

**Theorem 3** Given a poset \(A\), \(\mathcal{P}^\circ(A)\) is the free snack algebra generated by \(A\). That is, if \(\eta : A \rightarrow \mathcal{P}^\circ(A)\) is defined as \(\eta(x) = (x, \{x\})\), then for any snack algebra \(Sn\) and a monotone map \(f : A \rightarrow Sn\), there exists a unique snack homomorphism \(f^+ : \mathcal{P}^\circ(A) \rightarrow Sn\) such that \(f^+ \circ \eta = f\).

**Proof.** We omit verification that \(\mathcal{P}^\circ(A)\) is a snack algebra (in fact, the distributivity laws will be verified later in the greater generality when we consider the salad powerdomain).

Given a snack \(S = (U, \mathcal{L})\) where \(U = \{u_1, \ldots, u_n\}\) and \(\mathcal{L} = \{L_1, \ldots, L_k\}\), \(L_i = \{l_{i_1}, \ldots, l_{i_k}\}\), we have

\[
S = (\prod_{i=1}^{n} \eta(u_i))e + \sum_{i=1}^{k} \prod_{j=1}^{k_i} \eta(l_{i_j}^j)
\]

10
Then, if monotone \( f : A \to S^n \) is given, define \( f^+ : \mathcal{P}(A) \to S^n \) by

\[
f^+(S) = \left( \prod_{i=1}^{n} f(u_i) \right) e + \sum_{i=1}^{k} \prod_{j=1}^{k_i} f(l^i_j)
\]

Clearly, \( f^+(\emptyset, \emptyset) = e \) and \( f^+(\eta(x)) = f(x) \cdot e + f(x) = f(x) \). We must show that \( f^+ \) is a homomorphism.

We start with a few easy observations. First, notice that for a snack algebra \( + \) is monotone with respect to \( \leq \). Indeed, take \( a \leq b \) and observe that \( (a + b)(a + c) = a + bc = a + c \), hence \( a + b \leq a + c \). Let us now take three elements \( a \leq b \leq c \). We have: \( ae + c \leq ae + ae + c \leq ae + b + c \leq ae + c + c = ea + c \), hence \( ae + b + c = ae + c \). Furthermore, consider arbitrary \( a \) and \( b \), since \( abe(a + b) = abe \), we have \( abe \leq (a + b)e \). On the other hand, \( ae + be \) is below \( a \), \( b \) and \( e \), and hence \( ae + be \leq abe \). Thus, \( abe = (a + b)e \).

Let \( x \leq y \) in \( A \). Then \( f(x) \leq f(y) \) and hence \( f(x) \cdot f(y) = f(x) \). Therefore, if \( X \) and \( Y \) are two finite subsets of \( A \) equivalent with respect to \( \leq \), then \( \prod_{x \in X} f(x) = \prod_{y \in Y} f(y) \).

Furthermore, assume \( L \subseteq X \subseteq Y \) for \( U, X, Y \in \mathcal{P}(A) \). Then we have \( \prod_{u \in U} f(u) \cdot e \leq \prod_{x \in X} f(x) \leq \prod_{y \in Y} f(y) \) and therefore \( \prod_{u \in U} f(u) \cdot e + \prod_{x \in X} f(x) + \prod_{y \in Y} f(y) = \prod_{u \in U} f(u) \cdot e + \prod_{y \in Y} f(y) \). This observation shows that writing an expression for \( f^+(S_1 + S_2) \) and \( f^+(S_1 \cdot S_2) \) one may disregard all max and min operations. That is, for \( S_1 = (U, \mathcal{L}) \) and \( S_2 = (V, \mathcal{M}) \),

\[
(5) \quad f^+(S_1 + S_2) = \prod_{u \in U} f(u) \cdot \prod_{v \in V} f(v) \cdot e + \sum_{L \in \mathcal{L}} \prod_{l \in L} f(l) + \sum_{M \in \mathcal{M}} \prod_{m \in M} f(m)
\]

\[
(6) \quad f^+(S_1 \cdot S_2) = \prod_{u \in U} f(u) \cdot \prod_{v \in V} f(v) \cdot e + \sum_{L \in \mathcal{L}, M \in \mathcal{M}} \prod_{l \in L} f(l) \cdot \prod_{m \in M} f(m)
\]

That \( f^+(S_1 + S_2) = f^+(S_1^+) + f^+(S_2) \) follows immediately from (5).

Let us denote \( \prod_{x \in X} \) by \( \tilde{X} \). Then \( f^+(S_1 \cdot S_2) = \tilde{U} \tilde{V} e + \tilde{U} \tilde{e} \cdot \sum_{M} \tilde{M} + \tilde{V} \tilde{e} \cdot \sum_{L} \tilde{L} + \sum_{L} \tilde{L} \cdot \sum_{M} \tilde{M} \). The last summand is easily seen to be \( \sum_{L,M} \tilde{L} \cdot \tilde{M} \). Since \( \sum_{M} \tilde{M} \geq \tilde{V} \), the last summand is also greater than \( \tilde{V} \tilde{e} \cdot \sum_{L} \tilde{L} \) which can therefore be dropped. Similarly, \( \tilde{U} \tilde{e} \cdot \sum_{M} \tilde{M} \) can be dropped. Thus, \( f^+(S_1 \cdot S_2) = f^+(S_1^+) \cdot f^+(S_2) \) which shows that \( f^+ \) is a homomorphism. Its uniqueness follows from (3).

**3.4 The scone powerdomain**

Scones were introduced recently by Achim Jung and a few initial results were proved by Hermann Puhlmann. Given a poset \( A \), a *scone* over \( A \) is a pair \( (U, \mathcal{L}) \) where \( U \) is antichain, and \( \mathcal{L} = \{L_1, \ldots, L_k\} \) is a family of antichains which is itself and antichain with respect to \( \subseteq \).

In addition, a scone is required to satisfy the consistency condition: \( \forall L \in \mathcal{L} : \uparrow L \cap \uparrow U \neq \emptyset \).
Thus, the only difference between scones and snacks is the consistency condition. The ordering on scones is the snack ordering. It is not hard to show that the poset of scones over $A$, denoted by $P^s(A)$, is a meet-semilattice and the meet operation is the same as the meet for snacks (see the previous subsection).

If $x, y \in A$ and $x \triangleright y$, then $(x, \{y\})$ is a scone. Thus, we have the same problem as we had with sandwiches: it is no longer enough to start with $A$ itself as a generating poset if we want to represent scones as a free construction. That is, some information about consistency must be incorporated into the generating poset. Again, we consider the consistent closure $A \uparrow A$ of $A$ and the singleton function $\eta^i(x, y) = (x, \{y\})$.

Let us now describe the algebra. Recall that a left normal band is an algebra $(B, \cdot)$ where $\cdot$ is idempotent, associative and $x \cdot y \cdot z = x \cdot z \cdot y$ [8].

**Definition.** A scone algebra is an algebra $(Sc, +, \cdot, e)$ where $+$ is a semilattice operation with identity $e$, $\cdot$ is a left normal band operation. $+$ and $\cdot$ distribute over each other, the absorption laws hold and $e \cdot x = e$. Formally, in addition to $\cdot$ being left normal band and $+$ being semilattice operation, the following hold:

1. $x + y \cdot z = (x + y) \cdot (x + z)$;
2. $(x + y) \cdot z = x \cdot z + y \cdot z$;
3. $z \cdot (x + y) = z \cdot x + z \cdot y$;
4. $x + x \cdot y = x$;
5. $e + x = x + e = x$;
6. $e \cdot x = e$.

In other words, a scone algebra is “almost distributive lattice” – commutativity of one of the operations is replaced by the law of the left normal bands.

If $Sc$ is a scone algebra, define $x \cdot y = x \cdot y + y \cdot x$. It is an easy observation that $\cdot$ is a semilattice operation. An ordering on $Sc$ is defined according to this operation, that is, $x \leq y \iff xy = x$. Similarly to the case of snacks, this implies monotonicity of any homomorphism.

To give $P^s(A)$ the structure of a scone algebra we must show how to define $+$ and $\cdot$. The $+$ operation is defined as for snacks, and

$$(U, \mathcal{L}) \cdot (V, \mathcal{M}) = (U, \max^\sharp\{\min(L \cup M) \mid L \in \mathcal{L}, M \in \mathcal{M}\})$$

It is easy to check that $(U, \mathcal{L}) \cdot (V, \mathcal{M})$ satisfies the consistency condition. $e$ is the empty scone $(\emptyset, \emptyset)$. Similarly to the case of sandwiches, a definition of admissibility is needed to preserve the additional structure given by consistency closure of $A$.

**Definition.** Let $(Sc, +, \cdot, e)$ be a scone algebra. A monotone map $f : A \uparrow A \rightarrow Sc$ is called admissible if $f(u, l) \cdot f(v, m) = f(u, m) \cdot f(w, l)$ and $f(u, l) \cdot e = f(u, m) \cdot e$. 

12
Theorem 4 Given a poset A. $P^\alpha(A)$ is the free scone algebra generated by $A \uparrow A$. That is, for any scone algebra $Sc$ and an admissible map $f : A \uparrow A \to Sc$, there exists a unique scone homomorphism $f^+ : P^\alpha(A) \to Sc$ such that $f^+ \circ \eta^I = f$.

Proof. We shall verify the distributivity laws in the proof of algebraic characterization of the salad powerdomain in the next subsection. Distributivity laws for scones then follow from the observation that the second components of $(U, L) \cdot (V, M)$ and $(U, L) \ast (V, M)$ coincide. Equation 4) is immediate. Thus, $P^\alpha(A)$ is a scone algebra.

We now need some observations about the scone algebras. In what follows, $f$ is an admissible map from $A \uparrow A$ to a scone algebra $Sc$. The definition of admissibility can be rewritten to $f(u, l) \ast f(v, m) = f(u, l) \ast f(v, m)$ in the next subsection. Distributivity laws for scones then follow from the observation that the second components of $(U, L) \cdot (V, M)$ and $(U, L) \ast (V, M)$ coincide. Equation 4) is immediate. Thus, $P^\alpha(A)$ is a scone algebra.

Let $b \leq a$. Then $(a+c)(b+c) = (a+c) \ast (b+c) = a \ast b \ast b \ast a = c \ast ab = b+c$.

i.e. $b+c \leq a+c$.

2) $\cdot$ distributes over $\ast$.

$x(y+z) = x \ast (y+z) + (y+z) \ast x = x \ast y + y \ast x + x \ast z + z \ast x = xy + xz$.

3) If $a \leq b$, then $a \ast e \leq b \ast e$.

$(a \ast e) \ast (b \ast e) = a \ast b \ast e + b \ast a \ast e = (a \ast b + b \ast a) \ast e = (ab) \ast e = a \ast e$.

4) $f(x, y) + f(z, y) \leq f(x, y)$.

$f(x, y) + f(z, y) \cdot f(x, y) = (f(x, y) + f(z, y)) \ast f(x, y) + f(x, y) \ast (f(x, y) + f(z, y)) = (f(x, y) + f(x, y) \ast f(z, y)) + f(z, y) \ast f(x, y) = f(x, y) + f(z, y) \ast f(x, y) = f(x, y) + f(z, y)$.

5) If $a \leq b$, then $f(a, a) \ast e + f(b, b) \ast e = f(a, a) \ast e$.

First of all, $f(a, a) \ast e + f(b, b) \ast e = f(a, a) \ast e + f(b, b) \ast e = (f(a, a) + f(b, b)) \ast e \leq f(a, a) \ast e$ by 3) and 4). Furthermore, $f(a, a) = f(a, a) \ast f(a, a) \leq f(a, a) + f(b, b)$ by 1) and therefore $f(a, a) \ast e \leq (f(a, a) + f(b, b)) \ast e$ which finishes the proof.

6) If $a \leq b$ and $b \in \mathbb{R}$, then $f(x, a) \ast f(b, b) = f(x, a)$.

We have $f(x, a) \ast f(b, b) = f(x, a) \ast f(x, b) = f(x, b) \ast f(x, a)$. Hence $f(x, a) \ast f(b, b) = f(x, a) \ast f(x, b) + f(x, b) \ast f(x, a) = f(x, a) \cdot f(x, b) = f(x, a)$ because $f(x, a) \leq f(x, b)$.

7) For any $a \uparrow b$, $f(a, b) \ast f(b, a) \leq f(a, b)$.

It is easy to see that $(f(a, b) \ast f(b, a)) \cdot f(a, b) = f(a, b) \ast f(b, a)$. 

13
8) If $a \preceq b$, then $f(b, b) * f(a, a) = f(b, a)$.

By admissibility and 7), $f(b, b) * f(a, a) = f(b, a) * f(a, b) \leq f(b, a)$. On the other hand,

$$f(b, b) * f(a, a) = f(b, b) * f(a, a) + f(b, b) * f(a, a) = f(b, a) * f(b, a) * f(b, a) * f(b, a) * f(b, a) * f(b, a)$$

Hence, $f(b, a) \leq f(b, b) * f(a, a)$ which proves 8).

Since $\Pi$ is already used to denote repeated applications of $\cdot$, for many applications of $*$ we shall use $\otimes$.

Let $S = (U, L)$ be a scone over $A$. Since $\uparrow U \cap \uparrow L \neq \emptyset$ for all $L \in L$, there exists a pair $(u_i, l_{k_i})$ for every $j$ such that $u_i \uparrow l_{k_i}$. Let $i(j)$ and $k(j)$ be some indices such that $u_{i(i,j)} \uparrow l_{k(j)}$. Then $S$ can be represented as

$$S = \sum_{u \in U} \eta^1(u, u) * e + \sum_{L, l \in L} (\eta^1(u_{i(i,j)}, l_{k(j)}) * \otimes_{l \in L_j} \eta^1(l, l))$$

It is an easy observation that it does not matter how pairs $(i(j), k(j))$ are chosen.

Using (7), define

$$f^+(S) = \sum_{u \in U} f(u, u) * e + \sum_{L, l \in L} (f(u_{i(i,j)}, l_{k(j)}) * \otimes_{l \in L_j} f(l, l))$$

Our first goal is to verify that $f^+$ is well-defined, that is, it does not depend on how pairs $i(j), k(j)$ are chosen. To save space, denote $\otimes_{l \in L_j} f(l, l)$ by $\hat{L}$. First observe that any number of applications of $f$ to a consistent pair $(u, l)$ for $l \in L_j$ can be put after $f(u_{i(i,j)}, l_{k(j)})$ because, by admissibility, $f(u_{i(i,j)}, l_{k(j)}) * f(u, l) = f(u_{i(i,j)}, l_{k(j)}) * f(u, l)$ and $*$ is idempotent.

To finish the proof of well-definedness, it is enough to show that the following equation holds:

$$f(u, u) * e + f(u', u') * e + f(u, l) * \hat{L} = f(u, u) * e + f(u', u') * e + f(u', l') * \hat{L}$$

where $u, u' \in U$ and $l, l' \in L$. By distributivity, this reduces to showing that $f(u, u) * e + f(u', u') * e + f(u, l) * f(l', l') = f(u, u) * e + f(u', u') * e + f(u', l') * f(l, l)$. Because of the symmetry in this equation, it is enough to prove

$$f(u, u) * e + f(u', u') * e + f(u, l) * f(l', l') \leq f(u, u) * e + f(u', u') * e + f(u', l') * f(l, l)$$

Denote $f(u, u) * e + f(u', u') * e$ by $p$, $f(u, l) * f(l', l')$ by $q$ and $f(u', l') * f(l, l)$ by $r$. We must show $q + p \leq r + p$. By 2), $(q + p)(r + p) = rq + rp + qp + p$. By monotonicity of $+$ (see 1)), it enough to prove $qp \leq r$. We prove more. In fact, $p \leq r$. First observe that if $a \preceq b$, then $a * e \leq b * e$. Indeed, $(a * e) \cdot (b * c) = a * e + b * e = a * e$ by the same argument as in 5). Thus, we must show $p \leq f(u, l)$. Calculate $p \cdot f(u, l) = (f(u, u) + f(u', u')) * e \cdot f(u, l) = (f(u, u) + f(u', u')) * e * f(u, l) + (u, l) * (f(u, u) + f(u', u')) * e = (f(u, u) + f(u', u')) * e + f(u, l) * e = f(u, u) * e + f(u', u') * e = p$. Thus, $p \leq r$ and this finishes the proof of well-definedness.
Our next goal is to show, as we did for snacks, that if we drop max and min in defining operations on scones, formula (7) will remain true. That will make it much easier to prove that \( f^+ \) is a homomorphism.

First observe that if \( u \in U \) and \( v \succeq u \), then \( \hat{U} \circ e = \hat{U} \cup v \circ e \) (we use notation \( \hat{U} \) as a shorthand for \( \sum_{u \in U} f(u, u) \)). This follows immediately from \( 5 \).

Consider the \( L \)-part. In order to show that for \( l' \succeq l \in L \), the corresponding summand of (8) remains the same if \( f(l', l') \) is added, we must show \( f(u, l_0) \circ f(l, l) \circ f(l', l') = f(u, l_0) \circ f(l, l) \). The left hand side is equal to \( f(u, l_0) \circ f(l, l) \circ f(l', l') \) and by \( 6 \) \( f(l, l) \circ f(l, l') = f(l, l) \).

Finally, it must be shown that adding \( M \subseteq L \in L \) does not change the value of the right hand side of (8). Assume \( u \in U \), \( m \in M \) and \( l \in L \) are such that \( m \leq l \) and \( u \in \) \( f \) (we can find such because of the consistency condition and \( M \subseteq L \)). Let \( a = L \) and \( b = M \). We must show \( f(u, l) \circ a \circ f(u, m) \circ b = f(u, l) \circ a \) (it was already shown that it does not matter which consistent pair is chosen in representation (8)). Let \( a' = f(u, l) \circ a \) and \( b' = f(u, m) \circ b \).

\[ a' \cdot b' = (f(u, l) \circ f(u, m) \circ f(u, l)) \circ a \circ b = f(u, l) \circ f(u, m) \circ f(u, l) \circ a \circ b = f(u, l) \circ a \circ b = f(u, l) \circ a \circ b \]  

Since \( L \subseteq M \) and \( f(c, c) \circ f(d, d) = f(d, c) \) for \( d \succeq c \) by \( 8 \), we obtain \( a' \cdot b' = f(u, m) \circ b = b' \).

Hence \( b' \leq a' \) and \( a' + b' \leq a' \) by \( 1 \). To prove the reverse inequality, \( a' \leq a' + b' \), calculate \( (a' + b') = a' + f(u, l) \circ a \circ f(u, l) \circ f(u, m) \circ a \circ b + f(u, m) \circ f(u, l) \circ a \circ b \). By admissibility, \( a' \cdot b' = f(u, l) \circ a + f(u, l) \circ a \circ f(u, m) \circ b = a' + a' \cdot b' = a' \). Thus, \( a' \leq a' + b' \).

Now we are ready to prove that \( f^+ \) is a homomorphism. First, \( f^+(\emptyset, \emptyset) = e \circ e + e = e \).

Let \( S_1 = (U, L_1) \) and \( S_2 = (V, M) \). Writing expression (8) for \( f^+(S_1 + S_2) \) we can use \( U \cup V \) as the first component and \( L \cup M \) as the second. We know that it does not matter how we pick an element from \( U \cup V \) to be consistent with some element of a set from \( L \cup M \). For every \( L \in L \) choose \( u_L \in U \) which is consistent with some \( l_L \in L \) and similarly for every \( M \in M \) choose \( v_M \in V \) which is consistent with some \( m_M \in M \). Then we have

\[
f^+(S_1 + S_2) = \sum_{u \in U \cup V} f(u, u) \circ e + \sum_{L \in L} (f(u, l_L) \circ L) + \sum_{M \in M} (f(v_M, m_M) \circ M) = f^+(S_1) + f^+(S_2)
\]

Let \( a_L = f(u, l) \circ L \). \( c_M = f(v, m) \circ M \) where \( u \uparrow l, v \uparrow m, v \in V, u \in U, l \in L \in L \) and \( m \in M \in M \). Let \( b = \hat{U} \circ e \) and \( d = \hat{V} \circ e \). Then \( f^+(S_1) \circ f^+(S_2) = (\sum_{L \in L} (a_L + b)) \circ (\sum_{M \in M} (c_M + d)) = \sum_{L \in L, M \in M} (a_L \circ c_M + a_L \circ d + b \circ c_M + b \circ d) \). Since \( d = \hat{V} \circ e \), \( a_L \circ d = a_L \circ e \) and \( a_L \circ c_M + a_L \circ d = a_L \circ c_M + a_L \circ e = a_L \circ c_M \). Similarly, \( b \circ d = b \circ e \). Since \( b = \hat{U} \circ e, b = b \circ e \). Therefore, \( b \circ c_M = b \circ e \) and \( b \circ d = b \circ e = b \). Therefore, \( f^+(S_1) \circ f^+(S_2) = \sum_{L \in L, M \in M} (a_L \circ c_M) + b \). Consider \( a_L \circ c_M \). Since \( f(v, m) \) occurs inside the expression, by admissibility it can be changed to \( f(m, m) \). Therefore, \( a_L \circ c_M = f(u, l) \circ \hat{L} \circ \hat{M} \).
Thus.

\[ f^+(S_1) * f^+(S_2) = b + \sum_{\ell \in \mathcal{L}} f(u, \ell) * \hat{L} * \hat{M} = \sum_{u \in U} f(u, u) * e = \sum_{N \in \{L \cup M \mid L \in \mathcal{L}, M \in \mathcal{M}\}} f(u, l) * \hat{N} = f^+(S_1 * S_2) \]

Thus, \( f^+ \) is a homomorphism.

The uniqueness of \( f^+ \) follows from (7) and well-definedness of (8). Finally, \( f^+ (\eta^1(x, y)) = f(x, x) * e + f(x, y) * f(y, y) = f(x, y) * e + f(x, y) = f(x, y) \). This shows \( f^+ \circ \eta^1 = f \). Theorem is proved.

\[ \square \]

### 3.5 The salad powerdomain

In this section we describe a construction which can be seen as “all others put together with no restrictions”. This justifies the name of the salad powerdomain. Salads can be viewed as snacks or scones without the consistency condition.

**Definition.** A salad over a poset \( A \) is a pair \((U, \mathcal{L})\) where \( U \) is antichain, and \( \mathcal{L} = \{L_1, \ldots, L_k\} \) is a family of antichains which is itself an antichain with respect to \( \sqsubseteq \).

The family of all salads over \( A \) is denoted by \( \mathcal{P}^\star(A) \). It is ordered by the snack ordering. As usually, the salad powerdomain, \( \wp^\star(D) \), is defined as \( \text{Idl}(\mathcal{P}^\star(KD)) \). It is easy to see that \( \wp^\star(D) \) is isomorphic to \( \wp^\sharp(D) \times \wp^\#(D) \), where \( \wp^\#(D) \) is the iterated powerdomain: \( \wp^\#(D) = \wp^\#(\wp^\sharp(D)) (\cong \wp^\sharp(\wp^\sharp(D))) \). The isomorphism was established in [7, 13] and a simple proof was given in [18] describing explicitly isomorphism of \( \wp^\sharp(\wp^\star(A)) \) and \( \wp^\star(\wp^\sharp(A)) \).

Each of the two factors, \( \wp^\sharp(D) \) and \( \wp^\#(D) \), can be represented as a free construction [13, 12]. However, this does not give us a desired characterization of salads. Fortunately, there is a way to get one by combining the two known characterizations in a certain way. This way is described in the rest of the subsection.

**Definition:** A salad algebra, \( \langle Sd, +, \cdot, \Box, \Diamond \rangle \) is an algebra with two semilattice operations \( + \) and \( \cdot \) and two unary operation \( \Box \) and \( \Diamond \) such that the following equations hold:
1) \( x \cdot (y + z) = x \cdot y + x \cdot z \);
2) \( x = \Box x + \Diamond x \);
3) \( \Box (x + y) = \Box x + \Box y = \Box x \cdot \Box y = \Box (x \cdot y) \);
4) \( \Diamond (x + y) = \Diamond x + \Diamond y \);
5) \( \Diamond (x \cdot y) = \Diamond x \cdot \Diamond y \);
6) \( \Box x \cdot \Diamond y = \Box x \);
7) \( \Box x \cdot \Diamond y + \Diamond x = \Diamond x \);
8) \( \Diamond \Diamond x = \Diamond x \);
9) \( \Box \Box x = \Box x \);
Define an ordering $\leq$ on a salad algebra according to the operation $x \leq y$ iff $xy = x$. Then every homomorphism of salad algebras is monotone with respect to the ordering.

Define $\square Sd = \{\square x \mid x \in Sd\}$ and $\lozenge Sd = \{\lozenge x \mid x \in Sd\}$. Some useful properties of salads are summarized in the following proposition.

**Proposition 1** Given a salad algebra $Sd$, the distributivity law $x + yz = (x + y)(x + z)$ holds. Consequently, $+$, $\square$ and $\lozenge$ are monotone. In addition, the following holds:

(i) $\square x \leq x \leq \lozenge x$;

(ii) $\lozenge Sd$ is a distributive lattice;

(iii) $+$ and $\cdot$ coincide on $\square Sd$;

(iv) $\square \lozenge x = \lozenge \square y$.

**Proof.** Using 2) and distributivity law 1) calculate $(x + y)(x + z) = (\square x + \square y + \square x + \square y)(\square x + \square z + \square x + \square z) = (by~7)) = \square x + \square y + \square z + \square x + \square y \cdot \square z$. Similarly, $x + yz = \square x + \square x + (\square y + \square y)(\square z + \square z) = \square x + \square y + \square z + \square y \cdot \square z$. Hence, $(x + y)(x + z) = x + yz$. Now monotonicity of $+$ follows from the distributivity laws. That $\square$ and $\lozenge$ are monotone, follows from 4) and 6).

To prove (i), calculate $x \cdot \square x = (\square x + \square x)\square x = \square x + \square x \cdot \square x = \square x + \square x = \square x$. Moreover, $x \cdot \lozenge x = (\square x + \square x)\lozenge x = \square x \cdot \lozenge x + \lozenge x = \square x + \lozenge x = x$.

(ii) and (iii) follow immediately from the definitions.

(iii) By 7), $\square x \leq \lozenge \square y$; hence $\square \lozenge x \leq \lozenge \square y$ and by symmetry $\square \lozenge x = \lozenge \square y$. Similarly, $\square \lozenge x = \lozenge \square y$. Define $e_\square = \lozenge \square x$ and $e_\lozenge = \square \lozenge x$. The equations above show that $e_\square$ and $e_\lozenge$ are well-defined. Now calculate $e_\square + x = \lozenge \square x + x = \lozenge \square x + \lozenge x = \lozenge (\square + x) + x = \lozenge x + x = x$. Similarly, $e_\lozenge + x = \square \lozenge x + x = \square \lozenge x + \square x + x = \square (\lozenge x + x) + x = \square x + x = x$. Thus, both $e_\square$ and $e_\lozenge$ are identities for $\cdot$. Therefore, $e_\square = e_\square + e_\lozenge = e_\lozenge$. 

This lemma tells us that we can give the following equivalent definition of a salad algebra:

A salad algebra is a distributive bisemilattice $(Sd,+,\cdot)$ on which a projection $\square$ and a closure $\lozenge$ are defined such that $\square Sd$ is a semilattice. $\lozenge Sd$ is a lattice, $x = \square x + \lozenge x$ and $\forall x \in \square Sd \forall y \in \lozenge Sd: x \leq y$.

There is also one property of salad algebras that is worth mentioning and that follows directly from the definitions. Given a semilattice $(S, \vee)$ with bottom, a pair of ideals $I_1$ and $I_2$ is called a general decomposition of $S$ if bottom is the only common element of $I_1$ and $I_2$ and every $s$ in $S$ has a unique representation as $s = s_1 \vee s_2$ where $s_1 \in I_1$ and $s_2 \in I_2$ [16]. If $S$ is a bounded lattice, general decompositions become direct decompositions. For a large class of posets with partially defined lubs general decompositions are in 1-1 correspondence with neutral complemented ideals [16].
Proposition 2 Given a salad algebra \( S'd \). \( \square S'd \) and \( \Diamond S'd \) form a general decomposition of \( S'd \).

Proof. Let \( \leq_+ \) denote the ordering given by +, that is, \( x \leq_+ y \iff x + y = y \). Let \( x \leq_+ \Diamond y \). Then \( \Diamond x + \Diamond \Diamond y = \Diamond \Diamond y \), i.e. \( \Diamond x + e_\Diamond = e_\Diamond \) and \( \Diamond x = e_\Diamond \). Now \( x = \Diamond x + \Box x = e_\Diamond + \Box x = \Box x \). Hence \( x \in \Box S'd \), which shows that \( \Box S'd \) is an ideal. Similarly, \( \Diamond S'd \) is an ideal. It follows from (iii) of the lemma that \( \Box S'd \cap \Diamond S'd = \{e\} \) where \( e = e_\Box = e_\Diamond \). Finally, let \( x = \Box y + \Diamond z \). Then \( \Box x = \Box y + \Box \Diamond z = \Box y \) and similarly \( \Diamond x = \Diamond z \). Hence, \( x = \Box x + \Diamond x \) is a unique representation of \( x \) as a sum of elements from \( \Box S'd \) and \( \Diamond S'd \). Thus, \( \Box S'd \) and \( \Diamond S'd \) form a general decomposition. \( \square \)

Let us now show how the salad algebra operations are interpreted on \( \mathcal{P}^*(A) \). Operations + and \( \cdot \) are defined precisely as for snacks. For \( \Box \) and \( \Diamond \):

\[
\Box(U, \mathcal{L}) = (U, \emptyset) \quad \Diamond(U, \mathcal{L}) = (\emptyset, \mathcal{L})
\]

Theorem 5 Given a poset \( A \). \( \mathcal{P}^*(A) \) is the free salad algebra generated by \( A \). That is, if \( \eta : A \to \mathcal{P}^*(A) \) is defined by \( \eta(x) = (x, \{x\}) \), then for every monotone map \( f \) from \( A \) to a salad algebra \( S'd \) there exists a unique salad homomorphism \( f^+: \mathcal{P}^*(A) \to S'd \) such that \( f^+ \circ \eta = f \).

Proof. First verify that \( \mathcal{P}^*(A) \) is a salad algebra. We need to check the distributivity law and 7); all others are straightforward. Let \( S_1 = (U, \mathcal{L}), S_2 = (V, \mathcal{M}) \) and \( S_3 = (W, \mathcal{N}) \). Our goal is to show \( S_1 \cdot (S_2 + S_3) = S_1 \cdot S_2 + S_1 \cdot S_3 \). The first components of the left hand and the right hand sides coincide. It this case it is easier to work with filters rather than antichains - it allows us to drop max and min operations. In particular, it is enough to show that

\[
\{ \uparrow (L \cup K) | L \in \mathcal{L}, K \in \mathcal{M} \cup \mathcal{N} \} = \\
\{ \uparrow L_M | L_M \in \{L \cup M | L \in \mathcal{L}, M \in \mathcal{M} \} \} \cup \{ \uparrow L_N | L_N \in \{L \cup N | L \in \mathcal{L}, N \in \mathcal{N} \} \}
\]

Let \( C \) be an element of the left hand side, i.e. \( C = \uparrow (L \cup K) \). Without loss of generality, \( K \in \mathcal{M} \). Then \( C \) is in the right hand side. Conversely, if \( C \) is in the right hand side, say \( C = \uparrow L_M \) for \( L_M = L \cup M \), then \( C = \uparrow (L \cup M) \) and therefore is in the left hand side. This shows the equality above. Now, taking minimal elements for each filter and applying max to both collections would give us second components of the lhs and the rhs of the distributivity equation, which therefore are equal.

Now prove 7), that is, \( \Diamond(U, \mathcal{L}) \cdot \Diamond(V, \mathcal{M}) + \Diamond(U, \mathcal{L}) = \Diamond(U, \mathcal{L}) \). The first components of both sides are \( \emptyset \). The second component of the left hand side is \( \max^2(L \cup \max^2\{\min(L \cup M) | L \in \mathcal{L}, M \in \mathcal{M} \}) \). Since \( \min(L \cup M) \subseteq L \), this expression is equal to \( \max^2 \mathcal{L} = \mathcal{L} \). Hence, 7) holds. Thus, \( \mathcal{P}^*(A) \) is a salad algebra.
Now show that $P^*(A)$ is a free salad algebra. Given a salad $S = (U, L)$.

$$S = \bigodot \sum_{u \in U} \eta(u) + \bigodot \prod_{l \in L} \eta(l)$$

Therefore, given monotone $f : A \to Sd$, define

$$f^+(S) = \bigodot \sum_{u \in U} f(u) + \bigodot \prod_{l \in L} f(l)$$

We have: $f^+(\eta(x)) = f^+((x, \{x\})) = \bigodot f(x) + \bigodot f(x) = x$. Now we must show that $f^+$ is a homomorphism. First, it follows immediately from the properties of $\bigodot$ and $\bigodot$ and the fact that $e = \bigodot \odot x = \bigodot \odot y$ is the identity for $+$ (see lemma) that $f^+(\bigodot S) = \bigodot f^+(S)$ and $f^+(\bigodot S) = \bigodot f^+(S)$.

Assume $X \subseteq Y$ and let $x \in X$ below $y \in Y$. Then

$$\bigodot \sum_{x \in X} f(x) \cdot \bigodot \sum_{y \in Y} f(y) = \bigodot \left( \sum_{x \in X} f(x) + \sum_{y \in Y} f(y) \right) = \bigodot \sum_{x \in X} f(x) + \bigodot \sum_{y \in Y} f(y) = \bigodot \sum_{x \in X} f(x) + \bigodot \sum_{y \in Y} f(x)$$

Therefore, if $X$ and $Y$ are equivalent with respect to $\subseteq$: $\bigodot \sum_{x \in X} f(x) = \bigodot \sum_{y \in Y} f(y)$. Our next goal is to show that $\bigodot \prod_{x \in X} f(x) + \bigodot \prod_{y \in Y} f(y) = \bigodot \prod_{x \in X} f(x)$. Since $X \subseteq Y$, we have $\prod_{x \in X} f(x) \leq \prod_{y \in Y} f(y)$ and then the equation above follows from 7). Finally, let $x' \succeq x \in X$. Then $f(x') \succeq f(x)$ and $\prod_{x \in X} f(x) = f(x') \cdot \prod_{x \in X} f(x)$.

These three observations show that max and min operations can be disregarded when one writes an expression for $f^+$ on $S_1 + S_2$ or $S_1 \cdot S_2$. Therefore, for $S_1 = (U, L)$ and $S_2 = (V, M)$,

$$f^+(S_1 + S_2) = \bigodot \sum_{x \in U \cup V} f(x) + \bigodot \left( \prod_{l \in L} f(l) + \prod_{m \in M} f(m) \right) = f^+(S_1) + f^+(S_2)$$

To calculate $f^+(S_1 \cdot S_2)$, observe that $\bigodot \sum_{i \in I} \bigodot \sum_{j \in J} f_{ij} = \sum_{i} \bigodot f_{i} \cdot \bigodot f_{j} = \sum_{i} \bigodot f_{i}$. Therefore,

$$f^+(S_1 \cdot S_2) = \left( \bigodot \sum_{u \in U} f(u) + \bigodot \prod_{l \in L} f(l) \right) \cdot \left( \bigodot \sum_{v \in V} f(v) + \bigodot \prod_{m \in M} f(m) \right) =$$

$$\left( \bigodot \sum_{u \in U} f(u) \cdot \bigodot \sum_{v \in V} f(v) \right) + \left( \bigodot \sum_{u \in U} f(u) \cdot \bigodot \prod_{m \in M} f(m) \right) +$$

$$+ \left( \bigodot \sum_{v \in V} f(v) \cdot \bigodot \prod_{l \in L} f(l) \right) + \left( \bigodot \prod_{l \in L} f(l) \cdot \bigodot \prod_{m \in M} f(m) \right) =$$

$$\bigodot \sum_{u \in U} f(u) + \bigodot \sum_{v \in V} f(v) + \bigodot \left( \prod_{l \in L} f(l) \cdot \prod_{m \in M} f(m) \right) =$$

$$\bigodot \sum_{x \in U \cup V} f(x) + \bigodot \left( \prod_{l \in L} f(l) \cdot \prod_{m \in M} f(m) \right)$$

Thus, $f^+$ is a homomorphism. Its uniqueness follows from (9). Theorem is proved. $\square$
4 Relationship between the powerdomains

In the previous section we claimed to have gone all the way from the simplest powerdomain – mixed – to the most complicated one – salad. The purpose of this section is to justify this claim by describing relationship between the powerdomains and their algebras. That is, we will substantiate the assertion that by their “complexity” the powerdomains should be places as

Salads → Scones → Snacks → Sandwiches → Mixes

and algebras as

Salads → Scones → Snacks → Mixes

We deal with algebras in the first subsection, showing how less complicated algebras appear as reducts of the more complicated ones in the regular manner. In the second subsection we show that these reductions give us natural embeddings of the powerdomains.

4.1 Relationship between algebras

The general technique used in this subsection is the following. Given an algebra \((A, \Omega)\), let \(\Omega'\) be a subset of \(\Omega\) and \(\Omega''\) a set of derived operations. Let \(\Theta = (\Omega - \Omega') \cup \Omega''\). Then \(A\) can be considered as a \(\Theta\)-algebra which is called \(\Theta\)-reduct of \(\langle A, \Omega \rangle\) [10]. We denote a map that takes an \(\Omega\)-algebra \(\langle A, \Omega \rangle\) and returns the \(\Theta\)-algebra \(\langle A, \Theta \rangle\) by \(\varphi^{\Omega \rightarrow \Theta}\).

We now define reductions for the powerdomain algebras from the previous section.

Definition. a) Given a salad algebra \(Sd = \langle A, +, \cdot, \Box, \Diamond \rangle\), define its reducts as follows:

Scone reduct \(\varphi^{scone}(Sd) = \langle A, +, *, e \rangle\) where \(x * y = x \cdot \Diamond y\) and \(e = \Diamond \Box x\).

Snack reduct \(\varphi^{snack}(Sd) = \langle A, +, \cdot, e \rangle\) where \(e = \Diamond \Box x\).

Mix reduct \(\varphi^{mix}(Sd) = \langle A, +, \Box, e \rangle\) where \(e = \Diamond \Box x\).

b) Given a scone algebra \(Sc = \langle A, +, *, e \rangle\), define its reducts as follows:

Snack reduct \(\varphi^{scone}(Sc) = \langle A, +, \cdot, e \rangle\) where \(x \cdot y = x * y + y * x\).

Mix reduct \(\varphi^{mix}(Sc) = \langle A, +, \Box, e \rangle\) where \(\Box x = x * e\).

c) Given a snack algebra \(Sn = \langle A, +, \cdot, e \rangle\), define its mix reduct \(\varphi^{mix}(Sc)\) as \(\langle A, +, \Box, e \rangle\) where \(\Box x = x \cdot e\).
Our first goal is to show that the concepts above are well-defined, i.e., that a mix reduct is a mix algebra, scone reduct is a scone algebra etc. We then proceed to show that it does not matter which path we choose, i.e., a mix reduct of a scone reduct of a salad is a mix reduct of a salad etc.

**Proposition 3** The reducts above are well-defined.

**Proof.** We start with reducts of salads. First demonstrate that \( \varphi^{\rightarrow \Delta}(Sd) \) is a scone algebra. That \( e \) is the identity for + was already proved. Distributivity of \( * \) over + is obvious. We must show the other distributivity law: \( a + x \cdot y = (a + x) \cdot (a + y) \). To prove this, calculate

\[
a + x \cdot y = a + (a + x) \cdot y = a + a \cdot y + x \cdot y = a + a \cdot y + x \cdot y = (a + x)(a + y) = (a + x) \cdot (a + y).
\]

This proves distributivity. That \( * \) is a left normal band operation is obvious. We have \( e \cdot x = x \cdot e = x \cdot (a + x) = x \cdot a + x \cdot e = x \cdot a + x \cdot e = x \cdot e = e \). Finally, \( x + x \cdot y = x + (a + x) \cdot y = x + a + a \cdot y + x \cdot y = x + a \cdot y + x \cdot y = x \cdot y = x + x \cdot y = x \cdot y = x \). Therefore, \( \varphi^{\rightarrow \Delta}(Sd) \) is a scone algebra.

We have already shown in the previous section that + and \( \cdot \) distribute over each other; hence, \( \varphi^{\rightarrow \sigma}(Sd) \) is a snack algebra. To check that \( \varphi^{\rightarrow \text{mix}}(Sd) \) is a mix algebra, verify the equations of the mix algebra. The first two are also equations of the salad algebras, and we have shown already that \( x + \Box x = x \) and \( \Box x \leq x \). Thus, we must show \( x + \Box y \leq x \).

Calculate \( (x + \Box y) \cdot x = x + \Box y \cdot x = x + \Box x \cdot y + \Box y \cdot x = x + \Box y \cdot x = x + \Box x + \Box y = x + \Box y \). Hence, \( x + \Box y \leq x \).

Now consider reducts of sones. To show that \( \varphi^{\rightarrow \sigma}(Sc) \) is a scone algebra, we must verify the distributivity laws. One of them was verified in the proof of the characterization of sones. The other one is also easy: \( x + y \cdot z = x + y \cdot z + z \cdot y = (x + y) \cdot (z + y) \). The next step is to verify that \( \Box x = x \cdot e \) satisfies the equations of the mix algebras. We have \( x + \Box x = x + x \cdot e = x + x \cdot (x + e) = x + x \cdot (x + e) = x \cdot x \cdot e + x \cdot e \cdot x = x \cdot e = \Box x \), hence \( \Box x \leq x \). Finally, \( x + y \cdot e \cdot x = (x + y \cdot e) \cdot x + x \cdot (y \cdot e) = x + y \cdot e + x \cdot e = x + y \cdot e \).

Therefore, \( x + \Box y \leq x \) and \( \varphi^{\rightarrow \text{mix}}(Sc) \) is a mix algebra.

Finally, if in a snack algebra \( \Box x \) is defined as \( xe \), then \( x + xe = (x + x)(x + e) = x, xxe = xe \) and \( (x + ye)x = y + xe \leq x + x = x \). Thus, \( \varphi^{\rightarrow \text{mix}}(Sn) \) is a mix algebra and this finishes the proof of the proposition. \( \square \)

Our next goal is to show path independence, that is, it does not matter if we perform reduction from one algebra to another directly or via a number of steps. This can be formalized as follows.
Theorem 6 The following diagram commutes (where the arrow from $S_d$ to $S_n$ is $\varphi^{\star \rightarrow \Delta}$ and the arrow from $S_c$ to $Mix$ is $\varphi^{\Delta \rightarrow \text{mix}}$):

Proof. We have already shown that reductions are well-defined. Consider $\varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta} : S_d \to Mix$. The identity for $\star$ is $e = \Diamond \Box x$ and the box operation of the result, $\Box' x$, is defined as $\Box' x = x \star e = x \cdot \Diamond \Box x = (\Diamond x + \Diamond x) \cdot \Box x = \Diamond x \cdot \Box x + \Diamond x \cdot \Box x = \Diamond x + e = \Box x$. Hence, $\varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta} = \varphi^{\Delta \rightarrow \text{mix}}$. Now consider $\varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta} : S_d \to Mix$. The box operation of the result is $\Box' x = xe = (\Box x + \Diamond x) \cdot \Box x = \Diamond x + e = \Box x$, hence $\varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta} = \varphi^{\Delta \rightarrow \text{mix}}$. Then consider $\varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta} : S_d \to Mix$. The box operation of the result is $\Box' x = x \cdot S_n e = x \cdot e + e \cdot x = x \cdot \Diamond \Box x + \Diamond \Box x \cdot \Box x = x \cdot \Box x + \Diamond \Box x \cdot \Box x = \Box x + e = \Box x$. Thus, $\varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta} = \varphi^{\star \rightarrow \Delta}$. To show $\varphi^{\star \rightarrow \Delta} = \varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta}$, it is enough to show that $x \cdot y = x \cdot \Diamond y + y \cdot \Box x$. But this is easy: $x \cdot y = (\Box x + \Diamond x) \cdot (\Diamond y + \Diamond y) = \Box x \cdot \Diamond y + \Box x \cdot \Diamond y + \Diamond x \cdot \Diamond y = \Box x + \Diamond y + \Diamond x \cdot \Diamond y$ and $x \cdot \Diamond y + y \cdot \Box x = (\Box x + \Diamond x) \cdot \Diamond y + (\Diamond y + \Diamond y) \cdot \Box x = \Box x + \Diamond y + \Diamond x \cdot \Diamond y$. Finally, to show that $\varphi^{\Delta \rightarrow \text{mix}} = \varphi^{\Delta \rightarrow \text{mix}} \circ \varphi^{\star \rightarrow \Delta}$, observe that $x \star e + e \star x = x \star e + e = x \star e$ and therefore $\Box x$ is the same for both reductions. Theorem is proved.

4.2 Embeddings of the powerdomains

In this subsection we show that reductions from the previous subsection correspond to the embeddings of the powerdomain construction. The general idea is as follows. Assume that a poset $A$ is given and $P'$ and $P''$ are two powerdomain constructions such that $P'$ is "higher" than $P''$ in the hierarchy shown in the beginning of the section. That is, there is a reduction $\varphi$ that takes $P'(A)$ and makes it an algebra in the signature corresponding to $P''$. Depending on the generating poset for $P''(A)$, consider either $\eta(A)$ or $\eta'(A)$ which is a subset of $P''(A)$. Then the subalgebra of $\varphi(P'(A))$ generated by this subset is $P''(A)$. Moreover, this construction is "path independent" in the sense of theorem 6. To formalize it, we use the notation
The meaning of these arrows is: Take $\mathcal{P}''(A)$ and consider it as an algebra corresponding to $\mathcal{P}''$ (by means of $\varphi$). Then its subalgebra generated by $\eta(A)$ (or $\eta'(A)$) is $\mathcal{P}''(A)$.

**Theorem 7** In the following diagram all arrows are well-defined and the diagram commutes:

The arrows not shown on the diagram are:

$$[\eta(A)] \circ \varphi^{*-\Delta} : \mathcal{P}^*(A) \rightarrow \mathcal{P}^\Delta(A) \quad [\eta'(A)] \circ \varphi^{*-\Delta} : \mathcal{P}^*(A) \rightarrow \mathcal{P}^*(A)$$

$$[\eta(A)] \circ \varphi^{\Delta-\cos} : \mathcal{P}^\Delta(A) \rightarrow \mathcal{P}^\Delta(A) \quad [\eta'(A)] \circ \varphi^{\Delta-\cos} : \mathcal{P}^{\Delta}(A) \rightarrow \mathcal{P}^{\Delta}(A)$$

$$[\eta(A)] \circ \varphi^{\Delta-\cos} : \mathcal{P}^{\Delta}(A) \rightarrow \mathcal{P}^{\Delta}(A)$$

**Proof.** Full proof requires a lot of easy calculations so we only sketch it here. First observe that all definitions of new operations for reductions agree with their powerdomain interpretation. For example, given two heres $\mathcal{L} \cdot (V, M)$ in $\mathcal{P}^{\Delta}(A)$, the value of $(U, L) \cdot (V, M)$ in $\mathcal{P}^{\Delta}(A)$ is $\mathcal{P}^\Delta(\mathcal{L} \cdot (V, M)) = (U, L) \cdot (V, M) = \min(U \cup V), \max\{L \cup M | L \in \mathcal{L}, M \in \mathcal{M}\}$ which is indeed the infimum operation in $\mathcal{P}^\Delta(A)$. The verification
that other reductions agree with the powerdomain operations is also straightforward. Now representations of sandwiches (1), snacks (3), scones (7) and mixes as

\[(U, L) = \bigoplus_{u \in U} \eta(u) + \sum_{l \in L} \eta(l)\]

tell us that all arrows are well-defined. Commutativity follows in a straightforward way from the representations (1), (3), (7), (11) and theorem 6.

\[\square\]

5 Semantic issues

In this section we discuss semantics of databases with partial information given by the edible powerdomains. As it was briefly mentioned earlier, the idea of introducing those powerdomains is to approximate sets of partial descriptions. The \(U\)-part is the upper approximation, that is, any element of the approximated set must be above an element of \(U\). The \(L\)-part is the lower approximation. If the \(L\)-part is a set, it means that for every element of this set there is an element of the approximated set above it. If it is a family of sets \(L_1, \ldots, L_k\), then for every \(i\) there is an element in the approximated set above an element of \(L_i\).

Before we define the semantics formally, notice that there are two different approaches to selecting the kind of sets that can be approximated. In many domain theory motivated works [6, 16, 17, and also 14] it is assumed that the maximal elements of a domain are complete descriptions and sets of those are approximated. Others suggest that any set can be approximated, for example, [19]. The idea of the second approach comes from consideration of recursive types. For instance, solving the domain equation \(D = C \times D\) corresponding to a simple recursive record type would give us a domain whose maximal elements are infinite sequences of maximal elements of \(C\) and it is unlikely we would be interested in approximating those. If the type declaration is \(\text{person} = [\text{name: string, father: person}]\), we are interested in descriptions of finite length (ending with infinitely many bottom elements meaning null). Another example is unfinished experiments. They are just sequences of observations made, say, every day. Again, in this case partiality of information does not necessarily mean trying to approximate maximal elements, which are never reached.

Define formally two semantics of edible powerdomains. For powerdomains with multi-element \(L\)-part (snacks, scones, salads) it is:

\[\llbracket (U, \mathcal{L}) \rrbracket = \{X \in P_{\text{sn}}(A) \mid U \sqsubseteq^t X \text{ and } \forall i : \uparrow L_i \cap X \neq \emptyset\}\]

\[\llbracket (U, \mathcal{L}) \rrbracket_{\text{max}} = \{X \in P_{\text{sn}}(A^{\text{max}}) \mid U \sqsubseteq^t X \text{ and } \forall i : \uparrow L_i \cap X \neq \emptyset\}\]

and similarly for powerdomains with one-element \(L\)-part (mixes and sandwiches), where \(A^{\text{max}}\) is the set of maximal elements of \(A\). We refer the reader to [11] for discussion on semantics of mixes and sandwiches. In this section we concentrate on snacks and scones.
Let \( A \) be a three element chain \( a < b < c \) and \( S_1 = (a,b) \) and \( S_2 = (a,c) \) two snacks over \( A \). Then \([S_1]_{\text{max}} = [S_2]_{\text{max}} \) but \( S_1 \) is strictly below \( S_2 \) in the snack order. A more complicated example of incomparable \( S_1 \) and \( S_2 \) such that \([S_1]_{\text{max}} \subset [S_2]_{\text{max}} \) can also be found. Thus, the semantics in terms of maximal elements does not agree very well with the ordering of snacks which is supposed to mean being more partial. However, it is easy to show that

**Proposition 4** If \( S_1 \) and \( S_2 \) are two snacks, then \( S_1 \sqsubseteq S_2 \) iff \([S_2] \subseteq [S_1] \).

**Proof.** Let \( S_1 = (U,L) \) and \( S_2 = (V,M) \). Prove the \( 'if' \) part first. Assume \([S_2] \subseteq [S_1] \). Pick arbitrarily an element \( m_M \) from each \( M \in \mathcal{M} \). Then \( V' = V \cup \{m_M | M \in \mathcal{M} \} \in [S_2] \) and therefore \( V' \in [S_1] \) which means \( U \sqsubseteq V' \sqsubseteq V \). Hence, \( U \sqsubseteq V \). Assume that \( S_1 \not\sqsubseteq S_2 \); then \( \exists L \forall M \exists m \in M \forall l \in L : l \not\leq m \). Let \( L \in L \) be a set for which the statement above is true; then, selecting appropriate \( m \) for each \( M \in \mathcal{M} \) we obtain a set \( Q \) such that \( Q \cap M \neq \emptyset \) for all \( M \in \mathcal{M} \) and \( \forall l \in L \forall q \in Q : l \not\leq q \). In other words, \( \uparrow L \cap Q = \emptyset \). On the other hand, \( Q \in [S_2] \subseteq [S_1] \) and therefore \( \uparrow L \cap Q \neq \emptyset \) for all \( L \in L \). This contradiction shows \( S_1 \not\sqsubseteq S_2 \). To show the \( 'only if' \) part, assume \( S_1 \not\subseteq S_2 \) and \( Q \in [S_2] \). Then \( U \sqsubseteq V \sqsubseteq Q \) and, given \( L \in L \), there exist \( M \in \mathcal{M} \) such that \( \uparrow M \subseteq \uparrow L \) and therefore \( Q \cap \uparrow L \neq \emptyset \). This shows \( Q \in [S_1] \). Proposition is proved. \( \square \)

Unfortunately, this is no longer true for scones because, given the following \( A \):

![Diagram](attachment:image.png)

let \( S_1 = (a,b) \) and \( S_2 = (a,c) \) be two scones over \( A \). Then \( \{\top, \{a, \top\}\} = [S_1] = [S_2] \) but \( S_1 \) and \( S_2 \) are incomparable.

However, there is a very close connection between semantics of scones and snacks and the ordering. In some sense, the family of snacks over \( A \) is the maximal subclass of scones over \( A \) on which the semantics and the orderings agree. To formulate this rigorously, let \( S_1 \not\preceq S_2 \) iff \([S_2] \subseteq [S_1] \). Then \( \preceq \) is a preorder and the induced equivalence relation is denoted by \( \varepsilon_{\preceq} \).

**Proposition 5** For a bounded complete poset \( A \) (every two elements bounded above have a least upper bound), \((\mathcal{P}(A), \preceq) / \varepsilon_{\preceq} \cong \mathcal{P}(A)\).

**Proof.** If \( A \) is bounded complete, then for two finite sets \( U \) and \( L \) the set \( \min(\uparrow U \cap \uparrow L) \) is also finite. Hence, we define \( \psi : \mathcal{P}(A) \rightarrow \mathcal{P}(A) \) by \( \psi((U,L)) = (U, \{\min(\uparrow U \cap \uparrow L)\}) \).
Clearly, $[S] = [\nu(S)]$ and $\nu(\psi(S)) = \nu(S)$. According to proposition 4, $\psi(S)$ is the only snack in the $\varepsilon_\pi$-equivalence class of $S$. Moreover, $\psi$ is monotone because, if $U \subseteq^f V$ and $L \subseteq^f M$, then $\min(\uparrow L \cap \uparrow U) \subseteq^f \min(\uparrow M \cap \uparrow V)$. This finishes the proof of the proposition. □

Summing up, scones are the maximal subclass of salads with well-defined semantics and snacks are the maximal subclass of scones over $A$ on which the semantics and the orderings agree.

In the rest of the section we justify choosing the orderings $\sqsubseteq^{\text{mix}}$ and $\sqsubseteq^{\circ}$ using techniques of Libkin and Wong [20]. In [20] using the lower powerdomain ordering for sets and the upper powerdomain ordering for or-sets was justified in the following way. Let $X$ be a set of partial descriptions that does not contain comparable elements. How can we improve our knowledge about the situation described by $X$? One way is to take $x \in X$ and replace it by $y \geq x$; another way is to add a new element to $X$. Since we do not want comparable elements, this corresponds to the following transformations: $X \rightsquigarrow \max((X - x) \cup y)$ and $X \rightsquigarrow \max(X \cup y)$. We say that $X$ is more informative than $Y$ if a sequence of such transformations takes us from $X$ to $Y$, that is, $X \rightsquigarrow Y$. Similarly for or-sets, which are sets of disjunctive possibilities, the transformations are $X \mapsto \min(X - x)$ and $X \mapsto \min((X - x) \cup Y)$ where $y \geq x$ for all $y \in Y$. Then $\rightsquigarrow$ is the lower powerdomain ordering and $\Rightarrow$ is the upper powerdomain ordering.

We now introduce similar transformations for mixes and snacks (since the orderings used on the other powerdomains are either $\sqsubseteq^{\text{mix}}$ or $\sqsubseteq^{\circ}$, it is enough to consider these two only) and show that they give rise the respective orderings.

Updates for snacks and scones that make them more informative are:

1. $(U, \mathcal{L}) \rightarrow (U - u, \mathcal{L})$;
2. $(U, \mathcal{L}) \rightarrow (\min((U - u) \cup V), \mathcal{L})$ where $v \geq u$ for all $v \in V$;
3. $(U, \mathcal{L}) \rightarrow (U, \max^{\circ}(\mathcal{L} \cup L))$;
4. $(U, \mathcal{L}) \rightarrow (U, \max^{\circ}(\mathcal{L} - L) \cup (L - l))$;
5. $(U, \mathcal{L}) \rightarrow (U, \max^{\circ}(\mathcal{L} - L) \cup \min((L - l) \cup L'))$ where $l \leq l'$ for all $l' \in L'$.

Similarly for mixes and sandwiches we define the following transformations:

1. $(U, L) \rightarrow (U - u, L)$;
2. $(U, L) \rightarrow (\min((U - u) \cup V), L)$ where $v \geq u$ for all $v \in V$;
3. \((U, L) \rightarrow (U, L - l)\):

4. \((U, L) \rightarrow (U, \min((L - l) \cup L'))\) where \(l \leq l'\) for all \(l' \in L'\);

It is intuitively clear that all transformations lead to better descriptions. The next result shows that they completely determine the orderings.

**Theorem 8** For any two snacks (scones), \((U, L) \xrightarrow{\diamond} (V, M)\) iff \((U, L) \subseteq^\diamond (V, M)\). For any two mixes (sandwiches), \((U, L) \xrightarrow{\diamond} (V, M)\) iff \((U, L) \subseteq^{\text{mix}} (V, M)\).

**Proof.** First, it is easy to see that whenever \(S_1 \rightarrow S_2\) and both \(S_1\) and \(S_2\) are snacks (scones), it holds: \(S_1 \subseteq^\diamond S_2\). Hence, the transitive closure of \(\rightarrow\) is included in \(\subseteq^\diamond\). To prove the converse, let \((U, L) \subseteq^\diamond (V, M)\) (the proof will work for both snacks and scones). Since \(\mathcal{L}(\subseteq^\diamond)\mathcal{M}\), by [20] there is a sequence \(L \sim L_1 \sim \ldots \sim L_k \sim M\) such that \(L_i \subseteq \mathcal{L} \cup M\). In particular, each \((U, L_i)\) is a snack (scone) if \((U, L)\) and \((V, M)\) are snacks (scones). For transformation \(L_i \sim L_{i+1}\), there are two cases.

**Case 1.** \(L_{i+1} = \max^\diamond (L \cup L')\). In this case \((U, L_i) \rightarrow (U, L_{i+1})\) follows from the definitions.

**Case 2.** \(L_{i+1} = \max^\diamond ((L_i - L) \cup L')\) where \(L \subseteq^\diamond L'\). Then, by [20], there is a sequence \(L \mapsto L_1 \mapsto L_2 \mapsto \ldots \mapsto L_p \mapsto L'\) such that each \(L_j\) is a subset of \(L \cup L'\). In particular, this shows that \((U, \max^\diamond ((L_i - L) \cup L_{j+1}))\) is a snack or a scone respectively. Now there are two subcases. In the first subcase, \(L_{j+1} = \min(L_j - l)\) and then \((U, \max^\diamond ((L_i - L) \cup L_j)) \rightarrow (U, \max^\diamond ((L_i - L) \cup L_{j+1}))\) follows from the definition. Similarly, it holds for the second subcase when \(L_{j+1} = \min((L_j - l) \cup L')\).

Therefore, \((U, L_i) \xrightarrow{\diamond} (U, L_{i+1})\) which implies \((U, L) \xrightarrow{\diamond} (U, M)\). Now from [20] we have \(U \mapsto U_1 \mapsto U_2 \mapsto \ldots \mapsto U_r \mapsto V\) such that each \(U_i\) is a subset of \(U \cup V\). Since \(\uparrow V \subseteq \uparrow U\), this implies consistency condition for each \((U_i, M)\). Each \(U_i \rightarrow U_{i+1}\) is either \(U_i \rightarrow U_i - u\) or \(U_i \rightarrow \min((U_i - u) \cup U')\) where \(u \geq u\) for all \(u' \in U'\). In both cases, \((U_i, M) \rightarrow (U_{i+1}, M)\). Therefore, \((U, M) \xrightarrow{\diamond} (V, M)\) which finishes the proof of \((U, L) \xrightarrow{\diamond} (V, M)\). The result for mixes and sandwiches is easily proved along the same lines. \(\square\)

## 6 Outline of further research

The theory of edible powerdomains started just a few years ago and there are many topics to be investigated. First, the algebraic characterization given in this paper points out to an intimate connection between these constructions and various algebras with idempotent binary operations that have been extensively studied, most notably by A. Romanowska and J.D.H. Smith, see [8, 27, 26, 28]. In [26] they characterized freely generated meet-distributive bisemilattices, that is, bisemilattices satisfying only one distributive law.

27
We have seen four types of the consistency condition:

Type 1: $\uparrow L \subseteq \uparrow U$;

Type 2: $\uparrow U \cap \uparrow L \neq \emptyset$;

Type 3: $\exists W : L \sqsubseteq W$ and $U \sqsubseteq W$;

Type 4: no restrictions.

Consistency condition of each type can be used with either one-element $L$-part (as in mixes and sandwiches), or many-element $L$-part (as in snacks, scones and salads). That is, the edible powerdomains can be classified according to the $L$-part and the consistency condition as follows:

<table>
<thead>
<tr>
<th>$L$-part</th>
<th>type of consistency condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>one set</td>
<td>1</td>
</tr>
<tr>
<td>family of sets</td>
<td>snack</td>
</tr>
</tbody>
</table>

This table shows that scones are not to sandwiches what snacks are to mixes; in fact, the analog of a scone for one element $L$-part has not been characterized yet. One of the problems this table suggests is to fill out the missing entrees. This should not be too hard: the powerdomain with one-element $L$-part and type 2 consistency condition should be characterized using the mix signature with the scone-like operation $*$ and an appropriate admissibility condition. Analogs of sandwiches with multi-element $L$-part can be characterized as snack algebras with appropriate admissibility condition and the right signature for the unrestricted construction with one-element $L$-part is $\langle +, \Box, \Diamond \rangle$. The real challenge, however, is to find names for these constructions.

Edible powerdomains are used in approximations of partial data in databases and it is desirable to be able to program with them. Characterization of powerdomains as free algebras gives us two programming tools: strong monad arising from the adjunction and structural recursion. It is, however, unrealistic to expect a programmer to verify if the structural recursion is defined correctly. For example, for scone algebras that would amount to verifying a dozen of equations. A much more realistic approach is to program with unrestricted constructions (salads and their not yet discovered counterpart) and treat the consistency conditions as constraints. Since $\varphi^*(D) \simeq \varphi^*(D) \times \varphi^d(D)$, a salad is just a pair of an or-set and a set of or-sets and as such can be used by the language of Libkin and Wong [20] for or-sets. I plan to investigate the possibilities and limitations of using the language of [20] to program with approximation, and its possible extensions.
Acknowledgement: I would like to thank the Domain Theory Group, and especially Achim Jung and Hermann Puhlmann, for their hospitality during my visit to Darmstadt Technical University when this work began. Partial support was provided by AT&T Doctoral Fellowship and NSF Grant IRI-90-04137.

References


