Electronic Notes in Theoretical Computer Science 1 (1995)
URL: http://www.elsevier.nl/locate/entcs/volume1.html 21 pages

Types, Logic, and Semantics for Nested Databases

Achim Jung and Hermann Puhlmann [1]

Fachbereich Mathematik
Technische Hochschule Darmstadt
D–64289 Darmstadt
Germany
{jung,puhlmann}@mathematik.th-darmstadt.de

Abstract

This work presents first steps towards a denotational semantics for relational databases. It is argued that such a semantics will increase the chances of successfully incorporating relational databases into typed programming languages. Database relations are seen as sets of data of a common structure. The main problem therefore is to model a type of sets. We propose the *snack powerdomain* for this purpose.

Technically, the paper attempts to clarify two aspects of the domain theoretic background of this approach. We give a localic description of the snack powerdomain construction which reveals its logical simplicity. Second, we study a subdomain relation between Scott-domains on the denotational and the logical level. Again, the logical version is simple and intuitive. Such a relation is indispensable for introducing database operations such as 'natural join'.

1 Introduction

The theory of relational databases (see [16,11] for surveys) is highly developed and proves its usefulness in practice each day. The underlying mathematical structure of finite relations is simple to manipulate and may quickly be grasped even by the non-expert. Still, there are situations where the relational model fails to offer adequate formal support. We mention two. The first problem is that of *null values* or missing entries which occurs whenever certain information is unattainable (cf. [23]). While it is possible (and actually necessary) to fill the relation with fantasy values in these circumstances, it must be remembered that the usual relational operators may fail to make sense. For example, it is useless to test for equality in one attribute if in that attribute the information is not complete. The second problem is that relations are very rigid data structures in the sense that they can only be formed as sets of products

[1] Supported by Deutsche Forschungsgemeinschaft.

of basic types. A type "relation" is not allowed. This is in sharp contrast to the advances in functional programming over the last two decades which have resulted in languages treating all types on equal footing. Such an approach is highly desirable for databases as well, since it would greatly enlarge their flexibility, their adaptability to the functional environment, and, last but not least, would offer possibilities for optimization.

This paper adds to a research programme (initiated by Peter Buneman around 1985) which employs ordered structures to overcome these problems. First results were reported in [4]. There it was shown that by using ordered sets null values can be smoothly incorporated into the relational model without jeopardizing standard results and concepts. However, the problem that the collection of all (generalized) relations is of a different nature and hence is not subsumed by the new model remains. Proposals to remedy this situation were made and studied by Leonid Libkin in the last few years, [13,14]. From these studies it emerges that the concept of a finite anti-chain, which took the role of a relation in [4], has to be replaced by a more refined construction. Conceptually convincing and mathematically satisfying appears to be the idea of a *snack*, invented by Buneman and studied in [17,19]. (Related structures appear in [7,15].)

Snacks can be explained quite easily: If we take the point of view that a relation is basically a set structured in a certain way then once partial information comes into the picture, the natural choice is to work with *approximations to sets*. This approximation can be achieved in two different ways (nicely explicated in [7]), the first of which tries to enclose the set from outside, delineating an outer bound in which the set must lie. The second will give ranges (contained in the outer range) for which it is guaranteed that they will be met by the approximated set. One snack is better or more precise than another if it offers a tighter outer bound and more (and tighter) inner ranges.

Using the concept of snacks with the ordering described we are in the pleasing situation that applied to a Scott-domain (domain theoretic terminology follows [2]) we will get back a Scott-domain of snacks. The problem of higher order relations appears to be solved. However, we clearly need more evidence that the construction is in line with database needs. The present paper offers mathematical background on the basis of which the applicability of the snack construction has been demonstrated, [20]. This application concerns primarily the passage between different layers of nestedness in higher order relations. The study of these higher order relations (or *complex objects*, as they are sometimes called) was initiated in [21]. A most elegant approach, from the syntactical point of view, can be found in [3]. However, it is also a fact that nesting and unnesting of traditional (un-ordered) relations can contain unpleasant surprises, see [18], for example.

The mathematical tool developed in the sequel is based on Abramsky's "Domain Theory in Logical Form", [1], in which data elements are uniquely described and essentially replaced by the sets of properties which they satisfy. Our first result reports the surprising finding that the snack powerdomain construction, while of formidable domain theoretic complexity, has a simple and elegant logical counterpart. A basic ingredient of the theory in [4] was the notion of a *semi-factor* intended to serve as a substitute for relational schemes. Once more, we succeed in characterizing these in simple logical terms, revealing an analogy with conservative extensions in Logic. Finally, we employ the logical language to show that the strong sub-domain relation harmonizes well with an extensive list of database type constructions.

2 The Snack Powerlocale

The first step in the passage from the standard relational to the domain theoretic model is to replace the *sets* of basic values by *flat domains*. This is achieved by adjoining a least or bottom element to the set, representing missing information. While the use of more elaborate ordered structures (such as in [23]) may also be appropriate, flat domains already give the correct general picture, where the maximal elements constitute the complete data elements we are interested in and all other elements are approximations to these. (In a flat domain there is only one approximant, the bottom element.) Snacks, then, formalize the idea of approximating *sets* of maximal elements by giving an outer bound and specific information about individual elements, which we mentioned in the introduction. One of the advantages of having approximating elements around is that we can use *finitely* many of them to describe *infinite* sets of maximal elements.

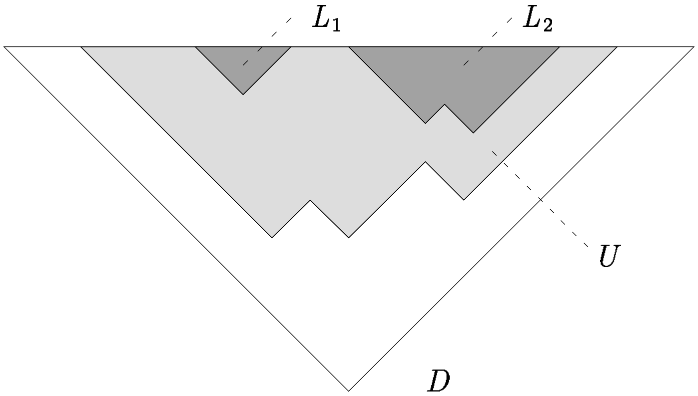
In order to give the precise definitions, we first fix some notation. For an ordered set $(P, \leq)$ we denote by $\mathcal{U}(P)$ the set of all finitely generated upper subsets of P ordered by the Smyth or upper order ($\leq^\sharp$). This is, for upper sets, just the superset relation. To exclude the empty set we write $\mathcal{U}_{\neq\emptyset}(P)$. The set of finite anti-chains of $\mathcal{U}_{\neq\emptyset}(P)$, ordered by the Hoare or lower order ($\leq^\flat$), is denoted by $\mathcal{L}(\mathcal{U}_{\neq\emptyset}(P))$.

Definition 2.1 A *snack* on an ordered set P is a pair $(U, L) \in \mathcal{U}(P) \times \mathcal{L}(\mathcal{U}_{\neq\emptyset}(P))$ such that $L_i \subseteq U$ for all $L_i \in L$. The set of all snacks on P is denoted by $\mathcal{S}^0(P)$.

The set $\mathcal{S}^0(P)$ is itself an ordered set. The order is inherited from $\mathcal{U}(P) \times \mathcal{L}(\mathcal{U}_{\neq\emptyset}(P))$, i. e. $(U, L) \leq (U', L')$ iff $U \supseteq U'$ and for all $L_i \in L$ there is an $L'_j \in L'$ such that $L_i \supseteq L'_j$.

If D is a Scott-domain, then the ideal completion $\mathsf{P}^S(D) := \mathsf{idl}(\mathcal{S}^0(KD))$ is the *snack powerdomain* of D.

A typical snack looks like the following diagram. Here, the first component U is to be read as the outer bound of the approximated set S, which we think of as a subset of $U \cap D_{\max}$. The second component L consists of L_1 and L_2. Each of these gives specific information about an individual element of S in the sense that there must be an element of S in each L_i. From this it is clear that $L_i \subseteq U$ since this element cannot be outside the outer bound U. With this interpretation in mind, the ordering on the set of snacks is easy to understand. In order to get a snack (U', L') that is better than the snack (U, L) one has to shrink the outer bound, i. e. $U' \subseteq U$, and each set approximating an individual element of S. This means that for each $L_i \in L$ there must be an $L'_j \in L'$ with $L'_j \subseteq L_i$. Note that the introduction of "new" sets $L'_j \in L'$ which are not related to the sets $L_i \in L$ is allowed so that one may add information about further elements of S.



We now could proceed with defining a type system for nested domain theoretic databases and operations on them. The definitions, however, would be complicated and hard to read. More insight is gained by employing Abramsky's method of describing data elements by the properties they satisfy [1]. The underlying mathematical structure is that of a prelocale.

Definition 2.2 A *coherent algebraic prelocale* A is a preordered algebra with two binary operations $\vee$ and $\wedge$, two nullary operations 0 and 1, and unary predicates E and T on A, such that $a \vee b$ is a supremum for $\{a, b\}$, $a \wedge b$ is an infimum for $\{a, b\}$, 0 is a least, and 1 is a largest element. The preorder on A is denoted by $\lesssim$, the corresponding equivalence relation by $\approx$. The predicate $\mathsf{E}(a)$ is required to hold if and only if a is not a least, the predicate $\mathsf{T}(a)$ if and only if a is not a largest element. Finally, every element of A must be equivalent to a finite join of $\vee$-primes.

A *Scott prelocale* has the additional property that the infimum of finite sets of $\vee$-primes is again $\vee$-prime or equivalent to 0.

This is not the place to explicate this definition and we must refer to [1,2,22,9,6] for a more gentle introduction into this theory. But it will help to think of a Scott prelocale as a syntactic description of the set $\mathsf{K}\Omega D$ of compact open subsets of a Scott-domain D. By $\mathsf{T}(a)$ we encode the information that the compact open subset described by a does not contain the bottom element of the domain ("Termination"). Similarly, we write $\mathsf{E}(a)$ to express the fact that a does not correspond to the empty set. To retrieve the domain described by a prelocale, the functor spec is used. Applied to a prelocale, it yields the ordered set of prime filters of the locale which is a domain.

Definition 2.3 Let A and B be Scott prelocales. A function $\phi\colon A \to B$ is called a *pre-isomorphism* if it is surjective, order-preserving and order-reflecting. If A is a Scott prelocale and D is a Scott domain and if further there is a pre-isomorphism $[\![\cdot]\!]\colon A \to \mathsf{K}\Omega(D)$ then we say that A is a *localic description* of D via $[\![\cdot]\!]$.

When searching for the localic description of the snack powerdomain we are not left alone in entirely new territory. The Plotkin- [1] and the mixed powerconstruction [7] are a good guideline. As for both of these, each element a of the underlying prelocale A gives us two generators $\Box a$ and $\Diamond a$ of the powerlocale. Axioms and rules of the preorder are then found by checking their validity in prelocales of compact open sets of a domain. Thus we get:

Definition 2.4 Let A be a prelocale which is a localic description of the Scott-domain D via $[\![\cdot]\!]_A$. We define the *snack powerlocale* $\mathsf{P}^S(A)$ over A as the term algebra over the generators

$$G_P = \{\Box a \mid a \in A\} \cup \{\Diamond a \mid a \in A\}$$

with the interpretation function $[\![\cdot]\!]\colon \mathsf{P}^S(A) \to \mathsf{K}\Omega(\mathsf{P}^S D)$ defined by

$$[\![\Box a]\!] = \{s \in \mathsf{P}^S(D) \mid \exists (U,L) \in s.\ U \subseteq [\![a]\!]_A\}$$
$$[\![\Diamond a]\!] = \{s \in \mathsf{P}^S(D) \mid \exists (U,L) \in s.\ \exists L_i \in L.\ L_i \subseteq [\![a]\!]_A\}$$

on the generators and extended to $\mathsf{P}^S(A)$ as a lattice homomorphism.

Preorder, E- and T-predicate are defined as follows:

Axioms.

$(\Box - \wedge)$	$\Box(\bigwedge_{i\in I} a_i) \approx \bigwedge_{i\in I} \Box a_i \qquad (I \text{ finite}),$
$(\Diamond - 0)$	$\Diamond(0) \approx 0,$
$(\Diamond - \wedge)$	$\Box a \wedge \Diamond b \lesssim \Diamond(a \wedge b),$
(dist)	$a \wedge (b \vee c) \approx (a \wedge b) \vee (a \wedge c),$
$(\mathsf{E} - 1)$	$\mathsf{E}(1),$
$(\mathsf{T} - 0)$	$\mathsf{T}(0),$
$(P - \mathsf{E} - \Box)$	$\mathsf{E}(\Box a).$

Rules.

$(P - \lesssim)$	If $a \lesssim b$ then $\Box a \lesssim \Box b$ and $\Diamond a \lesssim \Diamond b$,
$(P - \mathsf{E} - \Diamond)$	If $\mathsf{E}(a)$ then $\mathsf{E}(\Diamond a)$,
$(P - \mathsf{T} - \Box)$	If $\mathsf{T}(a)$ then $\mathsf{T}(\Box a)$,
$(P - \mathsf{T} - \Diamond)$	If $\mathsf{E}(a)$ then $\mathsf{T}(\Diamond a)$.

Further axioms and rules are needed which ensure that $\vee$ becomes a supremum, etc. We omit them for readability.

The claim, of course, is that with A being a localic description of the Scott-domain D the snack powerlocale $\mathsf{P}^S(A)$ is a localic description of the snack powerdomain $\mathsf{P}^S(D)$. The proof of this can be organized in a number of steps as given in [2]. Some of these steps do not depend on the particular construction considered. For these the reader is referred to [2].

First of all we show:

Lemma 2.5 *The interpretation function* $[\![\cdot]\!]$ *maps elements of* $\mathsf{P}^S(A)$ *to compact open subsets of* $\mathsf{P}^S(D)$.

Proof. The proof is by induction on the structure of the formula.

First note that the compact open subsets of a Scott-domain are those which are generated by finitely many compact elements of the domain. This immediately shows that $[\![0]\!] = \emptyset$ and $[\![1]\!] = \mathsf{P}^S(D)$ are compact open.

For generators of $\mathsf{P}^S(A)$ we can actually show more than that their interpretation is compact open. It is of the form $\uparrow c$ for c compact in D.

Consider a generator of the form $\Box a$. The least snack (U, L) with $U \subseteq [\![a]\!]_A$ is $([\![a]\!]_A, \emptyset)$. As $[\![a]\!]_A$ is a compact open set, this really is a snack. Every ideal in $[\![\Box a]\!]$ must contain it. The ideals which do are exactly $\downarrow([\![a]\!]_A, \emptyset)$ and all those larger than this. Hence $[\![\Box a]\!] = \uparrow_{\mathsf{P}^S(D)}(\downarrow([\![a]\!]_A, \emptyset))$, where $\downarrow([\![a]\!]_A, \emptyset))$ is compact in $\mathsf{P}^S(D)$.

For generators of the form $\Diamond a$ with $a \not\approx 0$ the least snack $(U, \{L_1, \ldots, L_k\})$ which has some $L_i \in \{L_1, \ldots, L_k\}$ with $L_i \subseteq [\![a]\!]_A$ is $(D, \{[\![a]\!]_A\})$. As above we conclude that $[\![\Diamond a]\!] = \uparrow_{\mathsf{P}^S(D)}(\downarrow(D, \{[\![a]\!]_A\}))$. If $a \approx 0$ then $[\![\Diamond 0]\!] = [\![0]\!] = \emptyset$ which is compact open.

Finally, in a Scott-domain both, union and intersection of compact open sets, are compact open. So with the assumption that $[\![\varphi]\!]$ and $[\![\psi]\!]$ are compact open we can conclude that so are $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \cap [\![\psi]\!]$ and $[\![\varphi \vee \psi]\!] = [\![\varphi]\!] \cup [\![\psi]\!]$. $\Box$

Next we have to prove soundness of the axioms and rules, i. e. that they translate via $[\![\cdot]\!]$ into valid statements about compact open subsets of $\mathsf{P}^S(D)$.

We begin with showing $(P - \precsim)$ to be sound. Given $a \precsim b$ in the prelocale A, we have $[\![a]\!]_A \subseteq [\![b]\!]_A$. If $s \in [\![\Box a]\!]$ then there is a snack $(U, L) \in s$ with $U \subseteq [\![a]\!]_A \subseteq [\![b]\!]_A$, whence it follows that $s \in [\![\Box b]\!]$. In the same way we get $[\![\Diamond a]\!] \subseteq [\![\Diamond b]\!]$: If $s \in [\![\Diamond a]\!]$ then there is a snack $(U, L) \in s$ with some $L_i \in L$ such that $L_i \subseteq [\![a]\!]_A \subseteq [\![b]\!]_A$, i. e. $s \in [\![\Diamond b]\!]$.

For the axiom $(\Box - \wedge)$, $[\![\Box(\bigwedge_{i \in I} a_i)]\!] \subseteq [\![\bigwedge_{i \in I} \Box a_i]\!]$ follows using soundness of $(P - \precsim)$. For the other direction, assume $s \in [\![\bigwedge_{i \in I} \Box a_i]\!] = \bigcap_{i \in I} [\![\Box a_i]\!]$. Then for all $i \in I$ there are snacks $(U^i, L^i) \in s$ with $U^i \subseteq [\![a_i]\!]_A$. As s is downward closed, it contains the snacks $(U^i, \emptyset)$ for all $i \in I$, and because it is directed we can conclude $(\bigcap_{i \in I} U^i, \emptyset) \in s$. Since $\bigcap_{i \in I} U^i \subseteq \bigcap_{i \in I} [\![a_i]\!]_A = [\![\bigwedge_{i \in I} a_i]\!]_A$ we have $s \in [\![\Box \bigwedge_{i \in I} a_i]\!]$.

Next we show soundness of $(\Diamond - 0)$. Of course, $0 \precsim \Diamond 0$. For the other implication, note that $[\![\Diamond 0]\!] = \{s \in \mathsf{P}^S(D) \mid \exists (U, L) \in s.\ \exists L_i \in L.\ L_i \subseteq \emptyset\}$ must be empty since the sets L_i in the second component of a snack are required to be nonempty.

Next for $(\Diamond - \wedge)$: If $s \in [\![\Box a \wedge \Diamond b]\!]$ then there is some $(U, L) \in s$ such that $U \subseteq [\![a]\!]_A$ and there is some $L_i \in L$ with $L_i \subseteq [\![b]\!]_A$. As $L_i \subseteq U$ we have $L_i \subseteq [\![a]\!]_A \cap [\![b]\!]_A = [\![a \wedge b]\!]_A$, i. e. $s \in [\![\Diamond(a \wedge b)]\!]$.

The axiom (dist) of distributivity is sound because $\vee$ and $\wedge$ translate via $[\![\cdot]\!]$ into the set-theoretic $\cup$ and $\cap$ for which we have distributivity.

The axioms $(\mathsf{E}-1)$ and $(\mathsf{T}-0)$ are sound because $[\![1]\!] = \mathsf{P}^S(D) \neq \emptyset = [\![0]\!]$. Above $[\![\Box a]\!]$ was shown to be $\uparrow_{\mathsf{P}^S(D)}(\downarrow([\![a]\!]_A, \emptyset)) \neq \emptyset$ from which we get soundness of $(P-\mathsf{E}-\Box)$. If $[\![a]\!]_A \neq D$, i.e. $\mathsf{T}(a)$, then $\uparrow_{\mathsf{P}^S(D)}(\downarrow([\![a]\!]_A, \emptyset)) \neq \mathsf{P}^S(D)$, so that $(P-\mathsf{T}-\Box)$ is sound. We also gave $[\![\Diamond a]\!]$ explicitly for $a \not\approx 0$, i.e. $\mathsf{E}(a)$. It could be seen that $[\![\Diamond a]\!] \neq \emptyset$ and $[\![\Diamond a]\!] \neq \mathsf{P}^S(D)$ in this case, so $(P-\mathsf{E}-\Diamond)$ and $(P-\mathsf{T}-\Diamond)$ are sound.

So we have shown:

Lemma 2.6 *The axioms and rules of the snack powerlocale construction are sound with respect to the interpretation function* $[\![\cdot]\!]$. $\Box$

Before going on we shall make a remark on the $\vee$-prime elements. They were made explicit in [2] by introducing the "C-predicate" which holds exactly for the $\vee$-prime elements. There this is needed to formulate the localic description of the function space. We will not use the function space and therefore can do without it. Instead we introduced the E-predicate to express that an element is not equivalent to 0, i.e. there Exists an element in the interpretation of it. In a prelocale, each element is required to be the finite join of $\vee$-primes. The empty join of $\vee$-primes is 0 and in this way we could have simulated the E-predicate by the C-predicate. However, in the application to databases it seems natural to ask if a database object is not empty so that we opted for including the corresponding predicate directly.

Still, it is useful to know how the $\vee$-prime elements look if we want to show that each element in the constructed structure is the join of finitely many $\vee$-primes. We claim that in the case of the snack powerlocale the $\vee$-prime elements are equivalent to those of the form

$$(\star) \qquad \Box a \wedge \bigwedge_{i=1}^{n} \Diamond b_i \qquad (0 \not\approx b_i \lesssim a \text{ for all } i \in \{1, \ldots, n\}.)$$

From above we know how generators $\Box a$ and $\Diamond b_i$ are interpreted. They are $\vee$-primes in the lattice of compact open subsets of $\mathsf{P}^S(D)$. The join of the generating snacks generates the interpretation of the meet of these generators. So $[\![\Box a \wedge \bigwedge_{i=1}^{n} \Diamond b_i]\!] = \uparrow_{\mathsf{P}^S(D)}\downarrow([\![a]\!]_A, \{[\![b_i]\!]_A \mid i = 1, \ldots, n\})$ if the b_i are as given above. Clearly, this is $\vee$-prime in $\mathsf{K}\Omega\mathsf{P}^S(D)$ so that our criterion for $\vee$-primeness in $\mathsf{P}^S(A)$ is sound. We postpone the proof of its completeness and first show the following:

Lemma 2.7 *Let A be a coherent prelocale. Each element $x \in \mathsf{P}^S(A)$ is the join of finitely many elements of the form $(\star)$.*

Proof. We first observe that $(\Diamond - \wedge)$ and the monotonicity of $\Box$ and $\Diamond$ imply the equivalence

$$(D) \qquad \Box a \wedge \Diamond b \approx \Box a \wedge \Diamond(a \wedge b)\,.$$

Now, given an expression in $\mathsf{P}^S(A)$ we first transform it into a disjunction of conjunctions by using the distributivity axiom. Thus it suffices to represent a term of the form

$$\bigwedge_{i \in I} \Box a_i \wedge \bigwedge_{j \in J} \Diamond b_j$$

as a disjunction of $\vee$-primes. But we can simplify further. Using $(\Box - \wedge)$ we can pack all $\Box$-generators into a single term $\Box a$ and by (D) we can assume that for each $j \in J$ we have $b_j \lesssim a$. So we get

$$\Box a \wedge \bigwedge_{j \in J} \Diamond b_j.$$

This is of the form $\star$ unless there is some $b_j \approx 0$. If that is the case, we make use of $(\Diamond - 0)$ and find the expression to be 0. So we can omit this expression in the considered disjunction of conjunctions and end up with a finite join of elements of the required form. (If it happens that we have to omit all conjunctions then we have the empty join of such elements.) $\Box$

With this being established we can restrict ourselves to showing completeness and definability for $\vee$-primes:

Lemma 2.8 *Let $\varphi, \psi \in \mathsf{P}^S(A)$ be of the form $(\star)$ and $[\![\varphi]\!] \subseteq [\![\psi]\!]$. Then $\varphi \lesssim \psi$.*

Proof. We have

$$\begin{aligned} \varphi &= \Box a \wedge \bigwedge_{i \in I} \Diamond b_i \qquad (0 \not\approx b_i \lesssim a \text{ for all } i \in I), \\ \psi &= \Box c \wedge \bigwedge_{j \in J} \Diamond d_j \qquad (0 \not\approx d_j \lesssim c \text{ for all } j \in J). \end{aligned}$$

From above we know what $[\![\varphi]\!]$ and $[\![\psi]\!]$ are: $[\![\varphi]\!] = \uparrow_{\mathsf{P}^S(D)} \downarrow([\![a]\!]_A, \{[\![b_i]\!]_A \mid i \in I\})$ and $[\![\psi]\!] = \uparrow_{\mathsf{P}^S(D)} \downarrow([\![c]\!]_A, \{[\![d_j]\!]_A \mid j \in J\})$. Because of $[\![\varphi]\!] \subseteq [\![\psi]\!]$ the snack generating $[\![\varphi]\!]$ is larger than the snack generating $[\![\psi]\!]$. Hence $[\![a]\!]_A \subseteq [\![c]\!]_A$ and for all $j \in J$ we can find some $i_j \in I$ with $[\![d_j]\!]_A \supseteq [\![b_{i_j}]\!]_A$. Using completeness of $[\![\cdot]\!]_A$ we infer that $a \lesssim c$ and $d_j \gtrsim b_{i_j}$ for all $j \in J$. Using monotonicity of $\Box$ and $\Diamond$ we get

$$\varphi = \Box a \wedge \bigwedge_{i \in I} \Diamond b_i \lesssim \Box a \wedge \bigwedge_{j \in J} \Diamond b_{i_j} \lesssim \Box c \wedge \bigwedge_{j \in J} \Diamond d_j = \psi.$$

$\Box$

This shows that the elements of the form $(\star)$ are in fact $\vee$-prime. We now show that all $\vee$-primes in $\mathsf{K}\Omega\mathsf{P}^S(D)$ are the interpretation of an element of the form $(\star)$.

Lemma 2.9 *For each $\vee$-prime element V of $\mathsf{K}\Omega\mathsf{P}^S(D)$ there is an element $\varphi \in \mathsf{P}^S(A)$ of the form $(\star)$ with $[\![\varphi]\!] = V$.*

Proof. Being $\vee$-prime in $\mathsf{K}\Omega\mathsf{P}^S(D)$, the set V is generated by an ideal that itself is generated by one snack $(U, \{L_1, \ldots, L_k\})$. As A is a localic description of D and $U, L_1, \ldots, L_k$ are compact open sets in D there are elements $a, a_1, \ldots, a_k$ of A with $[\![a]\!]_A = U$ and $[\![a_i]\!]_A = L_i$ for $i = 1, \ldots, k$. Moreover, each L_i $(i = 1, \ldots, k)$ is a nonempty subset of U, i. e. $0 \not\approx a_i \lesssim a$ for all

$i \in \{1, \ldots, k\}$. It is easily seen that $\Box a \wedge \bigwedge_{i=1}^{k} \Diamond a_i$ is of the required form and has the interpretation $\uparrow_{\mathsf{P}^S(D)}\downarrow(U, \{L_1, \ldots, L_k\}) = V$. $\Box$

In a general manner we can now conclude that $[\![\cdot]\!]$ is a pre-isomorphism, see [2]. The pre-isomorphism is lifted to an isomorphism via Stone duality. We finally wish to show that this isomorphism is natural with respect to the sub-prelocale relation.

Given a Scott prelocale B describing the Scott domain E and a sub-prelocale A of B describing the Scott domain D, there is an embedding from D to E which can be extended to compact open subsets of D: One embeds the generators of the compact open set and takes the upper set which is generated by the resulting set. For us it remains to show that the interpretation $[\![c]\!]_{\mathsf{P}^S{}_A}$ of a $\vee$-prime element $c \in \mathsf{P}^S(A)$ is thus mapped to the interpretation $[\![c]\!]_{\mathsf{P}^S{}_B}$. For a $\vee$-prime $c = \Box a \wedge \bigwedge_{i \in I} \Diamond b_i \in \mathsf{P}^S(A)$ we have $[\![c]\!]_{\mathsf{P}^S{}_A} = \uparrow_{\mathsf{P}^S{}_D}\downarrow([\![a]\!]_A, \{[\![b_i]\!]_A \mid i \in I\})$ and $[\![c]\!]_{\mathsf{P}^S{}_B} = \uparrow_{\mathsf{P}^S{}_E}\downarrow([\![a]\!]_B, \{[\![b_i]\!]_B \mid i \in I\})$. We only have to embed the compact element $\downarrow([\![a]\!]_A, \{[\![b_i]\!]_A \mid i \in I\})$ from $\mathsf{P}^S D$ into $\mathsf{P}^S E$. This is done by embedding the generators of the snack, i. e. the generators of $[\![a]\!]_A$ and the $[\![b_i]\!]_A$, and using the resulting elements to build a snack of the same structure. Practically, this means that $[\![a]\!]_A$ is replaced by $[\![a]\!]_B$ and each $[\![b_i]\!]_A$ is replaced by $[\![b_i]\!]_B$ and the structure of the formula is not altered. It thus becomes apparent that $[\![c]\!]_{\mathsf{P}^S{}_A}$ is mapped to $[\![c]\!]_{\mathsf{P}^S{}_B}$. From there, the general technique leads to the desired result:

Theorem 2.10 *Let A be a Scott-domain prelocale. Then*

$$\mathsf{P}^S(\mathsf{spec}A) \cong \mathsf{spec}(\mathsf{P}^S(A))$$

and this isomorphism is natural with respect to the sub-prelocale relation. $\Box$

3 Substructures

When proposing the use of Scott-domains to generalize relational databases [4], Buneman, Jung and Ohori found a suitable structure to play the role of a relational scheme in what was later called a semi-factor [10]:

Definition 3.1 A *stable subdomain* A in a Scott-domain D is a downward closed subset of D which is closed under all existing joins.

The projection p_A onto A is defined by $p_A(x) = \bigvee\{y \in A \mid y \leq x\}$ for all $x \in D$.

A stable subdomain A is called *semi-factor* if for all $x \in D$, $y \in A$ such that $p_A(x) \leq y$ it follows that $\{x, y\}$ is bounded.

Our aim is to define a type-system for databases with a subtype relation that is respected by the type constructors. Each type will have a domain associated with it, and in [4] the idea was that on the side of domains semi-factors of a domain will correspond to subtypes of a given type. It was shown that most type constructors of interest then preserve the subtype relation.

If, however, we add the snack powerconstruction as a means of building relation-valued types, this property no longer holds. A tedious proof reveals that the snack powerdomain of any Scott-domain has only trivial semi-factors. Thus we cannot use semi-factors to characterize subtypes.

A slight modification improves the situation. We refrain from requiring downward closedness. Additionally, the substructure will no longer have to be a subset of the domain. A connection via an embedding-projection pair will suffice.

Definition 3.2 A Scott-domain D is a *strong subdomain* of a Scott-domain E iff there is an embedding-projection pair $e\colon D \leftrightarrows E\colon p$ and for all $x \in E$, $y \in D$ with $p(x) \leq y$ it follows that $\{x, e(y)\}$ is bounded.

Before showing that the snack powerdomain functor preserves the strong subdomain relation which is our new notion of a substructure for subtypes on the side of domains, we will turn to the localic side of the game.

Definition 3.3 Let A and B be domain prelocales. We say that A is a *sub-prelocale* of B if the following conditions are satisfied:

(i) A is a subalgebra of B with respect to $\vee, \wedge, 0$ and 1.

(ii) The preorder on A is the restriction of the preorder on B to A.

(iii) $\mathsf{T}(A)$ equals $A \cap \mathsf{T}(B)$.

(iv) $\mathsf{E}(A)$ equals $A \cap \mathsf{E}(B)$.

Definition 3.4 A sub-prelocale A of a Scott prelocale B is a *strong sub-prelocale* of B (denoted $A \trianglelefteq_s B$) if for all $a \in A$, $b \in B$, if $a \wedge b \approx 0_B$ then there exists $a' \in A$ with $a' \gtrsim b$ and $a \wedge a' \approx 0_B$.

While the definition of "sub-prelocale" simply ensures that the larger logic is in harmony with the smaller one in that it doesn't redefine the logical operations, "strongness" adds one further aspect, that of conservativity. It may be understood as saying that whenever there is a contradiction in the larger theory in which the smaller is involved, then there was a contradiction in the smaller logic already.

A technical lemma precedes the proof of the equivalence of strong subdomains and strong sub-prelocales.

Lemma 3.5 *Let $e\colon D \leftrightarrows E\colon p$ be an embedding-projection pair between algebraic domains. For any compact upper subset U of E with generators in $e(D)$ and any compact open $V \subseteq E$ that contains U there is a compact open set W with generators in $e(D)$ and $U \subseteq W \subseteq V$.*

Proof. Every generator $e(x)$ of U is above some (compact) generator k of V. In the domain D we have $x = \bigvee^{\uparrow}(\downarrow x \cap K_D)$, hence $\bigvee^{\uparrow} e(\downarrow x \cap K_D) = e(\bigvee^{\uparrow}(\downarrow x \cap K_D)) = e(x) \geq k$ in E. As k is compact, there must be some $l \in \downarrow x \cap K_D$ with $e(l) \geq k$. Embeddings preserve compactness, so $e(l)$ is compact.

If we replace every generator $e(x)$ of U by an element $e(l)$ found in this way, we get the generators of an upper set W which is compact, open and between U and V. $\square$

Lemma 3.6 *If D is a strong subdomain of E via $e\colon D \leftrightarrows E : p$ then $\mathsf{K}\Omega D$ is a strong sub-prelocale of $\mathsf{K}\Omega E$.*

Proof. Strictly speaking, $\mathsf{K}\Omega D$ is not a sub-prelocale of $\mathsf{K}\Omega E$ since the compact open subsets of D are not subsets of E. But embedding the generators of a compact open subset U of D into E and taking the upper closure of the set of these elements yields a compact open subset of E which in fact is $p^{-1}(U)$. We can therefore think of $\mathsf{K}\Omega D$ as a sub-prelocale of $\mathsf{K}\Omega E$ and suppress the map p^{-1}. Similarly, we assume that $e\colon D \to E$ is just an inclusion and suppress it as well. The map p is then an idempotent on E with image D.

So let $U \in \mathsf{K}\Omega D$ and $V \in \mathsf{K}\Omega E$ be such that $U \cap V = \emptyset$. We have to find a set $U' \in \mathsf{K}\Omega D$ such that $U' \supseteq V$ and $U \cap U' = \emptyset$.

Let $\mathcal{F}$ be the set of all compact open supersets of $\uparrow p(V)$ and $\mathcal{F}' := \{F \cap U \mid F \in \mathcal{F}\}$. This is a filtered set of compact open sets. The Hofmann-Mislove Theorem [8,12] tells us that $\bigcap \mathcal{F}' \neq \emptyset$ if each $F \cap U$ is nonempty.

If that is so, we have $\uparrow p(V) \cap U = \bigcap \mathcal{F} \cap U = \bigcap \mathcal{F}' \neq \emptyset$. Then there are generators x of U and y of V such that $\{x, p(y)\}$ is bounded. With $x \in D$ and $p(y) \in D$ we have also $x \vee p(y) \in D$ since D is closed under existing joins. Consequently, there is some $z \in D$ with $z = x \vee p(y)$, and we have $p(y) \leq z$. Since D is a strong subdomain of E it follows that y and $z = x \vee p(y)$ are bounded. As $p(y) \leq y$, this implies that $\{x, y\}$ is bounded, i. e. $U \cap V \neq \emptyset$. This contradicts our choice of U and V.

So there must be a compact open superset W of $\uparrow p(V)$ such that $W \cap U = \emptyset$. This set W need not be in $\mathsf{K}\Omega D$. However, Lemma 3.5 guarantees the existence of a set $U' \in \mathsf{K}\Omega D$ with $W \subseteq U' \subseteq \uparrow p(V)$. This completes the proof. □

From [9] we recall:

Lemma 3.7 *Let I be an ideal in a distributive lattice, and F a filter which is maximal amongst those disjoint from I. Then F is prime.* □

Lemma 3.8 *If A is a strong sub-prelocale of B then $\mathsf{spec}A$ is a strong subdomain of $\mathsf{spec}B$.*

Proof. The projection-embedding pair is given by $p\colon \mathsf{spec}B \to \mathsf{spec}A$, $p(F) = F \cap A$, and $e\colon \mathsf{spec}A \to \mathsf{spec}B$, $e(G) = \uparrow_B G$, where F and G are prime filters of B and A, respectively (see [2], Proposition 7.3.6). To prove strongness, assume $p(F) \leq G$, i. e. $F \cap A \subseteq G$. We have to show that F and $\uparrow_B G$ are bounded, that is that there is a prime filter containing both of these. Clearly, $F \wedge \uparrow_B G := \{b \wedge a \mid b \in F, a \in \uparrow_B G\}$ is a filter, and it does not contain 0_B. If it did, we had some $b \in F$ and $a \in G$ with $a \wedge b \approx 0$. As A is a strong sub-prelocale of B we then could find a $b_a \gtrsim b$ in A with $b_a \wedge a \approx 0$. But $b_a \in F \cap A \subseteq G$ and therefore $0 \approx b_a \wedge a \in G$. However, G is a prime filter and therefore cannot contain 0.

We now consider the set of filters in B which contain $F \wedge \uparrow_B G$ but not 0. By Zorn's Lemma there is a maximal filter with this property, and by the prime filter theorem (Lemma 3.7) this filter is prime. □

In the remainder of this section we will show that the strong sub-prelocale relation is preserved by the snack powerconstruction and so is, as promised, the strong subdomain relation. The proof makes use of the following observation.

Lemma 3.9 *Let A be a Scott-prelocale. A conjunction $\Box a \wedge \bigwedge_{i=1}^{n} \Diamond b_i$ equals $0_{\mathsf{P}^S(A)}$ if and only if there is an $i_0 \in \{1,\ldots,n\}$ such that $a \wedge b_{i_0} \approx 0_A$.*

Proof. The "if" is immediate. Assume that for some $i_0 \in \{1,\ldots,n\}$ we have $a \wedge b_{i_0} \approx 0_A$. Then

$$\Box a \wedge \bigwedge_{i=1}^{n} \Diamond b_i \lesssim \Box a \wedge \Diamond b_{i_0} \lesssim \Diamond(a \wedge b_{i_0}) \approx \Diamond 0 \approx 0.$$

For the "only if" assume $\Box a \wedge \bigwedge_{i=1}^{n} \Diamond b_i \approx 0$. We transport the situation to $\mathsf{K}\Omega\mathsf{P}^S(\mathsf{spec}A)$ via the interpretation function $[\![\cdot]\!]$:

$$[\![\Box a \wedge \bigwedge_{i=1}^{n} \Diamond b_i]\!] = [\![\Box a]\!] \cap \bigcap_{i=1}^{n} [\![\Diamond b_i]\!] = \emptyset.$$

The interpretation of $\Box a$ is $\uparrow_{\mathsf{P}^S(\mathsf{spec}A)}\downarrow([\![a]\!]_A, \emptyset)$, the interpretation of $\Diamond b_i$ is $\uparrow_{\mathsf{P}^S(\mathsf{spec}A)}\downarrow(\mathsf{spec}A, \{[\![b_i]\!]_A\})$ $(i = 1,\ldots,n)$. The intersection of these sets is $\uparrow_{\mathsf{P}^S(\mathsf{spec}A)}\downarrow([\![a]\!]_A, \{[\![a]\!]_A \cap [\![b_i]\!]_A \mid i = 1,\ldots,n\})$ if the snack in this formula is correctly built. This is the case if and only if $[\![a]\!]_A \cap [\![b_i]\!]_A \neq \emptyset$ for all $i \in \{1,\ldots,n\}$. In that case, the intersection is nonempty. Otherwise, the generating snacks have no upper bound such that the intersection is empty. Then $[\![a]\!]_A \cap [\![b_{i_0}]\!]_A = \emptyset$ for some $i_0 \in \{1,\ldots,n\}$. This means $[\![a \wedge b_{i_0}]\!]_A = \emptyset$ and since $[\![\cdot]\!]_A$ is order-reflecting we can conclude $a \wedge b_{i_0} \approx 0$. □

Theorem 3.10 *If A is a strong sub-prelocale of the Scott-prelocale B then the same holds between $\mathsf{P}^S(A)$ and $\mathsf{P}^S(B)$.*

Proof. That the snack construction is monotone w.r.t. the sub-prelocale relation was shown in [2] in a general fashion. We have to prove strongness.

Let $a \in \mathsf{P}^S(A)$, $b \in \mathsf{P}^S(B)$ such that $a \wedge b \approx 0$. Both, a and b can be written as a disjunction of conjunctions a_i $(i = 1,\ldots,n)$ and b_j $(j = 1,\ldots,m)$ of generators of the respective powerlocales. Using distributivity, $a \wedge b$ is seen to be a disjunction of conjunctions $a_i \wedge b_j$ $(i = 1,\ldots,n; j = 1,\ldots,m)$ each of which must be 0. For each (i,j) we will find an element $a_{i,j} \gtrsim b_j$, $a_{i,j} \in \mathsf{P}^S(A)$, such that $a_i \wedge a_{i,j} \approx 0$.

Suppose we had this. Then

$$\begin{aligned} 0 &\approx (a_1 \wedge a_{1,1}) \vee (a_1 \wedge a_{1,2}) \vee \ldots \vee (a_n \wedge a_{n,m}) \\ &\approx (a_1 \wedge (a_{1,1} \vee \ldots a_{1,m})) \vee \ldots \vee (a_n \wedge (a_{n,1} \vee \ldots a_{n,m})) \\ &\gtrsim (a_1 \vee \ldots \vee a_n) \wedge \bigwedge_{i=1}^{n} (a_{i,1} \vee \ldots \vee a_{i,m}) \\ &\gtrsim a \wedge b \approx 0 \end{aligned}$$

where the last inequality holds because $a_{i,1} \vee \ldots \vee a_{i,m} \gtrsim b_1 \vee \ldots \vee b_m \approx b$ for all $i \in \{1,\ldots,n\}$. Hence $b \lesssim \bigwedge_{i=1}^{n}(a_{i,1} \vee \ldots \vee a_{i,m}) \in \mathsf{P}^S(A)$ and $\bigwedge_{i=1}^{n}(a_{i,1} \vee \ldots \vee a_{i,m})$ is the element we had to find in order to prove strongness.

It remains to find the elements $a_{i,j}$. We stated that each $a_i \wedge b_j$ must be equivalent to 0. Because of $(\Box - \wedge)$ we can assume a_i and b_j to be the conjunctions of one $\Box$-generator and finitely many $\Diamond$-generators of $\mathsf{P}^S(A)$ and $\mathsf{P}^S(B)$, respectively. Let us say, the respective $\Box$-generators are $\Box\hat{a}$ and $\Box\hat{b}$. Then, using Lemma 3.9, we know that among the $\Diamond$-generators of a_i or b_j there is one $\Diamond\hat{c}$ such that $\hat{a} \wedge \hat{b} \wedge \hat{c} \approx 0$. Now we use that A is a strong sub-prelocale of B. If $\Diamond\hat{c}$ is a subformula of a_i then this gives us an element $\bar{a} \gtrsim \hat{b}$ in $\mathsf{P}^S(A)$ with $\hat{a} \wedge \hat{c} \wedge \bar{a} \approx 0$. Again with Lemma 3.9 it follows that $\Box\hat{a} \wedge \Diamond\hat{c} \wedge \Box\bar{a} \approx 0$. As $\Box\hat{a} \wedge \Diamond\hat{c} \gtrsim a_i$ and $\Box\bar{a} \gtrsim b_j$ we can set $a_{i,j} := \Box\bar{a}$. If $\Diamond\hat{c}$ is a subformula of b_j then we find an element $\bar{a} \gtrsim b \wedge c$ in $\mathsf{P}^S(A)$ with $\hat{a} \wedge \bar{a} \approx 0$. It follows that $\Box\hat{a} \wedge \Diamond\bar{a} \approx 0$. As $\Box\hat{a} \gtrsim a_i$ and $\Diamond\bar{a} \gtrsim \Diamond(\hat{b} \wedge \hat{c}) \gtrsim \Box\hat{b} \wedge \Diamond\hat{c} \gtrsim b_j$ we can in this case set $a_{i,j} := \Diamond\bar{a}$. So the elements $a_{i,j}$ are found which completes the proof. $\Box$

Corollary 3.11 *If D is a strong subdomain of the Scott-domain E then the same holds between $\mathsf{P}^S(D)$ and $\mathsf{P}^S(E)$.*

Proof. If D is a strong subdomain of E then following Lemma 3.6 the prelocale $\mathsf{K}\Omega(D)$ is a strong sub-prelocale of $\mathsf{K}\Omega(E)$. Because of Theorem 3.10 we have that $\mathsf{P}^S(\mathsf{K}\Omega(D))$ is a strong sub-prelocale of $\mathsf{P}^S(\mathsf{K}\Omega(E))$. Using Lemma 3.8 we can conclude that $\mathsf{spec}(\mathsf{P}^S(\mathsf{K}\Omega D))$ is a strong subdomain of $\mathsf{spec}(\mathsf{P}^S(\mathsf{K}\Omega E))$. Now we employ Theorem 2.10 and get that $\mathsf{P}^S(\mathsf{spec}(\mathsf{K}\Omega D))$ is a strong subdomain of $\mathsf{P}^S(\mathsf{spec}(\mathsf{K}\Omega E))$. As $\mathsf{spec}(\mathsf{K}\Omega D)$ is isomorphic to D (and the same for E), we finally derive that $\mathsf{P}^S D$ is a strong subdomain of $\mathsf{P}^S E$. $\Box$

4 A Type System for Databases

In relational databases, the type of a relation is given by its scheme. The scheme is a set of attribute names each of which is associated with an unstructured set of possible values. A relation consists of tuples which are elements of the cartesian product of these sets of values. Thus, relational databases have only one type construction: That is the product of basic types to form the type of a tuple. Hereof one then takes sets to build relations, the first class citizens in relational databases. However, it is not possible to incorporate structured basic types or use other type constructors. Even the given construction of building sets of tuples cannot be applied in a nested fashion. To resolve these shortcomings, a number of proposals have been made such as using tree-like basic types [23] to capture null values or introducing nested relations [21]. We follow Buneman's idea of using Scott-domains [4] as the structured sets which the database objects are taken from. From domain theory, many type constructors are known that transform Scott-domains into Scott-domains. With the snack powerdomain we even have a construction which allows for a nesting of generalized relations. Starting with simple ground domains we can thus build domains for structured datatypes.

Our type expressions are built as follows:

$$\sigma ::= \mathbf{1} \mid \sigma \times \tau \mid \sigma \oplus \tau \mid \sigma_\perp \mid \mathsf{P}^S\sigma \mid t \mid \mathsf{rec}\, t.\sigma$$

These expressions are usually interpreted within the category of Scott domains as follows: $\mathbf{1}$ stands for the one-point domain; $\times$ is the cartesian product, $\oplus$ the coalesced sum of two domains; $()_\perp$ is the lift and P^S the snack powerdomain of a domain, t is a generic type variable necessary to formulate the recursive expression $\mathsf{rec}\ t.\sigma$, and the latter is interpreted as the initial solution of the domain equation $t = \sigma$.

The coalesced sum will primarily be used in order to form ground domains such as the flat booleans or integers. These can be defined as

$$\mathsf{bool} := (\mathbf{1})_\perp \oplus (\mathbf{1})_\perp \quad \text{and } \mathsf{int} := \mathsf{rec}\, t.(\mathbf{1})_\perp \oplus t.$$

In order to form record types, the *separated* sum will be more useful. We include it as a derived construction:

$$\sigma + \tau := (\sigma)_\perp \oplus (\tau)_\perp.$$

Database objects of type σ are elements of the domain which σ is interpreted by. If we add appropriate functions between domains we will derive a generalization of relational algebra. In the theory of relational databases, a logical query language, viz. that of relational calculus, stands besides relational algebra. This and a possible connection to program logics [5] suggests to investigate a logical interpretation of the types. We will use the logical interpretation in the category of prelocales which can be found in [1,2] and, for the snack powerlocale, in the present paper. We denote the two interpretations by $[\![\cdot]\!]_D$ and $[\![\cdot]\!]_L$. The basic relationship between the two interpretations is that of Stone duality and may be summarized in the following theorem.

Theorem 4.1 *If σ is a closed type expression then* $\mathsf{spec}([\![\sigma]\!]_L) \cong [\![\sigma]\!]_D$. $\square$

For the snack powerconstruction we have shown this in Section 2. For the other constructions we would have to carry out the same programme. However, some can be found explicitly in [1]. We therefore restrict ourselves to giving the respective prelocales by their generators and type-specific axioms and rules. Using the general theory from [1,2] the reader will easily see these constructions to be the Stone duals of the respective constructions on domains.

Construction	**Generators**
$A \times B$	$\{a_l \mid a \in A\} \cup \{b_r \mid b \in B\}$
Axioms, rules	
If $a \lesssim a'$ then $a_l \lesssim a'_l$	If $b \lesssim b'$ then $b_r \lesssim b'_r$
$(a \wedge a')_l \approx a_l \wedge a'_l$	$(b \wedge b')_r \approx b_r \wedge b'_r$
$(a \vee a')_l \approx a_l \vee a'_l$	$(b \vee b')_r \approx b_r \vee b'_r$
$0_l \approx 0_r \approx 0$	$1_l \approx 1_r \approx 1$
If $\mathsf{T}(a)$ then $\mathsf{T}(a_l)$	If $\mathsf{T}(b)$ then $\mathsf{T}(b_r)$
If $\mathsf{E}(a)$ then $\mathsf{E}(a_l)$	If $\mathsf{E}(b)$ then $\mathsf{E}(b_r)$

Construction	**Generators**
$A \oplus B$	$\{a \oplus f \mid a \in A\} \cup \{f \oplus b \mid b \in B\}$
Axioms, rules	
If $a \lesssim a'$ then $a \oplus f \lesssim a' \oplus f$	If $b \lesssim b'$ then $f \oplus b \lesssim f \oplus b'$
$(a \wedge a') \oplus f \approx (a \oplus f) \wedge (a' \oplus f)$	$f \oplus (b \wedge b') \approx (f \oplus b) \wedge (f \oplus b')$
$(a \vee a') \oplus f \approx (a \oplus f) \vee (a' \oplus f)$	$f \oplus (b \vee b') \approx (f \oplus b) \vee (f \oplus b')$
$0 \oplus f \approx f \oplus 0 \approx 0$	$1 \oplus f \approx f \oplus 1 \approx 1$
If $\mathsf{T}(a)$ then $\mathsf{T}(a \oplus f)$	If $\mathsf{T}(b)$ then $\mathsf{T}(f \oplus b)$
If $\mathsf{T}(a)$ and $\mathsf{T}(b)$ then $(a \oplus f) \wedge (f \oplus b) \approx 0$	
If $\mathsf{E}(a)$ then $\mathsf{E}(a \oplus f)$	If $\mathsf{E}(b)$ then $\mathsf{E}(f \oplus b)$

Construction	**Generators**
$A_\perp$	$\{a_\perp \mid a \in A\}$
If $a \lesssim a'$ then $a_\perp \lesssim a'_\perp$	
$(a \wedge a')_\perp \approx a_\perp \wedge a'_\perp$	$(a \vee a')_\perp \approx a_\perp \vee a'\perp$
$0_\perp \approx 0$	
$\mathsf{T}(a_\perp)$	$\mathsf{E}(1)$
If $\mathsf{E}(a)$ then $\mathsf{E}(a_\perp)$	

There are no rules for recursive expressions as the prelocale for a recursive type is just the union of the prelocales for the approximants of that type. In the subtype relation defined below we will not make use of the coalesced sum. Instead the separated sum is taken. Though this is a derived construction we will make its building rules explicit here. With the abbreviations

$$a + f = a_\perp \oplus f \text{ and } f + b = f \oplus b_\perp$$

we get the rules in the following table.

In databases, we want to manipulate the stored data. For instance, we want to look at a part of each "tuple" only. That is done by projection. Or we want to combine two relations depending on the entries in specified parts of the tuples which can be done by natural join. Both operations take subschemes of the schemes of the relations involved as arguments. With more general data types, we will have to replace the notion of a subscheme by that of a subtype. Though the generalized operations are not subject of this paper we can briefly indicate how they will use subtypes. In [4] it was observed that the domain theoretic concept of a "projection" can be employed for database needs. With σ being a subtype of τ we will have a projection p_σ that takes objects of type τ, i. e. elements of $[\![\tau]\!]_D$, and projects them onto the domain associated with σ. Natural join will use the largest common subtype of the two relations which are joined, and with giving a subtype explicitly, one may define equi- or θ-joins. Finally we will also have nest- and unnest-operations to switch between different levels of nestedness of the powerconstruction P^S. The nest operator will take a subtype which specifies the type of the nests that are produced.

Construction	**Generators**
$A+B$	$\{a+f \mid a \in A\} \cup \{f+b \mid b \in B\}$
Axioms, rules	
If $a \lesssim a'$ then $a+f \lesssim a'+f$	If $b \lesssim b'$ then $f+b \lesssim f+b'$
$(a \wedge a')+f \approx (a+f) \wedge (a'+f)$	$f+(b \wedge b') \approx (f+b) \wedge (f+b')$
$(a \vee a')+f \approx (a+f) \vee (a'+f)$	$f+(b \vee b') \approx (f+b) \vee (f+b')$
$0+f \approx f+0 \approx 0$	
$\mathsf{T}(a+f)$	$\mathsf{T}(f+b)$
$(a+f) \wedge (f+b) \approx 0$	
If $\mathsf{E}(a)$ then $\mathsf{E}(a+f)$	If $\mathsf{E}(b)$ then $\mathsf{E}(f+b)$

On the syntactic level, the subtype relation $\leq_s$ is defined inductively:

Axioms: $\mathbf{1} \leq_{\mathsf{s}} \sigma$

$\sigma \leq_{\mathsf{s}} \sigma$

$\sigma \leq_{\mathsf{s}} \sigma \times \tau$

$\tau \leq_{\mathsf{s}} \sigma \times \tau$

Rules: If $\sigma \leq_{\mathsf{s}} \tau$ and $\tau \leq_{\mathsf{s}} \upsilon$ then $\sigma \leq_{\mathsf{s}} \upsilon$.

If $\sigma \leq_{\mathsf{s}} \tau$ then $\sigma_\perp \leq_{\mathsf{s}} \tau_\perp$.

If $\sigma_1 \leq_{\mathsf{s}} \sigma_2$ and $\tau_1 \leq_{\mathsf{s}} \tau_2$ then $\sigma_1 \times \tau_1 \leq_{\mathsf{s}} \sigma_2 \times \tau_2$.

If $\sigma_1 \leq_{\mathsf{s}} \sigma_2$ and $\tau_1 \leq_{\mathsf{s}} \tau_2$ then $\sigma_1 + \tau_1 \leq_{\mathsf{s}} \sigma_2 + \tau_2$.

If $\sigma \leq_{\mathsf{s}} \tau$ then $\mathsf{P}^S\sigma \leq_{\mathsf{s}} \mathsf{P}^S\tau$.

If $\sigma \leq_{\mathsf{s}} \tau[t := \mathsf{rec}\, t.\tau]$ then $\sigma \leq_{\mathsf{s}} \mathsf{rec}\, t.\tau$.

Note that there is no rule for type expressions built with $\oplus$. These type expressions will therefore never have non-trivial subtypes. On the semantic side this corresponds to the fact that the strong subdomains of a coalesced sum are always trivial.

Generally we claim that that under the interpretations $[\![\cdot]\!]_D$ and $[\![\cdot]\!]_L$ the subtype relation is translated into the strong subdomain and the strong sublocale relation, respectively. For the proof we need the following lemma which allows us to extend the defining property of strong sublocales (Definition 3.4) from elements to joins of elements.

Lemma 4.2 *Let A be a sub-prelocale of B and $a \approx a_1 \vee \ldots \vee a_m \in A$, $b \approx b_1 \vee \ldots \vee b_n \in B$ be such that $a \wedge b \approx 0$. If for all $1 \leq i \leq m$, $1 \leq j \leq n$, there is a $b^{(i,j)} \in A$ with $b^{(i,j)} \gtrsim b_j$ and $a_i \wedge b^{(i,j)} \approx 0$ then there is a $b^a \in A$ with $b^a \gtrsim b$ and $a \wedge b^a \approx 0$.*

Proof. We set $b^a := \bigwedge_{k=1}^m \bigvee_{j=1}^n b^{(k,j)}$. Obviously, $b^a \in A$. Moreover, $\bigvee_{j=1}^n b^{(k,j)} \gtrsim b_1 \vee \ldots \vee b_n \approx b$ for all $k = 1, \ldots, m$, therefore $b^a \gtrsim b$. Finally

$$\begin{aligned}
a \wedge b^a &= (\bigvee_{i=1}^m a_i) \wedge (\bigwedge_{k=1}^m \bigvee_{j=1}^n b^{(k,j)}) \\
&\approx \bigvee_{i=1}^m (a_i \wedge \bigwedge_{k=1}^m \bigvee_{j=1}^n b^{(k,j)}) \\
&\lesssim \bigvee_{i=1}^m (a_i \wedge \bigvee_{j=1}^n b^{(i,j)}) \\
&\approx \bigvee_{i=1}^m \bigvee_{j=1}^n (a_i \wedge b^{(i,j)}) \\
&\approx 0,
\end{aligned}$$

hence $a \wedge b^a \approx 0$. □

Theorem 4.3 *If $\sigma \leq_s \tau$ then $[\![\sigma]\!]_L$ is a strong sub-prelocale of $[\![\tau]\!]_L$.*

Proof. The proof is by induction. The base cases $\mathbf{1} \leq_s \sigma$ and $\sigma \leq_s \sigma$ are trivial. From the two symmetric cases $\sigma \leq_s \sigma \times \tau$ and $\tau \leq_s \sigma \times \tau$ we do the first.

Clearly, $[\![\sigma]\!]_L$ is a sub-prelocale of $[\![\sigma \times \tau]\!]_L$. We have to prove strongness. For this, assume that $a \in [\![\sigma]\!]_L$, $x \in [\![\sigma \times \tau]\!]_L$ and $a_l \wedge x \approx 0$. We have to find some $x^a \in [\![\sigma]\!]_L$ with $x^a_l \gtrsim x$ and $a \wedge x^a \approx 0$. Each $x \in [\![\sigma \times \tau]\!]_L$ can be written as a disjunction of conjunctions of the special form $s_l \wedge t_r$ with $s \in [\![\sigma]\!]_L$ and $t \in [\![\tau]\!]_L$. Hence

$$a_l \wedge x \approx a_l \wedge \bigvee_{i=1}^{n} (s^i_l \wedge t^i_r) \approx 0.$$

By Lemma 4.2, we need to find some $x^i \in [\![\sigma]\!]_L$ for each $i \in \{1, \ldots, n\}$ with $x^i \gtrsim s^i_l \wedge t^i_r$ and $a_l \wedge x^i_l \approx 0$. For each $i \in \{1, \ldots, n\}$ we have $(a \wedge s^i)_l \wedge t^i_r \approx 0$ which can only be if $a \wedge s^i \approx 0$ or $t^i \approx 0$. In the latter case $x^i := 0$, in the former case $x^i := s^i$ are possible choices for x^i.

Now we turn to the inductive steps where we omit the trivial case of the transitivity rule. In all cases it is clear that the respective prelocales satisfy the required sub-prelocale relation. We therefore only have to prove strongness.

We begin with $\sigma \leq_s \tau \Rightarrow \sigma_\perp \leq_s \tau_\perp$. Let $a \in [\![\sigma_\perp]\!]_L$, $b \in [\![\tau_\perp]\!]_L$ such that $a \wedge b \approx 0$. We have to find $b^a \gtrsim a$, $b^a \in A$, such that $a \wedge b^a \approx 0$. If $a \approx 1$ then $b \approx 0$ and we can set $b^a := b$. If $b \approx 1$ then $a \approx 0$ and again $b^a := b$. Otherwise there are $a' \in [\![\sigma]\!]_L$, $b' \in [\![\tau]\!]_L$ with $a = a'_\perp$ and $b = b'_\perp$. Then $a \wedge b = a'_\perp \wedge b'_\perp \approx (a' \wedge b')_\perp \approx 0 \approx 0_\perp$ from which $a' \wedge b' \approx 0$ follows. As $[\![\sigma]\!]_L$ is assumed to be a strong sub-prelocale of $[\![\tau]\!]_L$ we get a $(b')^{a'} \in [\![\sigma]\!]_L$ with $(b')^{a'} \gtrsim b'$ and $a' \wedge (b')^{a'} \approx 0$. Then $b^a := ((b')^{a'})_\perp$ is the required element of $[\![\sigma_\perp]\!]_L$.

Next for $\sigma_1 \times \tau_1 \leq_s \sigma_2 \times \tau_2$. We assume $[\![\sigma_1]\!]_L$ resp. $[\![\tau_1]\!]_L$ to be a strong sub-prelocale of $[\![\sigma_2]\!]_L$ resp. $[\![\tau_2]\!]_L$. Let $a \in [\![\sigma_1 \times \tau_1]\!]_L$, $b \in [\![\sigma_2 \times \tau_2]\!]_L$ and $a \wedge b \approx 0$. Both, a and b can be written as disjunctions of conjunctions of generators in the following form:

$$a \approx \bigvee_{i=1}^{m} (s^i_l \wedge t^i_r), \qquad s^i \in [\![\sigma_1]\!]_L, t^i \in [\![\tau_1]\!]_L, i = 1, \ldots, m$$

$$b \approx \bigvee_{j=1}^{n} (u^j_l \wedge v^j_r), \qquad u^j \in [\![\sigma_2]\!]_L, v^j \in [\![\tau_2]\!]_L, j = 1, \ldots, n$$

By Lemma 4.2 it suffices to find elements $b^{(i,j)} \gtrsim u^j_l \wedge v^j_r$ in $[\![\sigma_1 \times \tau_1]\!]_L$ with $s^i_l \wedge t^i_r \wedge b^{(i,j)} \approx 0$ $(i = 1, \ldots, m;\ j = 1, \ldots, n)$. For these i and j we have

$$(s^i_l \wedge u^j_l) \wedge (t^i_r \wedge v^j_r) \approx 0.$$

As $[\![\sigma_1]\!]_L \trianglelefteq_s [\![\sigma_2]\!]_L$ and $[\![\tau_1]\!]_L \trianglelefteq_s [\![\tau_2]\!]_L$ we find some $(u^j)^{s^i} \gtrsim u^j$ in $[\![\sigma_1]\!]_L$ with $s^j \wedge (u^j)^{s^i} \approx 0$ in the first case and some $(v^j)^{t^i} \gtrsim v^j$ in $[\![\tau_1]\!]_L$ with $t^i \wedge (v^j)^{t^i} \approx 0$ in the second case. In the first case $b^{(i,j)} := (u^j)^{s^i}$, in the second case $b^{(i,j)} := (v^j)^{t^i}$ is the required element.

Now for $\sigma_1 + \tau_1 \leq_s \sigma_2 + \tau_2$. Again we assume $[\![\sigma_1]\!]_L \trianglelefteq_s [\![\sigma_2]\!]_L$ and $[\![\tau_1]\!]_L \trianglelefteq_s [\![\tau_2]\!]_L$. Given $a \in [\![\sigma_1 + \tau_1]\!]_L$, $b \in [\![\sigma_2 + \tau_2]\!]_L$ with $a \wedge b \approx 0$, we first consider the cases where $a \approx 1$ or $b \approx 1$. Then $b \approx 0$ resp. $a \approx 0$, and in either case $b \in [\![\sigma_1 + \tau_1]\!]_L$ so that nothing is to be done. Otherwise, both, a and b, can be written as disjunctions of conjunction of the form $s + f \wedge f + t$ and the proof can be completed in the same fashion as for the product.

The case of the snack powerconstruction was done in Theorem 3.10

Finally, the rule for the recursive types is trivial since $\tau[t := \mathsf{rec}\ t.\tau]$ and $\mathsf{rec}\ t.\tau$ are interpreted by the same prelocale. □

Because of Lemma 3.8 and Theorem 4.1 the domain theoretic version of Theorem 4.3 is also true:

Theorem 4.4 *If $\sigma \leq_{\mathrm{s}} \tau$ then $[\![\sigma]\!]_D$ is a strong subdomain of $[\![\tau]\!]_D$.* □

5 Conclusions

From the technical perspective we have seen how Abramsky's *Domain Theory in Logical Form* can usefully be employed in the development of domain theoretic constructions. Despite the underlying duality theory being one of the most advanced topics in Lattice Theory and Domain Theory, we get out of it logical descriptions which are simple and intuitive. This has worked both for the description of the snack powerconstruction itself and the subdomain relation.

As far as the overall project of providing a denotational semantics for relational databases is concerned, much remains to be done. Specifically, we need to demonstrate how the usual database operations can be interpreted in the model. This has been done successfully for the nest and unnest (cf. [21]) operators which are naturally associated with higher order relations, [20]. More operators remain to be investigated.

Finally, we would like to emphasize that the subdomain relation, which we have studied in the last section, is only a *qualitative* description of subtyping. It will be important in further developments to refine this into a more *computational* description. To this end we will replace the relation by an adjoint pair of explicit *coercion functions*, each of which can be deduced from the specific derivation showing that the subtype relation holds.

Acknowledgement

We are grateful to Reinhold Heckmann and an anonymous referee for pointing out misprints and suggesting improvements. We also thank the "Domain Theory Group" at Darmstadt for many stimulating discussions on the subject matter.

References

[1] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[2] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3. Clarendon Press, 1995.

[3] V. Breazu-Tannen, P. Buneman, and L. Wong. Naturally embedded query languages. In J. Biskup and R. Hull, editors, *Database Theory - ICDT'92*, volume 646 of *Lecture Notes in Computer Science*, pages 140–154. Springer Verlag, 1992.

[4] P. Buneman, A. Jung, and A. Ohori. Using powerdomains to generalize relational databases. *Theoretical Computer Science*, 91:23–55, 1991.

[5] P. Cousot. 15: Methods and Logics for Proving Programs. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 841–993. Elsevier Science Publishers, 1990.

[6] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices.* Springer Verlag, 1980.

[7] C. Gunter. The mixed power domain. *Theoretical Computer Science*, 103:311–334, 1992.

[8] K. H. Hofmann and M. Mislove. Local compactness and continuous lattices. In B. Banaschewski and R.-E. Hoffmann, editors, *Continuous Lattices, Proceedings Bremen 1979*, volume 871 of *Lecture Notes in Mathematics*, pages 209–248. Springer Verlag, 1981.

[9] P. T. Johnstone. *Stone Spaces*, volume 3 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, Cambridge, 1982.

[10] A. Jung, L. Libkin, and H. Puhlmann. Decomposition of domains. Technical Report 1416, Technische Hochschule Darmstadt, 1991.

[11] P. Kanellakis. Elements of relational database theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 17, pages 1073–1156. Elsevier Science Publishers B.V., 1990.

[12] K. Keimel and J. Paseka. A direct proof of the Hofmann-Mislove theorem. *Proceedings of the AMS*, 120:301–303, 1994.

[13] L. Libkin. A relational algebra for complex objects based on partial information. In J. Demetrovics and B. Thalheim, editors, *Mathematical Fundamentals of Database Systems–91*, volume 495 of *Lecture Notes in Computer Science*, pages 36–41. Springer-Verlag, 1991.

[14] L. Libkin. *Aspects of Partial Information in Databases.* PhD thesis, University of Pennsylvania, 1994.

[15] L. Libkin. Approximation in databases. In G. Gottlob and M. Vardi, editors, *International Conference on Database Theory*, volume 893 of *Lecture Notes in Computer Science*, pages 411–424. Springer Verlag, 1995.

[16] D. Maier. *The Theory of Relational Databases.* Computer Science Press, 1983.

[17] Teow-Hin Ngair. *Convex Spaces as an Order-theoretic Basis for Problem Solving.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, July 1992.

[18] H. Puhlmann. A Semantics for Generalized Database Relations. Master's thesis, Imperial College of Science, Technology and Medicine, London, 1991.

[19] H. Puhlmann. The snack powerdomain for database semantics. In A. M. Borzyszkowski and S. Sokołowski, editors, *Mathematical Foundations of Computer Science*, volume 711 of *Lecture Notes in Computer Science*, pages 650–659, 1993.

[20] H. Puhlmann. How to nest and unnest relations without loosing information. Notes., 1994.

[21] H.-J. Schek and M. Scholl. The relational model with relation-valued attributes. *Information Systems*, 11:137–147, 1986.

[22] S. J. Vickers. *Topology Via Logic.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[23] C. Zaniolo. Database relation with null values. *Journal of Computer and System Sciences*, 28:142–166, 1984.