Decomposition of Domains*

Achim Jung Leonid Libkin Hermann Puhlmann †

Abstract

The problem of decomposing domains into sensible factors is addressed and solved for the case of dI-domains. A decomposition theorem is proved which allows the represention of a large subclass of dI-domains in a product of flat domains. Direct product decompositions of Scott-domains are studied separately.

1 Introduction

This work was initiated by Peter Buneman's interest in generalizing relational databases, see [6]. He — quite radically — dismissed the idea that a database should be forced into the format of an n-ary relation. Instead he allowed it to be an arbitrary anti-chain in a Scott-domain. The reason for this was that advanced concepts in database theory, such as 'null values', 'nested relations', and 'complex objects' force one to augment relations and values with a notion of information order. Following Buneman's general approach, the question arises how to define basic database theoretic concepts such as 'functional dependency' for anti-chains in Scott-domains. For this one needs a way to speak about 'relational schemes' which are nothing but factors of the product of which the relation is a subset. Buneman successfully defined a notion of 'scheme' for Scott-domains and it is that definition which at the heart of this work. We show that his generalized 'schemes' behave almost like factors of a product decomposition. (Consequently, we choose the word *semi-factor* for them.) In the light of our results, Peter Buneman's theory of generalized databases becomes less miraculous: a large class of domains can be understood as sets of tuples.

Buneman's definition of scheme was discussed in [17] and an alternative definition was proposed. The idea of both definitions is that the elements of a domain are treated as objects, and projecting an element into a scheme corresponds to losing some information about this object. The definition of [17] is based on the assumption that the same piece of information is lost for every object. For example, if objects are records, it means that we lose information about some attributes' values. The idea of [6] is that every scheme has a sort of complement, and if we project one object to a scheme and the other to its

*Appeared in: *Proceedings of the Conference on Mathematical Foundations of Programming Semantics 91*, Springer Lecture Notes in Computer Science, **598** (1992), 235-258.

†Addresses: Achim Jung and Hermann Puhlmann: Fachbereich Mathematik, Technische Hochschule Darmstadt, Schloßgartenstraße 7, D-6100 Darmstadt, Germany. Leonid Libkin: Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia, PA 19104, U.S.A. L.Libkin was supported in part by NSF Grants IRI-86-10617 and CCR-90-57570 and ONR Grant NOOO14-88-K0634.

complement, then there exists a join of two projections, i.e. every object consists of two independent "pieces of information". Intuitively it means that the domain itself could be decomposed into two corresponding domains.

The definition of [6] is stronger than the definition of [17]. It is the first definition that is used in our decomposition theory while the second definition serves as a tool to describe direct product decompositions of domains. Combining the decomposition theorems, we will prove a formal statement that clarifies the informal reasonings from the previous paragraph.

There is also a more philosophical or pedagogical motivation for this work. A feature that novices to domain theory frequently find unsettling is the profusion of different definitions it offers. Often these definitions are laid out at the beginning and the relation to the semantics of programming languages is established only later. In particular, useful closure properties of the respective categories are derived. In his 'Pisa Lecture Notes' [18], Gordon Plotkin chose a rather more gentle approach. The 'domains' he considers are very primitive at the beginning, just sets, and step by step new constructs and properties are added to them: a bottom element transforms sets into flat domains, and thus the information order is introduced; next come slightly more complicated orders created by forming finite products of flat domains; function spaces call for the definition of dcpo and Scott-continuous function and, via bilimits and powerdomains, he finally arrives at bifinite domains. Furthermore, along the way he develops a syntax which allows to denote (most of) the elements of the domains, making them available for computation: the product appears as a set of arrays, the function space as a set of λ-terms, etc. (This aspect is also described elegantly and comprehensively in [1].) In this way, Plotkin creates the impression that all (bifinite) domains are built up from flat domains using various domain constructors. This may be reassuring for the novice but of course it is not explicitly confirmed in the text. Plotkin is just very carefully expanding his definitions and motivating each new concept. But we may still ask to what extent this first impression could be transformed into a theorem. To be more precise, we may ask: "Is it true, that every bifinite domain can be derived from flat domains using only lifting, product, coalesced sum, function space and convex powerdomain as constructors?" (A similar question was in fact asked — and found difficult — by Carl Gunter for the universal bifinite domain.)

How would one attack such a problem? We think the natural way to do it is to work backwards and to try to *decompose* domains into pieces that decompose no further. If we can show that the only irreducible domains are the flat domains then we are done.

At this point the informed reader may already have become nervous because he may know small finite counterexamples to the above question. But there are many variations of it which are equally interesting. We can restrict (or augment) the number of allowed constructions, we can change the class of domains we want to analyze, we can allow more (or fewer) primitive (i.e. irreducible) building blocks. The choice we have made for this paper is to consider Scott and dI-domains (cf. [4, 3]) and a single, albeit rather general, constructor, and instead of prescribing the irreducible factors, we are curious what they will turn out to be. The advantage of a decomposition theorem of this kind is apparent: instead of proving a property for general domains we can prove that it holds for the irreducible factors and that it is preserved under the constructions. We allow ourselves to compare this endeavor with the similar (and only recently completed) project of decomposing finite groups into finite simple groups, although the comparison is somewhat flattering: we cannot expect to find so much mathematically intriguing structure in domains.

What are the practical implications of our decomposition theorem? Well, in our particular setting we derive a very concrete representation of dI-domains as a set of 'tuples' which should simplify the implementation of dI-domains as abstract data types. Of course, there is a well-developed theory of effective representations (see [20, 15, 22, 16]), where one enumerates the set of compact elements and represents (a subset of) the infinite elements by recursively enumerable sets of compact approximations. However, this is more theoretical work and no one expects that we really ever use domains as data types represented this way. Instead, our representation is much more concrete. To give an example, consider a domain which is the product of two flat domains. The traditional effective domain theory simply enumerates all elements, and, if enumerations of the elements of the two factors are already given, then these are combined with the help of pairing functions. We work rather in the opposite direction. For a given domain we seek to decompose it as far as possible and we will only enumerate the bases of the (irreducible) factors in the traditional way. The representation of the original domain is then put together as a set of 'tuples'.

The paper is organized as follows. In the next section we shall quickly review some basic definitions from domain theory, mostly to fix notation and to remind the reader of a few less common concepts. In Section 3 we introduce semi-factors and prove basic properties of them. We apply these ideas and get a first decomposition theorem. This representation still contains a lot of redundancy and in Section 4 we show how to 'factor away' this redundancy. The resulting decomposition theorem yields a representation of dI-domains which is very tight. (These sections report work by the first and the third author.)

A direct product decomposition is a particular and interesting instance of our general goal and deserves more detailed study. In Sections 5 and 6 (which were written by the second author) this is done by establishing a relationship between these decompositions and particular instances of congruence relations and neutral ideals. The idea to describe direct product decompositions via neutral ideals is borrowed from lattice theory where neutral ideals describe decompositions of bounded lattices. For domains we will obtain a more general kind of decomposition including direct product and coalesced sum as limit cases. These decompositions are given by families of subsets of a domain such that every element of the domain has a unique representation as the join of suitably chosen representatives of these sets. Pairs of permutable complemented congruences also describe direct product decompositions as well as they describe decompositions of algebras. Having proved characterizations of decompositions, we establish the result showing the relationship between the two notions of scheme.

2 Definitions

We are using the standard definitions such as they can be found in [13] and in [1]. In particular, *dcpo's* are *directed-complete partial orders* and they have suprema for all directed sets. Most of the time they have a least element, which we denote by $\perp$. *Compact elements* in a domain are such that they cannot be below a supremum of a directed set without being below some element of that set already, and if there are enough compact elements such that every element is the supremum of a directed collection of them, we call the dcpo *algebraic*. More suprema than just those of directed sets can exist: if every bounded set has a join then we call the dcpo *bounded-complete*; if every set has a join then we have a *complete lattice*. In case a bounded-complete dcpo is also algebraic we call it a *Scott-domain*. The expression 'algebraic complete lattice' is shortened to *algebraic lattice*. We will mostly study distributive Scott-domains, for which it is sufficient to require the distributive law to hold in the principal ideals. (The standard textbook on distributive lattices is [2]). Even more restrictive is the definition of *dI-domains* (cf. [4, 3]): they are distributive Scott-domains in which every principal ideal generated by a compact element is finite. Because of this strong finiteness property we can usually derive theorems about dI-domains very quickly from the same theorems stated for finite distributive Scott-domains.

All our functions are *Scott-continuous*, which means they carry the supremum of a directed set to the supremum of the image of the set. We do not make much use of them in this generality but mostly consider *projections*, which are in addition idempotent and below the identity. Recall that projections always preserve existing infima and are completely determined by their image. Even the order between projections can be read off their image: it is simply inclusion. For more detailed information we refer to [8].

An element x in a lattice is *join– (meet–) irreducible* if from the equation $y \vee z = x$ ($y \wedge z = x$) we can deduce that x equals y or z. (In the presence of distributivity this is equivalent to the stronger property of *join– (meet–) primeness*, but we will not make much use of this.)

Domain theory also includes the concept of *ideal* which is a directed and downward closed subset. This is a generalization of 'ideal' as it is known in lattice theory, where these are sets which are downward closed and closed under finite suprema. We need a generalization which goes in a different direction:

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

The same concept is defined in [6] and in [7] where such subsets of Scott-domains are called *strong ideal* and *complete ideal*, respectively. We find either expression rather misleading as we are not dealing with a special kind of domain theoretic ideal but with a completely different concept. Instead we take the viewpoint that such subsets are special *substructures*, i.e. special subdomains. As it happens, they correspond one-to-one to images of projections p for which $y \leq x$ implies $p(y) = y \wedge p(x)$. (An even stronger property holds, see Proposition 2 (ii) below.) In domain theory such functions are known as *stable projections*, hence our terminology.

Factors of products of dcpo's with bottom have the property that there is always a canonical projection onto them. This is also true for stable subdomains in Scott-domains:

Lemma 1 *Let A be a stable subdomain of the Scott-domain D. Then $p_A\colon D \to D$, defined by*

$$p_A(x) = \bigvee(\downarrow x \cap A)$$

is a projection on D with image A.

Our first decomposition has the form of a general categorical *limit*. A concrete description is given in terms of certain elements of the product of the dcpo's involved.

Definition. Let $\mathcal{D}$ be a set of dcpo's and let $\mathcal{F}$ be a set of Scott-continuous functions between elements of $\mathcal{D}$ (in the language of category theory: a diagram in **DCPO**). Furthermore, let $\bar{x} = (x_D)_{D\in\mathcal{D}}$ be an element (a *tuple*) of the cartesian product of all elements of $\mathcal{D}$. We say that $\bar{x}$ is *commuting* if the equation $x_E = f(x_D)$ holds for all functions $f\colon D \to E$, $f \in \mathcal{F}$, and all elements D, E in $\mathcal{D}$. Similarly, it is called *hyper-commuting* if the inequality $x_E \geq f(x_D)$ holds.

The set of all commuting tuples forms the categorical limit of the diagram $(\mathcal{D}, \mathcal{F})$ and we denote it by $\mathsf{lim}_{\mathcal{F}}\mathcal{D}$. The set of hyper-commuting tuples we call the *hyper-limit* and we reserve the notation $\mathsf{hyperlim}_{\mathcal{F}}\mathcal{D}$ for it. The latter construction is a special case of a more general concept developed in the theory of 2-categories, namely, lax limits. It is easy to see that **DCPO** is closed under limits and this kind of lax limit. Whether any of the other properties generally associated with domains is preserved depends on the structure of the diagram. For more detailed information consult [21].

3 Stable subdomains, semi-factors, and the First Decomposition Theorem

We begin by recalling from [6] and [19] some of the properties of stable subdomains.

Proposition 2 *Let D be a Scott-domain. Then the following hold:*

(i) *$\{\bot_D\}$ and D are stable subdomains of D.*

(ii) *If x is an element of a stable subdomain A of D and if $p_A(y)$ is less than x then $p_A(y) = x \wedge y$.*

(iii) *If D is distributive then p_A preserves existing suprema.*

(iv) *The set Q_D of all stable subdomains of D ordered by inclusion is an algebraic lattice.*

(v) *If D is distributive then Q_D is distributive.*

(vi) *In Q_D, the finite meet of stable subdomains is given by their intersection and $p_{A\cap B} = p_A \circ p_B = p_B \circ p_A$.*

(vii) *If D is distributive then (arbitrary) suprema in Q_D can be calculated pointwise, and for $\mathcal{A} \subseteq Q_D, x \in D : p_{\bigvee\mathcal{A}}(x) = \bigvee_{A\in\mathcal{A}} p_A(x)$.*

(Proofs can be found in [6].)

The concept of 'stable subdomain' is still too general to serve as a definition of 'distinguished piece of a domain'. For example, every element x of a domain generates a stable subdomain $\downarrow x$, but in general such a principal ideal cannot be hoped to lead to a sensible decomposition. In [6] a more restrictive definition is introduced, that of a *scheme*, and it is motivated by the database applications we had in mind there. Here we can give a new motivation based on the desired decomposition result. Consider the following theorem:

Theorem 3 *Let D be a finite distributive Scott-domain and let $\mathcal{A}$ be a set of stable subdomains the supremum of which equals $D = \top_{Q_D}$. Let $\mathcal{F}$ be the set of projections $p_A|_B : B \to A$ where $A \subseteq B$ are two elements of $\mathcal{A}$. Furthermore, let $\hat{D}$ consist of those commuting tuples $\bar{x} = (x_A)_{A\in\mathcal{A}}$ for which the set $\{x_A \mid A \in \mathcal{A}\}$ is bounded in D. Then $\hat{D}$ is isomorphic to D with the isomorphisms*

$$\begin{aligned} \Psi\colon D \to \hat{D}, \Psi(x) &= (p_A(x))_{A\in\mathcal{A}} \\ \Phi\colon \hat{D} \to D, \Phi(\bar{x}) &= \bigvee\{x_A \mid A \in \mathcal{A}\}. \end{aligned}$$

The proof of this theorem is straightforward, one only has to bear in mind that suprema in Q_D are calculated pointwise. The theorem is unsatisfying, however, because in order to represent D through a set of stable subdomains, we need to include information that can only be gained by looking at D itself: the boundedness of the coordinates of $\bar{x}$. We shall now give a definition of a *semi-factor*, such that boundedness comes for free if only the tuple commutes.

Definition. A stable subdomain A of a Scott-domain D is called *semi-factor* if $p_A(x) \leq a$ implies that x and a are bounded, for all $x \in D$ and $a \in A$.

In [6] and in [19] it is shown that this definition works well in the test case of direct product decompositions: the semi-factors of a direct product $D \times E$ are in 1–1 correspondence with products of semi-factors of D and E. In particular, $D \times \{\perp_E\}$ and $\{\perp_D\} \times E$ are semi-factors in $D \times E$.

We collect the basic properties of semi-factors in a fashion similar to that for stable subdomains:

Proposition 4 *Let D be a distributive Scott-domain. Then the following hold:*

(i) *$\{\perp_D\}$ and D are semi-factors of D.*

(ii) *The set S_D of all semi-factors of D, ordered by inclusion, is a distributive, complete lattice.*

(iii) *If S and T are semi-factors of D, then so are $S \cap T$ and $S \vee T$, where again the join is taken pointwise. (The latter also holds for arbitrary joins.)*

(iv) *S_D is a sublattice of Q_D.*

(For the proofs see [6].)

The following lemma states that our definition yields the desired extension property:

Lemma 5 *Let $\mathcal{S}$ be a family of semi-factors of a finite distributive Scott-domain D and let $\mathcal{F}$ consist of all connecting projections as in Theorem 3 above. Let $\mathcal{S}$ be such that with $S, T \in \mathcal{S}$ we also have $S \cap T \in \mathcal{S}$. If $\bar{x} = (x_S)_{S\in\mathcal{S}}$ is a commuting tuple, then the set $\{x_S \mid S \in \mathcal{S}\}$ is bounded in D.*

Proof. We first show this for the case in which $\mathcal{S}$ consists of just three semi-factors, S, T and $S \cap T$. Let $\bar{x}$ be a commuting tuple in $S \times T \times S \cap T$.

$$\begin{array}{rcll} x_T & \geq & p_{S\cap T}(x_T) & (p_{S\cap T} \leq \mathsf{id}_D) \\ & = & x_{S\cap T} & (\bar{x} \text{ is commuting}) \\ & = & p_{S\cap T}(x_S) & (\text{ditto}) \\ & = & p_T \circ p_S(x_S) & (2-\text{vi} \ \& \ 4-\text{iii}) \\ & = & p_T(x_S) & (p_S\,|_T = \mathsf{id}_T) \end{array}$$

By the defining property of semi-factors, $\{x_S, x_T\}$ is bounded in D.

The general proof is by induction. Set $S = \bigvee_{i=1}^n S_i$ and $T = S_{n+1}$. By the induction hypothesis the join of $\{x_{S_1}, \dots, x_{S_n}\}$ exists and we may set $x_S = \bigvee_{i=1}^n x_{S_i}$. The tuple $(x_S, x_T, p_{S\cap T}(x_S))$ is commuting for the three semi-factors S, T and $S \cap T$, because projections preserve suprema by 2-(iii): $p_{S\cap T}(x_S) = p_T \circ p_S(x_S) = p_T(x_S) = p_T(\bigvee_{i=1}^n x_{S_i}) = \bigvee_{i=1}^n p_T(x_{S_i}) = \bigvee_{i=1}^n x_{S_i \cap T} = \bigvee_{i=1}^n p_{S_i}(x_T) = p_S(x_T) = p_S \circ p_T(x_T) = p_{S\cap T}(x_T)$. So we can apply the result for the three element case for the induction step. $\square$

In our decomposition theorem we want to use as few semi-factors as possible, which in turn should be as primitive as possible. As a first approximation we choose the set $J(S_D)$ of semi-factors which are join-irreducible in Q_D. This set has two properties which make it attractive: every semi-factor is a join of irreducibles (in the finite case, but it will generalize to dI-domains) and a join-irreducible cannot be reached by a join of strictly smaller semi-factors, so it is in a sense unavoidable. But in order to apply the previous lemma we need a set closed under finite intersections, and in general $J(Q_D)$ will not do us this favor. We need another preparatory lemma:

Lemma 6 *Let D be a finite distributive Scott-domain and let $J(S_D)$ be the set of join-irreducible semi-factors of D. Let $\bar{x} = (x_S)_{S\in J(S_D)}$ be a commuting tuple for $J(S_D)$ and the connecting projections $\mathcal{F}$. Let $\mathcal{F}'$ be the appropriately extended set of connecting projections for all of S_D. Then $\bar{x}$ can be extended uniquely to a commuting element $\bar{x}'$ for $S_D, \mathcal{F}'$.*

Proof. We first show that for two join-irreducible semi-factors U and V we have the following commutation rule: $p_U(x_V) = p_V(x_U)$. Indeed, if $U \cap V$ is the join of the join-irreducible semi-factors $U_1, \dots, U_n$ then we can calculate: $p_U(x_V) = p_U \circ p_V(x_V) = p_{U\cap V}(x_V) = \bigvee_{i=1}^n p_{U_i}(x_V) = \bigvee_{i=1}^n x_{U_i} = \bigvee_{i=1}^n p_{U_i}(x_U) = p_{U\cap V}(x_U) = p_V \circ p_U(x_U) = p_V(x_U)$.

We extend the tuple $\bar{x}$ to all of S_D by setting

$$x_S = \bigvee \{x_U \mid S \supseteq U \in J(S_D)\}.$$

We have to show that $\bar{x}' = (x_S)_{S\in S_D}$ is commuting, so let $S \subseteq T$ be two semi-factors of D. Then we have

$$\begin{array}{rcll} p_S(x_T) & = & p_S(\bigvee\{x_U \mid T \supseteq U \in J(S_D)\}) & (\text{by def.}) \\ & = & \bigvee\{p_S(x_U) \mid T \supseteq U \in J(S_D)\} & (2-\text{iii}) \\ & = & \bigvee\{p_V(x_U) \mid T \supseteq U \in J(S_D), S \supseteq V \in J(S_D)\} & (2-\text{vii}) \\ & = & \bigvee\{p_U(x_V) \mid T \supseteq U \in J(S_D), S \supseteq V \in J(S_D)\} & (\text{as shown before}) \end{array}$$

$$\begin{aligned}
&= \bigvee\{p_T(x_U) \mid S \supseteq V \in J(S_D)\} \quad (2-\text{vii})\\
&= \bigvee\{x_V \mid S \supseteq V \in J(S_D)\} \quad (V \supseteq S \supseteq T)\\
&= x_S \quad (\text{by def.}) \ \Box
\end{aligned}$$

We can now state

Theorem 7 (The First Decomposition Theorem) *Let $J(S_D)$ be the set of all join-irreducible semi-factors of the finite distributive Scott-domain D and let $\mathcal{F}$ be the set of connecting projections. Then D is isomorphic to the limit of $J(S_D)$ over $\mathcal{F}$. The isomorphisms are given by*

$$\begin{aligned}
\Psi\colon D &\to \lim_{\mathcal{F}} J(S_D)\\
x &\mapsto (p_S(x))_{S\in J(S_D)}
\end{aligned}$$

and

$$\begin{aligned}
\Phi\colon \lim_{\mathcal{F}} J(S_D) &\to D\\
(x_S)_{S\in J(S_D)} &\mapsto \bigvee_{S\in J(S_D)} x_S.
\end{aligned}$$

The proof of this theorem is contained completely in the previous lemma, where we showed how to extend a commuting tuple to all of S_D, in particular to $D \in S_D$ itself. $\Box$

We illustrate the First Decomposition Theorem for three finite domains.

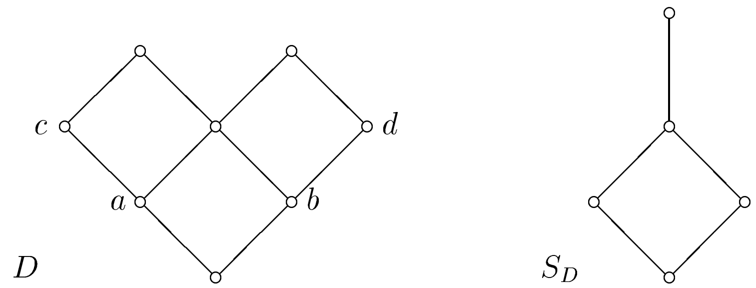
Example 1: $D = M_\perp$, M a finite set, i.e. D is a flat domain. We find that D possesses only the trivial semifactors $\{\perp\}$ and D, the latter being join-irreducible in S_D. Hence we conclude:

Observation 1: Flat domains are indecomposable.

Example 2: $D = 2 \times 2$, the four-element Boolean algebra. Since D is a lattice, it is isomorphic to its lattice of semi-factors. The join-irreducibles are $(\top,\perp)$ and $(\perp,\top)$ and the decomposition yields $D \simeq \downarrow(\top,\perp) \times \downarrow(\perp,\top)$, where $\downarrow(\top,\perp) \simeq \downarrow(\perp,\top) \simeq 2$. This is not a coincidence:

Observation 2: Direct product structure is recognized.

Example 3:



We find that D is join-irreducible in S_D and hence must be contained in any decomposition based on the First Decomposition Theorem. This is obviously not satisfactory and we shall derive a better decomposition theory below. Before doing so, let us study the situation for infinite domains. Here we have to deal with the following complication: the intersection of an infinite family of semi-factors is not necessarily a semi-factor again. We therefore do not know whether S_D is algebraic in general. We view this as the major open problem in our decomposition theory. In the case of dI-domains we are fine:

Proposition 8 *Let D be a dI-domain. Then S_D is algebraic and co-algebraic (i.e. S_D^{op} is algebraic).*

Proof. We only give an outline because we don't have the space to introduce the details of the theory of approximation via compact elements in domains in general and in our decomposition theory in particular.

One first observes that stable subdomains and semi-factors are completely determined by the set of compact elements they contain. Also, the canonical projection onto a stable subdomain can be seen as mapping each element onto the supremum of those compact elements of the subdomain which are below it: $p_A(x) = \bigvee {\downarrow} x \cap K(D) \cap A$. Furthermore, the canonical projection, as a Scott-continuous map, is completely determined by its behavior on compact elements. Since it is also sufficient to state the extension property of semi-factors for compact elements only, we have reduced the whole theory to $K(D)$, the set of compact elements in D. With this in mind, it is now easy to see that the arbitrary intersection of semi-factors is again a semi-factor: below a compact element in a dI-domain there are only finitely many elements at all and so for a particular compact element the intersection behaves as if it were over a finite index set.

Similarly, it is easy to see that the directed union of semi-factors yields a semi-factor again. Together this shows that the set S_D of semi-factors forms an inductive hull system on D, which implies algebraicity. A semi-factor is compact in S_D if and only if it is generated by a finite set of compact elements of D.

The co-compact elements are found as follows: suppose a semi-factor S does not contain a certain element x of D. By algebraicity of D it follows that there is a compact element c of D which S does not contain. Furthermore, because ${\downarrow} c$ is a finite distributive lattice, there is a join-irreducible k below c which again does not belong to S. On the other hand, if k is join-irreducible (hence: prime) in $K(D)$, then the join of all semi-factors which do not contain k, will again not contain this element. From this it follows along standard lines that any finite set of join-irreducible elements of $K(D)$ defines a co-compact semi-factor and that there are enough co-compact semi-factors to generate the whole lattice S_D. So it is co-algebraic as well. $\square$

From [8] we recall that algebraic lattices have an inf-basis of meet-irreducible elements, and so for a dI-domain D the distributive lattice S_D has both a sup-basis of join-irreducibles and an inf-basis of meet-irreducibles. We can therefore state:

Corollary 9 *The First Decomposition Theorem holds for dI-domains.*

4 Factoring by stable subdomains and the Second Decomposition Theorem

In group theory and in ring theory we are familiar with the following technique. For a given strong substructure (normal subgroup, ideal, respectively) one studies the equivalence relation which identifies those elements which differ only by an amount contained in the substructure. A similar notion works for ideals in distributive lattices: If A is an ideal in L then we can set $x \sim y$ if there is an $a \in A$ such that $x \vee a = y \vee a$. (for details see [2].) Since domains lack arbitrary suprema we have to rework this definition a little bit:

Definition. Let A be a stable subdomain in a distributive Scott-domain D. On D define a binary relation θ_A by setting $x\ \theta_A\ y$ if there is $a \in A$ such that $y = x \vee a$. Let Θ_A be the symmetric and transitive hull of θ that is the smallest equivalence relation containing θ_A. (Θ_A can be described concretely as $\bigcup_{n\in\mathbf{N}}(\theta_A^{-1} \circ \theta_A)^n$.)

This definition proves to be extremely fruitful. We list the following properties:

Proposition 10 *Let D be a finite distributive Scott-domain and let A be a stable subdomain in D. Then the following hold:*

(i) $x\ \theta_A\ y \Longrightarrow x \leq y$.

(ii) $x\ \theta_A\ y \Longrightarrow y = x \vee p_A(y)$, *and for all* $a \in A$,*if* $y = x \vee a$, *then* $a \leq p_A(y)$.

(iii) $\theta_A \circ \theta_A{=}\theta_A$.

(iv) $x\ \theta_A\ y, z \in D \Longrightarrow z \wedge x\ \theta_A\ z \wedge y$ *and* $z \vee x\ \theta_A\ z \vee y$. *(Provided the suprema exist.)*

(v) Θ_A *is a congruence relation on D with respect to finite infima and existing suprema.*

(vi) *Each equivalence class of* Θ_A *contains a least element.*

(vii) $\theta_A{=}\Theta_A \cap \leq$.

(viii) *Each equivalence class of* Θ_A *is order convex.*

(ix) $\Theta_A{=}\theta_A^{-1} \circ \theta_A$.

(x) p_A *is injective on every equivalence class of* Θ_A.

We denote the function which maps each element onto the smallest element in its equivalence class by q_A. With this notation we can add the following clauses:

(xi) q_A *is a projection on D.*

(xii) q_A *preserves existing suprema.*

Proof. (i) is trivial, for (ii) recall that p_A is join-preserving by 2-(iii).

(iii) $x\ \theta_A\ y\ \theta_A\ z \Longrightarrow y = x \vee a_1$ and $z = y \vee a_2 = x \vee a_1 \vee a_2$, and with a_1 and a_2 elements of A, their join is again in A.

(iv) $x\ \theta_A\ y \Longrightarrow y = x \vee a \Longrightarrow z \wedge y = z \wedge (x \vee a) = (z \wedge x) \vee (z \wedge a)$, and with $a \in A$, the element $z \wedge a$ is again in A. For suprema: $x\ \theta_A\ y \Longrightarrow y = x \vee a \Longrightarrow z \vee y = (z \vee x) \vee a$.

(v) It is immediate from the definition of Θ_A as a union of products of θ_A and θ_A^{-1} that (iv) also holds for Θ_A. Now, if $x\ \Theta_A\ y$ and $x'\ \Theta_A\ y'$ then $x \wedge x'\ \Theta_A\ y \wedge x'\ \Theta_A\ y \wedge y'$, and analogously for suprema.

(vi) follows from (v) by taking the infimum of the equivalence class.

(vii) Suppose $x \leq y$ and $x\ \Theta_A\ y$. Then by definition there is a chain $x_1, x_2, \ldots, x_n$ of elements such that $x = x_1\ \theta_A^{-1}\ x_2\ \theta_A\ x_3\ \theta_A^{-1}\ x_4 \ldots x_{n-1}\ \theta_A\ x_x = y$. By taking the supremum of each element of this sequence with x and then the infimum with y we derive a new sequence which is completely contained in the interval $[x, y]$. x_2 is then necessarily equal to x. We further shorten the sequence as follows: $x = x_2 = x_2 \wedge x_4\ \theta_A\ x_3 \wedge x_4\ \theta_A^{-1}\ x_4 \wedge x_4 = x_4\ \theta_A\ x_5 \ldots$ Since $x_3 \wedge x_4$ is below x_4 and in relation θ_A^{-1} it is actually equal to x_4, so the sequence now reduces to $x\ \theta_A\ x_4\ \theta_A\ x_5 \ldots$ Applying (iii) we find that x is in θ_A-relation to x_5 already. Continuing in this fashion will reduce the sequence eventually to $x\ \theta_A\ y$ which is what we want.

(viii) Assume $x\ \Theta_A\ y$ and $x \leq z \leq y$. By (vii) we have $x\ \theta_A\ y$ which implies $y = x \vee a$ for some $a \in A$. But then $z = z \wedge y = (x \vee z) \wedge (x \vee a) = x \vee (z \wedge a)$ which gives us $x\ \theta_A\ z$. The relation $z\ \theta_A\ y$ follows directly from $y = x \vee a$.

(ix) Combining (vi) and (vii) we find that the least element of an equivalence class is in θ_A-relation to each member.

(x) A projection always preserves infima and so if p_A maps two elements x and y to the same image a, it will map $x \wedge y$ to a as well, and, if $x\ \Theta_A\ y$ then $x \wedge y\ \Theta_A\ y$ and by (vii) $x \wedge y\ \theta_A\ y$. So consider w.l.o.g. $x\ \theta_A\ y$ and $p_A(x) = p_A(y)$. We directly get $y = x \vee p_A(y) = x \vee p_A(x) = x$.

(xi) We only have to show that q_A is monotone. So suppose $x \leq y$. By (vii) we have $q_A(y)\ \theta_A\ y$ which yields with (iv): $x \wedge q_A(y)\ \theta_A\ x \wedge y = x$. But $q_A(x)$ is the smallest element in the equivalence class of x. Hence $q_A(x) \leq x \wedge q_A(y) \leq q_A(y)$ follows.

(xii) From $z = x \vee y, q_A(x)\ \theta_A\ x, q_A(y)\ \theta_A\ y$ we conclude by (iv) that $q_A(x) \vee q_A(y)\ \theta_A\ x \vee y = z$. Since q_A is monotone it follows that $q_A(x) \vee q_A(y)$ must be equal to $q_A(z)$. □

Given a representation of a poset P as the cartesian product of two posets R and S we can understand P as follows: it consists of $|R|$ many copies S_x of S, and if $x \leq y$ in R then each element of S_x is below the corresponding element of S_y. A semi-factor S in a finite domain leads to a similar representation: for each element x in the image R of q_S we take the principal filter $F_x = \uparrow p_S(x)$ in S (instead of the whole semi-factor). These filters are connected as before, that is, if $x \leq y$ in R, then each element of F_x is below the corresponding element of F_y. However, there may be elements of F_x for which there is no corresponding element in F_y. This is the content of the following proposition. A picture illustrating this representation is given in Figure 1.

Knowing $S, \mathsf{im}\ q_S$ and the action of p_S on the image of q_S we can reconstruct the domain:

Proposition 11 *Let D be a finite distributive domain and let S be a semi-factor in D.*

(i) *The image of an equivalence class of Θ_S under p_S is upward closed in S.*

(ii) *D is isomorphic to the set $\hat{D} = \{(x, s) \in \mathsf{im}\ q_S \times S \mid p_S(x) \leq s\}$ ordered pointwise. The isomorphism is given by $q_S \times p_S \colon D \to \hat{D}$ and by the supremum function for the other direction.*

(iii) *If $\forall x \in D : p_S \circ q_S(x) = \perp_D$ then S is a direct factor of D.*

Proof. (i) If s is above $p_S(x)$ in S then by the extension property of semi-factors $s \vee x$ exists and is in θ_S^{-1}-relation to x. Also, $p_S(s \vee x) = p_S(s) \vee p_S(x) = s \vee p_S(x) = s$.

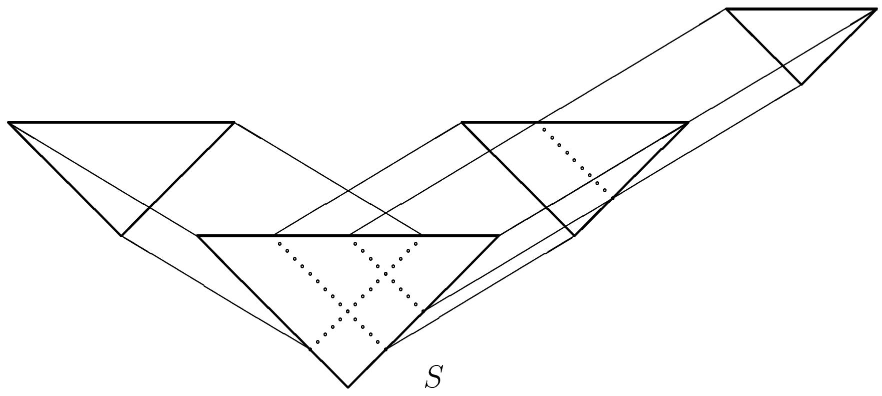


Figure 1: Decomposition of a domain by a semi-factor S.

(ii) For $x \in D$ we have $q_S(x) \leq x$ and therefore $p_S(q_S(x)) \leq p_S(x)$. So the pair $(q_S(x), p_S(x))$ belongs to $\hat{D}$. The mapping $q_S \times p_S$ is injective by Proposition 10-(x). We claim that the inverse is given by the supremum function. First of all, the supremum exists for the pairs in $\hat{D}$ because S is a semi-factor. It is clearly monotonic and it inverts $q_S \times p_S$ because $q_S \times p_S(x \vee s) = (q_S(x \vee s), p_S(x \vee s)) = (q_S(x) \vee q_S(s), p_S(x) \vee p_S(s)) = (x \vee \bot, p_S(x) \vee s) = (x, s)$ and for the other composition: $q_S(x) \vee p_S(x) = x$ because $q_S(x)\ \theta_S\ x$ by definition.

(iii) This follows because for $p_S \circ q_S = \bot_{[D \longrightarrow D]}$ the condition in the definition of $\hat{D}$ is always satisfied. $\square$

This proposition works with elements of the domain. But there is also a way of looking at this situation using congruence relations. Recall that every homomorphism $f\colon D \to E$ induces a canonical congruence relation on D, called the *kernel of* f ($\ker f$), which identifies exactly those elements of D which are mapped to the same element. Obviously, $\mathsf{ker} q_A = \Theta_A$. Let $Con(D)$ be the complete lattice of all congruences (with respect to finite infima and existing suprema) on D.

Proposition 12 *Let D be a finite distributive Scott-domain and let A be a stable subdomain in D. Then the following is true:*

(i) $\mathsf{ker}\ p_A$ *is a congruence with respect to arbitrary infima and arbitrary (existing) suprema.*

(ii) $\mathsf{ker}\ p_A \cap \Theta_A = \Delta_{D\times D} = 0_{Con(D)}$.

(iii) $\mathsf{ker}\ p_A \vee \Theta_A = D \times D = 1_{Con(D)}$.

Proof. (i) holds because p_A is a projection on a distributive domain, (ii) re-states 10-(x) and, finally, (iii) follows because every $x \in D$ is related to $\bot$ in the following way:
$x\ \Theta_A\ q_A(x)\ (\mathsf{ker} p_A)\ p_A(q_A(x))\ \Theta_A\ \bot$. $\square$

The results of this section extend to dI-domains:

Proposition 13 *Proposition 10 and Proposition 12 hold for dI-domains, in particular, equivalence classes of Θ_A and $\mathsf{ker} p_A$ are closed under directed suprema and q_A is Scott-continuous.*

Proof. The main technical difficulty is to prove that equivalence classes of Θ_A have a least element. For details we refer the reader to [19]. $\square$

We use factorization to improve on our First Decomposition Theorem. We observed that it produces representations which are redundant, namely, if two comparable semi-factors $S \subset T$ are join-irreducible in S_D then both take part in the representation, T repeating the information given by S. We shall now factor out this repeated information. Given a collection $\mathcal{S}$ of semi-factors we define for each element $S \in \mathcal{S}$ its *lower $\mathcal{S}$-cover* S' by $S' = \bigvee\{T \in \mathcal{S} \mid T \subset S\}$. Also, if $S \subseteq T \in S_D$ let $S/_T$ stand for $\mathsf{im} q_T\,|_S|$. With this notation we are now ready to formulate:

Theorem 14 (The Second Decomposition Theorem) *Let D be a finite distributive Scott-domain (a dI-domain) and let $J(S_D)$ be the set of all join-irreducible semi-factors of D. Define*

$$RJ(S_D) = \{S/_{S'} \mid S \in J(S_D)\}$$

and

$$\mathcal{F} = \{q_{S'} \circ p_S \Big|_{T/_{T'}} \mid S \subseteq T \in J(S_D)\}.$$

Then D is isomorphic to the hyper-limit of $RJ(S_D)$ over $\mathcal{F}$ with the isomorphisms

$$\begin{array}{rcl} \Psi\colon D & \to & \underset{\mathcal{F}}{\mathsf{hyperlim}}\ RJ(S_D) \\ x & \mapsto & (q_{S'} \circ p_S(x))_{S \in J(S_D)} \end{array}$$

and

$$\begin{array}{rcl} \Phi\colon \underset{\mathcal{F}}{\mathsf{hyperlim}}\ RJ(S_D) & \to & D \\ (x_S)_{S \in J(S_D)} & \mapsto & \bigvee\limits_{S \in J(S_D)} x_S. \end{array}$$

(The proof of this should be clear from the First Decomposition Theorem and Proposition 11.)

We illustrate the representation of domains provided by the Second Decomposition Theorem with Example 3 from the last section. The three join-irreducible semi-factors are $\downarrow a, \downarrow b$, and D itself. By factoring D through the join of $\downarrow a$ and $\downarrow b$ we can replace it by the three element domain $\{\perp, c, d\}$.

Decomposition into flat domains is particularly satisfying and one may wonder whether it is achievable for all distributive Scott-domains or for all dI-domains. The answer is 'no'; a counterexample is given in Figure 2.

However, it turns out that the category $\mathbf{F}$ of those distributive Scott-domains which are representable as hyperlimits of flat domains, is cartesian closed and contains strictly all *concrete domains* (cf. [14, 23]). Indeed, the connection to concrete domains seems to be very strong. Recent work by Geva and Brookes (see their contribution to this volume) suggests that every domain in $\mathbf{F}$ can be represented as a generalized concrete data structure.

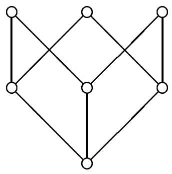


Figure 2: A non-flat indecomposable dI-domain.

5 Characterization of direct product decompositions

There exist several nice characterizations of the direct product decompositions of arbitrary algebras, see [5, 10]. In this section we will find the analogues to two of them for domains. There are several reasons to study the direct decompositions of domains. The first reason is, of course, purely theoretical. However, knowing domain decompositions may be important from the practical point of view. In the introduction we briefly described the idea of generalizing relational databases that defines a relation as a finite antichain in a Scott domain. There are several advantages of this approach. Firstly, it gives a formal framework for having attributes of arbitrary types, perhaps admitting null values. Secondly, it allows constructions more general than simply relations (matrices); for example, record and variant constructors can be applied to form very complex generalized records, see [6, 17]. Suppose that we have a Scott-domain whose finite antichains are considered as relations. A question arises: how far can this domain be seen as the direct product of its subdomains? In the other words, how far can our relations be seen as "usual" relations, that is, sets of tuples, and what are the attributes of these tuples? To answer this question we need a characterization of the direct decompositions of Scott-domains.

We will characterize direct product decompositions via complemented permutable congruences and neutral complemented ideals. Surprisingly, the characterization we will obtain is based on one of the concepts related to the domain approach to databases. In fact, all factors in the direct decompositions will be *schemes* as they were introduced in [17]. Schemes generalize semi-factors and serve as the main tool to introduce relational algebra and dependencies in the domain model of databases [17].

The decomposition theorems also enable us to prove a result which explains the other notion of scheme proposed in [6]. In fact, the definition of schemes of [6] (here we call them semi-factors) works for domains which are similar to domains of flat records. According to the definition of [17], a scheme is a stable subdomain such that the associated projection maps the maximal elements of the domain to the maximal elements of the stable subdomain. This definition emphasizes the fact that schemes are significant parts of domains. It is more general than the original definition of semi-factor which, in turn, is based on properties that schemes of domains of flat records should satisfy. We will prove that in a certain class of domains the condition that every scheme is a semi-factor implies that the domain is very similar to domain of flat records. Thus, our decomposition results are helpful in understanding the domain-theoretic generalization of databases.

This section contains five subsections. In the first we recall two well-known results from universal algebra. The second demonstrates how domains can be treated as partial algebras with operations of infinite arity, and introduces the concepts of ideal and congruence for them, the first one being identical to the stable subdomains. In the third subsection we give our basic lemma which reduces direct product decompositions of domains to those of principal ideals generated by maximal elements. Neutral ideals are studied in the fourth subsection. They give rise to so-called *general decompositions* which have coalesced sum and direct product decompositions as two limit cases. The fifth subsection deals with characterizations of direct product and general decompositions via congruences.

From now on all domains are Scott-domains, and we will write domain instead of Scott-domain. We will denote the sets of maximal elements of a domain D and of a stable subdomain A by D^{max} and A^{max}, respectively.

We will characterize only decompositions into a finite number of factors. Notice that in fact we do not need to consider more than two factors. Therefore, in all the results below only decompositions into two factors are characterized.

For the sake of brevity, only sketches of the proofs will be given, and the technical details will be omitted.

5.1 Algebraic preliminaries

Let us recall two techniques that are used in universal algebra and lattice theory to describe direct decompositions.

Given an algebra $\langle A, \Omega \rangle$, a congruence Θ on it is an equivalence relation on A such that for any $\omega \in \Omega$ of arity n and for any $x_i, y_i \in A, i \in [1, n]$, $\forall i \in [1, n] : x_i \Theta y_i$ implies $\omega(x_1, \ldots, x_n) \Theta \omega(y_1, \ldots, y_n)$. Congruences considered as binary relations form an algebraic lattice $Con(\langle A, \Omega \rangle)$ in which the bottom element is the equality and the top element is the total equivalence relation. Then direct product decompositions of $\langle A, \Omega \rangle$ (i.e. decompositions $\langle A, \Omega \rangle \simeq \langle A_1, \Omega \rangle \times \langle A_2, \Omega \rangle$) are in one-to-one correspondence with pairs (Θ_1, Θ_2) such that Θ_2 is a complement of Θ_1 in $Con(\langle A, \Omega \rangle)$ and Θ_1, Θ_2 are permutable, i.e. $\Theta_1 \circ \Theta_2 = \Theta_2 \circ \Theta_1$, see [5]. In fact, $\langle A_1, \Omega \rangle \simeq \langle A, \Omega \rangle / \Theta_1$ and $\langle A_2, \Omega \rangle \simeq \langle A, \Omega \rangle / \Theta_2$ or vice versa.

Let L be a lattice. An element $a \in L$ is called *neutral* if for every $x, y \in L$ the sublattice $\langle x, y, a \rangle$ generated by x, y, a is distributive. If L is bounded, then every direct product decomposition $L \simeq L_1 \times L_2$ can be represented as $L \simeq (a] \times (\overline{a}]$, where a is a neutral complemented element and $\overline{a}$ is its complement, see [10]. Both ideals $(a]$ and $(\overline{a}]$ are neutral elements of the ideal-lattice, i.e. so-called *neutral ideals*. It is also well-known that there is a one-to-one correspondence between direct product decompositions $L \simeq L_1 \times L_2$ and pairs (I_1, I_2) where I_1, I_2 are neutral ideals and I_2 is a complement of I_1 in the ideal-lattice. In fact, $L_1 \simeq I_1$ and $L_2 \simeq I_2$ or vice versa.

5.2 Domains as algebras. Congruences and ideals

In order to transfer the previous characterizations to domains, we have to introduce algebraic structure on domains. Let D be a domain. We consider it as a partial algebra containing the operations of infinite arity, namely infima and existing suprema for all possible subsets $X \subseteq D$. Thus, D becomes a partial algebra whose operations may be of arbitrary arity. It is well-known that the previous results about decompositions are not true for algebras with partial operations of infinite arity [5].

A subalgebra of this partial algebra could be called *subdomain* but in semantics this notion has no generally accepted meaning. In lattice theory an ideal is a downward closed set which is closed under finite joins. In our algebraic interpretation *ideals* are downward closed subsets of a domain which are closed under all existing joins, i.e. they are *stable subdomains*. A *congruence* is an equivalence relation Θ such that for any $x_i, y_i \in D, i \in I, I$ an arbitrary set of indices, $x_i \Theta y_i$ for all $i \in I$ implies $\bigwedge\{x_i : i \in I\} \Theta \bigwedge\{y_i : i \in I\}$, and if both $x = \bigvee\{x_i : i \in I\}$ and $y = \bigvee\{y_i : i \in I\}$ exist, then $x \Theta y$.

If D is a lattice, our definition of congruence coincides with the definition of *complete congruence* of a lattice introduced recently in [11, 12]. These congruences form a complete lattice denoted $Con(D)$. However, in contrast to the case of operations of finite arity, this lattice may fail to be algebraic. It was proved in [11] that for every complete lattice L there exists a lattice M such that L is isomorphic to the lattice of complete congruences of M. Moreover, M can be chosen among modular algebraic lattices [12]. Therefore, we cannot guarantee algebraicity of $Con(D)$.

If $x \in D$, then Θ^x is the restriction of Θ to $\downarrow x$, that is, $\Theta \cap (\downarrow x \times \downarrow x)$. The lattices of congruences of $\downarrow x$ will be denoted by $Con(x)$ if D is understood. $[x]\Theta$ denotes the equivalence class of x, i.e. $[x]\Theta = \{y : x\Theta y\}$. If Θ is a congruence, then $[\perp]\Theta$ is an ideal, that is, a stable subdomain.

We will also also need a concept of *scheme*. It was introduced in [17] in order to generalize the analogous concept of [6] known in this paper as semi-factor. A stable subdomain $A \subseteq D$ is called a *scheme* if $p_A(x)$ is a maximal element of A for every $x \in D^{max}$. Any semi-factor is a scheme. Although all the factors we will decompose domains into will be semi-factors, it is enough to require that they be schemes.

5.3 The main lemma

The following lemma generalizes the result of [10] which describes direct product decompositions of bounded lattices. Theorem 3.4.1 of [10] says that the direct product decompositions of a bounded lattice L into two factors are in one-to-one correspondence with neutral complemented elements $a \in L$. In fact, $L \simeq \downarrow a \times \uparrow a$, or equivalently $L \simeq \downarrow a \times \downarrow \overline{a}$. In the lemma below a pair of complemented schemes plays the role of $(a, \overline{a})$, and it is required that the projections of any maximal element x into these schemes be neutral and complementary in $\downarrow x$.

Notice also that according to [10] direct decompositions can be equivalently described by pairs of neutral complementary ideals (in fact, these ideals are just $\downarrow a$ and $\downarrow \overline{a}$), and we can view the pair of schemes as an analogy of these ideals rather than neutral elements. However, in contrast to the lattice case, neutral complementary elements of the lattice of ideals (stable subdomains) of a domain do not characterize direct decompositions, as we will see in the next subsection.

Lemma 15 *The direct product decompositions of a domain D are given by pairs of schemes (A_1, A_2) such that $A_1 \cap A_2 = \{\perp\}$, and $\forall x \in A_1^{max}, y \in A_2^{max} : x \vee y$ exists, $x \vee y \in D^{max}$, and x is a neutral element in the principal ideal $\downarrow(x \vee y)$. In fact, $D \simeq A_1 \times A_2$. Moreover, A_1 and A_2 are neutral complementary elements of the lattice Q_D of stable subdomains.*

Proof. Let $D \simeq D_1 \times D_2$. We can think of both D_1 and D_2 as being subsets of D, i.e. we can associate D_1 with $\{(x, \perp) : x \in D_1\}$ and D_2 with $\{(\perp, y) : y \in D_2\}$. Denote these subsets by A_1 and A_2. Obviously, both A_1 and A_2 are stable subdomains. We have $A_1 \cap A_2 = \{\perp_D\}$. If $x \in A_1^{max}$ and $y \in A_2^{max}$, then $x = \langle x', \perp \rangle, y = \langle \perp, y' \rangle$ where $x' \in D_1^{max}, y \in D_2^{max}$. Thus, $x \vee y = \langle x', y' \rangle \in D^{max}$. Obviously, x and y complement each other in $\downarrow(x \vee y)$. To prove that x is neutral we need to show that $\downarrow(x \vee y) \simeq \downarrow x \times \downarrow y$ [10]. This isomorphism is given by $\varphi((u, v)) = \langle (u, \perp), (\perp, v) \rangle$ for any $(u, v) \leq_D x \vee y$. The fact that A_1 and A_2 are neutral and complement each other in Q_D will follow from Theorem 17.

Let, conversely, A_1, A_2 be stable subdomains satisfying the conditions of the lemma. We prove $D \simeq A_1 \times A_2$. Consider two mappings: $\varphi : D \to A_1 \times A_2$ and $\psi : A_1 \times A_2 \to D$ where $\varphi(x) = \langle p_{A_1}(x), p_{A_2}(y) \rangle$ and $\psi(\langle u, v \rangle) = u \vee v$ (notice that this $u \vee v$ always exists). Then the following claims can be proved:

- If $x \in A_1, y \in A_2$, then $\varphi(x \vee y) = \langle x, y \rangle$;
- For any $z \in D$: $z = p_{A_1}(z) \vee p_{A_2}(z)$.

From these two claims it can be easily concluded that φ and ψ are mutually inverse bijections. Since they are both monotone, they establish the desired isomorphism. □

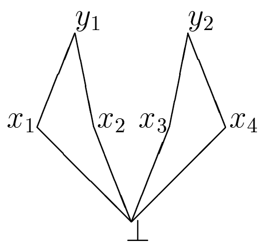
Remark: We did not exploit the fact that domains are algebraic cpos in the proof of lemma. In fact, the lemma is true for any bounded poset P in which greatest lower bounds exist for all pairs of elements and least upper bounds exist for all pairs bounded above. Here boundedness means that P is pointed, that is, it has the smallest element $\perp$, and each $x \in P$ is bounded above by some maximal $x' \in P^{max}$.

5.4 General decompositions of domains

It is a natural question whether all neutral complemented elements of the lattice of stable subdomains give rise to direct product decompositions, as it is the case for lattices. The answer, as we are going to show, is "no". In fact, neutral complemented ideals describe a much more general kind of decomposition which, for example, also includes coalesced sum decomposition.

Definition. A pair of stable subdomains A_1, A_2 of a domain D is called a *general decomposition* of D, which is denoted by $D = comp(A_1, A_2)$, if every element $x \in D$ has unique representation as $x = x_1 \vee x_2$ such that $x_1 \in A_1$ and $x_2 \in A_2$.

The direct decompositions $D \simeq A_1 \times A_2$ and the coalesced sum decompositions $D \simeq A_1 + A_2$, where A_1, A_2 are stable subdomains of D, are two main examples of the general decompositions; that is, in both cases $D = comp(A_1, A_2)$. However, there exist general decompositions different from the direct and coalesced sum decompositions. Consider the following domain D:



Then D has several general decompositions. For example, $D = comp(\{x_1, x_3, \perp\}, \{x_2, x_4, \perp\})$ or $D = comp(\{x_1, x_4, \perp\}, \{x_2, x_3, \perp\})$.

Obviously, if $D = comp(A_1, A_2)$ then $A_1 \cap A_2 = \{\perp\}$. Really, if $x \in A_1 \cap A_2$ and $x \neq \perp$, we have two representations $x = x \vee x$ and $x = x \vee \perp$.

Lemma 16 *If $D = comp(A_1, A_2)$, then the unique representation of any $x \in D$ as $x_1 \vee x_2$ where $x_1 \in A_1, x_2 \in A_2$, is $x = p_{A_1}(x) \vee p_{A_2}(x)$.* □

Theorem 17 *The general decompositions of a domain D are in one-to-one correspondence to pairs of neutral complementary elements of the lattice Q_D, i.e. $D = comp(A_1, A_2)$ iff A_1, A_2 are neutral elements of Q_D complementing each other.*

Proof. Let $D = comp(A_1, A_2)$. Then A_1 and A_2 complement each other in Q_D. Thus, we only need to prove that A_1 is neutral in Q_D. To do so, we will show that there is an isomorphism $\varphi : Q_D \to Q_{A_1} \times Q_{A_2}$ such that $\varphi(A_1) = \langle \mathbf{1}_{Q_{A_1}}, \mathbf{0}_{Q_{A_2}} \rangle$, where $\mathbf{1}$ and $\mathbf{0}$ stand for the top and bottom elements of a lattice. Then the neutrality will follow from [10].

Given $A \in Q_D$, let $\varphi(A) = \langle A \cap A_1, A \cap A_2 \rangle$. As in the proof of Lemma 15, introduce another map $\psi : Q_{A_1} \times Q_{A_2} \to Q_D$ by $\psi(\langle I_1, I_2 \rangle) = I_1 \vee I_2$. According to Lemma 16, for any $A \in Q_D : A = (A \cap A_1) \vee (A \cap A_2)$; that is, $A = \psi(\varphi(A))$.

Let $I_1 \subseteq A_1, I_2 \subseteq A_2$ be stable subdomains. Let $I = \{x_1 \vee x_2 : x_1 \in I_1, x_2 \in I_2\}$. Suppose $x \leq x_1 \vee x_2 \in I$. If $z = x_1 \vee x_2$, then $p_{A_i}(x) \leq p_{A_i}(z)$, $i = 1, 2$. By Lemma 16, $x_i = p_{A_i}(z)$. Thus, $p_{A_i}(x) \in I_i$, and $x = p_{A_1}(x) \vee p_{A_2}(x) \in I$. This shows that I is an ideal. It is also easy to show that I is a stable subdomain. Now, let $x = x_1 \vee x_2 \in I \cap A_1$. Then $\perp = p_{A_2}(x_1 \vee x_2) = x_2$, i.e. $x \in I_1$. This shows $(I_1 \vee I_2) \cap A_i = I_i, i = 1, 2$. In the other words, $\varphi(\psi(\langle I_1, I_2 \rangle)) = \langle I_1, I_2 \rangle$.

Therefore, φ and ψ are mutually inverse monotone bijections, which establishes the desired isomorphism. Since $\varphi(A_1) = \langle A_1, \{\perp\} \rangle$, A_1 is neutral.

Conversely, let A_1, A_2 be two complementary neutral elements of Q_D. Then, by [10] for any $A \in Q_D : A = (A \cap A_1) \vee (A \cap A_2)$. If $A = \downarrow x$ we have $\downarrow x = \downarrow p_{A_1}(x) \vee \downarrow p_{A_2}(x)$. Hence, $x = p_{A_1}(x) \vee p_{A_2}(x)$. If $x = x_1 \vee x_2$, where $x_i \in A_i, i = 1, 2$, then $\downarrow p_{A_1}(x) = \downarrow x \cap A_1 = (\downarrow x_1 \vee \downarrow x_2) \cap A_1 = (\downarrow x_1 \cap A_1) \vee (\downarrow x_2 \cap A_1) = \downarrow x_1 \cap A_1 = \downarrow x_1$. Thus, $x_1 = p_{A_1}(x)$. Analogously, $x_2 = p_{A_2}(x)$. This shows that $D = comp(A_1, A_2)$. The proof is complete. □

The neutral complemented elements of any lattice L form a Boolean sublattice often denoted $Cen(L)$. Notice that if Q_D does not have infinite chains, then $Cen(Q_D)$ is a finite Boolean lattice, and only representation of D as the composition of indecomposable factors is (in the sense of the operation of general decomposition) $D = comp_{A \in Atoms(Cen(Q_D))} A$, where $Atoms(Cen(Q_D))$ is the set of atoms of the finite lattice $Cen(Q_D)$. For example, if D does not have infinite chains, then the same is true of Q_D, and the following corollary holds.

Corollary 18 *If D does not have infinite chains, D can be factored into indecomposable subdomains in one and only one way.* □

It was mentioned before that direct product and coalesced sum decompositions appear now as two limit cases of general decompositions. In fact, the following holds.

Proposition 19 1) *A general decomposition $D = comp(A_1, A_2)$ is a direct product decomposition $D \simeq A_1 \times A_2$ iff $\forall x \in A_1, y \in A_2 : x \vee y$ exists.*

2) *A general decomposition* $D = comp(A_1, A_2)$ *is a coalesced sum decomposition* $D = A_1 + A_2$ *iff* $\forall x \in A_1, y \in A_2, x, y \neq \perp : x \vee y$ *does not exist.*

Proof. 1) If $D \simeq D_1 \times D_2$ and A_1, A_2 are defined as in the proof of lemma 15, then $D \simeq A_1 \times A_2$ and $D = comp(A_1, A_2)$. Obviously, $x \vee y$ exists for any $x \in A_1, y \in A_2$.

Now, suppose that $D = comp(A_1, A_2)$ and the condition of 1) holds. In order to prove $D \simeq A_1 \times A_2$ we must check the conditions of Lemma 15, i.e.

a) For any $x \in A_1^{max}, y \in A_2^{max} : x \vee y \in D^{max}$;

b) A_1, A_2 are the schemes;

c) x is neutral in $\downarrow(x \vee y)$.

To prove a) notice that if $v \geq x \vee y$ then $x = p_{A_1}(v)$ since $x \in A_1^{max}$. Analogously $y = p_{A_2}(v)$ and by Lemma 16 $v = x \vee y$. To prove b) consider any $v \in D^{max}$. If $w \geq p_{A_1}(v)$ and $w \in A_1^{max}$, then since $w \vee p_{A_2}(v)$ exists, $v = w \vee p_{A_2}(v)$, i.e. $p_{A_1}(v) = w \in A_1^{max}$, and A_1 is a scheme. Analogously A_2 is a scheme. To prove c) notice that for any $z \in \downarrow(x \vee y)$: $p_{A_1}(z) = z \wedge x$ and $p_{A_2}(z) = z \wedge y$; thus $z = (z \wedge x) \vee (z \wedge y)$. Then $\varphi : \downarrow(x \vee y) \to \downarrow x \times \downarrow y$ given by $\varphi(z) = \langle z \wedge x, z \wedge y \rangle$ is an injective monotone map. Since $\langle z_1, z_2 \rangle \to z_1 \vee z_2$ is its monotone inverse, φ is a lattice isomorphism, and x is neutral because $\varphi(x) = \langle x, \perp \rangle$.

2) is straightforward. $\square$

It follows from part 1) that if D is a lattice, its general decompositions are exactly direct decompositions. Therefore, from Corollary 18 we derive the well-known fact that a lattice without infinite chains admits exactly one representation as the direct product of nontrivial indecomposable factors (up to isomorphism and permutation of factors), see [5, 10].

5.5 Characterization via congruences

We have shown so far that, in contrast to the lattice case, the direct decompositions of domains are not described by their neutral complemented ideals. We had to introduce another kind of decomposition, called general decomposition, to be in 1-1 correspondence with these ideals.

In this subsection we examine another approach to describing decompositions. We will try to find and characterize pairs of congruences such that quotient sets are exactly factors of a decomposition. In universal algebra it is known that pairs of permutable complementary congruences exactly describe direct decompositions. This result holds for domains too, although domains are considered as algebras with partial operations of finite and infinite arity. Congruences also help understand the general decompositions better. It will be proved that factors of general decompositions are in 1-1 correspondence with pairs of congruences which are complementary and permutable when restricted to the principal ideals of maximal elements.

Define the following two mappings between the pairs of congruences Θ_1 and Θ_2 and pairs of stable subdomains of a domain $A_1, A_2 \subseteq D$:

(1) $$(\Theta_1, \Theta_2) \to (A_1, A_2) \quad : \quad A_1 = [\perp]\Theta_2, A_2 = [\perp]\Theta_1,$$

$$(2) \qquad (A_1, A_2) \to (\Theta_1, \Theta_2) \quad : \quad x\Theta_i y \text{ iff } p_{A_i}(x) = p_{A_i}(y), \quad i = 1, 2.$$

To be more precise, if Θ_1 and Θ_2 are congruences, then A_1, A_2 defined in (1) are stable subdomains, and for a pair of stable subdomains the relations defined in (2) are equivalence relations. However, they are congruences in the special cases considered below.

The following two theorems show that above defined (1) and (2) set up 1-1 correspondences between the factors of direct decompositions or general decompositions of a domain and the special pairs of congruences on this domain.

Theorem 20 *Let D be a domain. Then (1) and (2) form a one-to-one correspondence between the factors of the general decompositions $D = comp(A_1, A_2)$ and pairs of congruences (Θ_1, Θ_2) such that, for every $x \in D^{max}$, the congruences Θ_1^x and Θ_2^x are permutable, and Θ_2^x is a complement of Θ_1^x in $Con(x)$.*

Theorem 21 *The mappings (1) and (2) $(A_1, A_2) \leftrightarrow (\Theta_1, \Theta_2)$ form a one-to-one correspondence between the factors of the direct product decompositions $D \simeq A_1 \times A_2$ and pairs (Θ_1, Θ_2) of congruences such that Θ_1 and Θ_2 are permutable, and Θ_2 is a complement of Θ_1 in $Con(D)$.*

Before going to the proof of these two theorems, we formulate and prove one corollary which clarifies why, in the introduction to this subsection, we spoke of quotient sets rather than equivalence classes of bottom element.

Corollary 22 *In the previous theorems the factors A_1, A_2 the domain D is decomposed into are isomorphic to D/Θ_1 and D/Θ_2, respectively.*

For example, we have a perfect analogy of the congruence characterization of the decompositions of algebras. In fact, all the decompositions of a domain D are of form $D \simeq D/\Theta_1 \times D/\Theta_2$ for pairs (Θ_1, Θ_2) of complementary permutable congruences.

Proof. Let $\psi : D/\Theta_1 \to A_1$ be given by $\psi([x]\Theta_1) = p_{A_1}(x)$. This definition is correct by (2). It also follows from (2) that ψ is injective and since $\forall z \in A_1 : \psi([z]\Theta_1) = z$, it is surjective as well. Let $[x]\Theta_1 \leq [y]\Theta_1$ in D/Θ_1. Then $(x \wedge y)\Theta_1 x$; thus $p_{A_1}(x) \wedge p_{A_1}(y) \leq p_{A_1}(x)$, which proves the monotonicity of ψ. $\varphi : A_1 \to D/\Theta_1$ given by $\varphi(z) = [z]\Theta_1$ is the inverse of ψ, and it is also monotone. Thus, $A_1 \simeq D/\Theta_1$. The proof of $A_2 \simeq D/\Theta_2$ is the same. □

Proof. Two claims below are needed in the proofs of both theorems. After proving these claims, we outline the proofs of the theorems themselves.

Claim 1: Θ_1 and Θ_2 defined in (2) are congruences, if $D = comp(A_1, A_2)$ or $D \simeq A_1 \times A_2$.

It is enough to show Θ_1 is a congruence. Obviously, Θ_1 is an equivalence relation. Since $p_{A_1}(\bigwedge_{i \in I} x_i) = \bigwedge_{i \in I} p_{A_1}(x_i)$ for any set of indices I, Θ_1 preserves arbitrary meets. If $x \vee y$ exists, then $p_{A_1}(x \vee y) = p_{A_1}(x) \vee p_{A_1}(y)$. Really, since A_1 is a neutral element of Q_D by Theorem 17, if $z \in A_1$, such that $x \vee y \geq z > p_{A_1}(x) \vee p_{A_1}(y)$, then $z \in A_1 \wedge (\downarrow x \vee \downarrow y) = (A_1 \wedge \downarrow x) \vee (A_1 \wedge \downarrow y) = \downarrow p_{A_1}(x) \vee \downarrow p_{A_1}(y) = \downarrow p_{A_1}(x) \vee p_{A_1}(y) \not\ni z$, a contradiction. We conclude that the above equality holds and Θ_1 preserves finite joins. Now, let $x = \bigvee_{i \in I} x_i$. Then $p_{A_1}(x) \geq \bigvee_{i \in I} p_{A_1}(x_i)$. To prove the reverse inequality, suppose $y \leq p_{A_1}(x)$ and y is compact. Then $y \leq x$ and $y \leq \bigvee_{i\in I_0} x_i$ where $I_0 \subseteq I$ is finite. Thus, $y \leq p_{A_1}(\bigvee_{i\in I_0} x_i) = \bigvee_{i\in I_0} p_{A_1}(x_i) \leq \bigvee_{i\in I} p_{A_1}(x_i)$. Since any element is the join of compact elements below it, this proves the reverse inequality. Thus, p_{A_1} preserves arbitrary joins, and so does Θ_1, i.e. Θ_1 is a congruence. The claim is proved.

Claim 2: Let Θ_1, Θ_2 be permutable congruences on a domain D. Then their join in $Con(D)$ is $\Theta_1 \vee \Theta_2 = \Theta_1 \cdot \Theta_2$.

Since $\Theta_1 \cdot \Theta_2 \leq_{Con(D)} \Theta_1 \vee \Theta_2$, it suffices to show that $\Theta = \Theta_1 \cdot \Theta_2$ is a congruence. The only nontrivial part of the proof is to show that Θ preserves arbitrary existing joins. Assume that $x = \bigvee_{i\in I} x_i$ and $y = \bigvee_{i\in I} y_i$ exist, and $x_i \Theta y_i$ for all $i \in I$. Then for each i there is a z_i such that $x_i \Theta_1 z_i$ and $z_i \Theta_2 y_i$. Then $x_i \Theta_1 x_i \wedge z_i$. From $z_i \Theta_2 y_i$ we have $z \wedge x_i \Theta_2 z_i \wedge x_i \wedge y_i$. Analogously we have $y_i \wedge z_i \Theta_1 x_i \wedge y_i \wedge z_i$ and $y_i \Theta_2 y_i \wedge z_i$. Let $v = \bigvee_{i\in I}(x_i \wedge z_i), w = \bigvee_{i\in I}(x_i \wedge y_i \wedge z_i), u = \bigvee_{i\in I}(y_i \wedge z_i)$ (they exist since x and y exist). Then $x\Theta_1 v \Theta_2 w \Theta_1 u \Theta_2 y$ because $\Theta_1, \Theta_2 \in Con(D)$. Since they are permutable, we obtain $x\Theta y$. The claim is proved.

Now let us come back to the proof of Theorem 20. Let $D = comp(A_1, A_2)$. Then Θ_1, Θ_2 are congruences and it is easy to show that for any $x \in D^{max} : \Theta_1^x$ complements Θ_2^x in $Con(x)$. To show that they are permutable, note that if $a\Theta_1^x c \Theta_2^x b$, then $d = p_{A_2}(a) \vee p_{A_1}(b)$ exists and $a\Theta_2^x d \Theta_1^x b$.

Conversely, let Θ_1, Θ_2 be congruences defined in Theorem 20, and let A_1, A_2 be obtained as in (1). Then A_1, A_2 are stable subdomains. We need to prove $D = comp(A_1, A_2)$. The idea of the proof is the following. Since $\Theta_1^x \cdot \Theta_2^x$ is the total relation by claim 2, for any $y \leq x : y\Theta_1^x z \Theta_2^x \bot$ for some z. Then $z \in A_1$ and $y\Theta_1^x z \wedge y$; thus $y\Theta_1^x p_{A_1}(y)$. Similarly, $y\Theta_2^x p_{A_2}(y)$. If $x, y \in A_i$ and $x < y$, then $(x, y) \notin \Theta_i$. From this we can conclude that $x\Theta_i y \Leftrightarrow p_{A_i}(x) = p_{A_i}(y)$. Having proved this, we can use the fact that Θ_1^x and Θ_2^x complement each other to demonstrate that $y = p_{A_1}(y) \vee p_{A_2}(y)$ is the unique representation of y as the join of two elements from A_1 and A_2. Thus, $D = comp(A_1, A_2)$, which finishes the proof of Theorem 20.

Let us give the sketch of the proof of Theorem 21. If $D \simeq A_1 \times A_2$, then it is not hard to show that Θ_1 and Θ_2 defined in (2) are complementary and permutable.

If Θ_1 and Θ_2 are complementary and permutable in $Con(D)$, then so are Θ_1^x and Θ_2^x in $\downarrow x$ for any $x \in D^{max}$. Thus, for A_1, A_2 defined as in (1), we have $D = comp(A_1, A_2)$ and $x\Theta_i y \Leftrightarrow p_{A_i}(x) = p_{A_i}(y)$ by Theorem 20. Let $x \in A_1, y \in A_2$. Then $x\Theta_2 \bot \Theta_1 y$. Since the congruences are permutable, for some $z : x\Theta_1 z \Theta_2 y$. Thus, $p_{A_1}(z) = x, p_{A_2}(z) = y$ and $z \geq x, y$. Therefore, $x \vee y$ exists, and by Proposition 19, $D \simeq A_1 \times A_2$. Theorem 21 is proved. □

A simpler result can be stated for *qualitative* domains. Recall that a domain D is called *qualitative* iff $\downarrow x$ is a Boolean lattice for every $x \in D$ [9]. Since all congruences are permutable in a Boolean lattice, we have:

Corollary 23 *General decompositions of a qualitative domain are given by pairs of congruences (Θ_1, Θ_2) such that Θ_1^x and Θ_2^x are complementary elements in $Con(x)$ for every $x \in D^{max}$.* □

6 Schemes and semi-factors in qualitative domains: a database point of view

It has been mentioned several times before that the concepts of semi-factor and scheme first appeared as two different attempts to define an analogy of *scheme* for the domain model of databases. In this section we will apply our decomposition results to find out when these definitions coincide. But let us give some basic facts about the domain model of databases first.

For simplicity, consider the flat domain $\mathbf{N}_\perp$ of natural numbers and assume that we have a relational database which stores the information about the hotel rooms such as the room number, number of baths in the room, telephone extension and the date it becomes free. We further assume that all attributes' values are taken from $\mathbf{N}_\perp$. We need $\perp$ if a piece of information is unknown; for example, there might be no phone in the room, or we might not know when the occupant is going to leave. To represent the last attribute's values by natural numbers we may store the difference between the date the occupant leaves the room and the current date, decrementing it every day. Then the examples of records are:

$$\begin{aligned} r_1 &= \{\ \text{RoomNo} \Rightarrow \text{'121'},\ \text{Bath} \Rightarrow \text{'1'},\ \text{Phone} \Rightarrow \text{'9510'},\ \text{Free} \Rightarrow \text{'5'}\ \} \\ r_2 &= \{\ \text{RoomNo} \Rightarrow \text{'323'},\ \text{Bath} \Rightarrow \text{'2'},\ \text{Phone} \Rightarrow \text{'0752'},\ \text{Free} \Rightarrow \perp\} \\ r_3 &= \{\ \text{RoomNo} \Rightarrow \text{'323'},\ \text{Bath} \Rightarrow \text{'2'},\ \text{Phone} \Rightarrow \perp\ \text{Free} \Rightarrow \perp\} \end{aligned}$$

If $\mathcal{L}$ is the set of attributes, i.e. $\mathcal{L} = \{\text{RoomNo}, \text{Bath}, \text{Phone}, \text{Free}\}$, then the records stored in the database can be represented as functions from $\mathcal{L}$ to $\mathbf{N}_\perp$. We denote the domain of these records by $\mathcal{L} \to \mathbf{N}_\perp$. Of course, $\mathcal{L} \to \mathbf{N}_\perp \simeq \mathbf{N}_\perp^n$, where $n = |\mathcal{L}|$. In the above example $r_1 \in (\mathcal{L} \to \mathbf{N}_\perp)^{max}$ and $r_3 \leq r_2$.

Since records are elements of a domain, the relations are subsets of this domain. Since there is no need to store two comparable records (as r_2 and r_3) because one of them is just a worse description of the same object, the relations in our example are *finite antichains* in $\mathcal{L} \to \mathbf{N}_\perp$. This observation led the authors of [6] to the idea to generalize relational databases as finite antichains in Scott-domains. This idea was further developed in [17], where more complicated examples can be found. A generalized relational algebra was also constructed in [17].

One of the central concept of relational database theory is that of *scheme*. A scheme is simply a subset of attributes, but since they are used to define projections, one can alternatively associate a scheme with all records that are obtained as the projections into this scheme, that is, all records whose projections on attributes not in the scheme are bottom elements. Schemes in $\mathcal{L} \to \mathbf{N}_\perp$, therefore, are stable subdomains.

The way the definition of semi-factors is justified in [6] is the following. If we accept the perfect analogy between our arbitrary domain and the domain of flat records (such as $\mathcal{L} \to \mathbf{N}_\perp$), we may assume that every element of a domain which is a database object can be represented as two subobjects, each carrying an independent piece of information. Hence, projecting into a scheme is losing a certain piece of information, and the lost pieces and the projections are independent, i.e. the lost pieces can be added to the projections to get the records back. If we assume that we have two records, x and y, and projection of x into a scheme A is less than y, then adding information lost when x was projected to y is possible, and the result is more informative than both x and y. But this is exactly the definition of semi-factor: $p_A(x) \leq y \in S$ implies the existence of $x \vee y$.

An alternative approach of [17] does not make any assumption about the structure of the underlying domain. Just note that if we have a complete description, projecting into a scheme should not result in the loss of some piece of information that the scheme preserves, that is, complete descriptions are projected into complete descriptions, i.e. $p_A(D^{max}) = A^{max}$. And this is the definition of scheme.

Thus, a question arises: what are the domains in which the two concepts coincide (or, equivalently, every scheme is a semi-factor)? According to the justifications of the two definitions, we may expect these domains to be similar to the domain of flat records. We will prove the result for the *qualitative* domains. We need to define the blocks (analogous to flat domains) from which these domain will be built.

A domain D is called *simple* if it has no proper scheme, i.e. if it has no scheme but $\{\perp\}$ and itself. Since schemes appear as equivalence classes of congruences, this definition is motivated by the definition of a simple lattice, i.e. a lattice having no nontrivial congruences [5, 10].

Theorem 24 *Let D be a qualitative domain. Then every scheme in D is a semi-factor if and only if D is isomorphic to the direct product of simple domains.*

Proof. The *'if'* part is easy. To prove *'only if'*, suppose that any scheme is a semi-factor. Given a scheme A, define $\overline{A} = \{x \in D : p_A(x) = \perp\}$. Then $\overline{A}$ is a scheme too [17]. Since S_D is distributive and $\overline{A}$ is a complement of A in S_D, S_D is a Boolean lattice. Since it is algebraic by the corollary to the First Decomposition Theorem, it is atomistic, and $J(S_D)$ is the set of atoms of S_D. It then follows from the First Decomposition Theorem that D is isomorphic to the direct product of all elements of $J(S_D)$, since the limit over an empty set of connecting projections is the direct product. It is an easy observation that all schemes from $J(S_D)$ are simple domains, which finishes the proof. □

Corollary 25 *Let D be a qualitative domain in which every scheme is a semi-factor. Then S_D is an atomic Boolean lattice and the schemes of D are in 1-1 correspondence with subsets of the set of atoms of S_D.* □

This corollary gives a mathematical description of the assumption that every database object represented as an element of a domain can be decomposed into two "independent" subobjects, namely to its projection onto a scheme and its complement in S_D.

Acknowledgement

We would like to thank Peter Buneman for inventing the beautiful tool of semi-factors for us domain-theorists. Together with the first author he found a preliminary version of the First Decomposition Theorem while he visited Darmstadt in July 1989. We also thank Klaus Keimel for directing our attention to the classical theory of ideals in distributive lattices. He realized that our definition of the congruence Θ_S generalizes the classical definition to domains. G. Grätzer and E.T. Schmidt informed us about recent results on complete congruences and interrupted our vain attempts to prove algebraicity for complete congruence lattices of domains[1].

[1] While a revised version of this paper was being prepared, G. Grätzer and E.T. Schmidt wrote a paper titled *"On a congruence lattice of a Scott-domain"* in which they proved that every complete lattice is the lattice of congruences of a Scott-domain.

References

[1] S. Abramsky. Domain Theory in Logical Form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[2] R. Balbes and P. Dwinger. *Distributive Lattices.* University of Missouri Press, Columbia, 1974.

[3] G. Berry. Modèles Complèment Adéquats et Stables des Lambda-calculs typés. Thèse de Doctorat d'Etat, Université Paris VII, 1979.

[4] G. Berry. Stable Models of Typed λ-calculi. In *Proceedings of the 5th International Colloquium on Automata, Languages and Programming*, volume 62 of *Lecture Notes in Computer Science*, pages 72–89. Springer Verlag, 1978.

[5] G. Birkhoff. *Lattice Theory*, volume 25 of *AMS Colloq. Publ.* American Mathematical Society, Providence, third edition, 1967.

[6] P. Buneman, A. Jung, and A. Ohori. Using Powerdomains to Generalize Relational Databases. *Theoretical Computer Science*, 91:23–55, 1991.

[7] P.-L. Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming.* Pitman, 1986.

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

[9] J.-Y. Girard. The System F of Variable Types: Fifteen Years Later. *Theoretical Computer Science*, 45:159–192, 1986.

[10] G. Grätzer. *General Lattice Theory.* Birkhäuser Verlag, Basel, 1978.

[11] G. Grätzer. The complete congruence lattice of a complete lattice. *Proc. of Int. Conf. on Lattices, Semigroups, and Universal Algebra.* Plenum Press, New York (1990), 81–88.

[12] G. Grätzer and E. T. Schmidt. A representation of $m-$algebraic lattices. *Algebra Universalis*, to appear.

[13] A. Jung. *Cartesian Closed Categories of Domains*, volume 66 of *CWI Tracts.* Centrum voor Wiskunde en Informatica, Amsterdam, 1989.

[14] G. Kahn and G. Plotkin. Domaines Concrets. Technical Report 336, INRIA-Laboria, 1978.

[15] A. Kanda. Fully Effective Solutions of Recursive Domain Equations. In J. Beçvar, editor, *Mathematical Foundations of Computer Science*, volume 74. Springer-Verlag, 1979. Lecture Notes in Computer Science.

[16] K. G. Larsen and G. Winskel. Using Information Systems to Solve Recursive Domain Equations Effectively. In G. Kahn, D. B. MacQueen, and G. Plotkin, editors, *Semantics of Data Types*, volume 173 of *Lecture Notes in Computer Science*, pages 109–130. Springer-Verlag, 1984.

[17] 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.

[18] G. D. Plotkin. Post-Graduate Lecture Notes in Advanced Domain Theory (incorporating the "Pisa Notes"). Dept. of Computer Science, Univ. of Edinburgh, 1981.

[19] H. Puhlmann. Verallgemeinerung relationaler Schemata in Datenbanken mit Informationsordnung, 1990. (Diplomarbeit, Technische Hochschule Darmstadt.).

[20] M. B. Smyth. Effectively Given Domains. *Theoretical Computer Science*, 5:257–274, 1977.

[21] P. Taylor. Homomorphisms, Bilimits and Saturated Domains. Some Very Basic Domain Theory. Draft, Imperial College London, 15pp., 1987.

[22] K. Weihrauch and T. Deil. Berechenbarkeit auf cpo's. Technical Report 63, Rheinisch-Westfälische Technische Hochschule Aachen, 1980.

[23] G. Winskel. *Events in Computation.* PhD thesis, University of Edinburgh, 1981.