

Article

An Algebraic Theory of Information: An Introduction and Survey

Juerg Kohlas ^{1,*} and Juerg Schmid ²

¹ Department of Informatics, University of Fribourg, Bvd. de Perolles 90, CH-1700 Fribourg, Switzerland

² Mathematical Institute, University of Bern, Sidlerstrasse 5, CH-3012 Bern, Switzerland;
E-Mail: juerg.schmid@math.unibe.ch

* Author to whom correspondence should be addressed; E-Mail: juerg.kohlas@unifr.ch.

Received: 22 November 2013; in revised form: 8 April 2014 / Accepted: 8 April 2014 /

Published: 10 April 2014

Abstract: This review examines some particular, but important and basic aspects of information: Information is related to questions and should provide at least partial answers. Information comes in pieces, and it should be possible to aggregate these pieces. Finally, it should be possible to extract that part of a piece of information which relates to a given question. Modeling these concepts leads to an algebraic theory of information. This theory centers around two different but closely related types of information algebras, each containing operations for aggregation or combination of information and for extracting information relevant to a given question. Generic constructions of instances of such algebras are presented. In particular, the close connection of information algebras to logic and domain theory will be exhibited.

Keywords: information; information theory; universal logic; domain theory; order theory

1. Introduction: Modeling Information

The approach to the concept of information proposed here is based on the view that information comes in pieces, that information relates to questions and that it provides at least partial answers to questions. Since information comes in pieces, it should be aggregated or combined. It must be possible to extract from any piece of information the part pertaining to a given question, to focus the information on a given

question. This gives rise to algebraic structures capturing these aspects of information. The presentation and discussion of these structures is the aim of this survey.

In fact, we propose two different types of algebraic structures, each one emphasizing a different point of view, but shown to be essentially equivalent. So each of these two types may be used at convenience. The first type starts from the idea that each piece of information is associated with a specified question, a question it answers at least partially. The piece of information refers to the associated question by its *label*. Two or more pieces of information can be *combined* into a new piece of information whose label is determined by the labels of the pieces to be combined. This represents aggregation of information. Further a piece of information can be *transported* or *projected* to another question. This models extraction of information. These ideas are captured by algebraic structures with the two basic operations of combination and transport (or projection). Such structures are called *labeled information algebras*. Section 2 is devoted to an introduction and a first discussion of such algebras.

Alternatively, one may also consider pieces of information as abstract entities, unrelated to particular questions. Again such pieces of information can be *combined* to aggregate them into a new piece of information. Further it must be possible to *extract* the part relevant to any given question by extraction operations which associate to a piece of information others which represent its extracted parts. Again, we obtain algebraic structures with two basic operations, namely combination and extraction. Such structures are called *domain-free information algebras*. They are presented in Section 3.

Both types of algebraic structures are intimately related. They represent essentially two different views on the same subject, namely the concept of information. Each point of view has its own merits. The labeled algebras originated as a structure representing a certain computational approach. The domain-free algebras are better suited for theoretical investigations. We shall show that from each of the two types it is possible to pass to the other one and back: They represent two sides of the same coin. According to our purpose, we may thus work with one or the other type at our convenience. The domain-free version of an information algebra is generally more suited for theoretical investigations, while the labeled version is better suited for modeling and computational purposes.

There are many instances or models of information algebras. Some of them will be presented in this paper. Furthermore there are different general, generic ways to construct whole classes of such algebras. In particular, systems of *mathematical logic* provide important sources of information algebras, illustrating the close relationship between logic and information. Information algebras appear in this respect as objects of algebraic logic, adding some new features to this field. This is the subject of Section 4.

The algebraic structures presented here, especially the labeled ones, are based on a proposal for certain axiomatic algebraic structures in [1,2]. These structures, in turn, were proposed with the goal to generalize a successful computation scheme, presented in [3] for Bayesian networks. The algebraic structures proposed in [2] are now called *valuation algebras* [2,4–6]. So, valuation algebras provide a basis for generic, so-called *local computation schemes* [7]. Information algebras, in our sense, add however one important concept to valuation algebras, namely *idempotency*. That is, the combination of a piece of information with itself or part of itself gives nothing new. This allows to introduce an *order* between pieces of information, reflecting their *information content*. Order is an essential element

of the present theory of information algebras, representing a very important feature of any concept of “information”.

With this order structure, information algebras become related to *domain theory*. Domain theory essentially studies information order with the goal to capture some aspects of computation and semantics of programming languages. By introducing the concept of *finite information pieces*, information algebras as ordered structures become in fact *domains*. They are more rich, however, because of the operations of combination and especially information extraction. It turns out that many results and concepts of domain theory carry over to information algebras. This is discussed in Section 5.

In the concluding Section 6, an outlook on subjects not treated in detail in this survey and on open questions in the theory of information algebras is given. So far, there are not too many publications on information algebras available. Most of this survey is based on Chapter 6 in the book [6] and on [8]. A related paper is [9]. So our hope is that this survey may attract some interest to this new and promising theory of important aspects of information.

2. Labeled Algebras

2.1. Modeling Questions

According to the introductory section, we need to consider two types of elements in a labeled information algebra: pieces of information and questions. We start by addressing the concept of questions. First, there are, in general, many questions of interest in a given situation. Further, questions have a certain granularity. A question may be *refined*, made more precise, or on the contrary *coarsened*, made less precise. In addition, a question may be the refinement respectively the coarsening of several other questions. This can be captured by an ordering of questions where $x \leq y$ means that question x is coarser than y or y finer than x . So, we consider a family Q of elements thought to represent questions, and we assume that (Q, \leq) is a partially ordered set, that is, \leq satisfies:

- (1) *Reflexivity*: $x \leq x$ for all $x \in Q$.
- (2) *Antisymmetry*: $x \leq y$ and $y \leq x$ imply $x = y$.
- (3) *Transitivity*: $x \leq y$ and $y \leq z$ imply $x \leq z$.

The following example is based on the idea that a question may be identified with the set of all its possible answers. Let $\{X_i, i \in \mathbf{N}\}$ be a countable set of variables, and fix, for all $i \in \mathbf{N}$, a set of values V_i which X_i may take (the sets V_i need not be distinct). For any $I \subseteq \mathbf{N}$ the set $\prod_{i \in I} V_i$ represents all possible answers to the question “which combinations of values of the variables X_i with $i \in I$ occur?” The elements of $\prod_{i \in I} V_i$, or more precisely, the set of all maps s from I to $\cup_{i \in I} V_i$, satisfying $s(i) \in V_i$ for all $i \in I$, are called I -tuples with components in V_i .

Write Q for the collection of all such questions $\prod_{i \in I} V_i$, $I \subseteq \mathbf{N}$. It is natural to call a question I coarser than a question J , if $I \subseteq J$. This imposes a partial order \leq on Q . In this way, (Q, \leq) becomes a distributive lattice, where the lattice operations of meet (\wedge) and join (\vee) are given by

$$\begin{aligned} \prod_{i \in I} V_i \wedge \prod_{i \in J} V_i &= \prod_{i \in I \cap J} V_i \\ \prod_{i \in I} V_i \vee \prod_{i \in J} V_i &= \prod_{i \in I \cup J} V_i \end{aligned} \tag{1}$$

So (Q, \leq) is isomorphic to the power set $\mathcal{P}(N)$, ordered by set inclusion. A variant of this example is obtained by admitting only *finite* sets $I \subseteq \mathbb{N}$. The result is again a distributive lattice, although without a greatest element. We call this important model for a system of questions a *multivariate system* for further reference.

It is convenient to assume generally that (Q, \leq) is a lattice, although not necessarily a distributive one. The join of two question $x, y \in Q$, written $x \vee y$, is the coarsest question, finer than both x and y . It represents the *combined* question of x and y . The meet, written as $x \wedge y$, is the finest question, coarser than both x and y . It represents the common part of the questions x and y . A nondistributive model for a set of questions is given by the partitions of some universe U . The blocks of a partition \mathcal{P} are the possible answers of the question represented by the partition. A partition \mathcal{P}_1 is finer than a partition \mathcal{P}_2 , if every block of \mathcal{P}_1 is contained in some block of partition \mathcal{P}_2 . We write $\mathcal{P}_2 \leq \mathcal{P}_1$ in this case. Note that this is the reverse of the order usually considered when comparing partitions [10]. However, in our case this order makes sense semantically, since \mathcal{P}_1 has more possible answers than \mathcal{P}_2 , or any answer to \mathcal{P}_1 determines also an answer to \mathcal{P}_2 , but not vice versa. The join of two partitions in this order is defined by

$$\mathcal{P}_1 \vee \mathcal{P}_2 = \{P_1 \cap P_2 \neq \emptyset : P_1 \in \mathcal{P}_1, P_2 \in \mathcal{P}_2\} \tag{2}$$

The meet is a bit more difficult to define, we abstain from giving a definition here, see [10]. The lattice $part(U)$ of partitions of an universe U is not distributive except for trivial cases. In most cases sublattices of the lattice $part(U)$ are considered, which generally are nondistributive too.

Therefore, in the following, we assume (Q, \leq) to be a lattice. According to the examples above, questions are often represented by the set of their possible answers, representing the domain of the information pertaining to it. Therefore we also speak sometimes of domains instead of questions. Next we consider pieces of information in a labeled view, that is, pieces of information referring to specified questions, hence to elements of Q .

2.2. Labeled Information

Let Ψ be a set of elements, thought to represent labeled pieces of information, generic elements denoted by small Greek letters ψ, ϕ, \dots . We assume that each of these pieces of information pertains to a unique questions in a lattice (Q, \leq) whose elements will be denoted by lower case letters like x, y, \dots . That is, each element of Ψ is labeled with the question it refers to. Pieces of information can be combined to give new aggregated pieces of information. Further any piece of information can be projected to a question coarser than the question it refers to. More formally, three operations are defined in the pair (Ψ, Q) :

- (1) *Labeling*: $d : \Psi \rightarrow Q; \psi \mapsto d(\psi)$.

(2) *Combination*: $\cdot : \Psi \times \Psi \rightarrow \Psi; \psi, \phi \mapsto \psi \cdot \phi$.

(3) *Projection*: $\pi : \Psi \times Q \rightarrow \Psi; \psi, x \mapsto \pi_x(\psi)$, defined for $x \leq d(\psi)$.

The labeling operation retrieves the question $d(\psi) \in Q$ the information ψ pertains to. The combination operation returns the aggregated piece of information $\psi \cdot \phi$ of the two element ψ and ϕ . Projection finally represents the operation of extracting that part of information which pertains to a coarser question than the original one. It is a *partial* operation, defined only for pairs ψ, x such that $x \leq d(\psi)$. Thus, we have a two sorted algebra $(\Psi, Q; d, \cdot, \pi)$ with a unary and two binary operations, one of them partial. We collect the elements of Ψ having the same label into sets $\Psi_x = \{\psi \in \Psi : d(\psi) = x\}$ for every element $x \in Q$. Thus

$$\Psi = \bigcup_{x \in Q} \Psi_x \tag{3}$$

We impose the following axioms:

(1) *Semigroup*: $(\Psi; \cdot)$ is a commutative semigroup

(2) *Labeling*: For all $\psi, \phi \in \Psi$ and $x \in Q$ such that $x \leq d(\psi)$,

$$d(\psi \cdot \phi) = d(\psi) \vee d(\phi), \quad d(\pi_x(\psi)) = x$$

(3) *Unit and Null*: For all $x \in Q$ there exist unit and null elements 1_x and 0_x in Ψ_x such that $1_x \cdot \psi = \psi \cdot 1_x = \psi$ and $0_x \cdot \psi = \psi \cdot 0_x = 0_x$ for all ψ with $d(\psi) = x$. Further, for all $y \leq x$, $\pi_y(1_x) = 1_y$ and $0_y \cdot 1_x = 0_x$.

(4) *Projection*: For all $\psi \in \Psi$ and $x, y \in Q$ such that $x \leq y \leq d(\psi)$,

$$\pi_x(\pi_y(\psi)) = \pi_x(\psi)$$

(5) *Combination*: For all $\psi, \phi \in \Psi$, if $d(\psi) = x$ and $d(\phi) = y$, then

$$\pi_x(\psi \cdot \phi) = \psi \cdot \pi_{x \wedge y}(\phi)$$

(6) *Idempotency*: For all $\psi \in \Psi$ and $x \in Q$ such that $x \leq d(\psi)$,

$$\psi \cdot \pi_x(\psi) = \psi$$

An algebra $(\Psi, Q; d, \cdot, \pi)$ satisfying these axioms is called a *labeled information algebra*.

The *Semigroup* axiom states that information can be combined in any order without changing the resulting information. As a result, we do not need to write parentheses when combining several pieces of information. The *Labeling* axiom says that the combined information refers to the combined question of the questions of the factors to be combined. Similarly, if a piece of information is projected to a coarser question, the result refers to this coarser question. The axiom of *Unit and Null* means, together with the semigroup and labeling axioms, that all Ψ_x are subsemigroups of Ψ , each having unit and null elements. Unit elements represent *vacuous* information with respect to the corresponding questions: Combining

with whatever other information relating to the same question does not change this later information. Unit elements project to unit elements. Null elements represent contradiction, destroying any other information. According to the *Projection* axiom a projection may be carried out in two (or several) steps. The most powerful axiom is *Combination*. It permits to avoid large domains in combination, if a combination is followed by a projection. This is important for computations, but not exclusively. *Idempotency* finally tells us that to combine a piece of information with a part (a projection) of itself gives nothing new. This important property of information allows to order pieces of information, and permits to develop an order theory for information algebra, as we shall see later.

As an example, let us consider the multivariate model of questions introduced above. The lattice of questions is given by the lattice of subsets of variables. A piece of information R with label $d(R) = I$ is in this model an arbitrary set of I -tuples, with I finite (see Section 2.1). Note the similarity to relational algebra of relational database theory. A subset of I -tuples is called a *relation* over I there. So the elements of Ψ are relations over sets of variables I , or more precisely, pairs $\psi = (R, I)$, where R is a relation over I . The label of a relation ψ over I is just I , $d(\psi) = I$. Combination is essentially *relational join*. In order to define this operation define the label of an I -tuple s to be $d(s) = I$ and the projection π_J of I -tuples to J -tuples with $J \subseteq I$ to be the restriction of the map $s : I \rightarrow V$ to J , that is, $\pi_J(s)(j) = s(j)$ for $j \in J$. The relational join of two relations R and S over I and J respectively is then defined as

$$R \bowtie S = \{s : d(s) = I \cup J, \pi_I(s) \in R, \pi_J(s) \in S\} \tag{4}$$

Then, if $\psi = (R, I)$ and $\phi = (S, J)$, we define $\psi \cdot \phi = (R \bowtie S, I \cup J)$. This clearly defines a commutative semigroup; the null element in domain I is the pair (\emptyset, I) , and the unity element is the pair (T_I, I) , where T_I denotes the set of all I -tuples. Projection is based on *relational projection*. If R is a relation over I and $J \subseteq I$, then relational projection is defined as

$$\pi_J(R) = \{\pi_J(s) : s \in R\} \tag{5}$$

Projection in Ψ is then defined for $\psi = (R, I)$ by $\pi_J(\psi) = (\pi_J(R), J)$. It can be verified that $(\Psi, Q; d, \cdot, \pi)$ with labeling, combination and projection as just defined satisfies the axioms of a labeled information algebra. If we restrict ourselves to finite subsets of variables, then we still obtain a labeled information algebra, in fact, a subalgebra of the original one.

The pair notation is cumbersome, it is only necessary to distinguish the empty relations of no I -tuples. Once this is understood, we may drop the pair-notation and just work with relations R, S, \dots , their domains being implicitly defined. In this case we denote the pair (\emptyset, I) simply by \emptyset_I and call it the empty set on domain I . We refer to this example as a *relational information algebra*. Many labeled information algebras are *subalgebras* of a relational information algebra, especially with variables taking real numbers as values, see [6], as for instance the information algebras of convex sets, convex polyhedra or affine subspaces of linear spaces.

As a second example, let us consider the lattice (Q, \leq) of partitions of an universe U , discussed in the previous subsection. Pieces of information refer to a partition \mathcal{P} as label there. Similarly as for relational information algebras, we take as elements of Ψ pairs $\psi = (S, \mathcal{P})$, where S is a subset of blocks of partition \mathcal{P} . The label $d(\psi)$ of the piece of information ψ is \mathcal{P} . The idea here is that an unknown element of the universe U represents the state of a system or the actual world out of a set of possible worlds.

A partition \mathcal{P} represents a question about the unknown state. A set of blocks S is a (partial) answer to the question which block P of the partition \mathcal{P} contains the unknown state. In this sense ψ is a piece of information about the question represented by the partition \mathcal{P} . Such situations arise often in expert systems, especially diagnostics systems, where faults for instance may be decomposed or refined into finer descriptions. The partitions of an universe U form a lattice $Part(U)$ as we have seen in Section 2.1. However we consider only sublattices \mathcal{S} of $Part(U)$ satisfying an additional condition. For two elements $m, n \in U$ we write $m \equiv_{\mathcal{P}} n$ if they belong to the same block of the partition \mathcal{P} . Then we require, for any pair of partitions \mathcal{P} and \mathcal{Q} in the sublattice \mathcal{S} , that

$$m \equiv_{\mathcal{P} \wedge \mathcal{Q}} n \Rightarrow \exists l : m \equiv_{\mathcal{P}} l, l \equiv_{\mathcal{Q}} n \tag{6}$$

This condition is necessary and sufficient for a subset algebra to become an information algebra (see below). It is clear that a partition lattice in general does not satisfy this condition.

Let us now Φ denote the family of subsets of blocks of partitions of a sublattice \mathcal{S} satisfying Equation (6). Then we define combination and projection similarly as in the case of a relational algebra: If \mathcal{Q} is a partition coarser as \mathcal{P} , and P a block of \mathcal{P} , then define $\pi_{\mathcal{Q}}(P) = Q$, where Q is the block of \mathcal{Q} which contains P . The label $d(P)$ of a block of a partition \mathcal{P} is just \mathcal{P} . Then combination of two pieces of information $\psi = (S, \mathcal{P})$ and $\phi = (T, \mathcal{Q})$ is based on a kind of generalized relational join,

$$S \bowtie T = \{P : d(P) = \mathcal{P} \vee \mathcal{Q}, \pi_{\mathcal{P}}(P) \in S, \pi_{\mathcal{Q}}(P) \in T\} \tag{7}$$

by $\psi \cdot \phi = (S \bowtie T, \mathcal{P} \vee \mathcal{Q})$. Projection of a piece of information $\psi = (S, \mathcal{P})$ to a partition \mathcal{Q} coarser than \mathcal{P} is based on a generalization of relational projection:

$$\pi_{\mathcal{Q}}(S) = \{\pi_{\mathcal{Q}}(P) : P \in S\} \tag{8}$$

by $\pi_{\mathcal{Q}}(\psi) = (\pi_{\mathcal{Q}}(S), \mathcal{Q})$. Condition in Equation (6) is sufficient to guarantee that the structure $(\Phi, \mathcal{S}; \cdot, \circ)$ is a labeled information algebra. It is called a *subset information algebra*. Note that in a general lattice of partitions this is not true. Condition in Equation (6) is sufficient, and in fact also necessary, for $(\Phi, \mathcal{S}; \cdot, \circ)$ to satisfy the axioms of a labeled information algebra. We shall also see later that it is no coincidence that this example resembles relational information algebras so much (see Section 4.1).

To complete the picture let's consider a distributive lattice (L, \leq_L) with least element 0_L and greatest element 1_L . Meet and join in L are denoted by \wedge_L and \vee_L . Further consider any index set I and an associated indexed family of finite sets Ω_i for $i \in I$. For any finite subset s of I form the Cartesian product $\Omega_s = \prod_{i \in s} \Omega_i$. The generic elements of Ω_s are denoted by x, y, \dots . These elements can also be considered as maps x from s into Ω_s such that $x(i) \in \Omega_i$, that is, as s -tuples. If $t \subseteq s$, then let $\pi_t(x)$ denote the restriction of x to t . Let $(Q, \leq) = (\mathcal{P}(I), \subseteq)$ be the lattice of all subsets of I ordered under inclusion.

Lattice valuations on s are maps $\psi : \Omega_s \rightarrow L$. Denote the set of all such lattice valuations on s by Ψ_s and define

$$\Psi = \bigcup_{s \subseteq I} \Psi_s \tag{9}$$

Consider the following operations:

- (1) *Labeling*: For $\psi \in \Psi_s$, let $d(\psi) = s$.

(2) *Combination*: For $\psi, \phi \in \Psi$ with $d(\psi) = s, d(\phi) = t$, combination $\psi \cdot \phi$ is defined pointwise for all $x \in \Omega_{s \cup t}$ as

$$(\psi \cdot \phi)(x) = \psi(\pi_s(x)) \wedge_L \phi(\pi_t(x))$$

(3) *Projection*: For $\psi \in \Psi$ with $d(\psi) = s$ and $t \subseteq s$, the projection $\pi_t(\psi)$ is defined for all $x \in \Omega_t$ as

$$\pi_t(\psi)(x) = \vee_L \{ \psi(z) : z \in \Omega_s, \pi_t(z) = x \}$$

This defines a labeled information algebra, see [11]. The vacuous information on domain s is the lattice valuation $1_s(x) = 1_L$, the null element $0_s(x) = 0_L$ for all $x \in \Omega_s$. This is called a *Lattice Induced Information Algebra*. For the choice of L we have many possibilities. For instance we may take the Boolean lattice $\{0, 1\}$ with the natural order $0 \leq 1$. Here lattice valuations represent indicator functions. So, this gives an algebra similar to a subset information algebra. Secondly, we may take for L the real interval $[0, 1]$ with its natural order. Meet is then minimum and join maximum. Lattice valuations here can be thought to represent fuzzy sets or possibility distributions. Combination becomes

$$(\psi \cdot \phi)(x) = \min\{\psi(\pi_s(x)), \phi(\pi_t(x))\} \tag{10}$$

and projection is defined as

$$\pi_t(\psi)(x) = \max\{\psi(z) : z \in \Omega_s, \pi_t(z) = x\} \tag{11}$$

This corresponds to widely used definitions of fuzzy set intersection and fuzzy set projection. These are examples of generic constructions of labeled information algebras. We shall see more such generic constructions later (Section 4).

2.3. Transport of Information

The projection or extraction operator of a labeled information algebra is only defined as a partial map. We shall now extend it to a total map in two steps. So let $(\Psi, Q; d, \cdot, \pi)$ be a labeled information algebra. Consider an element $\psi \in \Psi$ and a question $x \geq d(\psi) = y$. Define $e_x(\psi) = \psi \cdot 1_x$. This is an information element on domain x (according to the Labeling axiom). If we project it back to the original domain y of ψ , we obtain by the Combination and Unit axioms that $\pi_y(\psi \cdot 1_x) = \psi \cdot \pi_y(1_x) = \psi \cdot 1_y = \psi$. So the extension of ψ to domain x by $e_x(\psi)$ can be inverted without loss of information. Also the combination of ψ with the vacuous information 1_x does only change the domain, but does not add information. Therefore the extension operation e_x is called *vacuous extension*. It still is a partial operation.

Still assume $d(\psi) = y$, but consider now an arbitrary element $x \in Q$. Then we define a transport operation t_x by

$$t_x(\psi) = \pi_x(e_{x \vee y}(\psi)) \tag{12}$$

That is, we extend ψ to the larger domain $x \vee y$ and then project the result back to domain x . Clearly this operation extracts the information pertaining to question x from ψ . The map $t : \psi \times Q \rightarrow \Psi$ defined by $\psi, x \mapsto t_x(\psi)$ is called the *transport operation*, since it transports ψ to domain x . It is a total map. Evidently, if $x \leq y = d(\psi)$, t_x reduces to the projection π . So it is indeed an extension of the partial map

π . If $x \geq d(\psi) = y$, then it reduces to vacuous extension; it is therefore also an extension of vacuous extension.

There are alternative, equivalent ways to define transport. Using the axioms of a labeled information algebra, it follows (see [6]) that for any $z \geq x \vee y$,

$$t_x(\psi) = \pi_x(e_z(\psi)) \tag{13}$$

and also

$$t_x(\psi) = e_x(\pi_{x \wedge y}(\psi)) \tag{14}$$

For the sake of completeness we mention a few other results on transport, which are easy to derive [6]

- (1) If $d(\psi) = x$, then $t_x(\psi) = \psi$,
- (2) If $y \leq z$, then $t_y(\psi) = t_y(t_z(\psi))$,
- (3) $t_y(t_x(\psi)) = t_y(t_{x \wedge y}(\psi))$,
- (4) $t_x(t_x(\psi) \cdot \phi) = t_x(\psi) \cdot t_x(\phi)$.

A piece of information ψ and its vacuous extension $e_x(\psi)$ represent in some way the *same* information, although with reference to different questions. More generally, if ψ and ϕ are two elements of Ψ with domains $d(\psi) = x$ and $d(\phi) = y$ and $t_y(\psi) = \phi$ and $t_x(\phi) = \psi$, then we may shuffle these elements around between the domains x and y , without any loss of information. Again, ψ and ϕ represent the *same* information, although each with reference to a different question. This observation can be captured by defining the following relation \equiv_σ :

$$\psi \equiv_\sigma \phi, \text{ if } d(\psi) = x \text{ and } d(\phi) = y \text{ together imply } t_y(\psi) = \phi \text{ and } t_x(\phi) = \psi.$$

This is an *equivalence relation* in Ψ . It is even more, namely a congruence with respect to combination and transport (see [6]). That means that if $\psi_1 \equiv_\sigma \phi_1$ and $\psi_2 \equiv_\sigma \phi_2$, then $\psi_1 \cdot \psi_2 \equiv_\sigma \phi_1 \cdot \phi_2$ and if $\psi \equiv_\sigma \phi$, then $t_x(\psi) \equiv_\sigma t_x(\phi)$ for any $x \in Q$. Furthermore $0_x \equiv_\sigma 0_y$ and $1_x \equiv_\sigma 1_y$ for all $x, y \in Q$. This allows us to consider the quotient algebra Ψ/σ , consisting of the equivalence classes $[\psi]_\sigma$ for all $\psi \in \Psi$. Then, since \equiv_σ is a congruence, the following operations between equivalence classes are all well-defined

$$[\psi]_\sigma \cdot [\phi]_\sigma = [\psi \cdot \phi]_\sigma \tag{15}$$

$$\epsilon_x([\psi]_\sigma) = [t_x(\psi)]_\sigma \tag{16}$$

So, Ψ/σ becomes a new type of algebra, again with a combination and an extraction operation, but this time applied to equivalence classes and not to individual elements of Ψ . We note that Ψ/σ is a commutative and idempotent semigroup under combination. The ϵ_x are mappings of Ψ/σ into itself for every $x \in Q$. Under composition these maps form also a commutative and idempotent semigroup, $\epsilon_x \circ \epsilon_y = \epsilon_y \circ \epsilon_x$ and $\epsilon_x \circ \epsilon_x = \epsilon_x$. Further, the maps ϵ_x have the interesting properties that (1) $\epsilon_x([1_y]_\sigma) = [1_y]_\sigma$, further (2) $[\psi]_\sigma \cdot \epsilon_x([\psi]_\sigma) = [\psi]_\sigma$ and finally (3) $\epsilon_x(\epsilon_x([\psi]_\sigma) \cdot [\phi]_\sigma) = \epsilon_x([\psi]_\sigma) \cdot \epsilon_x([\phi]_\sigma)$. This indicates an algebraic structure which is of interest in itself. It will be formalized and studied in the next section.

3. Domain-free Algebras

3.1. Another View

In this section a quite different view of an information algebra will be presented. It is motivated by the remarks at the end of the previous section. This time we start from a set Φ whose elements are thought of as pieces of information, considered as abstract entities independent of any domain or question. That is, the elements of Φ represent *domain-free* information. The generic elements of Φ are again denoted by lower case Greek letters like ϕ, ψ, \dots . And again the operation of *combination* is defined between elements of Φ :

$$\text{Combination: } \cdot : \Phi \times \Phi \rightarrow \Phi, \phi, \psi \mapsto \phi \cdot \psi.$$

As before, combination expresses the idea of aggregation of pieces of information, resulting in a new piece of information. Also, there is a unit or neutral element 1 for combination such that $1 \cdot \phi = \phi \cdot 1 = \phi$ for all $\phi \in \Phi$. It represents *vacuous information*. Further there is a null element 0 such that $0 \cdot \phi = \phi \cdot 0 = 0$. It represents *contradiction*, which destroys any information. Strictly speaking it does not really represent information as the other elements of Φ do, but it stands for the possibility that pieces of information may be contradictory.

Further we assume the existence of a family E of extraction operators, represented by mappings of Φ into itself:

$$\text{Extraction: } \epsilon : \Phi \rightarrow \Phi.$$

We may think of each element of E as an operation to extract information from the elements of Φ with respect to some question. This point of view will be clarified below. Guided by the algebraic structure derived from a labeled information algebra at the end of the previous section, we impose the following conditions on Φ , E and the interaction of combination and extraction:

1. *Combination:* Φ is a commutative and idempotent semigroup under the combination operation \cdot with a unit 1 and a null element 0.
2. *Extraction:* Each extraction operator $\epsilon \in E$ satisfies the following conditions:
 - (a) $\epsilon(0) = 0$,
 - (b) $\phi \cdot \epsilon(\phi) = \phi$ for all $\phi \in \Phi$,
 - (c) $\epsilon(\epsilon(\phi) \cdot \psi) = \epsilon(\phi) \cdot \epsilon(\psi)$ for all $\phi, \psi \in \Phi$.
3. *Composition:* E is a commutative and idempotent semigroup under ordinary composition of mappings.

An algebra $(\Phi, E; \cdot, \circ)$ satisfying these conditions is called a *Domain-Free Information Algebra*.

Item 1 states that combination of pieces of information can be done in any order with the same result. As stated above, the unit element 1 represents vacuous information and the null element 0 contradiction. Item 2 (a) says that extraction of information from the vacuous information results simply in vacuous information, there is no information to extract from vacuous information. Further, item 2 (b) requires that combining a piece of information with a part extracted from it gives nothing new, preserving

the original piece of information. This corresponds to the Idempotency axiom of labeled algebras. Item 2 (c) corresponds to the *Combination axiom* of labeled information algebras. Finally, item 3 says that if information is extracted with respect to two questions, the order in which this is done is irrelevant. And once information is extracted with respect to a fixed question, extracting information with respect to the same question gives nothing new.

As discussed at the end of the previous section, any labeled information algebra gives rise to a domain-free information algebra, if equivalent labeled information elements representing the same information are collected into equivalence classes. Combination is inherited from labeled combination and extraction from the transport operation. This hints already at a close connection between labeled and domain-free algebras. We shall in fact show that they are essentially two sides of a same coin and that we may switch between the two types of algebraic structures at our convenience.

If $(\Psi, Q; d, \cdot, \pi)$ is a labeled information algebra and the lattice of questions has a top element \top , that is a finest, universal question, then any element $\psi \in \Psi$ is equivalent to its vacuous extension to the top domain, $\psi \equiv_{\sigma} e_{\top}(\psi)$. In the quotient algebra Ψ/σ , we may therefore always take $e_{\top}(\psi)$ as a representative of the class $[\psi]_{\sigma}$. Since we then have

$$[e_{\top}(\psi)]_{\sigma} \cdot [e_{\top}(\phi)]_{\sigma} = [e_{\top}(\psi) \cdot e_{\top}(\phi)]_{\sigma} \tag{17}$$

$$\epsilon_x([e_{\top}(\psi)]_{\sigma}) = [e_{\top}(\pi_x(e_{\top}(\psi)))]_{\sigma} \tag{18}$$

we see that we may represent the domain-free version of the labeled algebra $(\Psi, Q; d, \cdot, \pi)$ essentially by the elements $e_{\top}(\psi)$ of the labeled algebra. So, e.g., in the case of a relational information algebra (see previous section), we may adjoin to the lattice Q of the finite subsets of \mathbf{N} the whole index set \mathbf{N} as top element. Let $f : \{1, 2, \dots\} \rightarrow \cup_{i \in \mathbf{N}} V_i$ be sequences of elements $f(i)$ with $f(i) \in V_i$ and define $\pi_I(f)$ to be the restriction of f to a subset I of variables. Then the vacuous extension of a relation R on I is its so-called cylindrical extension or saturation, defined by

$$e_{\top}(R) = \{f : \pi_I(f) \in R\} \tag{19}$$

Thus, the sets $e_{\top}(R)$ are subsets of $\prod_{i \in \mathbf{N}} V_i$, the set of all infinite sequences f . Hence, the domain-free version of a relational information algebra is essentially the algebra of subsets $e_{\top}(R)$ with intersection as combination and $\epsilon_I(e_{\top}(R)) = e_{\top}(\pi_I(e_{\top}(R)))$ as extraction.

To complete the picture let us present two further, simple examples of domain-free information algebras. Consider a finite alphabet Σ , the set Σ^* of finite strings over Σ , including the empty string ϵ , and the set Σ^{ω} of countably infinite strings over Σ . Let $\Sigma^{**} = \Sigma^* \cup \Sigma^{\omega} \cup \{0\}$, where 0 is a symbol not contained in Σ . For two strings $r, s \in \Sigma^{**}$, define $r \leq s$, if r is a prefix of s or if $s = 0$. Then, we define a combination operation in Σ^{**} as follows:

$$r \cdot s = \begin{cases} s, & \text{if } r \leq s, \\ r & \text{if } s \leq r, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Clearly, this is a commutative, idempotent semigroup. The empty string ϵ is the unit element, and the adjoined element 0 is the null element of combination. For extraction, we define operators ϵ_n for any $n \in \mathbf{N}$ and also for $n = \infty$. Let $\epsilon_n(s)$ be the prefix of length n of string s , if the length of s is at least

n , and let $\epsilon_n(s) = s$ otherwise. In particular, define $\epsilon_\infty(s) = s$ for any string s and $\epsilon_n(0) = 0$ for any n . So, any ϵ_n maps Σ^{**} into itself. It is easy to verify the each ϵ_n satisfies conditions (a) to (c) of item 2 above in the definition of a domain-free information algebra. Further, it can also be easily verified, that $E = \{\epsilon_n : n \in \mathbb{N} \cup \{\infty\}\}$ is a commutative and idempotent semigroup under composition of maps. So, the *string algebra* $(\Sigma^{**}, E, \cdot, \circ)$ is an instance of a domain-free information algebra.

As another example consider any set X . A *partial map* on X is a map $\sigma : S \rightarrow X$, where S is a subset of X . We denote the domain of σ by $d(\sigma)$. The set of all partial maps on X , including an adjoined new element 0 , is denoted by $X \rightarrow \circ \rightarrow X$. Partial maps are ordered by $\sigma \leq \tau$, if $d(\sigma) \subseteq d(\tau)$ and $\sigma(x) = \tau(x)$ for all $x \in d(\sigma)$ or if $\tau = 0$. We define combination in $X \rightarrow \circ \rightarrow X$ by

$$\sigma \cdot \tau = \begin{cases} \sigma & \text{if } \tau \leq \sigma, \\ \tau & \text{if } \sigma \leq \tau, \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Again, under this operation, $X \rightarrow \circ \rightarrow X$ is a commutative and idempotent semigroup, with the empty function ϵ , defined on the empty set, as unit, and 0 as null element. For any subset S of X , we define an extraction operation ϵ_S by restricting a map σ to S , more precisely,

$$\epsilon_S(\sigma) = \sigma|_{d(\sigma) \cap S} \tag{22}$$

and $\epsilon_S(0) = 0$. Each of these maps satisfies conditions (a) to (c) in item 2 of the definition of a domain-free information algebra. Further, $E = \{\epsilon_S : S \subseteq X\}$ is a commutative and idempotent semigroup under composition. Thus the *algebra of partial maps* $(X \rightarrow \circ \rightarrow X, E, \cdot, \circ)$ is a domain-free information algebra.

Further examples will be given later, in particular in Section 4. Here we shall first show that idempotency of Φ and E permits to define an order in each of these two sets. Especially important is the order of information in Φ . Many interesting features of an information algebra are in fact order-theoretic, as we shall see subsequently and in particular in Section 5. Order will be introduced in Section 3.2 and exploited in Section 3.3. Then, just as labeled information algebras induce associated domain-free algebras, we shall show in Section 3.4 that from a domain-free algebra we may also construct a labeled one. Finally, a further important structural property will be introduced and discussed in Section 3.5.

3.2. Order

Can pieces of information be compared as to their information content? Suppose $(\Phi, E; \cdot, \circ)$ to be a domain-free information algebra. If ϕ, ψ are two pieces of information such that $\phi \cdot \psi = \phi$, then we may say that ψ contains less information than ϕ , since knowing ϕ and ψ gives nothing more than ϕ . The piece of information ψ adds nothing to ϕ . So, in this case we write $\psi \leq \phi$, meaning that ψ contains less information than ϕ , is less informative than ϕ . This relation is indeed a *partial order* on Φ . This order has a number of simple consequences. So, $\phi, \psi \leq \phi \cdot \psi$. In fact, the combination of ϕ and ψ is the least upper bound, the *supremum* of the two elements, $\phi \cdot \psi = \sup\{\phi, \psi\}$. Further, 0 is an upper bound and 1 a lower bound for any element in Φ , $1 \leq \phi \leq 0$. This means that (Φ, \leq) is a *join-semilattice* with minimal and maximal elements 1 and 0 . We often write combination as *join*, when the order-theoretic

point of view is to be stressed, $\phi \cdot \psi = \phi \vee \psi$. We mention also that in many cases, as for example in the string algebra and the algebra of partial maps, combination is defined primarily by order.

In any commutative and idempotent semigroup there exist exactly two orders compatible with the semigroup operation. Here, we have selected the one which represents the information order. This choice implies that $1 \leq \phi \leq 0$; the vacuous information is less informative than any other information, and the contradiction 0, which is not really an information, dominates in this order any piece of information.

In the same way we may define an order in the commutative, idempotent semigroup E of extraction operators: Here we define $\epsilon \leq \eta$ if $\eta \circ \epsilon = \epsilon$. So, $\epsilon \leq \eta$ means that once information is extracted according to ϵ , another extraction according to η does no more change anything. Under this order, composition of extraction corresponds to the infimum, $\eta \circ \epsilon = \inf\{\eta, \epsilon\}$. In this way, E becomes a *meet-semilattice*, and composition can be written as meet, $\eta \circ \epsilon = \eta \wedge \epsilon$. In the examples presented above, we have seen that the elements of E are often indexed by another set Q , $E = \{\epsilon_x : x \in Q\}$. We may consider the elements x of Q as representing questions, and the maps ϵ_x as extraction operators, which extract the information pertaining to question x from the pieces of information. The order in E induces an order in Q : $x \leq y$ in Q , if $\epsilon_x \leq \epsilon_y$ in E . Here $x \leq y$ means that the question represented by x is coarser than the one represented by y ; once the information relating to x is extracted, the information relating to the finer question y is lost and cannot be retrieved.

In this view, in domain-free information algebras, the system Q of questions is only assumed to be a meet-semilattice, whereas in labeled information algebras Q is assumed to be a lattice. This is a difference between the domain-free and the labeled view of information. In many cases however, even in domain-free information algebras, Q is in fact also a lattice, see the examples above.

We adopt in the sequel the view that the family of extraction operators E is indexed by the elements of a question semilattice Q . Here, we give a few elementary results on the interplay between the order in Φ and the one in Q in a domain-free information algebra:

- (1) $\epsilon_x(1) = 1$ for all $x \in Q$,
- (2) $x \leq y$ implies $\epsilon_x(\phi) \leq \epsilon_y(\phi)$ for all $\phi \in \Phi$,
- (3) $\phi \leq \psi$ implies $\epsilon_x(\phi) \leq \epsilon_x(\psi)$ for all $x \in Q$,
- (4) $\epsilon_x(\phi) \leq \phi$ for all $\phi \in \Phi$ and all $x \in Q$,
- (5) $\epsilon_x(\phi) \cdot \epsilon_x(\psi) \leq \epsilon_x(\phi \cdot \psi)$.

In particular, conditions (a) to (c) in item 2 of the definition of a domain-free information algebra above may be rewritten as follows:

- (a) $\epsilon_x(0) = 0$,
- (b) $\epsilon_x(\phi) \leq \phi$ for all $\phi \in \Phi$,
- (c) $\epsilon_x(\epsilon_x(\phi) \vee \psi) = \epsilon_x(\phi) \vee \epsilon_x(\psi)$.

An operator on a semilattice, satisfying these conditions, is also called an *existential quantifier* in algebraic logic, see [12,13], although often the reversed order is used. This difference is explained by our choice of the order, dictated by the semantics of information. This is a first reference to the close

connection between information algebras and logic, in particular algebraic logic. This subject will be taken up in the next subsection and in particular in Section 4.

To conclude our discussion of order, let's mention that in just the same way an order may also be introduced between pieces of information in labeled information algebras. We take this for granted. This section can serve as a model how to do it. Many other concepts related to order, for instance ideal completion (Section 3.3) or compactness (Section 5) have their counterpart in the labeled version.

3.3. Ideal Completion

If an agent asserts a piece of information ϕ , then he should consistently also assert any lesser informative information $\psi \leq \phi$. So, the down set $\downarrow \phi = \{\psi : \psi \leq \phi\}$ is then a consistent body of information deduced from ϕ . More generally, if an agent asserts two pieces of information ϕ and ψ , then he should also consistently assert their combination $\phi \cdot \psi$. In other words, a consistent body of information should form an ideal in Φ . Assume $(\Phi, E; \cdot, \circ)$ is a domain-free information algebra. An ideal in Φ is then defined as follows:

A non-empty set $I \subseteq \Phi$ is called an *ideal* in Φ , if

- (1) $\phi \in I, \psi \in \Phi$ and $\psi \leq \phi$ jointly imply $\psi \in I$,
- (2) $\phi, \psi \in I$ imply $\phi \cdot \psi \in I$.

This corresponds to the usual concept of an ideal in semilattices or lattices, see [14]. In our context, an ideal represents a coherent and consistent *theory*. The down set $\downarrow \phi$ of an element ϕ of Φ is called a *principal ideal*, and if an ideal I is different from Φ it is called *proper*. Let I_Φ denote the set of all ideals of Φ . We show that these ideals form an information algebra.

For this purpose we consider ideals I of Φ as pieces of information and define combination of two ideal I_1 and I_2 by

$$I_1 \cdot I_2 = \{\phi \in \Phi : \phi \leq \phi_1 \cdot \phi_2 \text{ for some } \phi_1 \in I_1, \phi_2 \in I_2\} \tag{23}$$

Clearly, $I_1 \cdot I_2$ is an ideal in Φ , so combination is well defined. It is commutative and idempotent. The ideal $\{1\}$ is the unit element and the ideal Φ the null element of this combination operation.

Next, assume extraction operators in E to be indexed by elements $x \in Q$, where Q is a meet-semilattice. Define for $x \in Q$ and an ideal I of Φ

$$\bar{\epsilon}_x(I) = \{\phi \in \Phi : \phi \leq \epsilon_x(\psi) \text{ for some } \psi \in I\} \tag{24}$$

Again, $\bar{\epsilon}_x(I)$ is an ideal in Φ , so $\bar{\epsilon}_x(I)$ is well defined. It can be shown that $\bar{E} = \{\bar{\epsilon}_x : x \in Q\}$ is a commutative and idempotent semigroup under composition. It can also be shown that $\bar{\epsilon}_x$ as a map from the set of ideals I_Φ into itself satisfies conditions (a) to (c) of item 2 in the definition of a domain-free information algebra, that is, it is an existential quantifier, see [6,8]. Therefore, $(I_\Phi, \bar{E}, \cdot, \circ)$ is a domain-free information algebra. It is called the *ideal extension* of the information algebra $(\Phi, E; \cdot, \circ)$. Note that in the information order we have $I_1 \leq I_2$ exactly if $I_1 \subseteq I_2$. Thus I_2 is the larger theory than I_1 , hence more informative!

The original algebra $(\Phi, E; \cdot, \circ)$ is embedded into its ideal extension by the map $\phi \mapsto \downarrow \phi$, because $\downarrow(\phi \cdot \psi) = \downarrow \phi \cdot \downarrow \psi$, $\downarrow \epsilon_x(\phi) = \bar{\epsilon}_x(\downarrow \phi)$ and the map is one-to-one. To simplify notation we identify

henceforth Φ with its image under this map and also write ϵ_x instead of $\bar{\epsilon}_x$. Also $\phi \in I$ is often written as $\phi \leq I$, referring to the order in I_Φ .

The intersection of any family of ideals is again an ideal. So the collection of all ideals is a \cap -system; and any \cap -system is a complete lattice under set-inclusion [14]. Meet is just set-interesection and join is defined by ideal-combination. So, I_Φ is not just a join-semilattice under information order, but a complete lattice, where meet and join are defined for any family of ideals. The role of ideal completion will become clearer in Section 5.

3.4. Support

We claimed that an extraction operation ϵ_x extracts exactly the information pertaining to a question $x \in Q$ from any piece of information $\phi \in \Phi$. Now, it may be that a piece of information $\phi \in \Phi$ just pertains fully to a question x , that is extracting information relative to x does not change ϕ , $\epsilon_x(\phi) = \phi$. Then $x \in Q$ is called a *support* of ϕ . Note that, in general, not every element $\phi \in \Phi$ has a support. However, the identity map id of Φ into itself satisfies condition (a) to (c) of item 2 in the definition of a domain-free information algebra, so $E \cup \{id\}$ is still a commutative, idempotent semigroup, containing E as a sub-semigroup. So, if $(\Phi, E; \cdot, \circ)$ is a domain-free information algebra, then so is $(\Phi, E \cup \{id\}, \cdot, \circ)$. However, in this algebra every element of Φ has a support, namely id . To the extraction operator id we may adjoin a new top element \top in Q , satisfying $id = \epsilon_\top$. We mentioned already above that such a top element represents a finest question, which contains all possible answers. So, in this way, we may easily extend any domain-free information algebra in such a way that any element of Φ has a support. An algebra where each element has a support is called *supported*.

We list a few results concerning support:

- (1) $\epsilon_x(\epsilon_x(\phi)) = \epsilon_x(\phi)$, that is, x is a support of $\epsilon_x(\phi)$,
- (2) $\epsilon_x(\phi) = \phi$ implies $\epsilon_x(\epsilon_y(\phi)) = \epsilon_y(\phi)$, that is, if x is a support of ϕ , then x is a also a support of any $\epsilon_y(\phi)$,
- (3) $\epsilon_x(\phi) = \epsilon_y(\phi) = \phi$ implies $\epsilon_{x \wedge y}(\phi) = \phi$, that is, if x and y are supports of ϕ , then so is $x \wedge y$,
- (4) $\epsilon_x(\phi) = \phi$ implies $\epsilon_{x \wedge y}(\phi) = \epsilon_y(\phi)$, that is, if x is a support of ϕ , then $x \wedge y$ is a support of $\epsilon_y(\phi)$,
- (5) $\epsilon_x(\phi) = \phi$ and $x \leq y$ imply $\epsilon_y(\phi) = \phi$, that is, if x is a support ϕ , then any finer question y is a support of ϕ too,
- (6) $\epsilon_x(\phi) = \phi$ and $\epsilon_x(\psi) = \psi$ imply $\epsilon_x(\phi \cdot \psi) = \phi \cdot \psi$, that is, if x is a support of ϕ and ψ , then it is a support of $\phi \cdot \psi$ too,
- (7) If Q is a lattice, then $\epsilon_x(\phi) = \phi$ and $\epsilon_y(\psi) = \psi$ imply $\epsilon_{x \vee y}(\phi \cdot \psi) = \phi \cdot \psi$, that is, if x is a support of ϕ , and y a support of ψ , then $x \vee y$ is a support of $\phi \cdot \psi$.

The last item reminds of the *Labeling axiom* of a labeled information algebra. In fact, if Q is a lattice, then we may associate a labeled information algebra to a *supported* domain-free information algebra $(\Phi, E; \cdot, \circ)$. Here we assume as usual that E is labeled by the lattice Q . Consider the set of all pairs

(ϕ, x) , where $\phi \in \Phi$ and x is a support of ϕ . That is, we label all elements of Φ with all of its supports. Let then

$$\Psi = \{(\phi, x) : \phi \in \Phi, \epsilon_x(\phi) = \phi\} \tag{25}$$

Then we define a labeling operation on Ψ by

$$d(\phi, x) = x \tag{26}$$

Further, within Ψ , we define a *combination* operation by

$$(\phi, x) \cdot (\psi, y) = (\phi \cdot \psi, x \vee y) \tag{27}$$

Note that this operation is well defined by property (7) in the list above. Clearly, Ψ is a commutative and idempotent semigroup under this combination operation. It has $(1, x)$ and $(0, x)$ as unit and null elements on domains $x \in Q$. Finally, we define a *projection* operation for any $x \leq y$ by

$$\pi_x(\phi, y) = (\epsilon_x(\phi), x) \tag{28}$$

It can be verified that the algebra $(\Psi, Q; d, \cdot, \pi)$ indeed satisfies all axioms of a labeled information algebra [6]. If we go back from this labeled information algebra to the domain-free algebra by forming the quotient algebra Ψ/σ , then Ψ/σ will be isomorphic to the initial one, namely $(\Phi, E; \cdot, \circ)$. A formal definition of a morphism between information algebras will be given in Section 4.5, for the moment an intuitive understanding is sufficient. Similarly, if we start with a labeled information algebra $(\Psi, Q; d, \cdot, \pi)$, construct the domain-free algebra on Ψ/σ and take the associated labeled algebra using the pair construction above, then this labeled algebra will be isomorphic to the initial one. In this sense, labeled and domain-free algebra are closely associated. Therefore, in the sequel, we shall freely select at convenience either the labeled or the domain-free version of an information algebra, and we take it for granted that any concepts and constructions may be developed in parallel in both versions of an information algebra—although typically one of the two settings will be preferable in most cases.

3.5. Atoms

In many cases there are maximal, most informative information elements; and often any piece of information is somehow composed from such elements. This issue will be discussed in this section. Here we consider a *labeled information algebra* $(\Psi, Q; d, \cdot, \pi)$, although a similar development would also be possible for domain-free algebras.

An element $\alpha \in \Psi$ with label $d(\alpha) = x$ is called an *atom* on domain x , if

- (1) $\alpha \neq 0_x$,
- (2) For all $\psi \in \Psi$, $d(\psi) = x$ and $\alpha \leq \psi$ jointly imply either $\psi = \alpha$ or $\psi = 0_x$.

Therefore, atoms are the most informative elements on domain x (with the exception of the non-information 0_x , the contradiction). In a pure order-theoretic view atoms would be called co-atoms, since atoms there are the smallest nonzero elements. However, in the view of information order, the present notion of an atom is justified, since it turns out that a piece of information may be identified

in many important cases just with a set of atoms implying the information. In relational information algebras (see Section 2.2) the one-element relations, the relations containing exactly one tuple, are atoms in the domain of the relation. Of course, information algebras may have no atoms. Note also that in a subset information algebra (see Section 2.2) the individual blocks of a partition \mathcal{P} are atoms on the domain represented by this partition.

Here are a few properties of atoms:

- (1) If α is an atom on domain x , $\psi \in \Psi$ and $d(\psi) = x$, then either $\psi \leq \alpha$ or $\psi \cdot \alpha = 0_x$.
- (2) If α is an atom on domain x , and $y \leq x$, then $\pi_y(\alpha)$ is an atom on domain y .
- (3) If α and β are atoms on domain x , then either $\alpha = \beta$ or $\alpha \cdot \beta = 0_x$.

For a proof we refer to [6]. Two pieces of information ϕ and ψ on a same domain x are called *contradictory* (or *disjoint*), if $\phi \cdot \psi = 0_x$. So atoms are mutually contradictory and an atom and a piece of information on the same domain are either contradictory or the information is smaller than the atom.

Let $At(\Psi_x)$ denote the set of all atoms with domain x , $At(\Psi)$ the set of all atoms, and for $\psi \in \Psi$ with label $d(\psi) = x$, let $At(\psi)$ be the set of all atoms implying ψ ,

$$At(\psi) = \{\alpha \in At(\Psi_x) : \psi \leq \alpha\} \tag{29}$$

If $\alpha \in At(\psi)$ we say that α is *contained* in ψ . For instance, in relational algebras, the atoms are the one-tuple relations: $\{\alpha\}$ is an atom contained in R , exactly if $\alpha \in R$.

We introduce a few notions with respect to atoms and labeled information algebras:

- (1) A labeled information algebra is called *atomic* if for all $\psi \in \Psi$ different from 0_x the set $At(\psi)$ is not empty, *i.e.*, if every piece of information contains an atom.
- (2) A labeled information algebra is called *atomistic*, if for all $\psi \in \Psi$ different from 0_x , ψ is the infimum of the atoms it contains,

$$\psi = \wedge At(\psi) \tag{30}$$

- (3) A labeled information algebra is called *completely atomistic*, if it is atomistic and if for all $x \in Q$ and for every subset of atoms $A \subseteq At(\Psi_x)$, the infimum $\wedge A$ exists and belongs to Φ .

A relational information algebra is completely atomistic. If in a relational information algebra V is the set of real numbers and only convex subsets in each domain are considered, then we still have a labeled information algebra (a subalgebra of the relational one), which is atomistic but not completely atomistic, and this is the case for many other subalgebras of a relational information algebra.

For atoms we have essentially two basic operations, namely *Labeling*: $\alpha \mapsto d(\alpha)$ and *Projection*: $\alpha \mapsto \pi_x(\alpha)$. In an *atomic* information algebra the system $(At(\Psi), Q; d, \pi)$ has the following properties, inherited from the corresponding labeled information algebra (see [6]):

- (1) If $x \leq d(\alpha)$, then $d(\pi_x(\alpha)) = x$.
- (2) If $x \leq y \leq d(\alpha)$, then $\pi_x(\pi_y(\alpha)) = \pi_x(\alpha)$.

- (3) If $d(\alpha) = x$, then $\pi_x(\alpha) = \alpha$.
- (4) If $d(\alpha) = x$, $d(\beta) = y$ and $\pi_{x \wedge y}(\alpha) = \pi_{x \wedge y}(\beta)$, then there exists a γ such that $d(\gamma) = x \vee y$ and $\pi_x(\gamma) = \alpha$, $\pi_y(\gamma) = \beta$.
- (5) If $d(\alpha) = x$ and $x \leq y$, then there exists a γ such that $d(\gamma) = y$ and $\pi_x(\gamma) = \alpha$.

Note that the ordinary tuples of a relational algebra also satisfy these conditions. Therefore we call a system like $(At(\Psi), Q; d, \pi)$ a *tuple system*. More on such systems will be said in Section 4.1.

Still considering an atomic labeled information algebra $(\Psi, Q; d, \cdot, \pi)$. We call subsets of $At(\Psi_x)$ *generalized relations* on domain x and denote the family of all such relations by $\mathcal{R}(At(\Psi))$.

In $\mathcal{R}(At(\Psi))$, we may define an operation of combination among generalized relations R and S on x and y respectively, according to the model of relational join:

$$R \bowtie S = \{\alpha \in At(\Psi) : d(\alpha) = x \vee y, \pi_x(\alpha) \in R, \pi_y(\alpha) \in S\}$$

We may also define projection of a generalized relation R on domain x to some domain $y \leq x$ as in relational algebra:

$$\pi_y(R) = \{\pi_y(\alpha) : \alpha \in R\} \tag{31}$$

Then, exactly as in relational information algebras (see Section 2.2), we may define a *generalized relational information algebra* of pairs (R, x) of generalized relations in $At(\Psi)$ with domain $x \in Q$. We call this algebra $(\mathcal{R}(At(\Psi)), Q; d, \cdot, \pi)$ (see [6,8]). As before, we may drop the pair notation, and denote the pair (\emptyset, x) by \emptyset_x .

The map $\psi \mapsto At(\psi)$ between Ψ and $\mathcal{R}(At(\Psi))$ is a *homomorphism* (see Section 4.5 and [6]) between the atomic information algebra $(\Psi, Q; d, \cdot, \pi)$ and the generalized relational algebra $(\mathcal{R}(At(\Psi)), Q; d, \cdot, \pi)$, that is,

$$At(\psi \cdot \phi) = At(\psi) \bowtie At(\phi) \tag{32}$$

$$At(\pi_x(\psi)) = \pi_x(At(\psi)) \tag{33}$$

$$At(1_x) = At(\Psi_x) \tag{34}$$

$$At(0_x) = \emptyset_x \tag{35}$$

If the labeled information algebra $(\Psi, Q; d, \cdot, \pi)$ is *atomistic*, then this map is one-to-one, hence an *embedding* of $(\Psi, Q; d, \cdot, \pi)$ into the relational algebra $(\mathcal{R}(At(\Psi)), Q; d, \cdot, \pi)$. If the algebra $(\Psi, Q; d, \cdot, \pi)$ is *completely atomistic*, then the map is even onto $\mathcal{R}(At(\Psi))$, and hence an *isomorphism* [6].

A similar analysis can be carried out in the domain-free version of an information algebra, which links atomic algebras to information algebras of subsets. These results are partial answers to the *Representation Problem*: Is any information algebra (labeled or domain-free) isomorphic to an information algebra of sets, and if yes, how? More on this question will be said in Section 6.

4. Generic Constructions

Where do information algebras come from? Are there general ways to construct information algebras? There are! In this Section we describe several generic ways to construct information algebras. There are at least three different approaches: First, similarly to atomic algebras, information algebras may be obtained from abstract tuple systems as *generalized relational algebras*. This is presented in Section 4.1. Secondly, *logic* provides a rich source for information algebras. There are several ways information algebras are related to logic, especially algebraic logic. Sections 4.2 to 4.4 are devoted to this important subject. Finally, new information algebras may be obtained from given ones by constructions used in Universal Algebra and Category Theory. This will be discussed in Section 4.5.

4.1. Generalized Relational Algebras

Regarding atoms of an atomic information algebra, we have said that atoms form a tuple system. Here we give a formal definition of the concept of an abstract tuple system. Let Q be a lattice and T a set, whose generic elements will be denoted by t, s, \dots . Suppose that two operations representing labeling and projection are defined:

- (1) *Labeling*: $d : T \rightarrow Q, t \mapsto d(t)$,
- (2) *Projection*: $\pi : T \times Q, t, x \mapsto \pi_x(t)$, defined for $x \leq d(t)$.

The system $(T, Q; d, \pi)$ is called a *tuple system*, if the following conditions are satisfied for $t, s \in T$ and $x, y \in Q$:

- (1) If $x \leq d(t)$, then $d(\pi_x(t)) = x$,
- (2) If $x \leq y \leq d(t)$, then $\pi_x(\pi_y(t)) = \pi_x(t)$,
- (3) If $d(t) = x$, then $\pi_x(t) = t$,
- (4) If $d(t) = x, d(s) = y$ and $\pi_{x \wedge y}(t) = \pi_{x \wedge y}(s)$, then there exists $r \in T$ such that $d(r) = x \vee y$ and $\pi_x(r) = t, \pi_y(r) = s$,
- (5) If $d(t) = x$ and $x \leq y$, then there exists $s \in T$ such that $d(s) = y$ and $\pi_x(s) = t$.

The elements of T are called *generalized tuples*, since ordinary tuples evidently satisfy the conditions of a tuple system. The label $d(t)$ of a tuple t is called its domain.

As usual, the elements of Q may be thought of as representing questions. The tuples with domain x are the possible answers to the question represented by $x \in Q$. Projection of a tuple t to some coarser domain y links the answers to a question with the answers to a coarser question. If $\pi_y(t) = \pi_y(s)$, then s and t belong to the same answer to the coarser question. As we have seen, the atoms of an atomic labeled information algebra form a tuple system. So do the blocks of a partition lattice satisfying condition in Equation (6). In fact, the elements of any labeled information algebra $(\Psi, Q; d, \cdot, \pi)$ form a tuple system $(\Psi, Q; d, \pi)$ (see [6]).

If we fix $x \in Q$, then any subset R of tuples with domain x represents a partial answer to the question x . Such a subset R is called a *generalized relation* with domain x . Denote by $\mathcal{R}(T)$ the family of all such generalized relations. In the case of the tuple system of the blocks of a partition lattice, generalized relations are simply sets of blocks of a same partition. As for atomic labeled information algebras in Section 3.5, we may define generalized relational joins between generalized relations R and S with domain x and y respectively, as well as relational projection to $y \leq x$ of a generalized relation R with domain x :

$$R \bowtie S = \{t \in T : d(t) = x \vee y, \pi_x(t) \in R, \pi_y(t) \in S\} \tag{36}$$

$$\pi_y(R) = \{\pi_y(t) : t \in R\} \tag{37}$$

As in the atomic case (Section 3.5) it can be shown that the pairs (R, x) , where R is a generalized relation with domain x , form a labeled information, where the operations are defined as follows:

- (1) *Labeling*: $d(R, x) = x$,
- (2) *Combination*: $(R, x) \cdot (S, y) = (R \bowtie S, x \vee y)$,
- (3) *Projection*: $\pi_x(R, y) = (\pi_x(R), x)$.

We denote this algebra by $(\mathcal{R}(T), Q; d, \cdot, \pi)$ and call it the generalized relational algebra over the tuple system (T, Q, d, π) . Note that this is an information algebra whose elements are *sets* of tuples; it is a *set algebra*.

4.2. Quantifier Algebras

A *monadic algebra* [15,16] is a Boolean algebra Φ together with a map $\exists : \Phi \rightarrow \Phi$, which is an *existential quantifier* (see Section 3.2), that is,

- (1) $\exists(1) = 1$,
- (2) $\exists(\phi) \leq \phi$,
- (3) $\exists(\exists(\phi) \vee \psi) = \exists(\phi) \vee \exists(\psi)$.

Now, since only the join operation is involved in the definition of an existential quantifier, this concept applies to any join-semilattice, see [12]. Therefore, let Φ be a commutative, idempotent semigroup, hence a semilattice. We add a family of extraction operators in the following way: Let $\{\exists_i : i \in V\}$ be a finite set of commuting existential quantifiers, that is:

$$\exists_i(\exists_j(\phi)) = \exists_j(\exists_i(\phi)) \tag{38}$$

for all $i, j \in V$ and all $\phi \in \Phi$. Then we may define a map \exists_s for any subset $s = \{i_1, \dots, i_n\} \subseteq V$ by

$$\exists_s(\phi) = \exists_{i_n}(\exists_{i_{n-1}}(\dots \exists_{i_1}(\phi) \dots)) \tag{39}$$

Clearly, \exists_s is an existential quantifier, and for subsets s and t of V ,

$$\exists_{s \cup t}(\phi) = \exists_s(\exists_t(\phi)) \tag{40}$$

In this way, we obtain a *quantifier algebra* $(\Phi, \mathcal{P}(V); \cdot, \exists)$, where $\exists : \Phi \times \mathcal{P}(V) \rightarrow \Phi$, and where $\mathcal{P}(V)$ is the power set of V . This is closely related to an information algebra: We may define extraction operators ϵ_s for subsets s of V by

$$\epsilon_s(\phi) = \exists_{V-s}(\phi) \tag{41}$$

Then, $(\Phi, \mathcal{P}(V); \cdot, \epsilon)$ is a domain-free information algebra. In specific applications of this construction, existential quantification usually corresponds to the elimination of variables, which amounts to extract information pertaining to the remaining variables (see [6]). For this type of algebras we refer to *cylindric algebras* [17], which are an important concept in algebraic logic.

4.3. Consequence Operators

How can we express information? One possibility is to do this with propositions expressed by sentences of some appropriate formal language. The internal syntactic structure of this language is of no concern here. The essential point is that there is an *entailment relation* which permits to deduce further sentences from a set of given sentences. This captures the essence of a logic. The present section is essentially based on [18].

Let then \mathcal{L} be a set of sentences, representing the *language* to be considered. An *entailment relation* relates sets of sentences $X \subseteq \mathcal{L}$ to single sentences $s \in \mathcal{L}$, such that the following conditions are satisfied:

- (1) $X \vdash s$ for all $s \in X$,
- (2) If $X \vdash s$ for all $s \in Y$ and $Y \vdash t$, then $X \vdash t$.

Such entailment relations are used, for instance, in information systems introduced by Scott [14]. The relation $X \vdash s$ means that the sentence s can be deduced from X , or is entailed by X .

We may ask for the set of all sentences entailed by a set X . This leads to an operator C mapping the power set of \mathcal{L} into itself,

$$C(X) = \{s \in \mathcal{L} : X \vdash s\} \tag{42}$$

This mapping is a *consequence operator* (or also a *closure operator*); that is, it satisfies the following three conditions [6]:

- (1) $X \subseteq C(X)$,
- (2) $C(C(X)) = C(X)$,
- (3) if $X \subseteq Y$, then $C(X) \subseteq C(Y)$.

Sets X such that $X = C(X)$ are called *closed*; in particular any set of the form $C(X)$ is closed. We consider the closed sets as our pieces of information. Thus let Φ denote the family of all closed sets. Note that, in general, any closed set A can be written as $A = C(X)$ with $X \subseteq A$ in many different ways. Our interest usually is in finding small, at best finite sets X generating A in this way. So a piece of information A is coded nonuniquely by a set of sentences X contained in A . If we have two pieces

of information coded by X and Y respectively, then the combined information is the closure of the set $X \cup Y$. So, we define a combination operation within Φ by

$$C(X) \cdot C(Y) = C(X \cup Y) \tag{43}$$

Since $C(X \cup Y) = C(C(X) \cup C(Y))$, this operation is well defined among closed sets. Clearly, under this operation Φ is a commutative, idempotent semigroup. The set $C(\emptyset)$ of all tautologies is the unit element of this combination and \mathcal{L} is its null element. In fact, closed sets form a \cap -system, since the intersection of any family of closed systems is closed [14]. So Φ is a *complete lattice* under inclusion as order, intersection is the meet and the join is given by our combination operation.

In order to obtain an information algebra of closed sets, we need to introduce extraction operators into the semigroup Φ . For this purpose, we consider sublanguages $\mathcal{M} \subseteq \mathcal{L}$ of the original language. The information $C(X)$ restricted to \mathcal{M} is simply $C(X) \cap \mathcal{M}$. However, this set is not necessarily closed. Thus a candidate for an extraction operator relative to the sublanguage \mathcal{M} is obtained by taking the closure of this set,

$$C_{\mathcal{M}}(C(X)) = C(C(X) \cap \mathcal{M}) \tag{44}$$

This is a map $C_{\mathcal{M}} : \Phi \rightarrow \Phi$.

We want this map to be an existential quantifier on the semilattice Φ of closed sets (see Section 3.2) in order to obtain an information algebra. This means that conditions (a), (b) and (c) in Section 3.2 must be satisfied, which imposes additional conditions on the consequence operator C .

Condition (a) requires that $C(\mathcal{M}) = \mathcal{L}$, that is, the sublanguage \mathcal{M} by itself should not yet convey any information. Then $C_{\mathcal{M}}(\mathcal{L}) = C(\mathcal{L} \cap \mathcal{M}) = C(\mathcal{M}) = \mathcal{L}$. So condition (a) of an existential quantifier is satisfied. Condition (b) is also satisfied, since $C_{\mathcal{M}}(C(X)) \subseteq C(X)$.

Condition (c) requires that for any two closed sets $E, F \in \Phi$,

$$C_{\mathcal{M}}(C(C_{\mathcal{M}}(E) \cup F)) = C(C_{\mathcal{M}}(E) \cup C_{\mathcal{M}}(F)) \tag{45}$$

This is equivalent to the following condition on the consequence operator C , for any subsets E and F of \mathcal{L} [6,18],

$$C(C((E \cap \mathcal{M}) \cup F) \cap \mathcal{M}) = C((E \cap \mathcal{M}) \cup (F \cap \mathcal{M})) \tag{46}$$

Next, we are going to consider a family \mathcal{S} of sublanguages \mathcal{M} of \mathcal{L} which is closed under finite intersection, and such that each operator $C_{\mathcal{M}}$ is an existential quantifier, *i.e.*, satisfies in particular Equation (46) for all $\mathcal{M} \in \mathcal{S}$. These sublanguages \mathcal{M} may be thought of representing questions, namely what propositions in \mathcal{M} are asserted by a piece of information $C(X) \in \Phi$. Further, we require that $E = \{C_{\mathcal{M}} : \mathcal{M} \in \mathcal{S}\}$ becomes a commutative, idempotent semigroup of extraction operators $C_{\mathcal{M}} : \Phi \rightarrow \Phi$. A sufficient condition for this is

$$C_{\mathcal{M}} \circ C_{\mathcal{N}} = C_{\mathcal{M} \cap \mathcal{N}} \tag{47}$$

for any pair $\mathcal{M}, \mathcal{N} \in \mathcal{S}$. Then in this way $(\Phi, E; \cdot, \circ)$ becomes a domain-free information algebra; an algebra associated with an entailment system (\mathcal{L}, \vdash) and an intersection-semilattice \mathcal{S} of sublanguages of \mathcal{L} . This shows one way a system of logic may induce an information algebra. So, an

information algebra can be obtained from a consequence operator satisfying the additional conditions in Equations (46) and (47) on some language \mathcal{L} .

Condition in Equation (47) means, in terms of the consequence operator C , that for any closed set E

$$C(C(E \cap \mathcal{N}) \cap \mathcal{M}) = C(C(E \cap \mathcal{M}) \cap \mathcal{N}) = C(E \cap \mathcal{M} \cap \mathcal{N}) \tag{48}$$

This last condition has a natural characterization in terms of interpolation [18]: A consequence operator C has the *interpolation property* with respect to a class of sublanguages \mathcal{S} , if, for all sublanguages \mathcal{M} and \mathcal{N} of \mathcal{S} , from $s \in \mathcal{M}$, $X \subseteq \mathcal{N}$ and $s \in C(X)$ it follows that there exists a set of sentences $Y \subseteq \mathcal{M} \cap \mathcal{N}$ such that $Y \subseteq C(X)$ and $s \in C(Y)$. The set Y is called the interpoland between s and X . It can be shown [18] that a consequence operator C satisfies Equation (48) if and only if it has the interpolation property with respect to the family of sublanguages \mathcal{S} . A similar characterization of condition in Equation (46) is not known so far. In [18] a sufficient condition implying Equation (46) is given. Further, we refer to [19] for a discussion of information algebras related to various systems of logic.

4.4. Language and Models

In the previous section, information was defined on a purely syntactical level as closed theories. No semantics was considered. In this section we add *semantics* to language. As before, consider a set \mathcal{L} of sentences. Add now another set \mathcal{M} , whose elements are considered to be models or possible worlds. We consider triples $(\mathcal{L}, \mathcal{M}, \models)$, where \models is a binary relation $\models \subseteq \mathcal{L} \times \mathcal{M}$ between sentences and models. We write $m \models s$ (m models s) instead of $(s, m) \in \models$. Such triples are called *classifications* in the book [20] on *information flow*, and the terms *types* and *tokens* are used instead of our terms of sentences and models. Further, they occur in *formal concept analysis* [14,21] where the elements of \mathcal{L} and \mathcal{M} are called *attributes* and *objects*.

In our discussion, which is based on [9], we assume that the relation $m \models s$ means that model m satisfies sentence s or that m is a *model*, or a *possible interpretation*, of sentence s . An example is provided by *propositional logic*, where \mathcal{L} is a propositional language and the elements of \mathcal{M} are valuations, and $m \models s$ means that m satisfies s in the usual definition in propositional logic. Another example is provided by *predicate logic*. The language \mathcal{L} is given by formulae of predicate logic and models \mathcal{M} are structures used to interpret formulae. We refer to [6,22] for a discussion of information algebras relating to propositional and predicate logic, which are special cases of the structures considered here.

In a triple $(\mathcal{L}, \mathcal{M}, \models)$ a set X of sentences determines a set of models by

$$\hat{r}(X) = \{m \in \mathcal{M} : m \models s \text{ for all } s \in X\} \tag{49}$$

We call $\hat{r}(X)$ the models of X . We assume here that there is no model which satisfies all sentences of \mathcal{L} , that is, $\hat{r}(\mathcal{L}) = \emptyset$. Similarly, a set A of models determines a set X of sentences by

$$\check{r}(A) = \{s \in \mathcal{L} : m \models s \text{ for all } m \in A\} \tag{50}$$

The set $\check{r}(A)$ is called the theory of A . Note that $\check{r}(\emptyset) = \mathcal{L}$. The significance of \hat{r} and \check{r} will become clear in a moment. We note that

$$\hat{r}(X) \supseteq A \iff X \subseteq \check{r}(A) \tag{51}$$

This means that \hat{r} and \check{r} form a *Galois connection* [14]. We define further for $X \subseteq \mathcal{L}$ and $A \subseteq \mathcal{M}$,

$$(52)$$

It is well known that C_{\models} and C^{\models} are *consequence operators* in \mathcal{L} and \mathcal{M} respectively [14]. We consider the family of C_{\models} -closed sets $\Phi_{\models} \subseteq \mathcal{L}$ and C^{\models} -closed sets $\Phi^{\models} \subseteq \mathcal{M}$. Again, it is well known that Φ_{\models} and Φ^{\models} are complete lattices under inclusion [14].

We now construct two closely related information algebras based on the lattices Φ_{\models} and Φ^{\models} . Note that the models of a set of sentences X , $\hat{r}(X)$, are C^{\models} closed sets in \mathcal{M} [14]. We consider the models of X as the *information* about possible worlds expressed by a set of sentences X . Similarly, the theory $\check{r}(A)$ of a set of models is C_{\models} -closed. This is the *theory* of A . In Φ_{\models} we define combination, as in the preceding Section 4.3 by

$$C_{\models}(X) \cdot C_{\models}(Y) = C_{\models}(X \cup Y) \tag{53}$$

This corresponds to the information order $C_{\models}(X) \leq C_{\models}(Y)$ if $C_{\models}(X) \subseteq C_{\models}(Y)$. The larger a theory, the more informative it is. In Φ^{\models} however, we define

$$C^{\models}(A) \cdot C^{\models}(B) = C^{\models}(A) \cap C^{\models}(B) \tag{54}$$

This corresponds to the information order $C^{\models}(A) \leq C^{\models}(B)$ if $C^{\models}(A) \supseteq C^{\models}(B)$. The smaller a model set is the more precise and the more informative it is. This makes sense, if the questions we consider search for an unknown possible world, an unknown model. In both cases Φ_{\models} as well as Φ^{\models} become commutative, idempotent semigroups. In the first case the vacuous information is the tautology $C_{\models}(\emptyset)$, in the second case the whole model set \mathcal{M} . The null element in the first case is the whole language \mathcal{L} , in the second case it is the empty set.

Next, we introduce extraction operators. We do this here first with respect to Φ^{\models} ; corresponding extraction operators for Φ_{\models} will be derived later. For this purpose we consider a partition of \mathcal{M} and use the associated equivalence relation defined by $m \equiv n$ iff m and n belong to the same block of the partition. Extraction then means to restrict an element $\phi \in \Phi^{\models}$ to this partition. This is expressed by the *saturation operators* ϵ with respect to its partition. If A is any subset of \mathcal{M} , then

$$\epsilon(A) = \{m \in \mathcal{M} : \exists n \in A, m \equiv n\} \tag{55}$$

We require that the saturation of a C^{\models} -closed set is still C^{\models} -closed. This requirement is fulfilled if and only if closure and saturation commute, $C^{\models}(\epsilon(A)) = \epsilon(C^{\models}(A))$. Under this assumption it turns out that ϵ satisfies conditions (a) to (c) of item 2 of the definition of a domain-free algebra automatically. Or, in other words, ϵ is an existential quantifier with respect to the intersection semilattice of C^{\models} -closed sets (see Section 3.2),

- (a) $\epsilon(\emptyset) = \emptyset$,
- (b) $\epsilon(\phi) \supseteq \phi$,
- (c) $\epsilon(\epsilon(\phi) \cap \psi) = \epsilon(\phi) \cap \epsilon(\psi)$.

Note that here the information order is the reverse of set inclusion, join in the information order is therefore set intersection.

Consider now a lattice Q of partitions of \mathcal{M} , more precisely, a sublattice of the lattice $part(\mathcal{M})$ of partitions of \mathcal{M} . We require that the following *independence condition* is satisfied (see Section 2.2, Equation (6)): If \mathcal{P}, \mathcal{Q} are two partitions from the lattice Q , then

$$m \equiv_{\mathcal{P} \wedge \mathcal{Q}} n \Rightarrow \exists l : m \equiv_{\mathcal{P}} l, n \equiv_{\mathcal{Q}} l \tag{56}$$

Let $\epsilon_{\mathcal{P}}$ be the saturation operator relative to the partition \mathcal{P} and E the family of all operators $\epsilon_{\mathcal{P}}$ for all $\mathcal{P} \in Q$. Then the independence condition above is sufficient for E to be a commutative, idempotent semigroup under composition, or what is the same:

$$\epsilon_{\mathcal{P}} \circ \epsilon_{\mathcal{Q}} = \epsilon_{\mathcal{P} \wedge \mathcal{Q}} \tag{57}$$

So, $(\Phi^{\models}, E, \cdot, \circ)$ defined in this way is a domain-free information algebra.

To any $\phi \in \Phi^{\models}$ we may associate its theory $\check{r}(\phi) \in \Phi_{\models}$. We have already defined combination in Φ_{\models} . Now, we also define extraction operators for $\mathcal{P} \in Q$ by

$$\tau_{\mathcal{P}}(\psi) = \check{r}(\epsilon_{\mathcal{P}}(\hat{r}(\psi))) \tag{58}$$

for $\psi \in \Phi_{\models}$. Then, with $T = \{\tau_{\mathcal{P}} : \mathcal{P} \in Q\}$, the system $(\Phi_{\models}, T, \cdot, \circ)$ also becomes a domain-free information algebra. In fact, the maps $\phi \mapsto \check{r}(\phi)$ and $\epsilon_{\mathcal{P}} \mapsto \tau_{\mathcal{P}}$ define an isomorphism from $(\Phi^{\models}, E, \cdot, \circ)$ onto $(\Phi_{\models}, T, \cdot, \circ)$, see [9].

We refer also to [9] for a discussion of the associated labeled construction and a discussion of the relation to the notions of *infomorphism* and *channels* in [20].

This is a generic way to construct information algebras from models of propositional and predicate logic, of systems of linear equations and inequalities, etc. It extends and generalizes *semantic information theory* [23–29]. In all cases mentioned the lattice Q of partitions of \mathcal{M} is usually obtained from a multivariate system (see Section 2.1), since these model are based on sets of variables and models are either vectors or sequences, see [6,9]. We refer also to [19], where structures called *similarity models* are introduced, using a similar approach to the one presented here to construct commutative and idempotent semigroups of extraction operators.

4.5. Morphisms

Usually, new algebraic structures are constructed from algebras of the same type. For example, Cartesian products of information algebras should be again an information algebras. This will be shown in Section 5.3. Here we consider maps between information algebras. This is also a step towards a category theoretical study of information algebra. More on this subject is to be found in Section 5.3.

Consider two domain-free information algebras $(\Phi_1, E_1, \cdot, \circ)$ and $(\Phi_2, E_2, \cdot, \circ)$. To simplify notation we use the same symbol for combination and composition in both algebras; it will always be clear from the context which operation is meant. In the same way, \leq denotes the order either in Φ_1 or in Φ_2 . A map $f : \Phi_1 \rightarrow \Phi_2$ is called *order preserving*, if $\phi \leq \psi$ in Φ_1 implies $f(\phi) \leq f(\psi)$ in Φ_2 . Let $[\Phi_1 \rightarrow \Phi_2]$ denote the set of order preserving maps from Φ_1 into Φ_2 . We define in $[\Phi_1 \rightarrow \Phi_2]$ a combination operation pointwise by the combination in Φ_2 ,

$$(f \cdot g)(\phi) = f(\phi) \cdot g(\phi) \tag{59}$$

Clearly $f \cdot g$ is still an order-preserving map and this combination operation of order-preserving maps is associative, commutative and idempotent. So $[\Phi_1 \rightarrow \Phi_2]$ becomes a commutative and idempotent semigroup under combination. The unit element of combination is given by the map $\phi \mapsto 1$, where 1 is the unit in Φ_2 and the null element is given by the map $\phi \mapsto 0$, where here 0 is the null element in Φ_2 . The associated order is defined by $f \leq g$ if and only if $f(\phi) \leq g(\phi)$ for all $\phi \in \Phi_1$.

In order to introduce extraction operations on $[\Phi_1 \rightarrow \Phi_2]$, we consider the cartesian product $E_1 \times E_2$ and define maps $(\epsilon_1, \epsilon_2)f : \Phi_1 \rightarrow \Phi_2$ by

$$((\epsilon_1, \epsilon_2)f)(\phi) = \epsilon_2(f(\epsilon_1(\phi))) \tag{60}$$

Clearly, the map $(\epsilon_1, \epsilon_2)f$ is still order preserving. It further satisfies conditions (a) to (c) of item 2 in the definition of a domain-free information algebra. The family $E_1 \times E_2$ of these maps is further commutative and idempotent under composition. So, $([\Phi_1 \rightarrow \Phi_2], E_1 \times E_2, \cdot, \circ)$ with combination and extraction as defined above is a domain-free information algebra. We shall argue in Section 5.3 that these maps are the correct choice of *morphisms* for making the class of domain-free information algebras a category.

5. Compact Algebras

5.1. Finiteness

Computers can only treat “finite” information, and “infinite” information can often be approximated by “finite” elements. This point of view is developed in particular in *domain theory*, a subject at the interface between order theory and computer science [14,30–33]. Here, we treat the problem of finiteness in a similar way and show that in this respect information algebras extend domain theory and that many results of this theory can be extended to information algebras.

We discuss finiteness in the context of domain-free information algebras, although it could also be studied in the context of labeled algebras. The relation between finiteness in domain-free and labeled information algebras is interesting and it is not yet fully understood, see [8].

Consider then a domain-free information algebra $(\Phi, E; \cdot, \circ)$. In the set Φ of information elements we single out a subset Φ_f of elements which we consider to be finite. To qualify for this, they must of course satisfy a number of requirements, which we shall enumerate below. Before, we need a new concept, which serves to express the idea that any element of Φ should be approximated by finite elements. This is the concept of a *directed set*: A nonempty subset $X \subseteq \Phi$ is called *directed*, if for every pair $\phi, \psi \in X$ there is an element $\chi \in X$ such that $\phi, \psi \leq \chi$.

Here are now the conditions we impose on Φ_f :

- (1) *Subalgebra*: The system $(\Phi_f, E, \cdot, \circ)$ is a subalgebra of (Φ, E, \cdot, \circ) , that is it is closed under combination and extraction and the unit 1 and the null element 0 belong to Φ_f .
- (2) *Convergence*: If $X \subseteq \Phi_f$ is directed, then the supremum $\vee X$ exists in Φ .
- (3) *Density*: For all $\phi \in \Phi$ and $\epsilon \in E$,

$$\epsilon(\phi) = \bigvee \{ \psi \in \Phi_f : \psi = \epsilon(\psi) \leq \phi \}$$

- (4) *Compactness*: If $X \subseteq \Phi_f$ is directed, and $\psi \in \Phi_f$ satisfies $\psi \leq \vee X$, then there is a $\phi \in X$ such that $\psi \leq \phi$.

If these conditions are satisfied, we call the system $(\Phi, \Phi_f, E; \cdot, \circ)$ a *compact information algebra*. Sometimes weaker conditions are sufficient: [6,8]: For instance it is often sufficient that Φ_f is closed under combination, but not under extraction. Further, if the information algebra is supported, then from the density condition it follows also that

$$\phi = \bigvee \{\psi \in \Phi_f : \psi \leq \phi\} \tag{61}$$

This says that any element of Φ can be approximated by the finite elements below it. Conversely, this is not sufficient for the density condition above to hold, which says that any element supported by some domain can be approximated by finite elements supported by the same domain. Therefore it is called the *weak density*. We shall not discuss these refinements here, see [8].

As examples of compact information algebras we cite the *algebra of strings*, where the finite strings are the finite elements. In the information algebra of *convex sets*, the *convex polyhedra* are the finite elements, since any convex set can be approximated by the convex polyhedra it is contained in. Further a domain-free information algebra $(\Phi, E; \cdot, \circ)$, where Φ is a finite set is always compact. However, in a set algebra, the cofinite sets (the complements of finite sets) do not qualify as finite elements, since only weak density is satisfied.

Important examples of compact information algebras are provided by consequence operators. Consider an entailment system (\mathcal{L}, \vdash) and its associated consequence operator C (see Section 4.3). Assume that C satisfies conditions in Equations (46) and (48) specified there, such that the system $(\Phi, E; \cdot, \circ)$ associated with the entailment system (\mathcal{L}, \vdash) and an intersection-semilattice \mathcal{S} of sublanguages is an information algebra. Assume now further that C satisfies the following condition:

$$C(X) = \bigcup \{C(Y) : Y \subseteq X, Y \text{ finite}\} \tag{62}$$

This is equivalent to the condition that if $X \vdash s$, then there is a finite subset Y of X such that $Y \vdash s$. With this additional condition on C , the information algebra $(\Phi, E; \cdot, \circ)$ of closed sets becomes a compact algebra whose finite elements are the sets $C(X)$, where X is a finite set (see [6]).

Another example of a compact information algebra is the *ideal completion* of a domain-free information algebra, see Section 3.3. In fact, the elements of Φ or rather the principal ideals $\downarrow \phi$ for $\phi \in \Phi$ are the finite elements in the compact algebra $(I_\Phi, E; \cdot, \circ)$, see [6]. So, embedding an information algebra into its ideal completion is a way to compactify this algebra. We refer to [34] for a variant of this compactification, yielding an analogous result. On the other hand the following theorem is an extension of a well-known representation theorem of domain theory [35]: If $(\Phi, \Phi_f, E; \cdot, \circ)$ is a compact information algebra, then the ideal completion $(I_{\Phi_f}, \Phi_f, E; \cdot, \circ)$ is a compact information algebra, isomorphic to $(\Phi, \Phi_f, E; \cdot, \circ)$. For a proof we refer to [8].

The ideal completion can also be seen in another perspective. The set

$$I(X) = \{\phi \in \Phi : \phi \leq \phi_1 \cdot \dots \cdot \phi_m \text{ for some finite set } \phi_1, \dots, \phi_m \in X\} \tag{63}$$

is the ideal generated by the set $X \subseteq \Phi$. This is a consequence operator in Φ . If we take Φ to be a language, and the information order as an entailment relation, that is, $X \vdash \phi$ if $\phi \leq \vee X$, then $I(X)$

is the consequence operator associated with the entailment relation. Further, for any $\epsilon \in E$, let Φ_ϵ be the set of all element of Φ , supported by ϵ , that is $\Phi_\epsilon = \{\phi \in \Phi : \phi = \epsilon(\phi)\}$. The family Φ_ϵ for $\epsilon \in E$ is closed under finite intersections. It turns out that the family of I -closed sets is just the set I_Φ of ideals of Φ . Moreover, the consequence operator I satisfies conditions in Equations (46) and (48) and also Equation (62) above, see [6]. Therefore, the information algebra generated by the consequence operator I together with the family of sublanguages $\{\Phi_\epsilon : \epsilon \in E\}$ is just the ideal completion of the domain-free information algebra $(\Phi, E; \cdot, \circ)$. In this way, any compact information algebra is generated by a consequence operator, as described in Section 4.3.

If $(\Phi, \Phi_f, E; \cdot, \circ)$ is a compact information algebra, then it follows that Φ is a *complete lattice* under the information order, see [6]. Further, it follows that $\psi \in \Phi_f$ if and only if, for every directed set X contained in Φ , $\psi \leq \vee X$ implies that there is a $\phi \in X$ such that $\psi \leq \phi$. In order theory, elements of a complete partially ordered (cpo) set satisfying this condition are called *finite elements*, [14]. Thus the order theoretic concept of finiteness corresponds with our concept. Also $\psi \in \Phi_f$ if and only if, for every set $X \subseteq \Phi$, $\psi \leq \vee X$ implies that there is a finite subset F of X such that $\psi \leq \vee F$. In order theory, elements of a cpo satisfying this condition are called *compact elements*, [14]. Thus the order theoretic concept of compactness corresponds also to our concept of finite elements. Finally, all of this together means that Φ is an *algebraic lattice*, see [14] and $\Phi - \{0\}$ is an *algebraically complete order* such that $\phi \vee \psi \in \Phi - \{0\}$ if $\phi \vee \psi \neq 0$. Such a structure is called a *Scott-Ershov domain*. This links domain-free information algebras to *domain theory* [35]. Compact information algebras are domains with an additional operation, namely the extraction. Many results from domain theory extend to compact information algebras. In particular, compactness could be generalized to *continuity* by use of the *way-below* relation. see [8,34].

5.2. Continuous Maps

Maps between sets of pieces of information also represent information. In fact, we have seen in Section 4.5 that order preserving maps between domain-free information algebras themselves form an information algebra. In the case of *compact* information algebras the associated morphisms are not simply order preserving maps, but continuous maps. The concept of an continuous map is inherited from *cpo*s: Let $(\Phi_1, \Phi_{1,f}, E_1; \cdot, \circ)$ and $(\Phi_2, \Phi_{2,f}, E_2; \cdot, \circ)$ be two compact information algebras. A map $f : \Phi_1 \rightarrow \Phi_2$ is called *continuous*, if for all $\phi \in \Phi_1$,

$$f(\phi) = \bigvee \{f(\psi) : \psi \in \Phi_{1,f}, \psi \leq \phi\} \tag{64}$$

This is also equivalent to the condition that $f(\vee X) = \vee f(X)$ for every directed set $X \subseteq \Phi$. Extraction operators of a compact information algebra $(\Phi, \Phi_f, E; \cdot, \circ)$ are continuous, that is, $\epsilon(\vee X) = \vee \epsilon(X)$ for any directed set X and any $\epsilon \in E$, see [6]. Further, a continuous map is order preserving.

Just in the same way as we defined an information algebra of order preserving maps, we can also obtain an information algebra of continuous maps. Let $[\Phi_1 \rightarrow \Phi_2]_c$ be the set of continuous maps between the two compact algebras $(\Phi_1, \Phi_{1,f}, E_1; \cdot, \circ)$ and $(\Phi_2, \Phi_{2,f}, E_2; \cdot, \circ)$. Define combination and extraction as in Section 4.5 for the case of order preserving maps, for $f, g \in [\Phi_1 \rightarrow \Phi_2]_c$ and $\epsilon_1 \in E_1, \epsilon_2 \in E_2$,

- (1) *Combination*: $(f \cdot g)(\phi) = f(\phi) \cdot g(\phi)$,
- (2) *Extraction*: $((\epsilon_1, \epsilon_2)f)(\phi) = \epsilon_2(f(\epsilon_1(\phi)))$.

With these operations $([\Phi_1 \rightarrow \Phi_2]_c, E_1 \times E_2; \cdot, \circ)$ becomes a domain-free information algebra, a subalgebra of the algebra of order preserving maps, Section 4.5. In fact, this is a compact algebra. Its finite elements are defined using maps $s : Y \rightarrow \Phi_{1,f}$, where Y is a finite subset of $\Phi_{1,f}$. These maps are called *simple*. Let S be the set of simple maps. For any $s \in S$ let $Y(s)$ be its domain and define

$$\hat{s}(\phi) = \bigvee \{s(\psi) : \psi \in Y(s), \psi \leq \phi\} \tag{65}$$

If the set on the right hand side is empty, then $\hat{s}(\phi) = 1$. It turns out that these maps \hat{s} are the finite elements of the information algebra of continuous maps, see [6].

5.3. Categories of Information Algebras

To conclude this review of information algebras we are going to consider the following two *categories* of information algebras:

- (1) The category **IA** has as its *objects* domain-free information algebras $(\Phi, E; \cdot, \circ)$ and as its *morphisms* order preserving maps.
- (2) The category **CompIA** has as its *objects* compact information algebras $(\Phi, \Phi_f, E; \cdot, \circ)$ and as its *morphisms* continuous maps.

The category **CompIA** is a subcategory of **IA**. It turns out that these categories are both *Cartesian Closed*.

This means that these categories satisfy the following three conditions:

- (1) both have *terminal* objects,
- (2) both have *direct* products,
- (3) both have *exponentials*.

An object T is a terminal object if there is exactly one morphism from any other object to T . In our case this is simply the information algebra $(\{1\}, \{id\}; \cdot, \circ)$, where Φ and E both consist of only one element.

Secondly, the Cartesian product $(\Phi_1 \times \Phi_2, E_1 \times E_2; \cdot, \circ)$ of two information algebras, where both combination and extraction are defined component-wise, belongs to **IA**. If the two algebras are compact, then $(\Phi_1 \times \Phi_2, \Phi_{1,f} \times \Phi_{2,f}, E_{1,f} \times E_{2,f}; \cdot, \circ)$ is compact too, hence belongs to **CompIA**. In fact, these Cartesian products are the categorial *direct product*. That is, if the projections p_1 and p_2 are defined in the usual way, $p_1(\phi_1, \phi_2) = \phi_1$ and $p_2(\phi_1, \phi_2) = \phi_2$, then, for any pair of morphisms (order preserving or continuous maps) f_1 and f_2 from a third information algebra $(\Phi, E; \cdot, \circ)$ into $(\Phi_1, E_1, \cdot, \circ)$ and $(\Phi_2, E_2, \cdot, \circ)$, respectively, there is a morphism f from $(\Phi, E; \cdot, \circ)$ into $(\Phi_1 \times \Phi_2, E_1 \times E_2; \cdot, \circ)$ such that $p_1 \circ f = f_1$ and $p_2 \circ f = f_2$.

Finally both categories **IA** and **CompIA** have *exponentials*. In fact, the information algebras $([\Phi_1 \rightarrow \Phi_2], E_1 \times E_2; \cdot, \circ)$ and $([\Phi_1 \rightarrow \Phi_2]_c, E_1 \times E_2; \cdot, \circ)$ of order preserving or continuous maps respectively are the exponentials in their respective categories. Define the map $eval : [\Phi_1 \rightarrow \Phi_2]_c \times \Phi_1 \rightarrow \Phi_2$ by

$$eval(f, \phi) = f(\phi). \quad (66)$$

This map is continuous, hence a morphism in **CompIA**. Next let f be a morphism $f : \Phi \times \Phi_1 \rightarrow \Phi_2$ for a third compact information algebra $(\Phi, \Phi_f, E; \cdot, \circ)$. Then define a map $\lambda f : \Phi \rightarrow [\Phi_1 \rightarrow \Phi_2]_c$ by

$$(\lambda f(\chi))(\phi) = f(\chi, \phi) \quad (67)$$

It turns out that the map λf is continuous, hence a morphism of the category **CompIA** and that $eval \circ (\lambda f, id_{\Phi_1}) = f$. The same holds with respect to ordinary information algebras. This shows that the information algebras $([\Phi_1 \rightarrow \Phi_2], E_1 \times E_2; \cdot, \circ)$ and $([\Phi_1 \rightarrow \Phi_2]_c, E_1 \times E_2; \cdot, \circ)$ are the exponentials in their respective categories.

So, indeed both categories **IA** and **CompIA** are Cartesian closed. For proofs we refer to the manuscript [8].

6. Outlook

In this section we address a few issues not discussed above. For a start, we consider the question what part information algebras could have in a general theory of information as sketched, for instance, in [36]. It seems that the basic pragmatic purpose of information is to answer questions. This is already implicit in Shannon's theory. The question there is what symbol or sequence of symbols is transmitted over a channel. This implies that one should be concerned with extracting the part of the information relevant to the question one is interested in. Information extraction is indeed a common activity in internet search, query answering in databases, constraint solving *etc.* Surely also information comes usually in pieces, from different sources and sensors, from varying files or data bases. Therefore, aggregation is an unavoidable operation in processing information. Any theory of information must in one or another way cover these issues. Whether it does it the way information algebras proposes is another question. The fact that classical information structures of Computer Science like relational databases, different systems of logic, systems of equations, constraints, *etc.* can be modeled by information algebras is a strong argument for its relevance. Also many studies of semantic information turn around the concept of questions [26–29,37–42], although the algebraic structure of extraction is not explicitly discussed. These studies are also closely connected to semantic information theory, as discussed for instance in [23–25,43]. These last papers study language and information in a way similar to Section 4.4 where it is shown how language expresses information and how information algebras arise from language and its semantics. Finally we have shown that information algebras are also closely related to domain theory, another important part of Computer Science modeling information and computation. So, information algebras are very tightly linked to important pragmatic information concepts and information theories of Computer Science and Philosophy. This will be underlined even more strongly below.

The classical information theories are those of Shannon [44] and Solomonov, Kolmogorov and Chaitin [45–48]. All these theories address the question of how to measure information content of

a message or an object (both seen as a sequence of symbols). In this sense, they are very limited theories of information. In information algebras we have an information order, that is a qualitative comparison of information content. We may wonder whether we may associate a quantitative measure of information content to the elements of an information algebra. This measure should respect the information order. This seems only possible if the information considered is in some sense “finite”, that is, can be represented by a finite number of bits. In [9] it is proposed to use Hartley measure to measure *uncertainty* of a partial answer to a given question in an atomistic information algebra, and then to measure information content by the reduction of uncertainty. As an illustration, consider the very particular case of a string algebra, see Section 3.1. We consider only binary strings to simplify matters. Suppose we look for a string of length n . There are 2^n possible strings of this length. If we know nothing about our string, then the measure of the uncertainty according to Hartley [49] is the logarithm of possible cases, that is n . If as an information we are given the prefix of length m of the string, there remain only 2^{n-m} possible strings, that is the remaining uncertainty is $n - m$. The reduction of uncertainty is then $n - (n - m) = m$. So the measure of information content in the string of length m about the string of length n is exactly m bits. This makes sense in this particular case as it does also in more general situations [9]. Shannon’s theory can be seen as an extension of Hartley’s approach to a probabilistic situation. Probabilistic information will be shortly touched below, although we claim that information is primarily not probabilistic. It turns out that Shannon’s way to measure information content can consistently be applied to information algebras, if some finiteness conditions are satisfied. It is an open question whether and how information algebra can be related to algorithmic information theory.

It has been mentioned at the beginning that the axiomatics of information algebra were originally motivated by local computation schemes [1,2]. The inference problem to be solved in a labeled information algebra $(\Psi, Q; d, \circ, \pi)$ can be formulated as follows: Given $\phi_1, \dots, \phi_n \in \Psi$ and $x \in Q$ compute

$$\pi_x(\phi_1 \cdot \dots \cdot \phi_n) \quad (68)$$

Query answering in relational databases, constraint solving, solving systems of linear equations, computing Fourier transforms and many other problems can be put into this form, see [7] for many further examples. The factors in this combination have domains $x_i = d(\phi_i)$. Naive combination leads to an element on domain

$$d(\phi_1 \cdot \dots \cdot \phi_n) = x_1 \vee \dots \vee x_n \quad (69)$$

This is usually computationally infeasible, since the complexity of extraction and combination grows often (although not always) exponentially in the size of the domain [7]. Local computation schemes organize the computation in such a way that all operations can be executed “locally” on the domains x_1, \dots, x_n of the factors of the combination. This is already possible in valuation algebras where the idempotency property is missing [5–7]. However, idempotency allows to simplify the computational architectures further. Local computation is also related to the idea of (conditional) independence of information, a concept known especially from probability theory [50,51], but also from relational database theory [52,53]. This indicates that this concept can be generalized to valuation or information algebras [6,54,55]. In general this subject is treated in the framework of multivariate systems. However, the general framework would be the one of a lattice of questions or domains. In this respect not much has been done so far, indicating a fruitful topic of research.

In Section 3.5 we have seen that atomistic labeled information algebras can be embedded into a generalized relational information algebra, that is an algebra of subsets. The question arises, whether any information algebra (labeled or domain-free) can be represented by an information algebra of sets, more particularly a generalized relational algebra in the labeled case. And how can this set algebra be characterized? This is the subject of *representation theory*. In lattice theory, the well-known Stone respectively Priestley duality theories solve exactly this problem, by giving precise topological characterizations of the representing set algebras for the case of Boolean algebras respectively distributive lattices (see [14]). Is it possible to extend these theories to distributive lattice or Boolean information algebras, *i.e.*, algebras where Φ is a distributive lattice or a Boolean information algebra? These questions also remain open to a large extent so far. Similar questions have been treated for special information algebras like cylindrical algebras, see [17].

Finally, information, undoubtedly, is often *uncertain*. In the framework of information algebras uncertain information can be modeled as random maps with values in an information algebra, see [6,56]. These random maps form themselves again information algebras. A possible interpretation of a random map is that an information may depend on some unknown assumptions. One may then formulate hypotheses and study, given a random map (an uncertain information), under what assumptions and hence how likely it is that the hypothesis holds. This induces a kind of distribution function for the random map; it is a function *monotone of order* ∞ . Such functions are studied in *capacity theory* [57,58]. Many results from capacity theory can be used to study random maps and uncertain information [6,8,59]. The theory of uncertain information as understood here is a generalization of what is known as Dempster-Shafer theory [60,61]; and in fact, information algebras provide the natural general framework for this theory. Much of this theory, especially the beautiful *Theory of Allocations of Probabilities* (see [62]) can be extended to information algebras. Random maps are also related to the *Theory of Hints* [63] and finally also to *Probabilistic Argumentation Systems* [64]. Finally, from the point of view of measuring information content, this probabilistic approach puts also Shannon's theory of information in a more general perspective, see [65] for a preliminary discussion. Much remains to be done here.

To conclude we only mention a few other issues, where so far little or nothing is known to the best of our knowledge: One is the introduction of topology into information algebras, somehow like it is done for domains (see [66]). Another one is the question of *effective* information algebras, where combination and extraction are effectively computable. This should bring information algebras into the focus of computability. The theory is yet relatively new and much remains to be done.

Acknowledgments

We thank the anonymous referees for their important and valuable comments.

Author Contribution

Juerg Kohlas is responsible for the concept of the paper, a large part of the results presented and the writing. Juerg Schmid has contributed to the mathematical soundness of the theory. Both authors have read and approved the final published manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Shenoy, P.P.; Shafer, G. Propagating Belief Functions Using Local Computation. *IEEE Expert* **1986**, *1*, 43–52.
2. Shenoy, P.P.; Shafer, G. Axioms for Probability and Belief Function Propagation. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*; Yager, R.R., Liu, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 499–528.
3. Lauritzen, S.L.; Spiegelhalter, D.J. Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *J. R. Stat. Soc. Ser. B* **1988**, *50*, 157–224.
4. Shenoy, P.P. A Valuation-Based Language for Expert Systems. *Int. J. Approx. Reason.* **1989**, *3*, 383–411.
5. Kohlas, J.; Shenoy, P.P. Computation in Valuation Algebras. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*; Kohlas, J., Moral, S., Eds.; Springer: Dordrecht, The Netherlands, 2000; pp. 5–39.
6. Kohlas, J. *Information Algebras: Generic Structures for Inference*; Springer: Berlin, Germany, 2003.
7. Pouly, M.; Kohlas, J. *Generic Inference. A Unified Theory for Automated Reasoning*; Wiley: Hoboken, NJ, USA, 2011.
8. Kohlas, J.; Schmid, J. *Research Notes: An Algebraic Theory of Information*; Technical Report 06-03; Department of Informatics, University of Fribourg: Fribourg, Switzerland, 2013. Available online: http://diuf.unifr.ch/drupal/tns/sites/diuf.unifr.ch.drupal.tns/files/file/main_0.pdf (accessed on 9 April 2014).
9. Kohlas, J.; Schneuwly, C. Information Algebra. In *Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information*; Sommaruga, G., Ed.; Lecture Notes in Computer Science, Volume 5363; Springer: Berlin, Germany, 2009; pp. 95–127.
10. Grätzer, G. *General Lattice Theory*; Academic Press: London, UK, 1978.
11. Kohlas, J.; Wilson, N. Semiring Induced Valuation Algebras: Exact and Approximate Local Computation Algorithms. *Artif. Intell.* **2008**, *172*, 1360–1399.
12. Cignoli, R. Quantifiers on Distributive Lattices. *Discret. Math.* **1991**, *96*, 183–197.
13. Halmos, P.R.; Givant, S. *Logic as Algebra*; Dolciani Mathematical Expositions Book 21; The Mathematical Association of America: Washington, DC, USA, 1998.
14. Davey, B.; Priestley, H. *Introduction to Lattices and Order*; Cambridge University Press: Cambridge, UK, 1990.
15. Halmos, P.R. *Algebraic Logic*; Chelsea: New York, NY, USA, 1962.
16. Plotkin, B. *Universal Algebra, Algebraic Logic, and Databases*; Mathematics and its applications; Springer: Dordrecht, The Netherlands, 1994; Volume 272.
17. Henkin, L.; Monk, J.D.; Tarski, A. *Cylindric Algebras*; North-Holland: Amsterdam, The Netherlands, 1971.

18. Kohlas, J.; Staerk, R. Information Algebras and Consequence Operators. *Log. Universalis* **2007**, *1*, 1139–1165.
19. Wilson, N.; Mengin, J. Logical Deduction Using the Local Computation Framework. In Proceedings of European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, ECSQARU'99, London, UK, 5–9 July 1999; pp. 386–396.
20. Barwise, J.; Seligman, J. *Information Flow: The Logic of Distributed Systems*; Number 44 in Cambridge Tracts in Theoretical Computer Science; Cambridge University Press: Cambridge, UK, 1997.
21. Ganter, B.; Wille, R. *Formal Concept Analysis*; Translated to English by C. Franzke; Springer: Berlin, Germany, 1999.
22. Langel, J. Logic and Information: A Unifying Approach to Semantic Information Theory. Ph.D. Thesis, University of Fribourg, Fribourg, Switzerland, 2010.
23. Bar-Hillel, Y.; Carnap, R. *An Outline of a Theory of Semantic Information*; Technical Report 247; Research Laboratory of Electronics, Massachusetts Institute of Technology: Cambridge, MA, USA, 1952.
24. Bar-Hillel, Y.; Carnap, R. Semantic Information. *Br. J. Philos. Sci.* **1953**, *4*, 147–157.
25. Bar-Hillel, Y. *Language and Information: Selected Essays on Their Theory and Application*; Addison-Wesley: Boston, MA, USA, 1964.
26. Hintikka, J. On Semantic Information. In *Physics, Logic, and History*, Proceedings of the First International Colloquium, Denver, CO, USA, 16–20 May 1966; Hintikka, J., Suppes, P., Eds.; Springer: New York, NY, USA, 1970; pp. 3–27.
27. Hintikka, J. Surface Information and Depth Information. In *Information and Inference*; Hintikka, J., Suppes, P., Eds.; Springer: Dordrecht, The Netherlands, 1970; pp. 263–297.
28. Hintikka, J. *The Semantics of Questions and the Questions of Semantics*; Volume 28, Acta Philosophica Fennica; North-Holland: Amsterdam, The Netherlands, 1976.
29. Hintikka, J. Answers to Questions. In *Questions*; Hiz, H., Ed.; D. Reidel: Dordrecht, The Netherlands, 1978; pp. 279–300.
30. Scott, D.S. *Outline of a Mathematical Theory of Computation*; Technical Monograph PRG-2; Oxford University Computing Laboratory, Programming Research Group: Oxford, UK, 1970.
31. Scott, D.S. Continuous Lattices. In *Toposes, Algebraic Geometry and Logic*; Lecture Notes in Mathematics; Springer: Berlin/Heidelberg, Germany, 1972; pp. 97–136.
32. Scott, D.S. Domains for Denotational Semantics. In *Automata, Languages and Programming*; Nielsen, M., Schmitt, E.M., Eds.; Springer: Berlin/Heidelberg, Germany, 1982; pp. 577–610.
33. Scott, D.S. Computer Science Department, Carnegie Mellon University, PA, USA. A New Category? Domains, Spaces and Equivalence Relations. Unpublished manuscript, 1996.
34. Guan, X.; Li, Y. On Two Types of Continuous Information Algebras. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2012**, *20*, 655–671.
35. Stoltenberg-Hansen, V.; Lindstroem, I.; Griffor, E. *Mathematical Theory of Domains*; Cambridge University Press: Cambridge, UK, 1994.
36. Burgin, M. *Theory of Information: Fundamentality, Diversity and Unification*; World Scientific: Hackensack, NJ, USA, 2010.

37. Groenendijk, J.; Stokhof, M. Studies on the Semantics of Questions and the Pragmatics of Answers. Ph.D. Thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, 1984.
38. Groenendijk, J.; Stokhof, M. Questions. In *Handbook of Logic and Language*, 2nd ed.; van Benthem, J., ter Meulen, A., Eds.; Elsevier: London, UK, 2010; Chapter 25, pp. 1059–1131.
39. Groenendijk, J. Questions and answers: Semantics and logic. In Proceedings of the 2nd CologNET-ElsET Symposium. Questions and Answers: Theoretical and Applied Perspectives, Amsterdam, The Netherlands, 18 December 2003; pp. 12–23.
40. Groenendijk, J. The Logic of Interrogation: Classical Version. In Proceedings of the Ninth Conference on Semantic and Linguistic Theory, Santa Cruz, CA, USA, 19–21 February 1999; pp. 109–126.
41. Hintikka, J. Questions about Questions. In *Semantics and Philosophy*; Munitz, M., Unger, P., Eds.; New York University Press: New York, NY, USA, 1974; pp. 103–158.
42. Van Rooij, R. Comparing Questions and Answers: A Bit of Logic, a Bit of Language and Some Bits of Information. Available online: <http://staff.science.uva.nl/vanrooy/Sources.pdf> (accessed on 8 April 2014).
43. Barwise, J.; Seligman, J. *Information Flow: The Logic of Distributed Systems*; Cambridge University Press: Cambridge, UK, 1997.
44. Shannon, C. A Mathematical Theory of Communications. *Bell Syst. Tech. J.* **1948**, *27*, 379–432.
45. Kolmogorov, A.N. Three Approaches to the Quantitative Definition of Information. *Int. J. Comput. Math.* **1968**, *2*, 1–4.
46. Kolmogorov, A.N. Logical Basis for information Theory and Probability. *IEEE Trans. Inf. Theory* **1968**, *14*, 662–664.
47. Chaitin, G. *Algorithmic Information Theory*; Cambridge University Press: Cambridge, UK, 1987.
48. Li, M.; Vitanyi, P. *An introduction to Kolmogorov complexity and its applications*; Springer: New York, NY, USA, 1993.
49. Hartley, R. Transmission of Information. *Bell Syst. Tech. J.* **1928**, pp. 535–563.
50. Cowell, R.G.; Dawid, A.P.; Lauritzen, S.L.; Spiegelhalter, D.J. *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*; Springer: Berlin/Heidelberg, Germany, 1999.
51. Dawid, A.P. Separoids: A Mathematical Framework for Conditional Independence and Irrelevance. *Ann. Math. Artif. Intell.* **2001**, *32*, 335–372.
52. Maier, D. *The Theory of Relational Databases*; Pitman: London, UK, 1983.
53. Beeri, C.; Fagin, R.; Maier, D.; Yannakakis, M. On the Desirability of Acyclic Database Schemes. *J. ACM* **1983**, *30*, 479–513.
54. Studeny, M. Formal Properties of Conditional Independence in Different Calculi of AI. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*; Clarke, M., Kruse, R., Moral, S., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Germany, 1993; Volume 747, pp. 341–348.
55. Shenoy, P. Conditional Independence in Valuation-based Systems. *Int. J. Approx. Reason.* **1994**, *10*, 203–234.
56. Kohlas, J.; Eichenberger, C. Uncertain Information. In *Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information*;

- Sommaruga, G., Ed.; Lecture Notes in Computer Science, Volume 5363; Springer: Berlin, Germany, 2009; pp. 128–160.
57. Choquet, G. Theory of Capacities. *Annales de l'Institut Fourier* **1953–1954**, *5*, 131–295.
 58. Choquet, G. *Lectures on Analysis*; Benjaminm: New York, NY, USA, 1969.
 59. Kohlas, J. Support-and Plausibility Functions Induced by Filter-Valued Mappings. *Int. J. Gen. Syst.* **1993**, *21*, 343–363.
 60. Dempster, A. Upper and Lower Probabilities Induced by a Multivalued Mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
 61. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
 62. Shafer, G. Allocations of Probability. *Ann. Prob.* **1979**, *7*, 827–839.
 63. Kohlas, J.; Monney, P. *A Mathematical Theory of Hints: An Approach to the Dempster-Shafer Theory of Evidence*; Volume 425, Lecture Notes in Economics and Mathematical Systems; Springer: Berlin/Heidelberg, Germany, 1995.
 64. Haenni, R.; Kohlas, J.; Lehmann, N. Probabilistic Argumentation Systems. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*; Kohlas, J., Moral, S., Eds.; Springer: Dordrecht, The Netherlands, 2000; pp. 221–287.
 65. Pouly, M.; Kohlas, J.; Ryan, P. Generalized Information Theory for Hints. *Int. J. Approx. Reason.* **2013**, *54*, 17–34.
 66. Gierz, E.G. *Continuous Lattices and Domains*; Cambridge University Press: Cambridge, UK, 2003.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).