Computing and Visualizing Closure Objects using Relation Algebra and RelView

Rudolf Berghammer and Bernd Braßel

Institut für Informatik, Christian-Albrechts-Universität Kiel
Olshausenstraße 40, D-24098 Kiel
{rub|bbr}@informatik.uni-kiel.de

Abstract. Closure objects play an important role in mathematics and computer science. We develop relation-algebraic specifications to recognize several classes of them, compute the complete lattices they constitute and transform any of these closure objects into another. All specifications are algorithmic and can directly be translated into the language of RelView. We show that the system is well suited for computing and visualizing closure objects and their lattices.

1 Introduction

The procedure of computing the closure of a given object is an important basic technique for applications in mathematics and computer science alike. The approach followed most frequently is to employ so-called closure operations. Examples include operations that yield the transitive closure of a relation or the subgroup generated by a set of elements of a given group. The practical importance of closures is also documented by the fact that equivalent or at least very similar notions are frequently reinvented. For example, what [6] denotes as "full implicational system" is called a "full family of functional dependencies" in the theory of relational databases [7] or a "closed family of implications" in formal concept analysis [9]. Likewise a "dependency relation" [6, Section 2.2] is the same as a "contact relation" in the sense of [1] and one needs only to transpose such a relation and restrict its range to $2^X \setminus \{\emptyset\}$ in order to obtain an "entailment relation" in the sense of [8]. Moreover, on the lattice-theoretic side it has been proven that there exists a close correspondence between all of the aforementioned concepts and the concept of a closure operation on a set. All of these objects form complete lattices which are pairwise isomorphic [6].

The subject of this paper (a short version of [3], where all proofs can be found) is to give a relation-algebraic representation of closure objects. The resulting specifications are algorithmic and directly lead to corresponding programs for RelView [5], which is a computer algebra system for the special purpose of computing with relations. The developed formulas provide three basic algorithms for each closure object given as a finite relation, viz. *recognition of an object*, i.e., deciding whether a given relation is, for example, a dependency relation, *transformation between closure objects*, e.g., compute the closure operation corresponding to a given full implicational system, and *computation of the complete lattice* which is constituted by the set of all such closure objects. Employing the resulting programs, the computer algebra system RelView supports the study of specific closure objects in several ways, among which are *visualization* as Boolean matrices or as graphs, providing various graph layout algorithms and offering a number of ways to highlight selected portions, *retracing* the results of sub-expressions via step-wise execution, and *testing*, e.g., by providing random matrices of a specified degree of filling. Beyond the possibility to study individual closure objects, the presented algorithms scale well and are applicable to large examples. This is due to the fact that finite relations can be implemented efficiently with BDDs [4] and we have taken care that the developed formulas fit the setting of RelView. The only exception is the computation of all possible closure systems which is presented in Section 3.1, as the according problem is well known to be exponential for sets.

2 Relation Algebra

Since many years relation algebra [11, 10] is used by many scientists as conceptual and methodological base of their work. Its importance is due to the fact that many objects of discrete mathematics can be seen as specific relations. Relation algebra allows concise and exact problem specifications and extremely formal and precise calculations that drastically reduce the danger of making mistakes. In this section, we recall the basics of relation algebra, provide the reader with an introduction of how pairs and products can be modeled and show how to represent sets. More specific to the task at hand, its last part deals with the relation-algebraic specification of extremal elements of ordered sets and lattices.

2.1 Relations and Relation Algebra

We write $R : X \leftrightarrow Y$ if R is a relation with domain X and range Y, i.e., a subset of $X \times Y$. If the sets X and Y of R's *type* $[X \leftrightarrow Y]$ are finite, we may consider R as a Boolean matrix. Since this interpretation is well suited for many purposes and is also used by the computer algebra system RelView as one of its possibilities to depict relations, we often use matrix terminology and matrix notation. Especially we speak about the rows, columns and entries of a relation and write $R_{x,y}$ instead of $\langle x, y\rangle \in R$ or $x\,R\,y$. Relation algebra knows three basic relations. The *identity relation* $\mathsf{I} : X \leftrightarrow X$ satisfying for all $x, y \in X$ that $\mathsf{I}_{x,y}$ iff $x = y$, the *universal relation* $\mathsf{L} : X \leftrightarrow Y$ holding for all $x \in X$ and $y \in Y$, and the *empty relation* $\mathsf{O} : X \leftrightarrow Y$ which holds for no pair in $X \times Y$. The *transposition* of a given relation $R : X \leftrightarrow Y$ is denoted by $R^\mathsf{T} : Y \leftrightarrow X$ and satisfies for all x, y that $R^\mathsf{T}_{x,y}$ iff $R_{y,x}$. When viewing relations as sets it comes natural to form the *union* $R \cup S : X \leftrightarrow Y$ and *intersection* $R \cap S : X \leftrightarrow Y$ of two relations $R, S : X \leftrightarrow Y$ or to state the *inclusion* $R \subseteq S$. A lot of the expressive power of relation algebra is due to the possibility to express the *composition* $RS : X \leftrightarrow Y$

of two relations $R : X \leftrightarrow Z$ and $S : Z \leftrightarrow Y$. Its definition in predicate logic is that for all $x \in X$ and $y \in Y$ we have $(RS)_{x,y}$ iff there exists a $z \in Z$ such that $R_{x,z}$ and $S_{z,y}$. The *complementation* of a relation $R : X \leftrightarrow Y$ is denoted by $\overline{R} : X \leftrightarrow Y$ and corresponds to negation in predicate logic: for all x, y we have $\overline{R}_{x,y}$ iff $R_{x,y}$ does not hold.

By $\mathrm{syq}(R, S) := \overline{R^\mathsf{T}\,\overline{S}} \cap \overline{\overline{R}^\mathsf{T} S} : Y \leftrightarrow Z$ the *symmetric quotient* of two relations $R : X \leftrightarrow Y$ and $S : X \leftrightarrow Z$ is defined. A specification in predicate logic is that $\mathrm{syq}(R, S)_{y,z}$ iff for all x we have that $R_{x,y}$ iff $S_{x,z}$. In other words for all $y \in Y$ and $z \in Z$ we have $\mathrm{syq}(R, S)_{y,z}$ iff the y-column of R equals the z-column of S. Additional properties of this construct can be found in [10].

2.2 Pairing and Related Constructions

The *pairing* (or *fork*) $[R, S] : Z \leftrightarrow X \times Y$ of two relations $R : Z \leftrightarrow X$ and $S : Z \leftrightarrow Y$ is defined by demanding for all $z \in Z$ and $u = \langle u_1, u_2 \rangle \in X \times Y$ that $[R, S]_{z,u}$ iff R_{z,u_1} and S_{z,u_2}. It should be noted that throughout this paper pairs $u \in X \times Y$ are assumed to be of the form $\langle u_1, u_2 \rangle$.

Using identity and universal relations of appropriate types, pairing allows to define the *projection relations* $\pi : X \times Y \leftrightarrow X$ and $\rho : X \times Y \leftrightarrow Y$ of the direct product $X \times Y$ as $\pi := [\mathsf{I}, \mathsf{L}]^\mathsf{T}$ and $\rho := [\mathsf{L}, \mathsf{I}]^\mathsf{T}$. Then the above definition implies for all $u \in X \times Y$, $x \in X$ and $y \in Y$ that $\pi_{u,x}$ iff $u_1 = x$ and $\rho_{u,y}$ iff $u_2 = y$. Also the *parallel composition* (or *product*) $R \,\|\, S : X \times X' \leftrightarrow Y \times Y'$ of two relations $R : X \leftrightarrow Y$ and $S : X' \leftrightarrow Y'$, such that $(R \parallel S)_{u,v}$ is equivalent to R_{u_1,v_1} and S_{u_2,v_2} for all $u \in X \times X'$ and $v \in Y \times Y'$, can be defined by means of pairing. We get the desired property if we define $R \,\|\, S := [\pi R, \rho S]$, where $\pi : X \times X' \leftrightarrow X$ and $\rho : X \times X' \leftrightarrow X'$ are the projection relations on $X \times X'$.

2.3 The Representation of Sets

In relation algebra sets can be modeled using *vectors*, which are relations v with $v = v\mathsf{L}$. For a vector the range is irrelevant and we therefore consider vectors $v : X \leftrightarrow \mathbf{1}$ with a specific singleton set $\mathbf{1} = \{\perp\}$ as range and omit the second subscript, i.e., write v_x instead of $v_{x,\perp}$. Such a vector can be considered as a Boolean matrix with exactly one column, i.e., as a Boolean column vector, and *represents* the subset $\{x \in X \mid v_x\}$ of X. A non-empty vector v is said to be a *point* if $vv^\mathsf{T} \subseteq \mathsf{I}$, i.e., v is *injective*. This means that it represents a single element. In the Boolean matrix model a point $v : X \leftrightarrow \mathbf{1}$ is a Boolean column vector in which exactly one entry is 1.

To model sets we will also apply *membership-relations* $\mathsf{M} : X \leftrightarrow 2^X$ on X and its powerset 2^X. These specific relations are defined by demanding for all $x \in X$ and $Y \in 2^X$ that $\mathsf{M}_{x,Y}$ iff $x \in Y$. Using BDDs as implementation of relations as in RelView, the number of BDD-nodes for M is linear in the cardinality of X. See [4] for details.

Given an injective function $\imath$ from Y to X, we may consider Y as a subset of X by identifying it with its image under $\imath$. If Y is actually a subset of X and

$\imath$ is given as relation of type $[Y \leftrightarrow X]$ such that $\imath_{y,x}$ iff $y = x$ for all $y \in Y$ and $x \in X$, then the vector $\imath^{\mathsf{T}}\mathsf{L} : X \leftrightarrow \mathbf{1}$ represents Y as subset of X in the sense above. To model sets, we will apply as third technique that the transition in the other direction is also possible, i.e., the generation of a relation $\mathrm{inj}(v) : Y \leftrightarrow X$ from the vector representation $v : X \leftrightarrow \mathbf{1}$ of $Y \subseteq X$ such that for all $y \in Y$ and $x \in X$ we have $\mathrm{inj}(v)_{y,x}$ iff $y = x$.

A combination of injective functions with membership-relations allows a *column-wise enumeration* of sets of subsets. More specifically, if $v : 2^X \leftrightarrow \mathbf{1}$ represents a subset $\mathfrak{S}$ of the powerset 2^X in the sense defined above, then for all $x \in X$ and $Y \in \mathfrak{S}$ we get the equivalence of $(\mathsf{M}\,\mathrm{inj}(v)^{\mathsf{T}})_{x,Y}$ and $x \in Y$. This means that $S := \mathsf{M}\,\mathrm{inj}(v)^{\mathsf{T}} : X \leftrightarrow \mathfrak{S}$ is the relation-algebraic specification of membership on $\mathfrak{S}$, or, using matrix terminology, the elements of $\mathfrak{S}$ are represented precisely by the columns of S. Furthermore, a little reflection shows for all $Y, Z \in \mathfrak{S}$ the equivalence of $Y \subseteq Z$ and $\overline{S^{\mathsf{T}}\,\overline{S}}_{Y,Z}$. Therefore, $\overline{S^{\mathsf{T}}\,\overline{S}} : \mathfrak{S} \leftrightarrow \mathfrak{S}$ is the relation-algebraic specification of set inclusion on $\mathfrak{S}$.

2.4 Extremal Elements of Orders and Lattices

Given a relation $R : X \leftrightarrow X$, the pair (X, R) is an ordered set iff $\mathsf{I} \subseteq R$ (reflexivity), $R \cap R^{\mathsf{T}} \subseteq \mathsf{I}$ (antisymmetry) and $RR \subseteq R$ (transitivity). In the following we may omit the set X when clear from context and simply refer to R as a partial order. When dealing with ordered sets, one typically investigates extremal elements. Based upon the vector representation of sets we will use the following relation-algebraic specifications taken from [10].

$$\mathrm{lel}(R, v) := v \cap \overline{\overline{R}\,v} \qquad \mathrm{gel}(R, v) := \mathrm{lel}(R^{\mathsf{T}}, v)$$
$$\mathrm{glb}(R, v) := \mathrm{gel}(R, \overline{\overline{R}\,v}) \qquad \mathrm{lub}(R, v) := \mathrm{glb}(R^{\mathsf{T}}, v)$$

If $R : X \leftrightarrow X$ is a partial order relation and Y a subset of X that is represented by the vector $v : X \leftrightarrow \mathbf{1}$, then $\mathrm{lel}(R, v) : X \leftrightarrow \mathbf{1}$ is empty iff Y does not have a least element and is a point that represents the least element of Y, otherwise. Similarly, $\mathrm{gel}(R, v) : X \leftrightarrow \mathbf{1}$ ($\mathrm{glb}(R, v) : X \leftrightarrow \mathbf{1}$ and $\mathrm{lub}(R, v) : X \leftrightarrow \mathbf{1}$, respectively) is either empty or a point that represents the greatest element (greatest lower bound and least upper bound, respectively) of Y,

If the second arguments of the above specifications are not vectors but "proper" relations with a non-singleton range, then the corresponding extremal elements are computed column-wisely. E.g., in the case of the first specification this means the following. For all $A : X \leftrightarrow Y$ we obtain $\mathrm{lel}(R, A) : X \leftrightarrow Y$ and, furthermore, for all $x \in X$ and $y \in Y$ that $\mathrm{lel}(R, A)_{x,y}$ iff the least element of $\{z \in X \mid A_{z,y}\}$ exists and equals x. Hence, for all $y \in Y$ the y-column of $\mathrm{lel}(R, A)$ is either empty or a point that represents (with respect to R) the least element of the set the y-column of A represents.

For a partial order $R : X \leftrightarrow X$ we also need the following specifications, which both are of type $[X \times X \leftrightarrow X]$:

$$\mathrm{Inf}(R) := [R, R]^{\mathsf{T}} \cap \overline{[R, R]^{\mathsf{T}}\,\overline{R}} \qquad \mathrm{Sup}(R) := \mathrm{Inf}(R^{\mathsf{T}})$$

In [2] it is shown that for all $u \in X \times X$ and $x \in X$ we have $\mathrm{Inf}(R)_{u,x}$ iff x is the greatest lower bound of u_1 and u_2 and $\mathrm{Sup}(R)_{u,x}$ iff x is the least upper bound of u_1 and u_2. Hence, $\mathrm{Inf}(R)$ and $\mathrm{Sup}(R)$ relation-algebraically specify the two lattice prerations $\sqcap$ and $\sqcup$. As a consequence, an ordered set (X, R) constitutes a lattice $(X, \sqcup, \sqcap)$ iff $\mathsf{L} = \mathrm{Inf}(R)\mathsf{L}$ and $\mathsf{L} = \mathrm{Sup}(R)\mathsf{L}$, since the latter equations express that the two relations $\mathrm{Inf}(R)$ and $\mathrm{Sup}(R)$ are total (see [10]). Also complete lattices can easily be characterized by relation-algebraic means. If $R : X \leftrightarrow X$ is the partial order of a lattice $(X, \sqcup, \sqcap)$, then X is complete iff $\mathsf{L} = \mathsf{L}\,\mathrm{glb}(R, \mathsf{M})$ or, equivalently, iff $\mathsf{L} = \mathsf{L}\,\mathrm{lub}(R, \mathsf{M})$, where $\mathsf{M} : X \leftrightarrow 2^X$ is the membership relation.

3 Computing and Visualizing Closure Objects

In the introduction we have mentioned several concepts which we refer to as "closure objects". In the literature, all of these notions are usually defined on powersets; see e.g., [6]. But with the exception of dependency relations the restriction to such a specific class of lattices is not necessary. We therefore prefer to define all closure objects on more general ordered structures. In this section, we develop relation-algebraic specifications for recognizing, computing and transforming closure objects on complete lattices. The specifications will be algorithmic and, hence, can directly be translated into RelView-code. Since RelView only allows to treat relations on finite sets, we assume for the developments finite lattices if this is advantageous.

3.1 Closure Systems

Assume $(X, \sqcup, \sqcap)$ to be a complete lattice. Then $S \subseteq X$ is called a *closure system* (or a Moore family) of X if it is closed under arbitrary least upper bounds, that is, for all $X \subseteq S$ we have $\bigsqcap X \in S$. In the case of a finite carrier set X this second order definition is obviously equivalent to the two requirements 1) that $\top \in S$, where $\top$ denotes the greatest element of the lattice, and 2) that for all $x, y \in S$ also $x \sqcap y \in S$. The next theorem provides the transformation of this first-order specification to relation algebra.

Theorem 3.1.1 *Assume $R : X \leftrightarrow X$ to be the partial order of a finite lattice $(X, \sqcup, \sqcap)$ and let $S \subseteq X$ be represented by the vector $s : X \leftrightarrow \mathbf{1}$. Then S is a closure system of X iff* $\mathrm{gel}(R, \mathsf{L}) \subseteq s$ *and* $[s^\mathsf{T}, s^\mathsf{T}]^\mathsf{T} \subseteq \mathrm{Inf}(R)\, s$.

The formulae of Theorem 3.1.1 can immediately be translated into the programming language of the computer algebra system RelView. Hence, given a partial order $R : X \leftrightarrow X$ of a lattice and a vector $s : X \leftrightarrow \mathbf{1}$, the tool can be used to test whether s represents a closure system. Note that in Theorem 3.1.1 the type of s^T is $[\mathbf{1} \leftrightarrow X]$ and, thus, the type of $[s^\mathsf{T}, s^\mathsf{T}]$ is $[\mathbf{1} \leftrightarrow X \times X]$. This is advantageous for the implementation in RelView, which is based on BDDs. In general, the transposition of a relation requires that a new BDD has to be computed from the

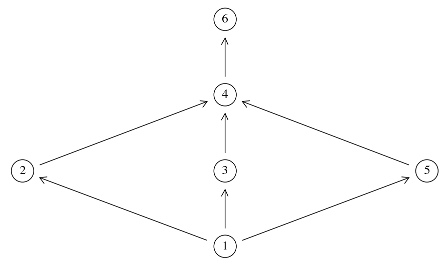

Fig. 1. Hasse-diagram of the partial order of a lattice with 6 elements

old one by exchanging the variables encoding the domain with those encoding the range. But in the case of a relation with domain or range $\mathbf{1}$ this process can be omitted since the BDD of the relation and its transpose coincide [4].

Having specified a single closure system within relation algebra, we turn to specify the set $\mathfrak{S}(X)$ of all closure systems of X as a subset of the powerset via a vector of type $[2^X \leftrightarrow \mathbf{1}]$. In the following theorem $\mathsf{M} : X \leftrightarrow 2^X$ is a membership relation, $\pi, \rho : X\times X \leftrightarrow X$ are the projection relations of $X\times X$, the left L has type $[X \leftrightarrow \mathbf{1}]$ and the remaining L is of type $[\mathbf{1} \leftrightarrow X\times X]$.

Theorem 3.1.2 *Let again $R : X \leftrightarrow X$ be the partial order of a finite lattice $(X, \sqcup, \sqcap)$. Then* $\mathrm{cls}(R) := (\mathrm{gel}(R,\mathsf{L})^{\mathsf{T}}\mathsf{M} \cap \overline{\mathsf{L}(\pi\mathsf{M} \cap \rho\mathsf{M} \cap \overline{\mathrm{Inf}(R)\,\mathsf{M}}\,)}\,)^{\mathsf{T}} : 2^X \leftrightarrow \mathbf{1}$ *is a vector-representation of the set $\mathfrak{S}(X)$ of all closure systems of X.*

Again, the transpositions occurring in the definition of $\mathrm{cls}(R)$ are motivated by the aim to obtain an efficient RelView program. Stating the formula in the given way, the relations affected during program execution are all of domain $\mathbf{1}$. If, in contrast, we would not have simplified the result by applying the rule $R^{\mathsf{T}}S^{\mathsf{T}} = (SR)^{\mathsf{T}}$ the resulting program would be less efficient and not scale anymore.

A significant fact about the set $\mathfrak{S}(X)$ is that it is itself a closure system of the powerset lattice $(2^V, \cup, \cap)$. Hence, it forms a complete lattice with intersection as greatest lower bound operation and set inclusion as partial order. A relation-algebraic specification of the partial order of $\mathfrak{S}(X)$ is rather simple. Using the technique described in Section 2.3, by

$$\mathrm{ClSys}(R) := \mathsf{M}\,\mathrm{inj}(\mathrm{cls}(R))^{\mathsf{T}} : X \leftrightarrow \mathfrak{S}(X)$$

the set $\mathfrak{S}(X)$ is enumerated by column and we immediately obtain from $\mathrm{ClSys}(R)$ that the set inclusion on $\mathfrak{S}(X)$ can be specified as

$$\mathrm{ClLat}(R) := \overline{\mathrm{ClSys}(R)^{\mathsf{T}}\,\overline{\mathrm{ClSys}(R)}} : \mathfrak{S}(X) \leftrightarrow \mathfrak{S}(X).$$

The number of different closure systems of a finite lattice grows very rapidly, see [6] for the numbers in the case of powersets 2^X up to $|X| = 6$. Therefore, a visualization of the according lattice is useful for rather small examples, only.

Example 3.1.1. Figure 1 contains a picture of the Hasse-diagram of the partial order of a lattice X^* with 6 elements $x_1, \ldots .x_6$. The picture demonstrates the

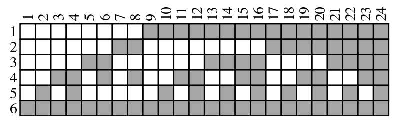

Fig. 2. Closure systems of the partial order of Figure 1

support in RelView to depict relations as directed graphs. Each vertex with label n represents the lattice element x_n, $1 \leq n \leq 6$.

Stating the two relation-algebraic specifications ClSys(R) and ClLat(R) above programs in the programming language of RelView, we computed that 24 of the 64 subsets of X^* are closure systems. The result of this computation is shown in Figure 2. There the 24 subsets are enumerated by column in a 6×24 Boolean matrix. In the picture a filled square denotes a 1-entry and an empty square a 0-entry. If we denote the closure system represented by column i with S_i, $1 \leq i \leq 24$, then, e.g., the first column represents the closure system $S_1 = \{x_6\}$ consisting of the greatest lattice element only, the second column represents the closure system $S_2 = \{x_5, x_6\}$, the third column represents the closure system $S_3 = \{x_4, x_6\}$ and the last column represents the closure system $S_{24} = X^*$ consisting of all lattice elements.

Finally, Figure 3 shows the Hasse-diagram of the lattice $\mathfrak{S}(X^*)$, where the vertex with label i corresponds to the closure system S_i, $1 \leq i \leq 24$, and an arrow denotes set inclusion. From Figure 2 it follows that the n-th layer of the graph exactly contains the vertices corresponding to the closure systems with cardinality n, $1 \leq n \leq 6$.

3.2 Closure Operations

For a given ordered set (X, R) a *closure operation* is a function $C : X \to X$ which is 1) extensive, 2) monotone, and 3) idempotent. Note that it is not necessary to restrict such operations to sets as arguments. In the next theorem we provide

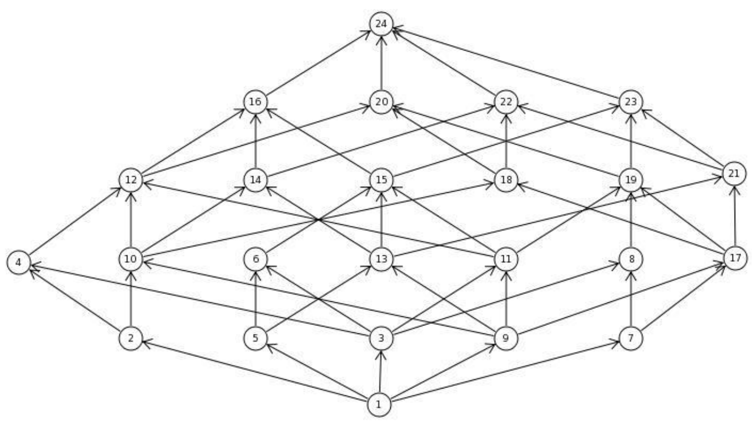

Fig. 3. Hasse-diagram of the lattice $\mathfrak{S}(X^*)$

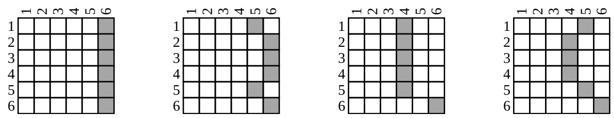

Fig. 4. Closure operations $C_1 \dots C_4$ of $\mathfrak{O}(X^*)$

a relation-algebraic characterization of closure operations of (X, R) as specific relations of type $[X \leftrightarrow X]$. Again the given formulae directly lead to a RelView-program for recognizing closure operations.

Theorem 3.2.1 *Given a partial order $R : X \leftrightarrow X$, a relation $C : X \leftrightarrow X$ is a closure operation of the ordered set (X, R) iff $C\overline{\mathsf{I}} = \overline{C}$, $C \subseteq R$, $R \subseteq CRC^{\mathsf{T}}$ and $CC \subseteq C$.*

On complete lattices $(X, \sqcup, \sqcap)$ there is a well-known one-to-one correspondence between the set $\mathfrak{O}(X)$ of all closure operations and the set $\mathfrak{S}(X)$ of all closure systems. The closure system corresponding to the closure operation $C \in \mathfrak{O}(X)$ is the set of all fixed points of C. Conversely, the closure operation corresponding to the closure system $S \in \mathfrak{S}(X)$ is such that $x \in X$ is mapped to $\bigsqcap\{z \in S \mid R_{x,z}\}$, where $R : X \leftrightarrow X$ is the partial order of the lattice. The following theorem states how these correspondences can be formulated as a pair of relation-algebraic specifications. To this end, we identify the subsets of X with their representation as vectors from $[X \leftrightarrow \mathbf{1}]$. This enables us to consider $\mathfrak{S}(X)$ to as a subset of the powerset $2^{[X \leftrightarrow \mathbf{1}]}$, and $\mathfrak{O}(X)$ as a subset of $[X \leftrightarrow X]$.

Theorem 3.2.2 *Let $R : X \leftrightarrow X$ be the partial order of a lattice $(X, \sqcup, \sqcap)$. Then, for all $C \in \mathfrak{O}(X)$ the vector $\mathrm{CloToCls}(C) := (C \cap \mathsf{I})\mathsf{L} : X \leftrightarrow \mathbf{1}$ represents the set of all fixed points of C and for all $s \in \mathfrak{S}(X)$ the relation $\mathrm{ClsToClo}(s) := \mathrm{glb}(R, s\mathsf{L} \cap R^{\mathsf{T}})^{\mathsf{T}} : X \leftrightarrow X$ fulfills for all $x, y \in X$ that $\mathrm{ClsToClo}(s)_{x,y}$ iff $y = \bigsqcap\{z \in V \mid s_z \wedge R_{x,z}\}$.*

The two functions $\mathrm{CloToCls} : \mathfrak{O}(X) \to \mathfrak{S}(X)$ and $\mathrm{ClsToClo} : \mathfrak{S}(X) \to \mathfrak{O}(X)$ of Theorem 3.2.2 are order-*reversing* (antiton) with respect to set inclusion for closure systems and the pointwise ordering of functions. The latter order on functions can be specified in relation algebra as $C_1 \leq C_2$ iff $C_1 \subseteq C_2R^{\mathsf{T}}$ since $C_1 \subseteq C_2R^{\mathsf{T}}$ says that for all $x, y \in X$ if C_1 maps x to y then C_2 maps x to a $z \in X$ with $R_{y,z}$.

Example 3.2.1. As the functions of Theorem 3.2.2 are order-reversing (antiton), a transposition of the Hasse-diagram in Figure 3 yields the Hasse-diagram of the lattice $\mathfrak{O}(X^*)$. Accordingly, in the resulting graph the vertex with label i represents the closure operation C_i corresponding to the closure system S_i for $1 \leq i \leq 24$ and Figure 4 depicts the relations $C_1, \dots, C_4$. The function C_1 maps all elements to the greatest lattice element. It is the greatest closure operation with respect to the pointwise ordering. The least closure operation is the identity relation I and corresponds to the greatest closure system S_{24}.

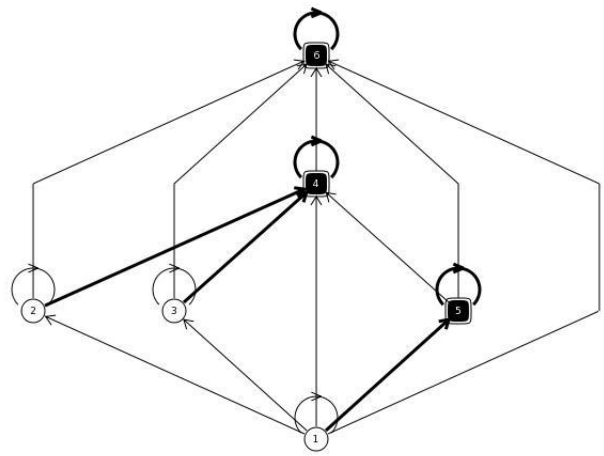

Fig. 5. Lattice X^*, emphasizing closure system S_4 and closure operation C_4

Figure 5 shows the partial order relation of the lattice X^* as directed graph, where the vertices of the closure system S_4 are emphasized as black squares and the arcs corresponding to the pairs of the closure operation C_4 are drawn boldface. Two of the three properties of closure operations can immediately be verified by examining the picture. The operation C_4 is extensive, since each arc of C_4 is an arc of the graph. It is idempotent, since each C_4-path leads into a loop over at most one non-loop edge. Monotonicity of C_4 can be recognized by pointwise comparisons. The picture also clearly visualizes that each element of the lattice is either a fixed point of C_4, i.e., contained in the corresponding closure system S_4, or is mapped to the least element of S_4 above it.

The *topological closure operations* of a lattice $(X, \sqcup, \sqcap)$ distribute over the $\sqcup$-operation and form an important subclass of $\mathfrak{O}(X)$. If the lattice X is finite, the corresponding closure systems are precisely the sublattices of X which contain the greatest element of X. Assuming $R : X \leftrightarrow X$ as partial order of X, a calculation shows that $C \in \mathfrak{O}(X)$ is topological iff $(C \parallel C)\,\mathrm{Sup}(R) = \mathrm{Sup}(R)\,C$. We have transformed this equation into RelView-code and computed for the above example lattice X^* that exactly 20 out of the 24 closure operations are topological. The four exceptions are C_{14}, C_{18}, C_{21} and C_{22}.

3.3 Full Implicational Systems and Join-Congruences

The origin of full implicational systems is relational database theory, where they are called families of functional dependencies (see e.g., [7]). In [6] full implicational systems are defined on powersets by a variant of the well known Armstrong axioms which require for the relation, written as arrow, and for all sets A, B, C, D that 1) if $A \to B$ and $B \to C$ then $A \to C$, 2) if $A \supseteq B$ then $A \to B$ and 3) if $A \to B$ and $C \to D$ then $A \cup C \to B \cup D$.

We generalize this description to finite (complete) lattices $(X, \sqcup, \sqcap)$ with partial order $R : X \leftrightarrow X$. In this sense, a *full implicational system* on X is a relation $F : X \leftrightarrow X$ that is 1) transitive, 2) contains R^{T} and 3) for all $x, y, x', y' \in X$ it holds that $F_{x,x'}$ and $F_{y,y'}$ imply $F_{x \sqcup y, x' \sqcup y'}$. As a side remark we note that axiom 3) could be generalized to arbitrary least upper bounds, i.e., arbitrary complete lattices. The resulting relation-algebraic formulation would be that 3') for all subrelations $D \subseteq F$ we have $vw^{\mathsf{T}} \subseteq F$, where $v := \mathrm{lub}(R, D\mathsf{L})$ specifies the least upper bound of all first components of pairs of D and $w := \mathrm{lub}(R, D^{\mathsf{T}}\mathsf{L})$ does the same for the second components. But as we restrict ourselves to finite relations, the following theorem considers the first version of the axiom only.

Theorem 3.3.1 *Given $R : X \leftrightarrow X$ as partial order of a finite lattice $(X, \sqcup, \sqcap)$, $F : X \leftrightarrow X$ is a full implicational system of X iff $FF \subseteq F$, $R^{\mathsf{T}} \subseteq F$ and $F \parallel F \subseteq \mathrm{Sup}(R)\, F\, \mathrm{Sup}(R)^{\mathsf{T}}$.*

If full implicational systems are ordered by inclusion, then the complete lattice induced by $(\mathfrak{O}(X), \leq)$ is isomorphic to the complete lattice induced by $(\mathfrak{F}(X), \subseteq)$, where $\mathfrak{F}(X)$ denotes the set of all full implicational systems of $(X, \sqcup, \sqcap)$. One direction of this isomorphism is given by mapping $C \in \mathfrak{O}(X)$ to the full implicational system $F \in \mathfrak{F}(X)$ that consists of all pairs $\langle x, y \rangle \in X \times X$ with $R_{y,C(x)}$. The converse direction is obtained by mapping $F \in \mathfrak{F}(X)$ to the closure operation $C \in \mathfrak{O}(X)$ such that $C(x) = \bigsqcup\{z \in X \mid F_{x,z}\}$ for all $x \in X$. The following theorem yields these correspondences formulated as a pair of relation-algebraic specifications.

Theorem 3.3.2 *Let $R : X \leftrightarrow X$ be the partial order of a lattice $(X, \sqcup, \sqcap)$. Then, for all $C \in \mathfrak{O}(X)$ the relation $\mathrm{CloToFis}(C) := CR^{\mathsf{T}} : X \leftrightarrow X$ fulfills for all $x, y \in X$ that $\mathrm{CloToFis}(C)_{x,y}$ iff $R_{y,C(x)}$, and, conversely, for all $F \in \mathfrak{F}(X)$ the relation $\mathrm{FisToClo}(F) := \mathrm{lub}(R, F^{\mathsf{T}})^{\mathsf{T}} : X \leftrightarrow X$ fulfills for all $x, y \in X$ that $\mathrm{FisToClo}(F)_{x,y}$ iff $y = \bigsqcup\{z \in X \mid F_{x,z}\}$.*

There is a very close relation between full implicational systems and join-congruence relations, which are generalizations of lattice congruences. Given a lattice $(X, \sqcup, \sqcap)$, a relation $J : X \leftrightarrow X$ is a *join-congruence* of X iff it is an equivalence relation and, in addition for all $x, y, z \in X$ from $J_{x,y}$ it follows that $J_{x \sqcup z, y \sqcup z}$. How to specify equivalence relations with relation-algebraic means is well-known; see e.g., [10]. The remaining requirement on join-congruences holds for J iff $J \parallel \mathsf{I} \subseteq \mathrm{Sup}(R)\, J\, \mathrm{Sup}(R)^{\mathsf{T}}$. This leads to the following result.

Theorem 3.3.3 *Let $R : X \leftrightarrow X$ be the partial order of a lattice $(X, \sqcup, \sqcap)$. Then $J : X \leftrightarrow X$ is a join-congruence of X iff $\mathsf{I} \subseteq J$, $J = J^{\mathsf{T}}$, $JJ \subseteq J$ and $J \parallel \mathsf{I} \subseteq \mathrm{Sup}(R)\, J\, \mathrm{Sup}(R)^{\mathsf{T}}$.*

In the case of a finite lattice $(X, \sqcup, \sqcap)$ there is a one-to-one correspondence between the set $\mathfrak{O}(X)$ of all closure operations of X and the set $\mathfrak{J}(X)$ of all join-congruences of X which again establishes a lattice isomorphism wrt. the lattices induced by the ordered sets $(\mathfrak{O}(X), \leq)$ and $(\mathfrak{J}(X), \subseteq)$. The join-congruence J associated with the closure operation $C \in \mathfrak{O}(X)$ is the kernel of the function C, i.e., we have for all $x, y \in X$ that $J_{x,y}$ iff $C(x) = C(y)$. Relation-algebraically

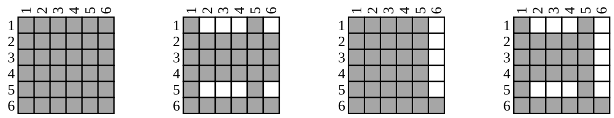

Fig. 6. Full implicational systems F_1 to F_4

this means that $J = \mathrm{CloToJc}(C)$, where $\mathrm{CloToJc}(C) := CC^{\mathsf{T}} : X \leftrightarrow X$. In the reverse direction, the closure operation C is obtained from the join-congruence $J \in \mathfrak{J}(X)$ as in the case of full implicational systems, i.e., by mapping each element $x \in X$ to $\bigsqcup\{z \in X \mid J_{x,z}\}$. Using Theorem 3.3.2 we get $C = \mathrm{JcToClo}(J)$ as $\mathrm{JcToClo}(J) := \mathrm{lub}(R, J)^{\mathsf{T}} : X \leftrightarrow X$ where $R : X \leftrightarrow X$ is the partial order of the finite lattice $(X, \sqcup, \sqcap)$.

Example 3.3.1. In Figure 6 the four full implicational systems F_1 to F_4 of our running example are shown as four RelView-pictures, where the relation F_i is the value of the specification $\mathrm{CloToFis}(C_i)$ with the closure operations C_i from Figure 4, $1 \leq i \leq 4$.

For our running example we already know that exactly 24 equivalence relations $J_i = \mathrm{CloToJc}(C_i)$, $1 \leq i \leq 24$, on X^* are join-congruences. Figure 7 shows the relations J_1 to J_4. Each column (or row) directly corresponds to a congruence class of the respective relation. In addition we used RelView to test which of the 24 join-congruences are also meet-congruences. Analogously, a meet-congruence M satisfies for all $x, y, z \in X$ that $M_{x,y}$ implies $M_{x \sqcap z, y \sqcap z}$. We obtained four positive answers: J_1, J_3, J_{22} (with the classes $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4, x_6\}, \{x_5\}$) and $\mathsf{I} = J_{24}$. The relation J_2, for example, is not a meet-congruence, since $x = z = x_2$ and $y = x_3$ is one of the 12 triples such that $\langle x, y \rangle$ is in J_2 but $\langle x \sqcap z, y \sqcap z \rangle$ is not in J_2.

3.4 Dependency Relations

In [1] Aumann introduced certain relations to formalize the essential properties of a "contact" between objects and sets of objects. His motivation was to obtain an access to topology which is more suggestive for beginners than the ones provided by "traditional" axiom systems. If we formulate Aumann's original definition in our notation, then a relation $D : X \leftrightarrow 2^X$ is a *contact* if 1) for all $x \in X$ and $Y, Z \in 2^X$ we have $D_{x,\{x\}}$, that 2) from $D_{x,Y}$ and $Y \subseteq Z$ it follows $D_{x,Z}$, and that 3) from $D_{x,Y}$ it follows $D_{x,Z}$ if $D_{y,Z}$ holds for all $y \in Y$.

Obviously, demands 1) and 2) are equivalent to the fact that $x \in Y$ implies $D_{x,Y}$ for all $x \in X$ and $Y \in 2^X$. Hence, Aumann's contacts are exactly the

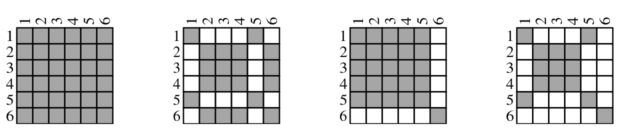

Fig. 7. Join congruences J_1 to J_4

dependency relations in the sense of [6]. In the following theorem, we present relation-algebraic versions of the two axioms given in [6].

Theorem 3.4.1 *Let* $\mathsf{M} : X \leftrightarrow 2^X$ *be a membership-relation. Then a relation* $D : X \leftrightarrow 2^X$ *is a dependency relation iff* $\mathsf{M} \subseteq D$ *and* $D\,\overline{\mathsf{M}^\mathsf{T}\,\overline{D}} \subseteq D$.

In [1] a one-to-one correspondence between the set $\mathfrak{O}(2^X)$ of all closure operations of $(2^X, \subseteq)$ and the set $\mathfrak{D}(X)$ of all contacts of type $[X \leftrightarrow 2^X]$ is established, which is also mentioned in [6] for dependency relations. The relation $D \in \mathfrak{D}(X)$ corresponding to $C \in \mathfrak{O}(2^X)$ is for all $x \in X$ and $Y \in 2^X$ given by $D_{x,Y}$ iff $x \in C(Y)$. Conversely, the closure operation associated with $D \in \mathfrak{D}(X)$ maps $Y \in 2^X$ to $\{x \in X \mid D_{x,Y}\}$. The next theorem contains corresponding specifications in relation algebra.

Theorem 3.4.2 *Assume* $\mathsf{M} : X \leftrightarrow 2^X$ *to be a membership-relation. Then, for all* $C \in \mathfrak{O}(2^X)$ *the relation* $\mathrm{CloToDep}(C) := \mathsf{M}C^\mathsf{T} : X \leftrightarrow 2^X$ *fulfills for all* $x \in X$ *and* $Y \in 2^X$ *that* $\mathrm{CloToDep}(C)_{x,Y}$ *iff* $x \in C(Y)$, *and, conversely, for all* $D \in \mathfrak{D}(X)$ *the relation* $\mathrm{DepToClo}(D) := \mathrm{syq}(D, \mathsf{M}) : 2^X \leftrightarrow 2^X$ *fulfills for all* $Y \in X$ *that* $\mathrm{DepToClo}(D)(Y) = \{x \in X \mid D_{x,Y}\}$.

In [1] Aumann mentions that his relations may also be used to investigate the notion of a contact in sociology or political science. For this, it is frequently necessary to replace M by a relation $M : X \leftrightarrow G$ with the interpretation "individual x is a member of a group g of individuals" of $M_{x,g}$. If $\mathrm{syq}(M, M) = \mathsf{I}$, then (G, R) is an ordered set, where $R := \overline{M^\mathsf{T}\,\overline{M}}$. In this general setting the one-to-one correspondence between closure operations and contacts is lost. It can only be shown that there is an order embedding from the set of closure operations to the set of these generalized contacts.

4 Conclusion

Closure systems and closure operations play an important role in both mathematics and computer science. Moreover, the literature contains many examples for concepts employed in practice which could be proven to be isomorphic to special closure systems. In this work we have presented relation-algebraic formulations of the connections between closure systems on the one hand and closure operations, full implication systems, join-congruences, and dependency relations on the other hand. The resulting algebraic representations are very compact. We have demonstrated that the formulas can directly be used to compute the transformation between concepts, e.g., transform a given finite dependency relation to the corresponding closure operation. Each of the definitions can also be used to efficiently test given relations for conformance, e.g., to compute whether a given relation is a dependency relation or not. We have used the computer algebra system RelView to compute both transformations and tests. As shown by examples, the system can also be used to visualize the results either as graphs or as Boolean matrices, whatever is more appropriate to the case.

A final word about scalability. We have taken care that the presented tests and transformations fit well into the setting of relations implemented using BDDs. As a consequence all of the resulting programs scale well and are also applicable to big relations with ten thousands of elements. The only exception is the enumeration of all possible closure systems on a given set provided with Theorem 3.1.2. As shown in [6] the number of such systems grows too rapidly to be subject to an efficient complete enumeration.

References

1. Aumann G.: Contact relations (in German). Bayerische Akademie der Wissenschaften, Math.-Nat. Klasse Sitzungsberichte 1970, 67-77 (1970).
2. Berghammer R.: Solving algorithmic problems on orders and lattices by relation algebra and RelView. In: Gansha V.G. et al. (eds.): Computer Algebra in Scientific Computing, LNCS 4194, Springer, 49-63 (2006).
3. Berghammer R., Braßel B.: Computing and visualizing closure objects using relation algebra and RelView. In: Gerdt V.P. et al. (eds.): Computer Algebra in Scientific Computing, LNCS 5743, Springer, 29-44 (2009).
4. Berghammer R., Leoniuk B., Milanese U.: Implementation of relation algebra using binary decision diagrams. In: de Swart H. (ed.): Relational Methods in Computer Science, LNCS 2561, Springer, 241-257 (2002).
5. Berghammer R., Neumann F.: RelView– An OBDD-based Computer Algebra system for relations. In: Gansha V.G. et al. (eds.): Computer Algebra in Scientific Computing, LNCS 3718, Springer, 40-51 (2005).
6. Caspard N., Monjardet B.: The lattices of closure systems, closure operators, and implicational systems on a finite set: a survey. Discr. Appl. Math. 127, 241 - 269 (2003).
7. Demetrovics J., Lipkin L.O., Muchnik J.B.: Functional dependencies in relational databases: a lattice point of view. Discr. Appl. Math. 40, 155-185 (1992).
8. Doignon J.P., Falmagne J.C.: Knowledge spaces. Springer (1999).
9. Ganter B., Wille R.: Formal concept analysis, Mathematical foundations. Springer (1998).
10. Schmidt G., Ströhlein T.: Relations and graphs. Discrete Mathematics for Computer Scientists, EATCS Monographs on Theor. Comp. Sci. Springer (1993).
11. Tarski A.: On the calculus of relations. J. Symb. Logic 6, 73-89 (1941).