Acta Informatica (2008) 45:211–236
DOI 10.1007/s00236-008-0072-5

ORIGINAL ARTICLE

Applying relation algebra and RELVIEW to solve problems on orders and lattices

Rudolf Berghammer

Received: 24 April 2007 / Accepted: 6 February 2008 / Published online: 14 March 2008

Abstract Relation algebra is well suited for dealing with many problems on ordered sets. Introducing lattices via order relations, this suggests to apply it and tools for its mechanization for lattice-theoretical problems, too. We combine relation algebra and the BDD-based specific purpose Computer Algebra system RELVIEW to solve some algorithmic problems on orders and lattices and to visualize their solutions.

1 Introduction

An ordered set $(X, \sqsubseteq)$ consists of a non-empty set X and a partial order relation $\sqsubseteq$ on X. It is a *lattice* if every pair of elements $x, y \in X$ has a greatest lower bound $x \sqcap y$ and a least upper bound $x \sqcup y$. Orders and lattices play an important role in many areas of computer science. These include, e.g., replacement systems, knowledge representation, data mining, information retrieval, cryptography and cryptanalysis, static program analysis, logic programming, algorithmics, and models of computation.

Relation algebra [27,33,38] generalizes lattices since it additionally uses complements, compositions with identities, and transpositions. Its use in computer science is mainly due to the fact that many datatypes can be modeled via relations, many problems on them can be naturally specified by relation-algebraic expressions and formulae, and, therefore, many solutions reduce to relation-algebraic computations. Finite relations can be implemented very efficiently. At Kiel University, we have developed a Computer Algebra system for the manipulation and visualization of relations and for relational programming, called RELVIEW [1,8]. It is written in C, runs under Sun-Solaris and Intel-based Linux systems, uses binary decision diagrams (BDDs) for representing relations [6,26,28], and makes full use of the X-windows GUI. The main purpose of RELVIEW is the evaluation of relation-algebraic expressions, which are constructed from the relations of its workspace using pre-defined operations and tests, user-defined relational functions, and user-defined relational programs. Relational functions

R. Berghammer (✉)
Institut für Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany
e-mail: rub@informatik.uni-kiel.de

are defined as is customary in mathematics, where the right-hand sides are relation-algebraic expressions over the relations of the system's workspace and the formal parameters. A relational program in RelView is much like a function procedure in programming languages like Pascal or Modula 2, except the only datatype used is relations.

As demonstrated, e.g., in [33], relation algebra is well suited for dealing with many problems concerning order relations in a component-free manner. Taking ordered sets as a starting point for introducing lattices (instead of algebras having two binary operations $\sqcap$ and $\sqcup$), lattices are nothing else than specific partial order relations. This suggests applying the formal apparatus of relation algebra and tools for its mechanization to lattice-theoretical problems, too. First examples for this approach are [7,8], where relation algebra and the Kiel RelView tool are combined for computing and visualizing cut completions and concept lattices. The material presented in this paper is a continuation of [7,8,33]. Having a Computer Algebra system for computations on orders and lattices is potentially very useful, and in the following we also want to demonstrate that RelView is especially suited to this task. The remainder of the paper is organized as follows. First, we collect some technical preliminaries of relation algebra and the relation-algebraic treatment of orders, extremal elements, and Hasse-diagrams in Sects. 2 and 3. Then, we concentrate in the main part on a series of applications in lattice theory (Sects. 4–8) and the computation of linear extensions (Sect. 9). In doing so, we also want to illustrate the advantages of RelView when using it for visualization purposes. Section 10 contains some concluding remarks.

2 Relational preliminaries

Given two sets $X \neq \emptyset$ and $Y \neq \emptyset$, we write $R : X \leftrightarrow Y$ if R is a (binary) relation with domain X and range Y, i.e., a subset of the direct product $X \times Y$. If the two sets X and Y of R's *type* $X \leftrightarrow Y$ are finite and of size m and n, respectively, we may consider R as a Boolean $m \times n$ matrix. Since this matrix interpretation is well suited for many purposes and also used by RelView as a possibility to visualize relations, in the following we often use matrix terminology and notation. Especially, we speak about entries/components, rows, and columns of a relation/matrix and write $R_{x,y}$ instead of $\langle x, y\rangle \in R$ or $x\ R\ y$. We assume the reader to be familiar with the basic operations on relations, viz. R^{T} (*transposition*), $\overline{R}$ (*complement*), $R \cup S$ (*union*), $R \cap S$ (*intersection*), and RS (*composition*), the predicate $R \subseteq S$ (*inclusion*), and the special relations O (*empty relation*), L (*universal relation*), and I (*identity relation*).

For each type, the set-theoretic operations $\overline{\ }, \cup, \cap$ and the constants O, L form a complete Boolean lattice with the order given by inclusion. Further well-known laws of relations are, e.g., $R^{\mathsf{T}\mathsf{T}} = R$, $Q(R \cap S) \subseteq QR \cap QS$, and $(RS)^{\mathsf{T}} = S^{\mathsf{T}}R^{\mathsf{T}}$. The theoretical framework for such laws holding is that of a relation algebra [27,33,38]. As constants and operations of this algebraic structure, we have those of the set-theoretic relations; its axioms are those of a complete Boolean lattice for $\overline{\ }, \cup, \cap, \subseteq, \mathsf{O}, \mathsf{L}$, associativity of composition with identity relations as neutral elements, the equivalence of $Q^{\mathsf{T}}\overline{S} \subseteq \overline{R}$, $QR \subseteq S$ and $\overline{S}R^{\mathsf{T}} \subseteq \overline{Q}$ (*Schröder equivalences*), and the equivalence of $R \neq \mathsf{O}$ and $\mathsf{L}R\mathsf{L} = \mathsf{L}$ (*Tarski rule*). In later proofs, we shall mention only the latter two axioms and their "non-obvious" consequences like the *Dedekind rule* $QR \cap S \subseteq (Q \cap SR^{\mathsf{T}})(R \cap Q^{\mathsf{T}}S)$. Well-known laws like those presented above or in Sects. 2.1–2.3 of [33] remain unmentioned.

By the definition $syq(R, S) = \overline{R^{\mathsf{T}}\overline{S}} \cap \overline{\overline{R}^{\mathsf{T}}S}$ the so-called *symmetric quotient* $syq(R, S) : Y \leftrightarrow Z$ of two relations $R : X \leftrightarrow Y$ and $S : X \leftrightarrow Z$ is introduced. Many properties of this

construct can be found in [33], for example. In the present paper, we will only use that for all $y \in Y$ and $z \in Z$ the equivalence of $syq(R, S)_{y,z}$ and $\forall\, x : R_{x,y} \leftrightarrow S_{x,z}$ holds.

Given a direct product $X \times Y$ of two sets X and Y, the two projection functions decompose a pair $u = \langle u_1, u_2 \rangle$ into its first component[1] u_1 and its second component u_2. We consider instead of these functions the corresponding *projection relations* $\pi : X \times Y \leftrightarrow X$ and $\rho : X \times Y \leftrightarrow Y$ such that for all $u \in X \times Y$, $x \in X$, and $y \in Y$ we have $\pi_{u,x}$ iff $u_1 = x$ and $\rho_{u,y}$ iff $u_2 = y$. Projection relations enable us to specify the well-known pairing operation of functional programming relation-algebraically as follows: For two relations $R : Z \leftrightarrow X$ and $S : Z \leftrightarrow Y$ with the same domain, we define their *pairing* or *fork* $[R, S] : Z \leftrightarrow X \times Y$ by $[R, S] = R\pi^\mathsf{T} \cap S\rho^\mathsf{T}$. Component-wisely we then have for all $z \in Z$ and $u \in X \times Y$ that $[R, S]_{z,u}$ iff R_{z,u_1} and S_{z,u_2}.

There are some relation-algebraic possibilities to model sets. Our first modeling uses *vectors*, which are relations v with $v = v\mathsf{L}$. Since for a vector the range is irrelevant, we consider in the following mostly vectors $v : X \leftrightarrow \mathbf{1}$ with a specific singleton set $\mathbf{1} = \{\bot\}$ as range and omit in such cases the subscript $\bot$, i.e., write v_x instead of $v_{x,\bot}$. 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 a *point* if $vv^\mathsf{T} \subseteq \mathsf{I}$, i.e., it is *injective*. This means that it represents a singleton subset of its domain or an element from it, if we identify a singleton set $\{x\}$ with the element x. In the Boolean matrix model, hence, a point $v : X \leftrightarrow \mathbf{1}$ is a Boolean column vector in which exactly one entry is 1. For all relations Q, R, S and points p, we have the equations $(Q \cap R)p = Qp \cap Rp$ and $\overline{Rp} = \overline{R}\,p$ and, furthermore, $Sp^\mathsf{T} \subseteq R$ iff $S \subseteq Rp$; see [33].

As a second way to model sets, we will apply the relation-level equivalents of the set-theoretic symbol $\in$, that is, *membership-relations* $\mathsf{M} : X \leftrightarrow 2^X$. These specific relations are defined by demanding for all elements $x \in X$ and sets $Y \in 2^X$ that $\mathsf{M}_{x,Y}$ iff $x \in Y$. A simple Boolean matrix implementation of membership-relations requires an exponential number of bits. However, in [6,26,28] an implementation of $\mathsf{M} : X \leftrightarrow 2^X$ using BDDs is presented, where the number of BDD-vertices is linear in the size of the base set X. This implementation is part of RelView.

Finally, we will use injective functions for modeling sets. Given an injective function $\iota : Y \to X$, we may consider Y as a subset of X by identifying it with its image under ι. If Y is actually a subset of X and ι is given as a relation of type $Y \leftrightarrow X$ such that $\iota_{y,x}$ iff $y = x$ for all $y \in Y$ and $x \in X$, then the vector $\iota^\mathsf{T}\mathsf{L} : X \leftrightarrow \mathbf{1}$ represents Y as a subset of X in the sense above. Clearly, the transition in the other direction is also possible, i.e., the generation of a relation $inj(v) : Y \leftrightarrow X$ from the vector representation $v : X \leftrightarrow \mathbf{1}$ of the subset Y of X such that for all $y \in Y$ and $x \in X$ we have $inj(v)_{y,x}$ iff $y = x$. A combination of such relations with membership-relations allows a *column-wise representation* of sets of subsets. More specifically, if the vector $v : 2^X \leftrightarrow \mathbf{1}$ represents a subset $\mathfrak{S}$ of 2^X in the sense above, then for all $x \in X$ and $Y \in \mathfrak{S}$ we get the equivalence of $(\mathsf{M}\,inj(v)^\mathsf{T})_{x,Y}$ and $x \in Y$. This means that the elements of $\mathfrak{S}$ are represented precisely by the columns of the relation $\mathsf{M}\,inj(v)^\mathsf{T} : X \leftrightarrow \mathfrak{S}$. A further consequence is that $\overline{S^\mathsf{T}\,\overline{S}} : \mathfrak{S} \leftrightarrow \mathfrak{S}$ is the relation-algebraic specification of set inclusion on $\mathfrak{S}$, that is, for all $Y, Z \in \mathfrak{S}$ we have the relationship $(\overline{S^\mathsf{T}\,\overline{S}})_{Y,Z}$ iff $Y \subseteq Z$.

In the standard model of relation algebra, the so-called point axiom of [33] holds. It says that for every relation $R \neq \mathsf{O}$ there exist points p and q such that $pq^\mathsf{T} \subseteq R$. As a consequence, for each vector $v \neq \mathsf{O}$ there exists a point p fulfilling $p \subseteq v$. The choice of an atom $atom(R)$ (i.e., a relation of the form pq^T with points p and q) contained in a

[1] Throughout this paper, we denote the first component of a pair u by u_1 and the second component of u by u_2. We also write $\sqcap u$ instead of $u_1 \sqcap u_2$ and $\sqcup u$ instead of $u_1 \sqcup u_2$.

relation $R \neq \mathsf{O}$ and a point p contained in a vector $v \neq \mathsf{O}$ are fundamental for a relational approach to algorithms. Our demands on $atom(R)$ are $atom(R) \subseteq R$ and that $atom(R)\mathsf{L}$ and $atom(R)^\mathsf{T}\mathsf{L}$ are points. As relation-algebraic axioms for $point(v)$, we demand $point(v) \subseteq v$ and that $point(v)$ is a point. Note that *atom* and *point* are (deterministic, partial) functions in the usual mathematical sense. Each call yields the same object so that, for instance, the equation $atom(R) = atom(R)$ holds for all non-empty relations R. Of course, the above axiomatizations allow different realizations. For instance, the specific implementation of *atom* in RELVIEW uses that the system deals only with finite base sets that are totally ordered by an internal enumeration. A call $atom(R)$ then yields the relation $\{\langle x, y\rangle\}$ that consists of the lexicographically smallest pair $\langle x, y\rangle$ of R.

3 Order relations, extremal elements, and Hasse-diagrams

Given a relation $R : X \leftrightarrow X$, it is rather easy to calculate relation-algebraic specifications of reflexivity, transitivity, and antisymmetry from the usual logical formulations of these properties. As result we obtain that R is a partial order relation and, hence, the pair (X, R) is an ordered set, iff the three inclusions $\mathsf{I} \subseteq R$, $RR \subseteq R$, and $R \cap R^\mathsf{T} \subseteq \mathsf{I}$ hold. Moreover, R is a total order relation iff additionally $R \cup R^\mathsf{T} = \mathsf{L}$ holds, i.e., there are no incomparable elements. Using these relation-algebraic specifications, to be a (totally) ordered set can be easily tested using the RELVIEW tool.

When dealing with ordered sets, one typically investigates extremal elements. Based upon the representation of sets as vectors, in [33] this is done using relation algebra. The descriptions of [33] lead to the following relation-algebraic specifications; they compute for a partial order relation $R : X \leftrightarrow X$ and a vector $v : X \leftrightarrow \mathbf{1}$ vectors of type $X \leftrightarrow \mathbf{1}$, which represent the set of lower bounds (least element, greatest lower bound, and minimal elements, respectively) of the set represented by v:

$$\begin{array}{rclcrcl} lbds(R, v) & = & \overline{\overline{R}\, v} & & glb(R, v) & = & lel(R^\mathsf{T}, lbds(R, v)) \\ lel(R, v) & = & v \cap lbds(R, v) & & min(R, v) & = & v \cap \overline{(R^\mathsf{T} \cap \bar{\mathsf{I}})v} \end{array} \tag{1}$$

Transposing the relation R immediately yields $upds(R, v) = lbds(R^\mathsf{T}, v)$ for the set of upper bounds, $gel(R, v) = lel(R^\mathsf{T}, v)$ for the greatest element, $lub(R, v) = glb(R^\mathsf{T}, v)$ for the least upper bound, and $max(R, v) = min(R^\mathsf{T}, v)$ for the set of maximal elements. Of course, extremal elements may not exist. In this case the relational functions yield the empty vector.

Geometrical representations of orders have been used and investigated by mathematicians and computer scientists for centuries, e.g., for visualization, the discovery of new results, and the construction of counter-examples (see, e.g., [18]). Usually, one draws the *Hasse-diagram* of a partial order relation R, which is the geometrical representation of the (unique) least relation S contained in R such that its reflexive and transitive closure $S^* = \bigcup_{n\in\mathbb{N}} S^n$ equals R. In [2] it is shown that every discrete (especially every finite) partial order relation possesses such a relation (also called the Hasse-diagram)

$$hasse(R) = R \cap \bar{\mathsf{I}} \cap \overline{(R \cap \bar{\mathsf{I}})(R \cap \bar{\mathsf{I}})} \tag{2}$$

that computes the Hasse-diagram of the partial order relation R.

4 From order relations to lattices

Assume $R : X \leftrightarrow X$ to be a partial order relation. The objective of this section is to obtain relation-algebraic specifications of the two lattice operations $\sqcap$ and $\sqcup$ as relations $Inf(R)$ and $Sup(R)$ of type $X \times X \leftrightarrow X$. We start with the development of $Inf(R)$. Assume $u \in X \times X$ and $x \in X$. Then, we have:

$$
\begin{aligned}
& x \text{ is the greatest lower bound of } u \\
\Leftrightarrow\ & R_{x,u_1} \wedge R_{x,u_2} \wedge \forall\, y : R_{y,u_1} \wedge R_{y,u_2} \rightarrow R_{y,x} \\
\Leftrightarrow\ & R_{x,u_1} \wedge R_{x,u_2} \wedge \neg\exists\, y : R_{y,u_1} \wedge R_{y,u_2} \wedge \overline{R}_{y,x} \\
\Leftrightarrow\ & [R, R]_{x,u} \wedge \neg\exists\, y : [R, R]_{y,u} \wedge \overline{R}_{y,x} \\
\Leftrightarrow\ & [R, R]^{\mathsf{T}}_{u,x} \wedge \neg\exists\, y : [R, R]^{\mathsf{T}}_{u,y} \wedge \overline{R}_{y,x} \\
\Leftrightarrow\ & [R, R]^{\mathsf{T}}_{u,x} \wedge (\overline{[R, R]^{\mathsf{T}}\,\overline{R}}\,)_{u,x} \\
\Leftrightarrow\ & ([R, R]^{\mathsf{T}} \cap \overline{[R, R]^{\mathsf{T}}\,\overline{R}}\,)_{u,x}
\end{aligned}
$$

In Boolean matrix terminology, the entry of the relation $[R, R]^{\mathsf{T}} \cap \overline{[R, R]^{\mathsf{T}}\,\overline{R}}$ at position u, x is 1 iff x is the greatest lower bound of the pair u and 0 otherwise. Hence, if we remove the two subscripts u and x from the last expression of this calculation, we get the first of the two relational functions we are looking for as follows:

$$Inf(R) = [R, R]^{\mathsf{T}} \cap \overline{[R, R]^{\mathsf{T}}\,\overline{R}} \tag{3}$$

The relational function $Sup(R) = Inf(R^{\mathsf{T}})$ for specifying the $\sqcup$-operation as relation of type $X \times X \leftrightarrow X$ is an immediate consequence of (3).

Having obtained relation-algebraic specifications for the two lattice operations $\sqcap$ and $\sqcup$, we also are able to test relation-algebraically an ordered set (X, R) to be a lattice. We have that

$$(X, R) \text{ constitutes a lattice } \Leftrightarrow\ \mathsf{L} = Inf(R)\mathsf{L} \text{ and } \mathsf{L} = Sup(R)\mathsf{L}, \tag{4}$$

since the two equations on the right-hand side of (4) exactly describe that the two relations $Inf(R) : X \times X \leftrightarrow X$ and $Sup(R) : X \times X \leftrightarrow X$ are total.

Here it should be remarked that all relation-algebraic specifications we have presented so far and all specifications, functions, and programs we will present in the remainder of the paper can be straightforwardly translated into the programming language of RelView. As an example, the translation of (3) into RelView-code looks as follows:

```
Inf(R) = [R,R]^ & -([R,R]^ * -R)
```

Hence, RelView can be used for dealing with ordered sets and lattices. Animation and visualization is also possible since RelView allows besides fully automatic executions also step-wise executions and offers a representation of relations as directed graphs, too. In addition, sophisticated algorithms for drawing graphs nicely and some possibilities for marking vertices and edges are available.

5 Problems concerning modularity and distributivity

We now pass from general lattices to specific classes and start with *modular lattices*. These are lattices (X, R) such that for all $x, y, z \in X$ the property $R_{x,z}$ implies the *modular equation* $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap z$. (Since $R : X \leftrightarrow X$ is the partial order relation of the lattice,

$R_{x,z}$ corresponds to the usual notation $x \sqsubseteq z$.) Various equivalent formulations to this definition exist. For developing a relational modularity-test, we use in the following (see, e.g., [12,14]) that a lattice (X, R) is modular iff for all $x, y \in X$ the existence of $z \in X$ with $x \sqcap z = y \sqcap z$ and $x \sqcup z = y \sqcup z$ and $R_{x,y}$ imply $x = y$.

Concentrating on the premise of the implication of the just mentioned universal quantification, we can calculate as given below. In doing so, we assume that the pair $\pi, \rho : X \times X \leftrightarrow X$ denotes the projection relations on the direct product $X \times X$ as introduced in Sect. 2. Furthermore, we abbreviate the relations $Inf(R)$ and $Sup(R)$ for the lattice operations $\sqcap$ and $\sqcup$ by I and S, respectively, which yields the equivalence of $\sqcap u = \sqcap v$ and $(II^{\mathsf{T}})_{u,v}$ and the equivalence of $\sqcup u = \sqcup v$ and $(SS^{\mathsf{T}})_{u,v}$. Let $x, y \in X$. Then, we have:

$$
\begin{array}{ll}
 & (\exists\, z : x \sqcap z = y \sqcap z \wedge x \sqcup z = y \sqcup z) \wedge R_{x,y} \\
\Leftrightarrow & (\exists\, u\, \exists\, v : u_1 = x \wedge v_1 = y \wedge u_2 = v_2 \wedge \sqcap u = \sqcap v \wedge \sqcup u = \sqcup v) \wedge R_{x,y} \\
\Leftrightarrow & (\exists\, u : u_1 = x \wedge \exists\, v : v_1 = y \wedge u_2 = v_2 \wedge \sqcap u = \sqcap v \wedge \sqcup u = \sqcup v) \wedge R_{x,y} \\
\Leftrightarrow & (\exists\, u : \pi_{u,x} \wedge \exists\, v : \pi_{v,y} \wedge (\rho\rho^{\mathsf{T}})_{u,v} \wedge (II^{\mathsf{T}})_{u,v} \wedge (SS^{\mathsf{T}})_{u,v}) \wedge R_{x,y} \\
\Leftrightarrow & (\exists\, u : \pi^{\mathsf{T}}_{x,u} \wedge \exists\, v : (\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})_{u,v} \wedge \pi_{v,y}) \wedge R_{x,y} \\
\Leftrightarrow & (\exists\, u : \pi^{\mathsf{T}}_{x,u} \wedge ((\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})\pi)_{u,y}) \wedge R_{x,y} \\
\Leftrightarrow & (\pi^{\mathsf{T}}(\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})\pi)_{x,y} \wedge R_{x,y} \\
\Leftrightarrow & (\pi^{\mathsf{T}}(\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})\pi \cap R)_{x,y}
\end{array}
$$

Because of the last expression of this derivation and the above characterization, we get the following relation-algebraic specification of modularity:

$$\text{Lattice } (X, R) \text{ is modular } \Leftrightarrow\ \pi^{\mathsf{T}}(\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})\pi \cap R \subseteq \mathsf{I} \tag{5}$$

Distributive lattices form a subclass of the modular lattices. They are important in many applications of computer science and mathematics because they can be easily associated with set families. Originally defined by the distributive law $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$ (or $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$) to hold for all $x, y, z \in X$, we apply the following equivalent characterization (see again [12,14]): A lattice (X, R) is distributive iff for all $x, y \in X$ from the existence of $z \in X$ such that $x \sqcap z = y \sqcap z$ and $x \sqcup z = y \sqcup z$ it follows $x = y$. Compared with the above characterization of modularity, only the relationship $R_{x,y}$ is missing within the premise. Hence, an immediate consequence of (5) is the following relation-algebraic specification of distributivity:

$$\text{Lattice } (X, R) \text{ is distributive } \Leftrightarrow\ \pi^{\mathsf{T}}(\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})\pi \subseteq \mathsf{I} \tag{6}$$

Other popular characterizations of modular and distributive lattices are given by forbidden substructures. Here two lattices play a decisive role, viz. the "pentagon lattice" N_5, whose Hasse-diagram is shown as the left one of the RelView-pictures of Fig. 1, and the "diamond lattice" M_3, whose Hasse-diagram is shown as the right RelView-picture of Fig. 1. It is well known (cf. again [12,14]) that a lattice is non-modular iff it contains a sublattice that is isomorphic to the lattice N_5, and a modular lattice is non-distributive iff it contains a sublattice that is isomorphic to the lattice M_3. In the remainder of this section, we show how relation algebra and RelView can be combined to compute for a non-modular lattice (X, R) a sublattice that is isomorphic to N_5 and to visualize its Hasse-diagram in the drawing of (X, R). The technique we use can be applied to a modular and non-distributive lattice, too, to compute and visualize in such a case a sublattice that is isomorphic to M_3.

In what follows, we assume $R : X \leftrightarrow X$ to be the partial order relation of a non-modular lattice (X, R) and $\pi, \rho : X \times X \leftrightarrow X$ to be the projection relations as introduced in Sect. 2. Furthermore, we let I abbreviate $Inf(R)$ and S abbreviate $Sup(R)$. In the first step, we consider

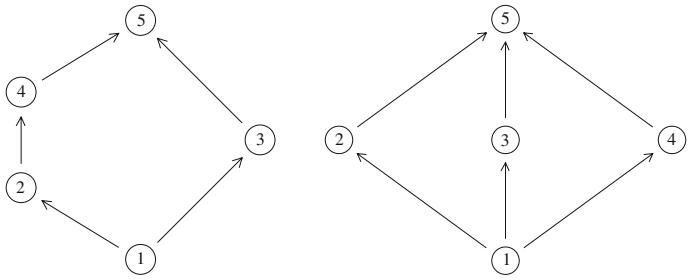

Fig. 1 Hasse diagrams of N_5 and M_3

the relation

$$U = \overline{R \cup R^{\mathsf{T}}} \tag{7}$$

of type $X \leftrightarrow X$ and develop with its help a relation C of the same type such that the equivalence

$$C_{x,y} \;\Leftrightarrow\; \exists\, z : x \sqcap z = y \sqcap z \wedge x \sqcup z = y \sqcup z \wedge U_{x,z} \wedge U_{y,z}$$

holds for all elements $x, y \in X$. Due to definition (7), the relationships $U_{x,z}$ and $U_{y,z}$ of this component-wise specification of C say that the two pairs $\langle x, z\rangle, \langle y, z\rangle \in X \times X$ are incomparable with respect to the partial order relation R. As a consequence, the intersection $C \cap R \cap \bar{\mathsf{I}}$ contains exactly the pairs $\langle x, y\rangle \in X \times X$ that are elements of a sublattice of (X, R) being isomorphic to N_5, such that x corresponds to vertex 2 and y corresponds to vertex 4 of the above RELVIEW-picture of N_5. To obtain a relation-algebraic specification from the component-wise specification of C, we have to modify only the above development of the expression $(\pi^{\mathsf{T}}(\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}})\pi)_{x,y}$ accordingly. The result is as follows:

$$C = \pi^{\mathsf{T}}(\rho\rho^{\mathsf{T}} \cap II^{\mathsf{T}} \cap SS^{\mathsf{T}} \cap \pi U\rho^{\mathsf{T}} \cap \rho U\pi^{\mathsf{T}})\pi \tag{8}$$

Since the lattice (X, R) is assumed to be non-modular, the intersection $C \cap R \cap \bar{\mathsf{I}}$ is non-empty. In the second development step, we first select an atom from $C \cap R \cap \bar{\mathsf{I}}$ by means of the operation *atom* of Sect. 2 and define afterwards with the atom's help two points $p, q : X \leftrightarrow \mathbf{1}$ in the following way:

$$p = atom(C \cap R \cap \bar{\mathsf{I}})\mathsf{L} \qquad q = atom(C \cap R \cap \bar{\mathsf{I}})^{\mathsf{T}}\mathsf{L} \tag{9}$$

If the point p represents the element $x \in X$ and the point q represents the element $y \in X$, then $\langle x, y\rangle$ is a pair from $C \cap R \cap \bar{\mathsf{I}}$. Because of the above described meaning of x and y with respect to the lattice N_5 it suffices to compute a third point, say $r : X \leftrightarrow \mathbf{1}$, that represents the element z of a sublattice of (X, R) we are searching for, such that z corresponds to vertex 3 of the RELVIEW-picture of N_5. The properties z has to fulfill are $U_{x,z}$, $U_{y,z}$, $x \sqcap z = y \sqcap z$, and $x \sqcup z = y \sqcup z$. This leads to

$$c = Up \cap Uq \cap (A \cap II^{\mathsf{T}}B)^{\mathsf{T}}\mathsf{L} \cap (A \cap SS^{\mathsf{T}}B)^{\mathsf{T}}\mathsf{L} \tag{10}$$

as the vector representation $c : X \leftrightarrow \mathbf{1}$ of all candidates for z, where the auxiliary relations $A, B : X \times X \leftrightarrow X$ are defined as follows:

$$A := \pi p\mathsf{L} \cap \rho \qquad B := \pi q\mathsf{L} \cap \rho$$

To verify that the vector $Up : X \leftrightarrow \mathbf{1}$ represents the elements of X that are incomparable to x and the vector $Uq : X \leftrightarrow \mathbf{1}$ does the same for y is trivial. Here is the justification that the vector $(A \cap II^{\mathsf{T}}B)^{\mathsf{T}}\mathsf{L} : X \leftrightarrow \mathbf{1}$ represents the elements $z \in X$ such that $x \sqcap z = y \sqcap z$:

$$\begin{aligned} x \sqcap z = y \sqcap z &\Leftrightarrow \exists\, u : u_1 = x \wedge u_2 = z \wedge \exists\, v : v_1 = y \wedge v_2 = z \wedge (II^{\mathsf{T}})_{u,v} \\ &\Leftrightarrow \exists\, u : (\pi p)_u \wedge \rho_{u,z} \wedge \exists\, v : (\pi q)_v \wedge \rho_{v,z} \wedge (II^{\mathsf{T}})_{u,v} \\ &\Leftrightarrow \exists\, u : (\pi p\mathsf{L} \cap \rho)_{u,z} \wedge \exists\, v : (II^{\mathsf{T}})_{u,v} \wedge (\pi q\mathsf{L} \cap \rho)_{v,z} \\ &\Leftrightarrow \exists\, u : (A \cap II^{\mathsf{T}}B)^{\mathsf{T}}_{z,u} \wedge \mathsf{L}_u \\ &\Leftrightarrow ((A \cap II^{\mathsf{T}}B)^{\mathsf{T}}\mathsf{L})_z \end{aligned}$$

Replacing I by the relation S shows that $(A \cap SS^{\mathsf{T}}B)^{\mathsf{T}}\mathsf{L} : X \leftrightarrow \mathbf{1}$ represents the elements $z \in X$ such that $x \sqcup z = y \sqcup z$. Finally, r is obtained by selecting it as a point from the vector c of (10) using the operation *point* of Sect. 2:

$$r = point(c) \tag{11}$$

The third step consists of the application of the two relational functions *glb* and *lub* of Sect. 3, to get a further vector $v : X \leftrightarrow \mathbf{1}$ as follows:

$$v = p \cup q \cup r \cup glb(R,\, p \cup r) \cup lub(R,\, q \cup r) \tag{12}$$

A little reflection shows that the two elements of X represented by the two points $glb(R,\, p \cup r) : X \leftrightarrow \mathbf{1}$ and $lub(R,\, q \cup r) : X \leftrightarrow \mathbf{1}$ correspond exactly to the vertices 1 and 5, respectively, of the lattice N_5 in the RelView-picture of Fig. 1. Hence, the vector $v : X \leftrightarrow \mathbf{1}$ of (12) represents the carrier set of a "pentagon sublattice" of the lattice (X, R) we are searching for. The partial order relation of this sublattice is immediately obtained (as a subrelation of R to facilitate a RelView-visualization via the marking of edges) as intersection $R \cap vv^{\mathsf{T}} : X \leftrightarrow X$.

It is straightforward to translate the relation-algebraic specifications (7) to (12) into a RelView-program that computes for the input relation R the vector v of (12). The RelView-picture of Fig. 2 demonstrates how then the system can be used for visualization purposes.

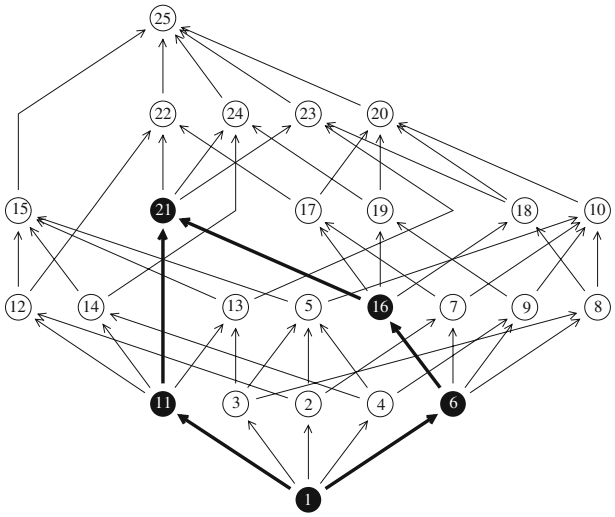

Fig. 2 The non-modular product lattice $N_5 \times M_3$

It shows the graphical representation of the Hasse-diagram of the direct product of the lattices N_5 and M_3. The product-lattice $N_5 \times M_3$ is non-modular. In the graph this fact is visualized by the Hasse-diagram of a sublattice that is isomorphic to N_5, which is emphasized by bold-face edges and black vertices.

6 Investigating pseudo complements

The theory of Boolean algebras is an algebraic version of classical propositional logic. For dealing with intuitionistic propositional logic, (relative) pseudo complements and Heyting algebras (and some variants like Brouwerian lattices [12], pseudo Boolean lattices [21], and semi-Boolean lattices [29]) have been introduced in the following sense: Given a lattice (X, R) and $x, y \in X$, the greatest element of the set $\{z \in X \mid R_{z \sqcap x, y}\}$ is called the *relative pseudo complement* of x with respect to y and denoted by $x * y$ (or $x \to y$). If (X, R) is a distributive lattice with a least element $\bot$ and a greatest element $\top$ and the relative pseudo complement $x * y$ exists for all elements $x, y \in X$, then the lattice is called a *Heyting algebra*. In Heyting algebras $x * \bot$ is defined as the *pseudo complement* x^* of x.

In the following, we consider a lattice (X, R) with least element $\bot \in X$ and greatest element $\top \in X$. Furthermore, using the two relational functions *lel* and *gel* of Sect. 3, we define two points $b, t : X \leftrightarrow \mathbf{1}$ as given below, where L is an universal vector of type $X \leftrightarrow \mathbf{1}$:

$$b = lel(R, \mathsf{L}) \qquad t = gel(R, \mathsf{L})$$

Hence, b represents the least element $\bot$ and t represents the greatest element $\top$ of the lattice. Finally, we assume $\pi, \rho : X \times X \leftrightarrow X$ to be the projection relations on $X \times X$. Our goal is to develop relation-algebraic specifications $RelPcompl(R) : X \times X \leftrightarrow X$ for relative pseudo complements and $Pcompl(R) : X \leftrightarrow X$ for pseudo complements, which allow to compute and visualize these constructions using RelView and which also lead to tests for a lattice to be a Heyting algebra or some of its variants.

We start with the relative pseudo complement and calculate a relation-algebraic specification of the partial function that maps a pair to the relative pseudo complement. Given $u \in X \times X$ and $c \in X$, the component-wise specification of symmetric quotients of Sect. 2 yields

$$c \text{ relative pseudo complement of } u_1 \text{ wrt. } u_2 \;\Leftrightarrow\; syq(S, R)_{u,c},$$

where the auxiliary relation $S : X \leftrightarrow X \times X$ satisfies $S_{z,u}$ iff $R_{z \sqcap u_1, u_2}$ for all $u \in X \times X$ and $z \in X$. It remains to develop a relation-algebraic specification of S. To reach this goal, we calculate for all $u \in X \times X$ and $z \in X$ as follows:

$$\begin{aligned}
R_{z \sqcap u_1, u_2} &\Leftrightarrow \exists\, a : a = z \sqcap u_1 \;\wedge\; R_{a,u_2} \\
&\Leftrightarrow \exists\, v : v_1 = z \wedge v_2 = u_1 \wedge \exists\, a : a = \sqcap v \wedge R_{a,u_2} \\
&\Leftrightarrow \exists\, v : \pi_{v,z} \wedge (\rho\pi^\mathsf{T})_{v,u} \wedge \exists\, a : Inf(R)_{v,a} \wedge (R\,\rho^\mathsf{T})_{a,u} \\
&\Leftrightarrow \exists\, v : \pi_{v,z} \wedge (\rho\pi^\mathsf{T})_{v,u} \wedge (Inf(R)\; R\; \rho^\mathsf{T})_{v,u} \\
&\Leftrightarrow \exists\, v : \pi^\mathsf{T}_{z,v} \wedge (\rho\pi^\mathsf{T} \cap Inf(R)\; R\; \rho^\mathsf{T})_{v,u} \\
&\Leftrightarrow (\pi^\mathsf{T}(\rho\pi^\mathsf{T} \cap Inf(R)\; R\; \rho^\mathsf{T}))_{z,u}
\end{aligned}$$

A removal of the two subscripts u and z from the last expression of this calculation yields the equation $S = \pi^\mathsf{T}(\rho\pi^\mathsf{T} \cap Inf(R)\; R\; \rho^\mathsf{T})$. If we unfold this description of S in $syq(S, R)_{u,c}$ and remove after that the two subscripts u and c, we get the following relation-algebraic

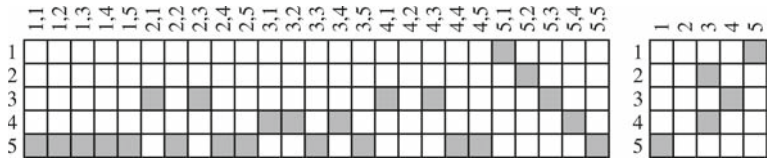

Fig. 3 (Relative) pseudo complement relation of N_5

specification:

$$RelPcompl(R) = syq(\pi^{\mathsf{T}}(\rho\pi^{\mathsf{T}} \cap Inf(R)\, R\, \rho^{\mathsf{T}}),\, R) \tag{13}$$

Pseudo complements are relative pseudo complements with respect to the least element $\bot$. Relation-algebraically this specialization is described by the expression $[\mathsf{I}, \mathsf{L}b^{\mathsf{T}}]$ $RelPcompl(R)$. To enhance efficiency, in the following we develop a specification of the pseudo complement relation $Pcompl(R)$ (of type $X \leftrightarrow X$) that does not use a fork and the relative pseudo complement relation. We start with the same idea as in the case of relative pseudo complements and obtain for all $c, x \in X$ that

$$c \text{ pseudo complement of } x \;\Leftrightarrow\; syq(S, R)_{x,c},$$

where now the auxiliary relation S of type $X \leftrightarrow X$ is component-wisely defined by demanding for all $z, x \in X$ that $S_{z,x}$ iff $R_{z\sqcap x,\bot}$. Similar to the case above, we can show that the equivalence of $R_{z\sqcap x,\bot}$ and $(R^{\mathsf{T}}(R \cap \overline{b}\,\mathsf{L}))_{z,x}$ holds for all $z, x \in X$, yielding the following relational function:

$$Pcompl(R) = syq(R^{\mathsf{T}}(R \cap \overline{b}\,\mathsf{L}),\, R) \tag{14}$$

We have translated (13) and (14) into RelView-code and then applied to the partial order relation of the lattice N_5. The results are shown in the pictures of Fig. 3. On the left, we present the Boolean matrix of the relative pseudo complement relation of N_5. For reasons of space, it is depicted in its transposed form, i.e., with 5 rows and 25 columns. The picture on the right shows the pseudo complement relation of N_5 as Boolean 5×5 matrix. A black square of such a RelView-matrix means that the elements are in relationship (i.e., the entry is 1) and a white square means that they are not (i.e., the entry is 0). So, e.g., from the pictures we see that $\langle 4, 2\rangle$ is the only pair without a relative pseudo complement.

At the end of this section, we want to show that (assuming again b and t as point representations of the least element and the greatest element of the lattice (X, R), respectively, and π and ρ as the projection relations on $X \times X$) the relational function

$$Compl(R) = \pi^{\mathsf{T}}(\rho \cap (Inf(R)\, b \cap Sup(R)\, t)\mathsf{L})\,. \tag{15}$$

for the complement relation $Compl(R) : X \leftrightarrow X$ can be calculated similar to the above specifications. Assuming an arbitrarily chosen pair $u \in X \times X$, we have the following property:

$$\begin{aligned} & u_1 \sqcap u_2 = \bot \wedge u_1 \sqcup u_2 = \top \\ & \quad \Leftrightarrow (\exists\, x : Inf(R)_{u,x} \wedge b_x) \wedge (\exists\, x : Sup(R)_{u,x} \wedge t_x) \\ & \quad \Leftrightarrow (Inf(R)\, b)_u \wedge (Sup(R)\, t)_u \\ & \quad \Leftrightarrow (Inf(R)\, b \cap Sup(R)\, t)_u \end{aligned}$$

Hence, the vector $Inf(R)\, b \cap Sup(R)\, t : X \times X \leftrightarrow \mathbf{1}$ represents the set of pairs $\langle u_1, u_2\rangle$ such that u_2 is the complement of u_1, i.e., it represents the complement relation $\{\langle x.\, \overline{x}\,\rangle \mid x \in X\}$

as subset of $X \times X$. The specification (15) now immediately is obtained by transforming the vector into the corresponding relation of type $X \leftrightarrow X$ following the construction of [33], page 164.

7 Computing and comparing completions

Embeddings into complete lattices play an important role in order and lattice theory and have a lot of applications. In this section, we show how such "completions" can be computed and visualized by combining relation algebra and the RelView tool.

Let us start with completion of ordered sets via cuts. We assume for the following (X, R) to be an ordered set. Then a subset Y of X is called a *cut* of (X, R) if it coincides with the lower bounds of its upper bounds. It is well known (see, e.g., [12,14]) that the set $\mathfrak{C}_R$ of all cuts of (X, R) together with set inclusion constitutes a complete lattice $(\mathfrak{C}_R, \subseteq)$, called the (Dedekind–McNeille) *cut completion* of (X, R) since it contains a suborder that is order-isomorphic to (X, R). This suborder is given by the the set of all *principal cuts* $[x] = \{y \in X \mid R_{y,x}\}$, where $x \in X$. The isomorphism is established via

$$\sigma : X \to \mathfrak{C}_R \qquad \sigma(x) = [x], \tag{16}$$

as this function is an *order-embedding* of (X, R) into $(\mathfrak{C}_R, \subseteq)$, i.e., satisfies $R_{x,y}$ iff $\sigma(x) \subseteq \sigma(y)$ for all elements $x, y \in X$.

Using the two relational operations *syq* and *inj* of Sect. 2 and the membership-relation $\mathsf{M} : X \leftrightarrow 2^X$, in [7] the following three relational functions for computing the cut completion of (X, R) are developed from formal predicate logic specifications:

$$\begin{aligned} \mathit{CutList}(R) &= \mathsf{M}\,\mathit{inj}((\mathit{syq}(\mathsf{M}, \mathit{lbds}(R, \mathit{upds}(R, \mathsf{M}))) \cap \mathsf{I})\mathsf{L})^{\mathsf{T}} \\ \mathit{CutLat}(R) &= \overline{\mathit{CutList}(R)^{\mathsf{T}}\ \overline{\mathit{CutList}(R)}} \\ \mathit{Sigma}(R) &= \mathit{syq}(R, \mathit{CutList}(R)) \end{aligned} \tag{17}$$

The columns of the relation $\mathit{CutList}(R) : X \leftrightarrow \mathfrak{C}_R$ represent the subset $\mathfrak{C}_R$ of 2^X as explained in Sect. 2, the relation $\mathit{CutLat}(R) : \mathfrak{C}_R \leftrightarrow \mathfrak{C}_R$ is the relation-algebraic description of the set inclusion on $\mathfrak{C}_R$ (see again Sect. 2), and the relation $\mathit{Sigma}(R) : X \leftrightarrow \mathfrak{C}_R$ specifies the order-embedding (16) relation-algebraically, that is, for all elements $x \in X$ and cuts $Y \in \mathfrak{C}_R$ we have $\mathit{Sigma}(R)_{x,Y}$ iff $\sigma(x) = Y$.

It is straightforward to translate the relational functions of (17) into the programming language of RelView. This allows to compute and visualize cut completions by means of the system. We demonstrate this in the following, using a small example. To start, consider an ordered set, the graphical representation of which in RelView looks as given in Fig. 4.

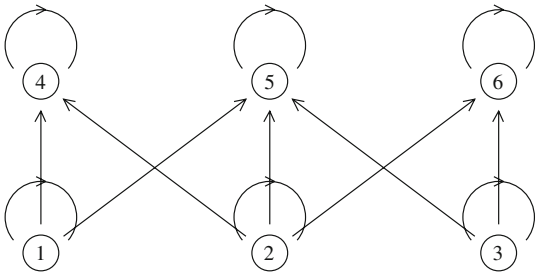

Fig. 4 An ordered set with six elements

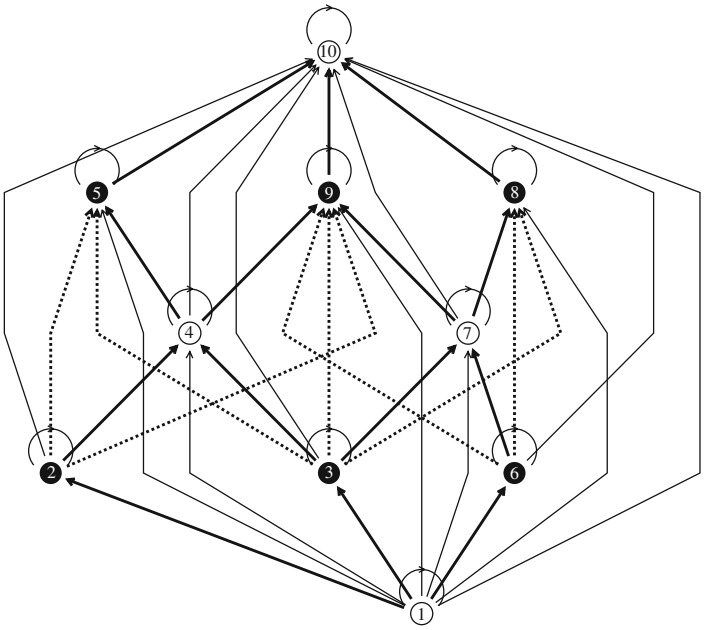

Fig. 5 An order as part of the cut completion

The RELVIEW-picture of Fig. 5 shows the cut completion of this ordered set as a directed graph and emphasizes the completion's Hasse-diagram as bold-face edges. Furthermore, it visualizes the embedding of the ordered set of Fig. 4 into the cut completion by drawing the elements of the suborder that is order-isomorphic to it as black vertices and the Hasse-diagram of the suborder as dotted bold-face edges.

It is also well known since a long time that the cut completion $(\mathfrak{C}_R, \subseteq)$ of an ordered set (X, R) is in a certain sense the least complete lattice containing a suborder that is order-isomorphic to (X, R). Formally, this is expressed by the following fact (see, e.g., [35] for a proof): if the ordered set (X, R) is order-isomorphic to a suborder of a complete lattice (V, Q) via the order-embedding $\phi : X \to V$, then also $(\mathfrak{C}_R, \subseteq)$ is order-isomorphic to a suborder of (V, Q) via the order-embedding

$$\psi : \mathfrak{C}_R \to V \qquad \psi(Y) = \bigsqcup \{\phi(x) \mid x \in Y\} \tag{18}$$

and, furthermore, the equation $\phi(x) = \psi(\sigma(x))$ holds for all $x \in X$. The latter means that ϕ equals the composition of ψ after σ.

As continuation of the work of [7], in the following we show how to calculate a relation-algebraic specification of the order-embedding (18) as a relation of type $\mathfrak{C}_R \leftrightarrow V$ and how afterwards to apply it in the context of another well-known completion procedure. In doing so, we abbreviate the column-wise representation $CutList(R)$ of the set $\mathfrak{C}_R$ of all cuts of (X, R) as C and assume the function $\phi : X \to V$ to be given as a relation $\Phi : X \leftrightarrow V$. The latter means that we assume the equivalence of $\phi(x) = z$ and $\Phi_{x,z}$ for all $x \in X$ and $z \in V$. Then, for all $Y \in \mathfrak{C}_R$ and $z \in V$ we are able to compute as follows, where the equivalence of $y \in Y$ and $C_{y,Y}$ (which we use in the fourth step) follows from the fact that $C : X \leftrightarrow \mathfrak{C}_R$

represents the set $\mathfrak{C}_R$ column-wisely:

$$\begin{aligned}
&\psi(Y) = z \\
&\Leftrightarrow \bigsqcup\{\phi(x) \mid x \in Y\} = z \\
&\Leftrightarrow (\forall\, y : y \in Y \to Q_{\phi(y),z}) \wedge (\forall\, x : (\forall\, y : y \in Y \to Q_{\phi(y),x}) \to Q_{z,x}) \\
&\Leftrightarrow (\neg\exists\, y : y \in Y \wedge \neg Q_{\phi(y),z}) \wedge (\forall\, x : (\neg\exists\, y : y \in Y \wedge \neg Q_{\phi(y),x}) \to Q_{z,x}) \\
&\Leftrightarrow (\neg\exists\, y : C_{y,Y} \wedge \overline{\Phi Q}_{\,y,z}) \wedge (\forall\, x : (\neg\exists\, y : C_{y,Y} \wedge \overline{\Phi Q}_{\,y,x}) \to Q_{z,x}) \\
&\Leftrightarrow \overline{C^{\mathsf{T}}\,\overline{\Phi Q}}_{\,Y,z} \wedge (\forall\, x : \overline{C^{\mathsf{T}}\,\overline{\Phi Q}}_{\,Y,x} \to Q_{z,x}) \\
&\Leftrightarrow \overline{C^{\mathsf{T}}\,\overline{\Phi Q}}_{\,Y,z} \wedge (\neg\exists\, x : \overline{C^{\mathsf{T}}\,\overline{\Phi Q}}_{\,Y,x} \wedge \overline{Q}^{\mathsf{T}}_{x,z}) \\
&\Leftrightarrow (\overline{C^{\mathsf{T}}\,\overline{\Phi Q}})_{Y,z} \wedge (\overline{\overline{C^{\mathsf{T}}\,\overline{\Phi Q}}\;\overline{Q}^{\mathsf{T}}})_{Y,z} \\
&\Leftrightarrow (\overline{C^{\mathsf{T}}\,\overline{\Phi Q}} \cap \overline{\overline{C^{\mathsf{T}}\,\overline{\Phi Q}}\;\overline{Q}^{\mathsf{T}}})_{Y,z}
\end{aligned}$$

Removing the subscripts Y and z from the last expression of the above calculation and replacing the abbreviation C by the function call $CutList(R)$, we arrive at the following relational function for computing the order-embedding $\psi : \mathfrak{C}_R \to V$ of (18):

$$Psi(R, \Phi, Q) = \overline{CutList(R)^{\mathsf{T}}\,\overline{\Phi Q}} \cap \overline{\overline{CutList(R)^{\mathsf{T}}\,\overline{\Phi Q}}\;\overline{Q}^{\mathsf{T}}} \tag{19}$$

To demonstrate an application of (19), we now consider a further completion procedure for embedding an ordered set (X, R) into a complete lattice. As carrier set of the complete lattice we take the set $\mathfrak{I}_R$ of all *order ideals* (also called downsets; cf. [14]) of (X, R), i.e., all subsets I of X such that for all $x \in I$ and $y \in X$ from $R_{y,x}$ it follows $y \in I$. Like cuts, order ideals are ordered by set inclusion. The complete lattice $(\mathfrak{I}_R, \subseteq)$ is called the (order) *ideal completion* (or downset completion) of (X, R). Again a suborder that is order-isomorphic to (X, R) is given by the set of principal cuts—in this context called *principal ideals*—and the isomorphism is established via $\phi : X \to \mathfrak{I}_R$, where $\phi(x) = \{y \in X \mid R_{y,x}\}$.

It is not hard to calculate a vector representation of the subset $\mathfrak{I}_R$ of 2^X. Assuming an arbitrarily chosen set $I \in 2^X$ and $\mathsf{M} : X \leftrightarrow 2^X$ as membership-relation, we have the following equivalence:

$$\begin{aligned}
\forall\, x, y : x \in I \wedge R_{y,x} \to y \in I \;&\Leftrightarrow \forall\, x : x \in I \to \overline{\neg\exists\, y : R_{y,x} \wedge y \notin I} \\
&\Leftrightarrow \forall\, x : \mathsf{M}_{x,I} \to \overline{\overline{\mathsf{M}}^{\mathsf{T}} R}_{\,I,x} \\
&\Leftrightarrow \neg\exists\, x : \mathsf{M}^{\mathsf{T}}_{I,x} \wedge (\overline{\overline{\mathsf{M}}^{\mathsf{T}} R})_{I,x} \\
&\Leftrightarrow \overline{(\mathsf{M}^{\mathsf{T}} \cap \overline{\mathsf{M}}^{\mathsf{T}} R)\mathsf{L}}_{\,I}
\end{aligned}$$

Because of the last expression of this calculation, the vector $\overline{(\mathsf{M}^{\mathsf{T}} \cap \overline{\mathsf{M}}^{\mathsf{T}} R)\mathsf{L}}$ of type $2^X \leftrightarrow \mathbf{1}$ represents the set $\mathfrak{I}_R$ of all order ideals of the ordered set (X, R). The following relational functions for the column-wise representation of the set $\mathfrak{I}_R$, the inclusion order on $\mathfrak{I}_R$, and the order-embedding ϕ from X to $\mathfrak{I}_R$ are immediate consequences of this fact:

$$\begin{aligned}
IdealList(R) &= \mathsf{M}\, inj(\overline{(\mathsf{M}^{\mathsf{T}} \cap \overline{\mathsf{M}}^{\mathsf{T}} R)\mathsf{L}}\,)^{\mathsf{T}} \\
IdealLat(R) &= \overline{IdealList(R)^{\mathsf{T}}\,\overline{IdealList(R)}} \\
Phi(R) &= syq(R, IdealList(R))
\end{aligned} \tag{20}$$

Translating (20) into RelView-code, we have computed the ideal completion of the ordered set of Fig. 4 and have compared it with the cut completion (shown in Fig. 5). The

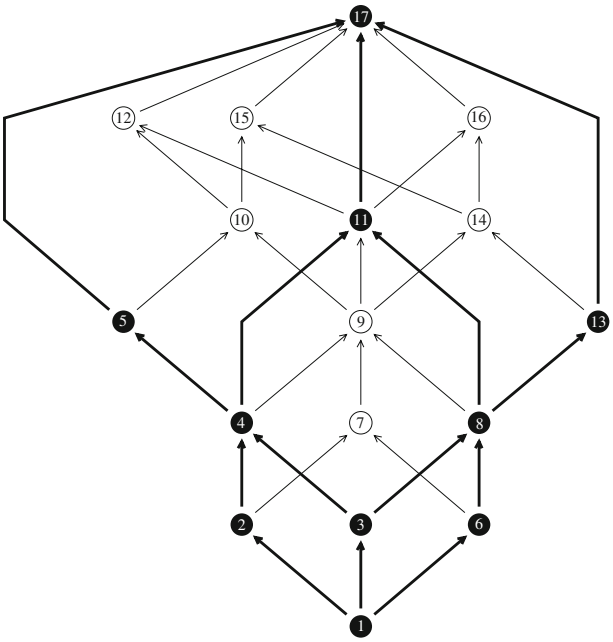

Fig. 6 A cut completion as part of the ideal completion

marked directed graph of Fig. 6 visualizes the containedness of the cut completion in the ideal completion. To obtain this picture, in a first step we have computed the Hasse-diagrams $H_{\mathcal{I}}$ of the ideal completion of the original 6-element ordered set as well as the Hasse-diagrams $H_{\mathcal{C}}$ of its cut completion. Then, we have adapted the type of $H_{\mathcal{C}}$ to the type of $H_{\mathcal{I}}$. In Boolean matrix terminology this means that we have translated the 10×10 matrix $H_{\mathcal{C}}$ into a 17×17 matrix $\tilde{H}_{\mathcal{C}}$ by adding empty rows and columns (i.e., with 0 as entries only) for all elements of the ideal completion not being contained in the cut completion. This easily is possible using the relation-algebraic specification (19) of the order embedding ψ, since $\tilde{H}_{\mathcal{C}} = \Psi^{\mathsf{T}} H_{\mathcal{C}} \Psi$ with Ψ being the relation corresponding to ψ. In the third step, we have joined $H_{\mathcal{I}}$ and $\tilde{H}_{\mathcal{C}}$. Finally, we have drawn this union as a directed graph and marked in the graph the arcs corresponding to the pairs of $\tilde{H}_{\mathcal{C}}$ by boldface arcs and the elements represented by the vector $\Psi^{\mathsf{T}}\mathsf{L}$ as black vertices.

8 Modeling groups and determining subgroup lattices

Let $(G, \cdot, ^{-1}, 1)$ be a group, $\mathfrak{S}_G$ be the set of all subgroups of G, and $\mathfrak{N}_G$ be the set of all normal subgroups of G, Then the ordered sets $(\mathfrak{S}_G, \subseteq)$ and $(\mathfrak{N}_G, \subseteq)$ constitute two lattices, called the *lattice of subgroups*, respectively, the *lattice of normal subgroups* of G. In these lattices $Y \sqcap Z$ corresponds to the intersection $Y \cap Z$ and $Y \sqcup Z$ to the least (normal) subgroup of G containing $Y \cup Z$. The latter construction usually is denoted by $\langle Y \cup Z \rangle$. In the case of normal subgroups Y and Z it can be simplified to the complex product $YZ = \{yz \mid y \in Y, z \in Z\}$ of Y and Z.

Lattices of subgroups and of normal subgroups are powerful tools in group theory. This is mainly due to the fact that many group-theoretical properties are determined by these lattices and vice versa. As a well-known example, we mention a theorem of O. Ore (see [31]) saying

that a group G is locally cyclic (i.e., every finitely generated subgroup is cyclic) iff the lattice of subgroups of G is distributive. A lot of further results in this vein can be found in the monograph [34].

In this section, we show how relation algebra and the RELVIEW tool can be used to compute and visualize lattices of subgroups and of normal subgroups, respectively. In what follows, we assume a group $(G, \cdot, {}^{-1}, 1)$ that is modeled by a multiplication relation $R : G \times G \leftrightarrow G$ for the binary group operation $\cdot : G \times G \to G$, an *inversion relation* $I : G \leftrightarrow G$ for the unary group operation ${}^{-1} : G \to G$, and a *neutral point* $n : G \leftrightarrow \mathbf{1}$ for the neutral element 1 of G, i.e., demand for all $u \in G \times G$ and $x, y \in G$ the equivalences

$$R_{u,x} \Leftrightarrow u_1 u_2 = x \qquad I_{x,y} \Leftrightarrow x^{-1} = y \qquad n_x \Leftrightarrow x = 1$$

to be valid. (As usual in mathematics, we write $u_1 u_2$ instead of $u_1 \cdot u_2$.) Note that the multiplication relation R can be used on its own to model the group G, because the neutral point n relation-algebraically can be specified as

$$n = \overline{\rho^{\mathsf{T}}\,\overline{(\pi \cap R)\mathsf{L}}} \tag{21}$$

and, based on it, the inversion relation becomes

$$I = \pi^{\mathsf{T}}(\rho \cap Rn\mathsf{L}), \tag{22}$$

where $\pi, \rho : G \times G \leftrightarrow G$ are the projection relations on $G \times G$. We prove only the first property (21) and calculate for all $x \in G$ as follows:

$$\begin{aligned}
\overline{\rho^{\mathsf{T}}\,\overline{(\pi \cap R)\mathsf{L}}}_x &\Leftrightarrow \neg\exists\, u : \rho^{\mathsf{T}}_{x,u} \wedge \overline{(\pi \cap R)\mathsf{L}}_u \\
&\Leftrightarrow \forall\, u : \rho_{u,x} \to ((\pi \cap R)\mathsf{L})_u \\
&\Leftrightarrow \forall\, u : u_2 = x \to \exists\, y : \pi_{u,y} \wedge R_{u,y} \wedge \mathsf{L}_y \\
&\Leftrightarrow \forall\, u : u_2 = x \to \exists\, y : u_1 = y \wedge R_{u,y} \\
&\Leftrightarrow \forall\, u : u_2 = x \to u_1 u_2 = u_1
\end{aligned}$$

Hence, $\overline{\rho^{\mathsf{T}}\,\overline{(\pi \cap R)\mathsf{L}}}_x$ iff x is the neutral element of G, and the decided specification follows. A proof of (22) is left to the reader.

As a small example, we consider the well-known *Kleinian group* V_4. Its carrier set consists of four elements e, a, b, c and the group structure is completely determined by demanding that e is the neutral element and that $aa = bb = e$ (see [25] for example). The three pictures of Fig. 7 show the multiplication relation and inversion relation of V_4 as depicted by RELVIEW as Boolean matrices and the point for the neutral element e as depicted by RELVIEW as a Boolean vector. For reasons of space, the multiplication relation is shown in transposed form.

Now, suppose G to be a *finite group* and $Y \in 2^G$ to be arbitrarily chosen. Then the set Y is closed with respect to both group operations iff it is closed with respect to the binary group operation only. This well-known characterization of subgroups of finite groups is the

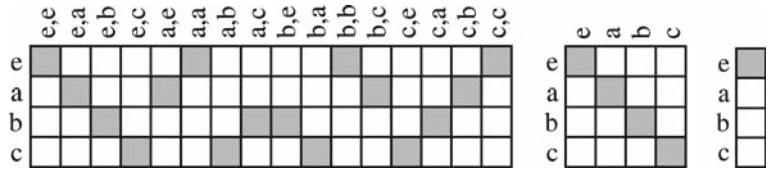

Fig. 7 Relational model of the Kleinian group V_4

starting point of the following calculation, where $\mathsf{M} : G \leftrightarrow 2^G$ is a membership-relation, $\pi, \rho : G \times G \leftrightarrow G$ are the two projection relations on $G \times G$, and two universal vectors of different type are used (in the last expression the L of $\mathsf{M}^\mathsf{T}\mathsf{L}$ has domains 2^X and the L inside the negation has domains $X \times X$):

$$
\begin{array}{ll}
 & Y \text{ is a subgroup} \\
\Leftrightarrow & Y \neq \emptyset \wedge \forall\, u : u_1 \in Y \wedge u_2 \in Y \rightarrow u_1 u_2 \in Y \\
\Leftrightarrow & (\exists\, x : x \in Y) \wedge \forall\, u : u_1 \in Y \wedge u_2 \in Y \rightarrow \exists\, z : u_1 u_2 = z \wedge z \in Y \\
\Leftrightarrow & (\exists\, x : x \in Y) \wedge \forall\, u : u_1 \in Y \wedge u_2 \in Y \rightarrow \exists\, z : R_{u,z} \wedge z \in Y \\
\Leftrightarrow & (\exists\, x : \mathsf{M}_{x,Y} \wedge \mathsf{L}_x) \wedge \forall\, u : (\pi\mathsf{M})_{u,Y} \wedge (\rho\mathsf{M})_{u,Y} \rightarrow \exists\, z : R_{u,z} \wedge \mathsf{M}_{z,Y} \\
\Leftrightarrow & (\mathsf{M}^\mathsf{T}\mathsf{L})_Y \wedge \forall\, u : (\pi\mathsf{M})_{u,Y} \wedge (\rho\mathsf{M})_{u,Y} \rightarrow (R\mathsf{M})_{u,Y} \\
\Leftrightarrow & (\mathsf{M}^\mathsf{T}\mathsf{L})_Y \wedge \neg\exists\, u : (\pi\mathsf{M})_{u,Y} \wedge (\rho\mathsf{M})_{u,Y} \wedge \overline{R\mathsf{M}}_{u,Y} \\
\Leftrightarrow & (\mathsf{M}^\mathsf{T}\mathsf{L})_Y \wedge \overline{\neg\exists\, u : (\pi\mathsf{M} \cap \rho\mathsf{M} \cap \overline{R\mathsf{M}})^\mathsf{T}_{Y,u} \wedge \mathsf{L}_u} \\
\Leftrightarrow & (\mathsf{M}^\mathsf{T}\mathsf{L} \cap \overline{(\pi\mathsf{M} \cap \rho\mathsf{M} \cap \overline{R\mathsf{M}})^\mathsf{T}\mathsf{L}})_Y
\end{array}
$$

If we remove the subscript Y from the last expression of this calculation and apply after that some well-known relation-algebraic laws to transpose[2] only a "row vector" instead of relations of types $G \leftrightarrow 2^G$ and $G \times G \leftrightarrow 2^G$, respectively, we get the following relation-algebraic specification (23) of the vector $SgVect(R) : 2^G \leftrightarrow \mathbf{1}$ representing $\mathfrak{S}_G$ as subset of the powerset 2^G:

$$SgVect(R) = \overline{(\mathsf{L}^\mathsf{T}\mathsf{M} \cap \overline{\mathsf{L}^\mathsf{T}(\pi\mathsf{M} \cap \rho\mathsf{M} \cap \overline{R\mathsf{M}})})^\mathsf{T}} \tag{23}$$

At this place it should be mentioned that in regard to computability by a tool like RelView the restriction $|G| < \infty$ is irrelevant. Its only reason is to simplify the above calculation and to arrive at a more efficient solution.

To obtain a solution for the set $\mathfrak{N}_G$, we additionally demand $Y \in \mathfrak{S}_G$ and calculate as follows:

$$
\begin{array}{ll}
 & Y \text{ is a normal subgroup} \\
\Leftrightarrow & \forall\, u : u_2 \in Y \rightarrow u_1 u_2 u_1^{-1} \in Y \\
\Leftrightarrow & \forall\, u : (\rho\mathsf{M})_{u,Y} \rightarrow \exists\, v : u_1 u_2 = v_1 \wedge u_1^{-1} = v_2 \wedge v_1 v_2 \in Y \\
\Leftrightarrow & \forall\, u : (\rho\mathsf{M})_{u,Y} \rightarrow \exists\, v : R_{u,v_1} \wedge (\pi I)_{u,v_2} \wedge v_1 v_2 \in Y \\
\Leftrightarrow & \forall\, u : (\rho\mathsf{M})_{u,Y} \rightarrow \exists\, v : R_{u,v_1} \wedge (\pi I)_{u,v_2} \wedge \exists\, z : v_1 v_2 = z \wedge z \in Y \\
\Leftrightarrow & \forall\, u : (\rho\mathsf{M})_{u,Y} \rightarrow \exists\, v : R_{u,v_1} \wedge (\pi I)_{u,v_2} \wedge \exists\, z : R_{v,z} \wedge \mathsf{M}_{z,Y} \\
\Leftrightarrow & \forall\, u : (\rho\mathsf{M})_{u,Y} \rightarrow \exists\, v : [R, \pi I]_{u,v} \wedge (R\mathsf{M})_{v,Y} \\
\Leftrightarrow & \forall\, u : (\rho\mathsf{M})_{u,Y} \rightarrow ([R, \pi I]R\mathsf{M})_{u,Y} \\
\Leftrightarrow & \neg\exists\, u : (\rho\mathsf{M})^\mathsf{T}_{Y,u} \wedge \overline{[R, \pi I]R\mathsf{M}}^\mathsf{T}_{Y,u} \\
\Leftrightarrow & \overline{\neg\exists\, u : (\rho\mathsf{M} \cap \overline{[R, \pi I]R\mathsf{M}})^\mathsf{T}_{Y,u} \wedge \mathsf{L}_u} \\
\Leftrightarrow & \overline{(\rho\mathsf{M} \cap \overline{[R, \pi I]R\mathsf{M}})^\mathsf{T}\mathsf{L}}_Y .
\end{array}
$$

A removal of the subscript Y from the last expression of this calculation followed by again some simple transformations to improve efficiency in view of an implementation of relations via BDDs, and an intersection of the result with the vector representation (23) of the set $\mathfrak{S}_G$ leads to the following vector representation $NsgVect(R, I) : 2^G \leftrightarrow \mathbf{1}$ of the set $\mathfrak{N}_G$:

$$NsgVect(R, I) = SgVect(R) \cap \overline{\mathsf{L}^\mathsf{T}(\rho\mathsf{M} \cap \overline{[R, \pi I]R\mathsf{M}})}^\mathsf{T} \tag{24}$$

[2] Using a BDD-implemention of relations, transposition of a relation with domain or range $\mathbf{1}$ only means to exchange domain and range; the BDD remains unchanged. See [28] for details.

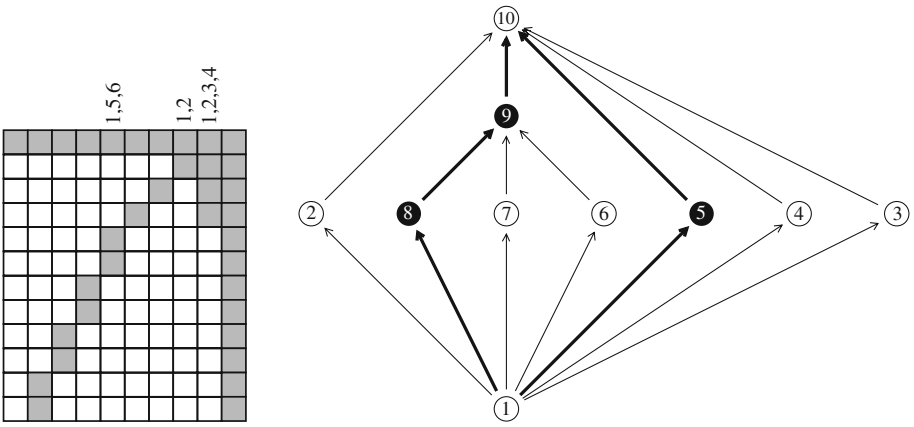

Fig. 8 Enumeration of subgroups and subgroup lattice of A_4

From (23) and (24) column-wise enumerations of the two sets $\mathfrak{S}_G$ and $\mathfrak{N}_G$ by two relations of type $G \leftrightarrow \mathfrak{S}_G$ and $G \leftrightarrow \mathfrak{N}_G$, respectively, and relation-algebraic specifications of set inclusion on $\mathfrak{S}_G$ and on $\mathfrak{N}_G$ (i.e., the partial order relations of the corresponding lattices) immediately are obtained as in the case of cut completion and ideal completion.

Let us consider an example. More than 100 years ago Dedekind proved in [15] that lattices of normal subgroups are modular. But lattices of general subgroups need not be modular. This is e.g., shown in [21] using the lattice of subgroups of the *alternating group* A_4 of the 12 even permutations on the set {1, 2, 3, 4} given in the well-known cycle-notation the following table, where group multiplication means composition of functions:

$$
\begin{array}{llll}
p_1 = () & p_2 = (1\,2)(3\,4) & p_3 = (1\,3)(2\,4) & p_4 = (1\,4)(2\,3) \\
p_5 = (1\,2\,3) & p_6 = (1\,3\,2) & p_7 = (1\,2\,4) & p_8 = (1\,4\,2) \\
p_9 = (1\,3\,4) & p_{10} = (1\,4\,3) & p_{11} = (2\,3\,4) & p_{12} = (2\,4\,3)
\end{array}
$$

We verified this example with the aid of RelView. In doing so, we started with the multiplication tables of A_4 and transformed it into the relational model of A_4 using a small program in a conventional programming language. We loaded the relations of this model (which are too large to present here) from an ASCII-file into RelView and computed the subgroups of A_4 and the partial order relation of the subgroup lattice using the programs resulting from the above specifications.

Figure 8 shows the results of these computations. The Boolean matrix on the left-hand side column-wisely enumerates the 10 subgroups of the alternating group A_4, where row i corresponds to permutation p_i of the above table. So e.g., the first column describes the smallest subgroup $\{p_1\}$ of $(\mathfrak{S}_{A_4}, \subseteq)$. The directed graph on the right-hand side depicts the inclusion order on $\mathfrak{S}_{A_4}$ by means of the Hasse diagram. Additionally we have labeled three columns of the enumeration matrix, where the labels indicate the permutations forming the respective subgroup, have drawn the corresponding nodes of the graph as black circles, and have emphasized the subgraph generated by the black nodes and the nodes 1 (corresponding to $\{p_1\}$) and 10 (corresponding to A_4) by boldface arrows. From the relationships drawn as boldface arrows, we immediately see that the subgraph lattice of A_4 contains a pentagon sublattice. Hence, it is not modular.

9 Computing linear extensions

Almost all relation-algebraic specifications we have developed so far are non-recursive functions in the usual mathematical sense. However, the solutions of many problems require more sophisticated algorithmic principles than non-recursive functions provide. Therefore, as already mentioned in the introduction, besides relational functions the language of RelView allows to formulate relational programs, with all basic constructs of while-programs over the datatype of relations. This conceptual simplicity is very suitable for *formal* development and verification by intertwining relation-algebraic calculations with well-known techniques from computer science. Graph-theoretic applications can be found, e.g., in [3–5]. In the following, we demonstrate this approach by means of an order-theoretic problem. We combine relation algebra with a problem specification via a pre- and postcondition pair and the well-known invariant technique (see [16,20] for example) to develop two relational programs for computing linear extensions.

As input of the programs, we suppose a partial order relation $R : X \leftrightarrow X$ on a *finite* set X. Hence, the precondition $pre(R)$ of both programs consists of the following three formulae:

$$\mathsf{I} \subseteq R \qquad RR \subseteq R \qquad R \cap R^{\mathsf{T}} \subseteq \mathsf{I} \tag{25}$$

The result of both programs is a *linear extension* of R; a total order relation that contains R. If we use in both cases S as variable for storing the result, then in terms of relation algebra the postcondition $post(R, S)$ of both programs we will develop is specified by the following five formulae:

$$R \subseteq S \qquad \mathsf{I} \subseteq S \qquad SS \subseteq S \qquad S \cap S^{\mathsf{T}} \subseteq \mathsf{I} \qquad S \cup S^{\mathsf{T}} = \mathsf{L} \tag{26}$$

Our first relational program is based on an idea from [37]. In this paper Szpilrajn proves that every partial order relation possesses a linear extension. This theorem follows from Zorn's lemma and the fact that, given a partial order relation and an incomparable pair $\langle a, b\rangle$, there exists an extension of the order relation that contains the pair. Using set-theoretic notation, the extension consists of all pairs $\langle x, y\rangle$ such that $\langle x, y\rangle$ is contained in the original order or both pairs $\langle x, a\rangle$ and $\langle b, y\rangle$ are contained in the original order. How to describe the extension of the original order with relation-algebraic means is expressed by the body of the while-loop of the following program:

$$\begin{array}{l} S = R; \\ \textbf{while } S \cup S^{\mathsf{T}} \neq \mathsf{L} \textbf{ do} \\ \quad A = atom(\overline{S \cup S^{\mathsf{T}}}); \\ \quad S = S \cup SAS \textbf{ od} \end{array} \tag{27}$$

To show correctness of (27) w.r.t. the precondition (25) and the postcondition (26), we use the first four formulae of (26) as loop invariant $inv(R, S)$. Then the formulae of the loop invariant together with the exit-condition $S \cup S^{\mathsf{T}} = \mathsf{L}$ of the loop constitute the postcondition. Since $A \subseteq \overline{S \cup S^{\mathsf{T}}} \subseteq \overline{S}$ and $A = \mathsf{I}A\mathsf{I} \subseteq S \cup SAS$ imply that S is strictly enlarged by each loop iteration, from the loop invariant and the finiteness of X we obtain that the program terminates.

Hence, it remains to verify that the loop invariant is established by the initialization and maintained by the body of the loop. The proof obligation $inv(R, R)$ of the initialization $S = R$ coincides with the precondition. To show maintenance of the loop invariant, we assume $inv(R, S)$ and $S \cup S^{\mathsf{T}} \neq \mathsf{L}$ and prove the four formulae of $inv(R, S \cup SAS)$ one after

another, where we use $A = atom(\overline{S \cup S^{\mathsf{T}}})$. The inclusion

$$R \subseteq S \cup SAS$$

follows from $R \subseteq S$. Reflexivity

$$\mathsf{I} \subseteq S \cup SAS$$

is a consequence of $\mathsf{I} \subseteq S$. The third formula, transitivity of $S \cup SAS$, is shown by the calculation

$$\begin{array}{lll}
(S \cup SAS)(S \cup SAS) & & \\
\quad = SS \cup SSAS \cup SASS \cup SASSAS & & \\
\quad \subseteq S \cup SAS \cup SAS \cup SA\mathsf{L}AS & & SS \subseteq S \\
\quad \subseteq S \cup SAS & & A\mathsf{L}A = A,
\end{array}$$

where the equation $A\mathsf{L}A = A$ used in the last step is verified by

$$\begin{array}{lll}
A\mathsf{L}A \subseteq \mathsf{L}A \cap A\mathsf{L} & & \\
\quad \subseteq (\mathsf{L} \cap A\mathsf{L}A^{\mathsf{T}})(A \cap \mathsf{L}A\mathsf{L}) & & \text{Dedekind} \\
\quad \subseteq A & & A\mathsf{L}A^{\mathsf{T}} = A\mathsf{L}(A\mathsf{L})^{\mathsf{T}} \subseteq \mathsf{I}\ (A\mathsf{L}\text{ point}) \\
\quad \subseteq \mathsf{L}A \cap A\mathsf{L} & & \\
\quad \subseteq (\mathsf{L} \cap A\mathsf{L}A^{\mathsf{T}})(A \cap \mathsf{L}A\mathsf{L}) & & \text{Dedekind} \\
\quad \subseteq A\mathsf{L}A. & &
\end{array}$$

To prove antisymmetry of the relation $S \cup SAS$, i.e., the fourth formula of $inv(R, S \cup SAS)$, we start with the equation

$$\begin{array}{l}
(S \cup SAS) \cap (S \cup SAS)^{\mathsf{T}} \\
\quad = (S \cup SAS) \cap (S^{\mathsf{T}} \cup S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}}) \\
\quad = (S \cap S^{\mathsf{T}}) \cup (S \cap S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}}) \cup (SAS \cap S^{\mathsf{T}}) \cup (SAS \cap S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}}).
\end{array}$$

Because of the antisymmetry of the relation S, we are done if the second, third, and fourth expression of the last expression of this equality are empty. Emptiness of $S \cap S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}}$ is shown by the calculation

$$\begin{array}{lll}
S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}} \subseteq S^{\mathsf{T}}\,\overline{S}\,S^{\mathsf{T}} & & A \subseteq \overline{S \cup S^{\mathsf{T}}} \subseteq \overline{S^{\mathsf{T}}} \\
\quad \subseteq \overline{S}\,S^{\mathsf{T}} & & \text{Schröder, } SS \subseteq S \\
\quad \subseteq \overline{S} & & \text{Schröder, } SS \subseteq S
\end{array}$$

and emptiness of $SAS \cap S^{\mathsf{T}}$ follows from this result due to the equality $SAS \cap S^{\mathsf{T}} = (S \cap S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}})^{\mathsf{T}} = \mathsf{O}$. The last case is slightly more complicated. Here we calculate

$$\begin{array}{lll}
SAS \cap S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}} & & \\
\quad \subseteq (S \cap S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}}S^{\mathsf{T}}A^{\mathsf{T}})(AS \cap S^{\mathsf{T}}S^{\mathsf{T}}A^{\mathsf{T}}S^{\mathsf{T}}) & & \text{Dedekind} \\
\quad \subseteq (SASASAS)^{\mathsf{T}} & & SS \subseteq S \\
\quad = \mathsf{O} & & ASA = \mathsf{O}.
\end{array}$$

where the property $ASA = \mathsf{O}$ used in the last step follows from $A \subseteq \overline{S^{\mathsf{T}}}$ in combination with the already shown equation $A\mathsf{L}A = A$. Here is the proof:

$$\begin{array}{lll} A \subseteq \overline{S^{\mathsf{T}}} & & \\ \Leftrightarrow\ A\mathsf{L}A \subseteq \overline{S^{\mathsf{T}}} & & A\mathsf{L}A = A \\ \Leftrightarrow\ A^{\mathsf{T}}\mathsf{L}(A\mathsf{L})^{\mathsf{T}} = A^{\mathsf{T}}\mathsf{L}A^{\mathsf{T}} \subseteq \overline{S} & & \\ \Leftrightarrow\ A^{\mathsf{T}}\mathsf{L} \subseteq \overline{S}\,A\mathsf{L} = \overline{SA\mathsf{L}} & & A\mathsf{L} \text{ point (see Sect. 2)} \\ \Rightarrow\ A^{\mathsf{T}}\mathsf{L} \subseteq \overline{SA} & & \\ \Leftrightarrow\ ASA \subseteq \mathsf{O} & & \text{Schröder} \end{array}$$

This ends the formal verification that the relational program (27) is correct wrt. precondition (25) and postcondition (26).

Computer scientists frequently refer to linear extensions as topological sortings. An efficient algorithm for this task goes back to Kahn [23]. The basic idea is to start with a minimal element x_1 of the given set X. Next, one chooses a minimal element x_2 of the set $X \setminus \{x_1\}$. Continuing until the set becomes empty leads to an enumeration $x_1, x_2, \ldots, x_n$ of X, where $n = |X|$. Finally, the total order relation one is looking for is the reflexive and transitive closure of $\{\langle x_i, x_{i+1}\rangle \mid 1 \leq i \leq n-1\}$. A relational version of this procedure that avoids the intermediate list and directly computes the result S with the help of the relational function *min* of (1) and an auxilioary vector v looks as follows:

$$\begin{array}{l} S = \mathsf{I}; \\ v = \mathsf{O}; \\ \textbf{while } v \neq \mathsf{L} \textbf{ do} \\ \quad p = point(min(R,\ \overline{v}\,)); \\ \quad S = S \cup vp^{\mathsf{T}}; \\ \quad v = v \cup p \ \textbf{od} \end{array} \tag{28}$$

To show correctness of (28) w.r.t. the precondition (25) and the postcondition (26), we generalize the postcondition to a loop invariant $inv(R, S, v)$ by choosing the following formulae:

$$\begin{array}{ccc} R \cap vv^{\mathsf{T}} \subseteq S & \mathsf{I} \subseteq S & SS \subseteq S \\ S \cap S^{\mathsf{T}} \subseteq \mathsf{I} & S \cup S^{\mathsf{T}} = vv^{\mathsf{T}} \cup \mathsf{I} & Rv \subseteq v \end{array} \tag{29}$$

In terms of the above basic idea the vector v represents the set of elements that have already been inserted into the list. Based on this interpretation, the inclusion $R \cap vv^{\mathsf{T}} \subseteq S$ says that S is an extension of the suborder generated by the elements of the list. We leave the translation of $S \cup S^{\mathsf{T}} = vv^{\mathsf{T}} \cup \mathsf{I}$ and $Rv \subseteq v$ into "usual" terminology to the reader.

As in the case of the relational program (27) it is rather trivial to verify that the initialization of S and v establishes the loop invariant (29) and that the loop invariant in combination with the exit condition $v = \mathsf{L}$ of the loop implies the postcondition (26). The definedness of the choice-expression of the body of the loop as well as the termination of the loop follow from the finiteness of the set. X[3] To verify the remaining proof obligation, i.e., that the loop invariant is maintained by the body of the loop, we assume $inv(R, S, v)$, $v \neq \mathsf{L}$, and $p = point(min(R,\ \overline{v}\,))$.

[3] Since the carrier set X is finite, the relation $R^{\mathsf{T}} \cap \overline{\mathsf{I}}$ is progressively finite in the sense of [33]. The latter property means that $w \subseteq (R^{\mathsf{T}} \cap \overline{\mathsf{I}})w$ implies $w = \mathsf{O}$ for all vectors w. Now, contraposition, the choice $w := v$, and (1) show that $\overline{v} \neq \mathsf{O}$ implies $min(R,\ \overline{v}\,) \neq \mathsf{O}$.

To prove $inv(R, S \cup vp^\mathsf{T}, v \cup p)$, we need two preparatory properties. First, we have the inclusion

$$Sv \subseteq (S \cup S^\mathsf{T})v = (vv^\mathsf{T} \cup \mathsf{I})v = vv^\mathsf{T}v \cup v = v$$

due to $vv^\mathsf{T}v \subseteq v\mathsf{L} = v$. The second property is

$$S^\mathsf{T}p \subseteq (S \cup S^\mathsf{T})p = (vv^\mathsf{T} \cup \mathsf{I})p = vv^\mathsf{T}p \cup p \subseteq vv^\mathsf{T}\overline{v} \cup p = p.$$

This calculation uses $p \subseteq min(R, \overline{v}) \subseteq \overline{v}$ and the Schröder equivalences to obtain $v^\mathsf{T}\overline{v} \subseteq \mathsf{O}$ from $v\mathsf{L} \subseteq v$. Inclusion $Sv \subseteq v$ says that the set represented by the vector v is downwards-closed w.r.t. S, and inclusion $S^\mathsf{T}p \subseteq p$ (that actually is an equality due to $p = \mathsf{I}p \subseteq S^\mathsf{T}p$) says that the element represented by p is its only upper bound w.r.t. S.

Here is the verification of the first formula of $inv(R, S \cup vp^\mathsf{T}, v \cup p)$, where the equation used in the last step follows from the Dedekind rule and the inclusion $Rv \subseteq v \subseteq \overline{p}$ of (29) by $pv^\mathsf{T} \cap R \subseteq (p \cap Rv)(v^\mathsf{T} \cap p^\mathsf{T}R) = \mathsf{O}$:

$$\begin{array}{lll}
R \cap (v \cup p)(v \cup p)^\mathsf{T} & & \\
\quad = R \cap (vv^\mathsf{T} \cup vp^\mathsf{T} \cup pv^\mathsf{T} \cup pp^\mathsf{T}) & & \\
\quad = (R \cap vv^\mathsf{T}) \cup (R \cap vp^\mathsf{T}) \cup (R \cap pv^\mathsf{T}) \cup (R \cap pp^\mathsf{T}) & & \\
\quad \subseteq (R \cap vv^\mathsf{T}) \cup vp^\mathsf{T} \cup (R \cap pv^\mathsf{T}) \cup \mathsf{I} & & p \text{ point} \\
\quad \subseteq S \cup vp^\mathsf{T} \cup (R \cap pv^\mathsf{T}) \cup \mathsf{I} & & R \cap vv^\mathsf{T} \subseteq S \\
\quad = S \cup vp^\mathsf{T} \cup (R \cap pv^\mathsf{T}) & & \mathsf{I} \subseteq S \\
\quad = S \cup vp^\mathsf{T} & & R \cap pv^\mathsf{T} = \mathsf{O}
\end{array}$$

Next, we show that $S \cup vp^\mathsf{T}$ is a partial order relation. Reflexivity of this relation follows from $\mathsf{I} \subseteq S$. To shown its transitivity, we calculate:

$$\begin{array}{lll}
(S \cup vp^\mathsf{T})(S \cup vp^\mathsf{T}) & & \\
\quad = SS \cup Svp^\mathsf{T} \cup vp^\mathsf{T}S \cup vp^\mathsf{T}vp^\mathsf{T} & & \\
\quad \subseteq S \cup vp^\mathsf{T} \cup vp^\mathsf{T}S \cup vp^\mathsf{T}vp^\mathsf{T} & & SS \subseteq S, Sv \subseteq v \\
\quad \subseteq S \cup vp^\mathsf{T} \cup vp^\mathsf{T}vp^\mathsf{T} & & p^\mathsf{T}S = (S^\mathsf{T}p)^\mathsf{T} \subseteq p^\mathsf{T} \\
\quad \subseteq S \cup vp^\mathsf{T} & & vp^\mathsf{T}vp^\mathsf{T} \subseteq v\mathsf{L}p^\mathsf{T} = vp^\mathsf{T}
\end{array}$$

The still missing proof of antisymmetry of $S \cup vp^\mathsf{T}$ starts with the following inclusion:

$$\begin{array}{lll}
(S \cup vp^\mathsf{T}) \cap (S \cup vp^\mathsf{T})^\mathsf{T} & & \\
\quad = (S \cap S^\mathsf{T}) \cup (S \cap pv^\mathsf{T}) \cup (S \cap pv^\mathsf{T})^\mathsf{T} \cup (vp^\mathsf{T} \cap pv^\mathsf{T}) & & \\
\quad \subseteq \mathsf{I} \cup (S \cap pv^\mathsf{T}) \cup (S \cap pv^\mathsf{T})^\mathsf{T} \cup (vp^\mathsf{T} \cap pv^\mathsf{T}) & & S \cup S^\mathsf{T} \subseteq \mathsf{I}
\end{array}$$

Since $pv^\mathsf{T} \cap S \subseteq (p \cap Sv)(v^\mathsf{T} \cap p^\mathsf{T}S) = \mathsf{O}$ due to the Dedekind rule and $Sv \subseteq v \subseteq \overline{p}$ and in a very similar way the inclusion $vp^\mathsf{T} \cap pv^\mathsf{T} \subseteq \mathsf{I}$ follows from the Dedekind rule and the point property of p, the desired antisymmetry of $S \cup vp^\mathsf{T}$ holds. A verification of the next formula of $inv(R, S \cup vp^\mathsf{T}, v \cup p)$ is rather straightforward:

$$\begin{array}{lll}
(S \cup vp^\mathsf{T}) \cup (S \cup vp^\mathsf{T})^\mathsf{T} & & \\
\quad = S \cup S^\mathsf{T} \cup vp^\mathsf{T} \cup pv^\mathsf{T} & & \\
\quad = vv^\mathsf{T} \cup \mathsf{I} \cup vp^\mathsf{T} \cup pv^\mathsf{T} & & S \cup S^\mathsf{T} = vv^\mathsf{T} \cup \mathsf{I} \\
\quad = vv^\mathsf{T} \cup vp^\mathsf{T} \cup pv^\mathsf{T} \cup pp^\mathsf{T} \cup \mathsf{I} & & p \text{ point} \\
\quad = (v \cup p)(v \cup p)^\mathsf{T} \cup \mathsf{I} & &
\end{array}$$

Fig. 9 A RELVIEW program for computing linear extensions

```
Szpilrajn(R)
  DECL S, A
  BEG  S = R;
       WHILE -eq(S | S^,L(R)) DO
         A = atom(-(S | S^));
         S = S | S*A*S OD
       RETURN S
  END.
```

Certainly mindful readers have noticed that until now we have used only that the point p is contained in $\overline{v}$. And, interestingly, the property $Rp \subseteq v \cup p$ used in the last step of the proof

$$\begin{array}{lll} R(v \cup p) = Rv \cup Rp & & \\ \subseteq v \cup Rp & & Rv \subseteq v \\ \subseteq v \cup p & & Rp \subseteq v \cup p \end{array}$$

of the last formula of $inv(R, S \cup vp^{\mathsf{T}}, v \cup p)$ is the only place where it is required that p represents a minimal element. From this we namely obtain the auxiliary inclusion of the proof as follows:

$$\begin{array}{lll} p \subseteq min(R, \overline{v}) \Rightarrow p \subseteq \overline{(R^{\mathsf{T}} \cap \bar{\mathsf{l}})\,\overline{v}} & & \\ \Leftrightarrow (R^{\mathsf{T}} \cap \bar{\mathsf{l}})\,\overline{v} \subseteq \overline{p} & & \\ \Leftrightarrow (R \cap \bar{\mathsf{l}})\,p \subseteq v & & \text{Schröder} \\ \Leftrightarrow Rp \cap \bar{\mathsf{l}}\,p \subseteq v & & p \text{ point (see Sect. 2)} \\ \Leftrightarrow Rp \cap \overline{p} \subseteq v & & p \text{ point (see Sect. 2)} \\ \Leftrightarrow Rp \subseteq v \cup p & & \end{array}$$

To give an impression how relational programs look in the programming language of the RELVIEW system, the RELVIEW-version of (27) is shown in Fig. 9. In this RELVIEW-code the predefined operation `eq` tests the equality of relations of the same type and the predefined operation `L` computes for a given relation a universal relation of the same type.

Besides computing a single linear extension of a partial order relation it is also of great interest to generate all linear extensions. For example, a lot of scheduling problems with precedence constraints can be solved by first generating all linear extensions of the given precedence order and then picking a best extension. Knowing all linear extensions is also very important for the investigation and management of distributed systems, since such a system is essentially an ordered set $(E, \rightarrow)$ of events $e \in E$, where $\rightarrow$ denotes Lamport's happened-before relation (see [24]). Obviously, every run of the system corresponds to a linear extension of the partial order relation $\rightarrow$. In the remainder of this section, we sketch how relation algebra and RELVIEW can be combined to generate all linear extensions.

Our approach for obtaining all linear extensions is based on a result of Bonnet and Pouzet (published in [13]) that bijectively links the set of linear extensions of a partial order relation $R : X \leftrightarrow X$ with the set of maximal chains of the lattice $(\mathfrak{I}_R, \subseteq)$ of order ideals. More precisely, if $S : X \leftrightarrow X$ is a linear extension of R, then the set $\mathfrak{I}_S$ of order ideals of (X, S) is a maximal chain of the lattice $(\mathfrak{I}_R, \subseteq)$. On the other hand, if $C \subseteq \mathfrak{I}_R$ is a maximal chain of $(\mathfrak{I}_R, \subseteq)$, then there exists a unique linear extension $S : X \leftrightarrow X$ of R such that $C = \mathfrak{I}_S$.

In what follows, we assume (X, R) to be a finite ordered set and n to be the cardinality of the set X. Then the vector $MaxchainVect(IdealLat(R))$ is of type $2^{\mathfrak{I}_R} \leftrightarrow \mathbf{1}$ and represents the set $\mathfrak{M}_R$ of maximal chains of the ideal completion $(\mathfrak{I}_R, \subseteq)$ of (X, R), where the relational

function[4]

$$Maxchain\ Vect(Q) = gel(\mathsf{S}, (\mathsf{L}^\mathsf{T}\mathsf{M})^\mathsf{T} \cap \overline{\mathsf{L}^\mathsf{T}(\mathsf{M} \cap \overline{Q \cup Q^\mathsf{T}}\,\mathsf{M})}^\mathsf{T}) \tag{30}$$

computes for a partial order relation $Q : Y \leftrightarrow Y$ the vector representation of the set of maximum chains of the ordered set (Y, Q). Using the membership-relation $\mathsf{M} : Y \leftrightarrow 2^Y$, the development of the vector representation

$$(\mathsf{L}^\mathsf{T}\mathsf{M})^\mathsf{T} \cap \overline{\mathsf{L}^\mathsf{T}(\mathsf{M} \cap \overline{Q \cup Q^\mathsf{T}}\,\mathsf{M})}^\mathsf{T} : 2^Y \leftrightarrow \mathbf{1}$$

of the set of chains of (Y, Q) from a formal predicate logic specification of chains is left to the reader. That then $MaxchainVect(IdealLat(R))$ represents the maximal chains of $(\mathfrak{I}_R, \subseteq)$ is due to the fact that each maximal chain C in $(\mathfrak{I}_R, \subseteq)$ can be written as ascending sequence

$$\emptyset \subset \{x_1\} \subset \{x_1, x_2\} \subset \ldots \subset \{x_1, x_2, \ldots, x_n\}$$

of maximum length $n + 1$ that leads from the least element $\emptyset$ to the greatest element X of the ideal completion $(\mathfrak{I}_R, \subseteq)$. The linear extension $S : X \leftrightarrow X$ of the partial order relation R corresponding to the chain C is given as the reflexive and transitive closure of the relation

$$\{\langle x_i, x_{i+1}\rangle \mid 1 \leq i \leq n-1\}.$$

It is an easy exercise to compute this relation (and, hence, also S) from the relation R, the vector representation $c : \mathfrak{I}_R \leftrightarrow \mathbf{1}$ of the chain C, and the partial order relation of the ideal completion by means of a small RelView-program. Hence, all linear extensions of R can be generated by calling this program for all compositions $\mathsf{M}p$ as c, where p ranges over all points contained in the vector $MaxchainVect(IdealLat(R))$ and M is the membership-relation of type $\mathfrak{I}_R \leftrightarrow 2^{\mathfrak{I}_R}$.

10 Conclusion

We have used relation algebra and the specific purpose Computer Algebra system RelView for solving order- and lattice-theoretic problems and for visualizing their solutions. We have demonstrated this fruitful combination by means of some small examples. Space restrictions did not allow to represent larger and more impressive examples for RelView's computational power and visualization possibilities, like Dilworth chain partitions (based upon a maximum bipartite matching RelView-program), the more efficient computation of cut completions and ideal completions as sequences of vectors generated by union or intersection from a given basis (inspired by [30]), the construction of more specific lattices (e.g., lattices of maximum antichains and lattices obtained by doublings), the computation of concept lattices and the clarification of incidence relations in Formal Concept Analysis (see [19]), and the computation or approximation of free lattices from partial order relations on the set of generators (guided by the algorithms presented in [17]). To give at least an impression of the potential of RelView regarding the visualization of such advanced applications, Fig. 10 shows the Dilworth chain partition of a 32-element Boolean lattice. The black vertices of this Hasse-diagram depict a maximum antichain of size $\binom{5}{3} = 10$ (cf. Sperner's theorem [36]) and the boldface arcs represent the 10 chains of the chain partition.

[4] Here S is the so-called *size-comparison-relation* that relates two sets Y and Z iff $|Y| \leq |Z|$. For $\mathsf{S} : 2^X \leftrightarrow 2^X$ an BDD-implementation has been developed in [28] which exactly uses $2 + |X|(|X| + 1)$ BDD-vertices. This implementation is part of RelView.

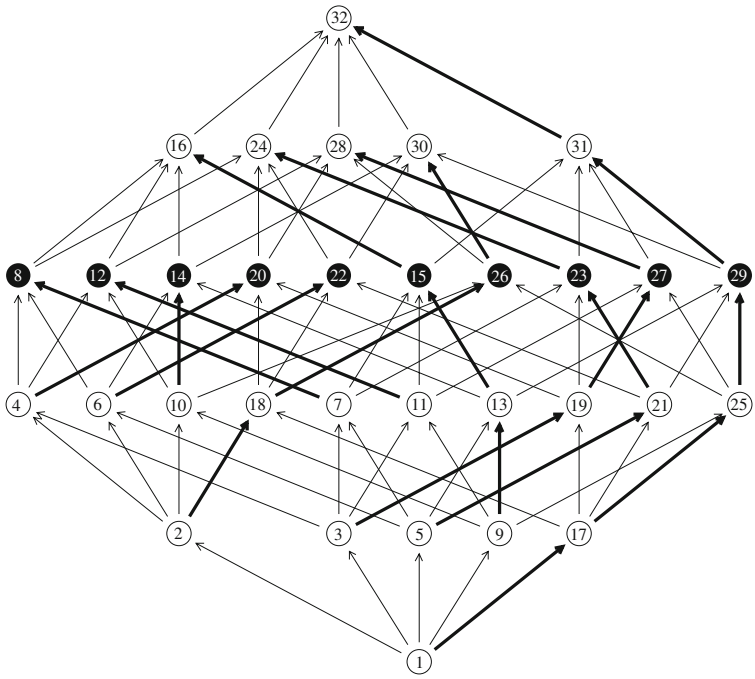

Fig. 10 A Dilworth chain partition

Of course, in spite of the fact that the system implements relations very efficiently with the help of BDDs, frequently RELVIEW-programs cannot compete with special programs tailored for problems of the kind we have considered in the main part of the paper or just have mentioned (cf. e.g., the times of [22] for computing all normal subgroups of some large groups or the times of [32] for generating all linear extensions of fency orders or grid orders)—although in the case of NP-hard problems or problems with a result set of potentially exponential size (like the set of all cuts, all order ideals, all linear extensions, and all extremal chains/antichains) the complexities are usually the same. Nevertheless, a lot of experiments have shown that precisely the computation of huge result sets via membership- and size-comparison-relations is a strength of RELVIEW. To give an impression of the potential of the system and the positive effects of the BDD-implementation of relations in this respect, we mention that (on a Sun-Fire 880 workstation running the Solaris 9 operating system at 750 MHz) the system required 0.05 s to filter out from the 4,096 subsets of the alternating group A_4 the 10 subsets that form subgroups and to compute the partial order relation of the subgroup lattice. As larger application, we mention the counting of linear extensions. We have counted the 1,942,503,128,726 maximal chains of the ideal completion of the product lattice $N_5 \times N_5$ in 330 s, where RELVIEW needed 271 s to determine the Boolean $1{,}184 \times 1{,}184$ matrix of the ideal completion order and the remaining 59 s to compute a Boolean $2^{1184} \times 1$ vector with the above number of 1-entries from this matrix. The reader is e.g., referred to [6–10,26], where further examples of the potential of the RELVIEW tool when dealing with such enumeration and counting tasks are presented.

Nowadays, systematic experiments are accepted as a way for obtaining new mathematical insights. As a consequence, tools for symbolic manipulation, prototypic computation,

animation, and visualization become increasingly important as one proceeds in investigations. We believe that the attraction of RelView in this area lies in its flexibility, its large application area, its computational power when dealing with enumerations of huge sets of "interesting objects" (e.g., to verify an example or to construct a counter-example), its manifold animation and visualization possibilities, and the concise form of its programs. Of course, applications cover all relation-based discrete structures, but also objects which at first glance do not seem to be closely connected to relations. See e.g., [10], where RelView is used as a SAT-solver, or [26], where RelView is applied to compute permanents of 0/1-matrices. New properties and types of and problems on such structures and objects are introduced (discovered and investigated, respectively) all the time and RelView proved to be an ideal tool for experimenting with a lot of the new concepts while avoiding unnecessary overhead. RelView-programs are built very quickly and, combining relation algebra with predicate logic and other formal tools (e.g., assertion logics or fixed point calculus), their correctness is guaranteed by the completely formal developments.

At this place, also the advantages of the system when using it in teaching should be mentioned. We found it very attractive to use RelView for producing good examples. These frequently have been proven for students to be the key of fully understanding an advanced concept. We have further recognized that it is sometimes very helpful to demonstrate how a certain algorithm works. In RelView this is possible by executing computations in a stepwise fashion.

Our present and future investigations pertain two fields. First, we seek possibilities to improve some of the developed algorithms. As an example, we presently investigate a procedure to get the normal subgroups of a group without using a membership-relation, viz. as normal closures of the conjugacy classes. This approach uses that relation algebra easily allows to specify the conjugacy-relation on a group G with multiplication relation $R : G \times G \leftrightarrow G$ as $\rho^{\mathsf{T}}(RR^{\mathsf{T}} \cap \pi\rho^{\mathsf{T}})\pi : G \leftrightarrow G$, where $\pi, \rho : G \times G \leftrightarrow G$ are the projection relations on $G \times G$. Second, we explore new applications of our approach, like the use of orders in preference modeling and multicriteria decision. Here frequently the rearranging of a Boolean matrix into a specific form (like upper triangle block form) seems to be very helpful for fully understanding the meaning of the order relation it visualizes. In [11] first results in this direction can be found. They concern, e.g., the transformation of matrix-representations of semi-order and interval-order relations into staircase form and the computation of interval representations of such relations. Our ultimate goal is to generate a library of RelView-programs for order- and lattice-theoretical tasks that hides most of the relation-algebraic notations and terminology and, therefore, facilitates the use of the tool and its manifold possibilities for people not being specialists in relation algebra.

Acknowledgments I want to express my gratitude to Peter Jipsen, Gunther Schmidt, and Georg Struth for many inspiring discussions and helpful suggestions. I am also grateful to a referee for very detailed and helpful comments.

References

1. Behnke, R., Berghammer, R., Meyer, E., Schneider, P.: RelView—a system for calculation with relations and relational programming. In: Astesiano, E. (ed.) Proceedings of 1st Conference on Fundamental Approaches to Software Engineering, pp. 318–321. LNCS, vol. 1382. Springer, Heidelberg (1998)
2. Berghammer, R., Schmidt, G.: Discrete ordering relations. Discr. Math. **43**, 1–7 (1983)
3. Berghammer, R.: Combining relational calculus and the Dijkstra-Gries method for deriving relational programs. Inf. Sci. **119**, 155–171 (1999)

4. Berghammer, R., Hoffmann, T.: Deriving relational programs for computing kernels by reconstructing a proof of Richardson's theorem. Sci. Comput. Prog. **38**, 1–25 (2000)
5. Berghammer, R., Hoffmann, T.: Relational depth-first-search with applications. Inf. Sci. **139**, 167–186 (2001)
6. Berghammer, R., Leoniuk, B., Milanese, U.: Implementation of relation algebra using binary decision diagrams. In: de Swart, H. (ed.): Proceedings of 6th International Workshop Relational Methods in Computer Science, pp. 241–257. LNCS, vol. 2561. Springer, Heidelberg (2002)
7. Berghammer, R.: Computation of cut completions and concept lattices using relational algebra and RelView. J. Relat. Meth. Comput. Sci. **1**, 50–72 (2004)
8. Berghammer, R., Neumann, F.: RelView—An OBDD-based Computer Algebra system for relations. In: Gansha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) Proceedings of 8th International Workshop Computer Algebra in Scientific Computing, pp. 40–51. LNCS, vol. 3718. Springer, Heidelberg (2005)
9. Berghammer, R., Fronk, A.: Exact computation of minimum feedback vertex sets with relational algebra. Fund. Inform. **70**, 301–316 (2006)
10. Berghammer, R., Milanese, U.: Relational approach to Boolean logic problems. In: MacCaull, W., Dünsch, I., Winter, M. (eds.) Proceedings of 8th International Workshop Relational Methods in Computer Science, pp. 48–59. LNCS, vol. 3929. Springer, Heidelberg (2006)
11. Berghammer, R., Schmidt, G.: Algebraic visualization of relations using RelView. In: Gansha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) Proceedings of 10th International Workshop Computer Algebra in Scientific Computing, pp. 58–72. LNCS, vol. 4770. Springer, Heidelberg (2007)
12. Birkhoff, G.: Lattice theory, 3rd edn. American Math. Society Coll. Publ., Vol. XXV, American Math. Society, Providence (1967)
13. Bonnet, R., Pouzet, M.: Extensions et stratifications d'ensembles disperses. C.R. Acad. Sci. **268**(Serie A), 1512–1515 (1969)
14. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge (1990)
15. Dedekind, R.: Über die von drei Moduln erzeugte Dualgruppe. Math. Ann. **53**, 371–403 (1900)
16. Dijkstra, E.W., Scholten, C.S.: Predicate Calculus and Program Semantics. Springer, Heidelberg (1990)
17. Freese, R., Jezek, J., Nation, J.B.: Free lattices. Mathematical Surveys and Monographs, vol. 42. American Math. Society, Providence (1995)
18. Freese, R.: Automated lattice drawing. In: Eklund, P. (ed.) Proceedings of International Conference on Formal Concept Analysis, pp. 112–127. LNCS, vol. 2961. Springer, Heidelberg (2004)
19. Ganter, B., Wille, R.: Formal concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
20. Gries, D.: The Science of Computer Programming. Springer, Heidelberg (1981)
21. Hermes, H.: Einführung in die Verbandstheorie. Grundlehren der math. Wissenschaften in Einzeldarstellungen, 2nd edn. Springer, Heidelberg (1967)
22. Hulpke, A.: Computing normal subgroups. In: Proc. Int. Symposium on Symbolic and Algebraic Computation, pp. 194–198. ACM Press, New York (1998)
23. Kahn, A.B.: Topological sorting of large networks. Commun. ACM **5**, 558–562 (1962)
24. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Commun. ACM **21**, 558–565 (1978)
25. Lang, S.: Algebra. Graduate Texts in Mathematics, 3rd edn. Springer, Heidelberg (2002)
26. Leoniuk, B.: ROBDD-basierte Implementierung von Relationen und relationalen Operationen mit Anwendungen. Dissertation, Univ. Kiel (2001)
27. Maddux, R.: Relation algebras. Studies in Logic and the Fundations of Mathematics, vol. 150. Elsevier, Amsterdam (2006)
28. Milanese, U.: Zur Implementierung eines ROBDD-basierten Systems für die Manipulation und Visualisierung von Relationen. Dissertation, Univ. Kiel (2003)
29. Nemitz, W.C.: Semi-Boolean lattices. Notre Dame Journal of Formal Logic **16**, 235–238 (1969)
30. Nourine, L., Raynaud, O.: A fast algorithm for building lattices. Inform. Proc. Let. **70**, 259–264 (1999)
31. Ore, O.: Structures and group theory II. Duke Math. J. **4**, 247–269 (1938)
32. Pruesse, G., Ruskey, F.: Generating all linear extensions fast. SIAM J. Comput. **23**, 373–386 (1994)
33. Schmidt, G., Ströhlein, T.: Relations and graphs. Discrete Mathematics for Computer Scientists, EATCS Monographs on Theoret. Comput. Sci., Springer Verlag (1993)
34. Schmidt, R.: Subgroup lattices of groups. de Gruyter Expositions in Mathematics, vol. 14, de Gruyter (1994)
35. Skornjakow, L.A.: Elemente der Verbandstheorie. Akademie-Verlag, Berlin (1973)
36. Sperner, E.: Ein Satz über Untermengen einer endlichen Menge. Math. Zeitschrift **27**, 544–548 (1928)
37. Szpilrajn, E.: Sur l'extension de l'ordre partiel. Fundamenta Math. **16**, 386–389 (1930)
38. Tarski, A.: On the calculus of relations. J. Symbolic Logic **6**, 73–89 (1941)