Computing the Cut Completion of a Partially Ordered Set – An Example for the Use of the RELVIEW-System

Rudolf Berghammer

Fakultät für Informatik
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
D-8014 Neubiberg

Abstract

The RELVIEW-system is a totally interactive and completely video-oriented computer system for the manipulation of concrete relations. This paper contains a brief description of the present Version 2.0 of the RELVIEW-system and describes an application, viz. the computation of the cut completion of a partially ordered set. Also the main topics of future work on RELVIEW are sketched.

1 Introduction

People working with relations (e.g., in the theory of partial orderings, lattice theory, or graph theory) very often use a more or less small example and manipulate it with pencil and paper in order to prove or disprove some property. For supporting such a task by machine (and also since manipulation by hand is no more feasible with bigger examples), at the Universität der Bundeswehr München a computer system – called RELVIEW – for the interactive and video-oriented manipulation of concrete relations has been developed. The system is written in C and is currently available for Sun workstations with American National Standard C and Sunview 4.0. Part of its specification has been based on a prototype written at the TU München. Version 1.0 of RELVIEW is in use since the middle of 1989.

The RELVIEW-system considers relations as Boolean matrices. Such matrices can be drawn and manipulated on the screen using a mouse. Execution of the system's commands is possible by clicking command buttons. Besides some administration commands and the commands realizing the basic operations on relations, the system also provides commands for higher operations and test on relations, and commands which realize operations important in relation algebraic domain description. In its present form (Version 2.0), the system allows also the user to define and apply its own functions on relations.

Since the year 1989 a lot of experiments have been performed with the RELVIEW-system. This paper describes one such experiment, viz. the computation of the cut completion of a partially ordered set. The procedure we will present also shows a nice application of the notion of a symmetric quotient as introduced in [Berghammer et al. 86]

2 Relation Algebraic Preliminaries

In this section we briefly introduce the basic concepts of the algebra of relations to illustrate which kind of operations and tests on relations can be treated by the RELVIEW-system. For more details concerning the algebraic theory of relations, see [Schmidt Ströhlein 89].

2.1 Basic Operations on Relations

If X and Y are sets, then a subset R of the Cartesian product $X \times Y$ is called a relation on X and Y with *domain* X and *range* Y. It is said to be *homogeneous* if $X = Y$. Otherwise we call it *heterogeneous*.

Following the notation of the specification language Z (see [Spivey 89]), we write $R : X \leftrightarrow Y$ if and only if R is a relation on X and Y.

If we consider instead of sets the corresponding characteristic predicates, then a relation R on X and Y becomes a function $R : X \times Y \to \mathbb{B}$, where $\mathbb{B}$ denotes the set $\{0, 1\}$ of truth-values. If X and Y are finite and of cardinality m and n, respectively, then we may consider R as a Boolean matrix with m rows and n columns. This matrix interpretation of relations is well suited for a graphical representation and also used within the RELVIEW-system.

Using Boolean matrix notation, the basic operations on relations may be introduced as follows: Given $R : X \leftrightarrow Y$, $S : X \leftrightarrow Y$, and $T : Y \leftrightarrow Z$, we define

$$\begin{array}{lrr}
\textit{Join}: & R \cup S : X \leftrightarrow Y & (R \cup S)_{xy} :\Longleftrightarrow R_{xy} \vee S_{xy} \\
\textit{Meet}: & R \cap S : X \leftrightarrow Y & (R \cap S)_{xy} :\Longleftrightarrow R_{xy} \wedge S_{xy} \\
\textit{Composition}: & RT : X \leftrightarrow Z & (RT)_{xz} :\Longleftrightarrow \exists\, y \in Y : R_{xy} \wedge T_{yz} \\
\textit{Negation}: & \overline{R} : X \leftrightarrow Y & \overline{R}_{xy} :\Longleftrightarrow \neg R_{xy} \\
\textit{Transposition}: & R^{\mathrm{T}} : Y \leftrightarrow X & R^{\mathrm{T}}_{yx} :\Longleftrightarrow R_{xy}.
\end{array}$$

Considered on a set of relations, negation and transposition are total operations, but join, meet and composition are not. Join and meet are only defined if both relations have the same domain and the same range and composition is only defined if the range of the first argument equals the domain of the second argument.

The empty relation is denoted by O, the universal relation by L, and the identity relation (the equality test on a set) by I. In Boolean matrix notation we have

$$\begin{array}{lrr}
\textit{Empty relation}: & O : X \leftrightarrow Y & O_{xy} :\Longleftrightarrow 0 \\
\textit{Universal relation}: & L : X \leftrightarrow Y & L_{xy} :\Longleftrightarrow 1 \\
\textit{Identity relation}: & I : X \leftrightarrow X & I_{x_1 x_2} :\Longleftrightarrow (x_1 = x_2).
\end{array}$$

The identity relation is the object-level (= relation-level) description of the meta-level symbol =. Finally, the inclusion of relations is introduced as follows: Given two relations $R : X \leftrightarrow Y$ and $S : X \leftrightarrow Y$, we define

$$\textit{Inclusion}: \qquad R \subseteq S \;:\Longleftrightarrow\; \forall\, x \in X, y \in Y : R_{xy} \to S_{xy}.$$

Like join and meet also inclusion is only defined if both relations have the same domain and the same range.

2.2 Special Relations

The basic operations and constants of the last subsection are very helpful for defining simple properties on relations in a component-free manner. In the remainder of the paper we need the following special relations:

Partial orderings: Assume $R : X \leftrightarrow X$ to be a homogeneous relation. R is reflexive if $I \subseteq R$, transitive if $R\,R \subseteq R$, and antisymmetric if $R \cap R^{\mathrm{T}} \subseteq I$. A partial ordering is a reflexive, antisymmetric, and transitive relation. By $R \cap \overline{I}$ we denote the irreflexive part of a partial ordering R, which is transitive and antisymmetric.

Mappings: Let $R : X \leftrightarrow Y$ be a relation. Then R is said to be functional if $R^{\mathrm{T}} R \subseteq I$, and total if $R\,L = L$. As usual, a functional and total relation is said to be a mapping. Following again Z, for a mapping we write $R : X \to Y$ instead of $R : X \leftrightarrow X$. A relation R is injective if R^{T} is functional and surjective if R^{T} total. Usually, an injective and surjective mapping is said to be bijective.

Homomorphisms: Let $R : X \leftrightarrow X$ and $S : Y \leftrightarrow Y$ be two homogeneous relations. A mapping $\Phi : X \to Y$ is called a homomorphism from R to S if $R\,\Phi \subseteq \Phi\,S$ holds. If R and S are partial orderings, then Φ is also called an order homomorphism or monotonic. Clearly, the composition of two homomorphisms is also a homomorphism. A homomorphism Φ from R to S is called an isomorphism between R and S if Φ^{T} is a homomorphism (from S to R), too. An isomorphism Φ between R and S, thus, is a bijective mapping which fulfills the equation $R\,\Phi = \Phi\,S$.

2.3 Description of Subsets with Relation Algebraic Means

Relations can also be used for characterizing the subsets of a given set. In principle, there are two possibilities.

Firstly, vectors can be used for describing subsets. A relation $v : X \leftrightarrow Y$ with $v = v\,L$ is called row constant. This condition means: Whatever set Z and universal relation $L : Y \leftrightarrow Z$ we choose, an element x from X is either in relation $v\,L$ to none of the elements of Z or to all elements of Z. Relations of this kind may be considered as subsets of the set X, predicates on X, or vectors. As for a vector $v : X \leftrightarrow Y$ the set Y is without relevancy, we consider in the following only vectors $v : X \leftrightarrow \mathbf{1}$ with a specific singleton set $\mathbf{1}$ as range and write v_x instead of v_{xj}. Such a vector can be considered as a Boolean matrix with exactly one column, i.e., as a Boolean column vector, and describes the subset $\{x \in X : v_x \Leftrightarrow 1\}$ of X. Every vector $v : X \leftrightarrow \mathbf{1}$ is functional, since $I = L$ for the identity relation $I : \mathbf{1} \leftrightarrow \mathbf{1}$ and the universal relation $L : \mathbf{1} \leftrightarrow \mathbf{1}$.

As in matrix calculus, in the following we write capital letters for arbitrary relations and lower case letters for vectors.

A vector $v : X \leftrightarrow \mathbf{1}$ is said to be a point if and only if it is injective und surjective. Injectivity means that it describes a set containing at most one element and surjectivity means that it describes a set containing at least one element. If $p : X \leftrightarrow \mathbf{1}$ is a point, then its transpose is a (constant) mapping, i.e., $p^{\mathrm{T}} : \mathbf{1} \to X$.

Besides vectors, we have injective mappings as second concept for representing subsets. Given an injective mapping $\imath : Y \to X$, we call Y a subset of X given by $\imath$. Clearly, if Y is a subset of X given by $\imath$, then the vector $\imath^{\mathrm{T}} L : X \leftrightarrow \mathbf{1}$ describes Y in the above sense. Since we deal only with concrete relations, the transition in the other direction, i.e., the construction of an injective mapping $\imath : Y \to X$ from a given vector $v : X \leftrightarrow \mathbf{1}$, is also possible.

2.4 Residuals and Quotients

Residuals are the greatest solutions of certain inclusions. The left residual of S over R (in symbols S/R) is the greatest relation X such that $X\,R \subseteq S$; the right residual of S over R (in symbols $R \setminus S$) is the greatest relation X such that $R\,X \subseteq S$. Both residuals may also be represented using the basic operations. We obtain

$$
\begin{array}{ll}
\textit{Left residual}: & S\,/\,R = \overline{\overline{S}\,R^{\mathrm{T}}} \\
\textit{Right residual}: & R \setminus S = \overline{R^{\mathrm{T}}\,\overline{S}}.
\end{array}
$$

In the following we will frequently need relations which are left and right residuals simultaneously, viz. symmetric quotients. The symmetric quotient $syq(R,S)$ of two relations R and S is defined as the greatest relation X such that $R\,X \subseteq S$ and $X\,S^{\mathrm{T}} \subseteq R^{\mathrm{T}}$. In terms of the basic operations we have

$$
\textit{Symmetric quotient}: \qquad syq(R,S) = \overline{R^{\mathrm{T}}\,\overline{S}} \cap \overline{\overline{R}^{\mathrm{T}}\,S}.
$$

The left residual is only defined if both relations have the same range and the right residual and the symmetric quotient are only defined if both relations have the same domain.

Translating the above relation algebraic expressions into component-wise formulations, the left residual $S/R : X \leftrightarrow Y$ of two relations $S : X \leftrightarrow Z$ and $R : Y \leftrightarrow Z$ fulfills the equivalence

$$
(S/R)_{xy} \Longleftrightarrow \forall\, z \in Z : R_{yz} \to S_{xz}.
$$

Thus, if we consider a relation as a Boolean matrix, then the left residual associates an element $x \in X$ with an element $y \in Y$ if and only if the x-row of S is contained in the y-row of R. Similarly, for two relations $S : Z \leftrightarrow X$ and $R : Z \leftrightarrow Y$ the right residual $R \setminus S : Y \leftrightarrow X$ fulfills the equivalence

$$
(R \setminus S)_{yx} \Longleftrightarrow \forall\, z \in Z : S_{zx} \to R_{zy}
$$

and the symmetric quotient $syq(R,S) : X \leftrightarrow Y$ fulfills the equivalence

$$
syq(R,S)_{xy} \Longleftrightarrow \forall\, z \in Z : S_{zx} \leftrightarrow R_{zy}.
$$

Using again Boolean matrix terminology for R and S, then the right residual associates an element $x \in X$ with an element $y \in Y$ if and only if the x-column of S is contained in the y-column of R and the symmetric quotient associates $x \in X$ with $y \in X$ if and only if these two columns coincide.

3 A Short Look at the RELVIEW-System

This section contains a brief description of the present version of the RELVIEW-system. For details we refer to [Abold et al. 89] and [Landgraf Dombetzki 91].

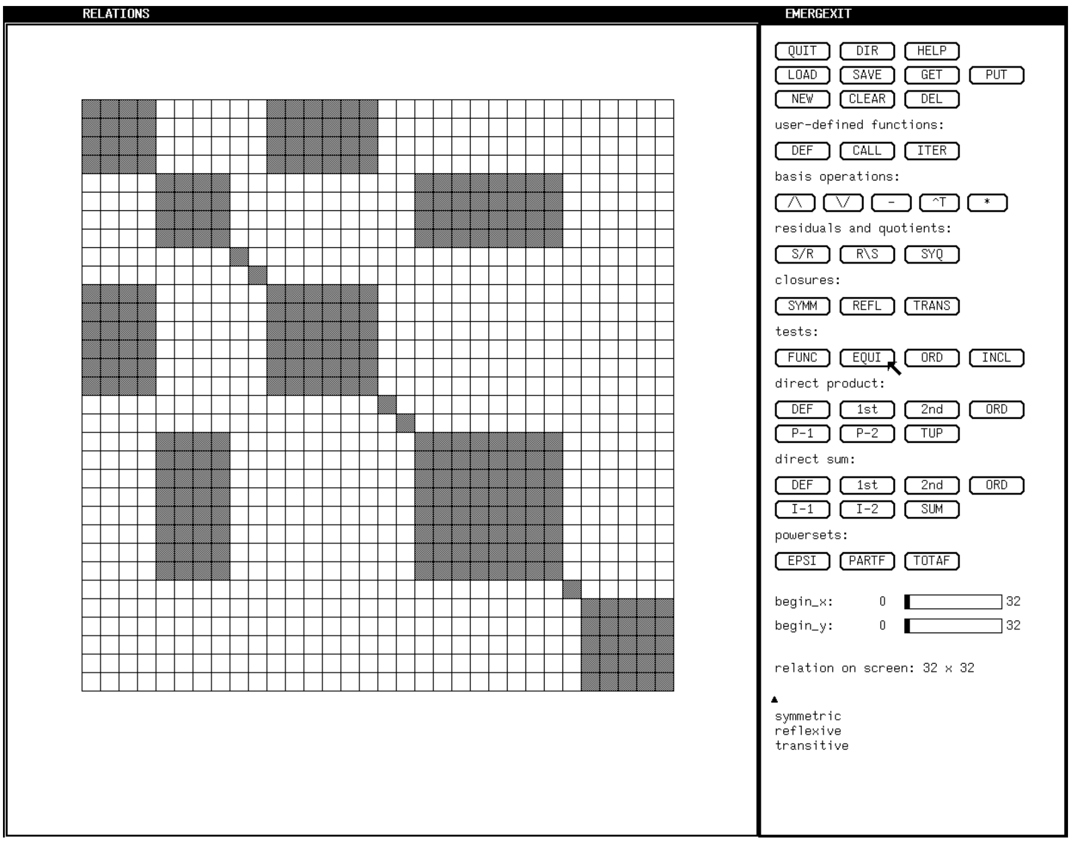
3.1 Overview

As already mentioned, the RELVIEW-system is a totally interactive and completely video-oriented computer system for the manipulation of concrete relations. The main components of the RELVIEW code are window handling, transition of the screen pictures into the logical structure of a relation, implementation of operations on relations, I/O commands, and management of relations and relational domains. The latter components are completely written in ordinary C. Only the window handling uses specific commands from the Sunview system for the generation and the management of windows, the realization of the command buttons and the scrollbars, and for event handling (management of mouse and key inputs).

A particularity of RELVIEW is that the graphical part uses only one Sunview command. The graphical manipulation of a relation is not immediately done on the screen but within a virtual rectangle of the working memory. As a consequence, the entire graphic uses only that Sunview command which draws a virtual rectangle onto the screen and reads the screen into a virtual rectangle, respectively. Besides gain of efficiency, the system's graphical part, thus, becomes more easily transferable to another window system. Meanwhile RELVIEW is also available for the X Window System.

3.2 The Graphical User Interface

The screen of Version 2.0 of RELVIEW looks as follows:



The left part of the screen is the drawing-window. Using the mouse pointer, here the user can edit relations as Boolean matrices. One relation, the so-called *working copy*, is displayed on the screen for editing. A whole collection of relations can be kept in the working memory during a working session. Such a collection may also be saved on permanent memory, e.g., on a hard disk. If a stored relation from the memory is displayed into the drawing-window for editing, a duplicate working copy is created. Editing with the mouse does only affect the working copy and thus does not change the original. To overwrite the original by the (changed) working copy a specific command has to be used.

The right part of the screen of RELVIEW contains the command buttons and the scrollbars. The scrollbars can be used for showing a part of a relation the size of which exceeds the maximal window size (which at present is 256 rows and columns). Also textual input (e.g., dimensions or names of relations) and output (e.g., results of tests, error messages) is requested and shown, respectively, in this part. In the above figure the output indicates that the relation of the drawing-window is an equivalence relation.

Execution of a command consists in the choice of the respective command button (using the mouse pointer), pressing the left mouse button, and, if necessary, the input of the arguments. The command buttons of row 1 through 3 serve only for management. In the next row 4 one finds the command buttons for the definition and application on user-declared functions on relations. The remaining rows contain the command buttons for the operations and tests on relations: Row 5 contains the buttons for the basic operations on relations. In row 6 we have the command buttons for the two residuals and the symmetric quotient. The command buttons for computing closures are listed in row 7. Rows 8 contains the buttons for certain tests. And, finally, the rows 9 through 13 contain the buttons which correspond to a relational description of domains. The buttons of rows 9 and 10 are for the direct product, those of rows 11 and 12 for the direct sum, and those of row 13 are devoted to powersets and function domains.

There are four kinds of input, viz. names (identifiers) of relations, integers, file names and relational terms. The identifiers are strings of at most 12 characters, where the first character has to be alphanumeric, the remaining characters can be chosen arbitrarily with letters, digits or the underscore-character. A relational term is built up with the help of identifiers (denoting the collection of relations in the working memory), the operations of row 5 and 6, and parentheses. Blanks in terms are not allowed! The representation of the operations of Subsection 2.1 and Subsection 2.3 in RELVIEW and their priority are:

		Priority
Join :	`R|S`	5
Meet :	`R&S`	5
Composition :	`R*S`	4
Negation :	`-R`	1
Transposition :	`R^`	2
Left residual :	`S/R`	6
Right residual :	`R\S`	6
Symmetric quotient :	`syq(R,S)`	3

The operations with higher priority bind more tightly and the parentheses around the operations can be dropped.

After clicking the `DEF` button (row 4), the user can declare a new function. The present Version 2.0 of RELVIEW only allows the declaration of unary functions, where the symbol % stands for the (abstracted) variable. The arguments of `DEF` are the name and the body of the function, e.g:

$$\texttt{Hasse} \qquad (\% \,\&\, -\texttt{I}) \,\&\, -\, ((\% \,\&\, -\texttt{I}) * (\% \,\&\, -\texttt{I}))$$

These two arguments declare a RELVIEW-function Hasse which computes, for the set X being finite, the so-called Hasse-diagram $H_R := (R \cap \overline{I}) \cap \overline{(R \cap \overline{I})\,(R \cap \overline{I})}$ of the irreflexive part of a partial ordering $R : X \leftrightarrow X$, that is the smallest relation which is contained in $R \cap \overline{I}$ and has the same transitive closure as $R \cap \overline{I}$. (See also [Berghammer Schmidt 83].)

The button CALL (row 4) applies a user-defined function F to an argument R, i.e., yields $F(R)$. If only one argument is given, then the matrix of the drawing-window is taken as the second one. In the same way, the button ITER (row 4) computes the sequence R, $F(R)$, $F^2(R)$, ... of relations and yields $F^i(R)$ if $F^{i+1}(R) = F^i(R)$, i.e., if the sequence becomes stationary. For instance, if $R : X \leftrightarrow X$ is a homogeneous relation and one declares a function

$$\texttt{Initial} \qquad -(\texttt{R} * -\%),$$

then ITER applied to this function and the empty vector $O : X \leftrightarrow \mathbf{1}$ yields a vector. It describes the subset of the domain X of R containing exactly those elements from which only paths of finite length emerge. (This subset is also called the initial part of a relation, cf. [Schmidt Ströhlein 89], Section 6.3.)

With the exception of HELP (row 1), every command is blocking, i.e., during a command execution no further execution can be started. If a command requires arguments, then execution starts not before the last argument is given. Thus, if the user inadvertently has chosen a wrong button, undo consists in choosing the correct button – provided the argument input yet not has been finished.

A detailed description of how to draw on the drawing-window, how to use the scrollbars, and how to execute a command (inclusive parameter passing and result delivery) is given in [Abold et al. 89] and [Landgraf Dombetzki 91]. The first report also presents some implementation details, e.g., the internal representation of relations, and outlines fast algorithms for computing products, symmetric quotients, and residuals of relations.

4 Computing the Cut Completion Using RELVIEW

In classical mathematics, the method of Dedekind cuts in the rational numbers $\mathbb{Q}$ is one of the ways for introducing the real numbers $\mathbb{R}$. It has been generalized to a procedure for constructing completions of partially ordered sets. In this section we show how such a cut completion can be treated in the calculus of relations and computed using the RELVIEW-system. To obtain this, we have to distinguish between the meta-level symbols $\in$ and $\subseteq$ and the is-element-of relation and the set inclusion relation on the object-level. In the sequel, we use on the object level the relations $\epsilon : X \leftrightarrow 2^X$ and $\sqsubseteq : 2^X \leftrightarrow 2^X$.

4.1 Cut Completion of a Poset

Assume $R : X \leftrightarrow X$ to be a partial ordering. Then we call the pair (X, R) a partially ordered set, *poset* in short. If, in addition, every subset $Y \subseteq X$ has a least upper bound and a greatest lower bound, then (X, R) is called a *complete lattice*. We assume the reader to be familiar with the basic notions of posets and lattice theory. Details can be found in the textbooks [Hermes 67] or [Erné 82], for example.

Now, the method of Dedekind cuts for constructing completions of posets is as follows: Let (X, R) be a poset. One considers two functions on the powerset 2^X of X, viz.

$$\begin{array}{ll} Ma : 2^X \to 2^X & Ma(Y) := \{x \in X : \text{x is an upper bound of Y}\} \\ Mi\ : 2^X \to 2^X & Mi(Y)\ := \{x \in X : \text{x is a lower bound of Y}\}. \end{array}$$

Then one defines an element $M \in 2^X$ to be a *cut* if and only if $Mi(Ma(M)) = M$. Obviously, for each element $p \in X$ the set

$$(p) := \{x \in X : R_{xp} \Leftrightarrow 1\}$$

is a cut, called the *principal cut generated by p*. Let $\mathcal{C}_X$ denote the set of cuts of X and $\mathcal{P}_X$ denote the set of principal cuts of X. Furthermore, let $\sqsubseteq_{\mathcal{C}} : \mathcal{C}_X \leftrightarrow \mathcal{C}_X$ and $\sqsubseteq_{\mathcal{P}} : \mathcal{P}_X \leftrightarrow \mathcal{P}_X$ denote the restrictions of set inclusion $\sqsubseteq : 2^X \leftrightarrow 2^X$ to the cuts and principal cuts, respectively. Then $(\mathcal{C}_X, \sqsubseteq_{\mathcal{C}})$ is a complete lattice, having $(\mathcal{P}_X, \sqsubseteq_{\mathcal{P}})$ as sub-poset. Furthermore, the mapping

$$e : X \to \mathcal{C}_X \qquad e(x) := (x)$$

is an injective order homomorphism from R to $\sqsubseteq_{\mathcal{C}}$. The lattice $(\mathcal{C}_X, \sqsubseteq_{\mathcal{C}})$ is said to be the *cut completion* of the poset (X, R). If the poset (X, R) is even a lattice, then $(\mathcal{P}_X, \sqsubseteq_{\mathcal{P}})$ is a sub-lattice of $(\mathcal{C}_X, \sqsubseteq_{\mathcal{C}})$ and e is an injective lattice homomorphism.

4.2 A Relational Algebra Approach to Cut Completion

Throughout this subsection we assume a fixed poset (X, R). We want to construct the cut completion of (X, R) with relation algebraic means. This construction consists of several steps, each of which easily can be performed within the RELVIEW-system, as will be shown in this subsection, too.

A relation algebraic construction of the cut completion $(\mathcal{C}_X, \sqsubseteq_{\mathcal{C}})$ of the poset (X, R) and the injective order homomorphism e proceeds as follows:

(1) If we describe a subset of X by a vector $v : X \leftrightarrow \mathbf{1}$ in the sense of Subsection 2.3, then $\overline{\overline{R}^{\mathsf{T}} v}$ is the vector of the upper bounds of v and $\overline{\overline{R}\, v}$ is the vector of the lower bounds of v. E.g., in the first case this can be seen from the following derivation: For all $x \in X$

$$\begin{aligned} (\overline{\overline{R}^{\mathsf{T}} v})_x &\Longleftrightarrow \neg(\exists\, y \in X : \overline{R}^{\mathsf{T}}_{xy} \wedge v_y) \\ &\Longleftrightarrow \forall\, y \in X : \neg(\overline{R}_{yx} \wedge v_y) \\ &\Longleftrightarrow \forall\, y \in X : R_{yx} \vee \neg v_y \\ &\Longleftrightarrow \forall\, y \in X : v_y \to R_{yx}. \end{aligned}$$

The above relational descriptions of upper and lower bounds as vectors originate from [Riguet 48]. Guided by it, we now introduce two functions on relations, viz.

$$\begin{array}{ll} Ma : (X \leftrightarrow Y) \to (X \leftrightarrow Y) & Ma(S) := \overline{\overline{R}^{\mathsf{T}} S} \\ Mi\ : (X \leftrightarrow Y) \to (X \leftrightarrow Y) & Mi(S)\ := \overline{\overline{R}\, S}. \end{array}$$

Let $S : X \leftrightarrow Y$ be a relation. Using Boolean matrix terminology, we obtain for an element $y \in Y$ that the y-column of $Ma(S)$ is the vector of the upper bounds wrt. R of the y-column of S and the y-column of $Mi(S)$ is the vector of the lower bounds wrt. R of the y-column of S.

For an application of RELVIEW, we assume the Boolean matrix R to be contained in the system's working memory. Then the declaration of Ma and Mi in RELVIEW is possible using the command `DEF` twice, where the arguments are:

`Ma`	$-(-\mathtt{R}\hat{}*\%)$
`Mi`	$-(-\mathtt{R}*\%)$

(2) Aiming at the computation of cuts, next we consider the is-element-of-relation $\epsilon : X \leftrightarrow 2^X$ and compare it with the relation $Mi(Ma(\epsilon)) : X \leftrightarrow 2^X$. By

$$s : 2^X \leftrightarrow \mathbf{1} \qquad\qquad s := (syq(Mi(Ma(\epsilon)), \epsilon) \cap I)\, L,$$

where $I : 2^X \leftrightarrow 2^X$ and $L : 2^X \leftrightarrow \mathbf{1}$ are the identity and universal relation, respectively, we obtain the vector describing the subsets of 2^X which are cuts. For a proof, assume a set $m \in 2^X$. Using the correspondences between the identity relation and the meta-level equality symbol, we obtain for the m-component of the vector s

$$\begin{aligned} s_m &\iff ((syq(Mi(Ma(\epsilon)), \epsilon) \cap I)\, L)_m \\ &\iff \exists\, n \in 2^X : syq(Mi(Ma(\epsilon)), \epsilon)_{mn} \wedge I_{mn} \wedge L_n \\ &\iff \exists\, n \in 2^X : syq(Mi(Ma(\epsilon)), \epsilon)_{mn} \wedge (m = n) \\ &\iff syq(Mi(Ma(\epsilon)), \epsilon)_{mm} \\ &\iff \forall\, x \in X : Mi(Ma(\epsilon))_{xm} \leftrightarrow \epsilon_{xm}. \end{aligned}$$

This implies that $s_m \Leftrightarrow 1$ holds if and only if the subset of X described by the m-column of the is-element-of relation $\epsilon : X \leftrightarrow 2^X$ is a cut.

In the RELVIEW-system, the command `EPSI` (row 13) computes for any homogeneous relation $Q : X \leftrightarrow X$ (representing the set X) the is-element-of relation ϵ. Thus, using this button in combination with the button `SYQ` (symmetric quotient, row 6), the buttons for the basic operations (row 5), and the user-declared functions `Ma` and `Mi` of Step (1), a computation of the vector of cuts s in RELVIEW is obvious.

(3) Since the cuts are ordered by set inclusion, in the third step of our cut completion procedure we consider the relation $\sqsubseteq\, : 2^X \leftrightarrow 2^X$. If we use the correspondences between the two relations $\sqsubseteq$ and ϵ and the meta-level symbols $\subseteq$ and $\in$, then we have for arbitrary sets $m, n \in 2^X$ the equivalence

$$\begin{aligned} \sqsubseteq_{mn} &\iff m \subseteq n \\ &\iff \forall\, x \in X : x \in m \rightarrow x \in n \\ &\iff \forall\, x \in X : \epsilon_{xm} \rightarrow \epsilon_{xn} \\ &\iff (\epsilon \setminus \epsilon)_{mn}. \end{aligned}$$

This shows that set inclusion equals as a relation $\sqsubseteq$ on the object level the right residual $\epsilon \setminus \epsilon$. Hence, the two RELVIEW commands `EPSI` (row 13) and `R \ S` (right residual, row 6) suffice to compute it.

(4) In this step, firstly, we consider an injective mapping $\imath : \mathcal{C}_X \to 2^X$ such that $\mathcal{C}_X$ is a subset of 2^X given by $\imath$ in the sense of Subsection 2.3. Using $\imath$, it becomes obvious, how the restriction $\sqsubseteq_\mathcal{C} : \mathcal{C}_X \leftrightarrow \mathcal{C}_X$ of set inclusion to the subset of cuts can be computed:

$$\sqsubseteq_\mathcal{C} \; = \; \imath \sqsubseteq \imath^{\mathrm{T}}$$

If we change to common function notation for the mapping $\imath$, this equation follows from the fact that for all cuts $m, n \in \mathcal{C}_X$

$$\begin{aligned} \sqsubseteq_{\mathcal{C}mn} &\iff \exists\, m_1, n_1 \in 2^X : m_1 = \imath(m) \wedge n_1 = \imath(n) \wedge m_1 \sqsubseteq n_1 \\ &\iff \exists\, m_1 \in 2^X : m_1 = \imath(m) \wedge (\exists\, n_1 \in 2^X : m_1 \sqsubseteq n_1 \wedge n_1 = \imath(n)) \\ &\iff \exists\, m_1 \in 2^X : m_1 = \imath(m) \wedge (\sqsubseteq \imath^{\mathrm{T}})_{m_1 n} \\ &\iff (\imath \sqsubseteq \imath^{\mathrm{T}})_{mn}. \end{aligned}$$

At this point the following problem arises. In Version 2.0 of RELVIEW, there exists no command which performs the transition from a vector to an injective mapping such that the vector is a subset given by the mapping. Hence, at present a computation of $\imath : \mathcal{C}_X \to 2^X$ from the result $s : 2^X \leftrightarrow \mathbf{1}$ of Step (2) can only be done by drawing on the drawing-window. Although such a computation by drawing is straightforward, it is no more feasible with a bigger set X. Therefore, we work on an extension of RELVIEW. This version is almost completed and contains (among other things) an additional command for the transition from a vector to a corresponding injective mapping.

(5) As the last step of relational cut completion it remains to describe the injective order homomorphism e from X to $\mathcal{C}_X$ with relation algebraic means. In terms of the is-element-of-relation $\epsilon : X \leftrightarrow 2^X$ of Step (2) and the injective mapping $\imath : \mathcal{C}_X \to 2^X$ of Step (4), we obtain

$$e : X \to \mathcal{C}_X \qquad\qquad e = syq(R, \epsilon\, \imath^{\mathrm{T}}).$$

This representation is easily established: Assume an element $x \in X$ and a cut $m \in \mathcal{C}_X$. If we use again the common function notation for $\imath$ and also the meta-level symbols $\in$ and $=$, then we have for all $x \in X$ and $m \in \mathcal{C}_X$ the equivalence

$$\begin{aligned} syq(R, \epsilon\, \imath^{\mathrm{T}})_{xm} &\iff \forall\, y \in X : R_{yx} \leftrightarrow (\epsilon\, \imath^{\mathrm{T}})_{ym} \\ &\iff \forall\, y \in X : R_{yx} \leftrightarrow (\exists\, z \in 2^X : y \in z \wedge \imath(m) = z) \\ &\iff \forall\, y \in X : R_{yx} \leftrightarrow y \in \imath(m) \\ &\iff \forall\, y \in X : y \in (x) \leftrightarrow y \in \imath(m) \\ &\iff (x) = \imath(m). \end{aligned}$$

Hence, the relation $syq(R, \epsilon\, \imath^{\mathrm{T}})$ is a mapping and associates an element $x \in X$ to a cut $m \in \mathcal{C}_X$ if and only if $\imath(m) \in 2^X$ is the principal cut $(x) \in \mathcal{P}_X$ generated by the element x. The above description of e as the symmetric quotient of R and $\epsilon\, \imath^{\mathrm{T}}$ also shows that – based on ϵ and $\imath$ – a computation of the order homomorphism in RELVIEW is obvious.

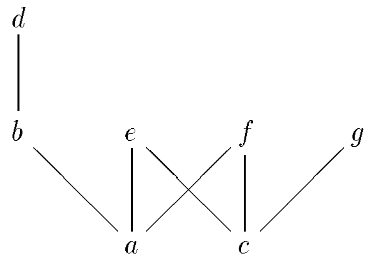
Before we illustrate the use of this procedure by means of an example, it should be pointed out that it not only can be used for computing the partial ordering $\sqsubseteq_\mathcal{C} : \mathcal{C}_X \leftrightarrow \mathcal{C}_X$ but also for computing a relation $C : X \leftrightarrow \mathcal{C}_X$ the columns of which describe exactly the cuts. Using the is-element-of relation ϵ of Step (2) and the injective mapping $\imath$ of Step (4) we have

$$
\begin{aligned}
(\epsilon\,\imath^{\mathrm{T}})_{xm} &\iff \exists\, z \in 2^X : \epsilon_{xz} \wedge (\imath(m) = z)\\
&\iff \exists\, z \in 2^X : x \in z \wedge (\imath(m) = z)\\
&\iff x \in \imath(m)
\end{aligned}
$$

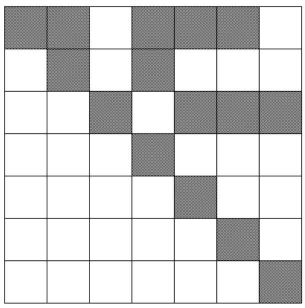
for all $x \in X$ and $m \in \mathcal{C}_X$. Hence, the relation describing the cuts is given by $C = \epsilon\,\imath^{\mathrm{T}}$.

4.3 Example

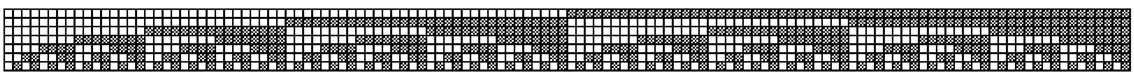
After having presented a procedure for constructing cut completions of posets with relation algebraic means and how to perform it within the RELVIEW-system in the previous subsection, we now deal with a concrete example. We consider a set $X := \{a, b, c, d, e, f, g\}$ with 7 elements and a partial ordering $R : X \leftrightarrow X$. Graphically, the Hasse-diagram of the irreflexive part of R looks as follows:



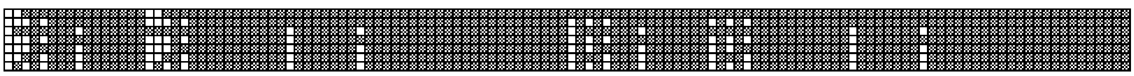
Representing R as Boolean matrix in RELVIEW, we obtain:



The following three pictures show the relations produced by the RELVIEW-system during Step (2) of our procedure. Firstly, we have the is-element-of relation $\epsilon : X \leftrightarrow 2^X$. This relation has 7 rows and 128 columns. Each column stands for a subset of X. For example, the first column corresponds to the empty set and row 8 corresponds to the set $\{e, f, g\}$.

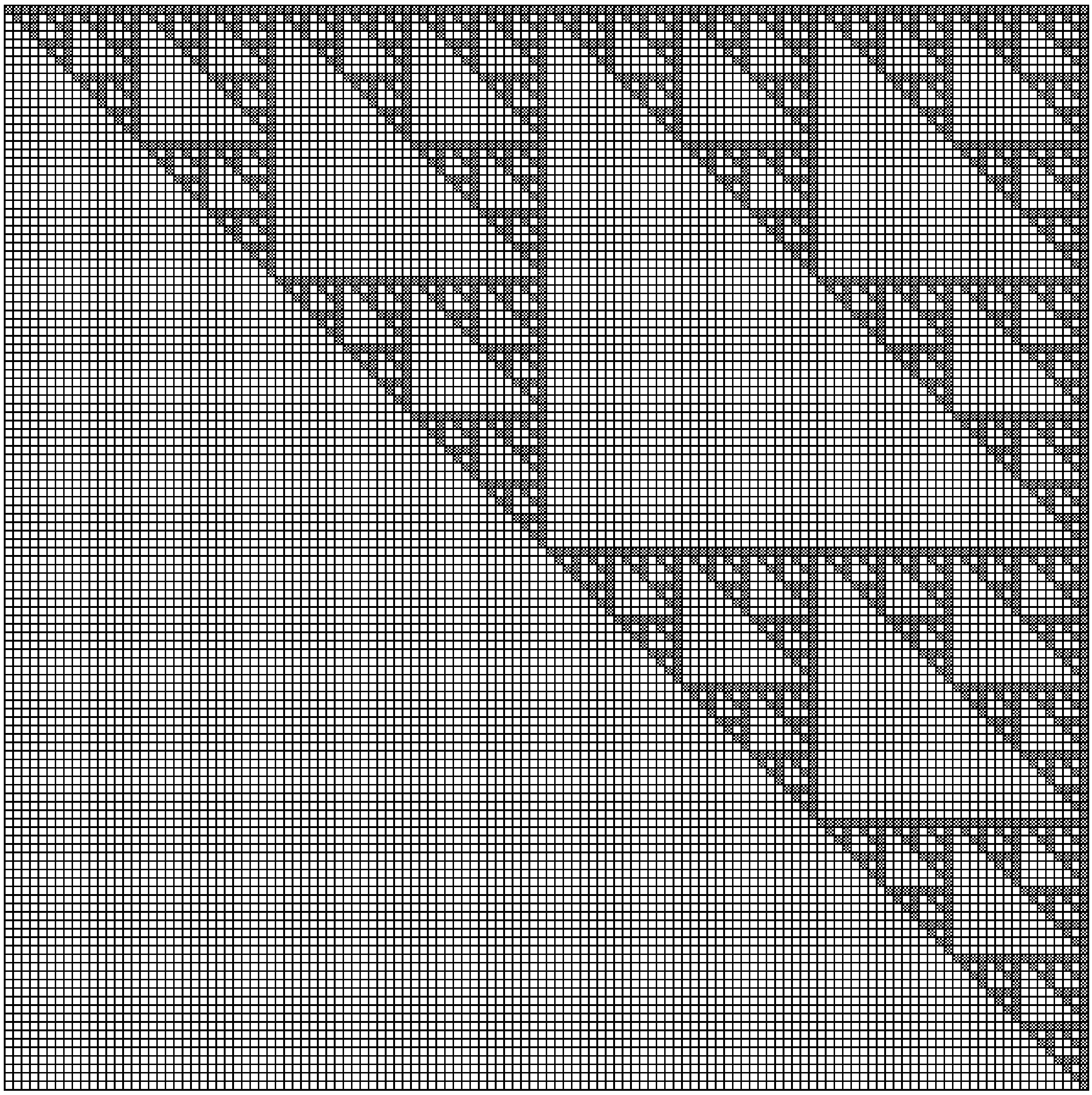


Next, an application of the user-declared RELVIEW-function `Ma` to this relation followed by an application of the user-declared function `Mi` to the result of `Ma` yields:

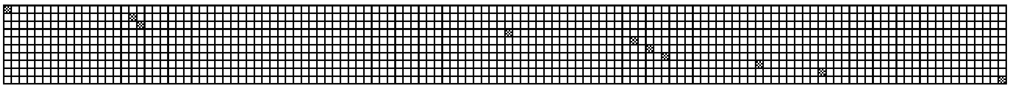


As described in Step (2), we compare this relation column-wise with the is-element-of relation $\epsilon : X \leftrightarrow 2^X$ via an application of the symmetric quotient, followed by an intersection with the identity relation $I : 2^X \leftrightarrow 2^X$ and a multiplication with the universal relation $L : 2^X \leftrightarrow \mathbf{1}$. The result is, finally, the vector of cuts $s : 2^X \leftrightarrow \mathbf{1}$. Its transpose $s^{\mathrm{T}} : \mathbf{1} \leftrightarrow 2^X$ is given in the following picture. From this Boolean row vector of length 128 we see that exactly 10 subsets of the set X are cuts, for instance the empty set (position 1) and the sets $\{c\}$ (position 17) and $\{c, g\}$ (position 18).

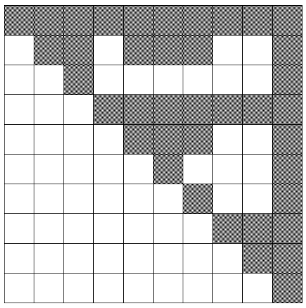
The next picture shows (cf. Step (3)) set inclusion $\sqsubseteq : 2^X \leftrightarrow 2^X$ as obtained by RELVIEW using the built-in command `EPSI`. For instance, the field in row 17 and column 18 of this Boolean matrix is black since $\{c\}$ is a subset of $\{c, g\}$, and the field in row 17 and column 8 is white since $\{c\}$ is not a subset of $\{e, f, g\}$.



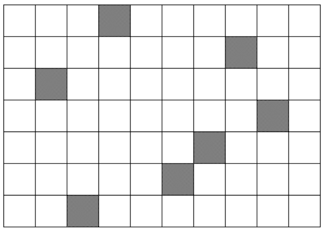
Now, we show the two relations of Step (4) as produced by the RELVIEW-system. Obviously, the construction of an injective mapping $\imath : \mathcal{C}_X \to 2^X$ from $s^{\mathrm{T}} : \mathbf{1} \leftrightarrow 2^X$ is straightforward. But, as already mentioned, in Version 2.0 of RELVIEW there exists no corresponding command. Hence, the following mapping must be drawn on the drawing-window of RELVIEW.



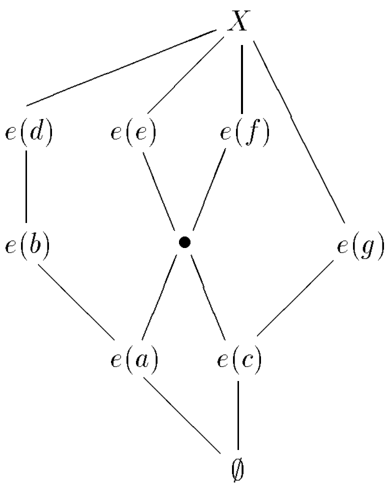
From this relation we obtain in combination with set inclusion $\sqsubseteq\, : \; 2^X \leftrightarrow 2^X$ as the partial ordering $\sqsubseteq_{\mathcal{C}} : \mathcal{C}_X \leftrightarrow \mathcal{C}_X$ of the cut completion $(\mathcal{C}_X, \sqsubseteq_{\mathcal{C}})$:



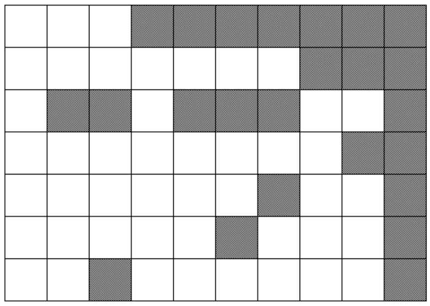
The relation to be computed in Step (5) of our procedure is the injective mapping $e : X \to \mathcal{C}_X$. Using $\imath : \mathcal{C}_X \to 2^X$, $\sqsubseteq\, : \; 2^X \leftrightarrow 2^X$, and the original partial ordering $R : X \leftrightarrow X$, we get:



In the following figure, the result of the cut completion is visualized. This picture graphically shows the Hasse-diagram of the irreflexive part of the partial ordering $\sqsubseteq_{\mathcal{C}} : \mathcal{C}_X \leftrightarrow \mathcal{C}_X$, where the bullet stands for the set $\{a, c\}$. Furthermore, the images $e(x)$, $x \in X$, of the mapping e are indicated.



As already mentioned, the procedure can also be used for computing a relation $C : X \leftrightarrow \mathcal{C}_X$ the columns of which describe exactly the cuts. In our example we obtain



i.e., $\mathcal{C}_X$ consists of the sets $\emptyset$, $\{c\}$, $\{c, g\}$, $\{a\}$, $\{a, c\}$, $\{a, c, f\}$, $\{a, c, e\}$, $\{a, b\}$, $\{a, b, d\}$, and X.

4.4 A Relational Description of Further Extremal Elements

In this subsection we describe further extremal elements of a poset by relation algebraic means. Also a small experiment with the RELVIEW-system is sketched, viz. testing whether a poset is a complete lattice.

Let (X, R) be a poset. If one has functions $Ma(S) := \overline{\overline{R}^{\mathrm{T}} S}$ and $Mi(S) := \overline{\overline{R} S}$ for computing upper and lower bounds wrt. R, it is very easy to define functions computing other extremal elements. We note (cf. [Schmidt Ströhlein 89], Section 3.3):

$$
\begin{array}{lll}
\textit{Greatest element}: & Ge(S) & := S \cap Ma(S) \\
\textit{Least element}: & Le(S) & := S \cap Mi(S) \\
\textit{Least upper bound}: & Lub(S) & := Le(Ma(S)) \\
\textit{Greatest lower bound}: & Glb(S) & := Ge(Mi(S))
\end{array}
$$

For instance, in Boolean matrix terminology for a relation $S : X \leftrightarrow Y$ every column of the relation $Ge(S) : X \leftrightarrow Y$ is a point or the empty vector. In the first case the point represents the greatest element wrt. R of the corresponding column of S; in the latter case such a greatest element does not exist.

A poset (X, R) is a complete lattice even if every subset $Y \subseteq X$ has a least upper bound. Since every subset of X is described by a column of the is-element-of relation $\epsilon : X \leftrightarrow 2^X$, the functional relation $Lub(\epsilon)^{\mathrm{T}} : 2^X \leftrightarrow X$ associates a set $m \in 2^X$ to an element $x \in X$ if and only if x is the least upper bound of m. As a simple consequence, (X, R) is a complete lattice if and only if $Lub(\epsilon)^{\mathrm{T}}$ is total, i.e., a mapping.

Based on this relational characterization, now testing the poset (X, R) to be a complete lattice in RELVIEW is obvious: Firstly, one declares a function `Ma` as in Step (1) of Subsection 4.2 and a further function

```
Le        % & - (-R * %)
```

for computing the relation of the least elements wrt. R. Then, one calls `Ma` with the is-element-of relation ϵ as argument and applies subsequently `Le` to the resulting relation. The transpose of a relation is total if and only if the relation is surjective. Hence, the last step of the procedure consists in an execution of the command `FUNC` (row 8) with the result of `Le` as argument. Among other things, this command tests also whether a relation is surjective.

5 Some Future Work on RELVIEW

Besides the experiments described in the last section, a lot of other studies have been performed with the RELVIEW-system including graph-theoretic algorithms, DAG-languages, domain constructions, relational specifications, and relational semantics. It turns out that the system in its present version is a good tool for the interactive manipulation of relations, but for more advanced scientific tasks some additional features will be very helpful and even necessary. Cf. the construction of the injective mapping $\imath$ in Step (4) of the procedure of Subsection 4.2 which is by hand no more feasible if the cardinality of X is greater than 10, for example. Therefore, our work is not finished. We see RELVIEW 2.0 only as an intermediate result.

As already mentioned, an extension of RELVIEW is under construction. It will take into account the experiences with the present Version 2.0.

The three main topics of the future work on RELVIEW are the following:

a) *Visualization of relations and vectors*: The Boolean matrix interpretation of relations used within the RELVIEW-system is well suited for a graphical representation and manipulation of relations. Unrestricted, however, this statement holds only if the system serves for doing tests concerning abstract problems on relations (like $I \subseteq syq(R, R)$) in order to go insight and/or to disprove them. If one uses the system to solve concrete problems on graphs or related structures by relational methods (as we did in Section 4 of the paper), then a Boolean matrix is not a good visualization for all relations. Instead, it seems better to depict some (homogeneous) relations $R : X \leftrightarrow X$ as usually graphically: The elements $x \in X$ are plotted as points on the drawing-window and two points x and y are joined by an oriented line segment if and only if $R_{xy} \Leftrightarrow 1$. This drawing displays certain typical features of a relation better than a Boolean matrix representation. (Nevertheless, for other features a matrix representation is more advantageous.)

   Therefore, the next version of RELVIEW will contain commands realizing a transition between Boolean matrices and graphs. Especially, it will be possible to edit a relation as a graph. For a visualization of results, furthermore, the user is given the tools to display a relation $R : X \leftrightarrow X$ on the screen as a directed graph and to emphasize a subset of the nodes described by a vector $v : X \leftrightarrow \mathbf{1}$.

b) *Many-place user-declared functions*: In the present version of RELVIEW, the user can declare only unary (one-place) functions. However, besides the built-in operations there are many operations on relations which depend on more than one arguments. Two typical examples used in relational semantics are

$$\begin{array}{ll} \textit{Demonic composition}: & R\,;S \;:= R\,S \cap \overline{R\,\overline{S\,L}} \\ \textit{Demonic join}: & R\,[]\,S := (R \cup S) \cap R\,L \cap S\,L. \end{array}$$

   These binary operations reflect the essence of demonic nondeterminism, viz. that an evaluation is guaranteed to yield a defined result if and only if any of the choices yields a defined result. The relational demonic join operation corresponds exactly to Dijkstra's demonic nondeterministic branching operator.

   Due to the fundamental importance of many-place operations, it is planned to incorporate also many-place user-declared functions in RELVIEW.

c) *Extension of relational terms*: In Version 2.0 of RELVIEW, relational terms are built from the relations of the working memory using the basic operations (row 5), the two residuals (row 6), and the symmetric quotient (row 6). Calls of user-declared functions in terms are not allowed. In the case of a more complex problem, this restriction leads to lengthy assembler-like computations. As a further disadvantage, it is not possible to use auxiliary functions to introduce structure and to master problem complexity; rather one obtains ugly function bodies. For instance, a declaration of a function for computing least upper bounds reads:

$$\texttt{Lub} \qquad -(-\texttt{R}\hat{} * \%)\,\&-(-\texttt{R} * -(-\texttt{R}\hat{} * \%))$$

Therefore, calls of user-declared functions in relational terms will be allowed in a future version of RELVIEW. This will lead to a simple programming language the programs of which are sequences of function declarations, e.g.:

$$\begin{array}{ll} \texttt{Ma} & -(-\texttt{R}\hat{} * \%) \\ \texttt{Mi} & -(-\texttt{R} * \%) \\ \texttt{Le} & \%\,\&\,\texttt{Mi}(\%) \\ \texttt{Lub} & \texttt{Le}(\texttt{Ma}(\%)) \end{array}$$

Besides these main points, we work also on some minor extensions of RELVIEW. In Subsection 4.2 we have already mentioned a future command for the transition from a vector to a corresponding injective mapping. Another example is a command which will transform a relation R fulfilling $R\,R^{\mathrm{T}}R \subseteq R$ – a so-called difunctional relation – by rearranging rows and columns into block form.

Finally, we are also concerned with interfaces to other systems. The ability for producing scientific papers on relations which mix text and drawings of Boolean matrices and graphs, respectively can be obtained by interfacing the RELVIEW-system with some typesetting system available for Sun workstations. Furthermore, an interface to the relational formula manipulation system RALF (also developed at the Universität der Bundeswehr München, see [Brethauer 91]) is planned.

6 References

[Abold et al. 89] Abold-Thalmann H., Berghammer R., Schmidt G.: Manipulation of concrete relations: The RELVIEW-system. Report Nr. 8905, Fakultät für Informatik, Universität der Bundeswehr München (1989)

[Berghammer Schmidt 83] Berghammer R., Schmidt G.: Discrete ordering relations. Discrete Mathematics 43, 1-7 (1983)

[Berghammer et al. 86] Berghammer R., Schmidt G., Zierer H.: Symmetric quotients. Report TUM-I8620, Institut für Informatik , Technische Universität München (1986)

[Brethauer 91] Brethauer R.: Ein Formelmanipulationssystem zur computergestützten Beweisführung in der Relationenalgebra. Diplomarbeit, Fakultät für Informatik, Universität der Bundeswehr München (1991)

[Erné 82] Erné M.: Einführung in die Ordnungstheorie. B.I.-Wissenschaftsverlag: Mannheim-Wien-Zürich (1982)

[Hermes 67] Hermes H.: Einführung in die Verbandstheorie. 2. Auflage, Springer: Berlin-Heidelberg-New York (1989)

[Landgraf Dombetzki 91] Landgraf J., Dombetzki N.: Erweiterung des RELVIEW-Systems. Trimesterarbeit, Fakultät für Informatik, Universität der Bundeswehr München (1991)

[Riguet 48] Riguet J.: Relations binaires, fermetures, correspondances de Galois. Bull. Soc. Math. France 76, 114-155 (1948)

[Schmidt Ströhlein 89] Schmidt G., Ströhlein T.: Relationen und Graphen. Springer: Berlin-Heidelberg-New York (1989) English version: Relations and graphs. Springer: Berlin-Heidelberg-New York (forthcoming)

[Spivey 89] Spivey J.M.: The Z notation: A reference manual. Prentice Hall: Englewood Cliffs (1989)