

Algebraic Visualization of Relations Using RELVIEW

Rudolf Berghammer

Institut für Informatik
Christian-Albrechts-Universität zu Kiel

joint work with Gunther Schmidt (Universität der Bundeswehr München)

Introduction

Relations and graphs are widely used as modeling tools.

For graphs there exist highly elaborated drawing algorithms that help getting an impression on how the graph is structured.

We concentrate here in an analogous way on [visualizing relations represented as Boolean matrices](#) in the relation-algebraic tool RELVIEW.

This means [rearranging the matrix appropriately, by permuting rows and columns](#) simultaneously or independently as required.

In this way, many complex situations may successfully be handled in various application fields.

We show how relation algebra and RELVIEW can be combined to solve such tasks and restrict us to some orders used by decision makers.

Relation Algebra

Relations:

- R is a relation with domain X and range Y :

$$R : X \leftrightarrow Y$$

$X \leftrightarrow Y$ is the **type** of R .

- Instead of $(x, y) \in R$ we use Boolean matrix notation:

$$R_{x,y}$$

Signature of relation algebra:

- Constants: $\mathbf{O}, \mathbf{L}, \mathbf{I}$.
- Operations: $R \cup S, R \cap S, RS, \overline{R}, R^T$.
- Tests: $R \subseteq S, R = S$.

The Relation-Algebraic Tool RELVIEW

The screenshot displays the RELVIEW software interface, which is used for relation algebra. The main window, titled "Kiel University Relview", shows a graph visualization of a relation E with 14 nodes. The graph consists of 14 nodes (labeled 1 to 14) and directed edges. The nodes are arranged in a roughly diamond shape, with node 1 at the bottom center, node 12 at the top center, and nodes 2 and 14 at the far right and left respectively. The edges are as follows:

- 1 → 2, 1 → 3, 1 → 4, 1 → 5
- 2 → 6
- 3 → 7, 3 → 8
- 4 → 9, 4 → 8
- 5 → 8, 5 → 11
- 6 → 10, 6 → 11
- 7 → 10, 7 → 11
- 8 → 11, 8 → 12
- 9 → 13, 9 → 10
- 10 → 11, 10 → 14
- 11 → 12, 11 → 13, 11 → 14
- 12 → 11
- 13 → 10
- 14 → 10

The interface includes several windows and panels:

- Evaluate Term:** A panel for evaluating terms, with fields for "Result", "Term", and "History", and buttons for "Clear History", "Eval", and "Close".
- Directory:** A panel showing a list of relations and domains. The "Relations" tab is active, showing:

Relation	Domain
E	=graph 14 X 14
T	=graph 14 X 14
\$	=graph 32 X 32

Buttons for "New", "Delete", and "Clear" are present.
- Programs/Functions:** A panel showing a list of programs and functions, including "Prim(E)" and "Kruskal(E)". A "Clear" button is at the bottom.
- View:** A panel with "Normal" and "Hidden" view options and a "Close" button.
- Terminal:** A terminal window at the bottom left showing the following commands and output:

```
floyd >>> !g
g SkriptVerbRel.ps
request id is 75
floyd >>> █
```
- Operations and user-defined functions:** A panel with buttons for "DEFI", "EVAL", "ITER", "TESTS", and "OPS".
- Files/Options/Info/Quit:** A panel with buttons for "FILES", "OPTIONS", "INFO", and "QUIT".
- Editors/Directories:** A panel with buttons for "RELATION", "GRAPH", "XRV/PROG", and "LABEL".
- NAME: E DIM: 14 X 14 (origin 1 : 1):** A window showing a 14x14 grid representing the relation E .

Example of a **relational function** in RELVIEW:

$$\text{Hasse}(C) = C \ \& \ \neg(C * C).$$

A call $\text{Hasse}(C)$ computes the **Hasse-diagram** $H_C = C \cap \overline{CC}$ of a **strict-order relation** C (i.e., $C \subseteq \bar{\mathbf{I}}$ and $CC \subseteq C$).

Example of a **relational program** in RELVIEW:

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

A call $\text{Szpilrajn}(E)$ computes a **linear extension** of a **partial order relation** E (i.e., $\mathbf{I} \subseteq E$, $E \cap E^T \subseteq \mathbf{I}$ and $EE \subseteq E$) using Szpilrajn's method.

A Motivating Example

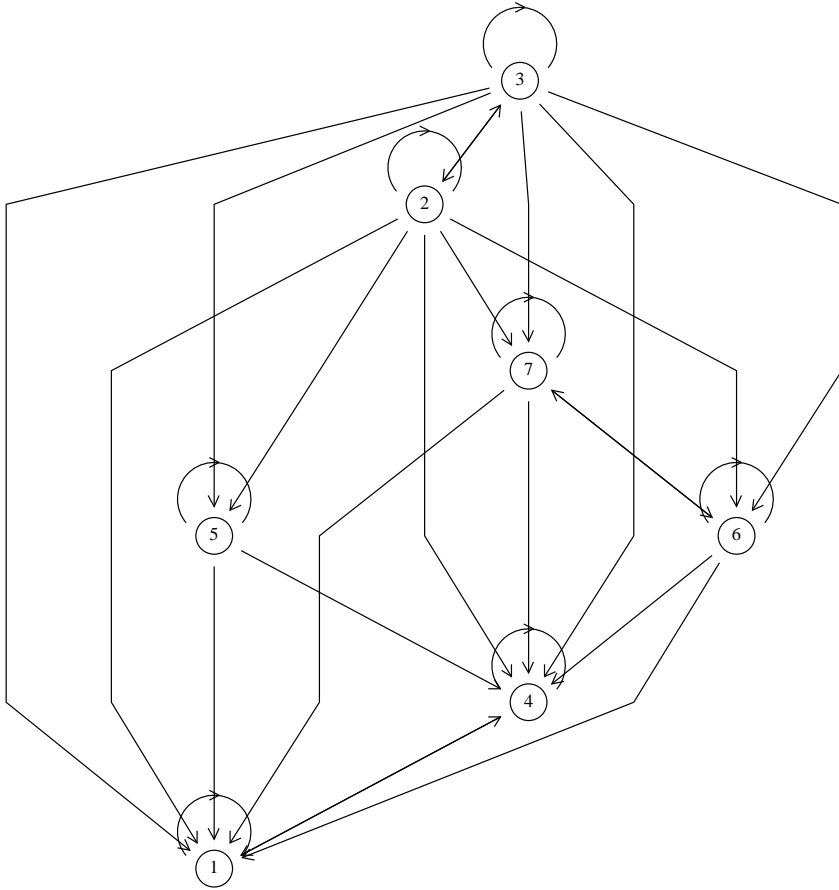
As a rather trivial example we consider a relation on elements $1, 2, \dots, 7$, represented with `RELVIEW` as the following Boolean matrix.

	1	2	3	4	5	6	7
1	1	0	0	1	0	0	0
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	1	0	0	1	0	0	0
5	1	0	0	1	1	0	0
6	1	0	0	1	0	1	1
7	1	0	0	1	0	1	1

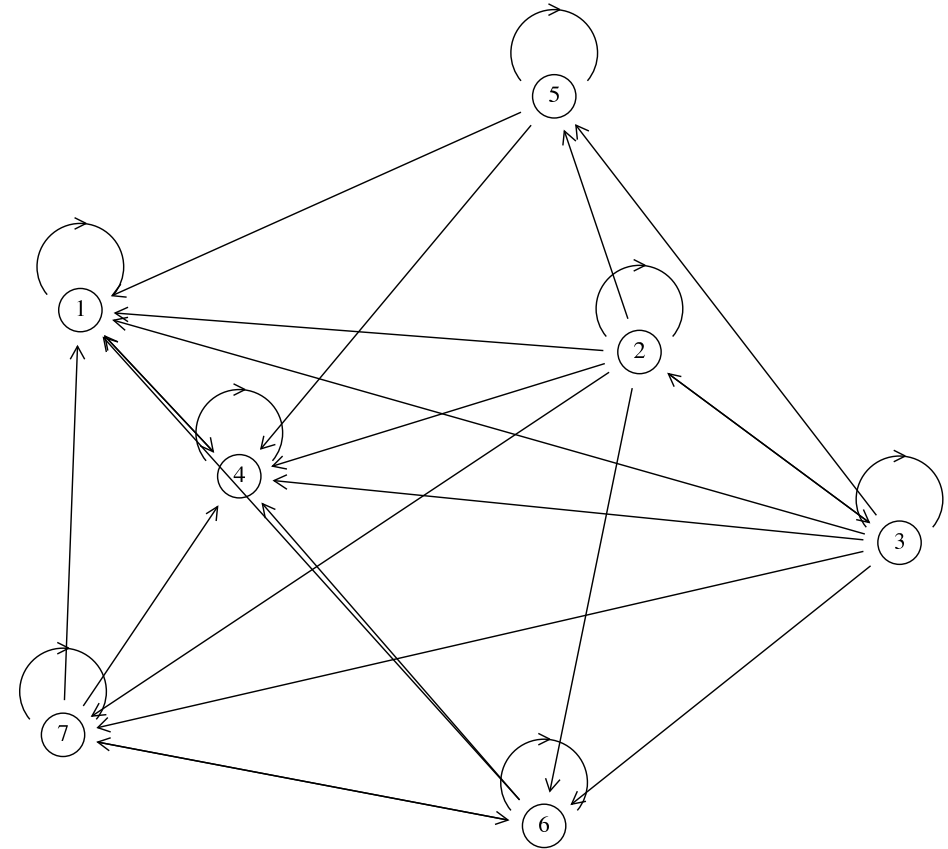
A black square stands for the matrix entry 1 (or *true*) and a white square stands for the entry 0 (or *false*).

It is easy to check that the relation Q represented by the matrix is a **pre-order relation** (i.e., $\mathbf{I} \subseteq Q$ and $QQ \subseteq Q$).

With a graph drawing algorithm one would obtain something as shown in the pictures (again produced by RELVIEW).



hierarchical polyline drawing



spring-embedder algorithm

More intuitive than these drawings is the left RELVIEW-matrix. It is an **upper right triangle of rectangles**, where the four rectangle-forming parts correspond to the sets $\{2, 3\}$, $\{5\}$, $\{6, 7\}$ and $\{1, 4\}$ of indices of the original matrix Q .

	2	3	5	6	7	1	4
2	■	■	■	■	■	■	■
3	■	■	■	■	■	■	■
5	□	□	■	□	□	■	■
6	□	□	□	■	■	■	■
7	□	□	□	■	■	■	■
1	□	□	□	□	□	■	■
4	□	□	□	□	□	■	■

	1	2	3	4	5	6	7
1	□	□	□	□	□	■	□
2	■	□	□	□	□	□	□
3	□	■	□	□	□	□	□
4	□	□	□	□	□	□	■
5	□	□	■	□	□	□	□
6	□	□	□	■	□	□	□
7	□	□	□	□	■	□	□

The key of the rearrangement is the **permutation relation** P (i.e., $PP^T = P^T P = \mathbf{I}$) shown as a RELVIEW-matrix on the right.

In general, computing a permutation relation P is the main task, since then

$$P^T R P$$

is the rearranged version of a relation R .

Rearranging Linear Strict-Orders

General assumption:

- Sets X between which a relation is defined we work with are equipped with a **linear strict-order relation**

$$\Omega_X : X \leftrightarrow X$$

(i.e., $\Omega_X \subseteq \bar{\mathbf{I}}$, $\Omega_X \Omega_X \subseteq \Omega_X$ and $\bar{\mathbf{I}} \subseteq \Omega_X \cup \Omega_X^T$), the **base strict-order**.

In RELVIEW:

- The base strict-order is implicitly given by the relation

$$succ : X \leftrightarrow X$$

which has, as Boolean matrix, 1-entries in the **upper secondary diagonal** and 0-entries otherwise.

- With $succ^+$ as the **transitive closure** of $succ$ we have $\Omega_X = succ^+$ and Ω_X is depicted as **full upper right triangle matrix**.

Input: Linear strict-order relation $C : X \leftrightarrow X$.

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^T C P$ is depicted as a **full upper right triangle** matrix.

The procedure:

- Compute the Hasse-diagrams of C and Ω_X :

$$H_C = C \cap \overline{CC} \qquad H_{\Omega_X} = \Omega_X \cap \overline{\Omega_X \Omega_X}$$

- Relate precisely the least element of (X, C) with that of (X, Ω_X) :

$$P_0 = \text{least}(C, \mathbf{L}) \text{least}(\Omega_X, \mathbf{L})^T$$

- Successively apply the relational function

$$\tau(R) = R \cup H_C^T R H_{\Omega_X}$$

to P_0 , leading to the finite chain $P_0 \subset \tau(P_0) \subset \dots \subset \tau^{|X|-1}(P_0)$.

- Define $P = \tau^{|X|-1}(P_0)$.

RELVIEW-implementation of the procedure:

```
PermLSO(C)
  DECL L, HC, HO, P, Q
  BEG  L = Ln1(C);
       HC = Hasse(C);
       HO = succ(L);
       P = least(C,L)*least(trans(HO),L)^;
       Q = P | HC^*P*HO;
       WHILE -eq(P,Q) DO
         P = Q;
         Q = P | HC^*P*HO OD
       RETURN P
  END.
```

Transformation of C into a full upper right triangle matrix via

$$P^T C P,$$

where the permutation relation P is the result of $\text{PermLSO}(C)$.

Application I: Pre-Orders

Input: Pre-order relation $Q : X \leftrightarrow X$.

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^T Q P$ is depicted as an upper right triangle of rectangles.

The procedure:

- Remove from Q the mutually comparable pairs to obtain a strict-order relation C :

$$C = Q \cap \overline{Q} \cap Q^T$$

- Compute a linear extension E of the reflexive closure $C \cup \mathbf{I}$ of C

$$E = \text{Szpilrajn}(C \cup \mathbf{I})$$

- Result a permutation relation that transforms the linear strict-order relation $E \cap \bar{\mathbf{I}}$ into a full upper right triangle:

$$P = \text{PermLSO}(E \cap \bar{\mathbf{I}})$$

Application II: Weak-Orders

Input: Weak-order relation $W : X \leftrightarrow X$ (i.e., $W^T \subseteq \overline{W}$ and $\overline{W} \overline{W} \subseteq \overline{W}$; the latter inclusion is called **negative transitivity**).

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^T W P$ is depicted as an **upper right block triangle** matrix.

The procedure:

- Compute the reflexive closure of W , which is a partial order relation:

$$E = W \cup \mathbf{I}$$

- Determine a linear extension E' of E :

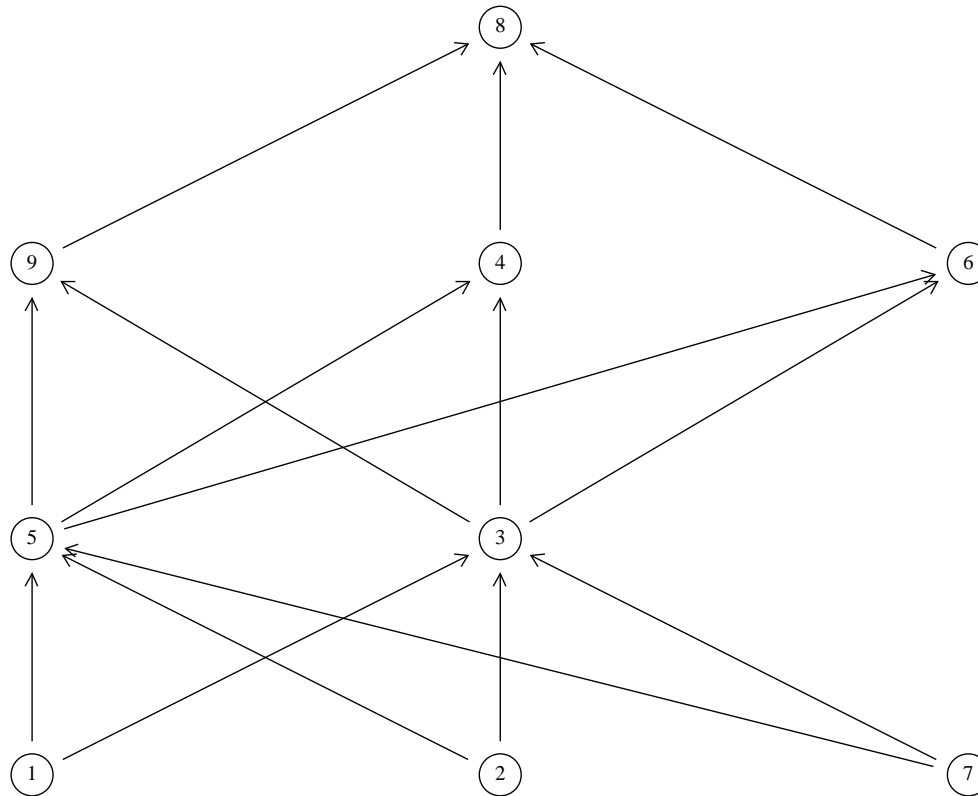
$$E' = \text{Szpilrajn}(E)$$

- Result a permutation relation that rearranges the linear strict-order relation $E' \cap \overline{\mathbf{I}}$ into a full upper right triangle:

$$P = \text{PermLSO}(E' \cap \overline{\mathbf{I}})$$

Informally, weak-orders are those strict-orders the Hasse-diagrams of which are composed by a series of complete bipartite strict-orders, one above another.

Example: Hasse-diagram of a weak-order relation on elements $1, 2, \dots, 9$ with 4 layers, drawn by RELVIEW:



From the block form boundary of the rearranged matrix the complete bipartite strict-orders and their arrangement immediately becomes apparent.

RELVIEW-matrices to the above example:

	1	2	3	4	5	6	7	8	9
1			■	■	■	■		■	■
2			■	■	■	■		■	■
3				■		■		■	■
4								■	
5			■	■	■	■		■	■
6								■	
7			■	■	■	■		■	■
8									
9								■	

original matrix

	1	2	3	4	5	6	7	8	9
1	■								
2		■							
3				■					
4						■			
5					■				
6							■		
7			■						
8									■
9								■	

permutation relation

	1	2	7	3	5	4	6	9	8
1				■	■	■	■	■	■
2				■	■	■	■	■	■
7				■	■	■	■	■	■
3						■	■	■	■
5						■	■	■	■
4									■
6									■
9									■
8									

rearranged matrix

bottom layer: 1, 2, 7
 second layer: 3, 5
 third layer: 4, 6, 9
 top layer: 8

Rearranging Semi-Orders via Weak-Orders

Input: Semi-order relation $S : X \leftrightarrow X$ (i.e., $S \subseteq \bar{\mathbf{I}}$, $\bar{S} \bar{S} \subseteq \overline{SS}$ and $S \bar{S}^T S \subseteq S$, where the latter two inclusions are called **semi-transitivity** and **Ferrers property**).

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^T S P$ is depicted as an **upper right block triangle** matrix with a **threshold**.

The procedure:

- Expand S to a weak-order relation W by adding whatever is missing for negative transitivity:

$$W = \bar{S}^T S \cup S \bar{S}^T$$

- Result a permutation relation that transforms W into an upper right block triangle form:

$$P = \text{PermWO}(W)$$

Here the relational program PermWO is assumed to be the result of Application II.

Interval-Orders

Input: Interval-order relation $J : X \leftrightarrow X$ (i.e., $J \subseteq \bar{\mathbf{I}}$ and $J\bar{J}^T J \subseteq J$).

Interval-order relations can be transformed into upper right staircase block form.

In contrast to the examples presented so far, such a rearrangement of interval-orders requires rows and columns to be permuted independently

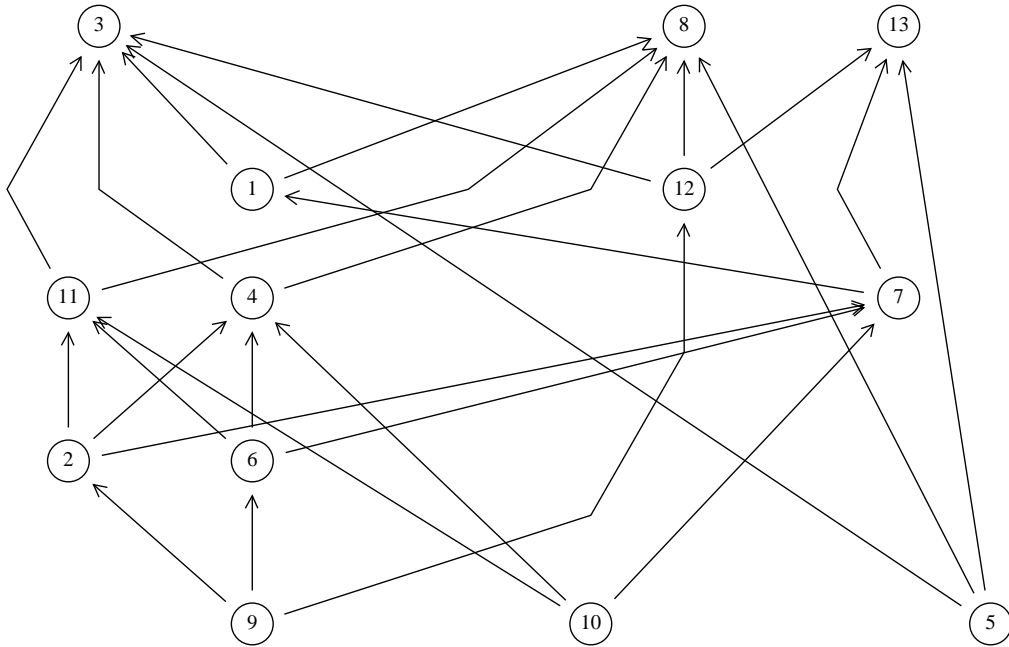
Result: We have to compute

$P_r : X \leftrightarrow X$ permutation relation for **rows**

$P_c : X \leftrightarrow X$ permutation relation for **columns**

such that $P_r^T J P_c$ is in **upper right staircase block** form.

An example:

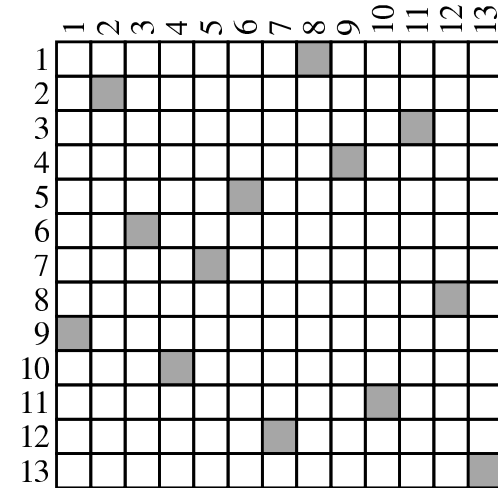
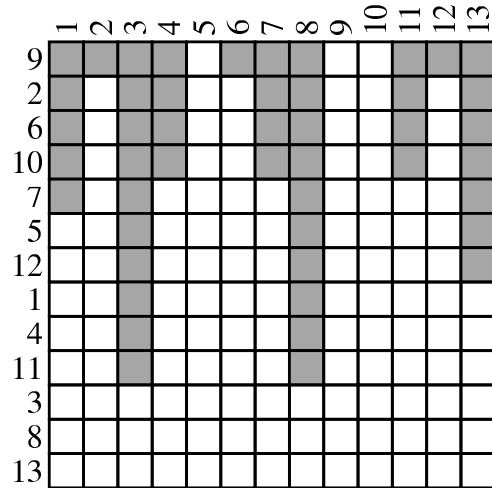


Hasse-diagram of J

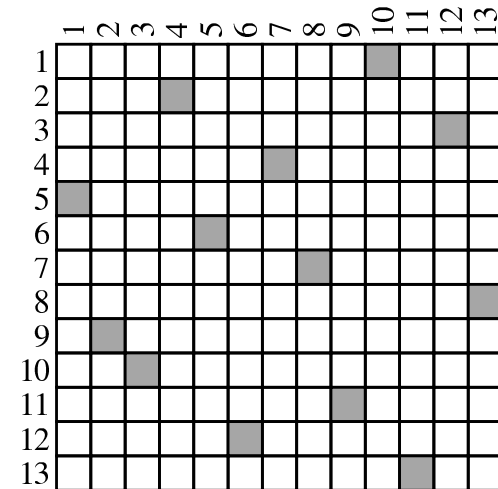
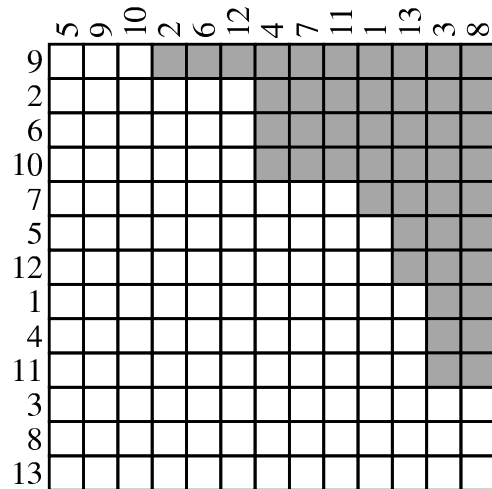
	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													

Boolean matrix representation of J

Sorting the rows of J in decreasing inclusion from top to bottom via $P_r^T J$, and the corresponding row permutation relation.



Sorting then the columns of $P_r^T J$ in increasing inclusion from left to right via $P_r^T J P_c$, and the corresponding column permutation relation.



Relation-algebraic specification of P_r :

$$P_r = \text{PermPreO}(\overline{J J^\top})$$

since the pre-order relation $\overline{J J^\top}$ relates $x, y \in X$ if the y -row of J is contained in the x -row of J :

$$\begin{aligned} (\overline{J J^\top})_{x,y} &\iff \neg \exists z : \overline{J}_{x,z} \wedge J^\top_{z,y} \\ &\iff \forall z : J^\top_{z,y} \rightarrow J_{x,z} \\ &\iff \forall z : J_{y,z} \rightarrow J_{x,z} \\ &\iff y\text{-row contained in } x\text{-row} \end{aligned}$$

Similar: Calculation of a relation-algebraic specification of P_c :

$$P_c = \text{PermPreO}(\overline{J^\top J})$$

Here the relational program PermPreO is assumed to be the result of Application I.

Some Further Applications

- Interval representations of interval-orders (see paper).

	a	b	c	d	e	f
1				■	■	
2		■				
3						■
4			■	■	■	
5	■	■	■	■		
6		■				
7			■			
8						■
9	■					
10	■	■				
11			■	■	■	
12		■	■	■		
13					■	■

intervals of a
linearly ordered
set with 6 elements

$$a < b < c < d < e < f$$

- Univalent and injective relations (i.e., $R^T R \subseteq \mathbf{I}$ and $RR^T \subseteq \mathbf{I}$).
- Partial equivalence relations (i.e., $R = R^T$ and $RR = R$).
- Maximum pair of independent sets rearrangements based on maximum matchings.
- Rearrangement of R according to its difunctional closure $R(R^T R)^+$

Future Aims

- Use of the efficiency and visualization power of RELVIEW to scan any given – even real-valued – matrix of moderate size for possibly hidden interesting properties.
In the real-valued case, one would use moving so-called cuts at different levels to arrive at Boolean matrices similar to the cuts used in the theory of fuzzy sets.
- Because of their close relationship to interval order relations we are also interested in the relation-algebraic treatment of interval graphs, e.g.:
 - Computation of interval representations
 - Matrix rearrangement based on perfect elimination orderings.
- Further development of the RELVIEW-tool in view of these new applications.
 - Additional basic operations.
 - Additional visualization features.