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:

$$\boxed{R : X \leftrightarrow Y}$$

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

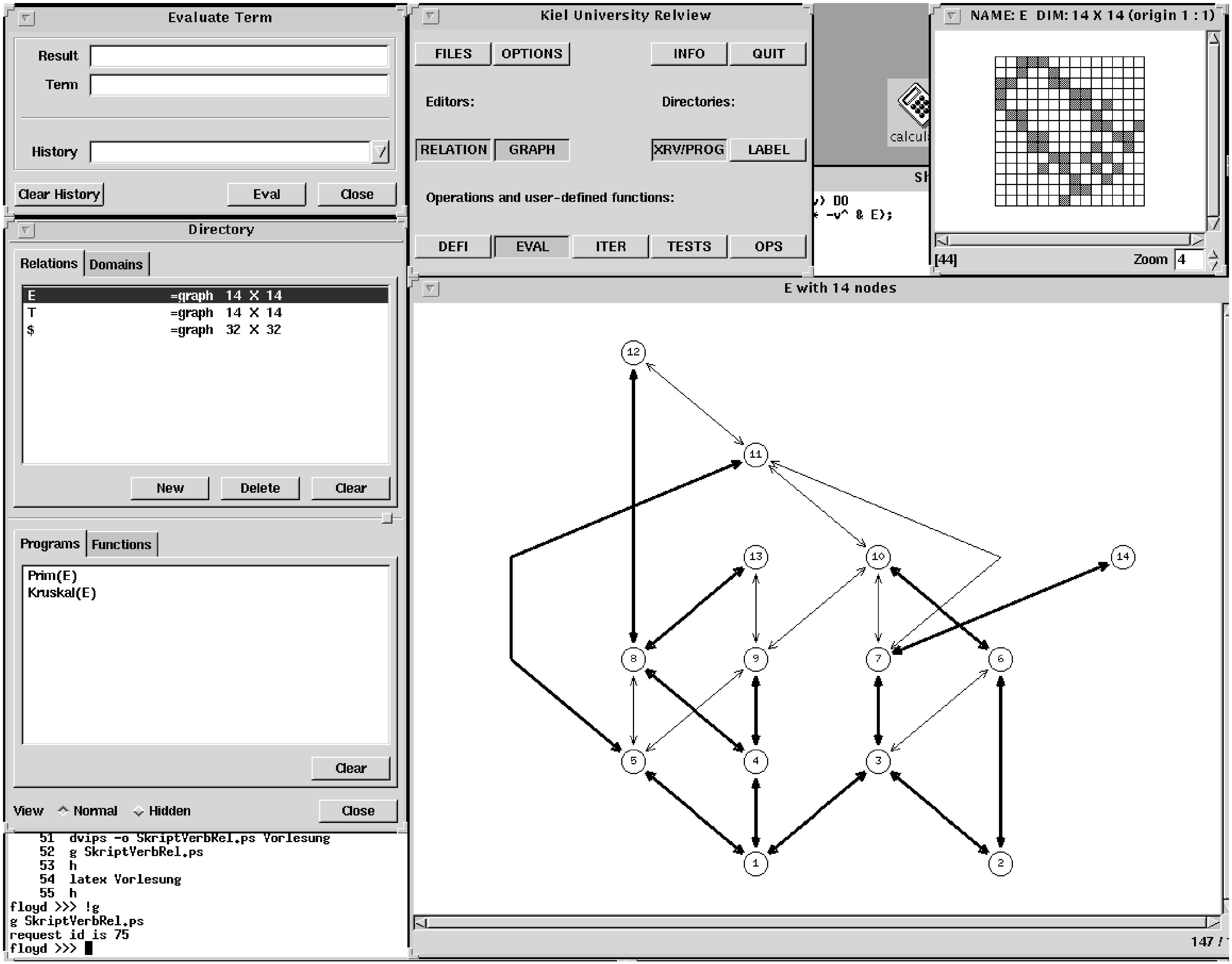
- Instead of $(x, y) \in R$ we use Boolean matrix notation:

$$\boxed{R_{x,y}}$$

Signature of relation algebra:

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

The Relation-Algebraic Tool RelView



Example of a relational function in RelView:

```
Hasse(C) = C & -(C*C).
```

A call 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 $CC \subseteq C$).

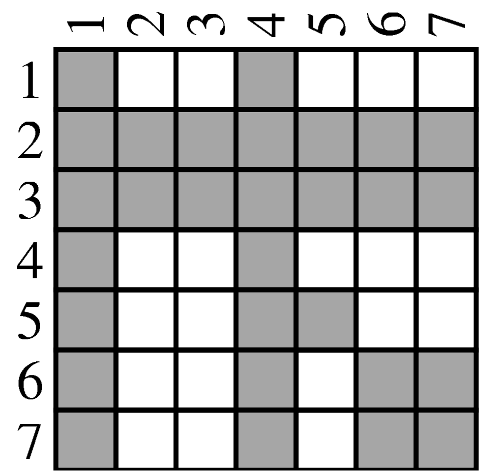
Example of a relational program in RelView:

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

A Motivating Example

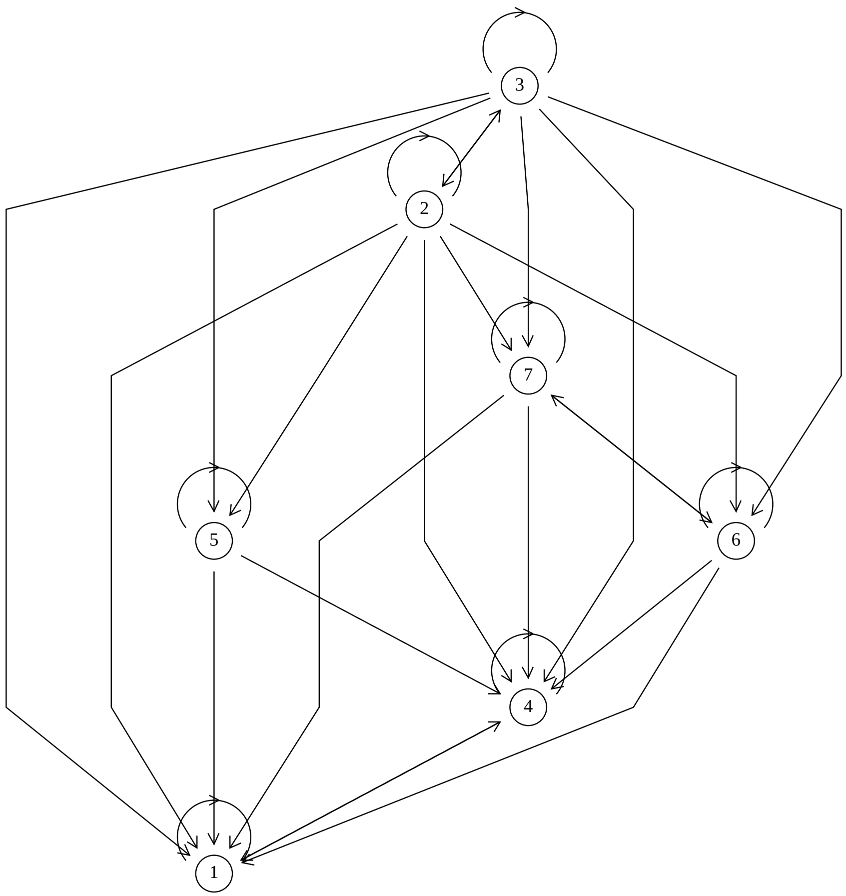
As a rather trivial example we consider a relation on elements $1, 2, \ldots 7$, represented with RelView as the following Boolean matrix.

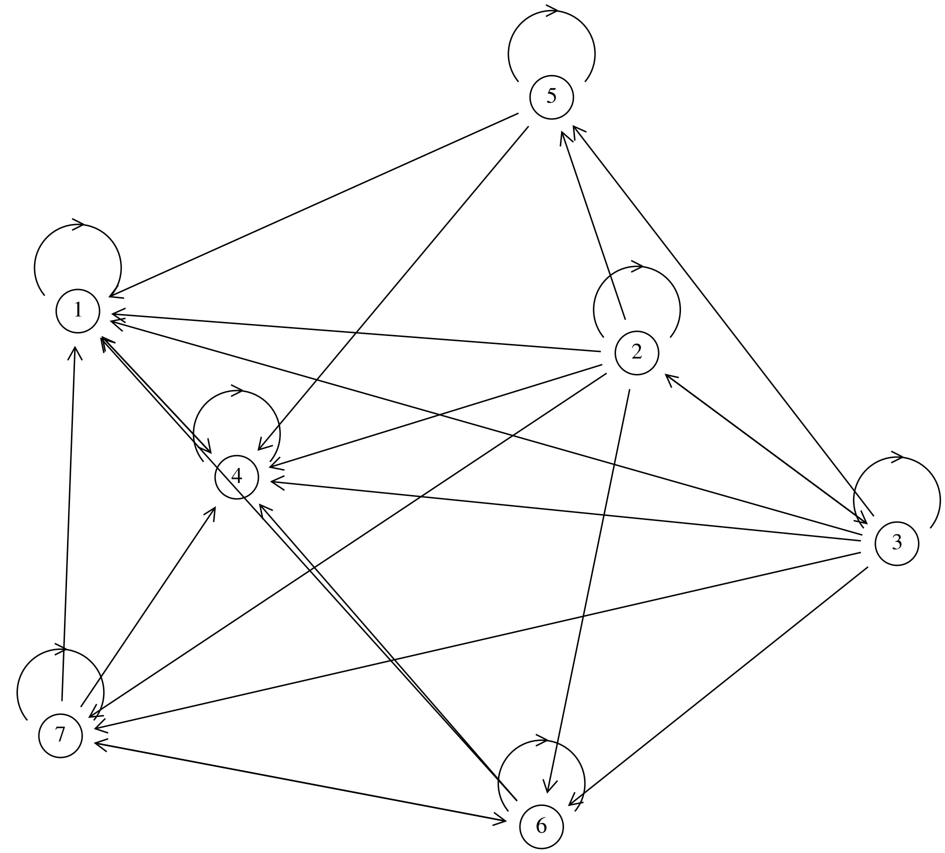


A black square stands for the matrix entry 1 (or *true*) and a white square stands for the entry 0 (or *false*).

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).

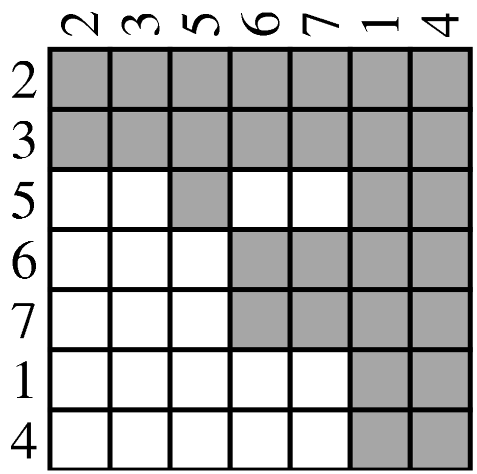


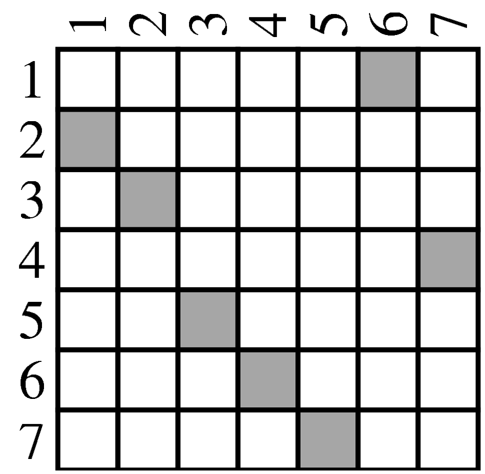


hierarchical polyline drawing

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.





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

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

$$\boxed{P^{\mathsf{T}}RP}$$

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 $\bar{\mathbf{I}} \subseteq \Omega_X \cup \Omega_X^{\mathsf{T}}$), 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 $P^{\mathsf{T}}CP$ is depicted as a full upper right triangle matrix.

The procedure:

- Compute the Hasse-diagrams of C and Ω_X:

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

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

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

- Successively apply the relational function

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

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

- 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

$$P^{\mathsf{T}}CP,$$

where the permutation relation P is the result of PermLSO(C).

Application I: Pre-Orders

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

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^{\mathsf{T}}QP$ 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:

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

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

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

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

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

Application II: Weak-Orders

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

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^{\mathsf{T}} W P$ 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:

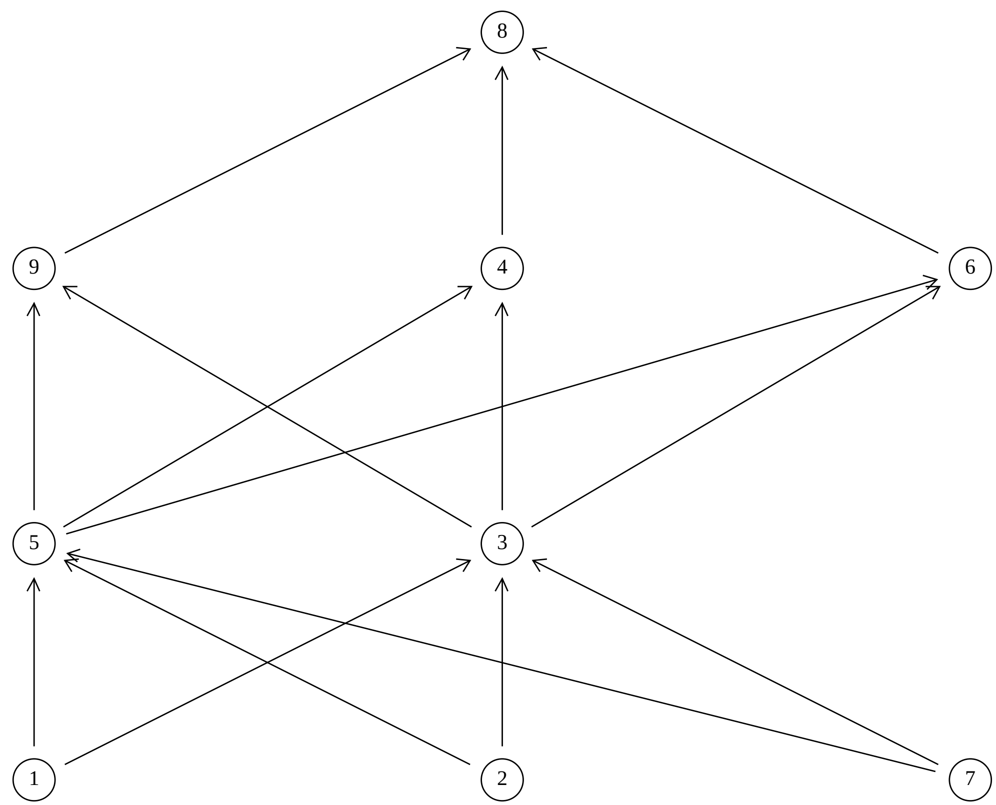
$$E' = \mathsf{Szpilrajn}(E)$$

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

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

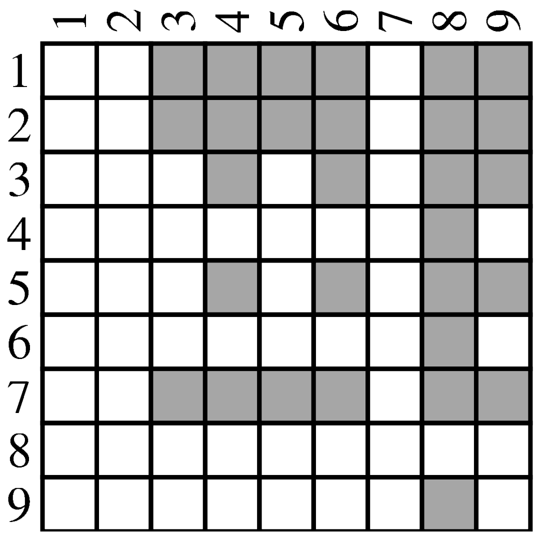
Informally, weak-orders are those strict-orders the Hasse-diagrams of which are composed by a series of complete bipartite strict-orders, one above another.

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

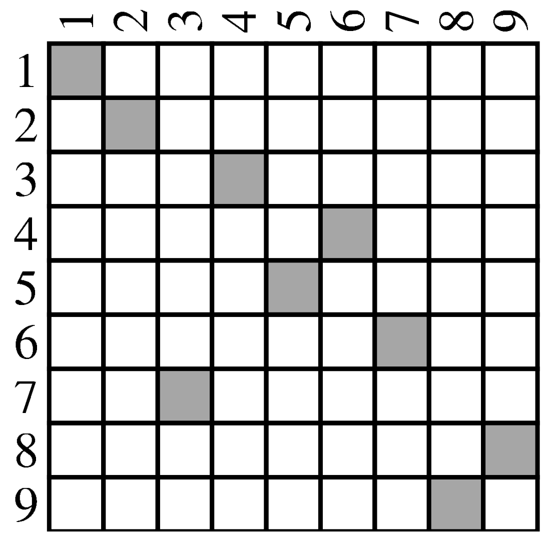


From the block form boundary of the rearranged matrix the complete bipartite strict-orders and their arrangement immediately becomes apparent.

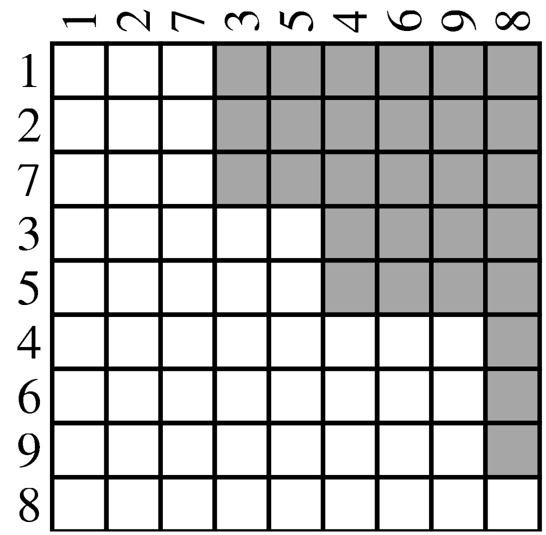
RelView-matrices to the above example:



original matrix



permutation relation



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., $S \subseteq \overline{\mathbf{I}}$, $\overline{S}\,\overline{S} \subseteq \overline{SS}$ and $S\,\overline{S}^{\mathsf{T}}S \subseteq S$, where the latter two inclusions are called semi-transitivity and Ferrers property).

Result: Permutation relation $P : X \leftrightarrow X$ such that $P^{\mathsf{T}}SP$ 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:
$$W = \overline{S}^{\mathsf{T}}S \cup S\,\overline{S}^{\mathsf{T}}$$
- Result a permutation relation that transforms W into an upper right block triangle form:
$$P = \mathsf{PermWO}(W)$$
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 $J\,\overline{J}^{\mathsf{T}} J \subseteq J$).

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

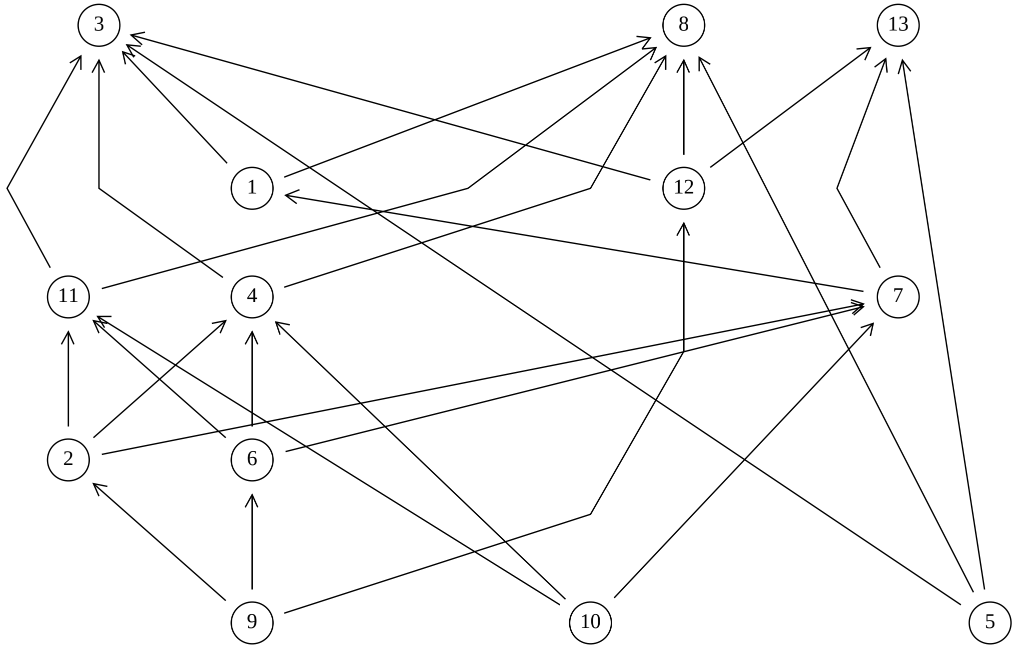
Result: We have to compute

$$P_r : X \leftrightarrow X \quad \text{permutation relation for rows}$$

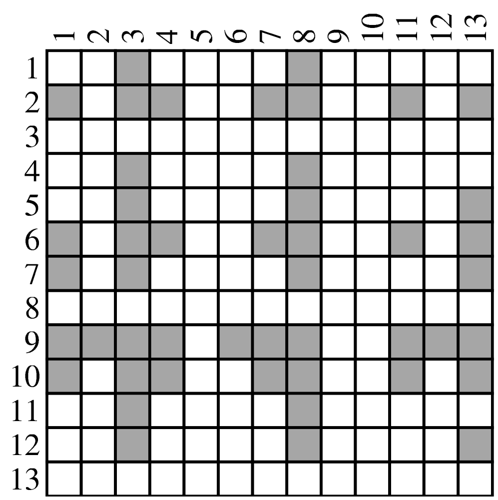
$$P_c : X \leftrightarrow X \quad \text{permutation relation for columns}$$

such that $P_r^{\mathsf{T}} J P_c$ is in upper right staircase block form.

An example:

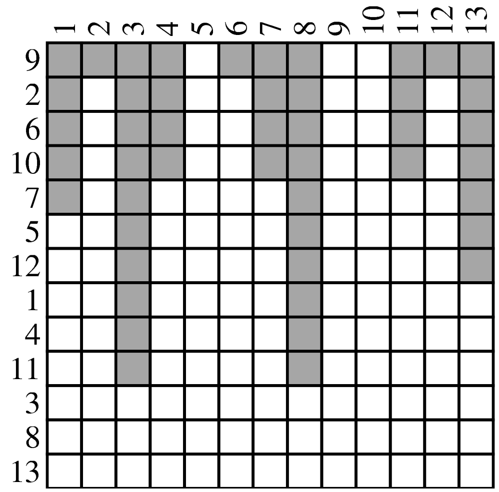


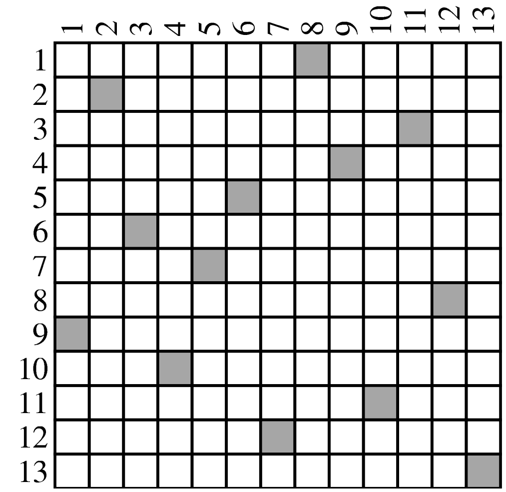
Hasse-diagram of J



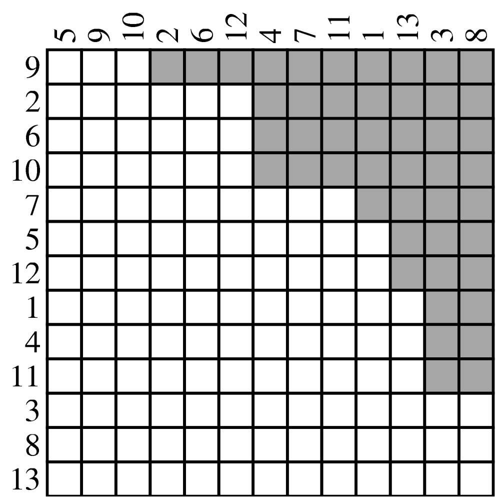
Boolean matrix representation of J

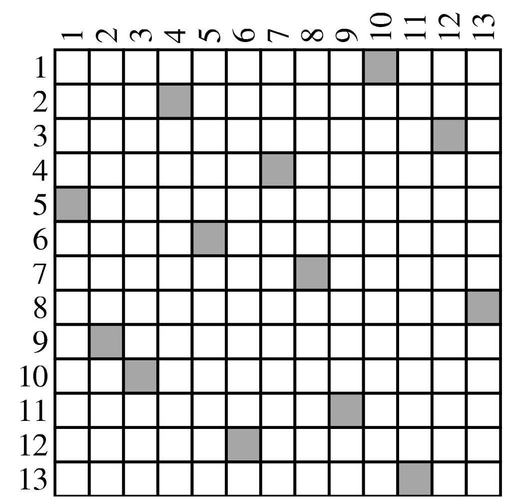
Sorting the rows of J in decreasing inclusion from top to bottom via $P_r^{\mathsf{T}} J$, and the corresponding row permutation relation.





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





Relation-algebraic specification of P_r:

$$P_r = \mathsf{PermPreO}(\,\overline{\overline{J}\,J^{\mathsf{T}}}\,)$$

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

$$\begin{array}{rcl} (\,\overline{\overline{J}\,J^{\mathsf{T}}}\,)_{x,y} & \Longleftrightarrow & \neg\exists\, z : \overline{J}_{x,z} \wedge J^{\mathsf{T}}_{z,y} \\ & \Longleftrightarrow & \forall\, z : J^{\mathsf{T}}_{z,y} \rightarrow J_{x,z} \\ & \Longleftrightarrow & \forall\, z : J_{y,z} \rightarrow J_{x,z} \\ & \Longleftrightarrow & y\text{-row contained in } x\text{-row} \end{array}$$

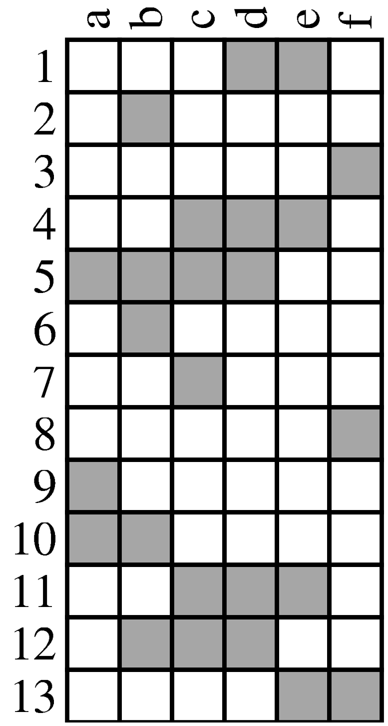
Similar: Calculation of a relation-algebraic specification of P_c:

$$P_c = \mathsf{PermPreO}(\,\overline{J^{\mathsf{T}}\,\overline{J}}\,)$$

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

Some Further Applications

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



intervals of a
linearly ordered
set with 6 elements

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

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

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.