Relational Data Analysis

Gunther Schmidt[1]

Institute for Software Technology, Department of Computing Science
Federal Armed Forces University Munich

e-Mail: `Schmidt@Informatik.UniBw-Muenchen.DE`

Abstract. Given a binary relation R, we look for partitions of its row set and its column set, respectively, that behave well with respect to selected algebraic properties, i.e., correspond to congruences related to R. Permutations are derived from these congruences that allow to rearrange R visualizing the decomposition.

1 Introduction

Known and new methods of decomposing a relation are presented together with methods of making the decomposition visible. Such aspects as difunctionality, Moore-Penrose inverses, independence and line covering, chainability, game decompositions, matchings, Hall conditions, term rank, and others are handled under one common roof.

$$A = \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\15\\16\\17 \end{array} \begin{array}{c} \begin{array}{ccccccccccccc} 1&2&3&4&5&6&7&8&9&10&11&12&13 \end{array} \\ \begin{pmatrix}
0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&1&0&0&0&0&0&0&0
\end{pmatrix} \end{array}$$

An initial example of a sparse relation

One area of application is multicriteria decision making. There, relations are given beforehand and one asks for dichotomies generated by the relation [1, 3]

[1] Cooperation and communication around this research was partly sponsored by the European COST Action 274: TARSKI (Theory and Application of Relational Structures as Knowledge Instruments), which is gratefully acknowledged.

following certain rational ideas. One may call this *theorem extraction* or theorem formulation — as opposed to *theorem proving*. Once formulated, theorem provers would certainly establish the theorem. But in practical situations it is more important *to find* the theorem on which one is willing to base decisions.

Much of the basic approach can be demonstrated by the initial example of a sparse boolean matrix A, which looks rather randomly distributed. It is a big problem in marketing, e.g., to analyze such raw data so as to finally arrive at $A_{\text{rearranged}}$. We have visualized that certain theorems are valid, namely that elements of $\{1, 9, 11, 15, 16, 17\}$ are only related to elements from $\{3, 6, 7, 13\}$, etc.

$$A_{\text{rearranged}} = \begin{array}{r} \\ 1 \\ 9 \\ 11 \\ 15 \\ 16 \\ 17 \\ 2 \\ 3 \\ 6 \\ 7 \\ 4 \\ 13 \\ 14 \\ 5 \\ 8 \\ 10 \\ 12 \end{array} \begin{array}{c} \begin{array}{cccc|ccc|cccc|cc} 3 & 6 & 7 & 13 & 1 & 4 & 8 & 2 & 5 & 9 & 10 & 11 & 12 \end{array} \\ \left(\begin{array}{cccc|ccc|cccc|cc} 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ \hline 0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&1&0&0 \\ \hline 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \end{array}\right) \end{array}$$

The initial example partitioned and rearranged

Having knowledge of this type may obviously be considered a major marketing advantage. Our basic question is, therefore: Are there methods to generate such theorems? One will certainly not blindly generate all theorems and scan them in some sort of a Laputa method [7] for interesting ones. Rather, there should be some concept as to how these theorems might look like. In the environment of RelMiCS, one will concentrate on "relation-algebraic theorem patterns". Several of these have been investigated and formulated as ontologies. By ontology we mean relational games, irreducibility, line covering, or matching concepts, e.g. With one of these ontologies, the relation in question is analyzed.

The full report underlying this article, obtainable via

`http://ist.unibw-muenchen.de/People/schmidt/DecompoHomePage.html`,

strives to provide the methodological basis together with a toolkit for the task of decomposing relations in various application disciplines. This report is not just a research report but also a Haskell program in literate style sufficient to run the programs.

2 Prerequisites and Tools

We have used Haskell [2] as the underlying programming language. It is a purely functional language, and it is currently widely accepted in research and university teaching. For more information see the Haskell WWW site at

`URL: http://www.haskell.org/`.

In the example presented in the introduction, the resulting permutation has only been shown via the permuted row and column numbers. The permutations, however, should be fully available in the program. There, they may be given as a function, decomposed into cycles, or as a permutation matrix. Either form has its specific merits. Sometimes also the inverted permutation is useful. Therefore, types and functions to switch from one form to the other, and to apply a permutation to some list are provided for. Also a basic technique is introduced determining permutations from partitions. Let, for example, a list of partitioning subsets be given as

```
[[False,True, False,True, False,False],
 [False,False,True, False,False,True ],
 [True, False,False,False,True, False]]
```

We are interested to obtain a permutation like `[5,1,3,2,6,4]`, which directs `True` entries of the first row to the front, followed by `True` entries of the second row, etc.

Relations are throughout handled as rectangular Boolean matrices. Often we represent their entries `True` by **1** and `False` by **0** when showing matrices in the text. The basic relational operators `&&&, |||, ***, <==` for intersection, union, composition, containment, etc., of relations are all formulated in Haskell.

3 Congruences

Whenever some equivalence behaves well with regard to some other structure, we are accustomed to call it a congruence. This is well-known for algebraic structures, i.e., those defined by mappings on some set. We define it correspondingly for the non-algebraic case, including heterogeneous relations. Let R be a relation and Ξ, Θ be equivalences on the source, resp. the target set of that relation. Then the pair (Ξ, Θ) is called a R**-congruence** if $\Xi ; R \subseteq R ; \Theta$.

We have formulated a generalisation of Birkhoffs famous theorem on the lattice of congruences for algebraic structures, extending it to relational structures.

Proposition 1. Let some finite heterogeneous relation R be given. Then all R-congruences (P, Q) which satisfy both, $R ; R^{\mathsf{T}} \subseteq P$ and $R^{\mathsf{T}} ; R \subseteq Q$, form a complete lattice, the least element of which is $(\Xi, \Theta) := ((R ; R^{\mathsf{T}})^*, (R^{\mathsf{T}} ; R)^*)$, the pair of natural congruences wrt. R.

Proof: Among other items, the proof has to show that the set

$\{(P, Q) \mid (P, Q) \text{ is an } R\text{-congruence satisfying } R ; R^{\mathsf{T}} \subseteq P \text{ and } R^{\mathsf{T}} ; R \subseteq Q\}$

is $\cap$-hereditary:

$$\begin{aligned}(P_1 \cap P_2);R &\subseteq P_1;R \cap P_2;R \subseteq R;Q_1 \cap R;Q_2\\ &\subseteq (R \cap R;Q_2;Q_1^{\mathsf{T}});(Q_1 \cap R^{\mathsf{T}};R;Q_2) \text{ using Dedekind's rule}\\ &\subseteq R;(Q_1 \cap Q_2;Q_2) = R;(Q_1 \cap Q_2)\end{aligned}$$ $\square$

Every congruence leads to a partition in equivalence classes, and these partitions in turn give rise to the permutations we compute for presentation purposes.

4 Difunctional Decomposition

Two different cases of decomposition will be distinguished: For *heterogeneous* relations between two sets — be it that they have equal cardinality —, we will permute rows and columns *independently* and for *homogeneous* relations rows and columns will be permuted *simultaneously.*

In the introductory example, difunctionality leads to an important decomposition. It groups rows as well as columns according to the "natural congruence".

A relation R is called **difunctional** if $R;R^{\mathsf{T}};R = R$. For every relation R, the least difunctional relation containing it is well defined and we define (according to J. Riguet, [4]) the difunctional closure as

$$h_{\text{difu}}(R) := \texttt{inf}\,\{H \mid R \subseteq H \text{ with } H \text{ difunctional}\}$$

We also ask for the practical aspects of this definition, which has long been discussed and is known among numerical analysts. To describe the main property of any diagonal block of the rearranged initial example, let a relation R be given that is conceived as a chessboard with dark squares or white according to whether R_{ik} is `True` or `False`. A rook shall operate on the chessboard in horizontal or vertical direction; however, it is only allowed to change direction on dark squares. Using this interpretation, a relation R is called **chainable** if the non-vanishing entries (i.e., the dark squares) can all be reached from one another by a sequence of "rook moves", or else if $h_{\text{difu}}(R) = \mathbb{T}$. The diagonal blocks mentioned are chainable when considered as a separate matrix. It is easy to prove that a total and surjective relation R is chainable precisely when its so-called edge-adjacency $K := \overline{\mathbb{I}} \cap R;R^{\mathsf{T}}$ is strongly connected.

The concept of being difunctional is related to linear algebra for numerical problems. A relation G is called a **Moore-Penrose inverse** of A if the following four conditions hold $A;G;A = A$, $G;A;G = G$, $(A;G)^{\mathsf{T}} = A;G$, $(G;A)^{\mathsf{T}} = G;A$. Moore-Penrose inverses are uniquely determined provided they exist. The Moore-Penrose inverse of a difunctional matrix always exists and turns out to be its converse.

5 Line Covering and Independence

Some relations may be decomposed in such a way, that there is a subset of row entries that is completely unrelated to a subset of column entries. In this context, a relation A may admit vectors x and y (with $\mathbb{⊥} \neq x \neq \mathbb{T}$ or $\mathbb{⊥} \neq y \neq \mathbb{T}$ to avoid degeneration), such that $A; y \subseteq x$ or, equivalently, $A \subseteq \overline{\overline{x}; y^\mathsf{T}}$. Given appropriate permutations P of the rows, and Q of the columns, respectively, we then have

$$P; A; Q^\mathsf{T} = \begin{pmatrix} * & \mathbb{⊥} \\ * & * \end{pmatrix} \qquad P; x = \begin{pmatrix} \mathbb{⊥} \\ \mathbb{T} \end{pmatrix} \qquad Q; y = \begin{pmatrix} \mathbb{⊥} \\ \mathbb{T} \end{pmatrix}.$$

Given $A; y \subseteq x$, to enlarge the $\mathbb{⊥}$-zone is not so easy a task, which may be seen in the case of the identity relation $\mathbb{I}$: All shapes from $1 \times (n-1)$, $2 \times (n-2)$, $\ldots (n-1) \times 1$ may be chosen. To avoid this counter-running, one usually studies this effect with one of the sets x and y negated. So we consider pairs of subsets (s, t) taken from the domain and from the range side and define

(s, t) is a **line covering** $\quad :\Longleftrightarrow \quad A; \overline{t} \subseteq s.$

(s, t) is a **pair of independent sets** $\quad :\Longleftrightarrow \quad A; t \subseteq \overline{s}.$

For the moment, call rows and columns lines. Then we are able to cover all entries **1** by $|\overline{y}|$ vertical plus $|x|$ horizontal lines. Given a relation A, the **term rank** is defined as the minimum number of lines necessary to cover all entries **1** in A, i.e. $\mathtt{min}\{|s| + |t| \mid A; \overline{t} \subseteq s\}$.

Consider $\begin{pmatrix} A_{11} & \mathbb{⊥} \\ A_{21} & A_{22} \end{pmatrix}; \begin{pmatrix} \mathbb{⊥} \\ \mathbb{T} \end{pmatrix} \subseteq \begin{pmatrix} \mathbb{⊥} \\ \mathbb{T} \end{pmatrix}$. Hoping to arrive at fewer lines than the columns of A_{11} and the rows of A_{22} to cover, one might start a first naive attempt and try to cover with s and t but with row i, e.g., omitted. If (s, t) is already minimal, there will be an entry in row i of A_{22} containing a **1** . Therefore, A_{22} is a total relation. In the same way, A_{11} turns out to be surjective. But we may also try to get rid of a set $x \subseteq s$ of rows and accept that a set of columns be added instead. It follows from minimality that regardless of how we choose $x \subseteq s$, there will be at least as many columns necessary to cover what has been left out. For a relation A and a set x, we therefore say that x satisfies the **Hall condition** if $|z| \leq |A^\mathsf{T}; z|$ holds for every subset $z \subseteq x$. If we have a line covering with $|s| + |t|$ minimal, then A_{11}^T as well as A_{22} will satisfy the Hall-condition.

We will later find minimum line coverings and maximum independent sets without just checking them all exhaustively. We postpone this, until further prerequisites are at hand and concentrate on the following aspect.

Proposition 2. Let a finite relation A be given. Then A is either chainable or it admits a pair (s, t) which is nontrivial, such that both s, t as well as $\overline{s}, \overline{t}$, constitute at the same time a pair of independent sets and a line covering.

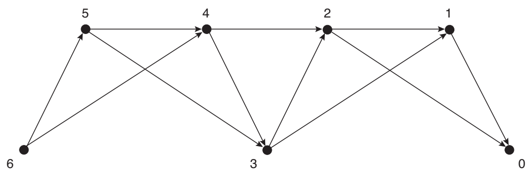
Difunctionality and line coverings are related in the following way.

Proposition 3. A relation A admits a pair (x, y) such that (x, y) and $(\overline{x}, \overline{y})$ are line coverings if and only if its difunctional closure admits these line coverings.

6 Game Decomposition

While so far heterogeneous relations have been treated by permuting their rows and columns independently, we now specialize to the homogeneous case and apply permutations simultaneously. There is a well-developed theory of standard iterations for Boolean matrices in order to solve a diversity of application problems such as matching, line covering, assignment, games, etc. We will present a general framework for executing these iterations.

In all of these cases, we need two antitone mappings between power sets, which we call $\sigma : \mathcal{P}(V) \to \mathcal{P}(W)$ and $\pi : \mathcal{P}(W) \to \mathcal{P}(V)$, according to [5, 6]. In case $V = W$, let an arbitrary homogeneous relation $B : V \leftrightarrow V$ be given with $\pi(x) = \sigma(x) = \overline{B;x}$. Two players are supposed to make moves alternatingly according to B in choosing a consecutive arrow to follow. The player who has no further move, i.e., who is about to move and finds an empty row in the relation B, or a terminal vertex in the graph, has lost.



$$\begin{array}{c|c} & \begin{matrix}0&1&2&3&4&5&6\end{matrix} \\ \begin{matrix}0\\1\\2\\3\\4\\5\\6\end{matrix} & \begin{pmatrix}0&0&0&0&0&0&0\\1&0&0&0&0&0&0\\1&1&0&0&0&0&0\\0&1&1&0&0&0&0\\0&0&1&1&0&0&0\\0&0&0&1&1&0&0\\0&0&0&0&1&1&0\end{pmatrix}\end{array}$$

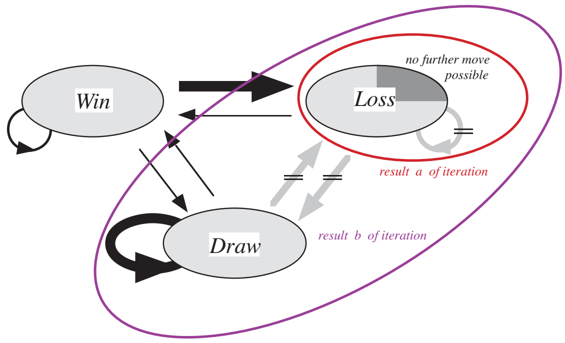
Situations in a game starting with a pile of 6 matches

Such a game is easily visualized by taking a relation B represented by a graph, on which players have to determine a path in an alternating way. We study the Nim game starting with 6 matches from which we are allowed to take 1 or 2.

We have a homogeneous relation, and we easily observe how with equalities in an alternating pattern an iteration evolves

$$\perp\!\!\!\perp \subseteq \overline{B;\top\!\!\!\top} = \overline{B;\overline{B;\perp\!\!\!\perp}} \subseteq \overline{B;\overline{B;\overline{B;\top\!\!\!\top}}} = \ldots \subseteq \ldots \subseteq \overline{B;\overline{B;\overline{B;\perp\!\!\!\perp}}} = \overline{B;\overline{B;\top\!\!\!\top}} \subseteq \overline{B;\perp\!\!\!\perp} = \top\!\!\!\top.$$

The limit of the iteration is characterised by the formulae $a = \pi(b)$ and $\sigma(a) = b$, which this time turn out to be $a = \overline{B;b}$ and $\overline{B;a} = b$. In addition, we will always have $a \subseteq b$. The smaller set a gives loss positions, while the larger one then indicates win positions as $\overline{b}$ and draw positions as $b \cap \overline{a}$. This is visualized by the following diagram for sets of win, loss, and draw, the arrows of which indicate moves that must exist (the thick black arrows), may exist (the thin arrows), or are not allowed to exist (the crossed out grey arrows).



Decompositions into sets of winning, draw, and losing positions

Proposition 4. Any finite homogeneous relation may by simultaneously permuting rows and columns be transformed into a matrix satisfying the following basic structure with square diagonal entries:

$$\begin{pmatrix} \perp\!\!\!\perp & \perp\!\!\!\perp & * \\ \perp\!\!\!\perp & \text{total} & * \\ \text{total} & * & * \end{pmatrix}$$ □

Of course, the win or the loss zone may contain no rows. If the win zone is nonempty, so is the loss zone. Nevertheless, the subdivision into loss/draw/win groups is uniquely determined, and indeed

$$a = \begin{pmatrix} \top\!\!\!\top \\ \perp\!\!\!\perp \\ \perp\!\!\!\perp \end{pmatrix} = \overline{\begin{pmatrix} \perp\!\!\!\perp & \perp\!\!\!\perp & * \\ \perp\!\!\!\perp & \text{total} & * \\ \text{total} & * & * \end{pmatrix} ; \begin{pmatrix} \top\!\!\!\top \\ \top\!\!\!\top \\ \perp\!\!\!\perp \end{pmatrix}} \qquad b = \begin{pmatrix} \top\!\!\!\top \\ \top\!\!\!\top \\ \perp\!\!\!\perp \end{pmatrix} = \overline{\begin{pmatrix} \perp\!\!\!\perp & \perp\!\!\!\perp & * \\ \perp\!\!\!\perp & \text{total} & * \\ \text{total} & * & * \end{pmatrix}} ; \begin{pmatrix} \top\!\!\!\top \\ \perp\!\!\!\perp \\ \perp\!\!\!\perp \end{pmatrix}$$

7 Matching and Assignment

An additional antimorphism situation is known to exist in connection with matchings and assignments. Let two matrices $Q, \lambda : V \leftrightarrow W$ be given, where $\lambda \subseteq Q$ is univalent and injective, i.e. a matching — possibly not yet of maximum cardinality.

We consider Q to be a relation of sympathy between a set of boys and a set of girls and λ the set of current dating assignments, assumed only to be established if sympathy holds. We now try to maximize the number of dating assignments.

Definition 5. Given the (possibly heterogeneous) relation Q, we call the relation λ a **Q-matching** provided it is a univalent and injective relation contained in Q, i.e., if

$$\lambda \subseteq Q \qquad \lambda ; \lambda^{\mathsf{T}} \subseteq \mathbb{I}, \qquad \lambda^{\mathsf{T}} ; \lambda \subseteq \mathbb{I}.$$

An iteration with $\pi(w) := \overline{\lambda ; w}$ and $\sigma(v) := \overline{Q^{\mathsf{T}} ; v}$ will end with two vectors (a, b) satisfying $a = \pi(b)$ and $\sigma(a) = b$ as before. Here, this means $\overline{a} = \lambda b$ and $\overline{b} = Q^{\mathsf{T}} ; a$.

In addition $\overline{a} = Q{;}b$. This follows from the chain $\overline{a} = \lambda{;}b \ \subseteq\ Q{;}b \ \subseteq\ \overline{a}$, which implies equality at every intermediate state. Only the resulting equalities for a, b have been used together with monotony and the Schröder rule to obtain this decomposition.

We find out that the pair a, b is an inclusion-maximal pair of independent sets for Q, or else $\overline{a}, \overline{b}$ is an inclusion-minimal line covering. As of yet, a, b need not be an inclusion-maximal pair of independent sets for λ, nor need $\overline{a}, \overline{b}$ be an inclusion-minimal line covering for λ! This will only be the case, when in addition $\overline{b} = \lambda^{\mathsf{T}}{;}a$.

It is thus not uninteresting to concentrate on the condition $\overline{b} = \lambda^{\mathsf{T}}{;}a$. After having found some matching relation and applying the iteration, it may not yet be satisfied. So let us assume $\overline{b} = \lambda^{\mathsf{T}}{;}a$ *not* to hold, which means that $\overline{b} = Q^{\mathsf{T}}{;}a \underset{\neq}{\supseteq} \lambda^{\mathsf{T}}{;}a$.

We make use of the formula $\lambda{;}\overline{S} = \lambda{;}\mathbb{T} \cap \overline{\lambda{;}S}$, which holds since λ is a univalent relation. The iteration finally ends with equations $\overline{b} = Q^{\mathsf{T}}{;}a$ and $\overline{a} = \lambda{;}b$. This easily expands to

$$\overline{b} = Q^{\mathsf{T}}{;}a = Q^{\mathsf{T}}{;}\overline{\lambda{;}b} = Q^{\mathsf{T}}{;}\overline{\lambda{;}\overline{Q^{\mathsf{T}}{;}a}} = Q^{\mathsf{T}}{;}\overline{\lambda{;}\overline{Q^{\mathsf{T}}{;}\overline{\lambda{;}\overline{Q^{\mathsf{T}}{;}a}}}} \ \ldots$$

from which the last but one becomes

$$\begin{aligned}\overline{b} &= Q^{\mathsf{T}}{;}a = Q^{\mathsf{T}}{;}\overline{\lambda{;}b} = Q^{\mathsf{T}}{;}\overline{\lambda{;}\mathbb{T} \cap \overline{\lambda{;}Q^{\mathsf{T}}{;}a}} = Q^{\mathsf{T}}{;}(\overline{\lambda{;}\mathbb{T}} \cup \lambda{;}Q^{\mathsf{T}}{;}a)\\ &= Q^{\mathsf{T}}{;}(\overline{\lambda{;}\mathbb{T}} \cup \lambda{;}Q^{\mathsf{T}}{;}(\overline{\lambda{;}\mathbb{T}} \cup \lambda{;}Q^{\mathsf{T}}{;}a))\end{aligned}$$

indicating how to prove that

$$\overline{b} = (Q^{\mathsf{T}} \cup Q^{\mathsf{T}}{;}\lambda{;}Q^{\mathsf{T}} \cup Q^{\mathsf{T}}{;}\lambda{;}Q^{\mathsf{T}}{;}\lambda{;}Q^{\mathsf{T}} \cup \ \ldots\ ){;}\overline{\lambda{;}\mathbb{T}}$$

If we have $\lambda^{\mathsf{T}}{;}a \underset{\neq}{\subseteq} \overline{b}$, we may thus find a point in

$$(Q^{\mathsf{T}} \cup Q^{\mathsf{T}}{;}\lambda{;}Q^{\mathsf{T}} \cup Q^{\mathsf{T}}{;}\lambda{;}Q^{\mathsf{T}}{;}\lambda{;}Q^{\mathsf{T}} \cup \ \ldots\ ){;}\overline{\lambda{;}\mathbb{T}} \cap \overline{\lambda^{\mathsf{T}}{;}a}$$

which leads to the well-known alternating chain algorithm.

When showing the result, some additional care will be taken concerning empty rows or columns in Q showing them at the first or at the last position, respectively. The Hall condition is made visible by arranging λ in a diagonal shape and by introducing an additional subdividing line. In principle, this gives a 4 by 4 pattern. Either or all of the first 3 row zones, or the last 3 column zones may be empty.

$$
\begin{array}{r c}
 & \begin{array}{ccccc|ccccccccc|c|cc} 3 & 15 & 13 & 17 & 4 & 9 & 14 & 1 & 5 & 7 & 12 & 8 & 16 & 2 & 11 & 6 & 10 \end{array} \\
\begin{array}{r} 7\\ 16\\ 6\\ 12\\ 14\\ 1\\ 2\\ 4\\ 5\\ 10\\ 3\\ 8\\ 9\\ 11\\ 13\\ 15\\ 17\\ 18\\ 19 \end{array} &
\left(\begin{array}{ccccc|ccccccccc|c|cc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\hline
1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\hline
1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
\hline
1&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0\\
1&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&0&0&1&0&0&0&1&0&1&0&0&1&1&0&0
\end{array}\right)
\end{array}
$$

Arbitrary heterogeneous relation with a rearrangement according to a cardinality-maximum matching — the diagonals

Proposition 6. Any given heterogeneous relation Q admits a cardinality-maximum matching $\lambda \subseteq Q$. By independently permuting rows and columns they can jointly be transformed into matrices of a 2 by 2 pattern with not necessarily square diagonal blocks of the following form:

$$
Q = \begin{pmatrix} \mathrm{Hall}^{\mathsf{T}} & \perp\!\!\!\perp \\ * & \mathrm{Hall} \end{pmatrix} \qquad \lambda = \begin{pmatrix} \text{univ. + surject. + inject.} & \perp\!\!\!\perp \\ \perp\!\!\!\perp & \text{univ. + total + inject.} \end{pmatrix}
$$

Looking back, we see that we have not just decomposed according to a pair of independent sets, but in addition ordered the rest of the relation so as to obtain matchings.

8 Conclusion and Outlook

Based on such investigations, it is possible to automatically develop *theories* combined with some relations, starting from a given ontology (difunctionality, game, e.g.). When applying a relational decomposition, this theory will change as it afterwards fits into the given ontology and, thus, includes the necessary predicates and theorems.

The following diagram shows the idea. There are concrete relations given and the sparse theory with which one may work on these. By this we mean hardly more than asking whether two elements are related. There is, however, also an ontology concerning games, irreducibility, or difunctionality, e.g. What we are constructing is in a sense the pushout. Afterwards, the given model may be viewed with the win-loss-draw structure, e.g., which the game ontology has provided. Then the theory will also contain certain closed formulae describing what holds between the new items.

$$\begin{array}{ccc} \text{Sparse theory} & \models & \text{Given model} \\ \downarrow & & \downarrow \\ \text{Ontology-enhanced theory} & \models & \text{Result model} \end{array}$$

With the methods presented it is possible to analyze a given relation with regard to different concepts and to visualize the results. Some programs are more efficient, others less. This was not a matter of concern in this paper, though. We had in mind data of a size not too big to be handled and even visualized for presentation purposes.

There is another point of possible criticism that matters more. Should the relations handled stem from rather fuzzy sources, taken by some α-cut, e.g., the results will heavily depend on single entries of the relation. On the other hand, such a single entry may be a rather "weak" one as to its origin from the α-cut as it has hardly passed the threshold. In the approach chosen, therefore, we have extremely high sensitivity depending on the initial data. This is a feature, one is usually not interested in. Rather, one would highly estimate results which stay more or less the same as data are changed only moderately. This would give more "meaning" to the results and would make it easier to base decisions on them.

Nonetheless, we hope that our exposition will lead to future research. We have scanned a diversity of topics for their algebraic properties. On several occasions, we have replaced counting arguments by algebraic ones. Our hope is that these algebraic properties will be of value in the following regard: Only recently, fuzzy relations have been investigated with more intensity. There, the entry of the matrix is not just **1** / **0** or yes/no. Instead, coefficients from some suitable lattice are taken. In this way it is possible to express given situations in more detail. Astonishingly, much of the algebraic structure of relation algebra still remains valid. In this modified context an investigation should be done using fuzzy relations. Reduced sensitivity of results with respect to given data may be hoped for.

Acknowledgment. The author gratefully acknowledges fruitful discussions with Michael Ebert as well as detailed comments of the referees.

References

1. Džeroski, S., and Lavrač, N., Eds. *Relational Data Mining*. Springer-Verlag, 2001.
2. Hudak, P., Jones, S. L. P., Wadler, P., et al. Report on the programming language Haskell, a non-strict purely functional language, version 1.2. *ACM SIGPLAN Notices 27*, 5 (1992). See also http://haskell.org/.

3. KITAINIK, L. *Fuzzy Decision Procedures With Binary Relations — Towards a Unified Theory*, vol. 13 of *Theory and Decision Library, Series D: System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, 1993.
4. RIGUET, J. Quelques propriétés des relations difonctionelles. *C. R. Acad. Sci. Paris 230* (1950), 1999–2000.
5. SCHMIDT, G., AND STRÖHLEIN, T. *Relationen und Graphen*. Mathematik für Informatiker. Springer-Verlag, 1989. ISBN 3-540-50304-8, ISBN 0-387-50304-8.
6. SCHMIDT, G., AND STRÖHLEIN, T. *Relations and Graphs — Discrete Mathematics for Computer Scientists*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1993. ISBN 3-540-56254-0, ISBN 0-387-56254-0.
7. SWIFT, J. Gulliver's travels, 1726.