

A FIXED POINT THEOREM FOR STRONGER ASSOCIATION RULES AND ITS COMPUTATIONAL ASPECTS

GÁBOR CZÉDLI

ABSTRACT. Each relation induces a new closure operator, which is (in the sense of data mining) stronger than or equal to the Galois one. The goal is to give some evidence that the new closure operator is often properly stronger than the Galois one. An easy characterization of the new closure operator as a largest fixed point of an appropriate contraction map leads to a (modest) computer program. Finally, various experimental results obtained by this program give the desired evidence.

1. INTRODUCTION TO A NEW CLOSURE OPERATOR

Although the terminology of formal concept analysis is frequently used in the first two sections, the paper belongs neither to formal concept analysis nor to other applied fields about data bases. These fields are used only as a part of motivations of the present study. However, in the converse direction, there is some hope that this section and mainly the next one may give some motivation to these fields. This section is devoted only to definitions and some basic properties; our motivations will be given in the next section.

Following Wille's terminology, cf. [11] or [7], a triplet

$$(A^{(0)}, A^{(1)}, \rho)$$

is called a *context* if $A^{(0)}$ and $A^{(1)}$ are nonempty sets and $\rho \subseteq A^{(0)} \times A^{(1)}$ is a binary relation. From what follows, we fix a context $(A^{(0)}, A^{(1)}, \rho)$ and let

$$\rho_0 = \rho \text{ and } \rho_1 = \rho^{-1}.$$

From now on, unless otherwise stated, i will be an arbitrary element of $\{0, 1\}$. So whatever we say including i without specification, it will be understood as prefixed by $\forall i$. The set of all subsets of $A^{(i)}$ will be denoted by $P(A^{(i)})$.

It is often, especially in the finite case, convenient to depict our context in the usual form: a binary table with row labels from $A^{(0)}$, column labels from $A^{(1)}$, and a cross in the intersection of the x -th row and the y -th column iff $(x, y) \in \rho$. We will refer to this table as the *context table*. For example, a context is given by Table 1. (In an appropriate sense, cf. Table 2, this is the smallest interesting example.) The

Date: August 21, 2007, revised July 15, 2008.

Key words and phrases. Association rule, database, data mining, lattice, poset, context, concept lattice, formal concept analysis, Galois connection, functional dependency.

This research was partially supported by the NFSR of Hungary (OTKA), grant no. T 049433 and K 60148.

concrete meaning of this context is not relevant for this paper¹.

	b_1	b_2	b_3	b_4
a_1	×	×		
a_2	×		×	
a_3	×	×		×
a_4	×			
a_5		×	×	×

Table 1

A mapping $\mathcal{D}^{(i)} : P(A^{(i)}) \rightarrow P(A^{(i)})$ is called a *closure operator* if it is *extensive* (i.e., $X \subseteq \mathcal{D}^{(i)}(X)$ for all $X \in P(A^{(i)})$), *monotone* (i.e., $X \subseteq Y$ implies $\mathcal{D}^{(i)}(X) \subseteq \mathcal{D}^{(i)}(Y)$), and *idempotent* (i.e., $\mathcal{D}^{(i)}(\mathcal{D}^{(i)}(X)) = \mathcal{D}^{(i)}(X)$ for all $X \in P(A^{(i)})$). By a *pair of extensive operators* we mean a pair $\mathcal{D} = (\mathcal{D}^{(0)}, \mathcal{D}^{(1)})$ where $\mathcal{D}^{(i)} : P(A^{(i)}) \rightarrow P(A^{(i)})$ is an extensive mapping for $i = 0, 1$. If these mappings are closure operators then \mathcal{D} is called a *pair of closure operators*.

If $\mathcal{D} = (\mathcal{D}^{(0)}, \mathcal{D}^{(1)})$ and $\mathcal{E} = (\mathcal{E}^{(0)}, \mathcal{E}^{(1)})$ are pairs of extensive operators then $\mathcal{D} \leq \mathcal{E}$ means that $\mathcal{D}^{(i)}(X) \subseteq \mathcal{E}^{(i)}(X)$ for all $i \in \{0, 1\}$ and all $X \in P(A^{(i)})$.

Now, associated with $(A^{(0)}, A^{(1)}, \rho)$, we define some pairs of closure operators. The motivation will be given afterwards. For $X \in P(A^{(i)})$ let

$$X\rho_i = \{y \in A^{(1-i)} : \text{for all } x \in X, (x, y) \in \rho_i\},$$

and, again for $X \in P(A^{(i)})$, define

$$\mathcal{G}^{(i)}(X) := (X\rho_i)\rho_{1-i} = \bigcap_{y \in X\rho_i} (\{y\}\rho_{1-i}).$$

Then $\mathcal{G} = (\mathcal{G}^{(0)}, \mathcal{G}^{(1)})$ is the well-known *pair of Galois closure operators*, which plays the main role in formal concept analysis, cf. Wille [11] and Ganter and Wille [7]. The visual meaning of

$$\mathcal{G} = \mathcal{G}(A^{(0)}, A^{(1)}, \rho)$$

is the following. The maximal subsets of ρ of the form $U^{(0)} \times U^{(1)}$ with $U^{(i)} \subseteq A^{(i)}$ are called the (formal) *concepts*, cf. [11] or [7]. Pictorially, they are the maximal full rectangles $U^{(0)} \times U^{(1)}$ of the context table. (Full means that each entry is a cross.) For $X_i \in P(A^{(i)})$ take all maximal full rectangles $U^{(0)} \times U^{(1)}$ such that $X \subseteq U^{(i)}$, then $\mathcal{G}^{(i)}(X)$ is the intersection of all the $U^{(i)}$'s.

Now we define a sequence \mathcal{C}_i , $i = 0, 1, 2, \dots$, of pairs of closure operators. For $X \in P(A^{(i)})$ let

$$X\psi_i := \{Y \in P(A^{(1-i)}) : \text{there is a surjection } \varphi : X \rightarrow Y \text{ with } \varphi \subseteq \rho_i\}.$$

Pictorially, the elements of $X\psi_i$ are easy to imagine. For example, let $i = 0$, i.e., let $X \subseteq A^{(0)}$ be a set of rows. Select a cross in each row of X , then the collection of the columns of the selected crosses is an element of $X\psi_0$, and each element of $X\psi_0$ is obtained this way. For example, if $X = \{a_1, a_2\}$ in Table 1 then $X\psi_0$ consists of $\{b_1\}$, $\{b_1, b_2\}$, $\{b_1, b_3\}$ and $\{b_2, b_3\}$.

¹This is a context about juggling, the details are at Publications/[77] in the author's web site.

Let $\mathcal{C}_0 = \mathcal{G}$. If \mathcal{C}_n is already defined then let

$$(1) \quad \mathcal{C}_{n+1}^{(i)}(X) := \mathcal{C}_n^{(i)}(X) \cap \bigcap_{Y \in X\psi_i} \bigcup_{y \in \mathcal{C}_n^{(1-i)}(Y)} \{y\}\rho_{1-i}.$$

This defines the pair $\mathcal{C}_{n+1} = (\mathcal{C}_{n+1}^{(0)}, \mathcal{C}_{n+1}^{(1)})$.

The easiest way to digest formula (1) is to think of it pictorially. For example, let $i = 0$ and $X \subseteq A^{(0)}$, and suppose that $\mathcal{C}_n = (\mathcal{C}_n^{(0)}, \mathcal{C}_n^{(1)})$ is already well-understood. Then a row z belongs to $\mathcal{C}_{n+1}^{(0)}(X)$ if and only if $z \in \mathcal{C}_n^{(0)}(X)$ and, in addition, for each set $Y \in X\psi_0$ of columns there is a column y in $\mathcal{C}_n^{(1)}(Y)$ such that y intersects the row z at a cross. (Notice that $X\psi_0$ has already been explained pictorially, $\mathcal{C}_n^{(0)}(X)$ and $\mathcal{C}_n^{(1)}(Y)$ are already well-known by assumption, and y need not be unique and it depends on Y .)

Finally, let

$$\mathcal{C} = (\mathcal{C}^{(0)}, \mathcal{C}^{(1)}) := \left(\bigcap_{n=0}^{\infty} \mathcal{C}_n^{(0)}, \bigcap_{n=0}^{\infty} \mathcal{C}_n^{(1)} \right),$$

which means that, for all $X \in P(A^{(i)})$,

$$\mathcal{C}^{(i)}(X) = \bigcap_{n=0}^{\infty} \mathcal{C}_n^{(i)}(X).$$

Although the above definitions look neither friendly nor natural at the first sight, they had a proper application in [3]; proper means that \mathcal{C} was heavily used when proving a theorem which has nothing to do with the notion of \mathcal{C} . Notice also that it was routine to prove in [3] that we have indeed defined pairs of closure operators.

Lemma 1. (cf. [3]) $\mathcal{C} = \mathcal{C}(A^{(0)}, A^{(1)}, \rho)$ and $\mathcal{C}_n = \mathcal{C}_n(A^{(0)}, A^{(1)}, \rho)$ ($n = 0, 1, \dots$) are pairs of closure operators. Further, $\mathcal{G} = \mathcal{C}_0 \geq \mathcal{C}_1 \geq \mathcal{C}_2 \geq \dots \geq \mathcal{C}$.

It is well-known and goes back to Galois that, for each context $(A^{(0)}, A^{(1)}, \rho)$, the complete lattices $(\{X \in P(A^{(0)}) : \mathcal{G}^{(0)}(X) = X\}, \subseteq)$ and $(\{X \in P(A^{(1)}) : \mathcal{G}^{(1)}(X) = X\}, \subseteq)$ are dually isomorphic. The analogous statement is far from being true for \mathcal{C} ; indeed, in case of the context given by Table 1, $|\{X \in P(A^{(0)}) : \mathcal{C}^{(0)}(X) = X\}| = 12$ while $|\{X \in P(A^{(1)}) : \mathcal{C}^{(1)}(X) = X\}| = 10$.

From now on we always assume that $(A^{(0)}, A^{(1)}, \rho)$ is finite. Then there are only finitely many pairs of operators, whence there is a smallest n with $\mathcal{C} = \mathcal{C}_n = \mathcal{C}_{n+1} = \mathcal{C}_{n+2} = \dots$. This raises the natural question how large this n can be. To shed more light on \mathcal{C} , we answer this question below.

Let us say that $(A^{(0)}, A^{(1)}, \rho)$ is a decomposable context if there are nonempty sets $B^{(i)}$ and $C^{(i)}$ with $B^{(i)} \cup C^{(i)} = A^{(i)}$ and $B^{(i)} \cap C^{(i)} = \emptyset$ such that

$$\rho = (\rho \cap (B^{(0)} \times B^{(1)})) \cup (\rho \cap (C^{(0)} \times C^{(1)})).$$

Otherwise $(A^{(0)}, A^{(1)}, \rho)$ is called an *indecomposable context*. We say that it is a *uniform context* if $|\{x\}\rho_i| = |\{y\}\rho_i|$ for all $x, y \in A^{(i)}$. For example, all finite block designs (P, B, I) and, in particular, all finite projective spaces (P, L, I) are uniform contexts. If $|\{x\}\rho_i| = |\{y\}\rho_i| = 2$ for all $x, y \in A^{(i)}$ then we speak of a *2-uniform context*. In the terminology of context tables, a context is 2-uniform iff there are exactly two crosses in each row and in each column.

Now, as an anonymous referee of [4] suggested, it is routine to extract from [3] that even if we restrict ourselves to finite indecomposable 2-uniform contexts, arbitrarily large numbers n with $\mathcal{G} = \mathcal{C}_0 > \mathcal{C}_1 > \mathcal{C}_2 > \mathcal{C}_3 \cdots \mathcal{C}_n$ occur. We close this section by recalling an open problem about \mathcal{C} ; for motivation and a possible application cf. [3].

Problem 1. *Is it true that for each indecomposable uniform context $(A^{(0)}, A^{(1)}, \rho)$ with $|A^{(0)}| \geq 3$ and $|A^{(1)}| \geq 3$ there exists an $i \in \{0, 1\}$ and there are $x, y, z \in A^{(i)}$ such that*

$$\mathcal{C}^{(i)}(\{x, y\}) \cap \mathcal{C}^{(i)}(\{y, z\}) \cap \mathcal{C}^{(i)}(\{z, x\}) = \emptyset?$$

2. MOTIVATIONS

Now we give our motivations, and this will explain why “association rule” occurs in the title of the paper. Closure operators have been playing an important role in the theory of relational databases and knowledge systems for a long time, cf. e.g., Caspard and Monjardet [2] for a survey. Nowadays most investigations of this kind belong to formal concept analysis, cf. Ganter and Wille [7] for an extensive survey. The theory of mining association rules goes back to Agrawal, Imielinski and Swami [1]; Lakhal and Stumme [8] gives a good account on the present status of this field.

For a data miner, the context is a *huge* binary database, and mining association rules is a popular knowledge discovery technique for warehouse basket analysis. In this case $A^{(0)}$ is the set of costumers’ baskets, $A^{(1)}$ is the set of items sold in the warehouse, and the task is to figure out which items are frequently bought together. This information, expressed by so-called “association rules”, can help the warehouse in developing appropriate marketing strategies. For example,

$$\{\text{cereal, coffee}\} \rightarrow \{\text{milk}\}$$

is an association rule (in many real warehouses), and this association rule says that, with a given probability p , costumers buying cereal and coffee also buy milk. When $\text{milk} \in \mathcal{G}^{(1)}\{\text{cereal, coffee}\}$ then this probability is 1 and we speak about a *strong* association rule.

However, the importance of looking for the hidden regularities and rules is not restricted only to *huge* databases. The success of formal concept analysis, cf. Ganter and Wille [7], or Mendeleyev’s classical periodic system of chemical elements show that exploring some rules in *small* databases may also lead to important results. From this aspect, the present paper offers \mathcal{C} , a mathematical tool, to formulate some regularities in abstract contexts. Since $\mathcal{C} \leq \mathcal{G}$, the “association rules” corresponding to \mathcal{C} are *stronger* than the previously mentioned ones. It would be nice to find some concrete contexts outside mathematics, say in natural or social sciences, where \mathcal{C} has some real applications and gives new insights that \mathcal{G} does not. This is much beyond the scope of the present paper but there is a hope, for finding associations is an integral part of any creative activity.

Now we use the context given by Table 1 to develop our ideas further. Let $X = \{a_1, a_2\} \subseteq A^{(0)} = \{a_1, \dots, a_5\}$. Then $\{a_1, \dots, a_4\} \times \{b_1\}$ is the only relevant maximal full rectangle to compute $\mathcal{G}^{(0)}(X) = \{a_1, \dots, a_4\}$. Since $Y = \{b_2, b_3\} \in X\psi_0$ but there is no $y \in \mathcal{G}^{(1)}(Y) = \{b_2, b_3, b_4\}$ with $a_4 \in \{y\}\rho_1$, formula (1) gives $a_4 \notin \mathcal{C}_1^{(0)}(X)$. After the trivial and therefore omitted details we can easily see that $\mathcal{C} = \mathcal{C}_1$ and $\mathcal{C}^{(0)}(X) = \{a_1, a_2, a_3\}$.

Suppose our whole knowledge is decoded in the context and we have to associate an element with X . Usually we want an element outside X , and we look for something similar, i.e., we want an element which shares the common attributes of the elements of X . So the first answer is that we should associate some element of $\mathcal{G}^{(0)}(X) \setminus X = \{a_3, a_4\}$. This way we obtain more than one element, but we may want to choose only a single one. For example, which of a_3 and a_4 should a scientist choose if the context represents something in his research field and choosing both is not permitted²? The unique element a_3 of $\mathcal{C}^{(0)}(X) \setminus X$? The other element, a_4 ? We cannot answer to this question of *decision making* in full generality.

The main motivation to investigate \mathcal{C} is that [3], where \mathcal{C} has a proper application, witnesses that \mathcal{C} is useful in algebra.

We have seen that there is chance that \mathcal{C} gives some insights and tools that \mathcal{G} , successfully used various fields, does not. But these are just hopes at present, and as a very first step to justify these expectations the paper gives some partial answers to the question how often \mathcal{C} is different from \mathcal{G} .

3. FROM A FIXED POINT THEOREM TO A PROGRAM

Given a context $(A^{(0)}, A^{(1)}, \rho)$, let $\mathbf{H} = \mathbf{H}(A^{(0)}, A^{(1)}, \rho)$ be the set of all pairs of extensive operators defined in the previous section. Similarly, the set of all pairs of closure operators will be denoted by $\mathbf{T} = \mathbf{T}(A^{(0)}, A^{(1)}, \rho)$. Then $\mathbf{H} = (\mathbf{H}, \leq)$ and $\mathbf{T} = (\mathbf{T}, \leq)$ are posets, \mathbf{T} is a sub-poset of \mathbf{H} , and $\mathcal{G}, \mathcal{C} \in \mathbf{T} \subseteq \mathbf{H}$. Motivated by formula (1), we define a mapping $f : \mathbf{H} \rightarrow \mathbf{H}$, $\mathcal{D} = (\mathcal{D}^{(0)}, \mathcal{D}^{(1)}) \mapsto \mathcal{E} = (\mathcal{E}^{(0)}, \mathcal{E}^{(1)})$ by

$$(2) \quad \mathcal{E}^{(i)}(X) := \mathcal{D}^{(i)}(X) \cap \bigcap_{Y \in X\psi_i} \bigcup_{y \in \mathcal{D}^{(1-i)}(Y)} \{y\} \rho_{1-i}.$$

Clearly, $\mathcal{E} = f(\mathcal{D}) \in \mathbf{H}$, f is a monotone mapping, and $f(\mathcal{C}_{n+1}) = \mathcal{C}_n$ for all n . Since $f(\mathcal{D}) \leq \mathcal{D}$ for all $\mathcal{D} \in \mathbf{H}$, we will call f a *contraction map*. The following proposition is mentioned to shed more light on the topic only, and will not be used in the sequel.

Proposition 1. *Given a finite context $(A^{(0)}, A^{(1)}, \rho)$, $\mathbf{T} = (\mathbf{T}, \leq)$, the set of pairs of closure operators over $(A^{(0)}, A^{(1)}, \rho)$, is an (upper) semimodular coatomistic meet-semidistributive lattice, and \mathbf{T} is closed with respect to the contraction map f .*

Proof. Let L_i be the poset of closure operators over $A^{(i)}$, $i = 0, 1$. Then, according to Corollaries 30 and 58 in Caspard and Monjardet [2], L_i is a lattice that has some nice properties, including those listed in the proposition. Notice that [2] attributes some of these properties to others, including Demetrovics, Libkin and Muchnik [5], Duquenne [6] and Ore [10]. Since \mathbf{T} is the direct product of L_0 and L_1 and the properties we consider are clearly preserved by finite direct products, the first part of the statement is shown. The statement about the contraction map is included, modulo notational changes, in the proof of Lemma 1 in [3]. \square

²For the mentioned concrete meaning of Table 1 the beginner may ask which one of a_3 and a_4 is easier to learn after a_1 and a_2 .

If $\mathcal{D} \in \mathbf{H}$ and $f(\mathcal{D}) = \mathcal{D}$ then \mathcal{D} is called a fixed point of f . As usual, for $\mathcal{D} \in \mathbf{H}$ the set $\{\mathcal{E} \in \mathbf{H} : \mathcal{E} \leq \mathcal{D}\}$ will be denoted by $(\mathcal{D}]$. Since f is a monotone contraction map, $(\mathcal{D}]$ is always closed with respect to f . Remembering that $(A^{(0)}, A^{(1)}, \rho)$ is assumed to be finite, we have the following theorem.

Theorem 1. \mathcal{C} is the largest fixed point of f in $(\mathcal{G}]$.

Proof. Since the context is finite, there is a k with $\mathcal{C}_k = \mathcal{C}_{k+1} = \mathcal{C}$. Hence $f(\mathcal{C}) = f(\mathcal{C}_k) = \mathcal{C}_{k+1} = \mathcal{C}$, so \mathcal{C} is a fixed point of f . Clearly, $\mathcal{C} \in (\mathcal{G}]$.

Now let $\mathcal{D} \in (\mathcal{G}]$ be an arbitrary fixed point. Then, using that f is monotone, $\mathcal{D} = f(\mathcal{D}) \leq f(\mathcal{G}) = \mathcal{C}_1$. So $\mathcal{D} = f(\mathcal{D}) \leq f(\mathcal{C}_1) = \mathcal{C}_2$, etc. Thus $\mathcal{D} \leq \mathcal{C}_k = \mathcal{C}$. \square

Even if the above theorem is a simple statement, it is useful from algorithmic point of view. The speed of the obvious algorithm for computing \mathcal{C} depends on how fast the sequence $\mathcal{C}_0 = \mathcal{G}, \mathcal{C}_1, \mathcal{C}_2, \dots$ decreases. If we follow what formula (1) says then we obtain \mathcal{C}_{n+1} from \mathcal{C}_n in two steps. In the first step we compute, say, $\mathcal{C}_{n+1}^{(0)}$ from $(\mathcal{C}_n^{(0)}, \mathcal{C}_n^{(1)})$, and then in the second step we compute $\mathcal{C}_{n+1}^{(1)}$ from $(\mathcal{C}_n^{(0)}, \mathcal{C}_n^{(1)})$. However, we could obtain a more rapidly decreasing sequence if we performed the second step from $(\mathcal{C}_{n+1}^{(0)}, \mathcal{C}_n^{(1)})$ instead of $(\mathcal{C}_n^{(0)}, \mathcal{C}_n^{(1)})$. (This would also mean less memory usage, which would save some additional time, too.) We will refer to this strategy as the *modified algorithm*.

Remark 1. The modified algorithm computes \mathcal{C} , and it is at least as fast as the straightforward algorithm suggested by formula (1). (In fact, it is usually faster.)

Indeed, for $i \in \{0, 1\}$ we define a mapping $f_i : \mathbf{H} \rightarrow \mathbf{H}$, $(\mathcal{D}^{(0)}, \mathcal{D}^{(1)}) \mapsto (\mathcal{E}^{(0)}, \mathcal{E}^{(1)})$ such that $\mathcal{E}^{(i)}$ is defined as in formula (2) and $\mathcal{E}^{(1-i)} = \mathcal{D}^{(1-i)}$. (It follows easily from Proposition 1 that \mathbf{T} is closed with respect to the contraction maps f_i , $i \in \{0, 1\}$, but we do not need this fact in the proof.) The modified algorithm produces the sequence

$$f_0(\mathcal{G}), f_1(f_0(\mathcal{G})), f_0(f_1(f_0(\mathcal{G}))), f_1(f_0(f_1(f_0(\mathcal{G})))), \dots$$

Computing two new members of this sequence needs a slightly less computer work than computing one new member of the sequence $\mathcal{C}_0 = \mathcal{G}, \mathcal{C}_1, \mathcal{C}_2, \dots$. Hence all we have to show is that, for every n , the $2n$ -th member of the first sequence is above \mathcal{C} and below \mathcal{C}_n . But this follows via a trivial induction, since $f(\mathcal{D}) \leq f_i(\mathcal{D}) \leq \mathcal{D}$ and $f(f(\mathcal{D})) \leq f_1(f_0(\mathcal{D})) \leq f(\mathcal{D})$ hold for all $\mathcal{D} \in \mathbf{H}$.

It is clear from the proof of Theorem 1 and the argument following Remark 1 that instead of the poset \mathbf{H} we could have worked only with the lattice \mathbf{T} . However, the advantage of \mathbf{H} is not only to make Theorem 1 stronger. In a practical calculation, like computing $\mathcal{C}^{(i)}(X)$ just for a single X , it gives a better theoretical background: we can reduce the $\mathcal{G}^{(i)}(Y_j)$'s for certain (not necessarily distinct) subsets Y_j of $A^{(0)}$ and $A^{(1)}$ according to (2) in an arbitrary order, and we do not have to care if the actual pair of operators is a pair of closure operators, the process converges to \mathcal{C} .

The algorithm given by Remark 1 was developed into a modest computer program which computes \mathcal{C} . The program is available at the author's home page. (Although the source code is in Borland's old Turbo Pascal 7.0 for MS DOS, the executable version runs in today's Windows environment as well.) When $|A^{(0)}|$ or $|A^{(1)}|$ is large then it is not possible to determine \mathcal{C} in the practice, at least not with *this* program, for we would need $2^{|A^{(0)}|} + 2^{|A^{(1)}|}$ steps even to store \mathcal{C} . However,

as it is clear from formula (1), we can determine $\mathcal{C}^{(i)}(X)$ for all X with $|X| \leq m$ and all $i \in \{0, 1\}$ without determining the whole \mathcal{C} , and this is much faster when m is not too large. The program allows $m \in \{2, \dots, 9\}$ when $|A^{(0)}|, |A^{(1)}| \leq 14$, and it allows only $m = 2$ when $|A^{(0)}|, |A^{(1)}| \leq 48$. However, the running time even for a single context with $m = 9$ and $|A^{(0)}| = |A^{(1)}| = 14$ is usually too long to wait for. The program can generate and test many random contexts. The experimental results obtained by the program are reported in the following section.

4. EXPERIMENTAL RESULTS OBTAINED BY THE PROGRAM

The computational results will be given by tables, which are followed by the explanations of their concise notations. Even if these results are not exact mathematical theorems, they shed some light on the goal of the paper, and they may induce exact numerical bounds or asymptotical statements in the future. Notice that when only some specific relations (e.g., partial orders) are considered then the frequency of $\mathcal{C} \neq \mathcal{G}$ can be quite different from what we can learn from the present section, cf. [4].

size	4	5	6	8	10	12	14	20	30	40	48
$ \{\text{tests}\} $	1000	1000	1000	100	100	100	100	100	100	100	100
$(-, -, -)$	549	757	889	98	100						
$([0, 4], -, -)$	549	757	889	98	100	100	100				
$([0, 2], -, -)$	535	707	844	97	100	100	100				
$(\{2\}, -, -)$	282	517	736	94	100	100	100	100	100	100	100
$(\{2\}, \subset, -)$	235	484	721	94	100	100	99	86	18	4	1
$([2, \infty), \subset, \neq)$	0	134	491	96	100						
$([2, 4], \subset, \neq)$	0	134	491	96	100	100	99				
$(\{2\}, \subset, \neq)$	0	134	470	92	100	100	99	86	18	4	1

Table 2

In Table 2, “size” denotes $|A^{(0)}| = |A^{(1)}|$, i.e., only “square” contexts have been tested. The number of contexts tested with the given size is denoted by $|\{\text{tests}\}|$. For a given context, $\mathcal{C} \neq \mathcal{G}$ can be due to some more or less trivial reason like $\mathcal{C}^{(i)}(\emptyset) \neq \mathcal{G}^{(i)}(\emptyset)$. Therefore the program counted those contexts for which there is an $i \in \{0, 1\}$ and an $X \in P(A^{(i)})$ such that $\mathcal{C}^{(i)}(X) \neq \mathcal{G}^{(i)}(X)$ and X satisfies some further conditions. Each of these further conditions is denoted by a vector (α, β, γ) . Here α is missing or it is a set of integers, like $[2, 4] = \{2, 3, 4\}$. If α is a set of integers then $|X|$ has to belong to α . If β not missing then it is the “ \subset ” sign and X has to satisfy $X \subset \mathcal{C}^{(i)}(X)$. If γ is not missing then it is the “ \neq ” sign and X has to satisfy $\mathcal{G}^{(i)}(X) \neq A^{(i)}$. For example, the row $(-, -, -)$ gives the number of contexts with $\mathcal{C} \neq \mathcal{G}$, and the entry 484 means that among 1000 random contexts there are 484 contexts containing a 2-element subset X with $\mathcal{C}^{(i)}(X) \neq \mathcal{G}^{(i)}(X)$ and $X \subset \mathcal{C}^{(i)}(X)$.

It is important to emphasize that, for each column, the program produced the given number of random contexts first, and counted those context that have the desired property only afterwards. In other words, different entries in the same column refer to the same set of random contexts. Due to the limited power of the program some entries are missing, but some obvious relations among the numbers in the same column give lower bounds for the missing entries.

We may also ask the question that if we take a random context table of size $n \times n$ and choose an $i \in \{0, 1\}$ and a subset X of $A^{(i)}$ randomly then what is the chance that $(-, -, -)$: $\mathcal{C}(X) \neq \mathcal{G}(X)$, or $([2, \infty), \subset, \neq)$: $2 \leq |X|$ and $X \subset \mathcal{C}^{(i)}(X) \neq \mathcal{G}^{(i)}(X) \neq A^{(i)}$. The experimental results for some values of n are reported in Table 3.

size	3	4	5	6	7	8	9
$ \{\text{tests}\} $	1000	1000	1000	1000	1000	1000	1000
$(-, -, -)$	17	39	82	185	306	402	571
$(\{2\}, \subset, \neq)$	0	0	0	12	35	54	86

Table 3

Table 2 gives the strong belief³ that a “medium sized” square context gives an “essentially new” \mathcal{C} with high probability. Here “essentially new” means that the condition $(\{2\}, \subset, \neq)$ holds for some X . However, this probability decreases rapidly when the size of the context grows.

Let us call a random context a p -random context, $0 < p < 1$, if we put a cross to each entry with probability p , independently from other entries. So far we have considered 0.5-random contexts. However, we may get different results with other values of p . For example, we tested 100 p -random 40×40 -sized contexts with different values of p , and counted the essentially new contexts among them (in the sense of $(\{2\}, \subset, \neq)$). The result is given by Table 4.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$(\{2\}, \subset, \neq)$	100	68	14	5	2	3	23	64	77

Table 4

Finally, we tested some contexts from the real life: essentially all those contexts from Ganter and Wille [7] which are given by simple context tables (with \times being the only entry) and whose size fits into the program. (Sometimes the context was given by a multi-valued table and we had to reduce it.) Each column represents a context, and the column label tells us which context of [7] is considered. “Yes” for a context means that \mathcal{C} is distinct from \mathcal{G} and the condition (α, β, γ) given in the row label is satisfied.

	1.1	1.5a	1.5b	1.13	1.16	1.21	1.23	1.24	2.4	2.13	2.15
$ A^{(0)} $	8	8	5	5	14	6	6	8	7	12	14
$ A^{(1)} $	9	5	4	25	16	12	8	8	7	9	9
$(\{2\}, \subset, \neq)$	no	no	no	yes	no	no	yes	no	yes	no	yes
$([0, 6], -, -)$	yes	no	no	yes		yes	yes	no	yes	no	yes

Table 5

REFERENCES

- [1] Agarwal, R., Imielinski, T., and Swami, A. Mining association rules between sets of items in large databases. in *Proceedings of the 1993 ACM SIGMOD international conference on Management of Data (SIGMOD'93)*, pages 207-216, ACM Press, May 1993.

³Theoretically there could be some unknown hidden connection between \mathcal{C} and the built-in random number generator and this could mislead us, but the chance of this is minimal.

- [2] Caspard, N., and Monjardet, B. The lattice of closure systems, closure operators, and implicational systems on a finite set: a survey. *Discrete Applied Mathematics* 127(2):241–269, 2003.
- [3] Czédli, G. 2-uniform congruences in majority algebras and a closure operator. *Algebra Universalis* 57(1):63–73, 2007.
- [4] Czédli, G. A stronger association rules in lattices and posets, *Order*, submitted.
- [5] Demetrovics, J., Libkin, L.O., and Muchnik, I.B. Functional dependencies in relational databases: a lattice point of view. *Discrete Applied Mathematics* 40(2):155–185, 1992.
- [6] Duquenne, V. The lattice of all 1-meet-subsemilattices of a finite lattice. 1987 (unpublished notes).
- [7] Ganter B., and R. Wille, R. *Formal Concept Analysis: Mathematical foundations*. Springer-Verlag, Berlin — Heidelberg, 1999.
- [8] Lakhal L., and Stumme, G. Efficient mining of association rules based on formal concept analysis. in *Formal Concept Analysis*, Lecture Notes in Computer Science, vol. 3626/2005 (editors: B. Ganter, G. Stumme and R. Wille), pp. 180–195, Springer-Verlag, Berlin — Heidelberg, 2005.
- [9] Polster, B. *The mathematics of juggling* Springer-Verlag, New York, 2003.
- [10] Ore, O. Combinations of closure relations. *Ann. Math.* 14:514–533, 1943.
- [11] Wille, R. Restructuring lattice theory: an approach based on hierarchies of concepts. *Ordered sets* (Banff, Alta., 1981) *NATO Adv. Study Inst. Ser. C: Math. Phys. Sci.* **83**, pages 445–470, Reidel, Dordrecht-Boston, Mass., 1982.

UNIVERSITY OF SZEGED, BOLYAI INSTITUTE, SZEGED, ARADI VÉRTANÚK TERE 1, HUNGARY
6720

E-mail address: czedli@math.u-szeged.hu

URL: <http://www.math.u-szeged.hu/~czedli/>