

Subdirect Decomposition of Contexts into Subdirectly Irreducible Factors

Jean-François Viaud¹, Karell Bertet¹, Christophe Demko¹, Rokia Missaoui²

¹ Laboratory L3i, University of La Rochelle, France
{jviaud, kbertet, cdemko}@univ-lr.fr

² University of Québec in Outaouais, Canada
rokoa.missaoui@uqo.ca

Abstract. The size of a concept lattice may increase exponentially with the size of the context. When the number of nodes is too large, it becomes very difficult to generate and study such a concept lattice. A way to avoid this problem is to break down the lattice into small parts. In the subdirect decomposition, the small parts are factor lattices which are meaningful in the Formal Concept Analysis (FCA) setting.

In this paper a walkthrough from a finite reduced context to its subdirect decomposition into subdirectly irreducible subcontexts and factors is given. The decomposition can be reached using three different points of view, namely factor lattices, arrow relations and compatible subcontexts. The approach is mainly algebraic since it is based on abstract lattice theory, except for the last point which is inherited from FCA. We also propose a polynomial algorithm to generate the decomposition of an initial context into subcontexts. Such a procedure can be extended to conduct an interactive exploration and mining of large contexts, including the generation of few concepts and their neighborhood.

Keywords: concept lattice, congruence relation, factor lattice, arrow relation, arrow closed subcontext, compatible subcontext

1 Introduction

During the last decade, the computation capabilities have promoted Formal Concept Analysis (FCA) with new methods based on concept lattices. Though they are exponential in space/time in worst case, concept lattices of a reasonable size enable an intuitive representation of data organized by a context that links objects to attributes through a binary relation. Methods based on concept lattices have been developed in various domains such as knowledge discovery and representation, database management and information retrieval where some relevant concepts, *i.e.* possible correspondences between objects and attributes are considered either as classifiers, clusters or representative object/attribute subsets.

With the increasing size of data, a set of methods have been proposed in order to either generate a subset (rather than the whole set) of concepts and

their neighborhood (e.g. successors and predecessors) in an online and interactive way [8, 20] or better display lattices using nested line diagrams [13]. Such approaches become inefficient when contexts are huge. However, the main idea of lattice/context decomposition into smaller ones is still relevant when the classification property of the initial lattice is maintained. Many lattice decompositions have been defined and studied, both from algebraic [6, 16] and FCA points of view [13, 12]. We can cite Unique Factorisation Theorem [16], matrix decomposition [2], Atlas decomposition [13], subtensorial decomposition [13], doubling convex construction [5, 17, 14, 3] and subdirect decomposition. The latter has been widely studied many years ago, in the field of universal algebra [6, 11], and even in FCA [21–24, 12]. To the best of our knowledge, there is no new development or novel algorithms for subdirect decomposition of contexts.

In this paper we investigate the subdirect decomposition of a concept lattice as a first step towards an interactive exploration and mining of large contexts. The subdirect decomposition of a lattice L into factor lattices $(L_i)_{i \in \{1, \dots, n\}}$, denoted by $L \hookrightarrow L_1 \times \dots \times L_n$, is defined by two properties (see important results in [13]): (i) L is a sublattice of the direct product $L_1 \times \dots \times L_n$, and (ii) each projection of L onto a factor is surjective. The first property establishes that each factor lattice is the concept lattice of an arrow-closed subcontext, *i.e.* closed according to the arrow relation between objects and attributes. This means that the decomposition can be obtained by computing specific subcontexts. The second property states that there is an equivalence between arrow-closed subcontexts and congruence relations of L , *i.e.*, an equivalence relation whose equivalence classes form a lattice with elements closed by the meet and join operations. This means that the concepts of L can be retrieved from the factor lattices, and the classification property of L is maintained since each equivalence relation forms a partition of the elements. The last result establishes an equivalence between arrow-closed subcontexts and compatible subcontexts, *i.e.* subcontexts such that each concept corresponds to a concept of the initial lattice. This result gives a way to compute the morphism from L into the direct product, and thus to retrieve the concepts of L from the factor lattices. In this paper, we deduce from these results strong links between the following notions that have not been used yet together as far as we know:

- Factors of a subdirect decomposition,
- Congruence relations,
- Arrow-closed subcontexts and
- Compatible subcontexts.

As suggested in [13], the contexts of the factors of a particular subdirect decomposition, namely the subdirectly irreducible subcontexts, can be obtained by a polynomial processing of each row/object of the initial context. Therefore, the subdirect decomposition of a lattice can be extended to a subdirect decomposition of its reduced context into subdirect and irreducible subcontexts.

In this paper, we propose a subdirect and polynomial decomposition of a context into subcontexts by extending the subdirect decomposition of a lattice

into factor lattices. This decomposition leads to data storage saving of large contexts. Indeed, the generation of the whole set of factor lattices can be avoided by providing an interactive generation of a few (but not all) concepts and their neighborhood from large contexts. Moreover, a focus on a specific factor lattice can be proposed to the user by generating, partially or entirely, the concept lattice and/or a basis of implications.

There are at least two reasons for studying this case of pattern management. The first one comes from the fact that users tend to be overwhelmed by the knowledge extracted from data, even when the input is relatively small. The second reason is that the community of FCA has made progress in lattice construction and exploration, and hence existing solutions can be adapted and enriched to only target useful patterns (*i.e.* pieces of knowledge).

This paper is organized as follows. Section 2 introduces the subdirect decomposition and the three different points of view, namely factor lattices, arrow relations and compatible subcontexts. Section 3 contains the main contribution of this paper about the subdirect decomposition and the proposed algorithms. A preliminary empirical study is presented in Section 4 while Section 5 presents future work.

2 Structural framework

Throughout this paper all sets (and thus lattices) are considered to be finite.

2.1 Lattices and Formal Concept Analysis

Algebraic lattice Let us first recall that a *lattice* (L, \leq) is an ordered set in which every pair (x, y) of elements has a least upper bound, called *join* $x \vee y$, and a greatest lower bound, called *meet* $x \wedge y$. As we are only considering finite structures, every subset $A \subset L$ has a join and meet (e. g. finite lattices are complete).

Concept or Galois Lattice A (formal) *context* (O, A, R) is defined by a set O of objects, a set A of attributes, and a binary relation $R \subset O \times A$, between O and A . Two operators are derived:

- for each subset $X \subset O$, we define $X' = \{m \in A, j R m \forall j \in X\}$ and dually,
- for each subset $Y \subset A$, we define $Y' = \{j \in O, j R m \forall m \in Y\}$.

A (formal) *concept* represents a maximal objects-attributes correspondence by a pair (X, Y) such that $X' = Y$ and $Y' = X$. The sets X and Y are respectively called *extent* and *intent* of the concept. The set of concepts derived from a context is ordered as follows:

$$(X_1, Y_1) \leq (X_2, Y_2) \iff X_1 \subseteq X_2 \iff Y_2 \subseteq Y_1$$

The whole set of formal concepts together with this order relation form a complete lattice, called the *concept lattice* of the context (O, A, R) .

Different formal contexts can provide isomorphic concept lattices, and there exists a unique one, named the *reduced context*, defined by the two sets O and A of the smallest size.

This particular context is introduced by means of special concepts or elements of the lattice L , namely irreducible elements.

An element $j \in L$ is *join-irreducible* if it is not a least upper bound of a subset not containing it. The set of join irreducible elements is noted J_L . *Meet-irreducible* elements are defined dually and their set is M_L . As a direct consequence, an element $j \in L$ is join-irreducible if and only if it has only one immediate predecessor denoted j^- . Dually, an element $m \in L$ is meet-irreducible if and only if it has only one immediate successor denoted m^+ .

In Figure 1, join-irreducible nodes are labelled with a number and meet-irreducible nodes are labelled with a letter.

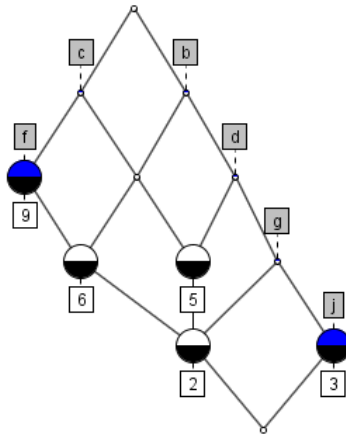


Fig. 1. A lattice with its irreducible nodes

Fundamental Bijection A fundamental result [1] establishes that any lattice (L, \leq) is isomorphic to the concept lattice of the context (J_L, M_L, \leq) , where J_L and M_L are the join and meet irreducible concepts of L , respectively. Moreover, this context is a reduced one.

As a direct consequence, there is a bijection between lattices and reduced contexts where objects of the context are associated with join-irreducible concepts of the lattice, and attributes are associated with meet-irreducible concepts.

Figure 2 shows the reduced context of the lattice in Figure 1.

	b	c	d	f	g	j
2	x	x	x	x	x	
3	x		x		x	x
5	x	x	x			
6	x	x		x		
9		x		x		

Fig. 2. The reduced context of the lattice in Figure 1

2.2 Compatible and Arrow-closed Subcontexts

This section is dedicated to the equivalence between compatible and arrow-closed subcontexts.

Compatible subcontexts A *subcontext* of a formal context (O, A, R) is a triple $(J, M, R \cap J \times M)$ such that $J \subset O$ and $M \subset A$. A subcontext $(J, M, R \cap J \times M)$ of (O, A, R) is *compatible* if for each concept (H, N) of (O, A, R) , $(J \cap H, M \cap N)$ is a concept of $(J, M, R \cap J \times M)$.

Arrow relations The arrow-closed subcontexts involved in the equivalence are based on the arrow relations between join and meet irreducible concepts of a lattice. Consider the reduced context (J_L, M_L, \leq) of a lattice (L, \leq) . Arrow relations [4, 15] form a partition of the relation $\not\leq$ (defined by not having $x \leq y$) by considering the immediate predecessor j^- of a join-irreducible j , and the unique immediate successor m^+ of a meet-irreducible m :

- $j \uparrow m$ if $j \not\leq m$, $j \leq m^+$ and $j^- \leq m$.
- $j \uparrow m$ if $j \not\leq m$, $j \leq m^+$ and $j^- \not\leq m$.
- $j \downarrow m$ if $j \not\leq m$, $j \not\leq m^+$ and $j^- \leq m$.
- $j \circ m$ if $j \not\leq m$, $j \not\leq m^+$ and $j^- \not\leq m$.

In Figure 3, the reduced context of Figure 2 is enriched with the four relations \uparrow , \downarrow , and \circ in the empty cells that both correspond to the case where $j \not\leq m$:

	b	c	d	f	g	j
2	x	x	x	x	x	\uparrow
3	x	\uparrow	x	\downarrow	x	x
5	x	x	x	\uparrow	\uparrow	\circ
6	x	x	\uparrow	x	\downarrow	\circ
9	\uparrow	x	\circ	x	\circ	\circ

Fig. 3. Arrow relation

As an illustration, let $j = 5$ and $m = f$ be join-irreducible and meet-irreducible nodes respectively (see Figure 1). Then, $j^- = 2$ and $m^+ = c$. The relation $5 \uparrow f$ holds since $5 \not\leq f$, $5 \leq c$ and $2 \leq f$.

Arrow-closed subcontext A subcontext $(J, M, R \cap J \times M)$ of a context (O, A, R) is an *arrow-closed subcontext* when the following conditions are met:

- If $j \uparrow m$ and $j \in J$ then $m \in M$
- If $j \downarrow m$ and $m \in M$ then $j \in J$

As an example, the first subcontext of Figure 4 is an arrow-closed subcontext of the reduced context of Figure 3 whereas the second one is not, due to the down-arrow $6 \downarrow g$.

	c	d	f	g
3	x		x	
5	x	x		
6	x		x	

	c	d	f	g
3	x		x	
5	x	x		

Fig. 4. Arrow-closed and non-arrow-closed subcontexts of the context in Figure 3

Equivalence theorem First let us introduce the first equivalence we need in this paper, whose proof can be found in [13]:

Theorem 1. *Let $(J, M, R \cap J \times M)$ be a subcontext of (O, A, R) . The following propositions are equivalent:*

- The subcontext $(J, M, R \cap J \times M)$ is a compatible one.
- The subcontext $(J, M, R \cap J \times M)$ is an arrow-closed one.

2.3 Congruence Relations and Factor Lattices

In this section, we introduce the equivalence between congruence relations and arrow-closed subcontexts.

Quotient An *equivalence relation* is a binary relation R over a set E which is reflexive, symmetric, and transitive. An equivalence class of $x \in E$ is:

$$x_R = \{y \in E \mid xRy\}$$

The set of equivalence classes, called the *quotient set* E/R , is:

$$E/R = \{x_R \mid x \in E\}$$

Factor lattice A congruence relation Θ on a lattice L is an equivalence relation such that:

$$x_1\Theta y_1 \text{ and } x_2\Theta y_2 \implies x_1 \wedge x_2\Theta y_1 \wedge y_2 \text{ and } x_1 \vee x_2\Theta y_1 \vee y_2$$

The quotient L/Θ verifies the following statement:

$$x_\Theta \leq y_\Theta \iff x\Theta(x \wedge y) \iff (x \vee y)\Theta y$$

With such an order, L/Θ is a lattice, called factor lattice. A standard theorem from algebra, whose proof is omitted, states that:

Theorem 2. *The projection $L \rightarrow L/\Theta$ is a lattice morphism onto.*

The second equivalence theorem We are now able to formulate the second equivalence whose proof can be found in [13]:

Theorem 3. *Given a lattice L , the set of congruence relations on L corresponds bijectively with the set of arrow-closed subcontexts of the reduced context of L .*

Congruence relations will be computed with this theorem. However, other algorithms exist [9, 10].

2.4 Subdirect decompositions

In this section, we introduce the equivalence between subdirect decompositions and sets of arrow-closed subcontexts.

Subdirect product

Definition 1. *A subdirect product is a sublattice of a direct product $L_1 \times \cdots \times L_n$ of lattices $L_i, i \in \{1, \dots, n\}$ such that each projection onto a factor is surjective. The lattices $L_i, i \in \{1, \dots, n\}$ are the factor lattices. A subdirect decomposition of a lattice L is an isomorphism between L and a subdirect product which can be denoted as:*

$$L \hookrightarrow L_1 \times \cdots \times L_n \twoheadrightarrow L_i$$

The third equivalence theorem The third and most important equivalence whose proof can be found in [13], makes a connection with sets of arrows-closed subcontexts when they cover the initial context:

Proposition 1. *Given a reduced context (O, A, R) , then the subdirect decompositions of its concept lattice L correspond bijectively to the families of arrow-closed subcontexts $(J_j, M_j, R \cap J_j \times M_j)$ with $O = \cup J_j$ and $A = \cup M_j$.*

3 Our contribution

3.1 Main Result

From the three previous equivalences found in [13], we deduce the following one:

Corollary 1. *Given a lattice L and its reduced context (O, A, R) , we have an equivalence between:*

1. *The set of arrow-closed subcontexts of (O, A, R) ,*
2. *The set of compatible subcontexts of (O, A, R) ,*
3. *The set of congruence relations of L and their factor lattices.*

Corollary 2. *Given a lattice L and its reduced context (O, A, R) , we have an equivalence between:*

1. *The families of arrow-closed subcontexts of (O, A, R) covering O and A ,*
2. *The families of compatible subcontexts of (O, A, R) covering O and A ,*
3. *The families $(\theta_i)_{i \in I}$ of congruence relations of L such that $\bigcap_{i \in I} \theta_i = \Delta$ with $x \Delta y \iff x = y$.*
4. *The set of subdirect decompositions of L and their factor lattices.*

In the following, we exploit these four notions that, to the best of our knowledge, have not been put together yet.

1. The subdirect decomposition ensures that L is a sublattice of the factor lattice product. Moreover, each projection from L to a factor lattice is surjective.
2. The congruence relations of L indicate that factor lattices correspond to their quotient lattices, and thus preserve partitions via equivalence classes.
3. The compatible subcontexts give a way to compute the morphism from L onto its factors.
4. Arrow-closed subcontexts enable the computation of the reduced context of the factor lattices.

In the following we present the generation of a particular subdirect decomposition and show a possible usage of factor lattices.

3.2 Generation of Subdirectly Irreducible Factors

In this section, we consider subdirect decompositions of a lattice L with its reduced context (O, A, R) as input. From Corollary 2, a subdirect decomposition of a lattice L can be obtained by computing a set of arrow-closed subcontexts of (O, A, R) that have to cover O and A . There are many sets of arrow-closed subcontexts and thus many subdirect decompositions. In particular, the decomposition from a lattice L into L itself is a subdirect decomposition, corresponding to the whole subcontext (O, A, R) which is clearly arrow-closed. A subdirect decomposition algorithm has been proposed in [12]. However, all congruence relations

are computed and then only pairs of relations are formed to get a decomposition. As a consequence, potentially multiple decompositions are produced with necessarily two factors.

In this article, we focus on the subdirect decomposition of a context into a possibly large number of small factors, *i.e.* factors that cannot be subdirectly decomposed. A factor lattice L is *subdirectly irreducible* when any subdirect decomposition of L leads to L itself. A nice characterization of subdirectly irreducible lattices can be found in [13]:

Proposition 2. *A lattice L is subdirectly irreducible if and only if its reduced context is one-generated.*

A context (O, A, R) is one-generated if it can be obtained by arrow-closing a context with only one $j \in J$. Thus (O, A, R) is the smallest arrow-closed subcontext containing $j \in J$.

Therefore, we deduce the following result:

Proposition 3. *Let L be a lattice. From L , we can deduce a product lattice $L_1 \times \dots \times L_n$ where each lattice L_i is:*

- the concept lattice of a one-generated subcontext,
- subdirectly irreducible,
- a factor lattice of the subdirectly irreducible decomposition.

From this result, we propose an algorithm (Algorithm 1) to compute in polynomial time the contexts of the factor lattices L_1, \dots, \dots, L_n of a subdirectly irreducible decomposition, with a reduced context (O, A, R) as input. The one-generated subcontexts for each $j \in J$ are obtained by arrow-closing (Algorithm 2). The subdirectly irreducible decomposition of L can then be obtained by computing the concept lattices of these subcontexts.

One can notice that the closure computed from join-irreducible concepts can also be calculated from meet-irreducible concepts.

Algorithm 1: Subdirect_Decomposition

Input: A context (O, A, R)

Output: List \mathcal{L} of the contexts (J_j, M_j, R_j) of the subdirectly irreducible factor lattices

```

1  $\mathcal{L} \leftarrow \emptyset;$ 
2 forall the  $j \in O$  do
3   Compute  $(J_j, M_j, R_j) = \mathbf{Arrow\_Closure}((j, \emptyset, \emptyset), (O, A, R))$ , the
   one-generated subcontext span by  $j$ ;
4   if  $\mathcal{L}$  does not contain any subcontext that covers  $(J_j, M_j, R_j)$  then
5      $\lfloor$  add  $(J_j, M_j, R_j)$  to  $\mathcal{L}$ 
6   if  $\mathcal{L}$  contains a subcontext covered by  $(J_j, M_j, R_j)$  then
7      $\lfloor$  delete it from  $\mathcal{L}$ 
8    $\lfloor$  return  $\mathcal{L};$ 

```

Algorithm 2: Arrow_Closure

Input: A subcontext $(\tilde{J}, \tilde{M}, \tilde{R})$ of a context (J, M, R)
Output: Arrow-closure of $(\tilde{J}, \tilde{M}, \tilde{R})$

- 1 $J_c = \tilde{J}; M_c = \tilde{M};$
- 2 $pred_J = 0; pred_M = 0;$
- 3 **while** $pred_M < card(M_c)$ or $pred_J < card(J_c)$ **do**
- 4 $pred_J = card(J_c);$
- 5 $pred_M = card(M_c);$
- 6 **forall the** $j \in J_c$ **do**
- 7 \lfloor add to M_c all $m \in M$ such that $j \uparrow m;$
- 8 **forall the** $m \in M_c$ **do**
- 9 \lfloor add to J_c all $j \in J$ such that $j \downarrow m;$
- 10 Return $(J_c, M_c, R \cap J_c \times M_c)$

j	Arrow_Closure			Con- tained in \mathcal{L}
	Input $(\tilde{J}, \tilde{M}, \tilde{R})$	J_c	M_c	
2	$(2, \emptyset, \emptyset)$	$\{2\}$	$\{j\}$	×
3	$(3, \emptyset, \emptyset)$	$\{3\}$	$\{c\}$	
5	$(5, \emptyset, \emptyset)$	$\{3, 5, 6\}$	$\{c, d, f, g\}$	×
6	$(6, \emptyset, \emptyset)$	$\{6\}$	$\{d\}$	
9	$(9, \emptyset, \emptyset)$	$\{9\}$	$\{b\}$	×

Fig. 5. Iterations of Algorithm 1 for the reduced context in Figure 2

Consider the reduced context in Figure 2. Each iteration of Algorithm 1 is described by Figure 5 for each value of j , the input and output of Algorithm 2, and the three one-generated subcontexts that belong to \mathcal{L} at the end of the process. Therefore we get three factor lattices (see Figure 6).

The first subcontext is the one on the left of Figure 4. The two other ones are: $(\{2\}, \{j\}, \emptyset)$ and $(\{9\}, \{b\}, \emptyset)$. The latter two subcontexts are interesting because:

- They show that the initial lattice has parts which are distributive. Indeed, these two subcontexts contain exactly one double arrow in each line and each column.
- They give us a dichotomy: any concept contains either 2 or j ; any concept contains either 9 or b
- In the reduced context, an arrow brings a deeper knowledge than a cross.

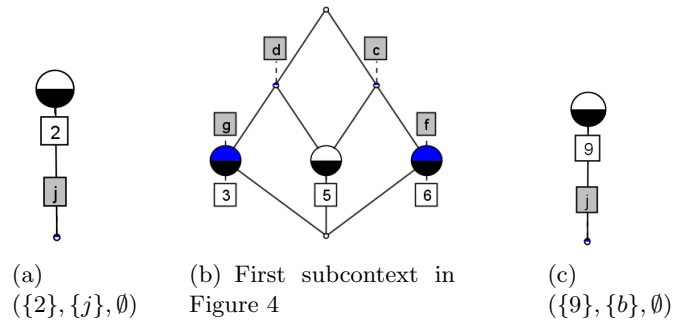


Fig. 6. The three factor lattices of the decomposition with their subcontext as caption

The context on the left hand-side of Figure 4 is tricky to understand. For the other ones, we have a simple relation $2 \uparrow j$ or $9 \uparrow b$, which means that, for instance, 2 and j are some kind of complement or converse.

Figure 7 shows a factor lattice and its corresponding congruence.

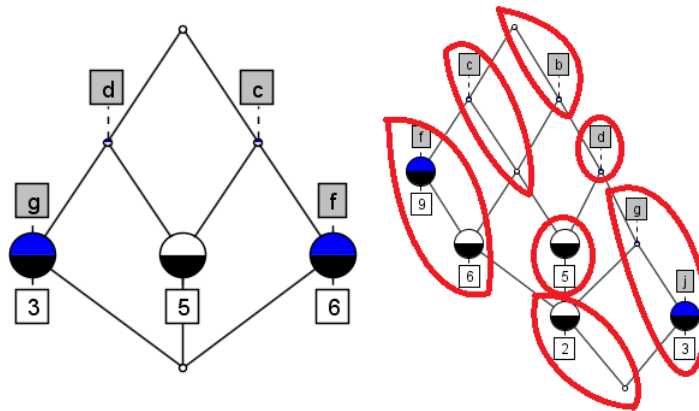


Fig. 7. Factor lattice and congruence

3.3 Onto Morphism and FCA

A subdirect decomposition of a lattice L into factor lattices L_1, \dots, L_n is relevant since there exists an onto morphism from L to the product lattice $L_1 \times \dots \times L_n$. This morphism is specified by the bijection between compatible subcontexts and congruence relations stated by Corollary 1:

Proposition 4. *Let $(J, M, R \cap J \times M)$ be a compatible subcontext, then the relation $\Theta_{J,M}$ defined by:*

$$(A_1, B_1)\Theta_{J,M}(A_2, B_2) \iff A_1 \cap J = A_2 \cap J \iff B_1 \cap M = B_2 \cap M$$

is a congruence relation, and its factor lattice is isomorphic to the concept lattice of the subcontext $(J, M, R \cap J \times M)$.

Algorithm 3 computes this morphism: each concept of L is computed as the product of concepts in each factor, and then marked in the product lattice. From Algorithm 3, we get the large tagged lattice product shown in Figure 8. Obviously, this algorithm is not intended to be used in a real application with large contexts since the product lattice is much more bigger than the original one, while the main goal of the decomposition is to get smaller lattices. We only use this algorithm in the empirical study.

Nevertheless, this morphism can be extended to develop basic FCA processing. Once the subdirectly irreducible decomposition of a reduced context (O, A, R) into the contexts C_1, \dots, C_n is computed, an interactive exploration and mining process can easily be considered by using the following basic tasks and avoiding the generation of the lattice for the whole context (O, A, R) :

- Compute the smallest concept of L that contains a given subset of objects or attributes, and identify its neighborhood
- Compute the smallest concept c_{ij} and its neighborhood in a subset of factors that contain a given collection of objects or attributes. Each factor L_i is a specific view of data.

Algorithm 3: Into_morphism

Input: Initial lattice L ;
Subcontexts (J_j, M_j, R_j) ;
Product lattice $P = L_1 \times \dots \times L_n$
Output: Product lattice P with nodes coming from L marked.

- 1 **forall the** $c = (A, B) \in L$ **do**
- 2 **forall the** (J_j, M_j, R_j) **do**
- 3 Compute $(A \cap J_j, B \cap M_j)$;
- 4 Mark, in P , the product node $\Pi_j(A \cap J_j, B \cap M_j)$;

4 Experiments

In this section, we conduct an empirical study in order to better understand the impact of the subdirect decomposition on contexts with different density levels. All the tests were done using the java-lattices library available at:

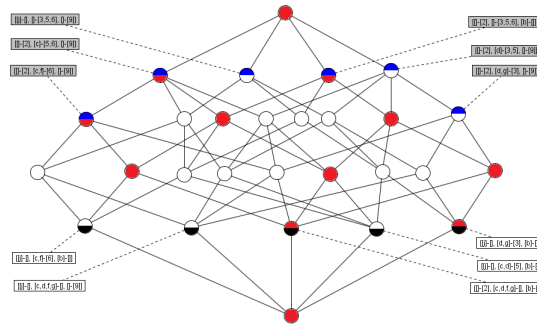


Fig. 8. Ugly tagged lattice product

<http://thegalactic.github.io/java-lattices/>

A thousand of contexts of 10 observations and 5 attributes were randomly generated. The experiments have been done three times using three density values, namely 20%, 50% and 80%. Figure 9 presents the number of generated lattices according to their size and their density. We can observe two histograms with a nice gaussian shape in the first two cases, but a strange behavior in the last case. However, we roughly observe that the higher the density is, the bigger the lattices are. Therefore, the context density will have an impact on the decomposition process.

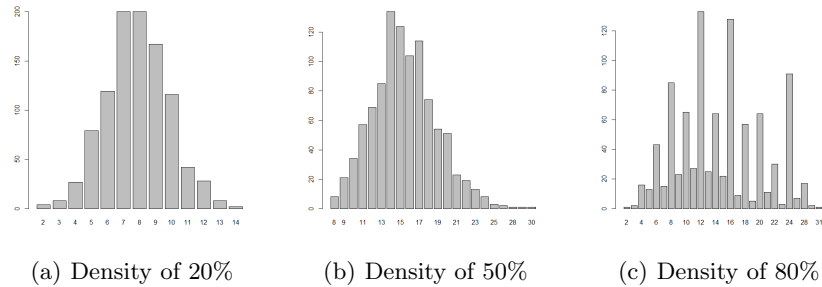


Fig. 9. Number of lattices per size.

The number of generated factors is given in Figure 10. One can observe that this number increases with the density of the initial context since the corresponding lattices have more edges. With a context density of 20%, we get 87.40% irreducible contexts. However, with a density of 80%, only 1.50% of contexts are irreducible and 70% of contexts have at least 4 factors in their decomposition. Thus, lattices are more decomposable when they come from dense contexts.

Factors	Density=20%	Density=50%	Density=80%
1	87.40%	70.50%	1.50%
2	9.70%	21.40%	9.90%
3	2.60%	6.20%	18.70%
4	0.30%	0.50%	23.40%
5		1.40%	28.50%
6			18.00%

Fig. 10. Proportion of the number of factors in the subdirect decomposition of the contexts according to their density

Let us now examine the size of factors with two different density values, namely 20% and 50%. Figure 11 gives the number of cases (initial lattices) according to the number of produced factors. Of course, we observe that smaller lattices give rise to smaller factors. We can also see that in these two density cases, the largest number is obtained for factors of size 2.

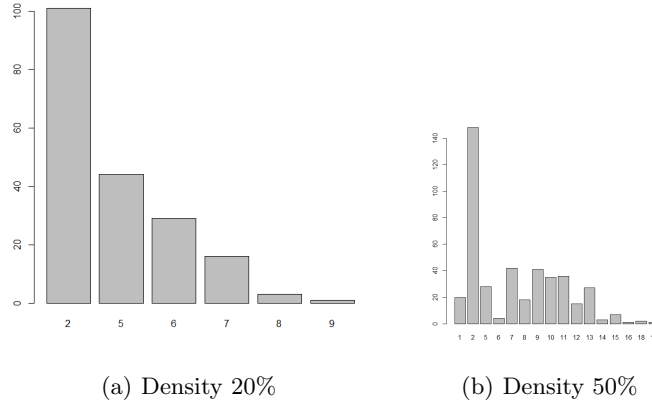


Fig. 11. Number of cases according to the number of generated factors.

The last part of the tests aims at using contexts with a large density of 80% and computing the ratio between the number of nodes (edges resp.) of the initial lattice and the number of nodes (edges resp.) of the product lattice.

We thus get Figure 12 which shows that the higher is the number of factors, the bigger is the product lattice. Consequently, the set of useless (void) nodes in the product becomes larger as the number of factors increases.

We have also conducted experiments with 40 observations and 15 attributes, and a hundred of contexts were generated using two density values: 20% and 50%. Unfortunately, all were subdirectly irreducible.

Factors	% of nodes	% of edges
1	100%	100%
2	97.90%	97.20%
3	89.20%	84.60%
4	85.00%	77.40%
5	80.60%	71.80%
6	68.00%	57.30%

Fig. 12. Proportion of untagged nodes and edges

5 Conclusion and future work

In this paper, we have presented a polynomial algorithm for the decomposition of a reduced context into subcontexts such that the concept lattice of each subcontext is a subdirectly irreducible factor. This decomposition is a direct consequence inferred from strong links between factors of a subdirect decomposition, congruence relations, arrow-closed subcontexts and compatible subcontexts established in [13].

To further investigate the subdirect decomposition, it would be interesting to conduct large-scale experiments on real world data not only to confirm/nullify the preliminary empirical tests but also to understand the semantics behind the generated irreducible contexts. In particular, attributes covered by several factors interfere in different views of data whose semantics must be interesting to understand. Moreover, it would be useful to allow the user to interactively select a few factors of the decomposition by mixing our approach with the one in [12].

From a theoretical point of view, we think that there are strong links between the implication basis in a quotient lattice and the one from the initial lattice. To the best of our knowledge, this issue has never been addressed and could have significant algorithmic impacts. However, we note that [19] tackle a similar issue in case of a vertical decomposition of a context into subcontexts.

Since the empirical study in [18] show that many real-life contexts are subdirectly irreducible, we plan to (i) identify cases in which a context is necessarily irreducible, and (ii) study, compare and combine other decompositions, in particular the Fratini congruence [7], and the reverse doubling convex construction [5, 17, 14, 3]. Finally, the construction of a lattice from its factor lattices can be done based on the optimization principles behind the relational join operation in databases.

References

1. Barbut, M., Monjardet, B. (eds.): L'ordre et la classification. Algèbre et combinatoire, tome II, Hachette (1970)
2. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences* 76(1), 3–20 (Feb 2010)

3. Bertet, K., Caspard, N.: Doubling convec sets in lattices: characterizations and recognition algorithms. Tech. Rep. TR-LACL-2002-08, LACL (Laboratory of Algorithms, Complexity and Logic), University of Paris-Est (Paris 12) (2002)
4. Crawley, P., Dilworth, R.: Algebraic theory of lattices. Prentice Hall, Englewood Cliffs (1973)
5. Day, A.: Congruence normality: The characterization of the doubling class of convex sets. *algebra universalis* 31(3), 397–406 (1994)
6. Demel, J.: Fast algorithms for finding a subdirect decomposition and interesting congruences of finite algebras. *Kybernetika (Prague)* 18(2), 121–130 (1982)
7. Duquenne, V.: Lattice drawings and morphisms. In: Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15–18, 2010. Proceedings. pp. 88–103 (2010)
8. Ferré, S.: Reconciling Expressivity and Usability in Information Access from File Systems to the Semantic Web. Ph.D. thesis, Univeristy Rennes 1 (2014)
9. Freese, R.: Computing congruence lattices of finite lattices. *Proceedings of the American Mathematical Society* 125(12), 3457–3463 (1997)
10. Freese, R.: Algorithms in finite, finitely presented and free lattices. Preprint, July 22, 159–178 (1999)
11. Freese, R.: Computing congruences efficiently. *Algebra universalis* 59(3-4), 337–343 (2008)
12. Funk, P., Lewien, A., Snelling, G.: Algorithms for concept lattice decomposition and their applications. Tech. rep., TU Braunschweig (December 1995)
13. Ganter, B., Wille, R.: Formal concept analysis - mathematical foundations. Springer (1999)
14. Geyer, W.: The generalized doubling construction and formal concept analysis. *algebra universalis* 32(3), 341–367 (1994)
15. Grätzer, G.: General lattice theory. Birkhäuser-Verlag, Basel (1978)
16. Mihók, P., Semanišin, G.: Unique factorization theorem and formal concept analysis. In: Yahia, S., Nguifo, E., Belohlavek, R. (eds.) *Concept Lattices and Their Applications*, Lecture Notes in Computer Science, vol. 4923, pp. 232–239. Springer Berlin Heidelberg (2008)
17. Nation, J.: Alan day’s doubling construction. *algebra universalis* 34(1), 24–34 (1995)
18. Snelling, G.: Concept lattices in software analysis. In: Formal Concept Analysis, Foundations and Applications. pp. 272–287 (2005)
19. Valtchev, P., Duquenne, V.: On the merge of factor canonical bases. In: International Conference on Formal Concept Analysis ICFCA, pp. 182–198. Springer Berlin Heidelberg (2008)
20. Visani, M., Bertet, K., Ogier, J.M.: Navigala: an Original Symbol Classifier Based on Navigation through a Galois Lattice. *International Journal on Pattern Recognition and Artificial Intelligence (IJPRAI)* (2011)
21. Wille, R.: Subdirekte produkte und konjunkte summen. *Journal für die reine und angewandte Mathematik* 0239_0240, 333–338 (1969)
22. Wille, R.: Subdirekte Produkte vollständiger Verbände. *J. reine angew. Math.* 283/284, 53–70 (1976)
23. Wille, R.: Subdirect decomposition of concept lattices. *Algebra Universalis* 17, 275–287 (1983)
24. Wille, R.: Subdirect product construction of concept lattices. *Discrete Mathematics* 63(2-3), 305–313 (1987)