TWO BASIC PROBLEMS OF INCREMENTAL CONSTRUCTION IN FORMAL CONCEPT ANALYSIS

Martin Kauer

Dissertation Thesis

Department of Computer Science
Faculty of Science
Palacky University Olomouc

2018

Author
Martin Kauer
Department of Computer Science
Faculty of Science
Palacký University Olomouc
17. listopadu 12
CZ–771 46 Olomouc
Czech Republic
kauer.martin@gmail.com

Supervisor
doc. RNDr. Michal Krupka, Ph.D.

Keywords: formal concept analysis, concept lattice construction, incremental algorithms, incidence removal, preconcept removal, substructures of concept lattices, generated complete sublattices, semi-closed subrelations, interval-preconcepts, basic theorem.

Hereby I declare that the thesis is my original work.
Most parts of this thesis are based on outcomes of the joint scientific work with Michal Krupka. All authors have even share in the results.

Dedicated to my family and to Michal.

Synopsis

Formal Concept Analysis (FCA) is a field of applied mathematics based on formalization of the notion of concept from cognitive psychology and has been widely studied in the last several decades. From a description of objects by their features FCA derives a hierarchy of concepts which is formalized by a complete lattice called a concept lattice. We explore some fundamental aspects of FCA. First, we focus on incremental concept lattice construction and analysis of its basic step–removal of an incidence–and propose two algorithms for incremental concept lattice construction. Second, we study generated complete sublattices and show how their corresponding closed subrelations can be efficiently computed. Lastly, we investigate a new type of subrelations from which a new formal rectangle type arises, we provide motivation from cognitive psychology for it and propose a basic theorem for lattices of such rectangles.

Acknowledgements

This thesis was supported by grant No. PrF 2018 030 of IGA of Palacky University Olomouc.

Contents

Preface

This thesis focuses on several fundamental tasks in Formal Concept Analysis and presents new results concerning incremental concept lattice construction and construction of substructures. Namely, we study how to accommodate the smallest change in a formal context into its concept lattice. Then, we turn our attention to the related problem of finding a formal context for a sublattice that is given by a set of generators. Particular parts of this thesis are based on the following articles:

[1] KAUER, Martin; KRUPKA, Michal. Removing an Incidence from a Formal Context. In: BERTET, Karell; RUDOLPH, Sebastian (eds.). Proc. CLA 2014, pp. 195–206.

[2] KAUER, Martin; KRUPKA, Michal. Subset-generated complete sublattices as concept lattices. In: YAHIA, Sadok Ben; KONECNY, Jan (eds.). Proc. CLA 2015, pp. 11-21.

[3] KAUER, Martin; KRUPKA, Michal. Generating complete sublattices by methods of formal concept analysis. Int. J. General Systems. 2017, vol. 46, no. 5, pp. 475–489.

The thesis is organized as follows. In the first chapter, we provide short preliminaries covering basic notions that are used in the following chapters. The main content of this thesis is split into two parts.

The first part, based on [1], can be found in Chapter 2. It is dedicated to a fundamental problem in incremental concept lattice construction–removing an incidence from a formal context. The first section of this chapter covers basic notions and draws useful correspondences. Building on that, Section 2.2 presents theoretical analysis of the problem at hand together with an algorithm for recomputing concepts after removal of an incidence. Section 2.3 extends the analysis and the algorithm by also covering structural changes. Experiments providing some insight into the performance of presented method can be found in Section 2.4. Further possible extensions are described in Section 2.5 and include removal of an arbitrary preconcept and discussion on possible parallelization. A summary of the first part, together with remarks on related work, can be found in Section 2.6.

The second part, based on [2, 3], is captured in Chapter 3 and it is devoted to the study of generated complete sublattices and subrelations. In the first section, we show a method for computing a closed subrelation for a generated complete sublattice without constructing any lattices. Section 3.2 contains experiments with presented method. Although we have an obvious upper bound for a complexity of our method, we believe that it can be tightened and Section 3.3 provides some insight into this. In Section 3.4, we identify the subrelations for which the closure to closed subrelation always exists and for an arbitrary subrelation we characterize all closed subrelations containing it. Section 3.5 introduces a new type of formal rectangle, draws its connections to other types, to block relations and presents two basic theorems for lattices of such rectangles. Discussion and remarks on related work can be found in Section 3.6.

Lastly, the thesis is closed by Chapter 4 containing concluding remarks.

Chapter 1

Preliminaries

In this chapter we provide a brief introduction to Formal Concept Analysis and other topics related to the content of the following chapters. We will not dwell on details here as all basic topics have been widely studied and all the details can be found in the cited sources.

1.1 Partially ordered sets, complete lattices and closures

Recall that a binary relation R on a set U is a *(partial) order*, if it satisfies the following conditions for all elements $u, v, w \in U$:

1. $\langle u, u\rangle \in R$ (reflexivity)
2. $\langle u, v\rangle \in R$ and $\langle v, u\rangle \in R$ implies $u = v$ (antisymmetry)
3. $\langle u, v\rangle \in R$ and $\langle v, w\rangle \in R$ implies $\langle u, w\rangle \in R$ (transitivity)

We usually denote the order relation R by $\leq$, its inverse R^{-1} by $\geq$, and we write $u < v$ for $u \leq v$ and $u \neq v$. Moreover, if it holds either $u \leq v$ or $v \leq u$ for every $u, v \in V$, then we call $\leq$ a *total order*. A set U together with a partial order on U is called a *partially ordered set* or *poset* for short. By the linear extension principle, every partial order $\leq$ on U can be extended to a total order $\preceq$ on U such that for every $u, v \in U$, if $u \leq v$, then $u \preceq v$. For $u, w \in U$, u is called a *lower neighbor* of w, if $u < w$ and there is no element v fulfilling $u < v < w$. In this case, w is called an *upper neighbor* of u, we write $u \prec w$ and we can also read it as *w covers u*. Lastly, for $u, w \in U$ the set $[u, w] = \{v \in U \mid u \leq v \leq w\}$ is called a *closed interval*.

A poset U is called a *complete lattice* if each subset $P \subseteq U$ has the least upper bound (*supremum*) and the greatest lower bound (*infimum*). We denote these by $\bigvee P$ and $\bigwedge P$, respectively. An element $u \in U$ is called $\bigvee$*-irreducible* (resp. $\bigwedge$*-irreducible*) if it cannot be expressed as a supremum of strictly smaller (resp. greater) elements of U. If the element is not $\bigvee$-irreducible (resp. $\bigwedge$-irreducible) we call it $\bigvee$*-reducible* (resp. $\bigwedge$*-reducible*). A subset $V \subseteq U$ is called $\bigvee$*-dense* (resp. $\bigwedge$*-dense*), if each element $u \in U$ can be obtained as suprema (resp. infima) of some elements from V. A subset $V \subseteq U$ is a $\bigvee$*-subsemilattice* (resp. $\bigwedge$*-subsemilattice*, resp. *complete sublattice*) *of* U, if for each $P \subseteq V$ it holds $\bigvee P \in V$ (resp. $\bigwedge P \in V$, resp. $\{\bigvee P, \bigwedge P\} \subseteq V$), i.e. the set V is closed under arbitrary suprema (resp. infima, resp. both previous). A subset $V \subseteq U$ is called an *order-embedded complete lattice*, if it is a complete lattice with the induced order (it does not have to be a sublattice). More details on order-embedded complete lattices can be found in [4].

For a subset $P \subseteq U$ we denote by $\mathrm{C}_{\bigvee} P$ the $\bigvee$-subsemilattice of U *generated by* P, i.e. the smallest (w.r.t. set inclusion) $\bigvee$-subsemilattice of U containing P. $\mathrm{C}_{\bigvee} P$ always exists and is equal to the intersection of all $\bigvee$-subsemilattices of U containing P. The $\bigwedge$-subsemilattice of U generated by P and the complete sublattice of U generated by P are defined similarly and are denoted by $\mathrm{C}_{\bigwedge} P$ and $\mathrm{C}_{\bigvee\bigwedge} P$, respectively. More on posets and lattices can be found in [5].

The operators $\mathrm{C}_{\bigvee}$, $\mathrm{C}_{\bigwedge}$ and $\mathrm{C}_{\bigvee\bigwedge}$ are closure operators on the set U. Recall that a *closure* (resp. *interior*) *operator on a set* X is a mapping $\mathrm{C}\colon 2^X \to 2^X$, where 2^X is the power-set of X (i.e. the set of all subsets of X), satisfying for all sets $A, A_1, A_2 \subseteq X$

1. $A \subseteq \mathrm{C}(A)$ (resp. $\mathrm{C}(A) \subseteq A$),
2. if $A_1 \subseteq A_2$, then $\mathrm{C}(A_1) \subseteq \mathrm{C}(A_2)$,
3. $\mathrm{C}(\mathrm{C}(A)) = \mathrm{C}(A)$.

An *isotone* (resp. *antitone*) *Galois connection* between two posets U and V is a pair of isotone (resp. antitone) functions $\langle f, g\rangle$ where $f : U \to V$, $g : V \to U$ satisfying

$$a \leq g(b) \text{ iff } f(a) \leq b \text{ (resp. } b \leq f(a)).$$

For an isotone (resp. antitone) Galois connection $\langle f, g\rangle$ the function composition $g \circ f$, given by $(g \circ f)(u) = g(f(u))$, is a closure operator on U and $f \circ g$

is an interior (resp. closure) operator on V. We define isotone (resp. antitone) Galois connection between two sets U and V as previously defined isotone (resp. antitone) Galois connection on their respective power-sets equipped with the subsethood ordering.

1.2 Formal Concept Analysis

Formal Concept Analysis was first introduced by R. Wille in [6] and has been widely studied ever since. The original motivation has its roots in human psychology and in the Port-Royal logic. Various generalizations and extensions of FCA were proposed over last years, see [7] for an overview. Our basic reference is [8].

A (*formal*) *context* is a triple $\langle X, Y, I\rangle$ where X is a set of objects, Y a set of attributes and $I \subseteq X \times Y$ a binary relation between X and Y specifying for each object its attributes.

For subsets $A \subseteq X$ and $B \subseteq Y$ we set

$$A^{\uparrow_I} = \{y \in Y \mid \text{for each } x \in A \text{ it holds } \langle x, y\rangle \in I\},$$
$$B^{\downarrow_I} = \{x \in X \mid \text{for each } y \in B \text{ it holds } \langle x, y\rangle \in I\}.$$

We call ${}^{\uparrow_I}, {}^{\downarrow_I}$ *derivation operators of* I. The pair $\langle {}^{\uparrow_I}, {}^{\downarrow_I}\rangle$ is an antitone Galois connection between the sets X and Y, therefore, the operator ${}^{\uparrow_I\downarrow_I}$ is a closure operator on X and the operator ${}^{\downarrow_I\uparrow_I}$ is a closure operator on Y.

A pair $\langle A, B\rangle$ satisfying $A^{\uparrow_I} = B$ and $B^{\downarrow_I} = A$ is called a (*formal*) *concept* of $\langle X, Y, I\rangle$. The set A is called *the extent of* $\langle A, B\rangle$, the set B *the intent of* $\langle A, B\rangle$. We denote $\mathrm{Ext}(X, Y, I)$ (resp. $\mathrm{Int}(X, Y, I)$) the set of all extents (resp. intents) of formal concepts of $\langle X, Y, I\rangle$. When there is no danger of confusion, we can use the term "an extent of I" instead of "the extent of a concept of $\langle X, Y, I\rangle$", similarly for intents, and "a concept of I" instead of "a concept of $\langle X, Y, I\rangle$". If the formal context is fixed we use terms "a concept", "an extent" and "an intent".

Several generalizations of the notion of formal concept have been proposed over the years. We call a pair $\langle A, A^{\uparrow_I}\rangle$ a $\sqcap$-*semiconcept* and a pair $\langle B^{\downarrow_I}, B\rangle$ a $\sqcup$-*semiconcept*. Combining the previous two notions we get a general notion of *semiconcept* [9]. We call a pair $\langle A, B\rangle$ satisfying $A^{\uparrow_I\downarrow_I} = B^{\downarrow_I}$ ($\Leftrightarrow B^{\downarrow_I\uparrow_I} = A^{\uparrow_I}$) a *protoconcept* [10]. Clearly, each semiconcept is also a potoconcept. These notions were motivated by their use for efficient description of formal concepts, namely, each protoconcept describes exactly one formal concept. Also, they were used to develop *Boolean Concept Logic* [10]. The most general notion of *preconcept* is a pair $\langle A, B\rangle$ satisfying $A \subseteq B^{\downarrow_I}$ and $B \subseteq A^{\uparrow_I}$ [11, 12]. Preconcepts are just formal rectangles in our data and motivation for this notion comes from cognitive psychology, namely, from J. Piaget stating that concepts originate in child development from images, ideas and preconcepts [13].

A partial order $\leq$ on the set $\mathcal{B}(X, Y, I)$ of all formal concepts of $\langle X, Y, I\rangle$ is defined by $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (iff $B_2 \subseteq B_1$). $\mathcal{B}(X, Y, I)$ along with $\leq$ is called *the concept lattice of* $\langle X, Y, I\rangle$. By the basic theorem on concept lattices [8, Theorem 3], $\mathcal{B}(X, Y, I)$ is a complete lattice with infima and suprema given by

$$\bigwedge_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \bigcap_{\iota\in\mathcal{I}} A_\iota, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle, \tag{1.1}$$

$$\bigvee_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \bigcap_{\iota\in\mathcal{I}} B_\iota \right\rangle. \tag{1.2}$$

Moreover, a complete lattice V is isomorphic to $\mathcal{B}(X, Y, I)$ if and only if there are mappings $\gamma_I : X \to V$ and $\mu_I : Y \to V$ such that $\gamma_I(X)$ is $\bigvee$-dense in V, $\mu_I(Y)$ is $\bigwedge$-dense in V and xIy is equivalent to $\gamma_I(x) \leq \mu_I(y)$ for all $x \in X$ and all $y \in Y$. In particular, V is isomorphic to $\mathcal{B}(V, V, \leq)$.

One of immediate consequences of (1.1) and (1.2) is that the intersection of any system of extents, resp. intents, is again an extent, resp. intent, and that it can be expressed as follows:

$$\bigcap_{\iota\in\mathcal{I}} B_\iota = \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I}, \quad \text{resp.} \quad \bigcap_{\iota\in\mathcal{I}} A_\iota = \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I},$$

for concepts $\langle A_\iota, B_\iota\rangle \in \mathcal{B}(X, Y, I)$, $\iota \in \mathcal{I}$.

Concepts $\mu_I(y) = \langle \{y\}^{\downarrow_I}, \{y\}^{\downarrow_I\uparrow_I}\rangle$ where $y \in Y$ are called *attribute concepts*, their extents are called *attribute extents* and intents are called *attribute intents*. According to the previous, each concept $\langle A, B\rangle$ is an infimum of some attribute concepts. More specifically, $\langle A, B\rangle$ is the infimum of attribute concepts $\langle \{y\}^{\downarrow_I}, \{y\}^{\downarrow_I\uparrow_I}\rangle$ for $y \in B$ and $A = \bigcap_{y\in B}\{y\}^{\downarrow_I}$.

Dually, concepts $\gamma_I(x) = \langle \{x\}^{\uparrow_I\downarrow_I}, \{x\}^{\uparrow_I}\rangle$ for $x \in X$ are called *object concepts*, they are $\bigvee$-dense in $\mathcal{B}(X, Y, I)$ and for each concept $\langle A, B\rangle$ we have $B = \bigcap_{x\in A}\{x\}^{\uparrow_I}$.

When the set of objects X and the set of attributes Y are fixed, we denote the concept lattice of $\langle X, Y, I\rangle$ just by $\mathcal{B}(I)$.

The *direct product* of formal contexts $\langle X_1, Y_1, I_1\rangle$ and $\langle X_2, Y_2, I_2\rangle$ is given by

$$\langle X_1, Y_1, I_1\rangle \times \langle X_2, Y_2, I_2\rangle = (X_1 \times X_2, Y_1 \times Y_2, K)$$
$$\text{where } (x_1, x_2)K(y_1, y_2) \Leftrightarrow x_1 I_1 y_1 \text{ or } x_2 I_2 y_2.$$

The concept lattice of the direct product of some formal contexts is called a *tensor product of concept lattices.*

For any set of preconcepts $Q \subseteq \mathcal{B}(X, Y, I)$ we set

$$\bigsqcup Q = \bigcup\{A \times B \mid \langle A, B\rangle \in Q\}.$$

$\bigsqcup Q$ is the subrelation of I equal to the union of rectangles given by preconcepts from Q.

Chapter 2

Concept lattice construction by incidence removals

We open this chapter with a fundamental question about concept lattice construction, specifically, what effect does removing an incidence from a formal context have on its concept lattice. This question is known as the problem of "killing a cross" which was coined by R. Wille in the early days of FCA. Solving this problem is desirable not only from the theoretical but also from the practical point of view because it leads us to an efficient method of computing concept lattices of two very similar formal contexts. Moreover, it seems that any incremental method for concept lattice construction has this problem rooted into it.

Traditionally, we need to recompute whole concept lattice upon the slightest change in the input data. Although there have been several incremental algorithms introduced (see [14, 15, 16, 17, 18, 19] and also [20] for a comparison of some of the algorithms) they usually operate on object (resp. attribute) level. We focus on a finer approach and study the problem of removing a single incidence from a formal context. Our goal is to provide a detailed analysis of this problem and based on it we propose two incremental algorithms for an efficient reconstruction of the concept lattice after the removal.

Throughout this chapter we consider a formal context $\langle X, Y, J\rangle$ which results from a formal context $\langle X, Y, I\rangle$ by removing a single incidence $\langle x_0, y_0\rangle$, i.e. $I = J \cup \langle x_0, y_0\rangle$ and $\langle x_0, y_0\rangle \notin J$. We denote the respective concept lattices by $\mathcal{B}(J)$ and $\mathcal{B}(I)$. Because we take the formal context $\langle X, Y, I\rangle$ as the starting point, we call it, and everything related to it (including derivation operators, $\mathcal{B}(I)$, ...), *initial*. Similarly, we call *final* everything related to the formal context $\langle X, Y, J\rangle$. We analyze necessary changes that are to be made in the initial concept lattice to obtain the final concept lattice.

2.1 Basic notions and correspondences

We start by examining how the derivation operators of initial formal context $({}^{\uparrow_I}, {}^{\downarrow_I})$ relate to the ones of final formal context $({}^{\uparrow_J}, {}^{\downarrow_J})$. In the case where we remove a single incidence, this relation is quite straightforward as can be seen in the following lemma.

Lemma 1. *For each $A \subseteq X$ and $B \subseteq Y$ it holds*

$$A^{\uparrow_J} = \begin{cases} A^{\uparrow_I} & \text{if } x_0 \notin A, \\ A^{\uparrow_I} \setminus \{y_0\} & \text{if } x_0 \in A, \end{cases} \qquad B^{\downarrow_J} = \begin{cases} B^{\downarrow_I} & \text{if } y_0 \notin B, \\ B^{\downarrow_I} \setminus \{x_0\} & \text{if } y_0 \in B. \end{cases}$$

In particular, $A^{\uparrow_J} \subseteq A^{\uparrow_I}$ and $B^{\downarrow_J} \subseteq B^{\downarrow_I}$.

Proof. Immediate from the fact that $\langle X, Y, I\rangle = \langle X, Y, J \cup \langle x_0, y_0\rangle\rangle$. □

It is obvious that not all initial concepts have to be influenced by the removal and there might be some concepts belonging into both $\mathcal{B}(I)$ and $\mathcal{B}(J)$. We call such concepts *steady* since they remain unchanged and do not require any reconstruction while computing $\mathcal{B}(J)$ from $\mathcal{B}(I)$. For this reason, it is important to identify steady concepts, and crucially, concepts that are not steady, *unsteady* for short, because they are the ones we need to reconstruct. As it turns out, unsteady initial concepts form a bounded sublattice of $\mathcal{B}(I)$. This sublattice is not generally complete and it is equal to the closed interval $[\gamma_I(x_0), \mu_I(y_0)]$. Therefore, this sublattice (resp. closed interval) is the only part of the concept lattice we need to focus on while pursuing our goal of computing $\mathcal{B}(J)$ based on $\mathcal{B}(I)$.

Lemma 2. *A concept $c \in \mathcal{B}(I)$ is unsteady iff $c \in [\gamma_I(x_0), \mu_I(y_0)]$.*

Proof. If $c = \langle A, B\rangle \notin [\gamma_I(x_0), \mu_I(y_0)]$, then either $x_0 \notin A$, or $y_0 \notin B$. If, for instance, $x_0 \notin A$, then by Lemma 1, $B = A^{\uparrow_I} = A^{\uparrow_J}$, showing B is the intent of a $d \in \mathcal{B}(J)$. Now by Lemma 1,

$$B^{\downarrow_J} = \begin{cases} B^{\downarrow_I} = A & \text{if } y_0 \notin B, \\ B^{\downarrow_I} \setminus \{x_0\} = A \setminus \{x_0\} = A & \text{if } y_0 \in B \end{cases}$$

and so $d = c$. The case $y_0 \notin B$ is dual.

To prove the opposite direction it is sufficient to notice that $c \in [\gamma_I(x_0), \mu_I(y_0)]$ is equivalent to $\langle x_0, y_0 \rangle \in A \times B$, excluding the case $\langle A, B \rangle \in \mathcal{B}(J)$. $\square$

Remark 3. A well-known result from the lattice theory states that each closed interval in a lattice is also its sublattice. Moreover, a closed interval in a complete lattice is, by itself, a complete lattice (with the induced order), however it is not necessarily a complete sublattice. It would be the case if the bounds of the interval coincide with the bounds of the whole lattice. Throughout this chapter we usually use the interval terminology to describe the set of all unsteady initial concepts.

Remark 4. We can construct a formal context K corresponding to the unsteady initial complete lattice by taking $K = \bigsqcup[\gamma_I(x_0), \mu_I(y_0)]$ and removing attributes $Y \setminus \{x_0\}^{\uparrow_I}$ and objects $X \setminus \{y_0\}^{\downarrow_I}$. This should become clearer after reading the second part of this thesis where we investigate more on the topic of substructures. Nevertheless, we can readily see that $\mathcal{B}(K)$ contains the initial unsteady interval but the bounds of the lattice does not have to coincide with the bounds of the interval. Removing said objects and attributes makes those bounds coincide without changing the elements of the interval because corresponding rows and columns are actually empty in K. Indeed, take an attribute $y_j \notin \{x_0\}^{\uparrow_I}$ and take any incidence $\langle x_i, y_j \rangle \in I$. Such incidence cannot be part of any unsteady initial concept (so it does not belong to K) due to $y_j \notin \{x_0\}^{\uparrow_I}$.

Lemma 5. *If a concept $\langle C, D \rangle \in \mathcal{B}(J)$ is unsteady, then either $x_0 \in C$ or $y_0 \in D$.*

Proof. Using contraposition, if $x_0 \notin C$ and $y_0 \notin D$ we get $C^{\uparrow_I} = C^{\uparrow_J}$ and $D^{\downarrow_I} = D^{\downarrow_J}$ showing steadiness of $\langle C, D \rangle$. $\square$

Moving forward with our analysis, we are going to chain derivation operators ${}^{\uparrow_I}, {}^{\downarrow_I}, {}^{\uparrow_J}, {}^{\downarrow_J}$, however, doing so quickly leads to lack of clarity even in otherwise simple statements. To help alleviate this problem, we introduce four *child operators* ${}_{\Box}, {}_{\boxtimes}, {}^{\Box}, {}^{\boxtimes}$ which we use throughout the rest of this chapter. The idea behind them is to relate concepts of $\mathcal{B}(I)$ to concepts of $\mathcal{B}(J)$ in a natural way that simplifies our analysis.

Definition 6 (child operators). For concepts $c = \langle A, B\rangle \in \mathcal{B}(I), d = \langle C, D\rangle \in \mathcal{B}(J)$ we set

$$c^{\Box} = \langle A^{\Box}, B^{\Box}\rangle = \langle A^{\uparrow_J\downarrow_J}, A^{\uparrow_J}\rangle, \qquad c_{\Box} = \langle A_{\Box}, B_{\Box}\rangle = \langle B^{\downarrow_J}, B^{\downarrow_J\uparrow_J}\rangle,$$
$$d^{\boxtimes} = \langle C^{\boxtimes}, D^{\boxtimes}\rangle = \langle D^{\downarrow_I}, D^{\downarrow_I\uparrow_I}\rangle, \qquad d_{\boxtimes} = \langle C_{\boxtimes}, D_{\boxtimes}\rangle = \langle C^{\uparrow_I\downarrow_I}, C^{\uparrow_I}\rangle.$$

Evidently, $c^{\Box}, c_{\Box} \in \mathcal{B}(J)$ and $d^{\boxtimes}, d_{\boxtimes} \in \mathcal{B}(I)$. $c^{\Box}$ (resp. $c_{\Box}$) is called *the upper* (resp. *lower*) *child of* c. It holds $d^{\boxtimes} = d_{\boxtimes}$ and it is the (unique) concept from $\mathcal{B}(I)$ containing, as a rectangle, the rectangle represented by d.

EXAMPLE 7. A basic example of the problem setting can be found in Fig. 2.1 and 2.2. The bold dot in the formal context marks the incidence we are removing. Dashed circles in the initial lattice mark unsteady initial concepts.

I	y_0	y_1	y_2
x_0	•	×	
x_1	×	×	
x_2			×

Figure 2.1: A formal context I. The bold dot marks the incidence $\langle x_0, y_0\rangle$ which is removed in order to obtain J.

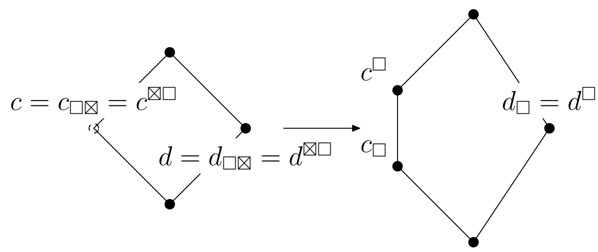


Figure 2.2: Two corresponding lattices $\mathcal{B}(I)$ (left) and $\mathcal{B}(J)$ (right) where $c = \langle\{x_0, x_1\}, \{y_0, y_1\}\rangle$ and $d = \langle\{x_2\}, \{y_2\}\rangle$.

REMARK 8. The equality $d^{\boxtimes} = d_{\boxtimes}$ would generally not hold if we would work in the settings where $I \setminus J$ has more than one element. Also, it is worth noting that every final concept is a semiconcept of $\mathcal{B}(I)$. Moreover, unsteady final concepts are all proper semiconcepts of $\mathcal{B}(I)$.

It is useful to show some basic properties of pairs of child operators $\langle^{\Box},^{\boxtimes}\rangle$ and $\langle_{\Box},_{\boxtimes}\rangle$ as we use them heavily in our analysis. These properties might remind you of the properties of Galois connections, and rightfully so, although the pairs $\langle^{\Box},^{\boxtimes}\rangle$ and $\langle_{\Box},_{\boxtimes}\rangle$ do not form Galois connections.

Lemma 9. *The mappings $c \mapsto c^{\square}$, $c \mapsto c_{\square}$, and $d \mapsto d^{\boxtimes}$ are isotone and satisfy*

$$\begin{array}{llll} c \le c^{\square\boxtimes}, & d \le d^{\boxtimes\square}, & c^{\square\boxtimes\square} = c^{\square}, & d^{\boxtimes\square\boxtimes} = d^{\boxtimes}, \\ c \ge c_{\square\boxtimes}, & d \ge d_{\boxtimes\square}, & c_{\square\boxtimes\square} = c_{\square}, & d_{\boxtimes\square\boxtimes} = d_{\boxtimes}. \end{array}$$

Proof. Isotony follows directly from definition.

Let $c = \langle A, B\rangle$. From Lemma 1 we have $A^{\uparrow_J} \subseteq A^{\uparrow_I}$. Thus, $A = A^{\uparrow_I\downarrow_I} \subseteq A^{\uparrow_J\downarrow_I}$, whence $c \le c^{\square\boxtimes}$. Similarly, for $d = \langle C, D\rangle$, $D^{\downarrow_J} \subseteq D^{\downarrow_I}$, whence $D^{\downarrow_I\uparrow_J} \subseteq D^{\downarrow_J\uparrow_J} = D$.

To prove $c^{\square\boxtimes\square} = c^{\square}$ it suffices to show that for the extent A of c it holds $A^{\uparrow_J\downarrow_I\uparrow_J} = A^{\uparrow_J}$. By Lemma 1, we have two possibilities: either $A^{\uparrow_J} = A^{\uparrow_I}$, or $A^{\uparrow_J} = A^{\uparrow_I} \setminus \{y_0\}$. In the first case $A^{\uparrow_J\downarrow_I\uparrow_J} = A^{\uparrow_J}$ holds trivially, in the second case $A^{\uparrow_J\downarrow_I} = A^{\uparrow_J\downarrow_J}$ (by the same lemma, because $y_0 \notin A^{\uparrow_J}$) and $A^{\uparrow_J\downarrow_I\uparrow_J} = A^{\uparrow_J\downarrow_J\uparrow_J} = A^{\uparrow_J}$. The equality $d^{\boxtimes\square\boxtimes} = d^{\boxtimes}$ can be proved similarly.

The assertions for lower children are dual. $\square$

Remark 10. The reason why the pair $\langle^{\square}, ^{\boxtimes}\rangle$ does not form an isotone Galois connection is due to only one direction holding in the following equivalence:

$$c^{\square} \le d \iff c \le d^{\boxtimes} \text{ means by the definition } A^{\uparrow_J\downarrow_J} \subseteq C \iff A \subseteq D^{\downarrow_I}.$$

$$\text{It only holds } A^{\uparrow_J\downarrow_J} \subseteq C \implies C^{\uparrow_J} = D \subseteq A^{\uparrow_J} \subseteq A^{\uparrow_I} \implies A = A^{\uparrow_I\downarrow_I} \subseteq D^{\downarrow_I}.$$

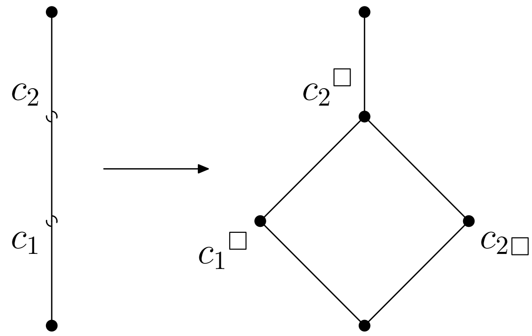
Even though the respective pairs of child operators do not form Galois connections, we still obtain closure (resp. interior) operators by their composition as summarized in the following corollary. We call them *compound child* operators.

Corollary 11 (compound child operators). *The mappings $c \mapsto c^{\square\boxtimes}$ and $d \mapsto d^{\boxtimes\square}$ are closure operators and the mappings $c \mapsto c_{\square\boxtimes}$ and $d \mapsto d_{\boxtimes\square}$ are interior operators.*

Remark 12. It is a well-known fact (see [21] for more details) that every closure operator can be obtained as a composition of suitable Galois connection and it might be tempting to specify it for our compound child operators. Unfortunately, we are not able to create a suitable Galois connection just by using child (resp. derivation) operators. Consider the following example.

I	y_0	y_1	y_2	y_3
x_0	•	×	×	
x_1	×	×		
x_2				

Given formal context.



The initial and final concept lattices.

Lets us try to find an isotone Galois connection $\langle f, g \rangle$ based on child (resp. derivation) operators that would make up the closure operator ${}^{\square\boxtimes}$. By an easy inspection we find it impossible. At first, notice that all initial concepts are fixpoints of the closure operator ${}^{\square\boxtimes}$. Operators ${}^{\boxtimes}$ and ${}_{\boxtimes}$ coincide, therefore, we have no choice in specifying map g. The only choice available lies within the image of the concept $c_2 = \langle\{x_0, x_1\}, \{y_0, y_1\}\rangle$ but by putting $f(c_2) = c_{2\square}$ we do not obtain an isotone function.

We now prove a lemma providing several equivalent conditions determining steadiness of initial concepts. We use it further to illuminate the connection between child operators and steady concepts.

Lemma 13. *The following assertions are equivalent for any $c = \langle A, B \rangle \in \mathcal{B}(I)$.*

1. *c is steady,*
2. *$A^{\uparrow_I} = A^{\uparrow_J}$,*
3. *$B^{\downarrow_I} = B^{\downarrow_J}$.*

Proof. “2 ⇒ 3”: by Lemma 1, $A \subseteq A^{\uparrow_J\downarrow_J} = B^{\downarrow_J} \subseteq B^{\downarrow_I} = A$.

“3 ⇒ 2”: dual.

The other implications follow by definition, since c is steady iff both 2. and 3. are satisfied. □

Utilizing the previous lemma, we can characterize steady concepts of $\mathcal{B}(I)$ and $\mathcal{B}(J)$ respectively. We already know from the previous observations that a concept $c \in \mathcal{B}(I)$ is steady if and only if it does not belong into the interval

$[\gamma_I(x_0), \mu_I(y_0)]$. This turns out to be equivalent of saying that children of c are–again–exactly c.

Lemma 14 (steady concepts in $\mathcal{B}(I)$). *The following assertions are equivalent for a concept $c \in \mathcal{B}(I)$:*

1. *c is steady,*
2. *$c \notin [\gamma_I(x_0), \mu_I(y_0)]$,*
3. *$c = c^{\square}$,*
4. *$c = c_{\square}$,*
5. *$c^{\square} = c_{\square}$.*

Proof. Directly from Lemma 13. □

Evidently, unsteady final concepts cannot form a closed interval in $\mathcal{B}(J)$ and their actual structure is more general. That being said, it still holds that a concept d is steady if and only if it is equivalent to each of its children.

Lemma 15 (steady concepts in $\mathcal{B}(J)$). *The following assertions are equivalent for a concept $d \in \mathcal{B}(J)$:*

1. *d is steady,*
2. *$d = d^{\boxtimes}$,*
3. *$d^{\boxtimes}$ is steady.*

Proof. Directly from Lemma 13. □

Remark 16. The fact that children of a steady concept c are equivalent to c plays nicely with our terminology.

Equipped with several equivalent characterizations of steady concepts, we are free to choose a suitable one for a given application. In the following proofs, we freely use all of the above characterizations, but in algorithms, one should always choose the less computationally expensive characterization despite having more complicated formal form.

2.2 Computing the final concepts

Building on basic notions and observations from the previous section, we will now turn our attention to the problem of computing concepts of $\mathcal{B}(J)$ given concepts of $\mathcal{B}(I)$. In the pursuit of this goal, we will make use of the child operators from the previous section.

We start with a technical lemma stating a relation between extents and intents of a concept and its children.

Lemma 17. *The following holds for $c = \langle A, B\rangle \in \mathcal{B}(I)$ and $d = \langle C, D\rangle \in \mathcal{B}(J)$: If $d = c^{\square}$, then $B \in \{D, D \cup \{y_0\}\}$ and if $d = c_{\square}$, then $A \in \{C, C \cup \{x_0\}\}$.*

Proof. By definition of $^{\square}$, $D = A^{\uparrow_J}$, which is by Lemma 1 either equal to B, or to $B \setminus \{y_0\}$. Similarly for $_{\square}$. □

Using the previous lemma we now prove a following theorem which is the backbone of our algorithm for computing the final concepts from the initial ones. It states that for each unsteady initial concept there exists exactly one unsteady final concept such that these two are related via the mappings $^{\square}, ^{\boxtimes}$ or $_{\square}, _{\boxtimes}$.

Theorem 18. *An unsteady concept $d \in \mathcal{B}(J)$ is a (upper or lower) child of exactly one concept $c \in \mathcal{B}(I)$. This concept is unsteady and satisfies $c = d^{\boxtimes} = d_{\boxtimes}$.*

Proof. Let $d = \langle C, D\rangle$. Since d is unsteady, then either $C^{\uparrow_I} \neq C^{\uparrow_J}$, or $D^{\downarrow_I} \neq D^{\downarrow_J}$. Suppose $C^{\uparrow_I} \neq C^{\uparrow_J}$ and set $A = C$, $B = C^{\uparrow_I}$. By Lemma 1, $x_0 \in C$, $y_0 \notin D$ and $B = D \cup \{y_0\}$. By the same lemma, $A = C = D^{\downarrow_J} = D^{\downarrow_I}$, whence A is an extent of I. Thus, $c = \langle A, B\rangle \in \mathcal{B}(I)$ and it is unsteady because $x_0 \in A$ and $y_0 \in B$ (Lemma 2). Since $D = C^{\uparrow_J} = A^{\uparrow_J}$, $d = c^{\square}$. $A = C$ yields $c = d_{\boxtimes}$.

We prove uniqueness of c. By Lemma 17, if for $c' = \langle A', B'\rangle \in \mathcal{B}(I)$ we have $d = c'^{\square}$, then either $B' = D$, or $B' = D \cup \{y_0\}$. The first case is impossible, because it would make D an intent of I and, consequently, d a steady concept. The second case means c' equals c above. There is a third case left: if $d = c'_{\square}$, then $C = B'^{\downarrow_J}$. Since $x_0 \in C$, we have $y_0 \notin B'$ (Lemma 1). Thus, $C = B'^{\downarrow_I}$ (Lemma 1 again). Consequently, $C^{\uparrow_I} = B'$ and since $y_0 \notin B'$, $B' = C^{\uparrow_J}$ (Lemma 1 for the last time). Thus, $d = c'$, which is a contradiction with unsteadiness of d.

The case $D^{\downarrow_I} \neq D^{\downarrow_J}$ is proved dually (in this case we obtain $d = c_{\square}$). □

The theorem leads to the following simple way of constructing $\mathcal{B}(J)$ from $\mathcal{B}(I)$. For each $c \in \mathcal{B}(I)$ the following has to be done:

1. If c is steady, then it has to be added to $\mathcal{B}(J)$.
2. If c is not steady, then each its unsteady child, i.e. each unsteady element of $\{c^{\square}, c_{\square}\}$, has to be added to $\mathcal{B}(J)$.

This method ensures that all proper elements will be added to $\mathcal{B}(J)$ (i.e. no element will be omitted) and each element will be added exactly once.

Steady (resp. unsteady) concepts can be identified by any means proposed previously. The following lemma shows a simple way of determining whether a child of an unsteady initial concept is steady. It also describes the role of fixpoints of the compound child operators.

Lemma 19. *Let c be an unsteady concept of $\mathcal{B}(I)$. Then*

- *$c^{\square}$ is unsteady iff c is a fixpoint of ${}^{\square\boxtimes}$,*
- *$c_{\square}$ is unsteady iff c is a fixpoint of ${}_{\square\boxtimes}$.*

Proof. If $c^{\square}$ is not steady, then $c = (c^{\square})^{\boxtimes}$ by Theorem 18. On the other hand, if $c^{\square}$ is steady, then $c^{\square\boxtimes} = c^{\square}$ by Lemma 15, which rules out $c^{\square\boxtimes} = c$, because in that case c would be equal to $c^{\square}$, which would make it steady by Lemma 14.

The proof for $c_{\square}$ is dual. □

Example 20. In Fig. 2.5-2.10 we can see several examples of formal contexts with initial concepts of different types w.r.t. compound child operators.

The proposed method is utilized in Algorithm 1 which computes the final concepts from the initial ones but does not take into the account the ordering.

I	y_0	y_1	y_2
x_0	•	×	×
x_1			
x_2			

Figure 2.5: The least concept is unsteady. It is a fixpoint of both operators and it has two unsteady children $\langle\emptyset, \{y_0, y_1, y_2\}\rangle$ and $\langle\{x_0\}, \{y_1, y_2\}\rangle$.

I	y_0	y_1	y_2	y_3
x_0	•	×	×	×
x_1	×	×		
x_2	×		×	
x_3	×			×

Figure 2.6: Several non-trivial unsteady concepts are fixpoints of both operators. There is a clear pattern where each concept $\langle\{x_0, x_{i>0}\}, \{y_0, y_{i>0}\}\rangle$ have two unsteady children, $\langle\{x_i\}, \{y_0, y_i\}\rangle$ and $\langle\{x_0, x_i\}, \{y_i\}\rangle$.

I	y_0	y_1	y_2
x_0	•	×	
x_1	×	×	
x_2		×	

Figure 2.7: The concept $\langle\{x_0, x_1\}, \{y_0, y_2\}\rangle$ is a fixpoint of $_{\Box\boxtimes}$ but not $^{\Box\boxtimes}$ and has an unsteady child $c_{\Box} = \langle\{x_1\}, \{y_0, y_1\}\rangle$.

I	y_0	y_1	y_2
x_0	•	×	
x_1	×	×	×
x_2			

Figure 2.8: The concept $\langle\{x_0, x_1\}, \{y_0, y_1\}\rangle$ is a fixpoint of $^{\Box\boxtimes}$ but not $_{\Box\boxtimes}$ and has an unsteady child $c^{\Box} = \langle\{x_0, x_1\}, \{y_1\}\rangle$.

I	y_0	y_1	y_2
x_0	•		
x_1	×	×	
x_2			

Figure 2.9: The concept $\langle\{x_0, x_1\}, \{y_0\}\rangle$ is not a fixpoint of any operator and so has no unsteady children.

I	y_0	y_1	y_2	y_3	y_4
x_0	•	×		×	
x_1	×	×	×		
x_2		×			
x_3	×			×	×
x_4				×	

Figure 2.10: The two concepts, $\langle\{x_0, x_1\}, \{y_0, y_1\}\rangle$, $\langle\{x_0, x_3\}, \{y_0, y_3\}\rangle$, are not fixpoints of any operator and so have no unsteady children.

Time complexity of Algorithm 1 is clearly $O(|\mathcal{B}(I)||X||Y|)$ in the worst case scenario. Indeed, the number of unsteady concepts is at most equal to $|\mathcal{B}(I)|$ and the computation of operators $^{\Box\boxtimes}$, $_{\Box\boxtimes}$ can be done in $O(|X| \cdot |Y|)$ time. It is worth noting that the time complexity is heavily affected by the size of the interval $[\gamma_I(x_0), \mu_I(y_0)]$ which can be much smaller than the size of $\mathcal{B}(I)$.

Algorithm 1 Transforming concepts of $\mathcal{B}(I)$ into concepts of $\mathcal{B}(J)$.

procedure TransformConcepts($\mathcal{B}(I)$)
  $\mathcal{B}(J) \leftarrow \mathcal{B}(I)$;
  for all $c = \langle A, B\rangle \in [\gamma_I(x_0), \mu_I(y_0)]$ **do**
    $\mathcal{B}(J) \leftarrow \mathcal{B}(J) \setminus \{c\}$;
    if $c = c_{\square\boxtimes}$ **then**
      $\mathcal{B}(J) \leftarrow \mathcal{B}(J) \cup \{c_\square\}$;
    if $c = c^{\square\boxtimes}$ **then**
      $\mathcal{B}(J) \leftarrow \mathcal{B}(J) \cup \{c^\square\}$;
  return $\mathcal{B}(J)$;

Remark 21. While implementing Algorithm 1 one should be aware of some possible optimizations. First, in order to compute the results of derivation operators, which are basis for computing child operators, we can use Lemma 1 and use already provided extents and intents. Second, tests like $c = c^{\square\boxtimes}$ can be actually performed without computing the corresponding compound child operators. For example, it is sufficient to compute $c^\square$ and compare its extent to the extent of c. This comes as a direct consequence of presented results.

2.3 Computing the final lattice

In order to analyze structural changes in a concept lattice after removal of an incidence we need to investigate additional properties of the closure operator $^{\square\boxtimes}$ and the interior operator $_{\square\boxtimes}$. We focus mostly on their fixpoints.

Remark 22. It is worth noting that the set of all fixpoints of $^{\square\boxtimes}$ within induced order is itself a complete lattice but it is generally not a complete sublattice of $\mathcal{B}(I)$. It is in fact $\bigwedge$-subsemilattice of $\mathcal{B}(I)$. We can make a dual observation about the set of all fixpoints of $_{\square\boxtimes}$. These observations follows from well-known mathematical theorems such as Knaster-Tarski Theorem [22].

Putting together several characterizations of steady concepts we obtain the following lemma postulating that steady initial concept is always fixpoint of both compound child operators.

Lemma 23. *Each steady concept is a fixpoint of both* $^{\square\boxtimes}$ *and* $_{\square\boxtimes}$.

Proof. Follows directly from Lemma 14 and Lemma 15. $\square$

By contraposition, the previous lemma states that if a concept is not a fixpoint of $^{\square\boxtimes}$ or $_{\square\boxtimes}$, then it is unsteady, i.e. it is part of the interval $[\gamma_I(x_0), \mu_I(y_0)]$.

Since $_{\square\boxtimes}$ is an interior operator and $^{\square\boxtimes}$ is a closure operator on $\mathcal{B}(I)$ we have for each $c \in \mathcal{B}(I)$, $c_{\square\boxtimes} \leq c \leq c^{\square\boxtimes}$. Therefore, we can consider the closed interval $[c_{\square\boxtimes}, c^{\square\boxtimes}] \subseteq \mathcal{B}(I)$ and explore properties of concepts in it.

Lemma 24. *For any* $c \in \mathcal{B}(I)$*, each concept from* $[c_{\square\boxtimes}, c^{\square\boxtimes}] \setminus \{c\}$ *is steady.*

Proof. First we prove that either $c^{\square\boxtimes}$ equals c, or is its upper neighbor. Let $c = \langle A, B\rangle$. By definition, the intent of $c^{\square\boxtimes}$ is equal to $A^{\uparrow_J\downarrow_I\uparrow_I}$. By Lemma 1, $A^{\uparrow_J} \in \{B, B \setminus \{y_0\}\}$. Thus, $A^{\uparrow_J\downarrow_I\uparrow_I} \in \{B, B \setminus \{y_0\}\}$. If it equals B, then $c^{\square\boxtimes} = c$. Otherwise the intents of c and $c^{\square\boxtimes}$ differ in exactly one attribute, which makes c and $c^{\square\boxtimes}$ neighbors. Also notice that in this case $c^{\square\boxtimes}$ is steady because its intent does not contain y_0 (Lemma 2).

Now let $c' \leq c^{\square\boxtimes}$ be unsteady. If $c = c^{\square\boxtimes}$, then $c' \leq c$. If $c < c^{\square\boxtimes}$, then c is unsteady (Lemma 23) whereas $c^{\square\boxtimes}$ is steady. Unsteady concepts in $\mathcal{B}(I)$ form a closed interval (Lemma 14). Thus, $c' \vee c$ is unsteady and should be less than $c^{\square\boxtimes}$. Hence, $c' \vee c = c$ (c is a lower neighbor of $c^{\square\boxtimes}$), concluding $c' \leq c$ again.

In a similar way we obtain $c' \geq c$ for each unsteady $c' \geq c_{\square\boxtimes}$. $\square$

The following lemma shows an important property of sets of fixpoints of compound child operators in the unsteady initial sublattice. Namely, the set of fixpoints of $^{\square\boxtimes}$ is a lower set whereas the set of fixpoints of $_{\square\boxtimes}$ is an upper set.

Lemma 25. *Let* $c \in \mathcal{B}(I)$ *be an unsteady concept. If* c *is a fixpoint of* $^{\square\boxtimes}$*, then each* $c' \leq c$ *is also a fixpoint of* $^{\square\boxtimes}$*. If* c *is a fixpoint of* $_{\square\boxtimes}$*, then each* $c' \geq c$ *is also a fixpoint of* $_{\square\boxtimes}$.

Proof. Let $c = c^{\square\boxtimes}$ and $c' \leq c$. If c' is steady, then the assertion follows by Lemma 23. Suppose c' is unsteady. By extensivity and isotony of $^{\square\boxtimes}$, $c' \leq c'^{\square\boxtimes} \leq c^{\square\boxtimes} = c$. Thus, $c'^{\square\boxtimes}$ is unsteady (Lemma 2) and $c'^{\square\boxtimes} = c'$ by Lemma 24. The case $c = c_{\square\boxtimes}$ is dual. $\square$

The above results provide an interesting insight into the structure of our fixpoints. This helps us restrict possible cases that we need to take into consideration when designing Algorithm 2 which computes the lattice $\mathcal{B}(J)$, i.e. final concepts together with information about their ordering. The algorithm is more complicated than the previous one. We provide a short description of the algorithm, together with some examples. Note that for the sake of readability we will leave out most dual parts of similar cases as well as references to used lemmas.

The algorithm processes all unsteady initial concepts in a bottom-up direction using an arbitrary linear extension of the ordering on $\mathcal{B}(I)$, i.e. an ordering $\preceq$ such that if $c_1 \leq c_2$, then $c_1 \preceq c_2$. Each concept is either modified (by removing x_0 from the extent or y_0 from intent), or disposed of entirely. Sometimes, new concepts are created. All concepts also get updated lists of their upper and lower neighbors. Now, let $c = \langle A, B\rangle$ be an arbitrary unsteady concept from $\mathcal{B}(I)$ ($c \in [\gamma_I(x_0), \mu_I(y_0)]$).

If $c = c^{\Box\boxtimes}$, $c = c_{\Box\boxtimes}$, then c has two unsteady children $c_\Box \leq c^\Box$.

- The concept c will "split" into its two children. The concept $c_\Box = \langle A \setminus \{x_0\}, B\rangle$ will be a lower neighbor of the concept $c^\Box = \langle A, B \setminus \{y_0\}\rangle$.
- If for a lower neighbor c_l of c it holds $c_l = {c_l}^{\Box\boxtimes}$, $c_l \neq c_{l\Box\boxtimes}$, then it will be a lower neighbor of $c^\Box$. It is necessary to check whether $c_\Box$ and $c_{l\Box\boxtimes}$ will be neighbors. It certainly holds $c_{l\Box\boxtimes} \leq c_\Box$ but there can be a concept k, such that $c_{l\Box\boxtimes} \leq k \leq c_\Box$. Dually for upper neighbors.
- If for an unsteady neighbor c_n of c it holds $c_n = {c_n}^{\Box\boxtimes}$, $c_n = c_{n\Box\boxtimes}$, i.e. the same conditions as for c (c_n will split into $c_{n\Box}$, ${c_n}^\Box$), then $c_\Box$, $c_{n\Box}$ and $c^\Box$, ${c_n}^\Box$ will be neighbors.
- All other upper (resp. lower) neighbors will be neighbors of $c^\Box$ (resp. $c_\Box$).

If $c = c^{\Box\boxtimes}$ and $c \neq c_{\Box\boxtimes}$, then c has one unsteady child $c^\Box$.

- We have $c^\Box = \langle A, B \setminus \{y_0\}\rangle$, i.e. c loses y_0 from its intent.
- If for an upper neighbor c_u it holds $c_u = c_{u\Box\boxtimes}$, $c_u \neq {c_u}^{\Box\boxtimes}$ (c_u will lose x_0 from its extent), then c_u and d will become incomparable. It is necessary to check whether $c_{\Box\boxtimes}, c_u$ and $c, {c_u}^{\Box\boxtimes}$ should be neighbors (again, there can be a concept between them).

If $c \neq c^{\Box\boxtimes}$ and $c = c_{\Box\boxtimes}$, then c has one unsteady child $c_{\Box}$.

- We have $c_{\Box} = \langle A \setminus \{x_0\}, B \rangle$, i.e. c loses x_0 from its extent.

If $c \neq c^{\Box\boxtimes}$ and $c \neq c_{\Box\boxtimes}$, then c has no unsteady child and vanishes.

- It is necessary to check whether $c_{\Box\boxtimes}$ and $c^{\Box\boxtimes}$ should be neighbors (again, a concept can exist between them).
- Denote by U the set of all upper neighbors of c except for $c^{\Box\boxtimes}$ and similarly by L the set of all lower neighbors of c except for $c_{\Box\boxtimes}$. There are no fixpoints of $^{\Box\boxtimes}$ in U and there are no fixpoints of $_{\Box\boxtimes}$ in L. Therefore, $U, L \subseteq [\gamma_I(x_0), \mu_I(y_0)]$.
- Concepts from U and L will not be neighbors. They will either become incomparable or either one will vanish.
- It holds $\forall c_l \in L: \;\; c_l \leq c \leq c^{\Box\boxtimes}$, but it is necessary to check if there is a concept between them. Similarly, it holds $\forall c_u \in U : c_{\Box\boxtimes} \leq c \leq c_u$ but again, it is necessary to check if there is a concept between them.

Example 26. In Fig. 2.11-2.14, we can see some examples of transformations of unsteady concepts (depicted with dashed circles) from $\mathcal{B}(I)$ into concepts of $\mathcal{B}(J)$.

In Algorithm 2 we assume that following functions are already defined:

- $UpperNeighbors(c)$ - returns upper neighbors of c;
- $LowerNeighbors(c)$ - returns lower neighbors of c;
- $Link(c_1, c_2)$ - introduces neighborhood relationship between c_1 and c_2;
- $Unlink(c_1, c_2)$ - cancels neighborhood relationship between c_1 and c_2.

The number of iterations in TransformConceptLattice is at most $|\mathcal{B}(I)|$ which occurs when each initial concept is unsteady. In each iteration, tests $c = c^{\Box\boxtimes}$ and $c = c_{\Box\boxtimes}$ are performed and one of the procedures SplitConcept, RelinkReducedIntent, UnlinkVanishedConcept is called. It can be easily seen that the tests can be performed quite efficiently and do not add to the time complexity.

Algorithm 2 Transforming the lattice $\mathcal{B}(I)$ into the lattice $\mathcal{B}(J)$.

```
procedure LinkIfNeeded(c1, c2)
    if ∄k ∈ B(I) : c1 < k < c2 then
        Link(c1, c2);

procedure SplitConcept(c)
    d1 = c_□; d2 = c^□;
    Link(d1, d2);
    for all u ∈ UpperNeighbors(c) do
        Unlink(c, u); Link(d2, u);
    for all l ∈ LowerNeighbors(c) do
        Unlink(l, c); Link(l, d1);
    for all u ∈ UpperNeighbors(c) do
        if u ≠ u^□⊠ then
            Unlink(d2, u); Link(d1, u); LinkIfNeeded(d2, u^□⊠);
    for all l = ⟨C, D⟩ ∈ LowerNeighbors(c) do
        if y0 ∉ D then
            Unlink(l, d1); Link(l, d2); LinkIfNeeded(l_⊠□, d1);
    return d1, d2;

procedure RelinkReducedIntent(c)
    for all u = ⟨C, D⟩ ∈ UpperNeighbors(c) do
        if u ≠ u^□⊠ then
            Unlink(c, u);
            LinkIfNeeded(c_□⊠, u); LinkIfNeeded(c, u^□⊠);

procedure UnlinkVanishedConcept(c)
    for all u ∈ UpperNeighbors(c) do
        Unlink(c, u); LinkIfNeeded(c_□⊠, u);
    for all l ∈ LowerNeighbors(c) do
        Unlink(l, c);

procedure TransformConceptLattice(B(I))
    for all c = ⟨A, B⟩ ∈ [γ_I(x0), μ_I(y0)] from least to largest w.r.t. ⊑ do
        if c = c^□⊠ and c = c_□⊠ then                    ▷ Concept will split.
            B(I) ← B(I) \ {c};
            B(I) ← B(I) ∪ SplitConcept(c);
        else if c ≠ c^□⊠ and c = c_□⊠ then               ▷ Extent will be smaller.
            A ← A \ {x0};
        else if c = c^□⊠ and c ≠ c_□⊠ then               ▷ Intent will be smaller.
            RelinkReducedIntent(c);
            B ← B \ {y0};
        else if c ≠ c^□⊠ and c ≠ c_□⊠ then               ▷ Concept will vanish.
            B(I) ← B(I) \ {c};
            UnlinkVanishedConcept(c);
```

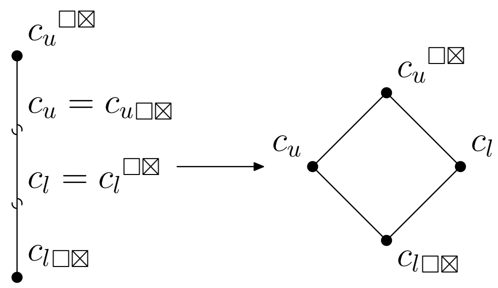


Figure 2.11: The concepts c_u, c_l become incomparable.

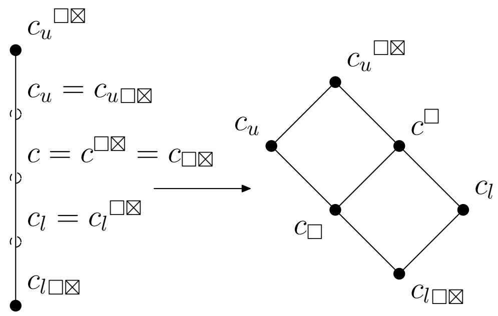


Figure 2.12: The concept c "splits" into its children.

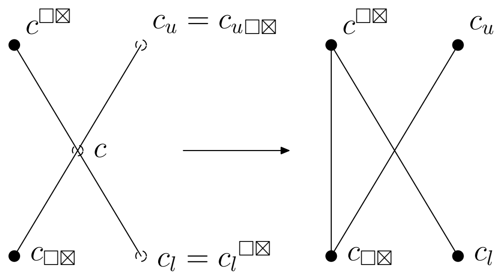


Figure 2.13: The concept c vanishes.

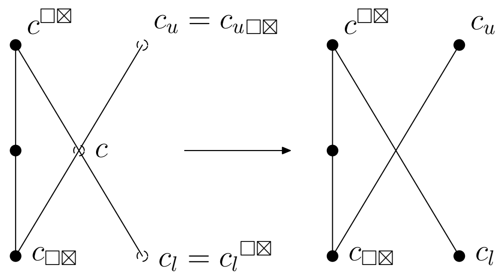


Figure 2.14: The concept c vanishes. There is already another concept between its children.

The most time consuming among the above three procedures is SplitConcept. It iterates through all upper (which can be bounded by $|X|$) and lower (which can be bounded by $|Y|$) neighbors of the concept c. For each of the neighbors it might be necessary to check if the interval between the neighbor and a certain other concept is empty (and we should make a new edge). This can be done by checking intents/extents of its neighbors.

The above considerations lead to the result that time complexity of Algorithm 2 is in the worst case $O(|\mathcal{B}(I)| \cdot |X|^2 \cdot |Y|)$.

Remark 27. We can apply several optimizations to Algorithm 2, however, we omit them in the description for the sake of readability. For example, for the test whether a concept is a fixed point of operator ${}^{\Box\boxtimes}$ (${}_{\Box\boxtimes}$) we actually do not need to compute the application of the operator and we can use some easier to compute, however equivalent, characterization. Another example is the check if there is a concept between a pair of comparable concepts. This check is provably searching for a steady concept in several cases (see Lemma 24). Taking this into consideration could speed up the search. Also note, in Algorithm 2, unsteady initial concepts are processed in an arbitrary linear extension of the initial ordering. That means that a concept will be processed only after all its lower neighbors are already processed. In general, it is not necessary to go through the initial concepts in this manner. One can easily design an algorithm, based on presented results, which will go through the concepts in any manner. We chose this processing order because it makes the description of the algorithm easier.

Example 28. An execution of Algorithm 2 on the concept lattice of the formal context from Fig. 2.10 is depicted in Fig. 2.15-2.18. Each picture captures the state after transformation of an unsteady concept. Unsteady concepts are drawn with dashed circles.

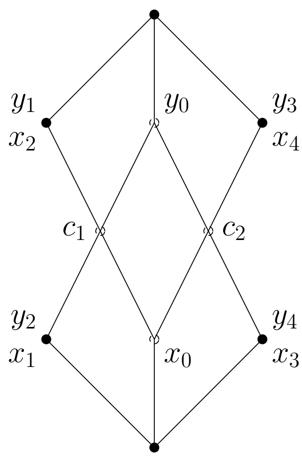


Figure 2.15: The initial state of $\mathcal{B}(I)$ for the formal context from Fig. 2.10. We see the unsteady interval $[\gamma_I(x_0), \mu_I(y_0)]$ containing four unsteady concepts $\gamma_I(x_0)$, c_1, c_2 and $\mu_I(y_0)$. Our algorithm starts at the bottom, so at first we transform the concept $\gamma_I(x_0)$. It holds $\gamma_I(x_0)_{\square\boxtimes} = \langle\emptyset, Y\rangle$ and $\gamma_I(x_0)^{\square\boxtimes} = \gamma_I(x_0)$ hence $\gamma_I(x_0)$ will lose y_0 from its intent. We also fix the neighborhood relation of $\gamma_I(x_0)$.

2.4 Experiments

We provide some insight into performance of our algorithms as well as some experimental evaluation. Comparing our algorithms to the traditional algorithms that recompute the whole final lattice does not make much sense as the difference proved to be immense in our preliminary experiments. This is caused

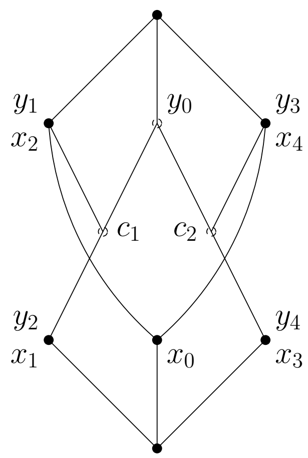


Figure 2.16: Now we transform the concept c_1. We have $c_{1\Box\boxtimes} = \gamma_I(x_1)$ and $c_1{}^{\Box\boxtimes} = \mu_I(y_1)$. Therefore, c_1 will vanish and we have to link steady concepts $\gamma_I(x_1)$ and $\mu_I(y_1)$ as well as $\gamma_I(x_1)$ and $\mu_I(y_0)$. We do not have to fix neighborhood relation of $\gamma_I(x_0)$ as it was fixed in the previous step.

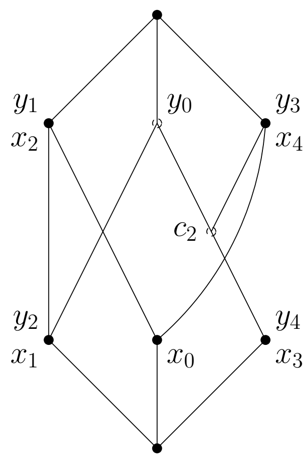


Figure 2.17: A mirror case of the previous step from Fig. 2.16. We transform concept c_2 which will in fact vanish.

by the obvious advantage of incremental methods as we usually need to recompute only a small portion of the initial lattice. Moreover, we can make use of previously computed concepts instead of just discarding them. Performance of our algorithms depends heavily on the size of unsteady initial interval. Hence, we provide experiments focusing on sizes of intervals corresponding to selected incidences. We used real world datasets as well as synthetic data. The former were taken from UC Irvine Machine Learning Repository[1] with an exception of dataset Drinks [23]. In order to obtain bivalent attributes we rescaled the

[1]http://archive.ics.uci.edu/ml/index.php

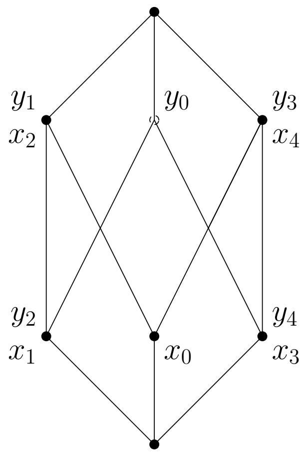


Figure 2.18: The transformation of the last unsteady concept $\mu_I(y_0)$. It holds $\mu_I(y_0)_{\square\boxtimes} = \mu_I(y_0)$ and $\mu_I(y_0)^{\square\boxtimes} = \langle X, \emptyset \rangle$ meaning that $\mu_I(y_0)$ will lose x_0 from its extent. The neighborhood relation was already fixed in previous steps.

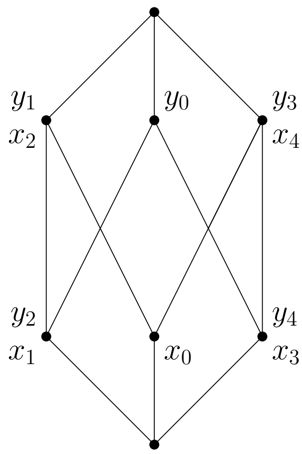


Figure 2.19: We transformed all unsteady concepts and arrived at the final concept lattice $\mathcal{B}(J)$.

attributes (as usual) using nominal scaling [8]. The details of used datasets can be found in Table 2.1.

For our experiments we selected a random incidence 10000 times and recorded the size of the corresponding interval. We provide maximal and average sizes as percentages of size of the whole concept lattice.

Remark 29. Evidently, not all intervals are determined by a single incidence but we are not interested in such intervals here.

	Objects	Attributes	Scaled Attrs.	Concepts
Mushrooms	8124	22	119	238710
Nursery	12960	8	32	183079
Post	90	9	25	1523
Zoo	101	17	28	379
Drinks	68	25	25	320

Table 2.1: Properties of datasets used in our experiments.

The results on real world datasets can be found in Table 2.2. We can see that for the larger datasets (Mushrooms and Nursery) the average size of the selected interval is well below 0.5%. For the smaller datasets, it is significantly larger although still within the 10%.

	Max size (%)	Avg size (%)
Mushrooms	4.46	0.46
Nursery	0.14	0.11
Post	8.27	3.92
Zoo	33.51	6.93
Drinks	51.56	8.83

Table 2.2: Sizes of intervals for real world data corresponding to randomly selected incidences.

The synthetic datasets were randomly generated with a fixed density (2%, 5%, 10%, 15%, 20%, 25%) and consisted of 500 objects and 100 attributes. The results can be seen in Table 2.3. All the recorded sizes, except for one, were withing 1%. Interestingly, both maximal and average size seems to be decreasing w.r.t. increasing density.

Remark 30. The results from Table 2.3 seem surprising as intuition might suggest that denser the context the larger the intervals corresponding to incidences. Indeed, consider an incidence $\langle x_0, y_0\rangle$ in a context $\langle X, Y, I\rangle$ where $\{x_0\}^{\uparrow_I} = Y$ and $\{y_0\}^{\downarrow_I} = X$. In such scenario we have $\mathcal{B}(I) = [\gamma_I(x_0), \mu_I(y_0)]$, i.e. the corresponding interval is equal to the whole concept lattice. Such case seems to be more probable in a dense context.

Density (%)	Max size (%)	Avg size (%)
2	1.73	0.47
5	0.86	0.25
10	0.73	0.18
15	0.75	0.15
20	0.72	0.14
25	0.78	0.14

Table 2.3: Sizes of intervals for synthetic data corresponding to randomly selected incidences.

The results suggest that an interval corresponding to a randomly selected incidence usually contains only a fraction of concepts w.r.t. the whole concept lattice. Recall, performance of our algorithms depends heavily on this size. Algorithm 1 can perform very well, especially if we take into consideration that the optimized version of the algorithm in fact computes just two derivation operators and two set comparisons in each iteration.

2.5 Extensions

This section is dedicated to investigation of various extensions of our method. In contrast with previous sections, we chose to approach this one in a less formal manner for the sake of readability. Instead of introducing more technical lemmas we usually provide more readable, although informal, descriptions. Nevertheless, every part of this section is based on firm foundations laid by the previous sections and given informal descriptions are sufficient for inferring corresponding formal forms.

First, we fix the set of objects X and the set of attributes Y. For the sake of readability we denote formal contexts of the form $\langle X, Y, I_i\rangle$ with just the name of the corresponding relation, e.g. we denote by I_1 a formal context $\langle X, Y, I_1\rangle$. If a formal context I_2 results from a formal context I_1 by removing a single incidence, we equip the child operators between such formal contexts with the greater index between the two formal contexts, e.g. ${}^{\Box_2}$, ${}^{\boxtimes_2}$, ${}_{\Box_2}$, ${}_{\boxtimes_2}$.

A natural extension of our method for removing an arbitrary number of incidences stems from repeated runs of the presented algorithms, i.e. removing incidences one by one. We obtain a sequence of formal contexts $I = I_0 \to I_1 \to I_2 \to \cdots \to I_n = J$ where $I_k = I_{k+1} \cup \{\langle x_{i_{k+1}}, y_{j_{k+1}}\rangle\}$. Evidently, there exist child operators between each pair of adjacent formal contexts and according to our extended notation we equip them with the greater index between the two formal contexts. We call initial (resp. final) concept unsteady if it is unsteady w.r.t. any removal step. To truly enlighten this, we provide a graphical representation in Fig. 2.20. By removing incidences one by one we are able to remove arbitrary number of incidences from any formal context.

$$\begin{tikzcd}[column sep=huge]
I = I_0 \arrow[r, bend left, "{\square_1,\, ^{\square_1}}"] & I_1 \arrow[l, bend left, "{\boxtimes_1,\, ^{\boxtimes_1}}"] \arrow[r, bend left, "{\square_2,\, ^{\square_2}}"] & I_2 \arrow[l, bend left, "{\boxtimes_2,\, ^{\boxtimes_2}}"] \arrow[r, bend left, "{\square_3,\, ^{\square_3}}"] & \cdots \arrow[l, bend left, "{\boxtimes_3,\, ^{\boxtimes_3}}"] \arrow[r, bend left, "{\square_n,\, ^{\square_n}}"] & I_n = J \arrow[l, bend left, "{\boxtimes_n,\, ^{\boxtimes_n}}"]
\end{tikzcd}$$

Figure 2.20: Removing n incidences one by one. We have $I_k = I_{k+1} \cup \{\langle x_{i_{k+1}}, y_{j_{k+1}}\rangle\}$.

Now, we take a closer look at compositions of child operators. At first, we focus on the simplest case, i.e. $I = I_0 \to I_1 \to I_2 = J$. In this scenario, we can calculate a composition of a child operator, e.g. $^{\square_1\square_2}$ or $_{\boxtimes_2\boxtimes_1}$, in a simple manner just by using the initial or final derivation operators as shown in the Lemma 31. Using the idea of this lemma repeatedly yields an easy way to calculate compositions of each type of child operator.

Lemma 31. *Let I_0, I_1, I_2 be formal contexts where $I_k = I_{k+1} \cup \{\langle x_{i_{k+1}}, y_{j_{k+1}}\rangle\}$. For $c \in \mathcal{B}(I_0), d \in \mathcal{B}(I_2)$ we have*

$$\begin{aligned}
c^{\square_1\square_2} &= \langle A^{\uparrow_{I_2}\downarrow_{I_2}}, A^{\uparrow_{I_2}}\rangle, & c_{\square_1\square_2} &= \langle B^{\downarrow_{I_2}}, B^{\downarrow_{I_2}\uparrow_{I_2}}\rangle,\\
d^{\boxtimes_2\boxtimes_1} &= \langle D^{\downarrow_{I_0}}, D^{\downarrow_{I_0}\uparrow_{I_0}}\rangle, & d_{\boxtimes_2\boxtimes_1} &= \langle C^{\uparrow_{I_0}\downarrow_{I_0}}, C^{\uparrow_{I_0}}\rangle.
\end{aligned}$$

Proof. We only show the part for $c^{\square_1\square_2}$ as the other cases are similar or dual. Let $c = \langle A, B\rangle \in \mathcal{B}(I_0)$. By the definition of child operator $^{\square}$ we have $c^{\square_1\square_2} = \langle A^{\uparrow_{I_1}\downarrow_{I_1}\uparrow_{I_2}\downarrow_{I_2}}, A^{\uparrow_{I_1}\downarrow_{I_1}\uparrow_{I_2}}\rangle$. It is sufficient to show that $A^{\uparrow_{I_1}\downarrow_{I_1}\uparrow_{I_2}} = A^{\uparrow_{I_2}}$. In the cases where either c or $c^{\square_1}$ is steady, the equation holds trivially. Suppose that both mentioned concepts are unsteady. Thus, we have $A^{\uparrow_{I_2}} = A^{\uparrow_{I_1}} \setminus \{y_{j_2}\} = A^{\uparrow_{I_1}\downarrow_{I_1}\uparrow_{I_1}} \setminus \{y_{j_2}\} = A^{\uparrow_{I_1}\downarrow_{I_1}\uparrow_{I_2}}$ by Lemma 1. $\square$

We already know that we can remove incidences one by one and now we show that it is also possible to remove an arbitrary number of incidences from an object in a single step (see Example 33). In fact, the presented method works practically as is by taking $I = J \cup \{\langle x_i, y_{j_1}\rangle, \langle x_i, y_{j_2}\rangle, \ldots, \langle x_i, y_{j_n}\rangle\}$ (J does not contain any of the incidences we are removing) and unsteady concepts to be from the union of all intervals determined by the removed incidences. To see this is indeed the case, consider removing the incidences one by one. We show that if an unsteady (w.r.t. any removal) concept c has two unsteady children, one of them will always remain steady (w.r.t. all consecutive removals) and the other can have at most one unsteady (w.r.t. any removal) child. This, together with already proven correctness of a single removal step shows that every concept of J can be computed using child operators.

Now, without loss of generality we can assume that $i = 0$ (we can freely reorder the objects) obtaining a sequence of formal contexts as in Fig. 2.20 where $I_k = I_{k+1} \cup \{\langle x_0, y_{j_{k+1}}\rangle\}$. Suppose that in k-th step there is an unsteady concept $c = \langle A, B\rangle$ with two unsteady children, namely $c_{\square_k} = \langle A \setminus \{x_0\}, B\rangle$ and $c^{\square_k} = \langle A, B \setminus \{y_{j_k}\}\rangle$. We notice immediately that $c_{\square_k}$ is steady and will remain steady in all consecutive steps due to not having x_0 in its extent. Now, the question is, if $c^{\square_k}$ can have two unsteady children in any of the consecutive steps. Without loss of generality (we can freely reorder the removals), we can assume unsteadiness of $c^{\square_k}$ in the next immediate step. For the sake of readability we put $(B \setminus \{y_{j_k}\}) = D$ and we show the steadiness of $(c^{\square_k})_{\square_{k+1}}$. By the definition, $(c^{\square_k})_{\square_{k+1}} = \langle D^{\downarrow_{k+1}}, D^{\downarrow_{k+1}\uparrow_{k+1}}\rangle$. Utilizing Lemma 1 we obtain $D^{\downarrow_{k+1}} = D^{\downarrow_k} \setminus \{x_0\} = A \setminus \{x_0\}$ and also $D^{\downarrow_{k+1}\uparrow_{k+1}} = (A \setminus \{x_0\})^{\uparrow_{k+1}} = D^{\downarrow_{k+1}\uparrow_k}$ showing $(c^{\square_k})_{\square_{k+1}} = c_{\square_k}$. Thus, $c^{\square_k}$ can only lose some attributes from its intent and, in borderline cases, it may vanish entirely. Lemma 31 ensures us that all the children may be computed by the same formulas as in the definition of child operators where we take $I = I_0$ as the initial and $J = I_n$ as the final formal context.

It is possible for more than one distinct unsteady concept from $\mathcal{B}(I_0)$ to be mapped (via composition of child operators) to a single unsteady concept $d \in \mathcal{B}(I_n)$. However, it can be easily seen that $d_{\boxtimes_n \cdots \boxtimes_1} = d^{\boxtimes_n \cdots \boxtimes_1}$ (see Remark 32) and so $d_{\boxtimes_n \cdots \boxtimes_1} = d^{\boxtimes_n \cdots \boxtimes_1} = c$ for exactly one concept $c \in \mathcal{B}(I_0)$. Thus, according to our method, d will be taken care of only while processing the concept c. Dually, we can show that it is possible to remove an arbitrary number of incidences from an attribute in a single step.

Remark 32. In the case of removing incidences from a single object, say $\langle x_0, y_{j_i}\rangle$ where $i \in \{1,\dots,n\}$, we obtain $d_{\boxtimes_n\dots\boxtimes_1} = d^{\boxtimes_n\dots\boxtimes_1}$ for $d = \langle C, D\rangle \in \mathcal{B}(I_n)$. If d remains steady, the equality is trivial. Supposed that d is not steady. By Lemma 31 we have $d_{\boxtimes_n\dots\boxtimes_1} = \langle C^{\uparrow_{I_0}\downarrow_{I_0}}, C^{\uparrow_{I_0}}\rangle$ and $d^{\boxtimes_n\dots\boxtimes_1} = \langle D^{\downarrow_{I_0}}, D^{\downarrow_{I_0}\uparrow_{I_0}}\rangle$. We show $D^{\downarrow_{I_0}} = C^{\uparrow_{I_0}\downarrow_{I_0}}$. First, we obtain $C = D^{\downarrow_{I_n}} \subseteq D^{\downarrow_{I_0}} \Rightarrow C^{\uparrow_{I_0}\downarrow_{I_0}} \subseteq D^{\downarrow_{I_0}}$. Second, it holds $D^{\downarrow_{I_0}} = D^{\downarrow_{I_n}} \cup \{x_0\} = C \cup \{x_0\} \subseteq C^{\uparrow_{I_0}\downarrow_{I_0}} \cup \{x_0\}$. However, it is easy to see that $x_0 \in C^{\uparrow_{I_0}\downarrow_{I_0}}$. This is because $x_0 \in D^{\downarrow_{I_n}} \subseteq D^{\downarrow_{I_0}}$ and $\bigcup_i^n\{y_{j_i}\} \subseteq \{x_0\}^{\uparrow_{I_0}}$ and also $C^{\uparrow_{I_0}} \subseteq \bigcup_i^n\{y_{j_i}\} \cup D$ together showing $x_0 \in (\bigcup_i^n\{y_{j_i}\} \cup D)^{\downarrow_{I_0}} \subseteq C^{\uparrow_{I_0}\downarrow_{I_0}}$.

Example 33. An example illustrating the removal of four marked incidences one by one is depicted in Fig. 2.22. The corresponding formal context can be found in Fig. 2.21. We obtain a sequence of formal contexts $I = I_0 \to I_1 \to I_2 \to I_3 \to I_4 = J$ where $I_k = I_{k+1} \cup \{\langle x_1, y_{k+1}\rangle\}$. It also illustrates the idea behind the removal of an arbitrary number of incidences from a single object (focus on the upper child of c_1).

I	y_1	y_2	y_3	y_4	y_5
x_1	•	•	•	•	×
x_2	×	×	×	×	×
x_3				×	×

Figure 2.21: The initial context with dots marking incidences to be removed.

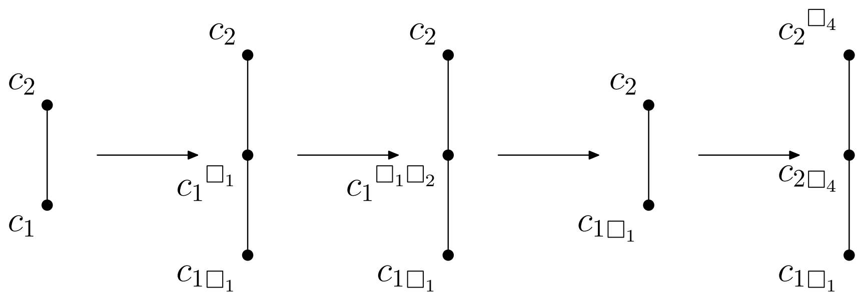


Figure 2.22: Concept lattices corresponding to formal contexts resulting from removing marked incidences from the formal context from Fig. 2.21, starting with the initial and ending with the final one where $c_1 = \langle\{x_1, x_2\}, Y\rangle$ and $c_2 = \langle X, \{y_4, y_5\}\rangle$.

We can go one step further and remove an arbitrary preconcept at once. This is now an easy extension of the case for removing an arbitrary number of incidences from a single object and its justification is very similar to it. Therefore we provide only a brief explanation. Consider removing incidences of a given preconcept in several steps such that in each step we remove all given incidences from a single object. Now, if in one of these steps an unsteady concept splits, i.e. it has two unsteady children, we can easily see that the upper child is steady and will remain steady due to not having any of the attributes corresponding to removed incidences in its intent. On the other hand, the lower child might still be unsteady w.r.t. any consecutive step. Nevertheless, it can never split as its upper child remains steady (see the previous paragraph for detailed explanation as this is practically its dual case).

Moving onwards, we consider removing arbitrary incidences at once based only on the concepts and derivation operators of the initial and final formal context. Take the formal context in Fig. 2.23. It is easy to derive the least and the greatest concept but there is no possibility to derive the intermediate concept using just our selected tools. Another way to easily see why this is impossible is to consider the formal context from Fig. 2.24. Evidently, given this formal context we would need to be able to derive 2^4 final concepts from a single initial concept.

I	y_1	y_2	y_3	y_4
x_1	$\times$	$\times$	$\times$	$\times$
x_2	$\times$	$\times$	$\bullet$	$\bullet$
x_3	$\times$	$\bullet$	$\bullet$	$\bullet$
x_4	$\times$	$\bullet$	$\bullet$	$\bullet$

Figure 2.23: It is not obvious how to efficiently derive the concept $\langle\{x_1, x_2\}, \{y_1, y_2\}\rangle$ from the concept $\langle X, Y\rangle$.

I	y_1	y_2	y_3	y_4
x_1	$\bullet$	$\times$	$\times$	$\times$
x_2	$\times$	$\bullet$	$\times$	$\times$
x_3	$\times$	$\times$	$\bullet$	$\times$
x_4	$\times$	$\times$	$\times$	$\bullet$

Figure 2.24: The concept lattice has to transform from the smallest possible into the largest.

Having found the limit for our extensions, we now turn our attention to the possibility of concurrently removing two–or more–incidences. In the case of corresponding intervals having no connection to each other (empty intersection and no neighborhood relation between any of the concepts), we can use presented algorithms concurrently without any modification. Lemma 34 provides a stronger condition, namely, a way to determine if corresponding intervals contain any comparable elements across them. In a more general case where the corresponding intervals have empty intersection (see Lemma 36) but there are connections between them, we would need to make some adjustments to the algorithms. Namely, we have to be careful with some special cases as can be seen in Example 37. In the most general case of non-empty intersection of corresponding intervals, we need to apply the transformations one by one in an arbitrary order for the elements from the intersection. Note that the intersection itself is also a closed interval (resp. sublattice). For Algorithm 2 we would need to make some adjustments in the way we are reconstructing the lattice structure, e.g. we can not assume the processing order which we used to make the description easier.

Lemma 34. *Let* $[c_1, c_2]$ *and* $[d_1, d_2]$ *be two closed intervals from* $\mathcal{B}(X, Y, I)$*. Then there exist comparable concepts* $c \in [c_1, c_2]$ *and* $d \in [d_1, d_2]$ *iff* $c_1 \leq d_2$ *or* $c_2 \leq d_1$*.*

Proof. Trivial, if there exist concepts $c \in [c_1, c_2]$ and $d \in [d_1, d_2]$ such that $c \leq d$ we get $c_1 \leq c \leq d \leq d_2$ and similarly for the dual case. The opposite direction is obvious. □

Remark 35. The exact condition for running our algorithms concurrently without any problem seems to be computationally more expensive and might not be worth using. Nevertheless, it might be useful to state it properly: unsteady intervals together with the set of the results of application of compound child operators to the concepts of the intervals should have an empty intersection. If this condition is fulfilled, we do not have to consider special cases like in Example 37.

Lemma 36. *Let* $c_1 = \langle A_1, B_1 \rangle \leq d_1 = \langle C_1, D_1 \rangle$ *and* $c_2 = \langle A_2, B_2 \rangle \leq d_2 = \langle C_2, D_2 \rangle \in \mathcal{B}(X, Y, I)$*. Then it holds*

$$[c_1, d_1] \cap [c_2, d_2] \neq \emptyset \text{ iff } \langle A_1 \cup A_2, D_1 \cup D_2 \rangle \text{ is a preconcept.}$$

Proof. "$\Rightarrow$": Suppose $[c_1, d_1] \cap [c_2, d_2] \neq \emptyset$, then there exists a concept $c = \langle A, D \rangle \in [c_1, d_1] \cap [c_2, d_2]$ and so $c_1 \leq c$ and $c_2 \leq c$. Thus $A_1 \subseteq A$ and $A_2 \subseteq A$ so $A_1 \cup A_2 \subseteq A$. Similarly, we obtain $D_1 \cup D_2 \subseteq D$ proving $\langle A_1 \cup A_2, D_1 \cup D_2 \rangle$ is preconcept.
"$\Leftarrow$": Assuming $\langle A_1 \cup A_2, D_1 \cup D_2 \rangle$ to be a preconcept implies $D_1 \cup D_2 \subseteq (A_1 \cup A_2)^{\uparrow_I}$ and so $\langle (A_1 \cup A_2)^{\uparrow_I \downarrow_I}, (A_1 \cup A_2)^{\uparrow_I} \rangle \in [c_1, d_1] \cap [c_2, d_2]$. □

Based on the ideas above we can now summarize what kind of changes to a formal context we are able to handle with our method.

- Removal of an arbitrary preconcept at once.
- Addition of an object (resp. attribute) can be achieved by adding full row (resp. column) to the underlying formal context. Evidently, this has no effect on the structure of the concept lattice (it just adds the object to all concepts). At last, we can remove unwanted incidences at once.
- Removal of an object (resp. attribute) can be done by removing all its incidences at once and afterwards removing the resulted empty row (resp. column). This is easy as we just have to check the greatest (resp. smallest) concept.
- Arbitrary change in an object (resp. attribute) is just a combination of the cases above (removal and addition).

Example 37. Let us have a formal context from Fig. 2.25. Consider removing incidences $\langle x_1, y_1\rangle$, $\langle x_2, y_3\rangle$ concurrently and take $I = I_1 \cup \langle x_1, y_1\rangle$, $I = I_2 \cup \langle x_2, y_3\rangle$, $J = I_1 \cap I_2$ where $\langle x_1, y_1\rangle \notin I_1$ and $\langle x_2, y_3\rangle \notin I_2$. The intervals determined by the incidences have empty intersection but there is a connection between them. The problem arises from concepts c_2 and c_3 transforming into the same concept. Fig. 2.26 shows concepts lattices $\mathcal{B}(I)$ and $\mathcal{B}(J)$.

I	y_1	y_2	y_3
x_1	$\bullet$	$\times$	
x_2	$\times$		$\bullet$

Figure 2.25: The initial context with dots marking incidences to be removed.

2.6 Discussion and related work

We analyzed changes in a concept lattice caused by removal of a single incidence from the associated formal context. We proved some theoretical results and presented two algorithms with time complexities $O(|\mathcal{B}| \cdot |X| \cdot |Y|)$ (Algorithm 1; without structure information) and $O(|\mathcal{B}| \cdot |X|^2 \cdot |Y|)$ (Algorithm 2; with structure information).

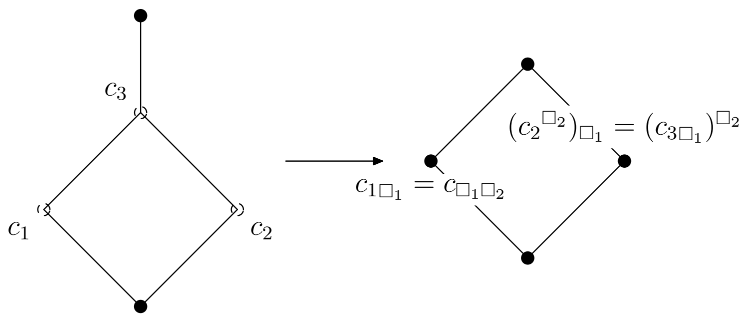


Figure 2.26: The initial and final concept lattice corresponding to the formal context from Fig. 2.25. The intervals $[c_1, c_3]$ and $[c_2, c_2]$ have empty intersection but there is a connection between them.

There exist several algorithms for incremental computation of concept lattice [14, 15, 16, 17, 18, 19] and they are usually based on adding/removing objects or attributes. Our approach is new in that we focus on the finer problem of recomputing a concept lattice after the removal of just one incidence. We believe that this problem is inherently rooted into every incremental algorithm for concept lattice construction. Amongst mentioned algorithms, the algorithm proposed by Nourine and Raynaud in [18] has the lowest time complexity of $O((|Y| + |X|) \cdot |X| \cdot |\mathcal{B}|)$. However, experiments presented in [15] indicate that this algorithm sometimes performs slower than some algorithms with time complexity $O(|\mathcal{B}| \cdot |X|^2 \cdot |Y|)$. In the case of our algorithms, presented experiments indicate that the size of the interval of unsteady concepts is usually relatively small, which substantially reduces the overall processing time of our algorithms.

We also looked into some possible extensions of our method and showed how it can be used to remove an arbitrary number of incidences from a single object (resp. attribute) at once. It turns out that we are also able to remove an arbitrary preconcept at once. Moreover, we are able to do it without any additional overhead. There is also the possibility of chaining applications of our algorithms providing a method to remove arbitrary incidences from a formal context. Utilizing these ideas we arrive at a general method for updating a concept lattice upon an arbitrary change in the underlying formal context.

The dual problem, adding an incidence to a formal context, does not share some nice properties as the problem of removing, e.g. the set of all unsteady final concepts has a more general structure than a closed interval and also not all unsteady initial concepts can be computed by the child operators.

Chapter 3

On sublattices and subrelations

One of basic theoretical results of FCA states a correspondence between closed subrelations of a formal context and complete sublattices of the corresponding concept lattice [8]. In this chapter, we study the related problem of constructing the closed subrelation for a complete sublattice generated by given set of elements.

A subrelation $J \subseteq I$ is called a *closed subrelation of* I [8] if each concept of $\langle X, Y, J \rangle$ is also a concept of $\langle X, Y, I \rangle$. There is the following correspondence between closed subrelations of I and complete sublattices of $\mathcal{B}(X, Y, I)$. For each closed subrelation $J \subseteq I$, $\mathcal{B}(X, Y, J)$ is a complete sublattice of $\mathcal{B}(X, Y, I)$, and for each complete sublattice $V \subseteq \mathcal{B}(X, Y, I)$ there exists a closed subrelation $J \subseteq I$ such that $V = \mathcal{B}(X, Y, J)$.

Throughout this chapter we consider a formal context $\langle X, Y, I \rangle$, its concept lattice $\mathcal{B}(X, Y, I)$, a set of concepts $P \subseteq \mathcal{B}(X, Y, I)$ and a complete sublattice $V \subseteq \mathcal{B}(X, Y, I)$ generated by the set P (i.e. $V = \mathrm{C}_{\bigvee\bigwedge} P$). Elements of P are called *generators*. We already know that there exists a closed subrelation $J \subseteq I$ with the concept lattice $\mathcal{B}(X, Y, J)$ equal to V. We show a method of constructing J without the need of constructing $\mathcal{B}(X, Y, I)$ first. We propose an efficient algorithm implementing the method and show illustrative examples and results of experiments.

We also investigate additional related problems. For a general subrelation $K \subseteq I$, we study the possibility of finding the least closed subrelation containing K. The problem does not always have a solution as the system of closed subrelations of I is not a closure system. We identify an important type of subrelations for which the solution always exists. We also provide some results on closed subrelations $J \supseteq K$ and their associated concept lattices.

From the investigation of subrelations a new type of formal rectangle arises. Such rectangles might serve as formalization of some notions from the field of cognitive psychology. We investigate properties of such formal rectangles and outline their relation to already known types. We also show that they are related to block relations. Lastly, we show how they can be structured into a lattice and we propose a basic theorem for lattices of such rectangles.

3.1 Closed subrelations for generated sublattices

Let us have a formal context $\langle X, Y, I\rangle$ and a subset P of its concept lattice. Denote by V the complete sublattice of $\mathcal{B}(X, Y, I)$ generated by P (i.e. $V = \mathrm{C}_{\bigvee\bigwedge} P$). Our goal is to find, without computing the lattice $\mathcal{B}(X, Y, I)$, the closed subrelation $J \subseteq I$ whose concept lattice $\mathcal{B}(X, Y, J)$ is equal to V.

If $\mathcal{B}(X, Y, I)$ is finite, V can be obtained by alternating applications of the closure operators $\mathrm{C}_{\bigvee}$ and $\mathrm{C}_{\bigwedge}$ on P: we set $V_1 = \mathrm{C}_{\bigvee} P$, $V_2 = \mathrm{C}_{\bigwedge} V_1$, …, and, generally

$$V_i \quad = \quad \begin{cases} \mathrm{C}_{\bigvee} V_{i-1} & \text{for odd } i, \\ \mathrm{C}_{\bigwedge} V_{i-1} & \text{for even } i. \end{cases} \tag{3.1}$$

The sets V_i are $\bigvee$-subsemilattices (for odd i) resp. $\bigwedge$-subsemilattices (for even i) of $\mathcal{B}(X, Y, I)$. Once $V_i = V_{i-1}$, we have the complete sublattice V. This situation is illustrated in Fig. 3.1 and Fig. 3.2.

$$\mathrm{C}_{\bigvee} P = V_1 \xrightarrow{\mathrm{C}_{\bigwedge}} V_2 \xrightarrow{\mathrm{C}_{\bigvee}} V_3 \overset{\dots}{\rightsquigarrow} V_{i-1} = V_i = V$$

Figure 3.1: One way to compute a (finite) complete sublattice generated by a set P stems from alternating computations of closures $\mathrm{C}_{\bigvee}$ and $\mathrm{C}_{\bigwedge}$ as given by (3.1).

Remark 38. Note that for infinite $\mathcal{B}(X, Y, I)$, V can be infinite even if P is finite. Indeed, denoting $FL(P)$ the free lattice generated by P [24, 25, 26] and setting $X = Y = FL(P)$, $I = \leq$ we have $FL(P) \subseteq V \subseteq \mathcal{B}(X, Y, I)$. $\mathcal{B}(X, Y, I)$ is the Dedekind-MacNeille completion of $FL(P)$ [8], we identify P

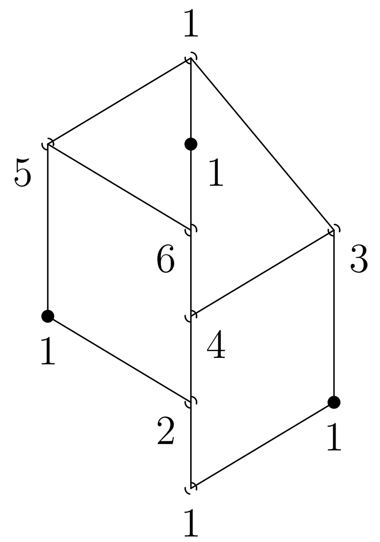


Figure 3.2: Computing the closures $C_{\bigvee}$ and $C_{\bigwedge}$ on a complete lattice. Bold dots mark the generators. Numbers mark the smallest i for which an element is contained in V_i.

with a subset of $FL(P)$, and P and $FL(P)$ with subsets of $\mathcal{B}(X, Y, I)$ as usual. Now, if $|P| > 2$, then $FL(P)$ is infinite [25], and so is V.

We always consider sets V_i together with the appropriate restriction of the ordering on $\mathcal{B}(X, Y, I)$. For each $i > 0$, V_i is a complete lattice that is order-embedded into $\mathcal{B}(X, Y, I)$ (but it is generally not a complete sublattice of $\mathcal{B}(X, Y, I)$).

In what follows, we construct formal contexts with concept lattices isomorphic to the complete lattices V_i, $i > 0$. By doing so, we obtain a sequence of formal context as shown in Fig. 3.3. We start by finding a formal context corresponding to the complete lattice V_1. Let $K_1 \subseteq P \times Y$ be given by

$$\langle\langle A, B\rangle, y\rangle \in K_1 \quad \text{iff} \quad y \in B. \tag{3.2}$$

As we can see, rows in the context $\langle P, Y, K_1\rangle$ are exactly intents of concepts from P.

$$\begin{tikzcd}
\mathrm{C}_{\bigvee} P = V_1 \arrow[r, "\mathrm{C}_{\bigwedge}"] \arrow[d, phantom, sloped, "\cong"] & V_2 \arrow[r, "\mathrm{C}_{\bigvee}"] \arrow[d, phantom, sloped, "\cong"] & V_3 \arrow[r, squiggly, "\cdots"] \arrow[d, phantom, sloped, "\cong"] & V_{i-1} = V_i = V \arrow[d, phantom, sloped, "\cong"] \\
\mathcal{B}(K_1) \arrow[r] \arrow[d, leftrightarrow] & \mathcal{B}(K_2) \arrow[r] \arrow[d, leftrightarrow] & \mathcal{B}(K_3) \arrow[r, squiggly, "\cdots"] \arrow[d, leftrightarrow] & \mathcal{B}(K_{i-1}) = \mathcal{B}(K_i) \arrow[d, leftrightarrow] \\
K_1 \arrow[r] & K_2 \arrow[r] & K_3 \arrow[r, squiggly, "\cdots"] & K_{i-1} = K_i
\end{tikzcd}$$

Figure 3.3: We compute a sequence of formal contexts K_i ($i > 0$) in order to obtain the closed subrelation with concept lattice equal to the complete sublattice generated by a set of concepts P.

Example 39. A basic example of a formal context $\langle X, Y, I\rangle$ (Fig. 3.4) and the corresponding formal context $\langle P, Y, K_1\rangle$ (Fig. 3.5) for $P = \{\langle\{x_1\}, \{y_1, y_2\}\rangle, \langle\{x_2\}, \{y_2, y_3\}\rangle\}$.

I	y_1	y_2	y_3
x_1	×	×	
x_2		×	×
x_3		×	

Figure 3.4: A formal context $\langle X, Y, I\rangle$. We take $P = \{\langle\{x_1\}, \{y_1, y_2\}\rangle, \langle\{x_2\}, \{y_2, y_3\}\rangle\}$.

K_1	y_1	y_2	y_3
c_1	×	×	
c_2		×	×

Figure 3.5: The corresponding formal context $\langle P, Y, K_1\rangle$ for Fig. 3.4.

Lemma 40. *The concept lattice $\mathcal{B}(P, Y, K_1)$ and the complete lattice V_1 are isomorphic. The isomorphism assigns to each concept $\langle B^{\downarrow_{K_1}}, B\rangle \in \mathcal{B}(P, Y, K_1)$ the concept $\langle B^{\downarrow_I}, B\rangle \in \mathcal{B}(X, Y, I)$.*

Proof. Concepts from V_1 are exactly those with intents equal to intersections of intents of concepts from P. The same holds for concepts from $\mathcal{B}(P, Y, K_1)$. $\square$

Next, we describe formal contexts for complete lattices V_i, $i > 1$. All of the contexts are of the form $\langle X, Y, K_i\rangle$, i.e. they have the set X as the set of objects and the set Y as the set of attributes (the relation K_1 is different in this regard). The relations K_i for $i > 1$ are defined in a recursive manner:

$$\text{for } i > 1,\ \langle x, y\rangle \in K_i \quad \text{iff} \quad \begin{cases} x \in \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I} & \text{for even } i, \\ y \in \{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}\uparrow_I} & \text{for odd } i. \end{cases} \tag{3.3}$$

Lemma 41. *For each $i > 1$,*

1. *$K_i \subseteq I$,*
2. *$K_i \subseteq K_{i+1}$.*

Proof. We prove both parts for even i; the assertions for odd i are proved similarly.

1. Let $\langle x, y\rangle \in K_i$. From $\{y\} \subseteq \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$ (which is true for both the special case $i-1 = 1$ and the case $i-1 > 1$) we get $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I} \subseteq \{y\}^{\downarrow_I}$. Thus, $x \in \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$ implies $x \in \{y\}^{\downarrow_I}$, which is equivalent to $\langle x, y\rangle \in I$.

2. As $K_i \subseteq I$, we have $\{x\}^{\uparrow_{K_i}\downarrow_{K_i}\uparrow_I} \supseteq \{x\}^{\uparrow_{K_i}\downarrow_{K_i}\uparrow_{K_i}} = \{x\}^{\uparrow_{K_i}}$. Thus, $y \in \{x\}^{\uparrow_{K_i}}$ yields $y \in \{x\}^{\uparrow_{K_i}\downarrow_{K_i}\uparrow_I}$. □

We can see that the definitions of K_i for even and odd $i > 1$ are dual. In what follows, we prove properties of K_i for even i and give the versions for odd i without proofs. Also, we always assume that i is a positive integer unless otherwise specified.

First, we show two basic properties of K_i that are equivalent to the definition. The first one says that K_i can be constructed as a union of some specific rectangles, the second one is a bit technical and it is used frequently in what follows. It shows how to construct K_i by individual columns (resp. rows).

Lemma 42. *Let $i > 1$.*

1. *If i is even, then $K_i = \bigsqcup_{y\in Y}\langle \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}, \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}\rangle$. If i is odd, then $K_i = \bigsqcup_{x\in X}\langle \{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}}, \{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}\uparrow_I}\rangle$.*
2. *If i is even, then for each $y \in Y$, $\{y\}^{\downarrow_{K_i}} = \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$. If i is odd, then for each $x \in X$, $\{x\}^{\uparrow_{K_i}} = \{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}\uparrow_I}$.*

Proof. We prove only the assertions for even i.

1. The "$\subseteq$" inclusion is evident. We now prove the converse inclusion. If $\langle x, y\rangle \in \bigsqcup_{y' \in Y}\langle \{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}, \{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}\rangle$, then there is $y' \in Y$ such that $x \in \{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$ and $y \in \{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$. The latter implies $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}} \subseteq \{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$, whence $\{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I} \subseteq \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$. Thus, x belongs to $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$ and by definition, $\langle x, y\rangle \in K_i$.

2. Follows directly from the obvious fact that $x \in \{y\}^{\downarrow_{K_i}}$ if and only if $\langle x, y\rangle \in K_i$. □

Remark 43. Informally, we can think of creating the sequence of formal contexts K_i as follows: for even i, we obtain the formal context K_i by stretching attribute intents of K_{i-1} over I; if i is odd, we stretch object extents of K_{i-1} over I. Another way of thinking about what happens at each iteration is the following: for even i we are fixing extents and for odd i we are fixing intents. Thinking in these informal terms could help with understanding the presented method.

A direct consequence of 2. of Lemma 42 is the fact that for every K_i we have either $\mathrm{Ext}(K_i) \subseteq \mathrm{Ext}(I)$ or $\mathrm{Int}(K_i) \subseteq \mathrm{Int}(I)$ as proven in the following lemma.

Lemma 44. *If i is even, then each extent of K_i is also an extent of I. If i is odd, then each intent of K_i is also an intent of I.*

Proof. Let i be even. 2. of Lemma 42 implies that each attribute extent of K_i is an extent of I. Thus, the lemma follows from the fact that each extent of K_i is an intersection of attribute extents of K_i.

The statement for odd i is proved similarly except for $i = 1$ where it follows by definition. □

There are several correspondences between the derivation operators of the contexts I and K_i and we provide some of them in the next lemma. Specifically, depending on the index i either attribute intents or object extents remain intact in K_{i+1}.

Lemma 45. *Let $i > 1$. If i is even, then for each $y \in Y$ it holds*

$$\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}} = \{y\}^{\downarrow_{K_i}\uparrow_{K_i}} = \{y\}^{\downarrow_{K_i}\uparrow_I}.$$

If i is odd, then for each $x \in X$ we have

$$\{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}} = \{x\}^{\uparrow_{K_i}\downarrow_{K_i}} = \{x\}^{\uparrow_{K_i}\downarrow_I}.$$

Proof. We prove the assertion for even i. By Lemma 44, $\{y\}^{\downarrow_{K_i}}$ is an extent of I. The corresponding intent is

$$\{y\}^{\downarrow_{K_i}\uparrow_I} = \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I\uparrow_I} = \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}} \tag{3.4}$$

(by Lemma 44, $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$ is an intent of I). Moreover, as $K_i \subseteq I$ (Lemma 41), we have

$$\{y\}^{\downarrow_{K_i}\uparrow_{K_i}} \subseteq \{y\}^{\downarrow_{K_i}\uparrow_I}. \tag{3.5}$$

We prove $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}} \subseteq \{y\}^{\downarrow_{K_i}\uparrow_{K_i}}$. Let $y' \in \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$. It holds

$$\{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}} \subseteq \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$$

($^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$ is a closure operator). Thus, $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I} \subseteq \{y'\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$ and so by 2. of Lemma 42, $\{y\}^{\downarrow_{K_i}} \subseteq \{y'\}^{\downarrow_{K_i}}$. Applying $^{\uparrow_{K_i}}$ to both sides we obtain $\{y'\}^{\downarrow_{K_i}\uparrow_{K_i}} \subseteq \{y\}^{\downarrow_{K_i}\uparrow_{K_i}}$, proving $y' \in \{y\}^{\downarrow_{K_i}\uparrow_{K_i}}$.

This, together with (3.4) and (3.5), proves the lemma. □

The following lemma helps us find isomorphisms between concept lattices of $\langle X, Y, K_i\rangle$ and subsemilattices V_i. It also allows us to draw correspondences between the derivation operators of K_i and I.

Lemma 46. *Let $i > 1$ be even. Then for each intent B of K_i it holds $B^{\downarrow_{K_i}} = B^{\downarrow_I}$. Moreover, if B is an attribute intent (i.e. there is $y \in Y$ such that $B = \{y\}^{\downarrow_{K_i}\uparrow_{K_i}}$), then $\langle B^{\downarrow_{K_i}}, B\rangle$ is a concept of I.*

If $i > 1$ is odd, then for each extent A of K_i it holds $A^{\uparrow_{K_i}} = A^{\uparrow_I}$. If A is an object extent (i.e. there is $x \in X$ such that $A = \{x\}^{\uparrow_{K_i}\downarrow_{K_i}}$), then $\langle A, A^{\uparrow_{K_i}}\rangle$ is a concept of I.

Proof. We prove the assertion for even i. Let B be an intent of K_i. It holds $B = \bigcup_{y \in B}\{y\}$ (obviously) and hence $B = \bigcup_{y \in B}\{y\}^{\downarrow_{K_i}\uparrow_{K_i}}$ (since ${}^{\downarrow_{K_i}\uparrow_{K_i}}$ is a closure operator). Therefore (2. of Lemma 42 and Lemma 45),

$$
\begin{aligned}
B^{\downarrow_{K_i}} &= \left(\bigcup_{y \in B}\{y\}\right)^{\downarrow_{K_i}} = \bigcap_{y \in B}\{y\}^{\downarrow_{K_i}} = \bigcap_{y \in B}\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I} \\
&= \left(\bigcup_{y \in B}\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}\right)^{\downarrow_I} = \left(\bigcup_{y \in B}\{y\}^{\downarrow_{K_i}\uparrow_{K_i}}\right)^{\downarrow_I} = B^{\downarrow_I},
\end{aligned}
$$

proving the first part.

Now let B be an attribute intent of K_i, $B = \{y\}^{\downarrow_{K_i}\uparrow_{K_i}}$. By the first already proven part we have $B^{\downarrow_I} = \{y\}^{\downarrow_{K_i}\uparrow_{K_i}\downarrow_I} = \{y\}^{\downarrow_{K_i}\uparrow_{K_i}\downarrow_{K_i}} = \{y\}^{\downarrow_{K_i}}$. By Lemma 45, $B^{\downarrow_I\uparrow_I} = \{y\}^{\downarrow_{K_i}\uparrow_I} = \{y\}^{\downarrow_{K_i}\uparrow_{K_i}} = B$. □

We already know that the sequence of formal contexts K_i ($i > 1$) is non-decreasing and now we show that for even i we have $\mathrm{Int}(K_{i-1}) \subseteq \mathrm{Int}(K_i)$ and for odd i we have $\mathrm{Ext}(K_{i-1}) \subseteq \mathrm{Ext}(K_i)$.

Lemma 47. *Let $i > 1$ be even. Then every intent of K_{i-1} is also an intent of K_i. If $i > 1$ is odd, then every extent of K_{i-1} is also an extent of K_i.*

Proof. Technically almost identical to the first part of the proof of Lemma 46, we show that for an intent B of K_{i-1} (i is even) we have $B^{\downarrow_{K_i}} = B^{\downarrow_I}$. The rest follows immediately from Lemma 41. □

Among other things, Lemma 46 (together with Lemma 47) shows that for even (resp. odd) $i > 1$ concepts of K_i are $\sqcup$-semiconcepts (resp. $\sqcap$-semiconcepts) of I. Indeed, for even i we have for an intent B of K_i the equality $B^{\downarrow_{K_i}} = B^{\downarrow_I}$ which means that $\langle B^{\downarrow_{K_i}}, B\rangle = \langle B^{\downarrow_I}, B\rangle$ is a $\sqcup$-semiconcept of I. Similarly for odd i.

Now we turn our attention to complete lattices V_i defined above. We have already shown in Lemma 40 that the complete lattice V_1 and the concept lattice $\mathcal{B}(P, Y, K_1)$ are isomorphic. Now we give a general result for $i > 0$.

Lemma 48. *For each $i > 0$, the concept lattice $\mathcal{B}(P, Y, K_i)$ (for $i = 1$) resp. $\mathcal{B}(X, Y, K_i)$ (for $i > 1$) and the complete lattice V_i are isomorphic. The isomor-*

phism is given by $\langle B^{\downarrow_{K_i}}, B\rangle \mapsto \langle B^{\downarrow_I}, B\rangle$ *if* i *is odd and by* $\langle A, A^{\uparrow_{K_i}}\rangle \mapsto \langle A, A^{\uparrow_I}\rangle$ *if* i *is even.*

Proof. We proceed by induction on i. The base step $i = 1$ has been already proven in Lemma 40. We will do the induction step for even i, the other case is dual.

As $V_i = \mathrm{C}_{\bigwedge} V_{i-1}$, we have to

1. show that the set $W = \{\langle A, A^{\uparrow_I}\rangle \mid A \text{ is an extent of } K_i\}$ is a subset of $\mathcal{B}(X, Y, I)$, containing V_{i-1} and
2. find for each $\langle A, A^{\uparrow_{K_i}}\rangle \in \mathcal{B}(X, Y, K_i)$ a set of concepts from V_{i-1} whose infimum in $\mathcal{B}(X, Y, I)$ has extent equal to A.

1. By Lemma 44, each extent of K_i is also an extent of I. Thus, $W \subseteq \mathcal{B}(X, Y, I)$. If $\langle A, B\rangle \in V_{i-1}$, then by the induction hypothesis B is an intent of K_{i-1} ($i - 1$ is odd). By Lemma 46 and Lemma 47, $B^{\downarrow_{K_i}} = B^{\downarrow_I} = A$ is an extent of K_i and so $\langle A, B\rangle \in W$.

2. Denote $B = A^{\uparrow_{K_i}}$. For each $y \in Y$, $\{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}$ is an intent of K_{i-1}. By Lemma 42 and the induction hypothesis,

$$\langle \{y\}^{\downarrow_{K_i}}, \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}\rangle = \langle \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}, \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}}\rangle \in V_{i-1}.$$

Now, the extent of the infimum (taken in $\mathcal{B}(X, Y, I)$) of these concepts for $y \in B$ is equal to $\bigcap_{y\in B}\{y\}^{\downarrow_{K_i}} = B^{\downarrow_{K_i}} = A$. □

If X and Y are finite, then 2. of Lemma 41 implies that there is a number $n > 1$ such that $K_{n+1} = K_n$. Denote this relation by J. According to Lemma 48, there are two isomorphisms of the concept lattice $\mathcal{B}(X, Y, J)$ and $V_n = V_{n+1} = V$. We will show that these two isomorphisms coincide and $\mathcal{B}(X, Y, J)$ is actually equal to V.

Lemma 49. $\mathcal{B}(X, Y, J) = V$*.*

Proof. Let $\langle A, B\rangle \in \mathcal{B}(X, Y, J)$. It suffices to show that $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$. As $J = K_{n+1} = K_n$, we have $J = K_i$ for some even i and also $J = K_i$ for some odd i. We can therefore apply both parts of Lemma 46 to J and get $A = B^{\downarrow_J} = B^{\downarrow_I}$ and $B = A^{\uparrow_J} = A^{\uparrow_I}$. □

Corollary 50. *The relation $J \subseteq I$ as defined above is a closed subrelation of I.*

Algorithm 3 uses our results to compute the subrelation J for given $\langle X, Y, I \rangle$ and P.

Algorithm 3 Computing the closed subrelation J.

Require: formal context $\langle X, Y, I \rangle$, subset $P \subseteq \mathcal{B}(X, Y, I)$
Ensure: the closed subrelation $J \subseteq I$ whose concept lattice is equal to $\mathrm{C}_{\bigvee\bigwedge} P$

```
Generate(K_1)                                  ▷ K_1 is given by (3.2)
procedure Generate(J)
    i ← 1
    repeat
        L ← J
        i ← i + 1
        if i is even then
            J ← {⟨x, y⟩ ∈ I | x ∈ {y}^{↓_L ↑_L ↓_I}}
        else
            J ← {⟨x, y⟩ ∈ I | y ∈ {x}^{↑_L ↓_L ↑_I}}
    until i > 2 & J = L
    return J
```

Lemma 51. *Algorithm 3 is correct and terminates after at most* $\max(|I|+1, 2)$ *iterations.*

Proof. Correctness follows from Lemma 49. The terminating condition ensures that we compare J and L only when they are both subrelations of the context $\langle X, Y, I \rangle$ (after the first iteration, L is a subrelation of $\langle P, Y, K_1 \rangle$ and the comparison would not make sense).

After each iteration, L holds the relation K_{i-1} and J holds K_i (3.3). Thus, except for the first iteration, we have $L \subseteq J$ before the algorithm enters the terminating condition (Lemma 41). As J is always a subrelation of I (Lemma 41), the number of iterations is not greater than $|I|+1$. The only exception is $I = \emptyset$. In this case, the algorithm terminates after 2 steps due to the first part of the terminating condition. □

Example 52. We now demonstrate execution of Alg. 3. Let $\langle X, Y, I \rangle$ be the formal context from Fig. 3.6 (left). The associated concept lattice $\mathcal{B}(X, Y, I)$

is depicted in Fig. 3.6 (right). Let $P = \{c_1, c_2, c_3\}$ where $c_1 = \langle\{x_1\}, \{y_1, y_4\}\rangle$, $c_2 = \langle\{x_1, x_2\}, \{y_1\}\rangle$, $c_3 = \langle\{x_2, x_5\}, \{y_2\}\rangle$ are concepts from $\mathcal{B}(X, Y, I)$. These concepts are depicted in Fig. 3.6 by filled dots.

I	y_1	y_2	y_3	y_4	y_5
x_1	×			×	
x_2	×	×	×		
x_3			×		×
x_4			×		
x_5		×			

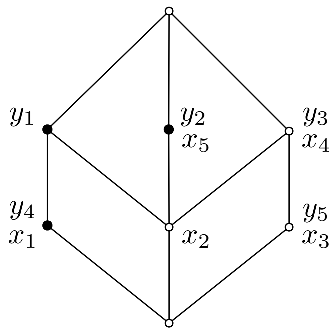


Figure 3.6: Formal context $\langle X, Y, I\rangle$ (left) and concept lattice $\mathcal{B}(X, Y, I)$ together with a subset $P \subseteq \mathcal{B}(X, Y, I)$, depicted by filled dots (right).

First, we construct the context $\langle P, Y, K_1\rangle$ (3.2). Rows in this context are intents of concepts from P (see Fig. 3.7, left). The concept lattice $\mathcal{B}(P, Y, K_1)$ (Fig. 3.7, center) is isomorphic to the $\bigvee$-subsemilattice $V_1 = \mathrm{C}_{\bigvee} P \subseteq \mathcal{B}(X, Y, I)$ (Fig. 3.7, right).

K_1	y_1	y_2	y_3	y_4	y_5
c_1	×			×	
c_2	×				
c_3		×			

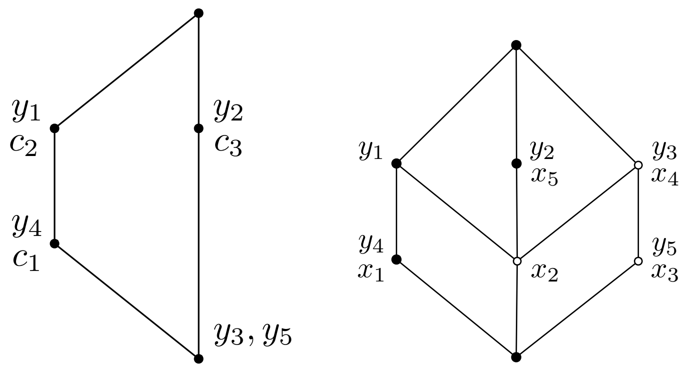


Figure 3.7: Formal context $\langle P, Y, K_1\rangle$ (left), the concept lattice $\mathcal{B}(P, Y, K_1)$ (center) and the $\bigvee$-subsemilattice $\mathrm{C}_{\bigvee} P \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(P, Y, K_1)$, depicted by filled dots (right).

It is easy to see that elements of $\mathcal{B}(P, Y, K_1)$ and corresponding elements of V_1 have the same intents.

Next step is to construct the subrelation $K_2 \subseteq I$. By (3.3), K_2 consists of elements $\langle x, y\rangle \in X \times Y$ satisfying $x \in \{y\}^{\downarrow_{K_1}\uparrow_{K_1}\downarrow_I}$. The concept lattice

$\mathcal{B}(X, Y, K_2)$ is isomorphic to the $\bigwedge$-subsemilattice $V_2 = \mathrm{C}_{\bigwedge} V_1 \subseteq \mathcal{B}(X, Y, I)$. K_2, $\mathcal{B}(X, Y, K_2)$, and V_2 are depicted in Fig. 3.8.

K_2	y_1	y_2	y_3	y_4	y_5
x_1	×			×	
x_2	×	×	•		
x_3			•		•
x_4			•		
x_5		×			

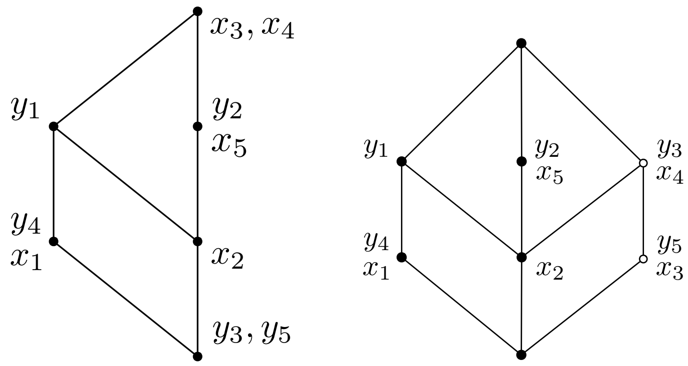


Figure 3.8: Formal context $\langle X, Y, K_2 \rangle$ (left), the concept lattice $\mathcal{B}(X, Y, K_2)$ (center) and the $\bigwedge$-subsemilattice $V_2 = \mathrm{C}_{\bigwedge} V_1 \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(X, Y, K_2)$, depicted by filled dots (right). Elements of $I \setminus K_2$ are depicted by dots in the table.

The subrelation $K_3 \subseteq I$ is computed again by (3.3). K_3 consists of elements $\langle x, y \rangle \in X \times Y$ satisfying $y \in \{x\}^{\uparrow_{K_2} \downarrow_{K_2} \uparrow_I}$. The result can be viewed in Fig. 3.9.

K_3	y_1	y_2	y_3	y_4	y_5
x_1	×			×	
x_2	×	×	×		
x_3			•		•
x_4			•		
x_5		×			

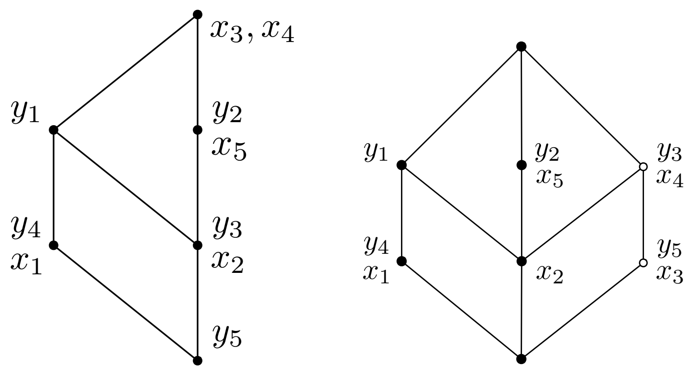


Figure 3.9: Formal context $\langle X, Y, K_3 \rangle$ (left), the concept lattice $\mathcal{B}(X, Y, K_3)$ (center) and the $\bigvee$-subsemilattice $V_3 = \mathrm{C}_{\bigvee} V_2 \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(X, Y, K_3)$, depicted by filled dots (right). Elements of $I \setminus K_3$ are depicted by dots in the table. As $K_3 = K_4 = J$, it is a closed subrelation of I and $V_4 = \mathrm{C}_{\bigwedge} V_3 = V_3$ is a complete sublattice of $\mathcal{B}(X, Y, I)$.

Notice that already $V_3 = V_2$ but $K_3 \neq K_2$. We cannot stop and have to perform another step. After computing K_4 we can easily check that $K_4 = K_3$. We thus obtained the desired closed subrelation $J \subseteq I$ and $V_4 = V_3$ is equal to the desired complete sublattice $V \subseteq \mathcal{B}(X, Y, I)$.

In our method, the relation K_1 differs from the other relations K_i for $i > 1$ in that it is a subset of $P \times Y$ instead of $X \times Y$. In the last part of this section, we present a modification of the method which replaces K_1 with a subrelation $K_1' \subseteq I$ given by

$$K_1' = \bigsqcup P, \tag{3.6}$$

where $\bigsqcup P$ is a union of rectangles determined by elements of P. The following lemma shows that after this replacement our method gives the same result as before.

Lemma 53. *Let $K_1' = \bigsqcup P$ and $L = \{\langle x, y\rangle \in X \times Y \mid x \in \{y\}^{\downarrow_{K_1'}\uparrow_{K_1'}\downarrow_I}\}$. Then $L = K_2$.*

Proof. By (3.3), it suffices to show that for each $y \in Y$, $\{y\}^{\downarrow_{K_1}\uparrow_{K_1}} = \{y\}^{\downarrow_{K_1'}\uparrow_{K_1'}}$. This is equivalent to saying that for each $y' \in Y$ the conditions

$$\{y\}^{\downarrow_{K_1}} \subseteq \{y'\}^{\downarrow_{K_1}} \tag{3.7}$$

and

$$\{y\}^{\downarrow_{K_1'}} \subseteq \{y'\}^{\downarrow_{K_1'}} \tag{3.8}$$

are equivalent. By the definition of K_1, (3.7) is satisfied iff $y \in B$ implies $y' \in B$ for each $\langle A, B\rangle \in P$.

Now, suppose (3.7) holds and let $x \in \{y\}^{\downarrow_{K_1'}}$. We have $\langle x, y\rangle \in K_1' = \bigsqcup P$. Therefore, there is a concept $\langle A, B\rangle \in P$ such that $x \in A$ and $y \in B$. By the assumption, we have also $y' \in B$, whence $\langle x, y'\rangle \in K_1'$, proving (3.8).

To prove the converse implication, suppose there is a concept $\langle A, B\rangle \in P$ satisfying $y \in B$. By (3.8), $A \subseteq \{y\}^{\downarrow_{K_1'}} \subseteq \{y'\}^{\downarrow_{K_1'}}$. Thus, $y' \in B$ and we have proved (3.7). $\square$

Note that the subrelation K_1' lacks the important property of the relation K_1 presented in Lemma 44. Namely, intents of K_1' need not be intents of I. Consequently, the concept lattice of K_1' does not have to be isomorphic to the complete lattice V_1, the property the relation K_1 has due to Lemma 48.

Remark 54. Even with the theoretical drawback of the relation K_1' it might be beneficial to use it while implementing our method because we obtain a unified type of all contexts involved, i.e. all with the same set of objects and attributes.

The above lemma allows for a slight modification of Alg. 3. We present it in Alg. 4. This algorithm uses the same procedure Generate as Alg. 3 but calls it with the relation K_1' instead of K_1.

Algorithm 4 Computing the closed subrelation J, alternative version.

Require: formal context $\langle X, Y, I\rangle$, subset $P \subseteq \mathcal{B}(X, Y, I)$

Ensure: the closed subrelation $J \subseteq I$ whose concept lattice is equal to $\mathrm{C}_{\bigvee\bigwedge} P$

  return Generate($\bigsqcup P$)

Example 55. Let $\langle X, Y, I\rangle$ and P be the same as in Example 52. As we can see, $K_1' = \bigsqcup P = K_2$. We have $\{x_2\}^{\uparrow_{K_1'}} = \{y_1, y_2\}$. This illustrates the mentioned fact that, in contrast to the subrelation K_1, intents of K_1' need not be intents of I.

The first iteration in Alg. 4 does not add any incidences to the built subrelation. However, as $i = 2$ at this stage, the algorithm does not stop and proceeds to the next iteration. From that, the execution continues exactly the same way as Alg. 3.

3.2 Experiments

Time complexity of Alg. 3 (and its variant Alg. 4) is clearly polynomial w.r.t. $|X|$ and $|Y|$. In Lemma 51 we proved that the number of iterations is less than or equal to $|I| + 1$. Our experiments indicate that this number might be much smaller in the practice. We used synthetic as well as real world datasets. More details about used datasets can be found in Section 2.4 and in Table 2.1.

The first batch of experiments involved real world datasets. The size of the set of generators P was given by percentage of corresponding number of concepts. For each size of P we randomly selected its elements 1000 times, ran our algorithm, and measured the number n of iterations, after which the algorithm terminated. The results for Mushrooms dataset can be seen in Table 3.1. For Nursery dataset, the results can be found in Table 3.2. Immediately, we can see that the average as well as maximal number of iterations seems to be decreasing w.r.t. increasing size of the set of generators. The peak in the number of iterations is in both cases achieved for a very small size of P, namely 0.003% and 0.005%. We recognized this trend in other datasets too.

$\lvert P\rvert$(%)	Max n	Avg n	$\lvert P\rvert$(%)	Max n	Avg n	$\lvert P\rvert$(%)	Max n	Avg n
0.003	16	7.81	0.040	8	4.57	0.40	6	3.4
0.005	12	7.48	0.050	7	4.48	0.45	5	3.30
0.010	10	6.13	0.10	6	4.44	0.5	5	3.33
0.015	10	5.8	0.15	6	4.28	1	4	3.25
0.020	12	5.23	0.20	6	4.01	2	4	3.22
0.025	8	4.91	0.25	6	3.95	3	4	3.11
0.030	8	4.69	0.30	6	3.62	4	4	2.99
0.035	8	4.57	0.35	6	3.6	5	4	2.89

Table 3.1: The results of experiments on Mushrooms dataset. The size of P is given by a percentage of the size of the concept lattice. There was no case where selected P generated the whole concept lattice.

$\lvert P\rvert$(%)	Max n	Avg n	$\lvert P\rvert$(%)	Max n	Avg n	$\lvert P\rvert$(%)	Max n	Avg n
0.003	10	3.17	0.040	4	3.03	0.40	3	3
0.005	20	7.32	0.050	3	3	0.45	3	2.99
0.010	10	5.41	0.10	3	3	0.5	3	3
0.015	7	4.28	0.15	3	3	1	3	2.99
0.020	6	3.68	0.20	3	3	2	3	2.99
0.025	5	3.27	0.25	3	3	3	3	2.99
0.030	5	3.14	0.30	3	3	4	3	2.98
0.035	4	3.01	0.35	3	3	5	3	2.91

Table 3.2: The results of experiments on Nursery dataset. The size of P is given by a percentage of the size of the concept lattice. Interestingly, we started to generate the whole concept lattice while selecting only 0.015% of its elements. By selecting 0.05% or more, we always generated the whole concept lattice.

Interestingly, while using Mushrooms dataset we never generated the whole concept lattice in our experiments. On the other hand, for Nursery dataset, we started generating it as soon as we hit 0.015% as the size of P and we always generated the whole concept lattice once the size of P was greater than or equal to 0.05%. This alludes to a question about the cause of this disparity. One possible answer is that the concept lattice of Mushrooms dataset has more intricate structure and more irreducible elements (we can not generate them from other elements).

As can be seen from the first batch of experiments on real world datasets, we usually obtain the highest number of iterations with a very small number of generators. Hence, for the rest of the experiments we chose to take fixed sizes of P, i.e. the size of P is no longer given as a percentage. Indeed, as we can see in Table 3.3 and 3.4 the peak occurs when we take $|P| \approx 10$.

$\lvert P \rvert$	Mushrooms		Nursery	
	Max n	Avg n	Max n	Avg n
3	9	3.39	6	2.27
4	18	4.77	8	2.59
5	17	6.34	9	3.36
6	18	7.51	14	4.37
7	14	7.27	12	4.78
8	19	7.77	16	5.99
10	12	7.63	18	7.45
15	12	6.84	10	5.94
20	10	6.32	8	5.18
25	11	6.2	7	4.56

Table 3.3: The results of experiments on larger real world datasets. The size of P is fixed, i.e. it is no longer given as a percentage. The whole lattice was generated only once, specifically for Nursery dataset and $|P| = 25$.

Lastly, we ran several experiments on synthetic datasets. Our synthetic contexts were randomly generated with fixed density and contained 500 objects and 100 attributes. For each density we generated 1000 formal contexts. For each such context and each fixed size of P we randomly selected generators 100 times and recorded the maximal and average number of iterations of our algorithm across all generated contexts. The results can be found in Table 3.5 and 3.6.

Investigating the results on synthetic data, we can observe the peak in both maximal and average number of iterations shifting from $|P| \approx 10$ to

$\|P\|$	Post		Zoo		Drinks	
	Max n	Avg n	Max n	Avg n	Max n	Avg n
3	8	2.81	8	2.92	8	3.09
4	16	3.77	11	3.68	12	3.6
5	24	5.40	13	4.20	9	3.90
6	28	7.35	12	4.46	9	4.15
7	25	8.76	12	4.68	10	4.15
8	24	9.31	13	4.94	9	4.13
10	28	9.51	11	5.06	9	4.11
15	18	7.78	11	4.85	10	4.06
20	15	6.82	9	4.72	8	3.75
25	13	6.42	11	4.43	8	3.53

Table 3.4: The results of experiments on smaller real world datasets. The size of P is fixed, i.e. it is no longer given as a percentage. In none of the cases P generated the whole concept lattice.

Density	5%			10%			20%		
$\|P\|$	Max n	Avg n	All	Max n	Avg n	All	Max n	Avg n	All
3	3	2.06	0	3	2.07	0	3	2.12	0
4	6	2.23	0	8	2.29	0	8	2.47	0
5	8	2.52	0	20	2.71	0.09	28	3.76	2.92
6	20	2.94	0.08	25	3.65	2.34	27	6.34	23.7
7	22	3.64	0.95	26	5.70	17.38	28	8.44	67.73
8	26	5.15	8.37	22	8.28	57.46	21	7.53	92.55
10	26	9.38	61.4	20	8.07	97.72	10	5.97	99.76
15	16	7.95	99.98	8	6.00	100	6	4.40	100
20	8	6.20	100	6	5.45	100	5	4.01	100
25	8	6.00	100	6	4.29	100	4	4.00	100

Table 3.5: The results of experiments on synthetic contexts with densities 5%, 10% and 20%. The column “All” specifies percentage of cases in which P generated the whole concept lattice.

$|P| \approx 6$ with increasing density of contexts. Interestingly, the peak in the average number of iterations seems slightly decreasing w.r.t. density, being 9.38 for $|P| = 10$ and 7.76 for $|P| = 6$.

Density	30%			40%			50%		
$\lvert P\rvert$	Max n	Avg n	All	Max n	Avg n	All	Max n	Avg n	All
3	3	2.16	0	3	2.19	0	18	2.23	0
4	29	2.75	0	37	3.17	0	30	3.62	0
5	28	5.26	7.6	37	6.11	6.7	35	6.78	3.75
6	28	7.78	31.39	28	7.91	25.61	40	7.76	15.12
7	26	7.78	64.69	27	7.24	50.97	22	6.86	34.95
8	20	6.54	85.13	22	6.14	73.64	20	5.98	56.49
10	9	5.35	98.81	12	4.97	95.32	10	4.80	88.05
15	6	4.03	100	5	4	99.99	8	4	99.97
20	4	4	100	4	3.99	100	4	3.98	100
25	4	3.99	100	4	3.93	100	4	3.70	100

Table 3.6: The results of experiments on synthetic contexts with densities 30%, 40% and 50%. The column "All" specifies percentage of cases in which P generated the whole concept lattice.

From the results we can see that the average number of iterations is very small compared to the number of objects and attributes. The maximal number of iterations is also small compared to the size of corresponding context. Both values usually peak for very small sizes of P and there is also an apparent decreasing trend for number of iterations for increasing size of P. Also, we observed the highest number of iterations being achieved for fixed sizes of P, no matter the size of the concept lattice.

The small average number of iterations for both synthetic and real world datasets indicates good performance for our algorithm as it just computes the closures of objects (resp. attributes) in its iterations.

3.3 The question of complexity

We already showed in Lemma 51 the obvious upper bound for the number of iterations of Algorithm 3. However, we believe this bound is loose and can be tightened. Our belief is that the number of iterations is in fact bounded by the sum of objects and attributes, i.e. $O(|X| + |Y|)$, however, the proof of this claim is a matter of future research. Nevertheless, we provide some examples of contexts containing certain patterns that can lead to a relatively high number of iterations together with some insight into those patterns. We focus on the finite cases in this section.

For now, let us think about the iterations as closures on the subsemilattices instead of operations over formal contexts. Since our goal is to bound the number of iterations by the sum of objects and attributes, we focus on $\bigvee$-irreducible and $\bigwedge$-irreducible elements (we are working with finite cases here). It is easy to see that every such closure has to add a new element otherwise the next immediate closure would start with the exact same set of elements as before, therefore it would be already closed w.r.t. this closure and the algorithm would terminate. Indeed, in the worst case scenario, we add an element with each closure and at the same time, it should be in such a way that minimizes sizes of $\bigvee$-irreducible and $\bigwedge$-irreducible sets. Let us have $V_i = \mathrm{C}_{\bigwedge} V_{i-1}$, making i even and let $i > 2$. It is easy to see that each element added in such iteration would be $\bigvee$-irreducible in V_i, otherwise it would have been added in the previous iteration. The problem now is that such concepts can make some previously $\bigvee$-irreducible elements $\bigvee$-reducible. The case for odd i is dual. Check Example 57 for a concrete example.

Remark 56. A tempting idea for showing a bound of the number of iterations of our algorithms is to consider free lattices [24, 25, 26]. However, this idea just strengthens our belief in the bound $O(|X| + |Y|)$. This is due to so called Whitman condition [24] and its consequences. One such consequence states that every element of a lattice satisfying Whitman condition (satisfied by free lattices) is either $\bigvee$-irreducible or $\bigwedge$-irreducible and we know that the number of such elements is a lower bound for the size of the underlying formal context.

Example 57. In Fig. 3.10 we can see a lattice with marked set of generators P and $\mathrm{C}_{\bigvee\bigwedge} P$ is equal to the whole lattice. Computing closures yields subsemilattices V_i and numbers next to the elements indicates the least index i for which V_i contains the corresponding element. Notice how previously $\bigvee$-irreducible or $\bigwedge$-irreducible elements are changed into reducible ones by adding new elements.

Example 58. In Fig. 3.11-3.16 we can see a formal context with marked set of generators P and first several iterations of our algorithm on such formal context. The number of iterations in this case is 10. In each step (except for

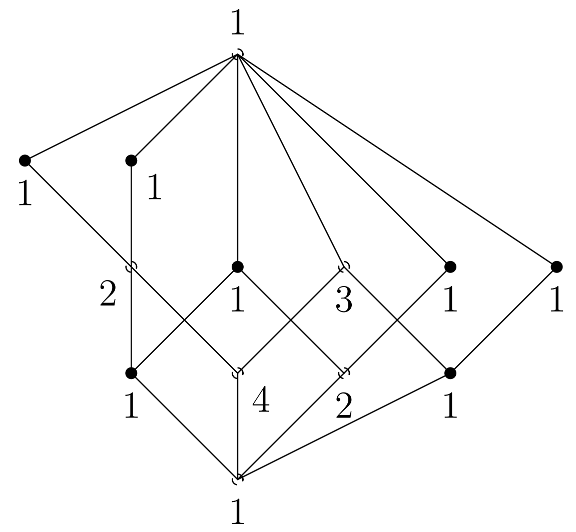


Figure 3.10: A lattice where for each element e, the numeric label marks the least index i such that $e \in V_i$. Generators are marked with bold dots.

the final ones) we generate a new concept. However, depending on the step number, such concept does not posses the correct extent or intent (w.r.t. the initial formal context). The following iteration fixes that, however, by doing so generates another new concept with the same issue.

Example 59. The formal context in Fig. 3.18 enlarges the formal context from Fig. 3.11. There is a clear pattern that can be further extended in a similar fashion. Let X be the set of objects and Y the set of attributes. Taking marked concepts as the set of generators, the number of iterations for such formal contexts can be calculated as $2|Y| - 5$. Since $|X| = |Y| + 1$, we obtain the following inequality $2|Y| - 5 < 2|Y| + 1$ showing that the number of iterations in such contexts is always strictly less than the sum of objects and attributes.

In Fig. 3.19 we can see a formal context extending the pattern from Example 58 with more attributes and we can extend it further in this fashion. Now, consider enlarging our pattern (same as Fig. 3.18), by letting selected object concepts of x_1, x_2, x_3 and x_4 to have not five, but ten attributes in their intents (we also need to add ten more object to accommodate our pattern). In this scenario, the number of iterations is 72 which is strictly greater than the sum of objects and attributes (in this case 66).

I	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1					×	×	×
x_2		×	×	×			
x_3	×	×	×	×	×	×	
x_4	×		×	×	×	×	
x_5	×		×	×	×		
x_6	×			×	×		
x_7	×			×			
x_8	×						

Figure 3.11: The initial formal context. Elements of P, i.e. generators, are marked with rectangles.

K_1	y_1	y_2	y_3	y_4	y_5	y_6	y_7
					×	×	×
		×	×	×			
	×						

Figure 3.12: The formal context K_1, its objects are elements of P. Rows are their intents.

K_2	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1					×	×	×
x_2		×	×	×			
x_3	×	×	×	×	•	•	
x_4	×						
x_5	×						
x_6	×						
x_7	×						
x_8	×						

Figure 3.13: The second iteration generates a new concept, namely, the object concept of x_3. However, its intent is not correct.

K_3	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1					×	×	×
x_2		×	×	×			
x_3	×	×	×	×	×	×	
x_4	×		•	•	•	•	
x_5	×						
x_6	×						
x_7	×						
x_8	×						

Figure 3.14: Fixing the object concept of x_3 generates additional concept, specifically, attribute concept of y_6 (although it does not posses correct extent).

K_4	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1					×	×	×
x_2		×	×	×			
x_3	×	×	×	×	×	×	
x_4	×		•	•	×	×	
x_5	×						
x_6	×						
x_7	×						
x_8	×						

Figure 3.15: A similar situation as for the second iteration but for object concept of x_4.

K_5	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1					×	×	×
x_2		×	×	×			
x_3	×	×	×	×	×	×	
x_4	×		×	×	×	×	
x_5	×		•	•			
x_6	×						
x_7	×						
x_8	×						

Figure 3.16: Now, the following iterations should already be obvious.

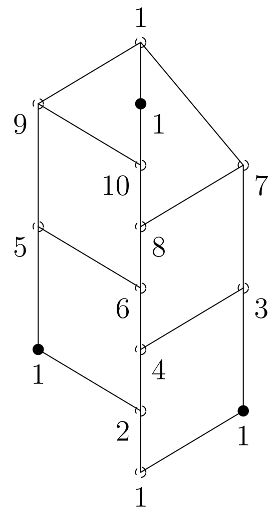


Figure 3.17: A concept lattice corresponding to the formal context from Fig. 3.11. Numbers mark iterations in which concepts appear for the first time (might be prior the fix of its extent/intent). The generators are marked with bold dots.

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}
x_1							×	×	×	×	×
x_2		×	×	×	×	×					
x_3	×	×	×	×	×	×	×	×	×	×	
x_4	×		×	×	×	×	×	×	×	×	
x_5	×		×	×	×	×	×	×	×		
x_6	×			×	×	×	×	×	×		
x_7	×			×	×	×	×	×			
x_8	×				×	×	×	×			
x_9	×				×	×	×				
x_{10}	×					×	×				
x_{11}	×					×					
x_{12}	×										

Figure 3.18: A formal context extending the pattern from Fig. 3.11.

3.4 Closed subrelations containing arbitrary relation

Procedure Generate from Algorithm 3 accepts any subrelation of I as its parameter so it does make sense to investigate what output it generates upon receiving an arbitrary subrelation of I. Interestingly, the next lemma shows that no matter what subrelation the procedure is called with, it always outputs

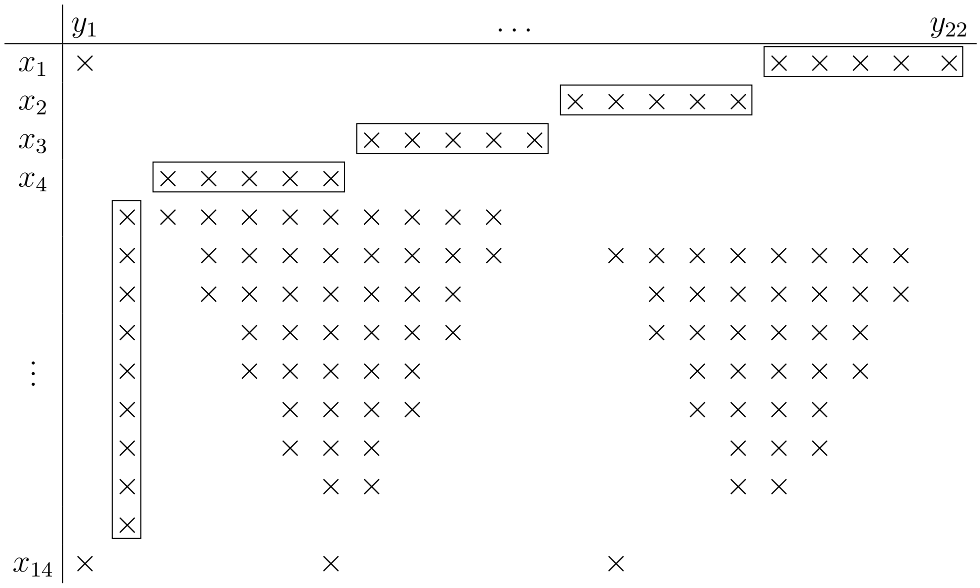


Figure 3.19: A formal context further extending the pattern from Fig. 3.18.

a closed subrelation of I. This may come as a surprise due to the fact that closed subrelations do not form a closure system.

Lemma 60. *For any subrelation $K \subseteq I$, the subrelation* Generate(K) *is a closed subrelation of I.*

Proof. For the relation $L = \{\langle x, y\rangle \in X \times Y \mid x \in \{y\}^{\downarrow_K \uparrow_K \downarrow_I}\}$, the attribute extent $\{y\}^{\downarrow_L}$ of any attribute y is equal to $\{y\}^{\downarrow_K \uparrow_K \downarrow_I}$ and hence is an extent of I. Therefore, each extent of L is, as an intersection of attribute extents, an extent of I (we have already used this argument before).

Now, for the subset P of the dual concept lattice $\mathcal{B}(Y, X, I^{-1})$ consisting of concepts whose intents are equal to extents of L, Alg. 3 outputs a closed subrelation of I^{-1}. This subrelation is evidently equal to the inverse of the subrelation Generate(K). □

The above result alludes to some questions about closed subrelations containing a given subrelation $K \subseteq I$. Namely, 1. whether there is the least such subrelation, and 2. what can be said about these subrelations and their concept lattices in general.

As mentioned in [8], the intersection of a system of closed subrelations needs not be a closed subrelation. Therefore, the system of all closed subrelations of I

is not a closure system and, consequently, there does not exist a closure operator assigning to each subrelation $K \subseteq I$ the least greater closed subrelation. Thus, the answer to the first question is, at least in general, negative. However, in what follows, we show that for some important type of subrelations of I (which we have already met) the answer is positive. Regarding the second question, we provide some basic results.

Definition 61 (block relation). A super-relation $L \supseteq I$ is called a *block relation of* I if for each $x \in X$ and $y \in Y$, $\{x\}^{\uparrow_L}$ is an intent and $\{y\}^{\downarrow_L}$ is an extent of I.

Definition 62 (semi-closed subrelation). We call a subrelation $L \subseteq I$ *semi-closed* if for each $x \in X$ and $y \in Y$, $\{x\}^{\uparrow_L}$ is an intent and $\{y\}^{\downarrow_L}$ is an extent of I.

The definition of semi-closed subrelation resembles closely the definition of a block relation [8, 27]. The only difference is that semi-closed relations are subrelations whereas block relations are super-relations of I. Semi-closed subrelations share many properties with block relations. We summarize some of them below. We omit proofs as they are technically exactly the same as proofs of corresponding properties of block relations which can be found in [8, 27, 28].

Remark 63. Despite what the name "semi-closed" might suggest, concepts of such subrelations are not generally semiconcepts of the initial relation. Indeed, check Example 76 for a counterexample. The name was chosen so it emphasizes the fact that such subrelations are in a sense closed as you can see in the following.

Lemma 64. *$L \subseteq I$ is a semi-closed subrelation iff each intent of L is also an intent of I and each extent of L is also an extent of I.*

Lemma 65. *The system of all semi-closed subrelations of I forms a closure system in I.*

We denote by $\mathrm{C}_{\mathcal{IV}}$ the closure operator associated with the closure system of all semi-closed subrelations of I. Algorithm 5 computes the value of $\mathrm{C}_{\mathcal{IV}}$ for any subrelation $L \subseteq I$. The same algorithm for block relations has been shown in [28].

Algorithm 5 Computing $\mathrm{C}_{\mathcal{IV}}$

Require: subrelation $L \subseteq I$
Ensure: $\mathrm{C}_{\mathcal{IV}}L$

$i \leftarrow 1$
repeat
  $L' \leftarrow L$
  $i \leftarrow i+1$
  if i is even **then**
    $L \leftarrow \{\langle x,y\rangle \in X \times Y \mid x \in \{y\}^{\downarrow_{L'}\uparrow_I\downarrow_I}\}$
  else
    $L \leftarrow \{\langle x,y\rangle \in X \times Y \mid y \in \{x\}^{\uparrow_{L'}\downarrow_I\uparrow_I}\}$
until $i > 2$ & $L = L'$
return L

Obviously, every closed subrelation of I is also semi-closed. Thus, we obtain a first property of closed subrelations containing K.

Lemma 66. *If J is a closed subrelation of I and $K \subseteq J$, then $\mathrm{C}_{\mathcal{IV}}K \subseteq J$.*

Proof. Follows from basic properties of closure operators and from the fact that J is semi-closed: $K \subseteq J$ implies $\mathrm{C}_{\mathcal{IV}}K \subseteq \mathrm{C}_{\mathcal{IV}}J = J$. □

The following lemma shows that despite the fact that closed subrelations do not form a closure system, in some important cases, there exists the least closed subrelation containing K.

Lemma 67. *Let $K = \bigsqcup P$ for some $P \subseteq \mathcal{B}(X,Y,I)$. Then $\mathrm{C}_{\mathcal{IV}}K$ is a closed subrelation of I and the concept lattice of $\mathrm{C}_{\mathcal{IV}}K$ is equal to $\mathrm{C}_{\bigvee\bigwedge}P$. $\mathrm{C}_{\mathcal{IV}}K$ can be computed by procedure* Generate *of Alg. 3.*

Proof. We show that for each $y \in Y$, $\{y\}^{\downarrow_K\uparrow_K} = \{y\}^{\downarrow_K\uparrow_I}$. Denote by P_y the set of all concepts $\langle A,B\rangle \in P$ such that $y \in B$. We have $\{y\}^{\downarrow_K} = \bigcup_{\langle A,B\rangle\in P_y} A$.

Thus (unions and intersections are always taken over all $\langle A, B\rangle \in P_y$),

$$\{y\}^{\downarrow_K\uparrow_K} = \Big(\bigcup A\Big)^{\uparrow_K} = \bigcap A^{\uparrow_K} = \bigcap A^{\uparrow_I} = \Big(\bigcup A\Big)^{\uparrow_I} = \{y\}^{\downarrow_K\uparrow_I}.$$

Therefore, the first iteration in Generate and Alg. 5 are identical.

Now, let K_1 be the same as in Sec. 3.1, i.e. given by (3.2). According to Lemma 53, the closed subrelation J for $\mathrm{C}_{\bigvee\bigwedge}P$ can be computed by means of Alg. 3. By Lemma 45 and Lemma 42, for each $i > 1$, K_i is equal to the union of rectangles given by some concepts of I. Thus, the above finding on K applies to all subrelations K_i, $i > 1$, and all iterations in Generate and Alg. 5 are identical. This proves that $J = \mathrm{C}_{\mathcal{IV}}K$. The rest follows from Lemma 49. □

Remark 68. Due to the previous lemma, it is easy to see that for a $L \subseteq I$ such that each concept of L is a protoconcept of I there exists the closure to the least closed subrelation. This is due to the fact that every protoconcept of I uniquely determines a concept of I. Every closed subrelation containing L has to contain these concepts and the rest follows immediately from Lemma 67.

For each concept $\langle A, B\rangle$ of a semi-closed subrelation of I, A is an extent and B an intent of I (there need not be any special relationship between A and B, i.e., in general, $B \neq A^{\uparrow_I}$ and $A \neq B^{\downarrow_I}$). We call such preconcepts *interval-preconcepts* and denote the set of all interval-preconcepts of I by $\mathcal{IV}(X, Y, I)$.

Remark 69. The name *interval-preconcepts* was chosen because it shows two defining properties of such rectangles: 1. they are preconcepts, 2. they uniquely determine an interval in the corresponding concept lattice.

Interval-preconcepts are ordered the same way as preconcepts [11, 12], i.e. for each $\langle A, B\rangle, \langle C, D\rangle \in \mathcal{IV}(X, Y, I)$ we have $\langle A, B\rangle \le \langle C, D\rangle$ iff $A \subseteq C$ and $D \subseteq B$.

Lemma 70. *$\mathcal{IV}(X, Y, I)$ together with the above ordering is a complete lattice with infima and suprema given by the same formulas as in* (1.1) *and* (1.2).

Proof. For infima: For any system $\langle A_\iota, B_\iota\rangle$, $\iota \in \mathcal{I}$, of interval-preconcepts, the tuple $\left\langle \bigcap_{\iota\in\mathcal{I}} A_\iota, (\bigcup_{\iota\in\mathcal{I}} B_\iota)^{\downarrow_I\uparrow_I}\right\rangle$ is evidently an interval-preconcept as well. Let $\langle C, D\rangle$ be another interval-preconcept, less than or equal to $\langle A_\iota, B_\iota\rangle$ for each

$\iota \in \mathcal{I}$. We have $C \subseteq A_\iota$ for each $\iota \in \mathcal{I}$, whence $C \subseteq \bigcap_{\iota\in\mathcal{I}} A_\iota$. Similarly, $D \supseteq \bigcup_{\iota\in\mathcal{I}} B_\iota$. Since D is an intent, it also holds $D \supseteq (\bigcup_{\iota\in\mathcal{I}} B_\iota)^{\downarrow_I\uparrow_I}$ and the proof is finished. The proof for suprema is similar. $\square$

For any semi-closed subrelation L of I, the concept lattice $\mathcal{B}(X,Y,L)$ is a subset of $\mathcal{IV}(X,Y,I)$. The following lemma characterizes all such subsets.

Lemma 71. *1. For any semi-closed subrelation $L \subseteq I$, $\mathcal{B}(X,Y,L)$ is a complete sublattice of $\mathcal{IV}(X,Y,I)$.*

2. A complete sublattice $U \subseteq \mathcal{IV}(X,Y,I)$ is equal to $\mathcal{B}(X,Y,L)$ for some semi-closed subrelation $L \subseteq I$ iff for each $\langle A,B\rangle, \langle C,D\rangle \in U$,

$$A \subseteq C \quad \textit{iff} \quad D \subseteq B. \tag{3.9}$$

It holds $L = \bigsqcup U$.

Proof. 1. Directly by Lemma 70.

2. Condition (3.9) is satisfied for all concept lattices. Thus, it suffices to show the converse implication, namely that if (3.9) holds, then $U = \mathcal{B}(X,Y,L)$ for $L = \bigsqcup U$. Let $\langle A,B\rangle \in U$. Condition (3.9) ensures that $A \times B$ is a maximal rectangle in L. Therefore, $\langle A,B\rangle \in \mathcal{B}(X,Y,L)$ and so $U \subseteq \mathcal{B}(X,Y,L)$.

To prove the converse inclusion, we show that each object concept of $\mathcal{B}(X,Y,L)$ belongs to U. This is sufficient as each concept is the supremum of a set of object concepts and U is a complete lattice. Let $x \in X$ be an object and let $S_x = \{\langle C,D\rangle \in U \mid x \in C\}$. Since $\langle X, X^{\uparrow_I}\rangle \in S_x$, S_x is nonempty. For $\langle A,B\rangle = \bigwedge S_x$ we have $x \in A$ by (1.1) and Lemma 70. Furthemore, if $y \in Y$ satisfies $\langle x,y\rangle \in L$, then there is $\langle C,D\rangle \in S_x$ such that $y \in D$. Therefore (1.1), $y \in B$ and so $B = \{x\}^{\uparrow_L}$. Now let $x' \in A$. If for $y \in Y$ it holds $\langle x',y\rangle \in L$, then $\langle x',y\rangle \in \langle C,D\rangle$ for the same $\langle C,D\rangle \in S_x$ for which $\langle x,y\rangle \in \langle C,D\rangle$. Thus, $x' \in B^{\downarrow_L}$ and so $A = B^{\downarrow_L}$. We conclude that $\langle A,B\rangle \in \mathcal{B}(X,Y,L)$ is the object concept of the object x. As $\langle A,B\rangle = \bigwedge S_x$, it holds $\langle A,B\rangle \in U$ by completeness and the proof is finished. $\square$

Remark 72. In the case of semi-closed subrelations, we don't have a one-to-one correspondence between semi-closed subrelations of I and complete sublattices of $\mathcal{IV}(X,Y,I)$. Check the following section for more details.

In the last lemma of this section, we give a characterization of all complete sublattices of the concept lattice $\mathcal{B}(X, Y, I)$ such that for each sublattice, the associated closed subrelation of I contains given subrelation $K \subseteq I$.

Before introducing the lemma, we will need some preliminary results. Each interval-preconcept $\langle A, B\rangle$ of I determines a closed interval in the concept lattice $\mathcal{B}(X, Y, I)$, namely the interval $[\langle A, A^{\uparrow_I}\rangle, \langle B^{\downarrow_I}, B\rangle]$. This correspondence between interval-preconcepts of I and closed intervals in $\mathcal{B}(X, Y, I)$ is evidently bijective. For each semi-closed subrelation $L \subseteq I$ we denote by S_L the system of all closed intervals of $\mathcal{B}(X, Y, I)$ determined by concepts of L (which are interval-preconcepts of I). This system corresponds to the concept lattice $\mathcal{B}(X, Y, L)$. Thus, the above Lemma 71 can be used for further investigation of the structure of S_L.

Lemma 73. *Let $K \subseteq I$ be an arbitrary subrelation, $U \subseteq \mathcal{B}(X, Y, I)$ a complete sublattice with the associated closed subrelation $J \subseteq I$. Then $J \supseteq K$ iff U has nonempty intersection with each interval from the system $\mathrm{S}_{\mathrm{C}_{\mathcal{IV}}K}$.*

Proof. As we know by Lemma 66, the condition $J \supseteq K$ is equivalent to $J \supseteq \mathrm{C}_{\mathcal{IV}}K$. The latter condition means that for each interval-preconcept $\langle A, B\rangle \in \mathcal{B}(X, Y, \mathrm{C}_{\mathcal{IV}}K)$ there is a concept $\langle C, D\rangle \in \mathcal{B}(X, Y, J)$ such that $A \subseteq C$ and $B \subseteq D$, which is equivalent to $\langle C, D\rangle \in [\langle A, A^{\uparrow_I}\rangle, \langle B^{\downarrow_I}, B\rangle]$. □

We already know that procedure Generate always outputs a closed subrelation given any subrelation $K \subseteq I$. However, dual computation, i.e. interchanging parts for even and odd indices, might lead to a different result. Notice that by executing the first two iterations of the procedure it is already determined what the result will be. This is because after running two iterations the intermediate resulting context can be written as a union of some concepts of I (Lemma 42). In fact, it is already decided after the first iteration as it leaves us with either correct intents or extents. Thus, to see what the result is going to be, we can focus just on the first iteration. Due to Lemma 73 we know how concepts of K determine closed intervals of $\mathcal{B}(I)$. Now, the result depends on which concepts we choose from corresponding intervals. Taking procedure Generate as is, the first iteration selects the upper bounds of all such intervals. Dual computation selects the lower bounds. Note that this selection can have an effect on the size (w.r.t. $\subseteq$) of the result (see Example 74).

EXAMPLE 74. In Fig. 3.20 we can see three formal contexts. Bold dots in the formal context K (resp. J_d) mark incidences of the initial formal context I that are not present in K (resp. J_d). Execution of GENERATE(K) leads to J. Dual computation results in J_d and we have $J_d \subset J$.

K	y_1	y_2
x_1	$\times$	$\bullet$
x_2	$\bullet$	
x_3		$\times$

J	y_1	y_2
x_1	$\times$	$\times$
x_2	$\times$	
x_3		$\times$

J_d	y_1	y_2
x_1	$\times$	$\times$
x_2	$\bullet$	
x_3		$\times$

Figure 3.20: Three formal contexts showing that dual computation of GENERATE(K) can lead to a smaller (w.r.t. $\subseteq$) closed subrelation.

3.5 Interval-preconcepts

We already met interval-preconcepts in the previous section where we studied them especially due to their connection to semi-closed and closed subrelations. Now, we take a closer look on such rectangles. We showed that they can be ordered in the same way as preconcepts and together with this ordering the set of all interval-preconcepts of a given formal context forms a complete lattice.

Interval-preconcepts have two defining properties: 1. they are preconcepts, 2. they uniquely determine an interval in the corresponding concept lattice. The second property plays a crucial role in our motivation to study such formal rectangles. First, we focus on a motivation from psychology and show where interval-preconcepts can fit in the theory of concepts. Second, we provide a more formal motivation relating interval-preconcepts to block relations.

There has been quite a lot of work done in the field of cognitive psychology concerning concepts especially in last 70 years (see [29] for a quick and [30] for a comprehensive overview). Nowadays, there are several definitions of concept in cognitive psychology that are sometimes almost contradictory and some approaches actually reject any definition of concept as it is impossible to define (see also [31]). They argue that for any such definition there exist an example (usually natural and very simple) that does not conform to it [32]. Nevertheless, we cannot possibly hope to mathematically define a notion that we are unable to grasp even with the help of vagueness of our language. Therefore, the notion of formal concept is based on the definition of concept from *The Classical Theory of Concepts* which dates back to antiquity to Plato [32].

It accepts a definition of concept as a structured mental representation that encodes necessary and sufficient conditions for its application. By application, we usually mean the ability to decide if an object is part of a concept or not, i.e. the process of categorization in the terms of cognitive psychology.

Accepting our notion of formal concept as a mathematical representation of the classical definition of concept leaves us with several possible interpretations of the notion interval-preconcept. In the study of concept acquisition, we can look at them as possible stages of the learning process. In formal terms, we represent a whole interval of concepts as a single entity and we are yet to acquire knowledge that would let us draw distinctions between concepts in this interval. We can also look at interval-preconcepts as on a kind of prototypical instances of a concept. Consider the following example that is formally captured in Figure 3.21. A child, lets call him Adam, living in a household with a single animal, a dog called Thor. For Adam, the concept of animal might be represented by a formal concept containing only Thor in its extension and his features in its intension. The features he is able to recognize might not be all the features of Thor but some subset of them. Adam might ignore some features because they seem unimportant at that point, it is because Adam has not yet seen any other animal except for Thor. Afterwards, Adam meets another dog, Loki, of a different breed and in order to distinguish between the two dogs he has to acknowledge some differentiating features and assimilate them to the concept containing Thor. The original concept containing only Thor in its extension and features now shared by both dogs in its intension becomes interval-preconcept and can be looked upon as a kind of prototype.

Other possible application of interval-preconcepts stems from the necessity of ignorance. It has been argued (see [30]) that people make adjustment to their process of categorization depending on current circumstances. One such adjustment includes ignorance of some features. This prevents overwhelming our mind and speeds up the process of categorization. By restraining the set of features only to a valid intension of a more general concept we obtain a way of thinking about more specific concepts in more general, but coherent, terms.

The second part of our motivation is purely formal. We have a new type of formal rectangle that arises from a notion of semi-closed subrelation which is closely related to the notion of block relation. We explore their connection and investigate potential areas where interval-preconcepts can help to represent more complex notions whilst providing a different viewing angle.

Adam	$\cdots$	woofs	fur	$\cdots$
$\vdots$				
Thor		$\times$	$\times$	
$\vdots$				

Adam	$\cdots$	woofs	fur	long tail	pointy ears	$\cdots$
$\vdots$						
Thor		$\boxed{\times}$	$\boxed{\times}$	$\times$		
Loki		$\times$	$\times$		$\times$	
$\vdots$						

Figure 3.21: State of the relevant part of Adam's knowledge, before and after meeting Loki, represented as a part of a formal context. The concept $\langle\{\text{Thor}\}, \{\text{woofs}, \text{fur}\}\rangle$, marked by the box, of the former formal context is an interval-preconcept of the latter formal context.

Definition 75 (interval-preconcept). An interval-preconcept of a formal context $\langle X, Y, I\rangle$ is a preconcept $\langle A, B\rangle$ such that $A = A^{\uparrow_I\downarrow_I}$ and $B = B^{\downarrow_I\uparrow_I}$.

To better understand relations of interval-preconcepts to other well-known types of rectangles we provide a summary in Fig. 3.22 and 3.23. Moreover, in Example 76 we can find proper instances of all types of rectangles.

	$A \subseteq B^{\downarrow}$ $B \subseteq A^{\uparrow}$	$A^{\uparrow\downarrow} = B^{\downarrow}$	$B = A^{\uparrow}$	$A = B^{\downarrow}$	$A = A^{\uparrow\downarrow}$ $B = B^{\downarrow\uparrow}$
preconcept	$\times$				
protoconcept	$\times$	$\times$			
interval-preconcept	$\times$				$\times$
$\sqcap$-semiconcept	$\times$	$\times$	$\times$		
$\sqcup$-semiconcept	$\times$	$\times$		$\times$	
formal concept	$\times$	$\times$	$\times$	$\times$	$\times$

Figure 3.22: A formal context of different types of formal rectangles and their properties.

Example 76. In Fig. 3.24 we can see proper examples of different types of formal rectangles. For example, we are able to identify a proper interval-preconcept $\langle\{x_2\}, \{y_2\}\rangle$. Obviously, it is not a formal concept. It also easy to see why it is not a protoconcept: $\{x_2\}^{\uparrow_I\downarrow_I} = \{x_2\} \neq \{x_1, x_2, x_3\} = \{y_2\}^{\downarrow_I}$.

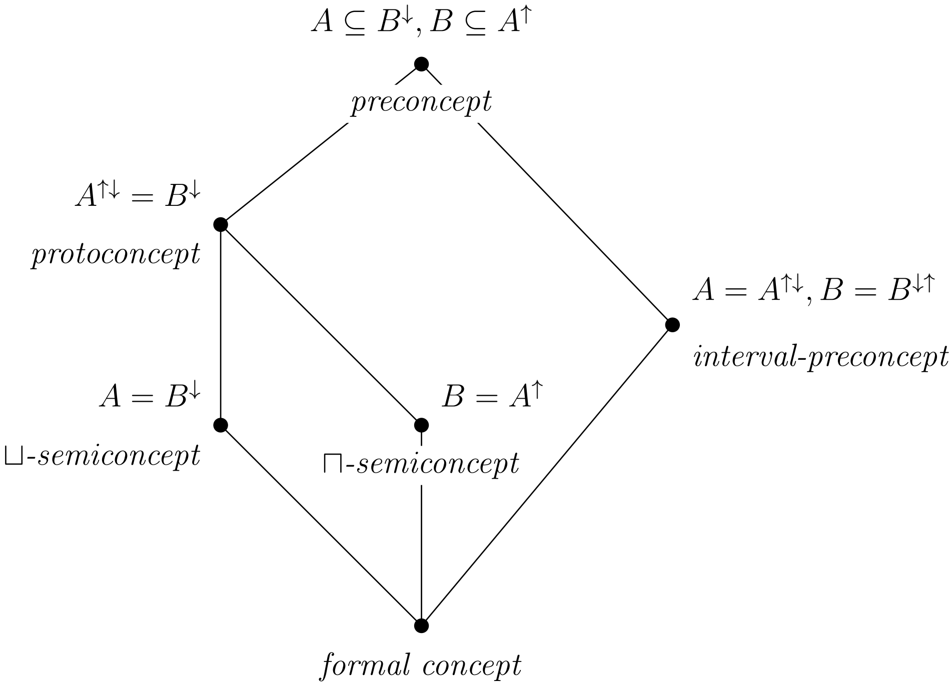


Figure 3.23: The concept lattice corresponding to the formal context from the Fig. 3.22 showing relations between different types of formal rectangles.

I	y_1	y_2	y_3
x_1		$\times$	
x_2	$\times$	$\times$	$\times$
x_3		$\times$	$\times$

Figure 3.24: Different types of formal rectangles can be found in this context. We can identify following examples: a proper preconcept $\langle\{x_2\}, \{y_3\}\rangle$, a proper interval-preconcept $\langle\{x_2\}, \{y_2\}\rangle$, a proper protoconcept $\langle\{x_3\}, \{y_3\}\rangle$ and proper semiconcepts $\langle\{x_3\}, \{y_2, y_3\}\rangle$ and $\langle\{x_2, x_3\}, \{y_3\}\rangle$.

The notion of interval-preconcept originates from our investigation of semi-closed subrelations. Such relations are defined similarly to block relations, however they are subrelations as opposed to being super-relations. Thus, we investigate the relation between interval-preconcepts and block relations. Recall, each interval-preconcept $\langle A, B\rangle \in \mathcal{IV}(X, Y, I)$ uniquely determines an interval in the concept lattice $\mathcal{B}(I)$, namely, $[\langle A, A^{\uparrow_I}\rangle, \langle B^{\downarrow_I}, B\rangle]$. We now show how a block relation corresponds to a set of interval-preconcepts.

Lemma 77. *Let $L \supseteq I$ be a block relation, then for each $\langle A, B\rangle \in \mathcal{B}(L)$, $c = \langle B^{\downarrow_I}, A^{\uparrow_I}\rangle$ is an interval-preconcept of I. Denote R_L the set of all such interval-preconcepts. R_L is an order-embedded complete lattice in $\mathcal{IV}(X, Y, I)$ with infima and suprema given by:*

$$\bigwedge_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota^{\uparrow_I}\right)^{\downarrow_K\uparrow_K\downarrow_I}, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle, \tag{3.10}$$

$$\bigvee_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota^{\downarrow_I}\right)^{\uparrow_K\downarrow_K\uparrow_I} \right\rangle. \tag{3.11}$$

Proof. From $\langle A, B\rangle \in \mathcal{B}(L)$ we have $A = A^{\uparrow_I\downarrow_I}$, $B = B^{\downarrow_I\uparrow_I}$ and since $B^{\downarrow_I} \subseteq B^{\downarrow_K} = A$ it holds $A^{\uparrow_I} \subseteq B^{\downarrow_I\uparrow_I}$. Similarly we obtain $B^{\downarrow_I} \subseteq A^{\uparrow_I\downarrow_I}$ proving $\langle B^{\downarrow_I}, A^{\uparrow_I}\rangle \in \mathcal{IV}(X, Y, I)$.

Denote $\rho : \mathcal{B}(K) \to \mathrm{R}_L$ and put $\rho(\langle A, B\rangle) = \langle B^{\downarrow_I}, A^{\uparrow_I}\rangle$. ρ is evidently a bijection with inverse $\rho^{-1}(\langle A, B\rangle) = \langle B^{\downarrow_I}, A^{\uparrow_I}\rangle$. ρ is order-preserving since for $\langle A, B\rangle \leq \langle C, D\rangle \in \mathcal{B}(K)$ we have $A \subseteq C$ (resp. $D \subseteq B$) implying $C^{\uparrow_I} \subseteq A^{\uparrow_I}$ (resp. $B^{\downarrow_I} \subseteq D^{\downarrow_I}$) showing $\rho(\langle A, B\rangle) \leq \rho(\langle C, D\rangle)$. Similarly we can show that ρ^{-1} is also order-preserving. Thus, R_L is a complete lattice isomorphic to $\mathcal{B}(K)$.

Now, we prove only 3.10 as 3.11 is dual. Let us have $\langle A_\iota, B_\iota\rangle \in R_L$ for $\iota \in \mathcal{I}$. We have

$$\begin{aligned}
\bigwedge_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle &= \rho(\bigwedge_{\iota\in\mathcal{I}} \rho^{-1}(\langle A_\iota, B_\iota\rangle)) \\
&= \rho(\bigwedge_{\iota\in\mathcal{I}} \langle B_\iota^{\downarrow_I}, A_\iota^{\uparrow_I}\rangle) \\
&= \rho\left\langle \bigcap_{\iota\in\mathcal{I}} B_\iota^{\downarrow_I}, \left(\bigcup_{\iota\in\mathcal{I}} A_\iota^{\uparrow_I}\right)^{\downarrow_K\uparrow_K} \right\rangle \\
&= \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota^{\uparrow_I}\right)^{\downarrow_K\uparrow_K\downarrow_I}, \left(\bigcap_{\iota\in\mathcal{I}} B_\iota^{\downarrow_I}\right)^{\uparrow_I} \right\rangle \\
&= \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota^{\uparrow_I}\right)^{\downarrow_K\uparrow_K\downarrow_I}, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle.
\end{aligned}$$

□

Evidently, the converse direction does not generally hold, i.e. not all subsets of $\mathcal{IV}(X, Y, I)$ correspond to block relations. Check Examples 79 and 80 for a simple counterexamples.

Corollary 78. *Let $V \subseteq \mathcal{IV}(X,Y,I)$ we have $V = \mathrm{R}_L$ for some block relation $L \supseteq I$ iff V is an order-embedded complete lattice in $\mathcal{IV}(X,Y,I)$ such that for each $\langle A,B\rangle, \langle C,D\rangle \in V$*

$$A \subseteq C \quad \textit{iff} \quad D \subseteq B. \tag{3.12}$$

and $L = \bigcup_{\langle A,B\rangle \in V}(B^{\downarrow_I} \times A^{\uparrow_I})$.

Example 79. Recall that a binary relation θ on a complete lattice V is called a *complete tolerance* if it is reflexive, symmetric and compatible with suprema and infima, i.e. we have

$$x_\iota \theta y_\iota \text{ for } \iota \in \mathcal{I} \Rightarrow \left(\bigvee_{\iota\in\mathcal{I}} x_\iota\right) \theta \left(\bigvee_{\iota\in\mathcal{I}} y_\iota\right) \text{ and } \left(\bigwedge_{\iota\in\mathcal{I}} x_\iota\right) \theta \left(\bigwedge_{\iota\in\mathcal{I}} y_\iota\right).$$

We call a subset W of V a block if $x\theta y$ holds for all $x, y \in W$ and W is maximal (w.r.t. $\subseteq$). There is a 1-1 correspondence between complete tolerances of $\mathcal{B}(I)$ and block relations of I.

Consider the formal context from Fig. 3.25 (the one on the right side), corresponding concept lattice is depicted in Fig. 3.26. Take $V = \{\langle\emptyset, \{y_4\}\rangle\}$. Concepts from corresponding interval are drawn with dashed circles.

Evidently, there is no block relation K for which $V = \mathrm{R}_K$ as the smallest complete tolerance containing this interval (determined by interval-preconcept $\langle\emptyset, \{y_4\}\rangle$) as a block also includes interval corresponding to the interval-preconcept $\langle\{x_2\}, \{y_2\}\rangle$.

I	y_1	y_2	y_3	y_4
x_1		×		×
x_2	×	×	×	
x_3		×		

	y_1	y_2	y_3	y_4
x_1	×	×	×	×
x_2				
x_3				

Figure 3.25: A formal context I where we take $V = \{\langle\emptyset, \{y_4\}\rangle\}$. Creating a relation in the same way as in Corollary 78 does not result in a block relation of I.

Example 80. Consider the formal context from Fig. 3.27. Corresponding concept lattice is depicted in Fig. 3.28 and interval-preconcept lattice can be seen in Fig. 3.31. Take $V = \{\langle\emptyset, \{y_1\}\rangle, \langle\{x_1\}, \{y_2\}\rangle, \langle\{x_3\}, \emptyset\rangle\}$. Now, create a relation L in the same way as in Corollary 78. We can see from Fig. 3.27 that we

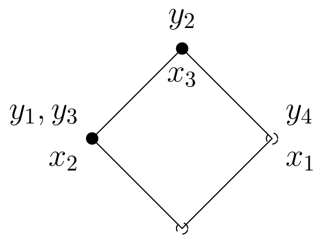


Figure 3.26: Concept lattice corresponding to formal context from Fig. 3.25 where concepts from selected interval are drawn with dashed circle.

obtain a block relation. However, V does not form an order-embedded complete lattice as there is no upper bound for V. Nevertheless, there exists a closure for V and it also contains interval-preconcept $\langle \emptyset, Y\rangle$. However, by adding it, the resulting set will not satisfy condition (3.12).

I	y_1	y_2	y_3
x_1	$\times$	$\times$	$\bullet$
x_2	$\bullet$	$\times$	$\bullet$
x_3			$\times$

Figure 3.27: A formal context I where we take $V = \{\langle \emptyset, \{y_1\}\rangle, \langle \{x_1\}, \{y_2\}\rangle, \langle \{x_3\}, \emptyset\rangle\}$. Incidences marked with bold dots are added in the same way as in Corollary 78. We obtain a block relation of I.

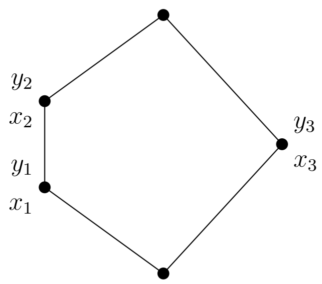


Figure 3.28: Concept lattice corresponding to formal context from Fig. 3.27.

We already showed that the set of all interval-preconcepts $\mathcal{IV}(X, Y, I)$ of given context is a complete lattice. Now, we show how to create a formal context K such that its concept lattice is isomorphic to $\mathcal{IV}(X, Y, I)$. For this purpose we adopt the following notation. For any set A we put $\overline{A} = \{\overline{x} \mid x \in A\}$.

Lemma 81. *For a formal context $\langle X, Y, I\rangle$ we have $\mathcal{IV}(X, Y, I) \cong \mathcal{B}(X \cup \overline{X}, Y \cup \overline{Y}, K_I)$ where*

$$K_I = \{\langle x, y\rangle, \langle \overline{x}, y\rangle, \langle x, \overline{y}\rangle \mid x \in X, y \in Y \text{ and } \langle x, y\rangle \in I\} \cup \overline{X} \times \overline{Y}.$$

Proof. It can be easily seen from the construction that concepts of $\mathcal{B}(K)$ are of the form $\langle A \cup \overline{C}, D \cup \overline{B}\rangle$ where $A \subseteq C$ and $D \subseteq B$ and $\langle A, B\rangle, \langle C, D\rangle \in \mathcal{B}(I)$. The rest follows immediately form the fact that interval-preconcepts and intervals of $\mathcal{B}(I)$ are in a 1-1 correspondence. □

Interestingly, the previous construction turns out to be equivalent to the direct product of the formal context $\langle X, Y, I\rangle$ and formal context from Fig. 3.29.

	y_1	y_2
x_1		
x_2		$\times$

Figure 3.29: A formal context with the concept lattice isomorphic to a three element chain.

Corollary 82. *For a formal context $\langle X, Y, I\rangle$ we have*

$$\mathcal{IV}(X, Y, I) \cong \mathcal{B}(\langle X, Y, I\rangle \times \langle \{x_1, x_2\}, \{y_1, y_2\}, \{\langle x_2, y_2\rangle\}\rangle).$$

The previous means that $\mathcal{IV}(X, Y, I)$ is isomorphic to a tensor product of certain concept lattices. Therefore, we can use any result about tensor product of concept lattices to investigate properties of $\mathcal{IV}(X, Y, I)$. Now, we provide a basic theorem on interval-preconcept lattices.

Theorem 83 (Basic Theorem on Interval-preconcept Lattices). *The interval-preconcept lattice $\mathcal{IV}(X, Y, I)$ is a complete lattice in which infima and suprema are given by:*

$$\bigwedge_{\iota \in \mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \bigcap_{\iota \in \mathcal{I}} A_\iota, \left(\bigcup_{\iota \in \mathcal{I}} B_\iota\right)^{\downarrow_I \uparrow_I} \right\rangle, \tag{3.13}$$

$$\bigvee_{\iota \in \mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota \in \mathcal{I}} A_\iota\right)^{\uparrow_I \downarrow_I}, \bigcap_{\iota \in \mathcal{I}} B_\iota \right\rangle. \tag{3.14}$$

In general, a complete lattice V with an element p admits an isomorphism α with interval-preconcept lattice $\mathcal{IV}(X,Y,I)$ with $\alpha(p) = \langle \emptyset^{\uparrow\downarrow}, \emptyset^{\downarrow\uparrow}\rangle$ if and only if there exist mappings $\gamma : X \to V$ and $\mu : Y \to V$ where

$$D_\vee = \bigcup_{x\in X} \{\gamma(x), \gamma(x) \wedge p\} \text{ is supremally dense in } V, \tag{3.15}$$

$$D_\wedge = \bigcup_{y\in Y} \{\mu(y), \mu(y) \vee p\} \text{ is infimally dense in } V, \tag{3.16}$$

such that for any $x \in X$, $y \in Y$ it holds

$$xIy \Leftrightarrow \gamma(x) \leq \mu(y) \Leftrightarrow \gamma(x) \wedge p \leq \mu(y) \wedge p \Leftrightarrow \gamma(x) \vee p \leq \mu(y) \vee p. \tag{3.17}$$

Proof. The first part is proven in Lemma 70. We now focus on the second part. For the rest of this proof, denote

$$D^p_\vee = \bigcup_{x\in X} \{\gamma(x) \wedge p\},$$
$$D^p_\wedge = \bigcup_{y\in Y} \{\mu(y) \vee p\}.$$

"$\Rightarrow$": We have an isomorphism α between V and $\mathcal{IV}(X,Y,I)$. First, we prove the statement for the special case $V = \mathcal{IV}(X,Y,I)$. Put

$$\gamma(x) = \langle \{x\}^{\uparrow_I\downarrow_I}, \{x\}^{\uparrow_I}\rangle,$$
$$\mu(y) = \langle \{y\}^{\downarrow_I}, \{y\}^{\downarrow_I\uparrow_I}\rangle.$$

We show that $D_\vee$ is $\bigvee$-dense, the proof for $D_\wedge$ is dual. First, recall that object concepts, i.e. elements of $\gamma(X)$, are $\bigvee$-dense in $\mathcal{B}(I)$ and so every extent can be formulated in the terms of object extents. Now, since $\alpha(p) = \langle \emptyset^{\uparrow\downarrow}, \emptyset^{\downarrow\uparrow}\rangle$ and due to the already proven first part we see that elements of $D^p_\vee = \bigcup_{x\in X}\{\gamma(x)\wedge p\} = \bigcup_{x\in X}\{\langle \emptyset^{\uparrow_I\downarrow_I}, \{x\}^{\uparrow_I}\rangle\}$, i.e. they have the smallest extent and object intents. Now, it can be easily seen that every element of $\mathcal{IV}(X,Y,I)$ can be written as suprema of elements from $D_\vee = \gamma(X) \cup D^p_\vee$. Specifically, extents are taken care of by elements of $\gamma(X)$ and intents by $D^p_\vee$.

Now, we prove the equalities (3.17). We have to show the equality of the following:

1. xIy,
2. $\gamma(x) \leq \mu(y)$,

3. $\gamma(x) \wedge p \leq \mu(y) \wedge p$,

4. $\gamma(x) \vee p \leq \mu(y) \vee p$.

"$1 \Rightarrow 2$": $xIy \Rightarrow \{y\} \subseteq \{x\}^{\uparrow_I} \Rightarrow \{x\}^{\uparrow_I\downarrow_I} \subseteq \{y\}^{\downarrow_I} \Rightarrow \gamma(x) \leq \mu(y)$ since both $\gamma(x)$, $\mu(y)$ are concepts.
"$2 \Rightarrow 3$": $\gamma(x) \leq \mu(y) \Rightarrow \gamma(x) \wedge p \leq \mu(y) \wedge p$.
"3. $\Rightarrow$ 4.": $\gamma(x) \wedge p \leq \mu(y) \wedge p \Leftrightarrow \langle \emptyset^{\uparrow_I\downarrow_I}, \{x\}^{\uparrow_I} \rangle \leq \langle \emptyset^{\uparrow_I\downarrow_I}, \{y\}^{\downarrow_I\uparrow_I} \rangle \Rightarrow \{y\}^{\downarrow_I\uparrow_I} \subseteq \{x\}^{\uparrow_I} \Rightarrow \{x\}^{\uparrow_I\downarrow_I} \subseteq \{y\}^{\downarrow_I} \Rightarrow \langle \{x\}^{\uparrow_I\downarrow_I}, \emptyset^{\downarrow_I\uparrow_I} \rangle \leq \langle \{y\}^{\uparrow_I}, \emptyset^{\downarrow_I\uparrow_I} \rangle \Leftrightarrow \gamma(x) \vee p \leq \mu(y) \vee p$.
"$4 \Rightarrow 1$": $\gamma(x) \vee p \leq \mu(y) \vee p \Leftrightarrow \langle \{x\}^{\uparrow_I\downarrow_I}, \emptyset^{\downarrow_I\uparrow_I} \rangle \leq \langle \{y\}^{\downarrow_I}, \emptyset^{\downarrow_I\uparrow_I} \rangle \Rightarrow x \in \{x\}^{\uparrow_I\downarrow_I} \subseteq \{y\}^{\downarrow_I} \Rightarrow xIy$.

More generally, if $V \cong \mathcal{IV}(X,Y,I)$ and $\beta : \mathcal{IV}(X,Y,I) \to V$ is an isomorphism, we define mappings γ and μ as

$$\gamma(x) = \beta(\langle \{x\}^{\uparrow_I\downarrow_I}, \{x\}^{\uparrow_I} \rangle),$$
$$\mu(y) = \beta(\langle \{y\}^{\downarrow_I}, \{y\}^{\downarrow_I\uparrow_I} \rangle).$$

and we can prove the required properties of the mappings in a similar fashion to the above. This concludes this part of the proof.

"$\Leftarrow$": Conversely, we start with a complete lattice V and mappings γ and μ satisfying properties stated above and we define

$$\alpha : V \to \mathcal{IV}(X,Y,I)$$

by

$$\alpha(v) = \langle \{x \in X \mid \gamma(x) \leq v\}, \{y \in Y \mid v \leq \mu(y)\} \rangle.$$

Before we demonstrate that α is well-defined, we show several equalities that we use in the rest of the proof. Also note that α is evidently order-preserving.

$$\gamma(x) \wedge p \leq v \Leftrightarrow \gamma(x) \wedge p \leq v \wedge p, \tag{3.18}$$
$$v \leq \mu(y) \vee p \Leftrightarrow v \vee p \leq \mu(y) \vee p, \tag{3.19}$$
$$\gamma(x) \leq \mu(y) \vee p \Leftrightarrow xIy \Leftrightarrow \gamma(x) \wedge p \leq \mu(y), \tag{3.20}$$

where $x \in X$, $y \in Y$ and $v, p \in V$. The first two equalities follow trivially from the properties of lattices. The third is a direct consequence of the first two and equalities (3.17).

Next, we show that for $v \in V$ we have

$$v \vee p = \bigvee\{\gamma(x) \vee p \mid \gamma(x) \leq v\}, \tag{3.21}$$

$$v \wedge p = \bigvee\{\gamma(x) \wedge p \mid \gamma(x) \wedge p \leq v\}. \tag{3.22}$$

The first equality holds due to the existence of $A \subseteq \gamma(X)$ and $C \subseteq D_\vee^p$ such that $v = \bigvee(A \cup C)$ since $v \vee p = (\bigvee A) \vee (\bigvee C) \vee p = \bigvee A \vee p$ due to $\bigvee C \leq \bigvee D_\vee^p \leq p$. The second one is obtained immediately from (3.18), $\bigvee$-density of $D_\vee$ and from the fact $v \wedge p \leq p$.

Lastly, we use the following equalities in the rest of the proof:

$$v \leq \mu(y) \Leftrightarrow v \wedge p \leq \mu(y), \tag{3.23}$$

$$\gamma(x) \leq v \Leftrightarrow \gamma(x) \leq v \vee p. \tag{3.24}$$

We prove the first equality, the second is dual. Left to right direction is trivial. Now, we show the converse also holds. By (3.22) we have $v \wedge p = \bigvee\{\gamma(x) \wedge p \mid \gamma(x) \wedge p \leq v\}$ and $v = (v \wedge p) \vee (\bigvee\{\gamma(x) \mid \gamma(x) \leq v\})$ due to $\bigvee$-density of $D_\vee$. We assumed $v \wedge p \leq \mu(y)$ and since $\gamma(x) \leq v$ implies $\gamma(x) \wedge p \leq v$ and due to equality $\gamma(x) \wedge p \leq \mu(y) \Leftrightarrow \gamma(x) \leq \mu(y)$ (3.20), we arrive at $v \leq \mu(y)$.

Now, we investigate how derivation operators can be read in V, let $v \in V$:

$$\begin{aligned}
&\text{for } A = \{x \in X \mid \gamma(x) \leq v\} \text{ we have } A^{\uparrow_I} = \{y \in Y \mid v \leq \mu(y) \vee p\},\\
&\text{for } C = \{x \in X \mid \gamma(x) \wedge p \leq v\} \text{ we have } C^{\uparrow_I} = \{y \in Y \mid v \leq \mu(y)\},\\
&\text{for } D = \{y \in Y \mid v \leq \mu(y)\} \text{ we have } D^{\downarrow_I} = \{x \in X \mid \gamma(x) \wedge p \leq v\},\\
&\text{for } B = \{y \in Y \mid v \leq \mu(y) \vee p\} \text{ we have } B^{\downarrow_I} = \{x \in X \mid \gamma(x) \leq v\}.
\end{aligned}$$

We prove the first two statements, the others are dual. For the first one, we know $v \leq \mu(y) \vee p \Leftrightarrow v \vee p \leq \mu(y) \vee p$ and from the previous $v \vee p = \bigvee_{x \in A} \gamma(x) \vee p$ (3.21) and also $xIy \Leftrightarrow \gamma(x) \vee p \leq \mu(y) \vee p$ proving the first statement. The second one follows directly from (3.22) and (3.23). Now, it should be fairly easy to see that A, C are extents and B, D are intents. Specifically, we have $A^{\uparrow_I} = B$ and $B^{\downarrow_I} = A$. Similarly, $C^{\uparrow_I} = D$ and $D^{\downarrow_I} = C$.

We are now well equipped to continue our proof and we start by showing that α is well-defined, i.e. for $v \in V$ we show that $\alpha(v)$ is an interval-preconcept of I. This comes directly from our investigation of derivation operators. Moreover, for the element p we obtain $\alpha(p) = \langle \emptyset^{\uparrow_I\downarrow_I}, \emptyset^{\downarrow_I\uparrow_I} \rangle$.

Now, lets define

$$\omega : \mathcal{IV}(X, Y, I) \to V$$

by

$$\omega(\langle A, B\rangle) = \left(\bigvee_{x\in A} \gamma(x)\right) \vee \left(\bigvee_{x\in B^{\downarrow_I}} \gamma(x) \wedge p\right).$$

Mapping ω is obviously order-preserving since for $\langle A, B\rangle \leq \langle C, D\rangle$ we get $A \subseteq C$ and $D \subseteq B \Rightarrow B^{\downarrow_I} \subseteq D^{\downarrow_I}$.

Lastly, we show that $\omega = \alpha^{-1}$:

$$\begin{aligned}
\omega(\alpha(v)) &= \omega(\langle \{x \in X \mid \gamma(x) \leq v\}, \{y \in Y \mid v \leq \mu(y)\}\rangle) \\
&= \left(\bigvee_{x\in A} \gamma(x)\right) \vee \left(\bigvee_{x\in B^{\downarrow_I}} \gamma(x) \wedge p\right) \\
&= \left(\bigvee_{\{x\in X\mid\gamma(x)\leq v\}} \gamma(x)\right) \vee \left(\bigvee_{\{x\in X\mid\gamma(x)\wedge p\leq v\}} \gamma(x) \wedge p\right) \\
&= v,
\end{aligned}$$

where $A = \{x \in X \mid \gamma(x) \leq v\}$ and $B = \{y \in Y \mid v \leq \mu(y)\}$. The previous holds due to $\bigvee$-density of $D_\vee$ and our investigation of derivation operators. □

Remark 84. The above basic theorem on interval-preconcept lattices can also be deduced from the results concerning tensor products in [8]. However, we present it here more specifically for our special case and provide it in similar wording to other basic theorems of other rectangle types. Also note that Theorem 83 does not specify the structure of interval-preconcept lattices and this flaw is rectified in Theorem 89.

We can immediately make several observations from the basic theorem and its proof. First, there are actually three isomorphisms between $\mathcal{B}(I)$ and certain parts of $\mathcal{IV}(X, Y, I)$, namely, intervals $[\langle \emptyset^{\uparrow_I\downarrow_I}, \emptyset^{\downarrow_I}\rangle, p]$, $[p, \langle \emptyset^{\downarrow_I}, \emptyset^{\downarrow_I\uparrow_I}\rangle]$ and the complete sublattice $\mathcal{IV}(X, Y, I) \cap \mathcal{B}(I)$. Second, we only need as many labels for $\mathcal{IV}(X, Y, I)$ as for $\mathcal{B}(I)$. Third, each element of $\mathcal{IV}(X, Y, I)$ is associated with two extents and two intents of I and it is easy to read them from the lattice.

Remark 85. There are some similarities between the basic theorem on interval-preconcept lattices and the basic theorem on preconcept lattices [12]. Mainly, the necessity of the "center" element p. For preconcept lattices, the element p is a singular element lesser or equal the supremum of all atoms (upper neighbors of the least element) and greater or equal the infimum of all coatoms (lower neighbors of the greatest element). Even though the element p has similar role in both theorems, in the case of interval-preconcept lattices we do not have a simple unique description for it.

Example 86. In Fig. 3.30 we can see a formal context $\langle X, Y, I\rangle$ and the corresponding formal context $\langle X \cup \overline{X}, Y \cup \overline{Y}, K_I\rangle$ (Lemma 81) with concept lattice isomorphic to interval-preconcept lattice of $\langle X, Y, I\rangle$. The interval-preconcept lattice $\mathcal{IV}(X, Y, I)$ is drawn in Fig. 3.31 and the concept lattice $\mathcal{B}(I)$ can be found in Fig. 3.28.

I	y_1	y_2	y_3
x_1	×	×	
x_2		×	
x_3			×

K_I	y_1	y_2	y_3	$\overline{y_1}$	$\overline{y_2}$	$\overline{y_3}$
x_1	×	×		×	×	
x_2		×			×	
x_3			×			×
$\overline{x_1}$	×	×		×	×	×
$\overline{x_2}$		×		×	×	×
$\overline{x_3}$			×	×	×	×

Figure 3.30: A formal context I and corresponding formal context K_I as given by Lemma 81. The concept lattice $\mathcal{B}(I)$ can be found in Fig. 3.28.

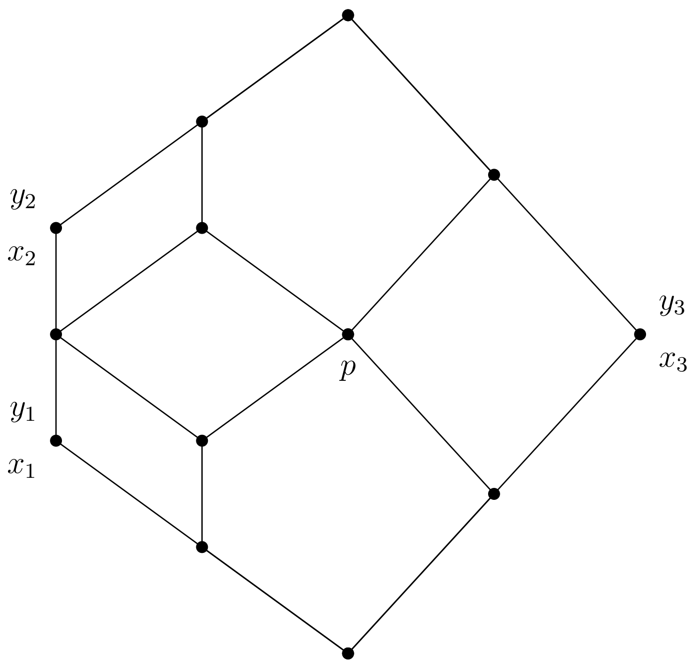


Figure 3.31: The interval-preconcept lattice $\mathcal{IV}(X, Y, I)$ corresponding to the formal context from Fig. 3.30.

We conclude this section with some final observations about the structure of interval-preconcept lattices.

Theorem 87.

$$\mathcal{IV}(I) \cong \langle\{\langle c_1, c_2\rangle \in \mathcal{B}(I) \times \mathcal{B}(I) \mid c_1 \le c_2\}, \sqsubseteq\rangle,$$

where $\langle c_1, c_2\rangle \sqsubseteq \langle c_3, c_4\rangle$ *iff* $c_1 \le c_3$ *and* $c_2 \le c_4$*.*

Proof. Denote $\mathcal{I}_{\mathcal{B}} = \langle\{\langle c_1, c_2\rangle \in \mathcal{B}(I) \times \mathcal{B}(I) \mid c_1 \le c_2\}, \sqsubseteq\rangle$ and define

$$\alpha : \mathcal{IV}(I) \to \mathcal{I}_{\mathcal{B}}$$

by

$$\alpha(\langle A, B\rangle) = \langle\langle A, A^{\uparrow_I}\rangle, \langle B^{\downarrow_I}, B\rangle\rangle.$$

Evidently α is the required isomorphism. Indeed, it is obviously well-defined, bijective and order-preserving with order-preserving inversion $\alpha^{-1}(\langle\langle A, B\rangle, \langle C, D\rangle\rangle) = \langle A, D\rangle$. $\square$

Corollary 88. *A complete lattice V isomorphic to the upper half of a Cartesian product $W \times W$ (i.e. the subset of $W \times W$ containing all pairs $\langle u, v\rangle$ where $u, v \in W$ and $u \le v$) of some complete lattice W is isomorphic to $\mathcal{IV}(W, W, \le)$.*

We call the complete lattice V from the previous corollary an *upper triangular complete lattice (of W)*. Lastly, we present a second version of basic theorem on interval-preconcept lattices. In this version, we properly identify the structure of interval-preconcept lattices and obtain the theorem in a more familiar form.

Theorem 89 (Second Basic Theorem on Interval-preconcept Lattices). *The interval-preconcept lattice $\mathcal{IV}(X, Y, I)$ is an upper triangular complete lattice of $\mathcal{B}(X, Y, I)$ in which infima and suprema are given by:*

$$\bigwedge_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \bigcap_{\iota\in\mathcal{I}} A_\iota, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle, \tag{3.25}$$

$$\bigvee_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \bigcap_{\iota\in\mathcal{I}} B_\iota \right\rangle. \tag{3.26}$$

In general, an upper triangular lattice V of W admits an isomorphism α with interval-preconcept lattice $\mathcal{IV}(X, Y, I)$ if and only if there exist mappings $\gamma : X \to V$ and $\mu : Y \to V$ such that

$$D_\vee = \bigcup_{x \in X} \{\gamma(x), \gamma(x) \wedge p\} \text{ is supremally dense in } V, \tag{3.27}$$

$$D_\wedge = \bigcup_{y \in Y} \{\mu(y), \mu(y) \vee p\} \text{ is infimally dense in } V, \tag{3.28}$$

where $p = \langle \bigvee \emptyset, \bigwedge \emptyset \rangle$ and for any $x \in X$, $y \in Y$ it holds

$$xIy \Leftrightarrow \gamma(x) \leq \mu(y) \Leftrightarrow \gamma(x) \wedge p \leq \mu(y) \wedge p \Leftrightarrow \gamma(x) \vee p \leq \mu(y) \vee p, \tag{3.29}$$

in particular, $V \cong \mathcal{IV}(W, W, \leq)$.

Proof. The proof follows from above observations, the proof of Theorem 83 and the fact that $\alpha(p) = \alpha(\langle \bigvee \emptyset, \bigwedge \emptyset \rangle) = \langle \emptyset^{\uparrow\downarrow}, \emptyset^{\downarrow\uparrow} \rangle$. □

3.6 Discussion and related work

An obvious advantage of our result on generating complete sublattices is that we avoid computation of any lattices and instead we work exclusively with contexts. In fact, our goal is to compute the closed subrelation corresponding to the given generated complete sublattice. The actual computation of the sublattice, if necessary, can be done with any well-known efficient algorithm for concept lattice construction. This should lead to shorter computation time, especially if the generated sublattice V is substantially smaller than $\mathcal{B}(X, Y, I)$.

In Lemma 51, we give an upper estimation of the number of iterations of our algorithms. It seems that this estimation could be improved. We provide some insight into the problem of complexity in Section 3.3 and outline situations that lead to a relatively high number of iterations. Nevertheless, at the time of writing, we were not able to construct any example with the number of iterations greater than $O(|X| + |Y|)$.

As far as related work goes, we are aware of only one published algorithm for generating sublattices [33]. Unfortunately, we cannot do any comparison as the algorithm in question is not correct. It does not always output a sublattice. In fact, it outputs an order-embedded lattice containing the generators which does not necessarily have to be a sublattice. In order to compare the result of our method with the one presented in [33] we would need to look at the result of our algorithm as on an order-embedded lattice, which it certainly is. However, this does not make any sense due to the fact, that order-embedded lattices do not form a closure system. This means that for a given set of generators there does not have to exist the smallest order-embedded lattice containing the generators. The algorithm from [33] is based on the following false claim: Given an element $e \in U$ and a set $P \subseteq U$ the smallest element $s \in U$ such that $e \leq s$ and $s \in \mathrm{C}_{\bigvee\bigwedge} P$ can be expressed as $s = \bigwedge\{p \in P \mid e \leq p\}$. Check Fig. 3.32 for a counterexample for this claim.

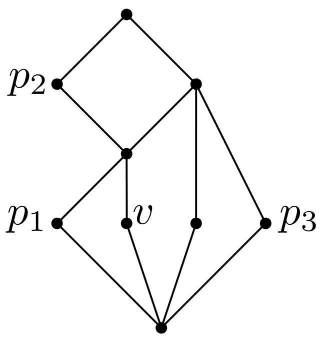


Figure 3.32: A counterexample for the claim that is the basis of the algorithm in [33], $P = \{p_1, p_2, p_3\}$. According to the claim, p_2 should be the smallest element such that it is greater or equal to v and it belongs to the $\mathrm{C}_{\bigvee\bigwedge} P$. However, such element is actually $p_1 \vee v = (p_1 \vee p_3) \wedge p_2$.

We also looked into the problem of characterizing all closed subrelations containing an arbitrary subrelation. We introduced a notion of semi-closed subrelation which is similar to that of block relation. In contrast with closed subrelations, semi-closed subrelations form a closure system. We showed how this notion can be used to solve the problem at hand. We also used it to identify an important type of subrelations for which we can always find a unique smallest closed subrelation containing it.

Investigation of semi-closed subrelations leads to a definition of interval-preconcepts which are a new type of formal rectangles. Interval-preconcepts uniquely determine an interval in the original concept lattice and they have a close relation to block relations and so to lattice factorization. We showed that together with the same ordering as preconcepts, they form a complete lattice. We studied properties of interval-preconcepts and we presented two versions of basic theorem on interval-preconcept lattices.

Chapter 4

Conclusion

We analyzed the basic step in incremental lattice construction–removal of an incidence–and based on this analysis we proposed two incremental algorithms for updating concepts and the corresponding concept lattice. As this is the smallest possible change in a formal context, we believe that this problem is in some form present in every incremental lattice construction method.

The performance of our algorithms depends heavily on the size of the order-embedded complete lattice (resp. interval) that contains exactly the concepts that are affected by the removal. Our experiments showed that the size of this interval is usually very small compared to the whole lattice. Combining it with proposed optimizations, the algorithm for updating concepts in fact computes two derivation operators and two set equality tests for each concept from the identified interval. Further extending presented method, we were able to remove an arbitrary preconcept at once without any additional overhead. By investigating possible extensions of our results, we arrived at a general method for updating a concept lattice upon an arbitrary change in the underlying context.

Afterwards, we focused on studying substructures, specifically complete sublattices generated by a set of elements. As it turns out, there is an efficient way of computing the closed subrelation corresponding to a complete sublattice generated by a set of elements. Computing such closed subrelation provides a full description of the corresponding generated complete sublattice and the actual construction of it, if necessary, can be done via any well-known efficient algorithm. Experiments with our method showed its efficiency and provided some insight into parameters that have an impact on its performance. Interestingly, the peak in the complexity of our method was achieved with small fixed sizes of the set of generators, i.e. the number of generators did not depend on the size of the lattices in these experiments.

The algorithm we proposed actually computes a closed subrelation for any given subrelation and in some sense the result seems minimal. This motivated us to further investigate this since it is in contrast with well-known result postulating that closed subrelations do not form a closure system. We introduced the notion of semi-closed subrelations that are more general than closed subrelations and indeed form a closure system. Using this notion we were able to identify an important type of subrelations for which there always exists the smallest closed subrelation containing given subrelation.

Our investigation of concepts of semi-closed relation lead us to a definition of a new type of formal rectangle that we call interval-preconcept. As most well-known types of formal rectangles have motivation in cognitive psychology, so does interval-preconcept and we showed some scenarios where it can serve as a formalization of some notion from cognitive psychology. We also explored their relations to other well-known types of formal rectangles and to block relations that are used for lattice factorization. Lastly, we showed how they can be structured into a complete lattice and proposed two versions of basic theorem on interval-preconcept lattices.

Bibliography

[1] KAUER, Martin; KRUPKA, Michal. Removing an Incidence from a Formal Context. In: BERTET, Karell; RUDOLPH, Sebastian (eds.). *Proc. CLA 2014*. 2014, pp. 195–206.

[2] KAUER, Martin; KRUPKA, Michal. Subset-generated complete sublattices as concept lattices. In: YAHIA, Sadok Ben; KONECNY, Jan (eds.). *Proc. CLA 2015*. 2015, pp. 11–21.

[3] KAUER, Martin; KRUPKA, Michal. Generating complete sublattices by methods of formal concept analysis. *Int. J. General Systems*. 2017, vol. 46, no. 5, pp. 475–489. Available from DOI: `10.1080/03081079.2017.1354964` (`http://dx.doi.org/10.1080/03081079.2017.1354964`).

[4] GANTER, Bernard. Order-embedded Complete Lattices. In: IGNATOV, Dmitry I.; NOURINE, Lhouari (eds.). *CLA*. CEUR-WS.org, 2018, vol. 2123, pp. 153–165. CEUR Workshop Proceedings. Available also from: `http://ceur-ws.org/Vol-2123/paper13.pdf`.

[5] BIRKHOFF, G. *Lattice Theory*. American Mathematical Society, 1973. American Mathematical Society colloquium publications. Available also from: `https://books.google.de/books?id=YEj4nQEACAAJ`.

[6] WILLE, Rudolf. Restructuring lattice theory: an approach based on hierarchies of concepts. In: RIVAL, I. (ed.). *Ordered Sets*. Boston, 1982, pp. 445–470.

[7] SINGH, Prem Kumar; KUMAR, Cherukuri Aswani; GANI, Abdullah. A comprehensive survey on formal concept analysis, its research trends and applications. *International Journal of Applied Mathematics and Computer Science*. 2016, vol. 26, no. 2, pp. 495–516. Available also from: `https://content.sciendo.com/view/journals/amcs/26/2/article-p495.xml`.

[8] GANTER, Bernhard; WILLE, Rudolf. *Formal Concept Analysis – Mathematical Foundations*. Springer, 1999.

[9] LUKSCH, Peter; WILLE, Rudolf. A Mathematical Model for Conceptual Knowledge Systems. In: BOCK, Hans-Hermann; IHM, Peter (eds.). *Classification, Data Analysis, and Knowledge Organization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 156–162. ISBN 978-3-642-76307-6.

[10] WILLE, Rudolf. Boolean Concept Logic. In: GANTER, Bernhard; MINEAU, Guy W. (eds.). *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 317–331. ISBN 978-3-540-44663-7.

[11] STAHL, Jürgen; WILLE, Rudolf. Preconcepts and set representation of contexts. In: GAUL, W.; SCHADER, M. (eds.). *Classification as a tool of research*. Amsterdam: North–Holland, 1986, pp. 431–438.

[12] BURGMANN, Christian; WILLE, Rudolf. The Basic Theorem on Preconcept Lattices. In: *Proceedings of the 4th International Conference on Formal Concept Analysis*. Dresden, Germany: Springer-Verlag, 2006, pp. 80–88. ICFCA'06. ISBN 3-540-32203-5, 978-3-540-32203-0. Available from DOI: `10.1007/11671404_5` (`http://dx.doi.org/10.1007/11671404_5`).

[13] PIAGET, Jean. *La formation du symbole chez l'enfant - imitation, jeu et reve - image et representation*. Delachaux et Niestle, 1959.

[14] CARPINETO, Claudio; ROMANO, Giovanni. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004. ISBN 0470850558.

[15] MERWE, Dean; OBIEDKOV, Sergei; KOURIE, Derrick. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: EKLUND, Peter (ed.). *Concept Lattices*. Springer Berlin Heidelberg, 2004, vol. 2961, pp. 372–385. Lecture Notes in Computer Science. ISBN 978-3-540-21043-6.

[16] OUTRATA, Jan. A Lattice-Free Concept Lattice Update Algorithm based on *CbO. In: OJEDA-ACIEGO, Manuel; OUTRATA, Jan (eds.). *CLA*. CEUR-WS.org, 2013, vol. 1062, pp. 261–274. CEUR Workshop Proceedings.

[17] NORRIS, E. M. An Algorithm for Computing the Maximal Rectangles in a Binary Relation. *Revue Roumaine de Mathématiques Pures et Appliquées*. 1978, vol. 23, no. 2, pp. 243–250.

[18] NOURINE, Lhouari; RAYNAUD, Olivier. A Fast Algorithm for Building Lattices. *Inf. Process. Lett.* 1999, vol. 71, no. 5-6, pp. 199–204. ISSN 0020-0190. Available from DOI: `10.1016/S0020-0190(99)00108-8` (`http://dx.doi.org/10.1016/S0020-0190(99)00108-8`).

[19] DOWLING, Cornelia E. On the Irredundant Generation of Knowledge Spaces. *J. Math. Psychol.* 1993, vol. 37, no. 1, pp. 49–62. ISSN 0022-2496. Available from DOI: `10.1006/jmps.1993.1003` (`http://dx.doi.org/10.1006/jmps.1993.1003`).

[20] KUZNETSOV, Sergei O.; OBIEDKOV, Sergei. Comparing Performance of Algorithms for Generating Concept Lattices. *Journal of Experimental and Theoretical Artificial Intelligence*. 2002, vol. 14, pp. 189–216.

[21] BLYTH, T.S. *Lattices and Ordered Algebraic Structures*. Springer London, 2005. Universitext. ISBN 9781852339050. Available also from: `https://books.google.cz/books?id=WgROkcmTxG4C`.

[22] TARSKI, Alfred. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.* 1955, vol. 5, no. 2, pp. 285–309. Available also from: `https://projecteuclid.org:443/euclid.pjm/1103044538`.

[23] BELOHLAVEK, Radim; TRNECKA, Martin. Basic Level of Concepts in Formal Concept Analysis. In: DOMENACH, Florent; IGNATOV, Dmitry I.; POELMANS, Jonas (eds.). *Formal Concept Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 28–44. ISBN 978-3-642-29892-9.

[24] WHITMAN, Philip M. Free Lattices. *Annals of Mathematics*. 1941, vol. 42, no. 1, pp. 325–330. ISSN 0003486X. Available also from: `http://www.jstor.org/stable/1969001`.

[25] WHITMAN, Philip M. Free Lattices II. *Annals of Mathematics*. 1942, vol. 43, no. 1, pp. pp. 104–115. ISSN 0003486X.

[26] FREESE, R.; FREESE, Ralph S.; JEŽEK, J.; NATION, J.B.; SOCIETY, American Mathematical. *Free Lattices*. American Mathematical Society, 1995. Advances in the Mathematical Sciences. ISBN 9780821803899.

Available also from: https://books.google.de/books?id=YdPxBwAAQBAJ.

[27] WILLE, Rudolf. Complete tolerance relations of concept lattices. In: EIGENTHALER, G.; AL., et (eds.). *Contributions to General Algebra*. Wien: Hölder-Pichler-Tempsky, 1985, vol. 3, pp. 397–415.

[28] BARTL, Eduard; KRUPKA, Michal. Residuated lattices of block relations: size reduction of concept lattices. *International Journal of General Systems*. 2016, vol. 45, no. 7-8, pp. 773–789. Available from DOI: 10.1080/03081079.2016.1144601 (http://dx.doi.org/10.1080/03081079.2016.1144601).

[29] MACHERY, Edouard. 100 Years of Psychology of Concepts: The Theoretical Notion of Concept and its Operationalization. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*. 2007, vol. 38, no. 1, pp. 63–84.

[30] MARGOLIS, E.; LAURENCE, S. *Concepts: Core Readings*. MIT Press, 1999. Bradford books. ISBN 9780262631938. Available also from: https://books.google.de/books?id=sj1gczQ-7K8C.

[31] FODOR, J.A.; GARRETT, M.F.; WALKER, E.C.T.; PARKES, C.H. Against definitions. *Cognition*. 1980, vol. 8, no. 3, pp. 263 –367. ISSN 0010-0277. Available also from: http://www.sciencedirect.com/science/article/pii/0010027780900086.

[32] PLATO. Five Dialogues: Euthyphro, Apology, Crito, Meno, Phaedo. 1981.

[33] BERTET, K.; MORVAN, M. Computing the sublattice of a lattice generated by a set of elements. In: *Proceedings of Third International Conference on Orders, Algorithms and Applications*. Montpellier, France, 1999.

TWO BASIC PROBLEMS OF INCREMENTAL CONSTRUCTION IN FORMAL CONCEPT ANALYSIS

Martin Kauer

Author Paper of Dissertation Thesis

Department of Computer Science

Faculty of Science

Palacky University Olomouc

2018

Uchazeč
Martin Kauer

Školitel
doc. RNDr. Michal Krupka, Ph.D.

Místo a termín obhajoby

Oponenti

S disertační prací a posudky se bude možné seznámit na katedře informatiky PřF UP, 17. listopadu 12, 771 46 Olomouc.

Synopsis – Formal Concept Analysis (FCA) is a field of applied mathematics based on formalization of the notion of concept from cognitive psychology and has been widely studied in the last several decades. From a description of objects by their features FCA derives a hierarchy of concepts which is formalized by a complete lattice called a concept lattice. We explore some fundamental aspects of FCA. First, we focus on incremental concept lattice construction and analysis of its basic step–removal of an incidence–and propose two algorithms for incremental concept lattice construction. Second, we study generated complete sublattices and show how their corresponding closed subrelations can be efficiently computed. Lastly, we investigate a new type of subrelations from which a new formal rectangle type arises, we provide motivation from cognitive psychology for it and propose a basic theorem for lattices of such rectangles.

Chapter 1

Preliminaries

In this chapter we provide a brief introduction to Formal Concept Analysis and other topics related to the content of the following chapters. We will not dwell on details here as all basic topics have been widely studied and all the details can be found in the cited sources.

1.1 Partially ordered sets, complete lattices and closures

Recall that a binary relation R on a set U is a *(partial) order*, if it satisfies reflexivity, antisymmetry and transitivity. We usually denote the order relation R by $\leq$, its inverse R^{-1} by $\geq$, and we write $u < v$ for $u \leq v$ and $u \neq v$. Moreover, if it holds either $u \leq v$ or $v \leq u$ for every $u, v \in V$, then we call $\leq$ a *total order*. A set U together with a partial order on U is called a *partially ordered set* or *poset* for short. For $u, w \in U$, u is called a *lower neighbor* of w, if $u < w$ and there is no element v fulfilling $u < v < w$. In this case, w is called an *upper neighbor* of u, we write $u \prec w$ and we can also read it as *w covers u*. Lastly, for $u, w \in U$ the set $[u, w] = \{v \in U \mid u \leq v \leq w\}$ is called a *closed interval*.

A poset U is called a *complete lattice* if each subset $P \subseteq U$ has the least upper bound (*supremum*) and the greatest lower bound (*infimum*). We denote these by $\bigvee P$ and $\bigwedge P$, respectively. An element $u \in U$ is called $\bigvee$-*irreducible* (resp. $\bigwedge$-*irreducible*) if it cannot be expressed as a supremum of strictly smaller (resp. greater) elements of U. If the element is not $\bigvee$-irreducible (resp. $\bigwedge$-irreducible) we call it $\bigvee$-*reducible* (resp. $\bigwedge$-*reducible*). A subset $V \subseteq U$ is called $\bigvee$-*dense* (resp. $\bigwedge$-*dense*), if each element $u \in U$ can be obtained as suprema (resp. infima) of some elements from V. A subset $V \subseteq U$ is a $\bigvee$-*subsemilattice* (resp. $\bigwedge$-*subsemilattice*, resp. *complete sublattice*) *of* U, if for each $P \subseteq V$ it holds $\bigvee P \in V$ (resp. $\bigwedge P \in V$, resp. $\{\bigvee P, \bigwedge P\} \subseteq V$), i.e. the set V is closed under arbitrary suprema (resp.

infima, resp. both previous). A subset $V \subseteq U$ is called an *order-embedded complete lattice*, if it is a complete lattice with the induced order (it does not have to be a sublattice). More details on order-embedded complete lattices can be found in [1].

For a subset $P \subseteq U$ we denote by $\mathrm{C}_{\bigvee} P$ the $\bigvee$-subsemilattice of U *generated by* P, i.e. the smallest (w.r.t. set inclusion) $\bigvee$-subsemilattice of U containing P. $\mathrm{C}_{\bigvee} P$ always exists and is equal to the intersection of all $\bigvee$-subsemilattices of U containing P. The $\bigwedge$-subsemilattice of U generated by P and the complete sublattice of U generated by P are defined similarly and are denoted by $\mathrm{C}_{\bigwedge} P$ and $\mathrm{C}_{\bigvee\bigwedge} P$, respectively. More on posets and lattices can be found in [2].

The operators $\mathrm{C}_{\bigvee}$, $\mathrm{C}_{\bigwedge}$ and $\mathrm{C}_{\bigvee\bigwedge}$ are closure operators on the set U. Recall that a *closure* (resp. *interior*) *operator on a set* X is a mapping $\mathrm{C} : 2^X \to 2^X$, where 2^X is the power-set of X (i.e. the set of all subsets of X), satisfying for all sets $A, A_1, A_2 \subseteq X$

1. $A \subseteq \mathrm{C}(A)$ (resp. $\mathrm{C}(A) \subseteq A$),
2. if $A_1 \subseteq A_2$, then $\mathrm{C}(A_1) \subseteq \mathrm{C}(A_2)$,
3. $\mathrm{C}(\mathrm{C}(A)) = \mathrm{C}(A)$.

An *isotone* (resp. *antitone*) *Galois connection* between two posets U and V is a pair of isotone (resp. antitone) functions $\langle f, g\rangle$ where $f : U \to V$, $g : V \to U$ satisfying

$$a \leq g(b) \text{ iff } f(a) \leq b \text{ (resp. } b \leq f(a)).$$

For an isotone (resp. antitone) Galois connection $\langle f, g\rangle$ the function composition $g \circ f$, given by $(g \circ f)(u) = g(f(u))$, is a closure operator on U and $f \circ g$ is an interior (resp. closure) operator on V. We define isotone (resp. antitone) Galois connection between two sets U and V as previously defined isotone (resp. antitone) Galois connection on their respective power-sets equipped with the subsethood ordering.

1.2 Formal Concept Analysis

Formal Concept Analysis was first introduced by R. Wille in [3] and has been widely studied ever since. The original motivation has its roots in human psychology and in the Port-Royal logic. Various generalizations and extensions of FCA were proposed over last years, see [4] for an overview. Our basic reference is [5].

A (*formal*) *context* is a triple $\langle X, Y, I\rangle$ where X is a set of objects, Y a set of attributes and $I \subseteq X \times Y$ a binary relation between X and Y specifying for each object its attributes.

For subsets $A \subseteq X$ and $B \subseteq Y$ we set

$$\begin{aligned} A^{\uparrow_I} &= \{y \in Y \mid \text{for each } x \in A \text{ it holds } \langle x, y\rangle \in I\}, \\ B^{\downarrow_I} &= \{x \in X \mid \text{for each } y \in B \text{ it holds } \langle x, y\rangle \in I\}. \end{aligned}$$

We call ${}^{\uparrow_I}, {}^{\downarrow_I}$ *derivation operators of I*. The pair $\langle {}^{\uparrow_I}, {}^{\downarrow_I}\rangle$ is an antitone Galois connection between the sets X and Y, therefore, the operator ${}^{\uparrow_I\downarrow_I}$ is a closure operator on X and the operator ${}^{\downarrow_I\uparrow_I}$ is a closure operator on Y.

A pair $\langle A, B\rangle$ satisfying $A^{\uparrow_I} = B$ and $B^{\downarrow_I} = A$ is called a (*formal*) *concept* of $\langle X, Y, I\rangle$. The set A is called *the extent of* $\langle A, B\rangle$, the set B *the intent of* $\langle A, B\rangle$. We denote $\mathrm{Ext}(X, Y, I)$ (resp. $\mathrm{Int}(X, Y, I)$) the set of all extents (resp. intents) of formal concepts of $\langle X, Y, I\rangle$. When there is no danger of confusion, we can use the term “an extent of I” instead of “the extent of a concept of $\langle X, Y, I\rangle$”, similarly for intents, and “a concept of I” instead of “a concept of $\langle X, Y, I\rangle$”. If the formal context is fixed we use terms “a concept”, “an extent” and “an intent”.

Several generalizations of the notion of formal concept have been proposed over the years. We call a pair $\langle A, A^{\uparrow_I}\rangle$ a $\sqcap$-*semiconcept* and a pair $\langle B^{\downarrow_I}, B\rangle$ a $\sqcup$-*semiconcept*. Combining the previous two notions we get a general notion of *semiconcept* [6]. We call a pair $\langle A, B\rangle$ satisfying $A^{\uparrow_I\downarrow_I} = B^{\downarrow_I}$ ($\Leftrightarrow B^{\downarrow_I\uparrow_I} = A^{\uparrow_I}$) a *protoconcept* [7]. Clearly, each semiconcept is also a potoconcept. These notions were motivated by their use for efficient description of formal concepts, namely, each protoconcept describes exactly one formal concept. Also, they were used to develop

Boolean Concept Logic [7]. The most general notion of *preconcept* is a pair $\langle A, B\rangle$ satisfying $A \subseteq B^{\downarrow_I}$ and $B \subseteq A^{\uparrow_I}$ [8, 9]. Preconcepts are just formal rectangles in our data and motivation for this notion comes from cognitive psychology, namely, from J. Piaget stating that concepts originate in child development from images, ideas and preconcepts [10].

A partial order $\leq$ on the set $\mathcal{B}(X, Y, I)$ of all formal concepts of $\langle X, Y, I\rangle$ is defined by $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (iff $B_2 \subseteq B_1$). $\mathcal{B}(X, Y, I)$ along with $\leq$ is called *the concept lattice of* $\langle X, Y, I\rangle$. By the basic theorem on concept lattices [5, Theorem 3], $\mathcal{B}(X, Y, I)$ is a complete lattice with infima and suprema given by

$$\bigwedge_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \bigcap_{\iota\in\mathcal{I}} A_\iota, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle, \tag{1.1}$$

$$\bigvee_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \bigcap_{\iota\in\mathcal{I}} B_\iota \right\rangle. \tag{1.2}$$

Moreover, a complete lattice V is isomorphic to $\mathcal{B}(X, Y, I)$ if and only if there are mappings $\gamma_I : X \to V$ and $\mu_I : Y \to V$ such that $\gamma_I(X)$ is $\bigvee$-dense in V, $\mu_I(Y)$ is $\bigwedge$-dense in V and xIy is equivalent to $\gamma_I(x) \leq \mu_I(y)$ for all $x \in X$ and all $y \in Y$. In particular, V is isomorphic to $\mathcal{B}(V, V, \leq)$.

Concepts $\mu_I(y) = \langle \{y\}^{\downarrow_I}, \{y\}^{\downarrow_I\uparrow_I}\rangle$ where $y \in Y$ are called *attribute concepts*, their extents are called *attribute extents* and intents are called *attribute intents*. According to the previous, each concept $\langle A, B\rangle$ is an infimum of some attribute concepts. Dually, concepts $\gamma_I(x) = \langle \{x\}^{\uparrow_I\downarrow_I}, \{x\}^{\uparrow_I}\rangle$ for $x \in X$ are called *object concepts*, they are $\bigvee$-dense in $\mathcal{B}(X, Y, I)$.

When the set of objects X and the set of attributes Y are fixed, we denote the concept lattice of $\langle X, Y, I\rangle$ just by $\mathcal{B}(I)$.

For any set of preconcepts $Q \subseteq \mathcal{B}(X, Y, I)$ we set

$$\bigsqcup Q = \bigcup \{A \times B \mid \langle A, B\rangle \in Q\}.$$

$\bigsqcup Q$ is the subrelation of I equal to the union of rectangles given by preconcepts from Q.

Chapter 2

Concept lattice construction by incidence removals

We open this chapter with a fundamental question about concept lattice construction, specifically, what effect does removing an incidence from a formal context have on its concept lattice. This question is known as the problem of "killing a cross" which was coined by R. Wille in the early days of FCA. Solving this problem is desirable not only from the theoretical but also from the practical point of view because it leads us to an efficient method of computing concept lattices of two very similar formal contexts. Moreover, it seems that any incremental method for concept lattice construction has this problem rooted into it.

Traditionally, we need to recompute whole concept lattice upon the slightest change in the input data. Although there have been several incremental algorithms introduced (see [11, 12, 13, 14, 15, 16] and also [17] for a comparison of some of the algorithms) they usually operate on object (resp. attribute) level. We focus on a finer approach and study the problem of removing a single incidence from a formal context. Our goal is to provide an analysis of this problem and based on it we propose two incremental algorithms for an efficient reconstruction of the concept lattice after the removal.

Throughout this chapter we consider a formal context $\langle X, Y, J\rangle$ which results from a formal context $\langle X, Y, I\rangle$ by removing a single incidence $\langle x_0, y_0\rangle$, i.e. $I = J \cup \langle x_0, y_0\rangle$ and $\langle x_0, y_0\rangle \notin J$. We denote the respective concept lattices by $\mathcal{B}(J)$ and $\mathcal{B}(I)$. Because we take the formal context $\langle X, Y, I\rangle$ as the starting point, we call it, and everything related to it (including derivation operators, $\mathcal{B}(I)$, $\ldots$), *initial.* Similarly, we call *final* everything related to the formal context $\langle X, Y, J\rangle$. We analyze necessary changes that are to be made in the initial concept lattice to obtain the final concept lattice.

2.1 Basic notions

It is obvious that not all initial concepts have to be influenced by the removal and there might be some concepts belonging into both $\mathcal{B}(I)$ and $\mathcal{B}(J)$. We call such concepts *steady* since they remain unchanged and do not require any reconstruction while computing $\mathcal{B}(J)$ from $\mathcal{B}(I)$. For this reason, it is important to identify steady concepts, and crucially, concepts that are not steady, *unsteady* for short. As it turns out, unsteady initial concepts form a bounded sublattice of $\mathcal{B}(I)$. This sublattice is not generally complete and it is equal to the closed interval $[\gamma_I(x_0), \mu_I(y_0)]$. Therefore, this sublattice is the only part of the concept lattice we need to focus on while pursuing our goal of computing $\mathcal{B}(J)$ based on $\mathcal{B}(I)$.

Lemma 1. *A concept $c \in \mathcal{B}(I)$ is unsteady iff $c \in [\gamma_I(x_0), \mu_I(y_0)]$.*

We introduce four *child operators* $_\square, _\boxtimes, ^\square, ^\boxtimes$ which we use throughout the rest of this chapter. The idea behind them is to relate concepts of $\mathcal{B}(I)$ to concepts of $\mathcal{B}(J)$ in a natural way that simplifies our analysis.

Definition 2 (child operators). For concepts $c = \langle A, B\rangle \in \mathcal{B}(I), d = \langle C, D\rangle \in \mathcal{B}(J)$ we set

$$\begin{aligned} c^\square &= \langle A^\square, B^\square\rangle = \langle A^{\uparrow_J\downarrow_J}, A^{\uparrow_J}\rangle, & c_\square &= \langle A_\square, B_\square\rangle = \langle B^{\downarrow_J}, B^{\downarrow_J\uparrow_J}\rangle, \\ d^\boxtimes &= \langle C^\boxtimes, D^\boxtimes\rangle = \langle D^{\downarrow_I}, D^{\downarrow_I\uparrow_I}\rangle, & d_\boxtimes &= \langle C_\boxtimes, D_\boxtimes\rangle = \langle C^{\uparrow_I\downarrow_I}, C^{\uparrow_I}\rangle. \end{aligned}$$

Evidently, $c^\square, c_\square \in \mathcal{B}(J)$ and $d^\boxtimes, d_\boxtimes \in \mathcal{B}(I)$. $c^\square$ (resp. $c_\square$) is called *the upper* (resp. *lower*) *child of* c. It holds $d^\boxtimes = d_\boxtimes$ and it is the (unique) concept from $\mathcal{B}(I)$ containing, as a rectangle, the rectangle represented by d.

Lemma 3 (compound child operators). *The mappings $c \mapsto c^{\square\boxtimes}$ and $d \mapsto d^{\boxtimes\square}$ are closure operators and the mappings $c \mapsto c_{\square\boxtimes}$ and $d \mapsto d_{\boxtimes\square}$ are interior operators.*

2.2 Computing the final concepts

Building on notions and observations from the previous section, we will now turn our attention to the problem of computing concepts of $\mathcal{B}(J)$ given concepts of $\mathcal{B}(I)$.

Theorem 4. *An unsteady concept $d \in \mathcal{B}(J)$ is a (upper or lower) child of exactly one concept $c \in \mathcal{B}(I)$. This concept is unsteady and satisfies $c = d^{\boxtimes} = d_{\boxtimes}$.*

The theorem leads to the following simple way of constructing $\mathcal{B}(J)$ from $\mathcal{B}(I)$. For each $c \in \mathcal{B}(I)$ the following has to be done:

1. If c is steady, then it has to be added to $\mathcal{B}(J)$.
2. If c is not steady, then each its unsteady child, i.e. each unsteady element of $\{c^{\square}, c_{\square}\}$, has to be added to $\mathcal{B}(J)$.

This method ensures that all proper elements will be added to $\mathcal{B}(J)$ (i.e. no element will be omitted) and each element will be added exactly once.

The following lemma shows a simple way of determining whether a child of an unsteady initial concept is steady. It also describes the role of fixpoints of the compound child operators.

Lemma 5. *Let c be an unsteady concept of $\mathcal{B}(I)$. Then*

– *$c^{\square}$ is unsteady iff c is a fixpoint of ${}^{\square\boxtimes}$,*

– *$c_{\square}$ is unsteady iff c is a fixpoint of ${}_{\square\boxtimes}$.*

The proposed method is utilized in Algorithm 1 which computes the final concepts from the initial ones but does not take into the account the ordering.

Time complexity of Algorithm 1 is clearly $O(|\mathcal{B}(I)||X||Y|)$ in the worst case scenario. Indeed, the number of unsteady concepts is at most equal to $|\mathcal{B}(I)|$ and the computation of operators ${}^{\square\boxtimes}, {}_{\square\boxtimes}$ can be done in $O(|X| \cdot |Y|)$ time. It is worth noting that the time complexity is heavily affected by the size of the interval $[\gamma_I(x_0), \mu_I(y_0)]$ which can be much smaller than the size of the whole lattice $\mathcal{B}(I)$.

Algorithm 1 Transforming concepts of $\mathcal{B}(I)$ into concepts of $\mathcal{B}(J)$.

procedure TRANSFORMCONCEPTS($\mathcal{B}(I)$)
    $\mathcal{B}(J) \leftarrow \mathcal{B}(I)$;
    for all $c = \langle A, B \rangle \in [\gamma_I(x_0), \mu_I(y_0)]$ **do**
        $\mathcal{B}(J) \leftarrow \mathcal{B}(J) \setminus \{c\}$;
        if $c = c_{\Box\boxtimes}$ **then**
            $\mathcal{B}(J) \leftarrow \mathcal{B}(J) \cup \{c_{\Box}\}$;
        if $c = c^{\Box\boxtimes}$ **then**
            $\mathcal{B}(J) \leftarrow \mathcal{B}(J) \cup \{c^{\Box}\}$;
    return $\mathcal{B}(J)$;

There are several optimizations that we can apply to Algorithm 1, e.g. in order to compute results of derivation operators, we can use already provided extents and intents. Also, tests like $c = c^{\Box\boxtimes}$ can actually be performed without computing the corresponding compound child operators.

2.3 Computing the final lattice

In order to analyze structural changes in a concept lattice after removal of an incidence we need to investigate additional properties of the closure operator ${}^{\Box\boxtimes}$ and the interior operator ${}_{\Box\boxtimes}$. We focus mostly on their fixpoints.

Lemma 6. *Each steady concept is a fixpoint of both* ${}^{\Box\boxtimes}$ *and* ${}_{\Box\boxtimes}$*.*

Lemma 7. *For any* $c \in \mathcal{B}(I)$*, each concept from* $[c_{\Box\boxtimes}, c^{\Box\boxtimes}] \setminus \{c\}$ *is steady.*

The following lemma shows an important property of sets of fixpoints of compound child operators in the unsteady initial sublattice. Namely, the set of fixpoints of ${}^{\Box\boxtimes}$ is a lower set whereas the set of fixpoints of ${}_{\Box\boxtimes}$ is an upper set.

Lemma 8. *Let* $c \in \mathcal{B}(I)$ *be an unsteady concept. If* c *is a fixpoint of* ${}^{\Box\boxtimes}$*, then each* $c' \leq c$ *is also a fixpoint of* ${}^{\Box\boxtimes}$*. If* c *is a fixpoint of* ${}_{\Box\boxtimes}$*, then each* $c' \geq c$ *is also a fixpoint of* ${}_{\Box\boxtimes}$*.*

The above results provide an interesting insight into the structure of our fixpoints. This helps us restrict possible cases that we need to take into consideration when designing Algorithm 2 which computes the lattice $\mathcal{B}(J)$. The time complexity of Algorithm 2 is in the worst case $O(|\mathcal{B}(I)| \cdot |X|^2 \cdot |Y|)$.

In Algorithm 2 we assume that following functions are already defined:

- $UpperNeighbors(c)$ - returns upper neighbors of c;
- $LowerNeighbors(c)$ - returns lower neighbors of c;
- $Link(c_1, c_2)$ - introduces neighborhood relationship between c_1 and c_2;
- $Unlink(c_1, c_2)$ - cancels neighborhood relationship between c_1 and c_2.

Example 9. An execution of Algorithm 2 on the concept lattice of the formal context from Fig. 2.1 is depicted in Fig. 2.2-2.6. Each picture captures the state after transformation of an unsteady concept. Unsteady concepts are drawn with dashed circles.

2.4 Experiments

We provide some insight into performance of our algorithms as well as some experimental evaluation. Comparing our algorithms to the traditional algorithms that recompute the whole final lattice does not make much sense as the difference proved to be immense in our preliminary experiments. This is caused by the obvious advantage of incremental methods as we usually need to recompute only a small portion of the initial lattice. Moreover, we can make use of previously computed concepts instead of just discarding them. Performance of our algorithms depends heavily on the size of unsteady initial interval. Hence, we provide experiments focusing on sizes of intervals corresponding to selected incidences. We used real world datasets as well as synthetic data. The former were taken from UC Irvine Machine Learning Repository[1] with an exception of dataset Drinks [18]. In order

[1] http://archive.ics.uci.edu/ml/index.php

Algorithm 2 Transforming the lattice $\mathcal{B}(I)$ into the lattice $\mathcal{B}(J)$.

procedure LinkIfNeeded(c_1, c_2)
  if $\nexists k \in \mathcal{B}(I) : c_1 < k < c_2$ **then**
    $Link(c_1, c_2)$;

procedure SplitConcept(c)
  $d_1 = c_{\Box}$; $d_2 = c^{\Box}$;
  $Link(d_1, d_2)$;
  for all $u \in UpperNeighbors(c)$ **do**
    $Unlink(c, u)$; $Link(d_2, u)$;
  for all $l \in LowerNeighbors(c)$ **do**
    $Unlink(l, c)$; $Link(l, d_1)$;
  for all $u \in UpperNeighbors(c)$ **do**
    if $u \neq u^{\Box\boxtimes}$ **then**
      $Unlink(d_2, u)$; $Link(d_1, u)$; $LinkIfNeeded(d_2, u^{\Box\boxtimes})$;
  for all $l = \langle C, D \rangle \in LowerNeighbors(c)$ **do**
    if $y_0 \notin D$ **then**
      $Unlink(l, d_1)$; $Link(l, d_2)$; $LinkIfNeeded(l_{\boxtimes\Box}, d_1)$;
  return d_1, d_2;

procedure RelinkReducedIntent(c)
  for all $u = \langle C, D \rangle \in UpperNeighbors(c)$ **do**
    if $u \neq u^{\Box\boxtimes}$ **then**
      $Unlink(c, u)$;
      $LinkIfNeeded(c_{\Box\boxtimes}, u)$; $LinkIfNeeded(c, u^{\Box\boxtimes})$;

procedure UnlinkVanishedConcept(c)
  for all $u \in UpperNeighbors(c)$ **do**
    $Unlink(c, u)$; $LinkIfNeeded(c_{\Box\boxtimes}, u)$;
  for all $l \in LowerNeighbors(c)$ **do**
    $Unlink(l, c)$;

procedure TransformConceptLattice($\mathcal{B}(I)$)
  for all $c = \langle A, B \rangle \in [\gamma_I(x_0), \mu_I(y_0)]$ from least to largest w.r.t. $\sqsubseteq$ **do**
    if $c = c^{\Box\boxtimes}$ **and** $c = c_{\Box\boxtimes}$ **then** ▷ Concept will split.
      $\mathcal{B}(I) \leftarrow \mathcal{B}(I) \setminus \{c\}$;
      $\mathcal{B}(I) \leftarrow \mathcal{B}(I) \cup SplitConcept(c)$;
    else if $c \neq c^{\Box\boxtimes}$ **and** $c = c_{\Box\boxtimes}$ **then** ▷ Extent will be smaller.
      $A \leftarrow A \setminus \{x_0\}$;
    else if $c = c^{\Box\boxtimes}$ **and** $c \neq c_{\Box\boxtimes}$ **then** ▷ Intent will be smaller.
      $RelinkReducedIntent(c)$;
      $B \leftarrow B \setminus \{y_0\}$;
    else if $c \neq c^{\Box\boxtimes}$ **and** $c \neq c_{\Box\boxtimes}$ **then** ▷ Concept will vanish.
      $\mathcal{B}(I) \leftarrow \mathcal{B}(I) \setminus \{c\}$;
      $UnlinkVanishedConcept(c)$;

I	y_0	y_1	y_2	y_3	y_4
x_0	•	×		×	
x_1	×	×	×		
x_2		×			
x_3	×			×	×
x_4				×	

Figure 2.1: A formal context with four unsteady concepts.

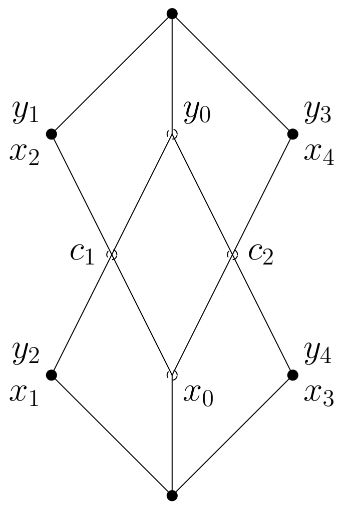


Figure 2.2: The initial state of the concept lattice.

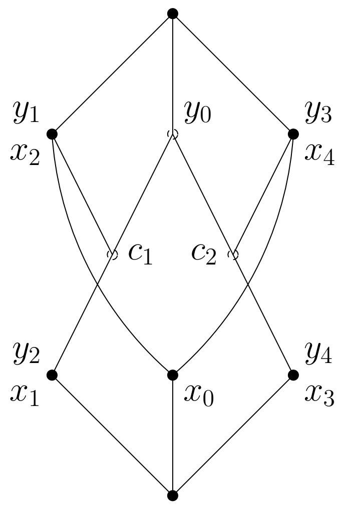


Figure 2.3: The transformation of $\gamma_I(x_0)$.

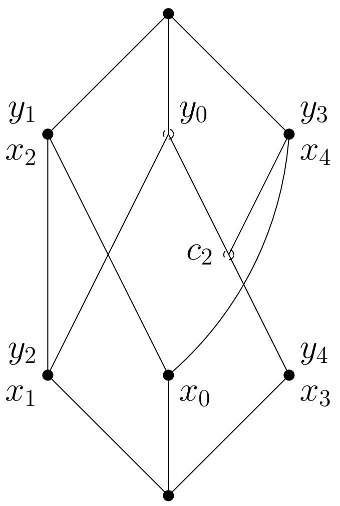


Figure 2.4: The concept c_1 vanishes.

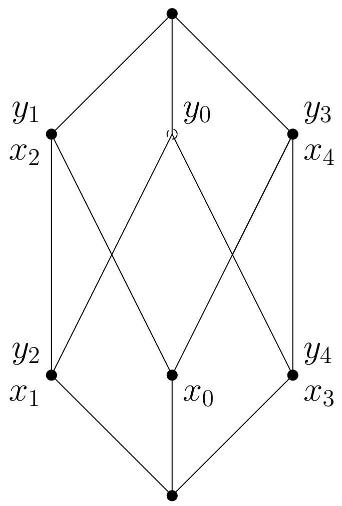


Figure 2.5: The concept c_2 vanishes.

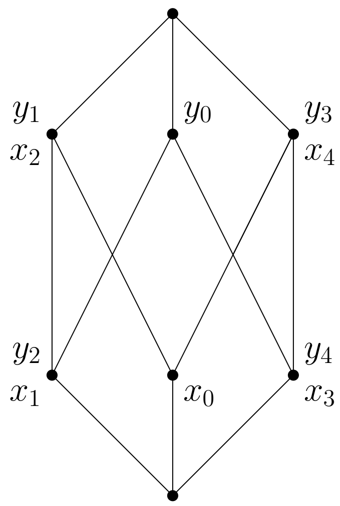


Figure 2.6: The transformation of $\mu_I(y_0)$.

to obtain bivalent attributes we rescaled the attributes (as usual) using nominal scaling [5]. The details of used datasets can be found in Table 2.1.

For our experiments we selected a random incidence 10000 times and recorded the size of the corresponding interval. We provide maximal and average sizes as percentages of size of the whole concept lattice.

The results on real world datasets can be found in Table 2.2. We can see that

	Objects	Attributes	Scaled Attrs.	Concepts
Mushrooms	8124	22	119	238710
Nursery	12960	8	32	183079
Post	90	9	25	1523
Zoo	101	17	28	379
Drinks	68	25	25	320

Table 2.1: Properties of datasets used in our experiments.

for the larger datasets (Mushrooms and Nursery) the average size of the selected interval is well below 0.5%. For the smaller datasets, it is significantly larger although still within the 10%.

	Max size (%)	Avg size (%)
Mushrooms	4.46	0.46
Nursery	0.14	0.11
Post	8.27	3.92
Zoo	33.51	6.93
Drinks	51.56	8.83

Table 2.2: Sizes of intervals for real world data corresponding to randomly selected incidences.

The synthetic datasets were randomly generated with a fixed density (2%, 5%, 10%, 15%, 20%, 25%) and consisted of 500 objects and 100 attributes. The results can be seen in Table 2.3. All the recorded sizes, except for one, were withing 1%. Interestingly, both maximal and average size seems to be decreasing w.r.t. increasing density.

The results suggest that an interval corresponding to a randomly selected incidence usually contains only a fraction of concepts w.r.t. the whole concept lattice.

Density (%)	Max size (%)	Avg size (%)
2	1.73	0.47
5	0.86	0.25
10	0.73	0.18
15	0.75	0.15
20	0.72	0.14
25	0.78	0.14

Table 2.3: Sizes of intervals for synthetic data corresponding to randomly selected incidences.

Algorithm 1 can perform very well, especially if we take into consideration that the optimized version of the algorithm in fact computes just two derivation operators and two set comparisons in each iteration.

2.5 Extensions

In this section we give a brief and informal description of selected extensions of our method. A natural extension of our method for removing an arbitrary number of incidences stems from repeated runs of the presented algorithms, i.e. removing incidences one by one. We obtain a sequence of formal contexts $I = I_0 \to I_1 \to I_2 \to \cdots \to I_n = J$ where $I_k = I_{k+1} \cup \{\langle x_{i_{k+1}}, y_{j_{k+1}} \rangle\}$. Evidently, there exist child operators between each pair of adjacent formal contexts. We call initial (resp. final) concept unsteady if it is unsteady w.r.t. any removal step. By removing incidences one by one we are able to remove arbitrary number of incidences from any formal context.

It is also possible to remove an arbitrary number of incidences from an object in a single step. In fact, the presented method works practically as is by taking $I = J \cup \{\langle x_i, y_{j_1} \rangle, \langle x_i, y_{j_2} \rangle, \ldots, \langle x_i, y_{j_n} \rangle\}$ (J does not contain any of the incidences we are removing) and unsteady concepts to be from the union of all intervals determined by the removed incidences. To see this is indeed the case, consider removing

$$\begin{tikzcd}
I = I_0 \arrow[r, bend left, "{\Box_1, \Box_1}"] & I_1 \arrow[l, bend left, "{\boxtimes_1, \boxtimes_1}"] \arrow[r, bend left, "{\Box_2, \Box_2}"] & I_2 \arrow[l, bend left, "{\boxtimes_2, \boxtimes_2}"] \arrow[r, bend left, "{\Box_3, \Box_3}"] & \cdots \arrow[l, bend left, "{\boxtimes_3, \boxtimes_3}"] \arrow[r, bend left, "{\Box_n, \Box_n}"] & I_n = J \arrow[l, bend left, "{\boxtimes_n, \boxtimes_n}"]
\end{tikzcd}$$

Figure 2.7: Removing n incidences one by one. We have $I_k = I_{k+1} \cup \{\langle x_{i_{k+1}}, y_{j_{k+1}} \rangle\}$.

the incidences one by one. We observe that if an unsteady (w.r.t. any removal) concept c has two unsteady children, one of them will always remain steady (w.r.t. all consecutive removals) and the other can have at most one unsteady (w.r.t. any removal) child. We can also show that every concept of $\mathcal{B}(I)$ can be computed by the same formulae as in the definition of the child operators. The case for removing an arbitrary number of incidences from an attribute in a single step is dual.

We can go one step further and remove an arbitrary preconcept at once. This is now an easy extension of the case for removing an arbitrary number of incidences from a single object (resp. attribute) and its justification is very similar to it.

Moving onwards, we consider removing arbitrary incidences at once based only on the concepts and derivation operators of the initial and final formal context. We can easily see that this is not possible. Consider a contranominal scale of size n, i.e. a formal context $\langle \{1, 2, \ldots, n\}, \{1, 2, \ldots, n\}, \neq \rangle$. Evidently, given such context we would need to be able to derive 2^n final concepts from a single initial concept.

Having found the limit for our extensions, we now turn our attention to the possibility of concurrently removing two–or more–incidences. The exact condition for running our algorithms concurrently without any problem seems to be computationally more expensive and might not be worth using. Nevertheless, it might be useful to state it properly: unsteady intervals together with the set of the results of application of compound child operators to the concepts of the intervals should have an empty intersection.

Based on the ideas above we can now summarize what kind of changes to a formal context we are able to handle with our method.

- Removal of an arbitrary preconcept at once.
- Addition of an object (resp. attribute) can be achieved by adding full row (resp. column) to the underlying formal context. Evidently, this has no effect on the structure of the concept lattice (it just adds the object to all concepts). At last, we can remove unwanted incidences at once.
- Removal of an object (resp. attribute) can be done by removing all its incidences at once and afterwards removing the resulted empty row (resp. column). This is easy, just check the greatest (resp. smallest) concept.
- Arbitrary change in an object (resp. attribute) is just a combination of the cases above (removal and addition).

2.6 Discussion and related work

We analyzed changes in a concept lattice caused by removal of a single incidence from the associated formal context. We showed selected theoretical results and presented two algorithms with time complexities $O(|\mathcal{B}| \cdot |X| \cdot |Y|)$ (Alg. 1; without structure information) and $O(|\mathcal{B}| \cdot |X|^2 \cdot |Y|)$ (Alg. 2; with structure information).

There exist several algorithms for incremental computation of concept lattice [11, 12, 13, 14, 15, 16] and they are usually based on adding/removing objects or attributes. Our approach is new in that we focus on the finer problem of recomputing a concept lattice after the removal of just one incidence. We believe that this problem is inherently rooted into every incremental algorithm for concept lattice construction. Amongst mentioned algorithms, the algorithm proposed by Nourine and Raynaud in [15] has the lowest time complexity of $O((|Y| + |X|) \cdot |X| \cdot |\mathcal{B}|)$. However, experiments presented in [12] indicate that this algorithm sometimes performs slower than some algorithms with time complexity $O(|\mathcal{B}| \cdot |X|^2 \cdot |Y|)$. In the case of our algorithms, presented experiments indicate that the size of the interval of unsteady concepts is usually relatively small, which substantially reduces the overall processing time of our algorithms.

We also looked into some possible extensions of our method and showed how it can be used to remove an arbitrary number of incidences from a single object (resp. attribute) at once. It turns out that we are also able to remove an arbitrary preconcept at once. Moreover, we are able to do it without any additional overhead. There is also the possibility of chaining applications of our algorithms providing a method to remove arbitrary incidences from a formal context. Utilizing these ideas we arrive at a general method for updating a concept lattice upon an arbitrary change in the underlying formal context.

The dual problem, adding an incidence to a formal context, does not share some nice properties as the problem of removing, e.g. the set of all unsteady final concepts has a more general structure than a closed interval and also not all unsteady initial concepts can be computed by the child operators.

Chapter 3

On sublattices and subrelations

One of basic theoretical results of FCA states a correspondence between closed subrelations of a formal context and complete sublattices of the corresponding concept lattice [5]. In this chapter, we study the related problem of constructing the closed subrelation for a complete sublattice generated by given set of elements.

A subrelation $J \subseteq I$ is called a *closed subrelation of* I [5] if each concept of $\langle X, Y, J\rangle$ is also a concept of $\langle X, Y, I\rangle$. There is the following correspondence between closed subrelations of I and complete sublattices of $\mathcal{B}(X, Y, I)$. For each closed subrelation $J \subseteq I$, $\mathcal{B}(X, Y, J)$ is a complete sublattice of $\mathcal{B}(X, Y, I)$, and for each complete sublattice $V \subseteq \mathcal{B}(X, Y, I)$ there exists a closed subrelation $J \subseteq I$ such that $V = \mathcal{B}(X, Y, J)$.

Throughout this chapter we consider a formal context $\langle X, Y, I\rangle$, its concept lattice $\mathcal{B}(X, Y, I)$, a set of concepts $P \subseteq \mathcal{B}(X, Y, I)$ and a complete sublattice $V \subseteq \mathcal{B}(X, Y, I)$ generated by the set P (i.e. $V = \mathrm{C}_{\bigvee\bigwedge} P$). Elements of P are called *generators*. We already know that there exists a closed subrelation $J \subseteq I$ with the concept lattice $\mathcal{B}(X, Y, J)$ equal to V. We show a method of constructing J without the need of constructing $\mathcal{B}(X, Y, I)$ first. We propose an efficient algorithm implementing the method and show illustrative examples and results of experiments.

We also investigate additional related problems. For an arbitrary subrelation $K \subseteq I$, we study the possibility of finding the least closed subrelation containing K. The problem does not always have a solution as the system of closed subrelations of I is not a closure system. We identify an important type of subrelations for which the solution always exists. We also provide some results on closed subrelations $J \supseteq K$ and their associated concept lattices.

From the investigation of subrelations a new type of formal rectangle arises. Such rectangles might serve as formalization of some notions from the field of cognitive psychology. We investigate properties of such formal rectangles and outline their relation to already known types. We also show that they are related to block relations. Lastly, we show how they can be structured into a lattice and we propose a basic theorem for lattices of such rectangles.

3.1 Closed subrelations for generated sublattices

Let us have a formal context $\langle X, Y, I\rangle$ and a subset P of its concept lattice. Denote by V the complete sublattice of $\mathcal{B}(X, Y, I)$ generated by P (i.e. $V = \mathrm{C}_{\bigvee\bigwedge} P$). Our goal is to find, without computing the lattice $\mathcal{B}(X, Y, I)$, the closed subrelation $J \subseteq I$ whose concept lattice $\mathcal{B}(X, Y, J)$ is equal to V.

If $\mathcal{B}(X, Y, I)$ is finite, V can be obtained by alternating applications of the closure operators $\mathrm{C}_{\bigvee}$ and $\mathrm{C}_{\bigwedge}$ on P: we set $V_1 = \mathrm{C}_{\bigvee} P$, $V_2 = \mathrm{C}_{\bigwedge} V_1$, …, and, generally

$$V_i \quad = \quad \begin{cases} \mathrm{C}_{\bigvee} V_{i-1} & \text{for odd } i, \\ \mathrm{C}_{\bigwedge} V_{i-1} & \text{for even } i. \end{cases} \tag{3.1}$$

The sets V_i are $\bigvee$-subsemilattices (for odd i) resp. $\bigwedge$-subsemilattices (for even i) of $\mathcal{B}(X, Y, I)$. Once $V_i = V_{i-1}$, we have the complete sublattice V.

$$\mathrm{C}_{\bigvee} P = V_1 \xrightarrow{\mathrm{C}_{\bigwedge}} V_2 \xrightarrow{\mathrm{C}_{\bigvee}} V_3 \overset{\cdots}{\rightsquigarrow} V_{i-1} = V_i = V$$

Figure 3.1: One way to compute a (finite) complete sublattice generated by a set P stems from alternating computations of closures $\mathrm{C}_{\bigvee}$ and $\mathrm{C}_{\bigwedge}$ as given by (3.1).

We always consider sets V_i together with the appropriate restriction of the ordering on $\mathcal{B}(X, Y, I)$. For each $i > 0$, V_i is a complete lattice that is order-embedded into $\mathcal{B}(X, Y, I)$ (but it is generally not a complete sublattice of $\mathcal{B}(X, Y, I)$).

In what follows, we construct formal contexts with concept lattices isomorphic to the complete lattices V_i, $i > 0$. By doing so, we obtain a sequence of formal context as shown in Fig. 3.2. We start by finding a formal context corresponding to the complete lattice V_1. Let $K_1 \subseteq P \times Y$ be given by

$$\langle\langle A, B\rangle, y\rangle \in K_1 \quad \text{iff} \quad y \in B. \tag{3.2}$$

As we can see, rows in the context $\langle P, Y, K_1\rangle$ are exactly intents of concepts from P.

$$\begin{tikzcd}
\mathrm{C}_{\bigvee} P = V_1 \arrow[r, "\mathrm{C}_{\bigwedge}"] \arrow[d, phantom, "\wr\|"] & V_2 \arrow[r, "\mathrm{C}_{\bigvee}"] \arrow[d, phantom, "\wr\|"] & V_3 \arrow[r, squiggly, "\cdots"] \arrow[d, phantom, "\wr\|"] & V_{i-1} = V_i = V \arrow[d, phantom, "\wr\|"] \\
\mathcal{B}(K_1) \arrow[r] \arrow[d, leftrightarrow] & \mathcal{B}(K_2) \arrow[r] \arrow[d, leftrightarrow] & \mathcal{B}(K_3) \arrow[r, squiggly, "\cdots"] \arrow[d, leftrightarrow] & \mathcal{B}(K_{i-1}) = \mathcal{B}(K_i) \arrow[d, leftrightarrow] \\
K_1 \arrow[r] & K_2 \arrow[r] & K_3 \arrow[r, squiggly, "\cdots"] & K_{i-1} = K_i
\end{tikzcd}$$

Figure 3.2: We compute a sequence of formal contexts K_i ($i > 0$) in order to obtain the closed subrelation with concept lattice equal to the complete sublattice generated by a set of concepts P.

Next, we describe formal contexts for complete lattices V_i, $i > 1$. All of the contexts are of the form $\langle X, Y, K_i\rangle$, i.e. they have the set X as the set of objects and the set Y as the set of attributes (the relation K_1 is different in this regard). The relations K_i for $i > 1$ are defined in a recursive manner:

$$\text{for } i > 1,\ \langle x, y\rangle \in K_i \quad \text{iff} \quad \begin{cases} x \in \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I} & \text{for even } i, \\ y \in \{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}\uparrow_I} & \text{for odd } i. \end{cases} \tag{3.3}$$

Lemma 10. *For each $i > 1$,*

1. $K_i \subseteq I$,
2. $K_i \subseteq K_{i+1}$.

Remark 11. Informally, we can think of creating the sequence of formal contexts K_i as follows: for even i, we obtain the formal context K_i by stretching attribute intents of K_{i-1} over I; if i is odd, we stretch object extents of K_{i-1} over I. Another way of thinking about what happens at each iteration is the following: for even i we are fixing extents and for odd i we are fixing intents. Thinking in these informal terms could help with understanding the presented method.

Lemma 12. *For each $i > 0$, the concept lattice $\mathcal{B}(P, Y, K_i)$ (for $i = 1$) resp. $\mathcal{B}(X, Y, K_i)$ (for $i > 1$) and the complete lattice V_i are isomorphic. The isomorphism is given by $\langle B^{\downarrow_{K_i}}, B\rangle \mapsto \langle B^{\downarrow_I}, B\rangle$ if i is odd and by $\langle A, A^{\uparrow_{K_i}}\rangle \mapsto \langle A, A^{\uparrow_I}\rangle$ if i is even.*

If X and Y are finite, then 2. of Lemma 10 implies that there is a number $n > 1$ such that $K_{n+1} = K_n$. Denote this relation by J. According to Lemma 12, there are two isomorphisms of the concept lattice $\mathcal{B}(X, Y, J)$ and $V_n = V_{n+1} = V$. These two isomorphisms coincide and in fact $\mathcal{B}(X, Y, J) = V$.

Corollary 13. *The relation $J \subseteq I$ as defined above is a closed subrelation of I and $\mathcal{B}(X, Y, J) = V$.*

Algorithm 3 Computing the closed subrelation J.

Require: formal context $\langle X, Y, I\rangle$, subset $P \subseteq \mathcal{B}(X, Y, I)$
Ensure: the closed subrelation $J \subseteq I$ whose concept lattice is equal to $\mathrm{C}_{\bigvee\bigwedge} P$

```
Generate(K_1)                                        ▷ K_1 is given by (3.2)
procedure Generate(J)
    i ← 1
    repeat
        L ← J
        i ← i + 1
        if i is even then
            J ← {⟨x, y⟩ ∈ I | x ∈ {y}^{↓L ↑L ↓I}}
        else
            J ← {⟨x, y⟩ ∈ I | y ∈ {x}^{↑L ↓L ↑I}}
    until i > 2 & J = L
    return J
```

Lemma 14. *Algorithm 3 is correct and terminates after at most* $\max(|I|+1,2)$ *iterations.*

Example 15. We now demonstrate execution of Alg. 3. Let $\langle X, Y, I\rangle$ be the formal context from Fig. 3.3 (left). The associated concept lattice $\mathcal{B}(X,Y,I)$ is depicted in Fig. 3.3 (right). Let $P = \{c_1, c_2, c_3\}$ where $c_1 = \langle\{x_1\},\{y_1,y_4\}\rangle$, $c_2 = \langle\{x_1,x_2\},\{y_1\}\rangle$, $c_3 = \langle\{x_2,x_5\},\{y_2\}\rangle$ are concepts from $\mathcal{B}(X,Y,I)$. These concepts are depicted in Fig. 3.3 by filled dots.

I	y_1	y_2	y_3	y_4	y_5
x_1	×			×	
x_2	×	×	×		
x_3			×		×
x_4			×		
x_5		×			

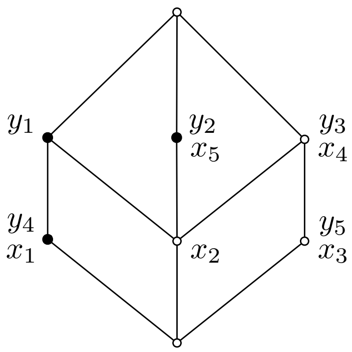


Figure 3.3: Formal context $\langle X, Y, I\rangle$ (left) and concept lattice $\mathcal{B}(X,Y,I)$ together with a subset $P \subseteq \mathcal{B}(X,Y,I)$, depicted by filled dots (right).

K_1	y_1	y_2	y_3	y_4	y_5
c_1	×			×	
c_2	×				
c_3		×			

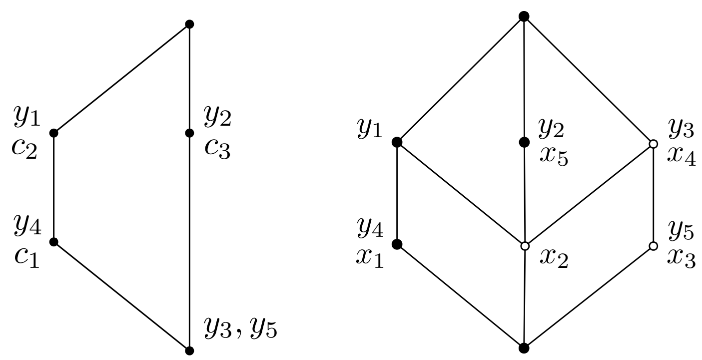


Figure 3.4: Formal context $\langle P, Y, K_1\rangle$ (left), the concept lattice $\mathcal{B}(P,Y,K_1)$ (center) and the $\bigvee$-subsemilattice $\mathrm{C}_{\bigvee}P \subseteq \mathcal{B}(X,Y,I)$, isomorphic to $\mathcal{B}(P,Y,K_1)$, depicted by filled dots (right).

K_2	y_1	y_2	y_3	y_4	y_5
x_1	×			×	
x_2	×	×	•		
x_3			•		•
x_4			•		
x_5		×			

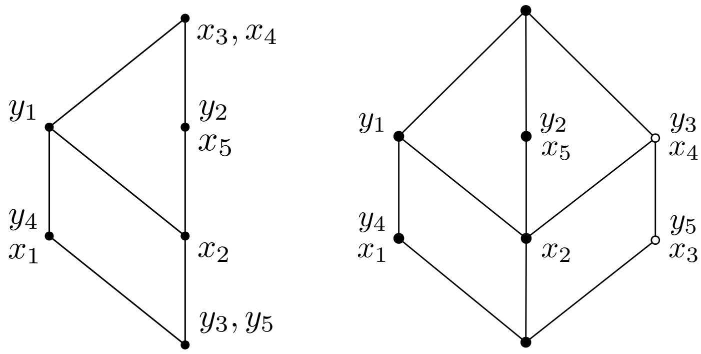


Figure 3.5: Formal context $\langle X, Y, K_2\rangle$ (left), the concept lattice $\mathcal{B}(X, Y, K_2)$ (center) and the $\bigwedge$-subsemilattice $V_2 = \mathrm{C}_{\bigwedge} V_1 \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(X, Y, K_2)$, depicted by filled dots (right). Elements of $I \setminus K_2$ are depicted by dots in the table.

K_3	y_1	y_2	y_3	y_4	y_5
x_1	×			×	
x_2	×	×	×		
x_3			•		•
x_4			•		
x_5		×			

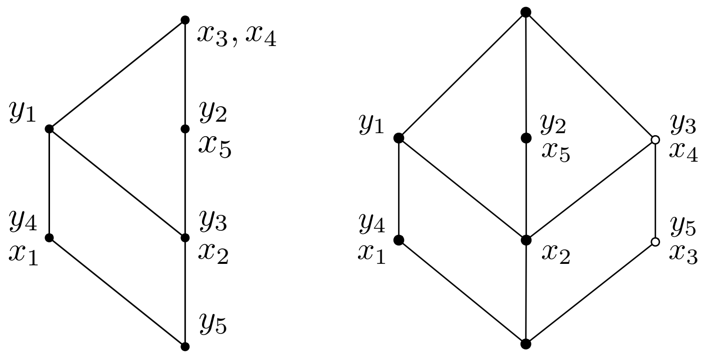


Figure 3.6: Formal context $\langle X, Y, K_3\rangle$ (left), the concept lattice $\mathcal{B}(X, Y, K_3)$ (center) and the $\bigvee$-subsemilattice $V_3 = \mathrm{C}_{\bigvee} V_2 \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(X, Y, K_3)$, depicted by filled dots (right). Elements of $I \setminus K_3$ are depicted by dots in the table. As $K_3 = K_4 = J$, it is a closed subrelation of I and $V_4 = \mathrm{C}_{\bigwedge} V_3 = V_3 = V$.

In our method, the relation K_1 differs from the other relations K_i for $i > 1$ in that it is a subset of $P \times Y$ instead of $X \times Y$. In the last part of this section, we present a modification of the method which replaces K_1 with a subrelation $K_1' \subseteq I$ given by

$$K_1' = \bigsqcup P, \tag{3.4}$$

where $\bigsqcup P$ is a union of rectangles determined by elements of P. We can easily show that after this replacement our method gives the same result as before. Note that the subrelation K_1' lacks one important property of the relation K_1. Namely, intents of K_1' need not be intents of I. Consequently, the concept lattice of K_1' does not have to be isomorphic to the complete lattice V_1, the property the relation K_1 has due to Lemma 12.

Algorithm 4 Computing the closed subrelation J, alternative version.

Require: formal context $\langle X, Y, I \rangle$, subset $P \subseteq \mathcal{B}(X, Y, I)$
Ensure: the closed subrelation $J \subseteq I$ whose concept lattice is equal to $\mathrm{C}_{\bigvee\bigwedge} P$
  return GENERATE($\bigsqcup P$)

3.2 Experiments

Time complexity of Alg. 3 (and its variant Alg. 4) is clearly polynomial w.r.t. $|X|$ and $|Y|$. In Lemma 14 we showed that the number of iterations is less than or equal to $|I|+1$. Our experiments indicate that this number might be much smaller in the practice. We used synthetic as well as real world datasets. More details about used datasets can be found in Section 2.4 and in Table 2.1.

The first batch of experiments involved larger real world datasets (Mushrooms and Nursery). The size of the set of generators P was given by percentage of corresponding number of concepts. For each size of P we randomly selected its elements 1000 times, ran our algorithm, and measured the number n of iterations, after which the algorithm terminated. In both cases we recognized that the peak in average as well as in the maximal number of iterations was achieved for a very small sizes of P. Also both values seemed decreasing w.r.t. increasing size of P. Therefore, in all following experiments we focused on fixed small sizes of P and the results can be found in Tables 3.1 and 3.2.

We also ran several experiments on synthetic datasets. Our synthetic contexts were randomly generated with fixed density and contained 500 objects and 100 attributes. For each density we generated 1000 formal contexts. For each such

$\lvert P\rvert$	Mushrooms		Nursery	
	Max n	Avg n	Max n	Avg n
3	9	3.39	6	2.27
4	18	4.77	8	2.59
5	17	6.34	9	3.36
6	18	7.51	14	4.37
7	14	7.27	12	4.78
8	19	7.77	16	5.99
10	12	7.63	18	7.45
15	12	6.84	10	5.94
20	10	6.32	8	5.18
25	11	6.2	7	4.56

Table 3.1: The results of experiments on larger real world datasets. The size of P is fixed, i.e. it is no longer given as a percentage. The whole lattice was generated only once, specifically for Nursery dataset and $|P| = 25$.

$\lvert P\rvert$	Post		Zoo		Drinks	
	Max n	Avg n	Max n	Avg n	Max n	Avg n
3	8	2.81	8	2.92	8	3.09
4	16	3.77	11	3.68	12	3.6
5	24	5.40	13	4.20	9	3.90
6	28	7.35	12	4.46	9	4.15
7	25	8.76	12	4.68	10	4.15
8	24	9.31	13	4.94	9	4.13
10	28	9.51	11	5.06	9	4.11
15	18	7.78	11	4.85	10	4.06
20	15	6.82	9	4.72	8	3.75
25	13	6.42	11	4.43	8	3.53

Table 3.2: The results of experiments on smaller real world datasets. The size of P is fixed, i.e. it is no longer given as a percentage. In none of the cases P generated the whole concept lattice.

context and each fixed size of P we randomly selected generators 100 times and recorded the maximal and average number of iterations of our algorithm across all generated contexts. The results can be found in Table 3.3 and 3.4.

Density	5%			10%			20%		
$\|P\|$	Max n	Avg n	All	Max n	Avg n	All	Max n	Avg n	All
3	3	2.06	0	3	2.07	0	3	2.12	0
4	6	2.23	0	8	2.29	0	8	2.47	0
5	8	2.52	0	20	2.71	0.09	28	3.76	2.92
6	20	2.94	0.08	25	3.65	2.34	27	6.34	23.7
7	22	3.64	0.95	26	5.70	17.38	28	8.44	67.73
8	26	5.15	8.37	22	8.28	57.46	21	7.53	92.55
10	26	9.38	61.4	20	8.07	97.72	10	5.97	99.76
15	16	7.95	99.98	8	6.00	100	6	4.40	100
20	8	6.20	100	6	5.45	100	5	4.01	100
25	8	6.00	100	6	4.29	100	4	4.00	100

Table 3.3: The results of experiments on synthetic contexts with densities 5%, 10% and 20%. The column “All” specifies percentage of cases in which P generated the whole concept lattice.

Investigating the results on synthetic data, we can observe the peak in both maximal and average number of iterations shifting from $|P| \approx 10$ to $|P| \approx 6$ with increasing density of contexts. Interestingly, the peak in the average number of iterations seems slightly decreasing w.r.t. density, being 9.38 for $|P| = 10$ and 7.76 for $|P| = 6$.

From the results we can see that the average number of iterations is very small compared to the number of objects and attributes. The maximal number of iterations is also small compared to the size of corresponding context. Both values usually peak for very small sizes of P and there is also an apparent decreasing trend for number of iterations for increasing size of P. We observed the highest number of iterations for fixed sizes of P, no matter the size of the concept lattice.

Density	30%			40%			50%		
$\|P\|$	Max n	Avg n	All	Max n	Avg n	All	Max n	Avg n	All
3	3	2.16	0	3	2.19	0	18	2.23	0
4	29	2.75	0	37	3.17	0	30	3.62	0
5	28	5.26	7.6	37	6.11	6.7	35	6.78	3.75
6	28	7.78	31.39	28	7.91	25.61	40	7.76	15.12
7	26	7.78	64.69	27	7.24	50.97	22	6.86	34.95
8	20	6.54	85.13	22	6.14	73.64	20	5.98	56.49
10	9	5.35	98.81	12	4.97	95.32	10	4.80	88.05
15	6	4.03	100	5	4	99.99	8	4	99.97
20	4	4	100	4	3.99	100	4	3.98	100
25	4	3.99	100	4	3.93	100	4	3.70	100

Table 3.4: The results of experiments on synthetic contexts with densities 30%, 40% and 50%. The column "All" specifies percentage of cases in which P generated the whole concept lattice.

3.3 Closed subrelations containing arbitrary relation

Procedure GENERATE from Algorithm 3 accepts any subrelation of I as its parameter so it does make sense to investigate what output it generates upon receiving an arbitrary subrelation of I. Interestingly, the next lemma shows that no matter what subrelation the procedure is called with, it always outputs a closed subrelation of I. This may come as a surprise due to the fact that closed subrelations do not form a closure system.

Lemma 16. *For any subrelation $K \subseteq I$, the subrelation* GENERATE(K) *is a closed subrelation of I.*

The above result alludes to some questions about closed subrelations containing a given subrelation $K \subseteq I$. Namely, 1. whether there is the least such subrelation, and 2. what can be said about these subrelations and their concept lattices in general.

As mentioned in [5], the intersection of a system of closed subrelations needs not be a closed subrelation. Therefore, the system of all closed subrelations of I is not a closure system and, consequently, there does not exist a closure operator assigning to each subrelation $K \subseteq I$ the least greater closed subrelation. Thus, the answer to the first question is, at least in general, negative. However, in what follows, we show that for some important type of subrelations of I (which we have already met) the answer is positive.

Definition 17 (semi-closed subrelation). We call a subrelation $L \subseteq I$ *semi-closed* if for each $x \in X$ and $y \in Y$, $\{x\}^{\uparrow_L}$ is an intent and $\{y\}^{\downarrow_L}$ is an extent of I.

The definition of semi-closed subrelation resembles closely the definition of a block relation [5, 19]. The only difference is that semi-closed relations are subrelations whereas block relations are super-relations of I.

We denote by $\mathrm{C}_{\mathcal{IV}}$ the closure operator associated with the closure system of all semi-closed subrelations of I. Algorithm 5 computes the value of $\mathrm{C}_{\mathcal{IV}}$ for any subrelation $L \subseteq I$. The same algorithm for block relations has been shown in [20].

Algorithm 5 Computing $\mathrm{C}_{\mathcal{IV}}$

Require: subrelation $L \subseteq I$
Ensure: $\mathrm{C}_{\mathcal{IV}}L$

```
i ← 1
repeat
    L' ← L
    i ← i + 1
    if i is even then
        L ← {⟨x, y⟩ ∈ X × Y | x ∈ {y}^{↓L' ↑I ↓I}}
    else
        L ← {⟨x, y⟩ ∈ X × Y | y ∈ {x}^{↑L' ↓I ↑I}}
until i > 2 & L = L'
return L
```

Lemma 18. *If J is a closed subrelation of I and $K \subseteq J$, then $\mathrm{C}_{\mathcal{IV}}K \subseteq J$.*

The following lemma shows that despite the fact that closed subrelations do not form a closure system, in some important cases, there exists the least closed subrelation containing K.

Lemma 19. *Let $K = \bigsqcup P$ for some $P \subseteq \mathcal{B}(X, Y, I)$. Then $\mathrm{C}_{\mathcal{IV}}K$ is a closed subrelation of I and the concept lattice of $\mathrm{C}_{\mathcal{IV}}K$ is equal to $\mathrm{C}_{\bigvee\bigwedge}P$. $\mathrm{C}_{\mathcal{IV}}K$ can be computed by procedure* Generate *of Alg. 3.*

For each concept $\langle A, B\rangle$ of a semi-closed subrelation of I, A is an extent and B an intent of I (there need not be any special relationship between A and B, i.e., in general, $B \neq A^{\uparrow_I}$ and $A \neq B^{\downarrow_I}$). We call such preconcepts *interval-preconcepts* and denote the set of all interval-preconcepts of I by $\mathcal{IV}(X, Y, I)$.

Interval-preconcepts are ordered the same way as preconcepts [8, 9], i.e. for each $\langle A, B\rangle, \langle C, D\rangle \in \mathcal{IV}(X, Y, I)$ we have $\langle A, B\rangle \leq \langle C, D\rangle$ iff $A \subseteq C$ and $D \subseteq B$.

Lemma 20. *$\mathcal{IV}(X, Y, I)$ together with the above ordering is a complete lattice with infima and suprema given by the same formulas as in* (1.1) *and* (1.2)*.*

Lemma 21. *1. For any semi-closed subrelation $L \subseteq I$, $\mathcal{B}(X, Y, L)$ is a complete sublattice of $\mathcal{IV}(X, Y, I)$.*

2. A complete sublattice $U \subseteq \mathcal{IV}(X, Y, I)$ is equal to $\mathcal{B}(X, Y, L)$ for some semi-closed subrelation $L \subseteq I$ iff for each $\langle A, B\rangle, \langle C, D\rangle \in U$,

$$A \subseteq C \quad \textit{iff} \quad D \subseteq B. \tag{3.5}$$

It holds $L = \bigsqcup U$.

Notice that each interval-preconcept $\langle A, B\rangle$ of I determines a closed interval in the concept lattice $\mathcal{B}(X, Y, I)$, namely the interval $[\langle A, A^{\uparrow_I}\rangle, \langle B^{\downarrow_I}, B\rangle]$. This correspondence between interval-preconcepts of I and closed intervals in $\mathcal{B}(X, Y, I)$ is evidently bijective. For each semi-closed subrelation $L \subseteq I$ we denote by S_L the system of all closed intervals of $\mathcal{B}(X,Y,I)$ determined by concepts of L (which are interval-preconcepts of I). This system corresponds to the concept lattice $\mathcal{B}(X,Y,L)$. Thus, the above Lemma 21 can be used for further investigation of the structure of S_L.

Lemma 22. *Let $K \subseteq I$ be an arbitrary subrelation, $U \subseteq \mathcal{B}(X,Y,I)$ a complete sublattice with the associated closed subrelation $J \subseteq I$. Then $J \supseteq K$ iff U has nonempty intersection with each interval from the system* $\mathrm{S}_{\mathrm{C}_{\mathcal{I}\mathcal{V}}K}$.

We already know that procedure Generate always outputs a closed subrelation given any subrelation $K \subseteq I$. However, dual computation, i.e. interchanging parts for even and odd indices, might lead to a different result. Notice that by executing the first iteration as it leaves us with either correct intents or extents. Due to Lemma 22 we know how concepts of K determine closed intervals of $\mathcal{B}(I)$. Now, the result depends on which concepts we choose from corresponding intervals. Taking procedure Generate as is, the first iteration selects the upper bounds of all such intervals. Dual computation selects the lower bounds. Note that this selection can have an effect on the size (w.r.t. $\subseteq$) of the result.

3.4 Interval-preconcepts

Now, we take a closer look on interval-preconcepts. We already showed that they can be ordered in the same way as preconcepts and together with this ordering the set of all interval-preconcepts of a given formal context forms a complete lattice.

Interval-preconcepts have two defining properties: 1. they are preconcepts, 2. they uniquely determine an interval in the corresponding concept lattice. The second property plays a crucial role in our motivation to study such formal rectangles. First, we focus on a motivation from psychology and show where interval-preconcepts can fit in the theory of concepts. Second, we provide a more formal motivation relating interval-preconcepts to block relations.

There has been quite a lot of work done in the field of cognitive psychology concerning concepts especially in last 70 years (see [21] for a quick and [22] for a comprehensive overview). Nowadays, there are several definitions of concept in cognitive psychology that are sometimes almost contradictory and some approaches actually reject any definition of concept as it is impossible to define (see also [23]). They argue that for any such definition there exist an example (usually natural and very simple) that does not conform to it [24]. Nevertheless, we cannot possibly hope to mathematically define a notion that we are unable to grasp even with the help of vagueness of our language. Therefore, the notion of formal concept is based on the definition of concept from *The Classical Theory of Concepts* which dates back to antiquity to Plato [24]. It accepts a definition of concept as a structured mental representation that encodes necessary and sufficient conditions for its application. By application, we usually mean the ability to decide if an object is part of a concept or not, i.e. the process of categorization in the terms of cognitive psychology.

Accepting our notion of formal concept as a mathematical representation of the classical definition of concept leaves us with several possible interpretations of the notion interval-preconcept. In the study of concept acquisition, we can look at them as possible stages of the learning process. In formal terms, we represent a whole interval of concepts as a single entity and we are yet to acquire knowledge that would let us draw distinctions between concepts in this interval.

Other possible application of interval-preconcepts stems from the necessity of ignorance. It has been argued (see [22]) that people make adjustment to their process of categorization depending on current circumstances. One such adjustment includes ignorance of some features. This prevents overwhelming our mind and speeds up the process of categorization. By restraining the set of features only to a valid intension of a more general concept we obtain a way of thinking about more specific concepts in more general, but coherent, terms.

The second part of our motivation is purely formal. We have a new type of formal rectangle that arises from a notion of semi-closed subrelation which is closely related to the notion of block relation. We explore their connection and investigate potential areas where interval-preconcepts can help to represent more complex notions whilst providing a different viewing angle.

Definition 23 (interval-preconcept). An interval-preconcept of a formal context $\langle X, Y, I\rangle$ is a preconcept $\langle A, B\rangle$ such that $A = A^{\uparrow_I \downarrow_I}$ and $B = B^{\downarrow_I \uparrow_I}$.

To better understand relations of interval-preconcepts to other well-known types of rectangles we provide a summary in Fig. 3.7 and 3.8. Moreover, in Figure 3.9 we can find proper instances of all types of rectangles.

	$A \subseteq B^{\downarrow}$, $B \subseteq A^{\uparrow}$	$A^{\uparrow\downarrow} = B^{\downarrow}$	$B = A^{\uparrow}$	$A = B^{\downarrow}$	$A = A^{\uparrow\downarrow}$, $B = B^{\downarrow\uparrow}$
preconcept	$\times$				
protoconcept	$\times$	$\times$			
interval-preconcept	$\times$				$\times$
$\sqcap$-semiconcept	$\times$	$\times$	$\times$		
$\sqcup$-semiconcept	$\times$	$\times$		$\times$	
formal concept	$\times$	$\times$	$\times$	$\times$	$\times$

Figure 3.7: A formal context of different types of formal rectangles.

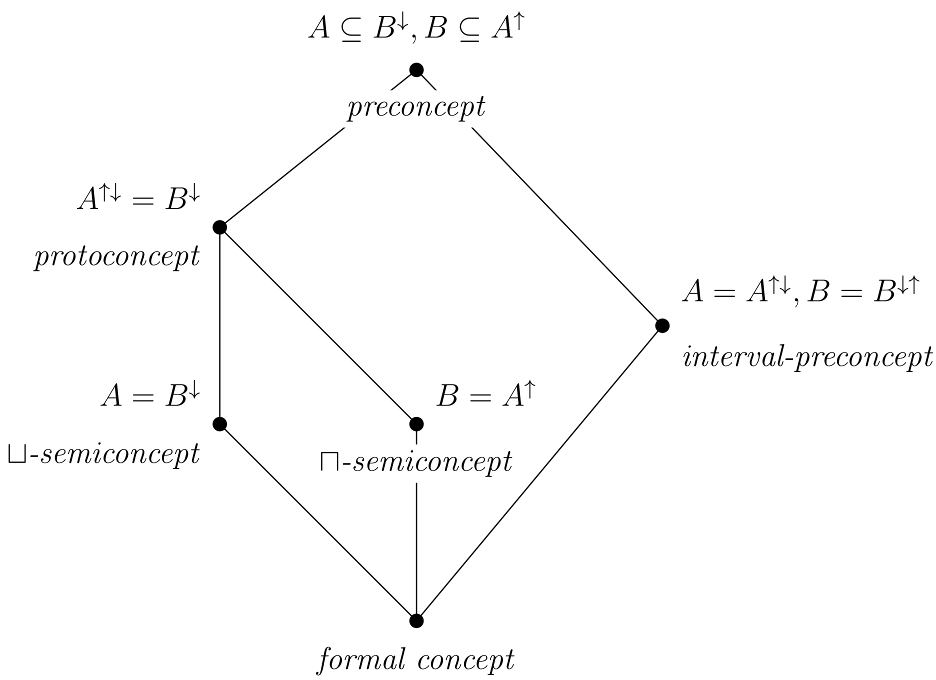


Figure 3.8: The concept lattice corresponding to the formal context from the Fig. 3.7 showing relations between different types of formal rectangles.

I	y_1	y_2	y_3
x_1		$\times$	
x_2	$\times$	$\times$	$\times$
x_3		$\times$	$\times$

Figure 3.9: Different types of formal rectangles can be found in this context. We can identify following examples: a proper preconcept $\langle\{x_2\},\{y_3\}\rangle$, a proper interval-preconcept $\langle\{x_2\},\{y_2\}\rangle$, a proper protoconcept $\langle\{x_3\},\{y_3\}\rangle$ and proper semiconcepts $\langle\{x_3\},\{y_2,y_3\}\rangle$ and $\langle\{x_2,x_3\},\{y_3\}\rangle$.

The notion of interval-preconcept originates from our investigation of semi-closed subrelations. Such relations are defined similarly to block relations, however they are subrelations as opposed to being super-relations. Thus, we investigate the relation between interval-preconcepts and block relations. Recall, each interval-preconcept $\langle A,B\rangle \in \mathcal{IV}(X,Y,I)$ uniquely determines an interval in the concept lattice $\mathcal{B}(I)$, namely, $[\langle A,A^{\uparrow_I}\rangle,\langle B^{\downarrow_I},B\rangle]$. We now show how a block relation corresponds to a set of interval-preconcepts.

Lemma 24. *Let $L \supseteq I$ be a block relation, then for each $\langle A,B\rangle \in \mathcal{B}(L)$, $c = \langle B^{\downarrow_I},A^{\uparrow_I}\rangle$ is an interval-preconcept of I. Denote R_L the set of all such interval-preconcepts. R_L is an order-embedded complete lattice in $\mathcal{IV}(X,Y,I)$ with infima and suprema given by:*

$$\bigwedge_{\iota\in\mathcal{I}}\langle A_\iota,B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota^{\uparrow_I}\right)^{\downarrow_K\uparrow_K\downarrow_I}, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle, \tag{3.6}$$

$$\bigvee_{\iota\in\mathcal{I}}\langle A_\iota,B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota^{\downarrow_I}\right)^{\uparrow_K\downarrow_K\uparrow_I} \right\rangle. \tag{3.7}$$

Evidently, the converse direction does not generally hold, i.e. not all subsets of $\mathcal{IV}(X,Y,I)$ correspond to block relations.

Lemma 25. *Let $V \subseteq \mathcal{IV}(X,Y,I)$ we have $V = \mathrm{R}_L$ for some block relation $L \supseteq I$ iff V is an order-embedded complete lattice in $\mathcal{IV}(X,Y,I)$ such that for each*

$\langle A, B\rangle, \langle C, D\rangle \in V$

$$A \subseteq C \quad \textit{iff} \quad D \subseteq B. \tag{3.8}$$

and $L = \bigcup_{\langle A,B\rangle \in V}(B^{\downarrow_I} \times A^{\uparrow_I})$.

We already showed that the set of all interval-preconcepts $\mathcal{IV}(X, Y, I)$ of given context is a complete lattice. Now, we show how to create a formal context K such that its concept lattice is isomorphic to $\mathcal{IV}(X, Y, I)$. For this purpose we adopt the following notation. For any set A we put $\overline{A} = \{\overline{x} \mid x \in A\}$.

Lemma 26. *For a formal context* $\langle X, Y, I\rangle$ *we have* $\mathcal{IV}(X, Y, I) \cong \mathcal{B}(X \cup \overline{X}, Y \cup \overline{Y}, K_I)$ *where* $K_I = \{\langle x, y\rangle, \langle \overline{x}, y\rangle, \langle x, \overline{y}\rangle \mid x \in X, y \in Y$ *and* $\langle x, y\rangle \in I\} \cup \overline{X} \times \overline{Y}$.

Interestingly, the previous construction turns out to be equivalent to the direct product of the formal context $\langle X, Y, I\rangle$ and formal context from Fig. 3.10.

Definition 27. The *direct product* of formal contexts $\langle X_1, Y_1, I_1\rangle$ and $\langle X_2, Y_2, I_2\rangle$ is given by

$$\langle X_1, Y_1, I_1\rangle \times \langle X_2, Y_2, I_2\rangle = (X_1 \times X_2, Y_1 \times Y_2, K)$$
$$\text{where } (x_1, x_2)K(y_1, y_2) \Leftrightarrow x_1 I_1 y_1 \text{ or } x_2 I_2 y_2.$$

The concept lattice of the direct product of some formal contexts is called a *tensor product of concept lattices.*

	y_1	y_2
x_1		
x_2		$\times$

Figure 3.10: A formal context with the concept lattice isomorphic to a three element chain.

Corollary 28. *For a formal context* $\langle X, Y, I\rangle$ *we have*

$$\mathcal{IV}(X, Y, I) \cong \mathcal{B}(\langle X, Y, I\rangle \times \langle \{x_1, x_2\}, \{y_1, y_2\}, \{\langle x_2, y_2\rangle\}\rangle).$$

The previous means that $\mathcal{IV}(X,Y,I)$ is isomorphic to a tensor product of certain concept lattices. Therefore, we can use any result about tensor product of concept lattices to investigate properties of $\mathcal{IV}(X,Y,I)$. Now, we provide a basic theorem on interval-preconcept lattices.

Theorem 29 (Basic Theorem on Interval-preconcept Lattices). *The interval-preconcept lattice $\mathcal{IV}(X,Y,I)$ is a complete lattice in which infima and suprema are given by:*

$$\bigwedge_{\iota\in\mathcal{I}}\langle A_\iota, B_\iota\rangle = \left\langle \bigcap_{\iota\in\mathcal{I}} A_\iota, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I}\right\rangle, \tag{3.9}$$

$$\bigvee_{\iota\in\mathcal{I}}\langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \bigcap_{\iota\in\mathcal{I}} B_\iota\right\rangle. \tag{3.10}$$

In general, a complete lattice V with an element p admits an isomorphism α with interval-preconcept lattice $\mathcal{IV}(X,Y,I)$ with $\alpha(p) = \langle\emptyset^{\uparrow\downarrow},\emptyset^{\downarrow\uparrow}\rangle$ if and only if there exist mappings $\gamma : X \to V$ and $\mu : Y \to V$ where

$$D_\vee = \bigcup_{x\in X}\{\gamma(x), \gamma(x)\wedge p\} \text{ is supremally dense in } V, \tag{3.11}$$

$$D_\wedge = \bigcup_{y\in Y}\{\mu(y), \mu(y)\vee p\} \text{ is infimally dense in } V, \tag{3.12}$$

such that for any $x \in X$, $y \in Y$ it holds

$$xIy \Leftrightarrow \gamma(x) \le \mu(y) \Leftrightarrow \gamma(x)\wedge p \le \mu(y)\wedge p \Leftrightarrow \gamma(x)\vee p \le \mu(y)\vee p. \tag{3.13}$$

We can immediately make several observations from the basic theorem. First, there are three isomorphisms between $\mathcal{B}(I)$ and certain parts of $\mathcal{IV}(X,Y,I)$, namely, intervals $[\langle\emptyset^{\uparrow_I\downarrow_I},\emptyset^{\downarrow_I}\rangle, p]$, $[p, \langle\emptyset^{\downarrow_I},\emptyset^{\downarrow_I\uparrow_I}\rangle]$ and the complete sublattice $\mathcal{IV}(X,Y,I)\cap\mathcal{B}(I)$. Second, we only need as many labels for $\mathcal{IV}(X,Y,I)$ as for $\mathcal{B}(I)$. Third, each element of $\mathcal{IV}(X,Y,I)$ is associated with two extents and two intents of I and it is easy to read them from the lattice. Note that Theorem 29 does not specify the structure of interval-preconcept lattices and this flaw is rectified in Theorem 32.

We conclude this section with some final observations about the structure of interval-preconcept lattices.

Theorem 30. *We have* $\mathcal{IV}(I) \cong \langle\{\langle c_1, c_2\rangle \in \mathcal{B}(I) \times \mathcal{B}(I) \mid c_1 \leq c_2\}, \sqsubseteq\rangle$, *where* $\langle c_1, c_2\rangle \sqsubseteq \langle c_3, c_4\rangle$ *iff* $c_1 \leq c_3$ *and* $c_2 \leq c_4$.

Corollary 31. *A complete lattice* V *isomorphic to the upper half of a Cartesian product* $W \times W$ *(i.e. the subset of* $W \times W$ *containing all pairs* $\langle u, v\rangle$ *where* $u, v \in W$ *and* $u \leq v$*) of some complete lattice* W *is isomorphic to* $\mathcal{IV}(W, W, \leq)$.

We call the complete lattice V from the previous corollary an *upper triangular complete lattice (of W)*.

Theorem 32 (Second Basic Theorem on Interval-preconcept Lattices). *The interval-preconcept lattice* $\mathcal{IV}(X, Y, I)$ *is an upper triangular complete lattice of* $\mathcal{B}(X, Y, I)$ *in which infima and suprema are given by:*

$$\bigwedge_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \bigcap_{\iota\in\mathcal{I}} A_\iota, \left(\bigcup_{\iota\in\mathcal{I}} B_\iota\right)^{\downarrow_I\uparrow_I} \right\rangle, \tag{3.14}$$

$$\bigvee_{\iota\in\mathcal{I}} \langle A_\iota, B_\iota\rangle = \left\langle \left(\bigcup_{\iota\in\mathcal{I}} A_\iota\right)^{\uparrow_I\downarrow_I}, \bigcap_{\iota\in\mathcal{I}} B_\iota \right\rangle. \tag{3.15}$$

In general, an upper triangular lattice V *of* W *admits an isomorphism* α *with interval-preconcept lattice* $\mathcal{IV}(X, Y, I)$ *if and only if there exist mappings* $\gamma : X \to V$ *and* $\mu : Y \to V$ *such that*

$$D_\vee = \bigcup_{x\in X} \{\gamma(x), \gamma(x) \wedge p\} \text{ is supremally dense in } V, \tag{3.16}$$

$$D_\wedge = \bigcup_{y\in Y} \{\mu(y), \mu(y) \vee p\} \text{ is infimally dense in } V, \tag{3.17}$$

where $p = \langle \bigvee \emptyset, \bigwedge \emptyset\rangle$ *and for any* $x \in X$, $y \in Y$ *it holds*

$$xIy \Leftrightarrow \gamma(x) \leq \mu(y) \Leftrightarrow \gamma(x) \wedge p \leq \mu(y) \wedge p \Leftrightarrow \gamma(x) \vee p \leq \mu(y) \vee p, \tag{3.18}$$

in particular, $V \cong \mathcal{IV}(W, W, \leq)$.

3.5 Discussion and related work

An obvious advantage of our result on generating complete sublattices is that we avoid computation of any lattices and instead we work exclusively with contexts. In fact, our goal is to compute the closed subrelation corresponding to the given generated complete sublattice. The actual computation of the sublattice, if necessary, can be done with any well-known efficient algorithm for concept lattice construction. This should lead to shorter computation time, especially if the generated sublattice V is substantially smaller than $\mathcal{B}(X, Y, I)$.

In Lemma 14, we give an upper estimation of the number of iterations of our algorithms. It seems that this estimation could be improved. At the time of writing, we were not able to construct any example with the number of iterations greater than $O(|X| + |Y|)$.

As far as related work goes, we are aware of only one published algorithm for generating sublattices [25]. Unfortunately, we cannot do any comparison as the algorithm in question is not correct. It does not always output a sublattice.

We also looked into the problem of characterizing all closed subrelations containing an arbitrary subrelation. We introduced a notion of semi-closed subrelation which is similar to that of block relation. In contrast with closed subrelations, semi-closed subrelations form a closure system. We showed how this notion can be used to solve the problem at hand. We also used it to identify an important type of subrelations for which we can always find a unique smallest closed subrelation containing it.

Investigation of semi-closed subrelations leads to a definition of new type of formal rectangles which we call interval-preconcepts. They uniquely determine an interval in the original concept lattice and they have a close relation to block relations and so to lattice factorization. We showed that together with the same ordering as preconcepts, they form a complete lattice. We studied properties of interval-preconcepts and we presented two versions of basic theorem on interval-preconcept lattices.

Chapter 4

Conclusion

We analyzed the basic step in incremental lattice construction, removal of an incidence, and based on this analysis we proposed two incremental algorithms for updating concepts and the corresponding concept lattice. As this is the smallest possible change in a formal context, we believe that this problem is in some form present in every incremental lattice construction method.

The performance of our algorithms depends heavily on the size of the order-embedded complete lattice (resp. interval) that contains exactly the concepts that are affected by the removal. Our experiments showed that the size of this interval is usually very small compared to the whole lattice. Combining it with some optimizations, the algorithm for updating concepts in fact computes two derivation operators and two set equality tests for each concept from the identified interval. Further extending presented method, we were able to remove an arbitrary preconcept at once without any additional overhead. By investigating possible extensions of our results, we arrived at a general method for updating a concept lattice upon an arbitrary change in the underlying context.

Afterwards, we focused on studying substructures, specifically complete sublattices generated by a set of elements. As it turns out, there is an efficient way of computing the closed subrelation corresponding to a complete sublattice generated by a set of elements. Computing such closed subrelation provides a full description of the corresponding generated complete sublattice and the actual construction of it, if necessary, can be done via any well-known efficient algorithm. Experiments with our method showed its efficiency and provided some insight into parameters that have an impact on its performance. Interestingly, the peak in the complexity of our method was achieved with small fixed sizes of the set of generators, i.e. the number of generators did not depend on the size of the lattices in these experiments.

The algorithm we proposed actually computes a closed subrelation for any given subrelation and in some sense the result seems minimal. This motivated us to further investigate this since it is in contrast with well-known result postulating that closed subrelations do not form a closure system. We introduced the notion of semi-closed subrelations that are more general than closed subrelations and indeed form a closure system. Using this notion we were able to identify an important type of subrelations for which there always exists the smallest closed subrelation containing given subrelation.

Our investigation of concepts of semi-closed relation lead us to a definition of a new type of formal rectangle that we call interval-preconcept. As most well-known types of formal rectangles have motivation in cognitive psychology, so does interval-preconcept and we showed some scenarios where it can serve as a formalization of some notion from cognitive psychology. We also explored their relations to other well-known types of formal rectangles and to block relations that are used for lattice factorization. Lastly, we showed how they can be structured into a complete lattice and proposed two versions of basic theorem on interval-preconcept lattices.

Bibliography

[1] GANTER, Bernard. Order-embedded Complete Lattices. In: IGNATOV, Dmitry I.; NOURINE, Lhouari (eds.). *CLA*. CEUR-WS.org, 2018, vol. 2123, pp. 153–165. CEUR Workshop Proceedings. Available also from: `http://ceur-ws.org/Vol-2123/paper13.pdf`.

[2] BIRKHOFF, G. *Lattice Theory*. American Mathematical Society, 1973. American Mathematical Society colloquium publications. Available also from: `https://books.google.de/books?id=YEj4nQEACAAJ`.

[3] WILLE, Rudolf. Restructuring lattice theory: an approach based on hierarchies of concepts. In: RIVAL, I. (ed.). *Ordered Sets*. Boston, 1982, pp. 445–470.

[4] SINGH, Prem Kumar; KUMAR, Cherukuri Aswani; GANI, Abdullah. A comprehensive survey on formal concept analysis, its research trends and applications. *International Journal of Applied Mathematics and Computer Science*. 2016, vol. 26, no. 2, pp. 495–516. Available also from: `https://content.sciendo.com/view/journals/amcs/26/2/article-p495.xml`.

[5] GANTER, Bernhard; WILLE, Rudolf. *Formal Concept Analysis – Mathematical Foundations*. Springer, 1999.

[6] LUKSCH, Peter; WILLE, Rudolf. A Mathematical Model for Conceptual Knowledge Systems. In: BOCK, Hans-Hermann; IHM, Peter (eds.). *Classification, Data Analysis, and Knowledge Organization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 156–162. ISBN 978-3-642-76307-6.

[7] WILLE, Rudolf. Boolean Concept Logic. In: GANTER, Bernhard; MINEAU, Guy W. (eds.). *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 317–331. ISBN 978-3-540-44663-7.

[8] STAHL, Jürgen; WILLE, Rudolf. Preconcepts and set representation of contexts. In: GAUL, W.; SCHADER, M. (eds.). *Classification as a tool of research*. Amsterdam: North–Holland, 1986, pp. 431–438.

[9] BURGMANN, Christian; WILLE, Rudolf. The Basic Theorem on Preconcept Lattices. In: *Proceedings of the 4th International Conference on Formal Concept Analysis*. Dresden, Germany: Springer-Verlag, 2006, pp. 80–88. ICFCA'06. ISBN 3-540-32203-5, 978-3-540-32203-0. Available from DOI: `10.1007/11671404_5` (`http://dx.doi.org/10.1007/11671404_5`).

[10] PIAGET, Jean. *La formation du symbole chez l'enfant - imitation, jeu et reve - image et representation*. Delachaux et Niestle, 1959.

[11] CARPINETO, Claudio; ROMANO, Giovanni. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004. ISBN 0470850558.

[12] MERWE, Dean; OBIEDKOV, Sergei; KOURIE, Derrick. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: EKLUND, Peter (ed.). *Concept Lattices*. Springer Berlin Heidelberg, 2004, vol. 2961, pp. 372–385. Lecture Notes in Computer Science. ISBN 978-3-540-21043-6.

[13] OUTRATA, Jan. A Lattice-Free Concept Lattice Update Algorithm based on *CbO. In: OJEDA-ACIEGO, Manuel; OUTRATA, Jan (eds.). *CLA*. CEUR-WS.org, 2013, vol. 1062, pp. 261–274. CEUR Workshop Proceedings.

[14] NORRIS, E. M. An Algorithm for Computing the Maximal Rectangles in a Binary Relation. *Revue Roumaine de Mathématiques Pures et Appliquées*. 1978, vol. 23, no. 2, pp. 243–250.

[15] NOURINE, Lhouari; RAYNAUD, Olivier. A Fast Algorithm for Building Lattices. *Inf. Process. Lett.* 1999, vol. 71, no. 5-6, pp. 199–204. ISSN 0020-0190. Available from DOI: `10.1016/S0020-0190(99)00108-8` (`http://dx.doi.org/10.1016/S0020-0190(99)00108-8`).

[16] DOWLING, Cornelia E. On the Irredundant Generation of Knowledge Spaces. *J. Math. Psychol.* 1993, vol. 37, no. 1, pp. 49–62. ISSN 0022-2496. Available from DOI: `10.1006/jmps.1993.1003` (`http://dx.doi.org/10.1006/jmps.1993.1003`).

[17] KUZNETSOV, Sergei O.; OBIEDKOV, Sergei. Comparing Performance of Algorithms for Generating Concept Lattices. *Journal of Experimental and Theoretical Artificial Intelligence*. 2002, vol. 14, pp. 189–216.

[18] BELOHLAVEK, Radim; TRNECKA, Martin. Basic Level of Concepts in Formal Concept Analysis. In: DOMENACH, Florent; IGNATOV, Dmitry I.; POELMANS, Jonas (eds.). *Formal Concept Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 28–44. ISBN 978-3-642-29892-9.

[19] WILLE, Rudolf. Complete tolerance relations of concept lattices. In: EIGENTHALER, G.; AL., et (eds.). *Contributions to General Algebra*. Wien: Hölder-Pichler-Tempsky, 1985, vol. 3, pp. 397–415.

[20] BARTL, Eduard; KRUPKA, Michal. Residuated lattices of block relations: size reduction of concept lattices. *International Journal of General Systems*. 2016, vol. 45, no. 7-8, pp. 773–789. Available from DOI: `10.1080/03081079.2016.1144601` (`http://dx.doi.org/10.1080/03081079.2016.1144601`).

[21] MACHERY, Edouard. 100 Years of Psychology of Concepts: The Theoretical Notion of Concept and its Operationalization. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*. 2007, vol. 38, no. 1, pp. 63–84.

[22] MARGOLIS, E.; LAURENCE, S. *Concepts: Core Readings*. MIT Press, 1999. Bradford books. ISBN 9780262631938. Available also from: `https://books.google.de/books?id=sj1gczQ-7K8C`.

[23] FODOR, J.A.; GARRETT, M.F.; WALKER, E.C.T.; PARKES, C.H. Against definitions. *Cognition*. 1980, vol. 8, no. 3, pp. 263 –367. ISSN 0010-0277. Available also from: `http://www.sciencedirect.com/science/article/pii/0010027780900086`.

[24] PLATO. Five Dialogues: Euthyphro, Apology, Crito, Meno, Phaedo. 1981.

[25] BERTET, K.; MORVAN, M. Computing the sublattice of a lattice generated by a set of elements. In: *Proceedings of Third International Conference on Orders, Algorithms and Applications*. Montpellier, France, 1999.