

# NextClosures with Constraints

Francesco Kriegel

Institute of Theoretical Computer Science  
Technische Universität Dresden, Dresden, Germany  
`francesco.kriegel@tu-dresden.de`

**Abstract.** In a former paper, the algorithm *NextClosures* for computing the set of all formal concepts as well as the canonical base for a given formal context has been introduced. Here, this algorithm shall be generalized to a setting where the data-set is described by means of a closure operator in a complete lattice, and furthermore it shall be extended with the possibility to handle constraints that are given in form of a second closure operator. As a special case, constraints may be predefined as implicational background knowledge. Additionally, we show how the algorithm can be modified in order to do parallel *Attribute Exploration* for unconstrained closure operators, as well as give a reason for the impossibility of (parallel) *Attribute Exploration* for constrained closure operators if the constraint is not compatible with the data-set.

**Keywords:** Formal Concept Analysis, Implication, Canonical Base, Constraint, Closure Operator, Parallel Computation, Parallel Exploration

## 1 Introduction

Recently, in [18, 20] a parallel algorithm for the computation of canonical bases (and the set of formal concepts as a byproduct) for formal contexts has been introduced. Furthermore, in [18, 19] some extensions have been provided that allow for the parallel exploration of canonical bases by means of experts. However, not all data-sets occurring in practical applications can efficiently be described as a formal context. Henceforth, it can be useful to describe a generalization of the *NextClosures* algorithm which allows to compute canonical bases for implications valid in a closure operator in a complete lattice. Of course, powersets are always complete lattices, and the composition of the two derivation operators of a formal context is a closure operator in the powerset lattice – hence this is indeed a generalization. Since the algorithm proceeds in a level-wise order that is compatible with the set-theoretic subset ordering, the generalization presupposes a strict order-homomorphism of the underlying lattice into the ordered set  $(\mathbb{N}, \leq)$  of the natural numbers with their usual ordering (called a *quasi-rank function*). Furthermore, this paper presents some possible applications, e.g., in *Formal Concept Analysis* [13], for interpretations in *Description Logics* [2], and for *Pattern Structures* [12].

First, the notion of a closure operator in a complete lattice is defined, and it is proven that the set of all closure operators in a fixed complete lattice forms a complete lattice itself. Consequently, it is possible to construct infima and suprema of sets of closure operators. As applications, we discuss what it means for implications to be valid in infima and suprema, and furthermore generalize the notion of *C-implications* (*constrained implications*, where *C* is a closure operator) in [4] from the special case of *Formal Concept Analysis* to the more general case of closure operators.

In [24], Stumme introduced an extension of Ganter's *Attribute Exploration* [10, 11] that can handle background knowledge in form of an implication set. A further generalization [9] allows for arbitrary propositional background knowledge. The input data is given as a formal context  $\mathbb{K} = (G, M, I)$  as well as an implication set  $\mathcal{L} \subseteq \text{Imp}(M)$  that is valid in  $\mathbb{K}$ . Then by generalizing the notions of pseudo-intents and canonical bases, it is possible to compute a minimal extension of  $\mathcal{L}$  which is sound and complete for  $\mathbb{K}$ . This minimal extension is called *canonical base* of  $\mathbb{K}$  *relative* to  $\mathcal{L}$ . However, it has not been addressed what can be done in the case where  $\mathcal{L}$  is not valid in  $\mathbb{K}$ . There are at least two possibilities to handle such cases. One may either compute a base for the implications that are valid in  $\mathbb{K}$  as well as are entailed by  $\mathcal{L}$ , or one can compute a base for the constrained implications of  $(\mathbb{K}, \mathcal{L})$ . We will discuss some details here, and provide solutions by utilizing infima and suprema of closure operators in lattices.

Higuchi has shown in [16] that the set of all closure operators on a set constitutes a complete lattice. More generally, this is also true for the set of all closure operators in a complete lattice. The order of closure operators is defined as follows: A closure operator  $\phi$  is smaller than another closure operator  $\psi$  if each closure of  $\psi$  is also a closure of  $\phi$ , i.e., if  $\phi$  is more *fine-grained* than  $\psi$  which means that each closure of  $\psi$  can be expressed as an infimum of closures of  $\phi$ . Furthermore, it is easy to verify that implications are valid in an infimum of two closure operators if, and only if, they are valid in each of the closure operators. The dual case considers implications valid in suprema, which are exactly those implications valid for all models that are closures of all involved closure operators. In general, one may think of infima describing unions of data-sets, and suprema as describing intersections (or mergings) of data-sets.

The last case investigated in this document considers the constrained implications. As input two closure operators are necessary, one describing the data-set and the other one describing the constraints to be met. Then a base for all those implications may be computed, for which both premise and conclusion is a closure of the constraining closure operator, and all common closures of both closure operators are models. In particular, we will show an application where we have some implications which have been proven to be correct, and a possibly faulty data-set. As a result, we may construct implications that are compatible with the background knowledge, and furthermore are valid w.r.t. all closures of the data-set which are also models of the background implications.

This document is structured as follows. In Section 2, the notion of a closure operator in a complete lattice is defined, and it is shown that the set of all closure operators constitute a complete lattice itself, i.e., we present an order on closure operators as well as give the equations for the corresponding infimum and supremum operations. Then in Section 3, we consider implications in a complete lattice and define the notion of validity w.r.t. closure operators, as well as entailment for implication sets. Next, Section 4 generalizes the parallel *NextClosures* algorithm from [20] to the case of closure operators (with constraints) in complete lattices. Proofs are omitted, as on the one hand they have already been given in [18], and on the other hand could be generalized from those given in [19, 20] without too much effort. Before finishing this document with some conclusions in Section 6, several possible applications are described, in particular, in *Formal Concept Analysis*, in *Description Logics*, and for *Pattern Structures*.

## 2 The Complete Lattice of Closure Operators in a Complete Lattice

Throughout the whole section, we assume that  $\underline{M} = (M, \leq, \bigwedge, \bigvee, \top, \perp)$  is a *complete lattice*, i.e.,  $M$  is a set;  $\leq$  is an order relation on  $M$ , i.e., it is reflexive, antisymmetric, and transitive; for each subset  $X \subseteq M$ ,  $\bigwedge X$  is the *infimum* of  $X$ , i.e.,  $\bigwedge X \leq x$  for all  $x \in X$ , and  $y \leq x$  for all  $x \in X$  implies  $y \leq \bigwedge X$ ; dually,  $\bigvee X$  is the *supremum* of  $X$ ;  $\top$  is the greatest element and  $\perp$  is the smallest element, i.e.,  $\perp \leq x \leq \top$  for all  $x \in M$ . We set  $x \wedge y := \bigwedge \{x, y\}$  as well as  $x \vee y := \bigvee \{x, y\}$ . For  $x \in M$ , its (*prime*) *ideal* is  $\downarrow x := \{y \in M \mid y \leq x\}$ , and its (*prime*) *filter* is  $\uparrow x := \{y \in M \mid x \leq y\}$ . For further information on (complete) lattices, the interested reader is referred to [5, 7, 13, 14].

A *closure operator* in  $\underline{M}$  is a mapping  $\phi: M \rightarrow M$  that is extensive, monotone, and idempotent, cf. [6, 7, 16].  $\text{ClOp}(\underline{M})$  denotes the set of all closure operators in  $\underline{M}$ . An element  $x \in M$  is a *closure* of  $\phi$  if  $x = x^\phi$ , and we shall denote the set of all closures of  $\phi$  by  $\text{Clo}(\phi)$ . Furthermore, it is well-known that the following statements are equivalent characterizations of a closure operator  $\phi$ :

1.  $x \leq x^\phi$ ,  $x \leq y \Rightarrow x^\phi \leq y^\phi$ , and  $x^{\phi\phi} = x^\phi$ , for all  $x, y \in M$ .
2.  $x \leq y^\phi \Leftrightarrow x^\phi \leq y^\phi$  for all  $x, y \in M$ .
3.  $x \vee y^{\phi\phi} \leq (x \vee y)^\phi$  for all  $x, y \in M$ .
4.  $x \leq x^\phi$ , and  $(x \vee y)^\phi = (x^\phi \vee y^\phi)^\phi$ , for all  $x, y \in M$ .

It is easy to verify that the following statements hold for a closure operator  $\phi$ :

1.  $(x \wedge y)^\phi \leq x^\phi \wedge y^\phi$  for all  $x, y \in M$ .
2.  $(x^\phi \wedge y^\phi)^\phi = x^\phi \wedge y^\phi$  for all  $x, y \in M$ .

A *closure system* in  $\underline{M}$  is a  $\bigwedge$ -closed subset of  $M$ . A subset  $P \subseteq M$  is a closure system in  $\underline{M}$  if, and only if,  $\{p \in P \mid x \leq p\} = \uparrow x \cap P$  has a smallest element for all  $x \in M$ . Then there is a one-to-one correspondence between closure operators and closure systems as follows. For each closure operator in  $\underline{M}$ , the set of its closures is a closure system in  $\underline{M}$ . Vice versa, if  $P$  is a closure system in  $\underline{M}$ , then  $\phi_P: x \mapsto \bigwedge(\uparrow x \cap P)$  is a closure operator in  $\underline{M}$ . The operations are mutually inverse.

It turns out that the set of all closure operators in a complete lattice constitutes a complete lattice itself, see also [16, 23]. Indeed, closure operators can be ordered by  $\preceq$ , where  $\phi \preceq \psi$  if  $\text{Clo}(\phi) \supseteq \text{Clo}(\psi)$ . This condition is equivalent to both  $\phi \circ \psi = \psi$  and  $\phi \leq \psi$  (w.r.t. pointwise order, i.e.,  $x^\phi \leq x^\psi$  for all  $x \in M$ ). Obviously, the smallest closure operator is given by the identity mapping  $\underline{\perp}: x \mapsto x$ , and the greatest closure operator is given by the constant mapping  $\overline{\top}: x \mapsto \top$ . If  $\Phi$  is a set of closure operators in  $\underline{M}$ , then its *infimum*  $\bigwedge \Phi$  and *supremum*  $\bigvee \Phi$  are given by the following equations:

$$\begin{aligned} \bigwedge \Phi: x \mapsto \bigwedge \{x^\phi \mid \phi \in \Phi\}, \\ \text{and } \bigvee \Phi: x \mapsto \bigwedge \{y \mid x \leq y \text{ and } y = y^\phi \text{ for all } \phi \in \Phi\}. \end{aligned}$$

Of course, all complete lattices are (isomorphic to) a formal concept lattice, cf. [13]. In particular, the lattice of closure operators is isomorphic to the concept lattice of  $(\text{ClOp}(\underline{M}), \text{ClOp}(\underline{M}), \preceq)$ , but also to the concept lattice of the smaller formal context  $(\text{ClOp}(\underline{M}), M, C)$  where  $\phi C x$  if  $x$  is a closure of  $\phi$ . The corresponding isomorphisms are

$$\phi \mapsto (\downarrow \phi, \text{Clo}(\phi)) \quad \text{and} \quad (\Phi, X) \mapsto \bigvee \Phi: x \mapsto \bigwedge(\uparrow x \cap X).$$

### 3 Implications in Closure Operators

An *implication* in  $M$  is an expression of the form  $p \rightarrow c$  with *premise*  $p \in M$  and *conclusion*  $c \in M$ . An element  $m \in M$  is a *model* of  $p \rightarrow c$  if  $p \leq m$  implies  $c \leq m$ . It is *valid* in  $\phi$  if all closures of  $\phi$  are models of  $p \rightarrow c$ , and we shall denote this by  $\phi \models p \rightarrow c$ . Furthermore, it can easily be shown that  $\phi \models p \rightarrow c$  is equivalent to  $c \leq p^\phi$ . We shall denote the set of all implications in  $M$  by  $\text{Imp}(M)$ . For an implication set  $\mathcal{L} \cup \{p \rightarrow c\} \subseteq \text{Imp}(M)$ , we say that  $\mathcal{L}$  *entails*  $p \rightarrow c$ , symbolized as  $\mathcal{L} \models p \rightarrow c$ , if each model of  $\mathcal{L}$ , i.e., each model of all implications in  $\mathcal{L}$ , is a model of  $p \rightarrow c$ . For each element  $x \in M$ , there is a smallest element  $x^\mathcal{L}$  above  $x$  that is a model of  $\mathcal{L}$ , and the corresponding closure operator satisfies  $x^\mathcal{L} = \bigvee \{x^{\mathcal{L},n} \mid n \geq 1\}$  where

$$x^{\mathcal{L},1} := x \vee \bigvee \{c \mid \exists p: p \rightarrow c \in \mathcal{L} \text{ and } p \leq x\},$$

and  $x^{\mathcal{L},n+1} := (x^{\mathcal{L},1})^{\mathcal{L},n}$  for each  $n \in \mathbb{N}$ .

Then,  $\mathcal{L}$  entails  $p \rightarrow c$  if, and only if,  $c \leq p^\mathcal{L}$ . Additionally, we define a similar closure operator  $\mathcal{L}^*$  where  $p \leq x$  is replaced with  $p \lesssim x$  in the definition of  $x^{\mathcal{L},1}$ . For an implication set  $\mathcal{L}$ , we symbolize by  $\mathcal{L}|_k$  the subset consisting of all implications from  $\mathcal{L}$  the premises of which have a quasi-rank not exceeding  $k$ .

A *pseudo-closure* of  $\phi$  is an element  $p \in M$  which is not a closure of  $\phi$ , but contains the closure of each strictly smaller pseudo-closure.  $\text{PsClo}(\phi)$  denotes the set of all pseudo-closures of  $\phi$ , and then  $\mathcal{B}_{\text{can}}(\phi) := \{p \rightarrow p^\phi \mid p \in \text{PsClo}(\phi)\}$  constitutes an *implicational base* for  $\phi$ , i.e., for all implications  $p \rightarrow c \in \text{Imp}(M)$ ,  $\phi \models p \rightarrow c$  if, and only if,  $\mathcal{B}_{\text{can}}(\phi) \models p \rightarrow c$ . We call  $\mathcal{B}_{\text{can}}(\phi)$  the *canonical base* of  $\phi$ , and furthermore it can be shown that it is a *minimal* implicational base, i.e., there is no base of smaller cardinality. Algorithm 1 can be utilized to compute the canonical base of  $\phi$ , for which the constraining closure operator  $\psi$  must be set to the identity mapping  $\perp$  that imposes no constraints at all.

#### 3.1 Implications in Infima

As we have seen in Section 2, the infimum of two closure operators  $\phi$  and  $\psi$  in a complete lattice  $\underline{M}$  is given as  $\phi \wedge \psi: x \mapsto x^\phi \wedge x^\psi$ . The corresponding closure system is the smallest that contains all closures of  $\phi$  as well as all closures of  $\psi$ . Consequently, an implication is valid in the infimum if, and only if, it is valid in both closure operators. For formal contexts  $\mathbb{K}_1$  and  $\mathbb{K}_2$ , the infimum of their intent closure operators can be obtained as the intent closure operator of the subposition  $\frac{\mathbb{K}_1}{\mathbb{K}_2}$ . In general, the infimum somehow corresponds to the union of two data-sets.

Using the infimum operation it is furthermore possible to construct implicational bases from streams. Suppose that  $(\phi_n)_{n \in \mathbb{N}}$  is a sequence of closure operators in  $\underline{M}$  such that  $\phi_n$  is only available at time point  $n$ , i.e., we do not have access to the whole sequence at once, but only to the most recent closure operator. However, for each time point  $n \in \mathbb{N}$ , it is desired to have a base for the implications that are valid in all closure operators  $\phi_\ell$  with  $\ell \leq n$ . This can easily be achieved as follows:

1. Set  $\mathcal{B}_0 := \mathcal{B}_{\text{can}}(\phi_0)$ .
2. Set  $\mathcal{B}_{n+1} := \mathcal{B}_{\text{can}}(\mathcal{B}_n \wedge \phi_{n+1})$  for each  $n \in \mathbb{N}$ .

By construction, then each  $\mathcal{B}_n$  is an implicational base for  $\bigwedge_{\ell=0}^n \phi_\ell$ , i.e., for the implications which are valid in  $\phi_0, \dots, \phi_n$ .

### 3.2 Implications in Suprema

Consider two closure operators  $\phi$  and  $\psi$  in a complete lattice  $\underline{M}$ . Then, their supremum  $\phi \vee \psi$  maps each element  $x \in M$  to the smallest element of  $M$  that is greater than  $x$ , and both a closure of  $\phi$  and  $\psi$ . The corresponding closure system is the smallest that contains the intersection  $\text{Clo}(\phi) \cap \text{Clo}(\psi)$ . Hence, the supremum corresponds to the intersection (or merging) of data-sets. An implication is valid in  $\phi \vee \psi$  if, and only if, it has all common closures of  $\phi$  and  $\psi$  as models. Note that there may be implications valid in  $\phi \vee \psi$  which are neither valid in  $\phi$ , nor valid in  $\psi$ .

### 3.3 Constrained Implications

As a special case of implications being valid in a supremum, we consider so-called *constrained implications*. This notion has first been defined by Belohlávek and Vychodil in [4] for formal contexts. Consider a formal context  $\mathbb{K} = (G, M, I)$  and a closure operator  $C$  on  $M$ , then a  $C$ -implication over  $M$  is an implication the premise as well as the conclusion of which are closures of  $C$ . Furthermore, a  $C$ -implication is valid in  $\mathbb{K}$  if it has as models all intents of  $\mathbb{K}$  that are closures of  $C$ . We can generalize this to the case of closure operators in complete lattices as follows.

Again, suppose that  $\phi$  and  $\psi$  are closure operators in a complete lattice  $\underline{M}$ . We then call the pair  $(\phi, \psi)$  a *closure operator with constraint* where  $\phi$  is the *constrained*, and  $\psi$  is the *constraining* closure operator. An implication  $p \rightarrow c$  is *constrained* by  $\psi$  if both its premise  $p$  and conclusion  $c$  are closures of  $\psi$ . Furthermore,  $p \rightarrow c$  is *valid* in  $(\phi, \psi)$  if it has all closures of the supremum  $\phi \vee \psi$  as models, i.e., if  $\phi \vee \psi \models p \rightarrow c$ . In order to construct bases for constrained implications in a closure operator with constraint, we shall adapt the notions of pseudo-closures and the canonical base accordingly. An element  $p \in M$  is a pseudo-closure of  $(\phi, \psi)$  if  $p$  is not a closure of  $\phi$ , but a closure of  $\psi$ , and furthermore  $q^{\phi \vee \psi} \leq p$  for all pseudo-closures  $q \leq p$  of  $(\phi, \psi)$ . Then, the set  $\mathcal{B}_{\text{can}}(\phi, \psi) := \{p \rightarrow p^{\phi \vee \psi} \mid p \in \text{PsClo}(\phi, \psi)\}$  is a minimal implicational base of  $(\phi, \psi)$  where  $\text{PsClo}(\phi, \psi)$  denotes the set of all pseudo-closures of  $(\phi, \psi)$ .

Comparing the two approaches for computing implications in a supremum, the first one in Section 3.2 only restricts the possible models, in particular, only closures of  $\phi$  are considered that are also closures of  $\psi$ . The second approach in Section 3.3 also imposes constraints on the implications, i.e., only implications are considered where both premise and conclusion are closures of  $\psi$ .

## 4 A Generalized NextClosures Algorithm

Let  $\underline{M}$  be a complete lattice. Throughout the whole section, we assume that there is a strict order-preserving function  $|\cdot|: (M, \leq) \rightarrow (\mathbb{N}, \leq)$ , i.e.,  $x \leq y$  implies  $|x| \leq |y|$  for all  $x, y \in M$ . W.l.o.g.  $|\perp| = 0$ . (If  $|\perp| = n$  for  $n > 0$ , then  $\|\cdot\|: x \mapsto |x| - n$  is also a strict order-homomorphism, and  $\|\perp\| = 0$ .) Then  $|\cdot|$  is a *quasi-rank function* on  $(M, \leq)$ , and we say that an element  $x \in M$  has *quasi-rank*  $|x|$ . In particular, for all *graded* complete lattices, the corresponding rank function  $|\cdot|$  is a quasi-rank function such that furthermore  $|x| + 1 = |y|$  if  $x$  is a lower neighbor of  $y$ . For our purposes, we do not need the additional property that the ranks of neighboring elements only differ by an amount of 1.

Consider two closure operators  $\phi$  and  $\psi$  in  $\underline{M}$ . Then the set containing all closures of  $\phi \curlywedge \psi$  and all pseudo-closures of  $(\phi, \psi)$  is a closure system in  $\underline{M}$ , and the corresponding closure operator is  $(\phi, \psi)^* := \mathcal{B}_{\text{can}}(\phi, \psi)^*$ . Our aim is to find an automatic method for constructing all closures of  $(\phi, \psi)^*$ . The following Algorithm 1 has been introduced in [18], and the special case of  $\phi$  being the intent closure operator induced by a formal context, and the constraining closure operator  $\psi$  being the identity  $\perp$  (i.e., no constraints are given), has been handled in [20]. The proof is left out here, as it can be found in [18], or can be generalized from the proof in [20]. We will describe Algorithm 1 in the following text.

According to the definition of a pseudo-closure, for all elements  $p \in M$ , it suffices to know all strictly smaller pseudo-closures  $q \lesssim p$  in order to correctly determine whether  $p$  is pseudo-closed. Hence, we may compute them in a level-wise approach w.r.t. increasing quasi-rank, since if we have computed all pseudo-closures  $q$  where  $|q| \lesssim |p|$ , then we can find those pseudo-closures with  $q \lesssim p$  among them. In particular, if  $\mathcal{L}$  is a set of  $\psi$ -implications that contains exactly all implications  $p \rightarrow p^{\phi \curlywedge \psi}$  where  $p$  is a  $(\phi, \psi)$ -pseudo-closure with quasi-rank not exceeding  $k$ , and some arbitrary implications with premise quasi-rank  $k + 1$ , then the closure operators  $(\phi, \psi)^*$  and  $\mathcal{L}^*$  coincide on all elements with a quasi-rank of at most  $k + 1$ , i.e.,  $|x| \leq k + 1$  implies  $x^{(\phi, \psi)^*} = x^{\mathcal{L}^*}$ . Furthermore, it can be proven that the following statements are satisfied:

1. If  $p \in M$  is a  $(\phi, \psi)$ -pseudo-closure, then there is neither a  $\phi \curlywedge \psi$ -closure nor a  $(\phi, \psi)$ -pseudo-closure strictly between  $p$  and  $p^{\phi \curlywedge \psi}$ .
2. If  $m \in M$  is a  $\phi \curlywedge \psi$ -closure, then the next  $\phi \curlywedge \psi$ -closures or  $(\phi, \psi)$ -pseudo-closures are of the form  $n^{(\phi, \psi)^*}$  for upper neighbors  $n \succ m$ .
3. If  $x, y \in M$  with  $x \lesssim y$  are neighboring  $(\phi, \psi)^*$ -closures, then  $y = z^{(\phi, \psi)^*}$  for all upper neighbors  $z \succ x$  with  $z \leq y$ .

---

**Algorithm 1.** NextClosures
 

---

**Input:** a complete lattice  $\underline{M} = (M, \leq, \wedge, \vee, \top, \perp)$   
**Input:** a quasi-rank function  $|\cdot|: (M, \leq) \rightarrow (\mathbb{N}, \leq)$   
**Input:** a closure operator with constraint  $(\phi, \psi)$  in  $\underline{M}$   
**Initialize:** a set  $\mathcal{I} := \emptyset$   
**Initialize:** an implication set  $\mathcal{L} := \emptyset$   
**Initialize:** a candidate set  $\mathbf{C} := \{\perp\}$

- 1 for  $k = 0, 1, \dots, |\top|$  do
- 2   for all  $c \in \mathbf{C}$  with  $|c| = k$  do in parallel
- 3     if  $c = c^{\mathcal{L}^*}$  then
- 4       if  $c \neq c^{\phi \curlywedge \psi}$  then
- 5           $\mathcal{L} := \mathcal{L} \cup \{c \rightarrow c^{\phi \curlywedge \psi}\}$
- 6           $\mathcal{I} := \mathcal{I} \cup \{c^{\phi \curlywedge \psi}\}$
- 7           $\mathbf{C} := \mathbf{C} \cup \{d \mid c^{\phi \curlywedge \psi} \prec d\}$
- 8     else
- 9        $\mathbf{C} := \mathbf{C} \cup \{c^{\mathcal{L}^*}\}$
- 10   Wait for termination of all parallel processes.

**Output:** the set  $\mathcal{I}$  of all  $(\phi, \psi)$ -closures  
**Output:** the canonical base  $\mathcal{L}$  of  $(\phi, \psi)$

---

Algorithm 1 manages a set  $\mathbf{C}$  of candidates, an implication set  $\mathcal{L}$ , and a set  $\mathcal{I}$  of closures. Initially,  $\perp$  is the only candidate in  $\mathbf{C}$ . Algorithm 1 is *in state*  $k$  if it has processed all candidates with a quasi-rank  $\leq k$ , but none of quasi-rank  $> k$ . In state  $k$ ,  $\mathbf{C}_k$  denotes the set of candidates,  $\mathcal{L}_k$  denotes the implication set, and  $\mathcal{I}_k$  denotes the set of closures. It can be shown that the following invariants are always satisfied during Algorithm 1's run:

1.  $\mathbf{C}_k$  contains all pseudo-closures of  $(\phi, \psi)$  with quasi-rank  $k + 1$ , and all closures of  $\phi \Upsilon \psi$  with quasi-rank  $k + 1$  that are not already contained in  $\mathcal{I}_k$ .
2.  $\mathcal{I}_k$  contains exactly all closures of  $\phi \Upsilon \psi$  with a quasi-rank not exceeding  $k$ .
3.  $\mathcal{L}_k$  contains exactly all implications  $p \rightarrow p^{\phi \Upsilon \psi}$  for which the premise  $p$  is a pseudo-closure of  $(\phi, \psi)$  with a quasi-rank of at most  $k$ .
4. Between the states  $k$  and  $k + 1$ , the closure operators  $(\phi, \psi)^*$  and  $\mathcal{L}^*$  coincide on all elements with quasi-rank  $k + 1$ , i.e., an element with quasi-rank  $k + 1$  is either a closure of  $\phi \Upsilon \psi$  or a pseudo-closure of  $(\phi, \psi)$  if, and only if, it is a closure of  $\mathcal{L}^*$ .

As a consequence, we obtain that in the final state  $|\top|$ , Algorithm 1 outputs the set of all closures of  $\phi \Upsilon \psi$  as well as the canonical base of  $(\phi, \psi)$ .

#### 4.1 A Generalized Parallel Attribute Exploration

As an extension, we suppose that the given closure operator only describes a part of the data-set, and furthermore there is an expert available (very much in the same way as for *Attribute Exploration* [10, 11] in the case of *Formal Concept Analysis*). The exploration of a closure operator in a lattice is explained in full detail in [18], and the special case of exploring a formal context is considered in [19] (including extensive proofs). However, we will give a short summary here.

An *expert* in a lattice  $\underline{M}$  is a partial mapping  $\chi: \text{Imp}(M) \rightarrow_p M$  such that the following conditions hold:

1. If  $\chi(p \rightarrow c)$  is defined, then the value is a counterexample against  $p \rightarrow c$ , i.e.,  $\chi(p \rightarrow c) = m$  implies  $p \leq m$  and  $c \not\leq m$ .
2. If  $\chi(p \rightarrow c)$  is undefined, then all counterexamples of  $\chi$  are models of  $p \rightarrow c$ , i.e., whenever  $\chi(q \rightarrow d) = m$ , then  $p \leq m$  implies  $c \leq m$ .

We say that  $\chi$  *accepts*  $p \rightarrow c$  if it does not provide a counterexample against  $p \rightarrow c$ , i.e.,  $\chi(p \rightarrow c)$  is undefined, and that  $\chi$  *refutes*  $p \rightarrow c$  otherwise. Furthermore,  $\chi$  is *induced* by a closure operator  $\phi$  if for all implications  $p \rightarrow c$ ,  $\chi(p \rightarrow c)$  is undefined if, and only if,  $p \rightarrow c$  is valid in  $\phi$ . The expert  $\chi$  is *optimal* if it accepts  $p \rightarrow c \wedge x$  whenever it refutes  $p \rightarrow c$  with counterexample  $x$ . For an arbitrary expert  $\chi$ , it is always possible to construct an optimal expert  $\hat{\chi}$  that accepts the same implications as follows: Let  $p \rightarrow c$  be an implication. Then  $\hat{\chi}$  accepts  $p \rightarrow c$  if  $\chi$  accepts  $p \rightarrow c$ . Otherwise, let  $c_0 := c$  and  $n := 0$ . While  $\chi$  rejects  $c_n$  with counterexample  $x_n$ , set  $c_{n+1} := c_n \wedge x_n$ , and increase  $n$ . Eventually, define  $\hat{\chi}(p \rightarrow c) := \bigwedge_{k=0}^n c_k$ .

As input the generalized *Parallel Attribute Exploration* requires a closure operator  $\phi$ , an expert  $\chi$ , as well as an implication set  $\mathcal{L}$  in  $\underline{M}$ , such that there is an inaccessible closure operator  $\delta$  describing the domain of interest where  $\chi$  is induced by  $\delta$ ,  $\delta \preceq \phi$ , i.e.,  $\text{Clo}(\phi) \subseteq \text{Clo}(\delta)$ , and  $\mathcal{L}$  is valid in  $\delta$ . W.l.o.g. we may assume that  $\chi$  is optimal. The goal is to compute an implicational base for the domain closure operator  $\delta$  by utilizing the background knowledge  $\mathcal{L}$ , and querying the expert  $\chi$ .

In order to insert new closures into the closure operator  $\phi$ , we define an operation  $\downarrow: \text{Clop}(\underline{M}) \times M \rightarrow \text{Clop}(\underline{M})$  by  $\phi \downarrow x := \phi \wedge \rho_x$  where  $\rho_x(m) := x$  if  $m \leq x$ , and  $\rho_x(m) := \top$  otherwise. It is readily verified that  $\phi \downarrow x$  is the greatest closure operator below  $\phi$  that has  $x$  as a closure. Then  $x \models y \rightarrow y^\phi$  implies  $y^{\phi \downarrow x} = y^\phi$ . Furthermore, if  $x$  is a model of all implications  $p \rightarrow p^\phi$  where  $p$  is a pseudo-closure of  $(\phi, \mathcal{L})$  with  $|p| \leq k$ , then the pseudo-closures of  $(\phi, \mathcal{L})$  and  $(\phi \downarrow x, \mathcal{L})$  with a quasi-rank of at most  $k$  are the same. As a corollary, we get that then also the canonical bases of  $(\phi, \mathcal{L})$  and  $(\phi \downarrow x, \mathcal{L})$  coincide for the implications the premises of which have a quasi-rank not exceeding  $k$ . By an inductive application, it follows that  $\mathcal{B}_{\text{can}}(\phi, \mathcal{L}) \upharpoonright_k = \mathcal{B}_{\text{can}}((\phi \downarrow x_1, \dots, x_n), \mathcal{L}) \upharpoonright_k$  if each  $x_i$  is a model of  $\mathcal{B}_{\text{can}}(\phi, \mathcal{L}) \upharpoonright_k$ . This fact enables us to execute a parallel attribute exploration.

---

**Algorithm 2.** ParallelAttributeExploration
 

---

**Input:** a complete lattice  $\underline{M} = (M, \leq, \wedge, \vee, \top, \perp)$   
**Input:** a quasi-rank function  $|\cdot|: (M, \leq) \rightarrow (\mathbb{N}, \leq)$   
**Input:** a closure operator  $\phi$  in  $\underline{M}$   
**Input:** an expert  $\chi$  in  $\underline{M}$   
**Input:** a set  $\mathcal{L}$  of background implications that are valid in  $\phi$   
**Initialize:** an implication set  $\mathcal{B} := \emptyset$   
**Initialize:** a candidate set  $\mathbf{C} := \{\perp\}$   
1 for  $k = 0, \dots, |\top| - 1$  do  
2   for all  $c \in \mathbf{C}$  with  $|c| = k$  do in parallel  
3     if  $c = c^{\mathcal{B}^*}$  and  $c = c^\mathcal{L}$  then  
4       while  $c \neq c^\phi$  and  $\chi(c \rightarrow c^\phi) = x$  do  
5           $\phi := \phi \downarrow x$   
6          if  $c \neq c^\phi$  then  
7            $\mathcal{B} := \mathcal{B} \cup \{c \rightarrow c^\phi\}$   
8            $\mathbf{C} := \mathbf{C} \cup \{d \mid d \succ c^\phi\}$   
9       else  
10        $\mathbf{C} := \mathbf{C} \cup \{c^{\mathcal{B}^* \vee \mathcal{L}}\}$   
11   Wait for termination of all parallel processes.  
**Output:** the refined closure operator  $\phi$   
**Output:** the canonical base  $\mathcal{B}$  of  $\phi$  relative to  $\mathcal{L}$

---

Algorithm 2 is *in state*  $k$  if it has processed all candidates of a quasi-rank not exceeding  $k$ , but none of a greater quasi-rank. Let  $\mathbf{C}_k$  denote the set of candidates in state  $k$ , and let  $\mathcal{B}_k$  denote the implication set in state  $k$ . Furthermore,  $x_k^1, \dots, x_k^{n_k}$  denote all counterexamples provided by the expert between states  $k$  and  $k+1$ , and  $\phi_k$  is the closure operator in state  $k$ , i.e.,  $\phi_k \downarrow x_k^1, \dots, x_k^{n_k} = \phi_{k+1}$ . Then the following statements are always satisfied.

1.  $\mathbf{C}_k$  contains all pseudo-closures of  $(\phi_{k+1}, \mathcal{L})$  with quasi-rank  $k+1$ .
2.  $\mathcal{B}_k$  consists of all implications  $p \rightarrow p^{\phi \vee \psi}$  where  $p$  is a pseudo-closure of  $(\phi_k, \mathcal{L})$  with a quasi-rank of at most  $k$ , i.e.,  $\mathcal{B}_{\text{can}}(\phi_k, \mathcal{L}) \upharpoonright_k = \mathcal{B}_k$ .
3. Between the states  $k$  and  $k+1$ , every element with quasi-rank  $k+1$  is a closure of  $\mathcal{B}^*$  if, and only if, it is either a closure of  $\phi_{k+1} \vee \mathcal{L}$  or a pseudo-closure of  $(\phi_{k+1}, \mathcal{L})$ .

As a corollary, we infer that in the final state  $|\top|$ , Algorithm 2 returns a refinement of  $\phi$  that has the same closures as  $\delta$ , and a minimal implicational base of  $\delta$  relative to  $\mathcal{L}$ , i.e., a minimal superset of  $\mathcal{L}$  that constitutes an implicational base of  $\delta$ . Additionally,



there is no algorithm that computes a minimal relative implicational base of  $\delta$ , but poses less questions to  $\chi$  than Algorithm 2.

#### 4.2 A Problem for the Exploration of Constrained Implicational Bases

A further generalization of Algorithm 2 to arbitrary closure operators  $\psi$  instead of implication sets  $\mathcal{L}$  valid in  $\phi$  is not easily possible. We may only replace  $\mathcal{L}$  by a closure operator  $\psi$  such that  $\psi \preceq \phi$ , otherwise when receiving counterexamples from the expert, they may not be inserted into the closure operator  $\phi$ , but must be inserted into the supremum  $\phi \vee \psi$ , since we cannot ensure that  $y^{(\phi \downarrow x) \vee \psi} = y^{\phi \vee \psi}$  if  $x$  is a  $\psi$ -model of the  $\psi$ -implication  $y \rightarrow y^{\phi \vee \psi}$ . This is due to the fact that the lattice of closure operators is not distributive, and this fact can be proven by means of [22, Lemmata 1 and 2].

### 5 Applications

In this section, we will present some applications of the generalized *NextClosures* algorithm. It is trivial that there is an application for *Formal Concept Analysis*, since the powerset of the attribute set constitutes a complete lattice, and the composition of the derivation operators is a closure operator. Further applications can be found for interpretations in *Description Logics*, and for *Pattern Structures*. We will shortly describe each of the cases in the following subsections.

#### 5.1 Application: Formal Concept Analysis

For each formal context  $\mathbb{K} = (G, M, I)$ , the mapping  $II: X \mapsto X^{II}$  is a closure operator in the powerset  $\wp(M)$ . The closure system induced by  $II$  is the set of all intents of  $\mathbb{K}$ , and thus it is easy to verify that an implication is valid in  $\mathbb{K}$  if, and only if, it is valid in  $II$ . Hence, Algorithm 1 can be utilized to compute both the set of intents of  $\mathbb{K}$  and the canonical base of  $\mathbb{K}$ . If furthermore a constraining closure operator  $C$  in  $\wp(M)$  is given, cf. [4] as well as Section 3.3, then Algorithm 1 is also applicable, and computes all  $C$ -intents as well as a base of the  $C$ -implications.

As a special case, we may consider the case where the constraining closure operator  $C$  is given by means of an implication set  $\mathcal{L} \subseteq \text{Imp}(M)$ , i.e.,  $C: X \mapsto X^{\mathcal{L}}$  assigns to each attribute set  $X \subseteq M$  the smallest superset that is a model of  $\mathcal{L}$ . Then there are two cases to consider: either  $\mathbb{K} \models \mathcal{L}$ , or  $\mathbb{K} \not\models \mathcal{L}$ .

First, assume that  $\mathcal{L}$  is valid in  $\mathbb{K}$ . Then each intent of  $\mathbb{K}$  is a model of  $\mathcal{L}$ , i.e.,  $\mathcal{L} \preceq II$ , and thus  $II \wedge \mathcal{L} = \mathcal{L}$  as well as  $II \vee \mathcal{L} = II$ . Consequently, it is uninteresting to consider the infimum or supremum. Furthermore, it can be shown that the canonical base of  $\mathbb{K}$  with constraint  $\mathcal{L}$  from Section 3.3 coincides with the canonical base of  $\mathbb{K}$  relative to  $\mathcal{L}$  from [9, 24], i.e., is a minimal extension of  $\mathcal{L}$  which constitutes an implicational base for  $\mathbb{K}$ . However, both documents [9, 24] have not addressed the remaining case where  $\mathcal{L}$  is not valid in  $\mathbb{K}$ .

Second, let  $\mathbb{K} \not\models \mathcal{L}$ . Then, an implication is valid in the infimum  $II \wedge \mathcal{L}$  if, and only if, it is valid in  $\mathbb{K}$  as well as is entailed by  $\mathcal{L}$ . This is useful for the case where a sequence  $(\mathbb{K}_n)_{n \in \mathbb{N}}$  of formal contexts is observed, and for each time point  $n$ , a base for the subposition of  $\mathbb{K}_0, \dots, \mathbb{K}_n$  shall be constructed. Note that this has already been discussed in Section 3.1 for the general case.

For the supremum, an implication is valid if, and only if, it has as models all intents of  $\mathbb{K}$  that are also models of  $\mathcal{L}$ . Consider the case where  $\mathbb{K}$  contains faulty objects, e.g., due to wrong observations inserted to  $\mathbb{K}$ , and where the implication set  $\mathcal{L}$  has been manually created or verified to ensure that all its implications are valid in the domain of interest. Then, constructing implications that are valid in  $II \vee \mathcal{L}$  is a suitable method, since only those intents of  $\mathbb{K}$  are allowed as models that are already models of  $\mathcal{L}$ , i.e., satisfy the existing implicational knowledge. A further restriction is imposed by also requiring the premises and conclusions to be models of  $\mathcal{L}$ , i.e., only considering implications that are  $\mathcal{L}$ -constrained. For both cases, Algorithm 1 can be used to compute a canonical base in parallel. However, it remains to investigate whether the implications valid in  $II \vee \mathcal{L}$ , or the  $\mathcal{L}$ -constrained implications of  $\mathbb{K}$ , are more useful in practise.

## 5.2 Application: Pattern Structures

A *pattern structure*, as introduced by Ganter and Kuznetsov in [12], is a triple  $(G, (M, \sqcap), \delta)$  consisting of a set  $G$  of *objects*, a semi-lattice  $(M, \sqcap)$  of *patterns*, and a *description function*  $\delta: G \rightarrow M$ , such that the set  $\{\delta(g) \mid g \in G\}$  induces a complete sub-semi-lattice  $(M_\delta, \sqcap)$  of  $(M, \sqcap)$ . Furthermore, then a galois connection between the powerset  $\wp(G)$  and the pattern lattice  $(M, \sqsubseteq)$  (where  $x \sqsubseteq y$  iff  $x \sqcap y = x$ ) is given as follows:

$$\begin{aligned} A^\square &:= \bigsqcap \{\delta(g) \mid g \in A\} \quad \text{for all } A \subseteq G, \\ d^\square &:= \{g \in G \mid \delta(g) \sqsubseteq d\} \quad \text{for all } d \in M. \end{aligned}$$

A *pattern implication* is a term  $x \rightarrow y$  where  $x, y \in M$ , and is valid in  $(G, (M, \sqcap), \delta)$  if  $x^\square \subseteq y^\square$ . Since  $(\cdot^\square, \cdot^\square)$  is a galois connection, it follows that the composition  $\cdot^{\square\square}$  is a closure operator in the lattice of all descriptions. Hence, it is possible to apply the generalized *NextClosures* algorithm for the case of pattern structures, too, provided that there is a quasi-rank function  $|\cdot|: (M, \sqsubseteq) \rightarrow (\mathbb{N}, \leq)$ .

## 5.3 Application: Description Logics

As a special case of pattern structures, we consider interpretations in *Description Logics* [2] as input data-sets. First, fix a finite signature  $(N_C, N_R)$ , i.e.,  $N_C$  is a set of *concept names*, and  $N_R$  is a set of *role names*. An *interpretation* is a pair  $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$  consisting of a non-empty set  $\Delta^\mathcal{I}$ , called *domain*, and an *extension function*  $\cdot^\mathcal{I}$  such that  $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$  for all concept names  $A \in N_C$ , and  $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$  for all role names  $r \in N_R$ . An  $\mathcal{EL}^\perp$ -*concept description* may be built according to the following inductive rule:

$$C ::= \perp \mid \top \mid A \mid C \sqcap C \mid \exists r. C.$$

Then, the extension function  $\cdot^\mathcal{I}$  is extended to all  $\mathcal{EL}^\perp$ -concept descriptions by means of the following definitions:

$$\begin{aligned} \perp^\mathcal{I} &:= \emptyset, \\ \top^\mathcal{I} &:= \Delta^\mathcal{I}, \\ (C \sqcap D)^\mathcal{I} &:= C^\mathcal{I} \cap D^\mathcal{I}, \\ \text{and } (\exists r. C)^\mathcal{I} &:= \{d \in \Delta^\mathcal{I} \mid \exists e \in \Delta^\mathcal{I}: (d, e) \in r^\mathcal{I} \text{ and } e \in C^\mathcal{I}\}. \end{aligned}$$

Please note that an implication between concept descriptions is rather called a *general concept inclusion*. An implicational base for the closure operator  $\cdot^{\mathcal{II}}$ , where the first  $\cdot^{\mathcal{I}}$  is the extension mapping, and the second  $\cdot^{\mathcal{I}}$  is the model-based most-specific concept description mapping, is a base of GCIs for  $\mathcal{I}$ , as it has been investigated by Baader and Distel in [1, 8].

The set of all  $\mathcal{EL}^\perp$ -concept descriptions constitutes a bounded lattice. First we define a quasi-order  $\sqsubseteq$ , the *subsumption order*, by  $C \sqsubseteq D$  if  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$  for all interpretations  $\mathcal{I}$ . The corresponding equivalence relation is denoted as  $\equiv$ . For simplicity, we do not distinguish between equivalence classes and their representatives. The set of all equivalence classes is a bounded lattice with the smallest element  $\perp$ , the greatest element  $\top$ , and where the *conjunction*  $\sqcap$  is the infimum operation, and the *least common subsumer mapping*  $\sqcup$  is the supremum operation. It is easy to verify that both  $\sqcap$  and  $\sqcup$  are compatible with  $\equiv$ , hence they are well-defined for the equivalence classes.

If we do not restrict the role-depth, and use descriptive semantics (as defined above) instead of greatest-fixpoint semantics, then however the lattice of concept descriptions is not complete, as there are infinitely many mutually distinct  $\mathcal{EL}^\perp$ -concept descriptions, e.g., the concept descriptions  $(\exists r)^n A$  for  $n \in \mathbb{N}$ , and conjunction is only allowed for a finite number of concept descriptions. It turns out, that this is no restriction for the application of Algorithm 1, since for finite interpretations (i.e., with a finite domain), the canonical base is finite, and hence for the computation of closures w.r.t. the current implication set in Algorithm 1 only finitary suprema are necessary.

For defining a quasi-rank function, we may first observe that the dual lattice of  $\mathcal{EL}^\perp$ -concept descriptions is well-founded, i.e., there are no infinite strictly ascending chains  $C_0 \sqsubset C_1 \sqsubset C_2 \sqsubset \dots$ , cf. Baader and Morawska in [3, Proposition 3.5]. Hence, a quasi-rank function could be defined by  $|C| := n$  where  $n$  is the maximal length of a chain starting at  $C$  and ending at  $\top$ . Since each concept description has only finitely many subsumers (for a finite signature), there are only finitely many mutually distinct chains in the interval  $[C, \top]$ , and thus  $|\cdot|$  as above is well-defined. For the same reason, the set of upper neighbors of an arbitrary concept description is computable.

## 6 Conclusion

In this document, the *NextClosures* algorithm has been generalized to arbitrary closure operators in complete lattices, and has been extended to handle constraints that are given in form of another closure operator. Furthermore, some exemplary applications have been presented, e.g., in the field of *Formal Concept Analysis*, *Description Logics*, and for *Pattern Structures*. While some experiments of an implementation specialized to formal contexts as input structures have shown that there is an almost inverse linear correlation between the computation time and the number of available processor cores, experiments for other types of input structures are outstanding. A complexity analysis taking into account the complexities of the lattice operations as well as of the closure operator could be interesting.

*Acknowledgements* The author gratefully thanks the anonymous reviewers for their constructive hints and helpful remarks.

## References

- [1] Franz Baader and Felix Distel. “A Finite Basis for the Set of EL-Implications Holding in a Finite Model”. In: *Formal Concept Analysis, 6th International Conference, ICFCA 2008, Montreal, Canada, February 25-28, 2008, Proceedings*. 2008, pp. 46–61.
- [2] Franz Baader and Carsten Lutz. “Description Logic”. In: *Handbook of Modal Logic*. Ed. by Patrick Blackburn, Johan van Benthem, and Frank Wolter. Elsevier, 2006. Chap. 13.
- [3] Franz Baader and Barbara Morawska. “Unification in the Description Logic EL”. In: *Logical Methods in Computer Science* 6.3 (2010).
- [4] Radim Belohlávek and Vilém Vychodil. “Formal Concept Analysis with Constraints by Closure Operators”. In: *Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures, ICCS 2006, Aalborg, Denmark, July 16-21, 2006, Proceedings*. 2006, pp. 131–143.
- [5] Garrett Birkhoff. “Lattice theory”. In: *Colloquium Publications*. 3rd ed. Vol. 25. Amer. Math. Soc., 1967.
- [6] Nathalie Caspard and Bernard Monjardet. “The Lattices of Closure Systems, Closure Operators, and Implicational Systems on a Finite Set: A Survey”. In: *Discrete Applied Mathematics* 127.2 (2003), pp. 241–269.
- [7] Brian A. Davey and Hilary Ann Priestley. *Lattices and Order*. 2nd ed. Cambridge University Press, Apr. 2002.
- [8] Felix Distel. “Learning description logic knowledge bases from data using methods from formal concept analysis”. PhD thesis. Dresden University of Technology, 2011.
- [9] Bernhard Ganter. “Attribute Exploration with Background Knowledge”. In: *Theor. Comput. Sci.* 217.2 (1999), pp. 215–233.
- [10] Bernhard Ganter. *Two Basic Algorithms in Concept Analysis*. FB4-Preprint 831. Darmstadt, Germany: Technische Hochschule Darmstadt, 1984.
- [11] Bernhard Ganter. “Two Basic Algorithms in Concept Analysis”. In: *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*. 2010, pp. 312–340.
- [12] Bernhard Ganter and Sergei O. Kuznetsov. “Pattern Structures and Their Projections”. In: *Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings*. 2001, pp. 129–142.
- [13] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. 1st ed. Springer, Dec. 1999.
- [14] George Grätzer. *General Lattice Theory*. 2nd ed. Birkhäuser Verlag, 1998.
- [15] Jean-Luc Guigues and Vincent Duquenne. “Famille minimale d’implications informatives résultant d’un tableau de données binaires”. In: *Mathématiques et Sciences Humaines* 95 (1986), pp. 5–18.
- [16] Akira Higuchi. “Lattices of closure operators”. In: *Discrete Mathematics* 179.1-3 (1998), pp. 267–272.
- [17] Francesco Kriegel. *Concept Explorer FX*. Software for Formal Concept Analysis, available at <https://github.com/francesco-kriegel/conexp-fx>. 2010–2016.
- [18] Francesco Kriegel. *NextClosures – Parallel Exploration of Constrained Closure Operators*. LTCS-Report 15-01. Dresden, Germany: Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2015.
- [19] Francesco Kriegel. “Parallel Attribute Exploration”. In: *Graph-based Modeling of Conceptual Structures - 22nd International Conference on Conceptual Structures, ICCS 2016, Annecy, France, July 5-7, 2016, Proceedings*. 2016.
- [20] Francesco Kriegel and Daniel Borchmann. “NextClosures: Parallel Computation of the Canonical Base”. In: *Proceedings of the 12th International Conference on Concept*

*Lattices and Their Applications (CLA 2015), Clermont-Ferrand, France, October 13-16, 2015.* 2015, pp. 181–192.

- [21] David Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.
- [22] José Morgado. “Note on the distributive closure operators of a complete lattice”. In: *Portugaliae Mathematica* 23 (1964), pp. 11–25.
- [23] Sebastian Rudolph. “On the Succinctness of Closure Operator Representations”. In: *Formal Concept Analysis - 12th International Conference, ICFCA 2014, Cluj-Napoca, Romania, June 10-13, 2014. Proceedings.* 2014, pp. 15–36.
- [24] Gerd Stumme. “Attribute Exploration with Background Implications and Exceptions”. In: *Data Analysis and Information Systems*. Ed. by Hans-Hermann Bock and Wolfgang Polasek. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, 1996, pp. 457–469.
- [25] Marcel Wild. “A Theory of Finite Closure Spaces Based on Implications”. In: *Advances in Mathematics* 108.1 (1994), pp. 118–139.