A Depth-first Search Algorithm for Computing Pseudo-closed Sets

Alexandre Bazin

A Depth-first Search Algorithm for Computing Pseudo-closed Sets

Alexandre Bazin[1,2]

(1) Université Clermont Auvergne, Université Blaise Pascal, LIMOS, BP 10448, F-63000 CLERMONT-FERRAND, FRANCE
(2) CNRS, UMR 6158, LIMOS, F-63178 AUBIERE, FRANCE
contact@alexandrebazin.com

Abstract

The question of the lower bounds for the delay in the computation of the Duquenne-Guigues implication basis in non-lectic orders is still open. As a step towards an answer, we propose an algorithm that can enumerate pseudo-closed sets in orders that do not necessarily extend the inclusion order using depth-first searches in a sequence of closure systems. Empirical comparisons with NextClosure on the runtime and number of closed sets computed are provided.

Keywords:
Implication, Pseudo-closed set

1. Introduction

Implications, linked to closure operators, are not only interesting mathematical objects worth studying in their abstract form but they also find their application in database theory and data mining in the form of functional dependencies and certain association rules. Implications are essentially relations between elements of a powerset 2^E and, as often with this type of constructs, their number is exponential in $|E|$ and most of them are redundant. As such, the interest revolves around finding smaller, non redundant sets of implications. The smallest such set, the subject of this work, is the Duquenne-Guigues basis.

The Duquenne-Guigues basis is constructed through the enumeration of so-called pseudo-closed sets. This enumeration has been studied most notably in the field of Formal Concept Analysis (FCA) and known results on counting, recognizing and computing these objects are mentioned in Section 2.2. The biggest open question concerns the delay of this enumeration in the general case. Indeed, it has been shown that pseudo-closed sets cannot be enumerated with a polynomial delay in the lectic or reverse lectic orders unless P = NP [9, 2] but no such result exists for other orders.

Two algorithms (and their variants) can be used to compute pseudo-closed sets. The first and most popular, NextClosure [11], uses the lectic order and thus cannot enumerate with a polynomial delay unless P = NP. The second [17] uses an incremental approach and is not necessarily limited to the lectic order. Both algorithms, however, have to enumerate closed sets along with pseudo-closed sets. As the number of closed sets can be exponential in the number of pseudo-closed sets, this severely reduces hopes of obtaining a good delay with these methods. We believe that the need to compute all closed sets is linked to the way pseudo-closed sets are recognized, which forces an enumeration in orders that extend the inclusion order. That is why we propose here an algorithm that can enumerate pseudo-closed sets in orders that do not necessarily extend the inclusion order.

Section 2 covers the basic required notions on lattices, closure operators and implications. In Section 3, we present a new way of recognizing a pseudo-closed set that does not require explicit knowledge of all its subsets and also propose an algorithm for finding a single pseudo-closed set. In Section 4, we describe an algorithm for computing the Duquenne-Guigues basis of a closure operator. Section 5 presents empirical results on the number of closed sets that our algorithm computes and comparisons with NextClosure.

2. Preliminaries

2.1. Lattices and Closure Systems

In this section, we recall the basic notions of lattice theory we will use in this work. For a deeper understanding of the subject, we refer the reader to [7].

Definition 1. *Let $(E, \leq)$ be a partially ordered set. For any $S \subseteq E$, we call infimum (noted $\bigwedge S$) the greatest $e \in E$ such that $\forall s \in S$, $e \leq s$. Dually, we call supremum (noted $\bigvee S$) the smallest $e \in E$ such that $\forall s \in S$, $s \leq e$. A lattice is a partially ordered set in which any pair of elements has a unique infimum and a unique supremum.*

For example, $(2^X, \subseteq)$ is the Boolean lattice composed of the subsets of X ordered by inclusion.

Definition 2. *Let $L = (E, \leq)$ be a lattice. For any $x, y \in E$, we say that x is a lower (resp. upper) cover of y if x is maximal (resp. minimal) such that $x \leq y$ (resp. $y \leq x$). An element $e \in E$ is said to be meet-irreducible (resp. join-irreducible) if it has a single upper (resp. lower) cover.*

The sets of meet- and join-irreducibles of L are denoted by $\mathcal{M}(L)$ and $\mathcal{J}(L)$ respectively. Any element $e \in E$ is the infimum (resp. supremum) of $\{m \in \mathcal{M}(L) \mid e \leq m\}$ (resp. $\{j \in \mathcal{J}(L) \mid j \leq e\}$). As such, both sets completely represent the lattice.

	a	b	c	d	e
o1	×	×			
o2		×	×	×	
o3		×		×	×
o4			×		×
o5				×	×

Table 1: A Formal Context

Definition 3. *In a lattice $(E, \leq)$, a chain is a set $C \subseteq E$ such that any two elements of C are comparable. A chain $C = c_1 \leq ... \leq c_n$ is maximal if c_{i+1} is an upper cover of c_i for any $1 \leq i < n$.*

Definition 4. *Let E be a set. An operator $c : 2^E \mapsto 2^E$ that is idempotent ($c(c(X)) = c(X)$), extensive ($X \subseteq c(X)$) and monotone ($X \subseteq Y \Rightarrow c(X) \subseteq c(Y)$) is a closure operator.*

A set $S \in 2^E$ such that $S = c(S)$ is said to be *closed* under c.

Proposition 1. *Let $X, Y \in 2^E$ be sets and c a closure operator. The set $c(X) \cap c(Y)$ is closed under c.*

Throughout this paper, we will use $\Phi_c = (\{X \mid X = c(X)\}, \subseteq)$ to denote the lattice of sets closed under a closure operator c ordered by the inclusion relation.

The most commonly used closure operator in FCA literature is the derivation operator $\cdot''$ that we will use as an example in this work.

Definition 5. *A formal context is a triple $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ where $\mathcal{O}$ is a set of objects, $\mathcal{A}$ a set of attributes and $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$ a binary relation that associates sets of attributes to objects.*

The derivation operator $\cdot''$, associated to the context, is the composition of the two operators

$$\cdot' : 2^{\mathcal{O}} \mapsto 2^{\mathcal{A}}$$

$$O' = \{a \in \mathcal{A} \mid \forall o \in O, (o, a) \in \mathcal{R}\}$$

and

$$\cdot' : 2^{\mathcal{A}} \mapsto 2^{\mathcal{O}}$$

$$A' = \{o \in \mathcal{O} \mid \forall a \in A, (o, a) \in \mathcal{R}\}$$

Example 1. *In the remainder of this work, we will abuse set notations for attribute sets and use, for example, abc to denote* $\{abc\}$*. In the context depicted in Table 1, the set of closed attribute sets is* $\{\emptyset$*, b, c, d, e, ab, bd, ce, de, bcd, bde, abcde*$\}$*.*

2.2. Implications

Now, let us present some definitions and results on implications.

Definition 6. *An implication is a pair* $(A, B) \in 2^E \times 2^E$*, often noted* $A \to B$*.*

Definition 7. *Let* $\mathcal{I}$ *be a set of implications. We denote by* $\mathcal{I}(\cdot)$ *the closure operator, sometimes called logical closure, that maps a set* X *to its smallest superset* Y *such that*

$$\forall A \to B \in \mathcal{I}, A \subseteq Y \Rightarrow B \subseteq Y$$

Example 2. *Let* $\mathcal{I} = \{a \to ab, bc \to bcd\}$ *be an implication set. We have* $\mathcal{I}(ad) = abd$ *and* $\mathcal{I}(ac) = abcd$*.*

Definition 8. *An implication set* $\mathcal{I}$ *is a basis for a closure operator* c *if* $\Phi_{\mathcal{I}} = \Phi_c$*.*

In FCA, an implication $A \to B$ is said to *hold* in a formal context when $A' \subseteq B'$.

Example 3. *The implications* $a \to ab$*,* $b \to b$ *and* $ad \to abcd$ *hold in the example context. The implication* $b \to b$ *holds trivially in any context.*

Multiple notions of bases have been studied in the literature [6] but the one we are interested in is the basis of minimum cardinality called the *Duquenne-Guigues basis* [13].

Definition 9. *A set* $P \in 2^E$ *is pseudo-closed if it is not closed and it contains the closure of all its proper subsets* ***that are pseudo-closed****.*

Definition 10. *The Duquenne-Guigues basis is the set*

$$\mathcal{B} = \{P \to c(P) \mid P \text{ is pseudo-closed}\}$$

Example 4. *The Duquenne-Guigues basis corresponding to the context shown in Table 1 is* $\{a \rightarrow ab, bc \rightarrow bcd, be \rightarrow bde, cd \rightarrow bcd, abd \rightarrow abcde, bcde \rightarrow abcde\}$.

Computing the Duquenne-Guigues basis is thus enumerating all the pseudo-closed sets for a particular closure operator. Having shown that the closed and pseudo-closed sets together form a closure system, Ganter proposed to use NextClosure [11], arguably the best-known and all-around most efficient algorithm. It enumerates pseudo-closed sets in the so-called *lectic order*, a total order that extends the order induced by the inclusion relation. As such, when the algorithm reaches a new pseudo-closed set, it has already computed the closure of all its subsets that are pseudo-closed and can, thus, recognize it. Other algorithms have been proposed such as variations on NextClosure [8, 18, 3] based on the lectic order, the one in [4] which can enumerate in any order that extends the inclusion or the attribute-incremental algorithm in [17]. All of these share the same property: they compute all the closed sets. The number of closed sets being potentially exponential in the number of pseudo-closed sets, none of these algorithms have a polynomial delay.

Some aspects of pseudo-closed sets have been studied in relation to their computation. We know that $|\mathcal{B}|$ can be exponential in $|\mathcal{A}|$ and $|\mathcal{R}|$ [16]. The problem of deciding whether a set is pseudo-closed given a context has been considered in [15, 1] and found to be coNP-complete. Special cases in the form of particular classes of lattices have been explored in [14, 20, 10]. In [9], the authors proved that it is impossible to enumerate pseudo-closed sets in the general case in the lectic order with a polynomial delay unless P = NP and went on to remove the restriction on the order but were unable to find a lower bound. Thus, the question of the complexity of enumerating pseudo-closed sets in non-lectic orders is still open. To the best of our knowledge, the problem of enumerating pseudo-closed sets in orders that do not extend the inclusion order has not yet been studied in the literature.

3. Computing a Pseudo-closed Set

3.1. Recognizing Pseudo-closed Sets

As mentioned in Section 2.2, given a formal context, the problem of recognizing a pseudo-closed set is coNP-complete. When enumerating in orders that extend the inclusion order, we are faced with the easier problem of recognizing pseudo-closed sets given a context, a set S and the knowledge of the closures of all the subsets of S encapsulated in the implication set $\mathcal{I} = \{A \rightarrow B \mid A \text{ is pseudo-closed and } A \subset S\}$. It is solved by computing $\mathcal{I}(S)$ and S''. However, if we want to enumerate in orders that do **not** extend the inclusion order, we cannot suppose we have all the information on the subsets and we thus have to be able to recognize a pseudo-closed set given only partial information, i.e. an implication set $\mathcal{I} \subseteq \{A \rightarrow B \mid A \text{ is pseudo-closed and } A \subset S\}$.

Looking back at the definition of a pseudo-closed set, we acknowledge that the minimal amount of information (about the closures of the subsets) needed to recognize a pseudo-closed S is one that ensures that S contains the closures of all its pseudo-closed proper subsets. Using implications as information, this minimal amount of information is an implication set of minimal cardinality for which we know with certainty that adding new implications will not change the logical closure of S. In other words, it is an implication set $\mathcal{I} \subseteq \mathcal{B}$ of minimal cardinality such that $\mathcal{I}(S) = S$ implies $\mathcal{L}(S) = S$ for any $\mathcal{I} \subseteq \mathcal{L} \subseteq \mathcal{B}$.

Proposition 2. *A set P is pseudo-closed if and only if it is not closed and there exists an implication set $\mathcal{I} \subseteq \mathcal{B}$ such that $P \in \Phi_{\mathcal{I}}$ and all its lower covers in $\Phi_{\mathcal{I}}$ are closed.*

Proof $\Rightarrow$. Let us suppose P is pseudo-closed and $\mathcal{I} = \mathcal{B} \setminus \{P \to c(P)\}$. From the definition of a pseudo-closed set, we have that P is not closed and $P \to c(P)$ does not follow from $\mathcal{I}$ so $P \in \Phi_{\mathcal{I}}$. We then have that all the lower covers of P in $\Phi_{\mathcal{I}}$ are closed (otherwise it would contain another pseudo-closed set that is not in $\mathcal{I}$).

$\Leftarrow$. Let us suppose P is not closed and $\mathcal{I} \subseteq \mathcal{B}$ is an implication set such that $P \in \Phi_{\mathcal{I}}$ and all the lower covers of P in $\Phi_{\mathcal{I}}$ are closed. From P belonging to $\Phi_{\mathcal{I}}$, we deduce that if $A \to c(A) \in \mathcal{I}$ and $A \subset P$, then $c(A) \subset P$. For any set $S \subset P$ in $\Phi_{\mathcal{I}}$, we have $S \subseteq C$ where C is a lower cover of P. If $S \neq c(S)$, we necessarily have $c(S) \subseteq C$ because C is closed. As every pseudo-closed set not in $\mathcal{I}$ is necessarily in $\Phi_{\mathcal{I}}$ (otherwise it would not contain the closure of one of its pseudo-closed proper subsets), P contains the closure of all its strict subsets that are pseudo-closed and is thus pseudo-closed. □

Definition 11. *A pseudo-closed set $P \in \Phi_{\mathcal{I}}$ for a closure operator c is said to be recognizable under $\mathcal{I}$ if all its lower covers in $\Phi_{\mathcal{I}}$ are closed.*

We use $Rec(\mathcal{I})$ to denote the set of pseudo-closed sets that are recognizable under $\mathcal{I}$.

Example 5. *In our running example, given $\mathcal{I} = \{cd \to bcd, be \to bde\}$ the lower covers of bc and bcde in $\Phi_{\mathcal{I}}$ are respectively $\{b, c\}$ and $\{bcd, bde, ce\}$. All of these sets being closed, both bc and bcde are recognizable pseudo-closed sets under $\mathcal{I}$. However, we have abd $\notin Rec(\mathcal{I})$ because one of its lower covers, ad, is not closed.*

Proposition 3. *$\forall \mathcal{I} \subset \mathcal{B}$, $Rec(\mathcal{I}) \neq \emptyset$.*

Proof Let P be minimal among pseudo-closed sets that are not premises of implications in $\mathcal{I}$. If there is a lower cover C of P in $\Phi_{\mathcal{I}}$ that is not closed, then there is a non-closed set $A \subseteq C$ in $\Phi_{\mathcal{I}}$ such that $A \to c(A)$ holds in the context.

If $c(A) \subset P$ then P is not minimal. If $c(A) \not\subseteq P$, then P is not pseudo-closed. Both cases lead to contradictions so all the lower covers of P are closed and P is recognizable from $\mathcal{I}$. Since we have $\mathcal{I} \subset \mathcal{B}$, the set of pseudo-closed sets that are not a premise in $\mathcal{I}$ is not empty, so it has minimal elements. Hence, $Rec(\mathcal{I}) \neq \emptyset$ if $\mathcal{I} \subset \mathcal{B}$. □

Viewing $\mathcal{I}$ as the set of implications "already found" at a given step during an enumeration, this property ensures we always have additional pseudo-closed sets to find if the basis is not completed.

As every element of $\Phi_{\mathcal{I}}$ is the intersection of meet-irreducibles, the number of lower covers of a set is bounded by $|\mathcal{M}(\Phi_{\mathcal{I}})|$. The intersection and the closure of sets can both be computed in time polynomial in the size of the context so deciding whether a set is a recognizable pseudo-closed set is polynomial in the size of $\mathcal{M}(\Phi_{\mathcal{I}})$.

3.2. Finding a Pseudo-closed Set

Now that we know how to recognize pseudo-closed sets, we want to be able to find one efficiently. We do not want to force an enumeration order more restrictive than the one imposed by the recognizability of a pseudo-closed set and randomly testing sets would be inefficient.

Proposition 4. *The lattice $\Phi_{\mathcal{I}}$ contains a recognizable pseudo-closed set P if and only if there is a maximal chain $C = X_1 \subset ... \subset X_n$ of non-closed sets that cannot be extended by proper subsets of X_1 with $X_1 = P$ and $X_n \in \mathcal{M}(\Phi_{\mathcal{I}})$.*

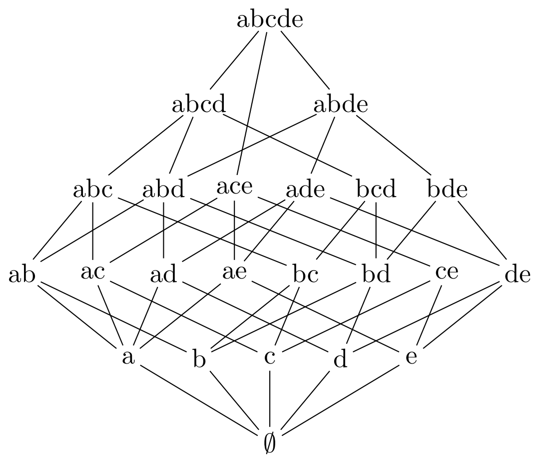
Proof $\Rightarrow$. Let $P \in \Phi_{\mathcal{I}}$ be a recognizable pseudo-closed set. Since P is in $\Phi_{\mathcal{I}}$, there must be a $\wedge$-irreducible S such that $P \subseteq S$ and $c(P) \not\subseteq S$. Obviously, S itself and any $T \in \Phi_{\mathcal{I}}$ such that $P \subseteq T \subseteq S$ are not closed. P being the minimal element of the chain follows from the definition of a recognizable pseudo-closed set.

$\Leftarrow$. Let us suppose the lattice contains a maximal chain of non-closed sets with a $\wedge$-irreducible as its maximal element and P as its minimal element. The set P being minimal in the maximal chain of non-closed sets, all its lower covers are closed. As such, it is a recognizable pseudo-closed set. □

Example 6. *Let $\mathcal{I} = \{cd \rightarrow bcd, be \rightarrow bde, bcde \rightarrow abcde\}$ be a subset of the Duquenne-Guigues basis of our running example. The lattice $\Phi_{\mathcal{I}}$, depicted in Figure 1, contains 8 meet-irreducibles $\mathcal{M}(\Phi_{\mathcal{I}}) = \{abcd, abde, abc, ace, ade, bcd, bde, ce\}$ and 2 recognizable pseudo-closed sets $Rec(\Phi_{\mathcal{I}}) = \{bc, a\}$. Both $abde > abd > ad > a$ and $abcd > abc > bc$ are examples of maximal non-closed chains between a meet-irreducible and a pseudo-closed set.*

We can thus find a pseudo-closed set by first finding a non-closed meet-irreducible X in $\Phi_{\mathcal{I}}$ and then computing the non-closed elements of the chain in a depth-first manner.

Figure 1: Lattice $\Phi_{\mathcal{I}}$ for $\mathcal{I} = \{cd \rightarrow bcd, be \rightarrow bde, bcde \rightarrow abcde\}$



Let us suppose that we know $\mathcal{M}(\Phi_{\mathcal{I}})$. As any set in the lattice is the infimum of a subset of $\mathcal{M}(\Phi_{\mathcal{I}})$, we can easily obtain the lower covers of any set $S \in \Phi_{\mathcal{I}}$ by computing the maximal elements of $\{S \cap X \mid X \in \mathcal{M}(\Phi_{\mathcal{I}}) \text{ and } S \not\subseteq X\}$. Algorithm 1 uses this to find a pseudo-closed set in $\Phi_{\mathcal{I}}$ given a non-closed $X \in \Phi_{\mathcal{I}}$.

Algorithm 1 *Descent*(S, M, c)

Require: A closure operator c on 2^E, a set of meet-irreducibles $M = \mathcal{M}(\Phi_{\mathcal{I}})$ and a non-closed set $S \in \Phi_{\mathcal{I}}$
Ensure: A recognizable pseudo-closed set $P \in \Phi_{\mathcal{I}}$
  $C = max(\{S \cap X \mid X \in \mathcal{M}(\Phi_{\mathcal{I}}) \text{ and } S \not\subseteq X)\})$
  if C contains a non-closed set P **then**
    Return *Descent*(P, M, c)
  else
    Return S
  end if

Proposition 5. *Algorithm 1 terminates and returns a recognizable pseudo-closed set.*

Proof There is a finite number of meet-irreducibles and elements so the algorithm performs a finite number of recursive calls. As per Propositions 2 and 4, the set returned is a recognizable pseudo-closed set. □

Computing the intersections of a set with meet-irreducibles is in $O(|E| \times |\mathcal{M}(\Phi_{\mathcal{I}})|)$ and the algorithm performs at most $|E|$ recursive calls. Isolating the maximal elements can be done in $O(|\mathcal{M}(\Phi_{\mathcal{I}})|^2)$. Thus, we can compute a new pseudo-closed set in time polynomial in $|\mathcal{M}(\Phi_{\mathcal{I}})|$ times the complexity of computing a closure.

The actual runtime of Algorithm 1 heavily depends on the order in which the meet-irreducibles are considered. If the first intersection always results in a non-closed lower cover, the pseudo-closed set is reached after $|E|$ closures whereas finding the non-closed set last can produce $|E| \times |\mathcal{M}(\Phi_{\mathcal{I}})|$ unnecessary closed sets.

4. Computing all Pseudo-closed Sets

From Section 3.2, we know how to find a recognizable pseudo-closed set given a context, an implication set $\mathcal{I}$ and $\mathcal{M}(\Phi_{\mathcal{I}})$. In practice (and in the case we are interested in), we begin with only a closure operator on 2^E and an empty set of implications. The lattice $\Phi_{\emptyset} = (2^E, \subseteq)$ has $|E|$ obvious meet-irreducibles: $\{E \backslash \{e\} \mid e \in E\}$ so applying Algorithm 1 to obtain a first pseudo-closed set only requires polynomial time. As a matter of fact, it corresponds to the algorithm proposed in [9] for computing the lectically first pseudo-closed set (without the order requirement). After the first implication is found and $\mathcal{I}$ is not empty anymore, we have to explicitly compute $\mathcal{M}(\Phi_{\mathcal{I}})$. Once this is done, we can apply Algorithm 1 to compute a new pseudo-closed set. Repeating this process will eventually yield the Duquenne-Guigues basis which, as per Proposition 4, has been computed once all the meet-irreducibles are closed.

Algorithm 2, described in pseudocode, computes the Duquenne-Guigues basis of a given closure operator.

Algorithm 2 $AllPC(E, c)$

Require: A closure operator c on 2^E
Ensure: The Duquenne-Guigues basis $\mathcal{B}$ corresponding to c
  $\mathcal{I} = \emptyset$
  $M = \{E \setminus \{e\} \mid e \in E\}$
  while $\exists S \in M$ such that $S \neq c(S)$ **do**
    $P = Descent(S, M, c)$
    $\mathcal{I} = \mathcal{I} \cup \{P \rightarrow c(P)\}$
    $M = \mathcal{M}(\Phi_{\mathcal{I}})$
  end while
  return $\mathcal{I}$

Proposition 6. *Algorithm 2 terminates and returns the Duquenne-Guigues basis of the input closure operator.*

Proof Propositions 5 and 2 ensure every iteration of the while loop adds a new implication $I \in \mathcal{B}$. Proposition 4 ensures the while loop stops when $\mathcal{I} = \mathcal{B}$. The number of implications in $\mathcal{B}$ being finite, the algorithm terminates and returns the Duquenne-Guigues basis of the context. □

We deliberately keep our options open regarding the actual implementation of the computation of the meet-irreducibles. The best general-case algorithm is the one proposed by Wild [19] which computes the meet-irreducibles of $\Phi_{\mathcal{I}}$ from those of $\Phi_{\mathcal{I}\setminus\{I\}}$, which makes it efficient for our purpose. However, there are other algorithms for special cases. In particular, Gély and Nourine [12] showed that we can compute $\mathcal{M}(\Phi_{\mathcal{I}})$ from $\mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}})$ in polynomial time when the premise of I is a singleton. More recently, Beaudou et al. [5] proposed an algorithm adapted to k-meet-semidistributive lattices.

Wild's algorithm computes the meet-irreducibles of $\Phi_{\mathcal{I}}$ from those of $\Phi_{\mathcal{I}\setminus\{I\}}$ by intersecting elements of $\mathcal{M}(\Phi_{\mathcal{I}}) \cap \mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}})$ (meet-irreducibles that stay) with elements of $\mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}}) \setminus (\mathcal{M}(\Phi_{\mathcal{I}}) \cap \mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}}))$ (meet-irreducibles removed by the new implication). Hence, it performs less than $|\mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}})|^2$ intersections. It then recognizes meet-irreducibles among the resulting sets. This can be done in polynomial time by checking whether a set is maximal among those that do not contain some element e. As such, both computing $\mathcal{M}(\Phi_{\mathcal{I}})$ and finding a new pseudo-closed set can be done in time polynomial in $|\mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}})|$. Unfortunately, $|\mathcal{M}(\Phi_{\mathcal{I}\setminus\{I\}})|$ can be exponential in $|\mathcal{I}|$.

5. Experimental Results

We implemented Algorithm 2 and used it to compute the Duquenne-Guigues bases of randomly generated closure operators. These were derivation operators corresponding to randomly generated formal contexts $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ such that (o, a) is in $\mathcal{R}$ with a probability $d = n/|\mathcal{A}|$. We call d the *density* of the context. The contexts contained 50 objects and a varying number of attributes. For each values of $|\mathcal{A}|$ and d, 1000 random contexts were generated. This section presents results on the number of closed sets computed and comparisons with NextClosure.

5.1. Number of Closures

For each closure operator, we compared the number of closed sets computed by Algorithm 2 to the total number of closed sets.

$$\sigma_{closed} = \frac{\text{number of closed sets computed}}{\text{number of unique closed sets}}$$

Figure 2 shows the values of σ_{closed} (sorted in increasing order) for the 11000 contexts with 12 attributes and a density varying from 1/12 to 11/12. We observe that Algorithm 2 computes less closed sets than the total number of closed sets in 42% of the contexts. The best and worst cases are respectively $\sigma_{closed} = 0.03$ and $\sigma_{closed} = 10.1$.

Figure 3 shows the average number of closed sets total and computed (left) and the corresponding average σ_{closed} (right) for each density d between 1/12 and 11/12.

We observe that Algorithm 2 is most efficient on contexts with a density above 7/12 while, in sparser contexts, it computes many times more closed sets than necessary. This is most likely due to the fact that denser contexts more often contain "clusters" of closed sets that are never computed, thus compensating for the multiple occurrences of the other ones.

5.2. Runtime

Even though the runtime depends heavily on the implementation, it can give insight into what needs to be improved. We compare the runtimes of Algorithm 2 and NextClosure through the ratio

$$\sigma_{runtime} = \frac{\text{runtime for Algorithm 2}}{\text{runtime for NextClosure}}$$

Figure 4 shows the average runtimes of NextClosure and Algorithm 2 together with the values of $\sigma_{runtime}$ for random contexts containing 13 attributes with densities between 1/13 and 12/13.

The variations of the average runtime unsurprisingly correlates with the numbers of closed sets enumerated (Figure 3). This means that Algorithm 2 becomes more efficient than NextClosure once it starts enumerating less closed sets, i.e. once the context is dense enough.

6. Discussion

The proposed algorithm allows for the computation of the Duquenne-Guigues basis in orders that do not extend the inclusion order. Moreover, it does not necessarily enumerate all the closed sets even though it can find some of them multiple times. While this is certainly a step towards more efficient algorithms for this problem, the version presented here suffers from high runtimes on contexts (closure operators) that are not dense enough. These runtimes are due to

- the difficulty of computing the meet-irreducibles elements of $\Phi_{\mathcal{I}\cup\{I\}}$ from those of $\Phi_{\mathcal{I}}$
- the fact that closed sets can be found multiple times when performing the depth-first searches along non-closed chains

As previously mentioned, computing the meet-irreducibles elements of $\Phi_{\mathcal{I}\cup\{I\}}$ from those of $\Phi_{\mathcal{I}}$ can be done in polynomial time when the premise of I is a singleton. It would be interesting to isolate other special cases as it could lead to

Figure 2: Sorted values of σ_{closed} for 11000 randomly generated contexts with 12 attributes and a density between 1/12 and 11/12 (1000 contexts for each density). The red line is $\sigma_{closed} = 1$.

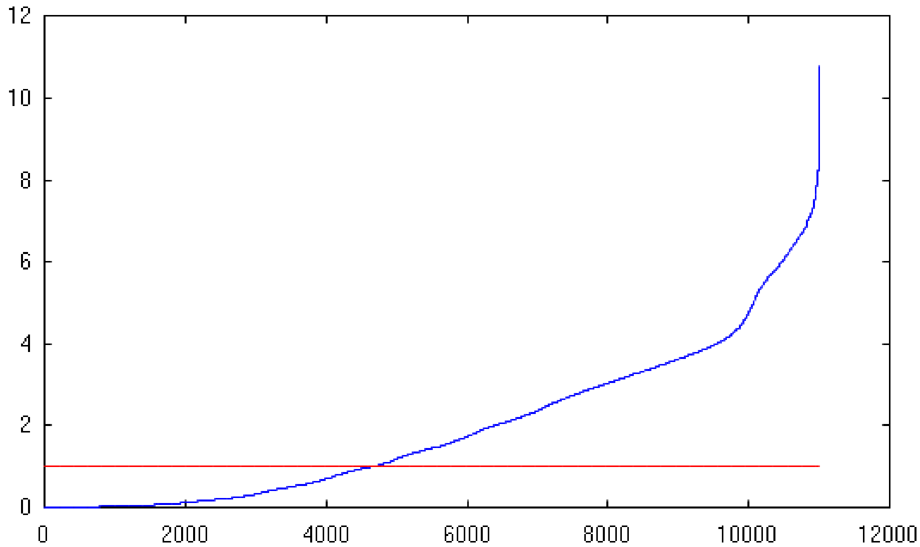


Figure 3: Left: average number of unique closed sets (blue) and computed closed sets (green) for each density. Right: average values of σ_{closed} for each density. The red line is $\sigma_{closed} = 1$.

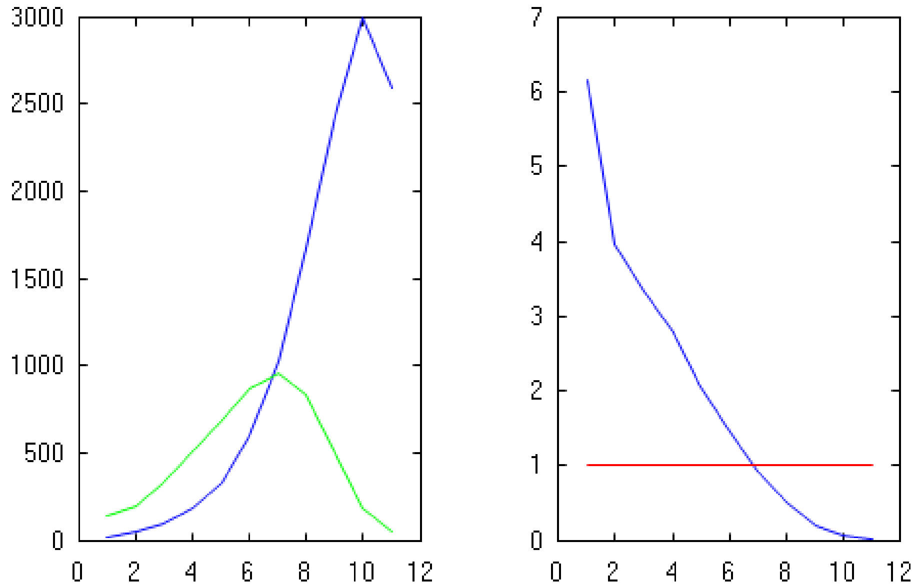
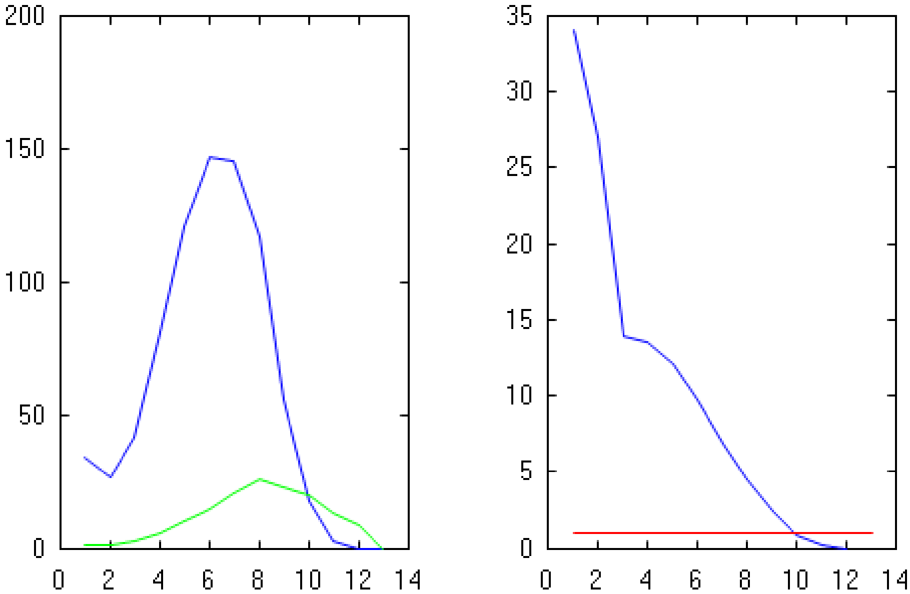


Figure 4: Left: average runtime in milliseconds of NextClosure (green) and Algorithm 2 (blue) for each density between 1/13 and 12/13. Right: values of $\sigma_{runtime}$ for each density



significant improvements on the runtime. Furthermore, studying the evolution of $\mathcal{M}(\Phi_{\mathcal{I}})$ when $\mathcal{I}$ grows would help us bound the delay as it is a function of both the number of meet-irreducibles and their structure. Of course, it would also be beneficial to investigate better general-case algorithms for this particular problem.

Concerning the computation of the non-closed chain, we could, potentially, always find a non-closed set first and thus never compute a single closed set with Algorithm 1 before reaching the pseudo-closed set. Achieving that in every case is probably impossible but heuristics could be used to find a non-closed set faster, which would considerably speed up the computation.

Acknowledgements

The author was supported by the European Union and the French Région Auvergne-Rhône-Alpes under the European Regional Development Fund (ERDF).

Bibliography

[1] Mikhail A. Babin and Sergei O. Kuznetsov. Recognizing Pseudo-intents is coNP-complete. In Marzena Kryszkiewicz and Sergei A. Obiedkov, editors, *Concept Lattices and Applications*, volume 672 of *CEUR Workshop Proceedings*, pages 294–301. CEUR-WS.org, 2010.

[2] Mikhail A. Babin and Sergei O. Kuznetsov. Computing Premises of a Minimal Cover of Functional Dependencies is Intractable. *Discrete Applied Mathematics*, 161(6):742–749, 2013.

[3] Konstantin Bazhanov and Sergei A. Obiedkov. Comparing Performance of Algorithms for Generating the Duquenne-Guigues Basis. In Amedeo Napoli and Vilém Vychodil, editors, *Concept Lattices and Applications*, volume 959 of *CEUR Workshop Proceedings*, pages 43–57. CEUR-WS.org, 2011.

[4] Alexandre Bazin and Jean-Gabriel Ganascia. Computing the Duquenne-Guigues Basis: An Algorithm for Choosing the Order. *International Journal of General Systems*, 45(2):57–85, 2016.

[5] Laurent Beaudou, Arnaud Mary, and Lhouari Nourine. Algorithms for k-meet-semidistributive lattices. *Theoretical Computer Science*, 658:391–398, 2017.

[6] Karell Bertet and Bernard Monjardet. The Multiple Facets of the Canonical Direct Unit Implicational Basis. *Theoretical Computer Science*, 411(22):2155–2166, 2010.

[7] Garrett Birkhoff. *Lattice Theory*, volume 25. American Mathematical Society, 1948.

[8] Daniel Borchmann. A Generalized Next-Closure Algorithm - Enumerating Semilattice Elements from a Generating Set. In Laszlo Szathmary and Uta Priss, editors, *Concept Lattices and Applications*, volume 972 of *CEUR Workshop Proceedings*, pages 9–20. CEUR-WS.org, 2012.

[9] Felix Distel and Barış Sertkaya. On the Complexity of Enumerating Pseudo-intents. *Discrete Applied Mathematics*, 159(6):450–466, 2011.

[10] Vincent Duquenne. The Core of Finite Lattices. *Discrete Mathematics*, 88(2):133–147, 1991.

[11] Bernhard Ganter and Klaus Reuter. Finding All Closed Sets: A General Approach. *Order*, 8(3):283–290, 1991.

[12] Alain Gély and Lhouari Nourine. About the Family of Closure Systems Preserving Non-unit Implications in the Guigues-Duquenne Base. In Rokia Missaoui and Jürg Schmidt, editors, *Formal Concept Analysis*, volume 3874 of *Lecture Notes in Computer Science*, pages 306–308. Springer, 2006.

[13] Jean-Louis Guigues and Vincent Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences humaines*, 95:5–18, 1986.

[14] Philippe Janssen and Lhouari Nourine. Minimum Implicational Basis for Meet-semidistributive Lattices. *Information Processing Letters*, 99(5):199–202, 2006.

[15] Sergei Kuznetsov and Sergei Obiedkov. Counting Pseudo-intents and #P-completeness. In Rokia Missaoui and Jürg Schmidt, editors, *Formal Concept Analysis*, volume 3874 of *Lecture Notes in Computer Science*, pages 306–308. Springer, 2006.

[16] Sergei O Kuznetsov. On the Intractability of Computing the Duquenne-Guigues Basis. *Journal of Universal Computer Science*, 10(8):927–933, 2004.

[17] Sergei Obiedkov and Vincent Duquenne. Attribute-incremental Construction of the Canonical Implication Basis. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):77–99, 2007.

[18] Gerd Stumme. Attribute Exploration with Background Implications and Exceptions. In *Data analysis and information systems*, pages 457–469. Springer, 1996.

[19] Marcel Wild. Computations with Finite Closure Systems and Implications. In *International Computing and Combinatorics Conference*, pages 111–120. Springer, 1995.

[20] Marcel Wild. Optimal Implicational Bases for Finite Modular Lattices. *Quaestiones Mathematicae*, 23(2):153–161, 2000.