Some Algorithmical Aspects Using the Canonical Direct Implicationnal Basis

K. Bertet

L3I - Université de La Rochelle - av Michel Crépeau - 17042 La Rochelle
karell.bertet@iniv-lr.fr

Abstract. Closure systems on a set S arises in many areas, in particular in formal concept analysis. Implicational systems represents an efficient and convenient tool to handle a closure system, and have been studied in various areas, with different terminology. This paper focuses on some algorithmical aspects of a particular unary implicationnal system called the *canonical direct basis* and proposes an incremental generation algorithm for this basis.

keywords: implicational system ; canonical direct basis ; lattice ; algorithm ; incremental generation ; closure operator ; closure system.

1 Introduction

Closure systems on a set S arise in many areas as formal concept analysis, lattice theory, relational databases, data-mining, artificial intelligence or logical programming, where we need efficient algorithms to handle them. Implicational systems represents an efficient and convenient tool to handle a closure system, and have been studied in various areas, with different terminology: they are denoted rules and proper implications in artificial intelligence [15], functional dependencies in databases [9], and Horn functions in logical programming ([1,5]). One can find many others representations of a closure system: representation by a table called a *context* in formal concept analysis ([7]) and data-mining ; representation by the canonical basis in data analysis ([8]) ; representation by a poset of irreducibles in lattice theory ([11]).

An unary implication system (UIS) Σ on a finite set S is a set of rules called Σ-implications of the kind $B \to x$, with $B \subseteq S$ and $x \in S$. A subset $X \subseteq S$ satisfies $B \to x$ when "$B \subseteq X$ implies $x \in X$" holds, so UIS can be used to describe constraints on sets of elements, such as dependency or causality between attributes. This paper focuses on algorithmical aspects of the representation of a closure system by a particular UIS called the *canonical direct basis*.

Section 2 recall all the definitions (definitions of a closure system, an UIS, a concept lattice). Section 3 defines the canonical direct basis, and its use to generate some closures of the associated lattice. All algorithms to generate the canonical direct basis have an exponential time complexity. In Section 4 we propose a new generation algorithm, based on an incremental addition of new implication $B \rightarrow x$ is a canonical direct basis Σ'_{cd}, to limit the exponential cost in $O(|S||\Sigma_{cd}|^{|B|+1})$.

2 Recalls and Definitions

In this paper, a subset $X = \{x_1, x_2, \ldots, x_n\}$ is written as the word $X = x_1x_2\ldots x_n$. A subset $x_ix_{i+1}\ldots x_j \subseteq X$, with $1 \leq i \leq j \leq n$, is written as the subword $X[i,j]$. Moreover, we abuse notation and use $X + x$ (respectively, $X \setminus x$) for $X \cup \{x\}$ (respectively, $X\backslash\{x\}$), with $X \subseteq S$ and $x \in S$.

Closure system. A *set system* on a set S is a family of subsets of S. A *closure system* $\mathbb{F}$ on a set S, also called a *Moore family*, is a set system stable by intersection and which contains S: $S \in \mathbb{F}$ and $F_1, F_2 \in \mathbb{F}$ implies $F_1 \cap F_2 \in \mathbb{F}$. The subsets belonging to a closure system $\mathbb{F}$ are called the *closed sets* of $\mathbb{F}$.

A *partially ordered set* $P = (S, \leq)$, also called a *poset*, is a set S equipped with an order relation $\leq$ where an order relation is a binary relation which is reflexive ($\forall x \in S, x \leq x$), antisymmetric ($\forall x \neq y \in S$, $x \leq y$ imply $y \not\leq x$) and transitive ($\forall x, y, z \in S$, $x \leq y$ and $y \leq z$ imply $x \leq z$). A poset $L = (S, \leq)$ is a *lattice* if any pair $\{x, y\}$ of elements of L has a *join* (i.e. a least upper bound) denoted by $x \vee y$ and a *meet* (i.e. a greatest lower bound) denoted by $x \wedge y$. Therefore, a lattice contains a minimum (resp. maximum) element according to the relation $\leq$ called the *bottom* (resp. *top*) of the lattice, and denoted $\bot_L$ or simply $\bot$ (resp. $\top_L$ or simply $\top$.)

The poset $(\mathbb{F}, \subseteq)$ is a lattice with, for each $F_1, F_2 \in \mathbb{F}$, $F_1 \wedge F_2 = F_1 \cap F_2$ and $F_1 \vee F_2 = \bigcap\{F \in \mathbb{F} \,|\, F_1 \cup F_2 \subseteq F\}$. Moreover, any lattice L is isomorphic to the lattice of closed sets of a closure system (see [3] for more details).

A *closure operator* on a set S is a map φ on $\mathcal{P}(S)$ satisfying the three following properties: φ is *isotone* (i.e. $\forall X, X' \subseteq S$, $X \subseteq X' \Rightarrow \varphi(X) \subseteq \varphi(X')$), *extensive* (i.e. $\forall X \subseteq S$, $X \subseteq \varphi(X)$) and *idempotent* (i.e. $\forall X \subseteq S$, $\varphi^2(X) = \varphi(X)$). Closure operators are in one-to-one correspondance with closure systems. On the first hand, the set of all closed elements of φ forms a closure system $\mathbb{F}_\varphi$:

$$\mathbb{F}_\varphi = \{F \subseteq S \,|\, F = \varphi(F)\} \tag{1}$$

Dually, given a closure system $\mathbb{F}$ on a set S, one defines the closure $\varphi_{\mathbb{F}}(X)$ of a subset X of S as the least element $F \in \mathbb{F}$ that contains X:

$$\varphi_{\mathbb{F}}(X) = \bigcap\{F \in \mathbb{F} \,|\, X \subseteq F\} \tag{2}$$

In particular $\varphi_{\mathbb{F}}(\emptyset) = \bot_{\mathbb{F}}$. Moreover for all $F_1, F_2 \in \mathbb{F}$, $F_1 \vee F_2 = \varphi_{\mathbb{F}}(F_1 \cup F_2)$ and $F_1 \wedge F_2 = \varphi_{\mathbb{F}}(F_1 \cap F_2) = F_1 \cap F_2$.

See the survey of Caspard and Monjardet [4] for more details about closure systems.

Implicationnal System. An *Unary Implicational System* (UIS for short) Σ on S is a binary relation between $\mathcal{P}(S)$ and S: $\Sigma \subseteq \mathcal{P}(S) \times S$. An ordered pair $(A, b) \in \Sigma$ is called a Σ*-implication* whose *premise* is A and *conclusion* is b. It is written $A \to b$, meaning "A implies b". A subset $X \subseteq S$ *respects* a Σ-implication $A \to b$ when $A \subseteq X$ implies $b \in X$ (i.e. "if X contains A then X contains b"). $X \subseteq S$ is Σ*-closed* when X respects all Σ-implications, i.e $A \subseteq X$ implies $b \in X$ for every Σ-implication $A \to b$. The set of all Σ-closed sets forms a closure system $\mathbb{F}_\Sigma$ on S:

$$\mathbb{F}_\Sigma = \{X \subseteq S \mid X \text{ is } \Sigma\text{-closed}\} \tag{3}$$

Then, as introduced in [16] and [17], we can associate to Σ a closure operator $\varphi_\Sigma = \varphi_{\mathbb{F}_\Sigma}$ which defines the closure of a subset $X \subseteq S$

$$\varphi_\Sigma(X) = \pi_\Sigma(X) \cup \pi^2_\Sigma(X) \cup \pi^3_\Sigma(X) \cup \ldots \text{ where} \tag{4}$$

$$\pi_\Sigma(X) = X \cup \bigcup \{b \mid A \subseteq X \text{ and } A \to b\} \tag{5}$$

Remark that S being finite, the procedure in (4) terminates. Moreover, $\varphi_\Sigma(X) = \pi^n_\Sigma(X)$ with $n \leq |S|$ being the first integer such that $\pi^n_\Sigma(X) = \pi^{n+1}_\Sigma(X)$.

An UIS Σ is a *generating system* of the closure system $\mathbb{F}_\Sigma$ (using Eq. (1)), and thus for the induced closure operator φ, and the induced lattice $(\mathbb{F}_\Sigma, \subseteq)$. When some UISs Σ and Σ' on S are generating systems for the same closure system, they are called *equivalent* (i.e. $\mathbb{F}_\Sigma = \mathbb{F}_{\Sigma'}$).

An UIS Σ is called *direct* or *iteration-free* if for every $X \subseteq S$, $\varphi(X) = \pi_\Sigma(X)$ (see Eq. (5)). An UIS Σ is *minimal* or *non-redundant* if $\Sigma \setminus \{X \to y\}$ is not equivalent to Σ, for all $X \to y$ in Σ. It is *minimum* if it is of least cardinality, i.e. if $|\Sigma| \leq |\Sigma'|$ for all UIS Σ' equivalent to Σ. A minimum UIS is trivially non-redundant, but the converse is false. Σ is *optimal* if $s(\Sigma) \leq s(\Sigma')$ for all UIS Σ' equivalent to Σ, where the *size* $s(\Sigma)$ of Σ is defined by: $s(\Sigma) = \sum_{A \to b \in \Sigma}(|A|+1)$ The *direct-optimal* property combines the directness and optimality properties: an UIS Σ is *direct-optimal* if it is direct, and if $s(\Sigma) \leq s(\Sigma')$ for any direct UIS Σ' equivalent to Σ. A minimal UIS is usually called a *basis* for the induced closure system (and thus for the induced lattice), and a *minimum basis* is then a basis of least cardinality. An implication $X \to x$ with $x \in X$ is called *trivial.* An UIS is called *proper* if it doesn't contains trivial implications. When an UIS is not proper, an equivalent proper UIS can be obtained by applying the following treatment T1. In this paper, all UISs will be considered to be proper UISs.

T1 *delete* $A \to b$ *from* Σ *when* $b \in A$.

In the litterature, an implicational system (IS for short) Σ can also be defined as a binary relation on $\mathcal{P}(S)$. A Σ-implication is then an ordered pair $(A, B) \in \Sigma$, written $A \to B$, with $A, B \in \mathcal{P}(S)$. An equivalent unary IS can be obtained by applying the following treatment:

T2 *replace* $A \to B = \{b_1, b_2, \ldots, b_n\}$ *by* $A \to b_1$*,* $A \to b_2$*,* $\ldots$ *and* $A \to b_n$.

Dually, an equivalent IS can be obtained from an unary IS by applying recursively the following treatment:

T3 *replace* $A \to B$ *and* $A \to B'$ *by* $A \to B \cup B'$.

Generating systems (also called *covers*) and bases can be also defined for IS. In this case, there exists an unique minimum basis, called the *canonical basis*, also denoted the *Guigues and Duquenne basis* ([8]), enabling to get all the others minimum basis. The *canonical direct basis*, aim of this paper, is the unique direct-optimal basis. We denote Σ_{can} the UIS deduced from the canonical basis by applying T2, and Σ_{cd} the UIS deduced from the canonical direct basis by applying T2. Other definitions and bibliographical remarks can be found in the survey of Caspard and Monjardet in [4].

Concept lattice and closure system. A *formal context* $K = (G, M, I)$ consists of two sets G and M, and a relation I between G and M. The elements of G are called the *objects*, and the elements of M are called the *attributes* of the context. We define the application f that associates to every element $o \in G$ the set $f(o) = \{a \in M \mid (o, a) \in I\}$, and the application g that associates to every element $a \in M$ the set $g(a) = \{o \in G \mid (o, a) \in I\}$. The extension of f to subsets $A \subseteq G$ provides:

$$f(A) = \cap_{o \in A} f(o) = \{a \in M \mid (o, a) \in I \ \forall\ o \in A\} \tag{6}$$

Dually, the extension of g to subsets $B \subseteq M$ provides:

$$g(B) = \cap_{a \in G} g(a) = \{a \in G \mid (o, a) \in I \ \forall\ a \in B\} \tag{7}$$

The two applications f and g forms a *Galois correspondance* between G and M.

A *formal concept* of a formal context K is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $f(A) = B$ et $g(B) = A$. Let $\mathbb{C}$ be the set of all the concepts of K, and $\leq_C$ be the following relation on $\mathbb{C}$, with (A_1, B_1) and (A_2, B_2) two concepts:

$$(A_1, B_1) \leq (A_2, B_2) \text{ iff } A_1 \subseteq A_2 \text{ (or equivalently } B_2 \subseteq B_1) \tag{8}$$

The set $\mathbb{C}$ equipped with the relation $\leq$ is a lattice called the *concept lattice*[1] Let $\mathbb{C}_G$ be the restriction of $\mathbb{C}$ to the objects where each concept (A, B) is replaced by the subset A of objects. Dually, let $\mathbb{C}_M$ be the restriction of $\mathbb{C}$ to the attributes. Then $\mathbb{C}_G$ is a closure system on the set G, with $h = f \circ g$ the associated closure operator. Dually, $\mathbb{C}_M$ is a closure system on the set M of attributes, with $h' = g \circ f$ the associated closure operator. Moreover, the three lattices $(\mathbb{C}_G, \subseteq)$, $(\mathbb{C}_M, \supseteq)$ and $(\mathbb{C}, \leq)$ are isomorphic.

In *formal concept analysis* concept lattices are used to analyse datas when organised by a binary relation between object and attributes. See the complete book of Ganter and Wille [7] for a complete description of formal concept analysis.

[1] This lattice is also denoted *Galois lattice* by reference to the Galois correspondance (f, g) of the formal context C.

3 The canonical direct implicationnal basis

3.1 Description

The canonical direct implicationnal basis Σ_{cd} has been introduced with different terminologies and definitions, satisfying various properties. In [2], the *direct-optimal basis* is introduced with a generation way (i.e. a generation algorithm from any equivalent UIS), and stated to verify the direct-optimal and the unicity properties. Wild in [17] introduces the *canonical iteration-free basis* defined by a caracterization of Σ-implication's premisses (that have to be *free subsets*). It also states the unicity and the directness properties. The *left-minimal basis* is a restriction of any direct UIS to implications where the premisse is of minimal cardinality. Such implications are used in data-mining area research where they are denoted *proper implications* ([15]), and in relationnal databases where they are denoted *functionnal dependencies* ([9]). Rush and Wille in [14] introduce the *weak-implication basis*, based on the definition of a minimal transversal of a subset, to establish a connection with the formal concept analysis.

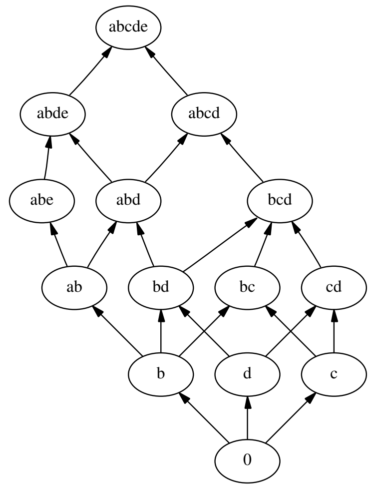
In a recent work [1], Bertet and Monjardet state the equality between these four basis[2]. The three main properties are the directness, canonical and minimality properties, thus the name *canonical direct implicational basis.* Moreover, a simple generation algorithm from any equivalent UIS is provided in [2].

Consider as example the closure system on the set $S = \{a, b, c, d, e\}$:

$$\mathbb{F} = \{\emptyset, b, c, d, ab, bd, bc, cd, bcd, abe, abd, abde, abcd, S\}$$

One can verify that $\mathbb{F}$ is stable by intersection. The lattice $(\mathbb{F}, \subseteq)$ is represented by its Hasse diagram and the canonical direct basis Σ_{cd} is:

$$\Sigma_{cd} = \begin{cases} (\ 1\ )\ a \to b \quad (\ 2\ )\ ac \to d \\ (\ 3\ )\ e \to a \quad (\ 4\ )\ e \to b \\ (\ 5\ )\ ce \to d \end{cases}$$



Remark that Σ_{cd} is proper UIS since for every implication conclusion is not included in premisse. Moreover, Σ_{cd} is a direct UIS [3]

[2] The equality is also stated with another basis, denoted the *dependance's relation basis*, defined from a particular relation between some elements of a lattice, called the *dependance relation* of a lattice.

[3] Consider the φ-closure of e: $\pi(e) = e + a + b$ by applying Σ_{cd}-implication (3) and (4) and $\pi(e) = \varphi(e)$. Consider the equivalent but non direct UIS Σ defined by deletion of the Σ_{cd}-implication (4). In this case, $\pi(e) = e + a$ by applying Σ-implication (3) and $\pi^2(e) = (e + a) + b$ by applying Σ-implication (1). Therefore $\varphi(e) \neq \pi(e)$, thus Σ is not direct.

3.2 Algorithmical aspects of the canonical direct basis

Computation of a closure $\varphi(X)$ ([2]). A number of problems related to closure systems, thus (concept) lattices or closure operators, can be answered by computing closures of the kind $\varphi_\Sigma(X)$, for some $X \subseteq S$. According to the definition (see Eq.(4)) $\varphi(X)$ can be obtained given an UIS Σ by iteratively scanning Σ-implications: $\varphi(X)$ is initialized with X then increased with b for each implication $A \rightarrow b$ such that $\varphi(X)$ contains A. The computation cost depends on the number of iterations, and in any case is bounded by $|S|$. It is worth noticing that for direct (or iteration-free) UISs the computation of $\varphi(X)$ requires only one iteration, since $\varphi_\Sigma(X) = \pi_\Sigma(X)$.

In [10], Mannila and Räihä propose the generation of a closure $\varphi(X)$ (algorithm *Linclosure*) in $O(|S|^2|\Sigma|)$, with a given Σ as input. This algorithm iteratively scans implications of an UIS Σ. In order to practically limit this number while keeping the same complexity, Wild in [17] modifies this algorithm using additional and sophisticated data structures. Another improvement consists in considering an UIS of minimal-size, as the canonical basis Σ_{can}.

Using the direct property (i.e. only one iteration on the implications is required) of the canonical direct basis Σ_{cd} (i.e. the direct UIS of minimal size), Bertet and Nebut in [2] propose the generation of a closure $\varphi(X)$ in $O(|X||\Sigma_{cd}|)$ when expressed with respect to X or in $O(s(\Sigma_{cd}))$ when expressed with respect to Σ_{cd}. Therefore, computation of a closure $\varphi(X)$ can be performed, with $|\Sigma_{cd}| > |\Sigma_{can}|$, in $O(|S|^2|\Sigma_{can}|)$ using the canonical basis Σ_{can}, or in $O(|S||\Sigma_{cd}|)$ or in $O(s(\Sigma_{cd}))$ using the canonical direct basis Σ_{cd}.

In the cases where few closures are needed, or where a small canonical basis is considered, it may be more efficient to iterate over Σ_{can}-implications. On the contrary, when lot closures are needed (for example when the whole family $\mathbb{F}$ has to be generated), or where a small canonical direct basis is considered, the second algorithm using the canonical direct basis Σ_{cd} is more efficient. Let us also notice that Σ_{can} gives a more concise description of the family than Σ_{cd}.

Generation of the family $\mathbb{F}$ or the set of concepts $\mathbb{C}$ ([1]). A particular problem that can be answered by computing some closures $\varphi(X)$ is the generation of the complete family $\mathbb{F}$, equivalent to the set of concepts $\mathbb{C}$. Any generation algorithm of $\mathbb{F}$ or $\mathbb{C}$ has to be analysed by considering the time-complexity per closure $\varphi(X)$ since $\mathbb{F}$ and $\mathbb{C}$ are exponential is the worst case.

The well-known algorithm generating $\mathbb{F}$ or $\mathbb{C}$ is the *Next-closure* algorithm due to Ganter ([6]) in the context of the formal concept analysis ([7]). It accepts a formal context (and more generally a closure operator) as input, and has a polynomial space-complexity (since the closed sets have not to be stored) and a time complexity in $O(|S|^3)$ per element. One can find various algorithms generating $\mathbb{F}$ or $\mathbb{C}$ using different inputs, with the same complexity as the Next-closure algorithm, i.e. in $O(|S|^3)$ per generated closed set. However, the algorithm with the best known complexity, using a poset or a formal context as input, is due to Nourine and Raynaud in [12]. It has a time-complexity in $O(|S|^2)$ per generated closed set, and an exponential space-complexity since all closed sets have to be stored in a tree structure.

Let us mention the Bertet and Nebut algorithm in [2] that generates $\mathbb{F}$ or $\mathbb{C}$ by computing some closures $\varphi(X)$. This algorithm has an exponential space-complexity since the closed sets have to be stored, and a time complexity in $O(|S|^2 + |S|c_\varphi)$ per element, where c_φ is the cost to generate one closure $\varphi(X)$. Therefore, their algorithm is in $O(|S|^3)$ using the canonical basis Σ_{can} as input, and in $O(|S|^2 + |S|s(\Sigma_{cd}))$ using the canonical direct basis Σ_{cd} as input.

4 Generation of the canonical direct basis

4.1 One-pass generation

Since Σ_{cd} is bounded by $2^{|S|}$ in the worst case, and by 1 in the best case, with a reasonable size in practice, any generation algorithm has to be analysed by considering the time-complexity per implication. Currently, there exists no algorithm with a polynomial generation per implication. Moreover, the existence of such a polynomial algorithm is still an open problem. Wild in [17] provides an algorithm with an IS Σ as input that has an exponential time complexity per implication. His algorithm computes an intermediate but larger UIS of exponential size in the worst case. Bertet and Nebut's algorithm, described by Proposition 1 ([2]) also generates an intermediate and exponential but direct UIS Σ_d (recursive treatment T4) before minimizing it (treatment T5), and computes Σ_{cd} in $O(|S||\Sigma_d|^2)$. Let us also mention in the area of data-mining the algorithm of Taouil and Bastide in [15] where the implications are called *proper implications*. It has the same exponential time and space complexity per implication. The algorithm of Mannila in [10], with the irreducibles elements of $\mathbb{F}$ as input, has an exponential time per implication in the worst case, and is based on the generation of all minimal transversals, open problem (no known polynomial algorithm, and no reduction to an NP-complete problem). Concerning the canonical basis Σ_{can}, let us mention the attribute-incremental algorithm of Duquenne generating Σ_{can} from a context ([13]).

Proposition 1. *[2] The* canonical direct basis Σ_{cd} *is obtained from any equivalent and proper UIS Σ as follows:*

1. *first apply recursively the following* make-direct treatment*to obtain a direct equivalent UIS* [4] *:*

   T4 *for all $A \to b$ and $C + b \to d$ with $d \neq b$ and $d \notin A$, add $A \cup C \to d$ to Σ*
2. *then apply the following* reduction treatment *to minimize premisses of the Σ-implications:*

   T5 *for all $A \to b$ and $C \to b$, if $C \subset A$ then delete $A \to b$ from Σ.*

[4] When Σ is not proper, this treatment has to be apply only when $b \notin A$ and $d \notin A \cup C$

Name: `CanonicalDirectBasis`
Input: An UIS $\Sigma = \{B_i \to x_i \,:\, i \in [1, m]\}$
Output: The canonical direct basis Σ_{cd}
begin
  $\Sigma tmp = \{B_1 \to x_1\}$
  for *i from* 2 *to* m **do**
    $\Sigma tmp =$`AddDirect`$(\Sigma tmp, B_i \to x_i)$
  return Σtmp
end

Algorithm 1: Incremental Generation of the Canonical Direct Basis

4.2 Incremental generation

Time complexity of the generation of the canonical direct basis Σ_{cd} from any equivalent UIS Σ can be improved by reducing the size of the intermediate, larger and direct UIS Σ_d that has to be generated by Bertet and Nebut's algorithm as described in Proposition 1 (also by Wild's algorithm in[17]). This intermediary UIS, larger and direct, is generated by the make-direct treatment T4, before being minimized by the reduction treatment T5 in order to obtain Σ_{cd}.

In the algorithm we propose, the principle is to alternate between the make-direct treatment T4 and the reduction treatment T5 in an incremental way in order to limit the size of the intermediate direct UIS generated. More formally, let $\Sigma = \{B_i \to x_i \,:\, i \in [1, n]\}$ the input UIS. Then Σ_{cd} can be obtained by successively compute $\Sigma_i = (\Sigma_{i-1} \cup \{B_i \to x_i\})_{cd}$ for $i \leq n$. Thus $\Sigma_1 = \{B_1 \to x_1\}$ and $\Sigma_n = \Sigma_{cd}$.

The basic step of this incremental algorithm then consists in computing $(\Sigma'_{cd} \cup \{B \to x\})_{cd}$, with Σ'_{cd} a canonical direct basis (i.e. satisfying the three properties that are directness, canonical and minimality properties), and $B \to x$ a new implication that can obviously be supposed to be proper (i.e. $x \notin B$) and Σ'_{cd}-minimal, with the minimality defined as follows:

Definition 2 (Σ-minimal implication). Let Σ an UIS. An implication $A \to b$ is said *Σ-minimal* if $A \not\subset B$ and $B \not\subset A$ for every Σ-implication $B \to b$.

Therefore, instead of applying the make-direct treatment to every pair of rules in $(\Sigma'_{cd} \cup \{B \to x\})^2$ in a recursive way as stated by Proposition 1, it can be reduced to the only pairs including $B \to x$ since Σ'_{cd} is already direct-optimal. Theorem 4 gives a more precise characterization of the pairs of $(\Sigma'_{cd} \cup \{B \to x\})^2$ for which the make-direct treatment has to be applied. It is based on the $\otimes_{\mathbb{D}}$-operator to represent the make-direct treatment as follows:

Definition 3 (The $\otimes_{\mathbb{D}}$ operator). The $\otimes_{\mathbb{D}}$ operator is a binary operator[5] defined on an UIS Σ, with $A \to b$ and $C \to d$ be two Σ-implications, by:

$$(A \to b \otimes_{\mathbb{D}} C \to d) = A \cup C \backslash b \to d \text{ when } b \in C$$
$$= \emptyset \to \emptyset \quad \text{ when } b \notin C$$

Theorem 4. *Let Σ'_{cd} be a canonical direct basis ; $B \to x$ be a proper and Σ'_{cd}-minimal implication and $P \to c$ be an added implication in $(\Sigma'_{cd} \cup B \to x)_{cd} \backslash \Sigma'_{cd}$. Then $P \to c$ verifies one of the two following cases:*

1. *$c = x$ and $P \to x$ is obtained by application of the make-direct treatment T4 on the set $\mathbb{K}$ of Σ'_{cd}-implications (in (9)) as follows:*

$$P \to x = (C_n \to x_n \otimes_{\mathbb{D}} (\ldots \otimes_{\mathbb{D}} (C_2 \to x_2 \otimes_{\mathbb{D}} (C_1 \to x_1 \otimes_{\mathbb{D}} B \to x))))$$
$$= (B \backslash (x_1 \ldots x_n) \cup \bigcup_{i \leq n} C_i) \to x$$

$$\mathbb{K} = \{C_1 \to x_1, \ldots, C_n \to x_n \ : \ x \notin C_i \ , \ x_i \in B \backslash (x_1 \ldots x_{i-1}), \forall i \leq n\} \quad (9)$$

2. *$c \neq x$ and $P \to c$ is obtained by application of the make-direct treatment T4 on the set of Σ'_{cd}-implications in (10) as follows:*

$$P \to c = (C_n \to x_n \otimes_{\mathbb{D}} (\ldots \otimes_{\mathbb{D}} (C_1 \to x_1 \otimes_{\mathbb{D}} (B \to x \otimes_{\mathbb{D}} A \to c))))$$
$$= (B \backslash (x_1 \ldots x_n) \cup A \backslash (x x_1 \ldots x_n) \cup \bigcup_{i \leq n} C_i) \to c$$

$$\mathbb{K} \bigcup \{A \to c \ : \ c \notin B \text{ and } x \in A\} \quad (10)$$

Proof. Then $P \to c$ is issued from a sequence of make-direct treatments T4, and can formally be described by an expression composed of some initial Σ'_{cd}-implications and the new implication $B \to x$ together with the binary $\otimes_{\mathbb{D}}$-operator. However, every such expression of $\otimes_{\mathbb{D}}$-operators doesn't give raise to a new implication since it can be deleted by the reduction treatment T5, and thus not being minimal. Some expressions have not to be considered to obtained minimal implications in Σ_{cd}, thus a limitation to the two cases stated by Theorem 4 obviously issued from the four following remarks:

1. Without loss of generalities one can reduce the expressions that have to be considered to those containing $B \to x$: every expression that not contains $B \to x$ either is already a Σ'_{cd}-implication, or is not minimal in Σ'_{cd} since Σ'_{cd} is a canonical direct basis, thus issued from the T4 and T5 treatments.
2. With the same kind of argument, each sub-expression that not contains $B \to x$ can equivalently be replaced by one Σ'_{cd}-implication using Proposition 5 since Σ'_{cd} verifies the directness property.

[5] This $\otimes_{\mathbb{D}}$ operator corresponds to the *accumulative* operator or the *pseudo-transitive* operator in databases[9]

3. Lemma 6 (case 3) states that implications on the right hand of $B \to x$ can be reduced to only one implication. Moreover, this implication has to be treated in first as stated by Lemma 6 (case 4).
4. Finally, implications that have to be treated at the left hand of $B \to x$ are restricted according to their conclusions as described by Lemma 6 (case 1).

Proposition 5. *Let Σ be a direct UIS. Let $P \to c$ and $P' + c \to c'$ be two Σ-implications. Then there exist a Σ-implication $P'' \to c'$ such that $P'' \subseteq P \cup P'$*

Proof. Since Σ is direct, the canonical direct basis Σ_{cd} is included in Σ by the minimality. Let us prove that $P'' \to c'$ such that $P'' \subseteq P \cup P'$ is a Σ'_{cd}-implication. The make-direct treatment T4 consists in the application of the $\otimes_{\mathbb{D}}$ operator as follows:

$$(P \to c \otimes_{\mathbb{D}} P' + c \to c') = P \cup P' \to c' \tag{11}$$

Then, if the reduction treatment T5 implies the deletion $P \cup P' \to c'$, this means that there exists a Σ'_{cd}-implication $P'' \to c'$ such that $P'' \subseteq P \cup P'$.

Lemma 6. case 1: *The following implications is not Σ_{cd}-minimal when for all $k \leq m$, $x_k \notin B \backslash (C_1 + \ldots + C_{k-1} + x_1 \ldots x_{k-1})$:*

$$(C_m \to x_m \otimes_{\mathbb{D}} (\ldots \otimes_{\mathbb{D}} (C_2 \to x_2 \otimes_{\mathbb{D}} (C_1 \to x_1 \otimes_{\mathbb{D}} B \to x)))) \tag{12}$$

case 2: *The following implications is not Σ_{cd}-minimal*

$$(C_m \to x_m \otimes_{\mathbb{D}} (\ldots \otimes_{\mathbb{D}} (C_1 \to x_1 \otimes_{\mathbb{D}} B \to x))) \otimes_{\mathbb{D}} A \to y \tag{13}$$

case 3: *The following implication is not Σ_{cd}-minimal:*

$$(B \to x \otimes_{\mathbb{D}} A_1 \to y_1) \otimes_{\mathbb{D}} A_2 \to y_2) \ldots \otimes_{\mathbb{D}} A_m \to y_m) \tag{14}$$

Proof. **case 1:** Let $P \to x$ be the implication issued from Eq 12. Suppose the condition $x_k \notin B \backslash (C_1 + \ldots + C_{k-1} + x_1 \ldots x_{k-1})$ is not verified for some $k \leq m$, and let us prove that $P \to x$ is then not minimal in Σ_{cd}. Using Definition 3, P is equal to:

$$P = B \backslash (x_1 \ldots x_m) \cup \bigcup_{i \leq n} C_i \backslash (x_{i+1} \ldots x_m) \tag{15}$$

Let $k \leq m$ be the first integer such that $x_k \notin B \backslash (C_1 + \ldots + C_{k-1} + x_1 \ldots x_{k-1})$. This imply that $x_i \in B \backslash (C_1 + \ldots + C_i + x_1 \ldots x_i)$ then $x_i \notin C_{i'}$ for every $i' < i < k$. Thus a refinement of the (C_i)'s when $i < k$:

$$P = B \backslash (x_1 \ldots x_m) \cup \bigcup_{i<k} C_i \backslash (x_k \ldots x_m) \cup \bigcup_{i \geq k} C_i \backslash (x_{i+1} \ldots x_m) \tag{16}$$

Since the $\otimes_{\mathbb{D}}$-operator has been applied to $C_k \to x_k$, we deduce from $x_k \notin B \backslash (C_1 + \ldots + C_{k-1} + x_1 \ldots x_{k-1})$ that there exists $k' < k$ such that $x_k \in C_{k'}$. Proposition 5 applied to the two Σ'_{cd}-implications $C_{k'} \to x_{k'}$ and $C_k \to x_k$ gives raise to the existence of the Σ'_{cd}-implication $P_{k'} \to x_{k'}$ such that $P_{k'} \subseteq C_k \cup C_{k'} \backslash x_k$. When deleting the implication $C_k \to x_k$ from Eq. 12, then replacing the implication $C_{k'} \to x_{k'}$ by $P_{k'} \to x_{k'}$, a new implication $P' \to x$ would be provided. The premisse P' of these new implication is deduced from P: it consists in replacing in P the subsets issued from the two implications $C_k \rightarrow x_k$ and $C_{k'} \rightarrow x_{k'}$ (i.e. $C_k \backslash (x_{k+1} \ldots x_m)$ and $C_{k'} \backslash (x_k \ldots x_m)$) by the subset issued from the implication $P_{k'} \rightarrow x_{k'}$ (i.e. $P_{k'} \backslash (x_{k+1} \ldots x_m)$ since $k' < k$). Moreover, these new subsets is included in P:

$$\begin{aligned} P_{k'} \backslash (x_{k+1} \ldots x_m) &\subseteq (C_k \cup C_{k'} \backslash x_k) \backslash (x_{k+1} \ldots x_m) \\ &\subseteq C_k \backslash (x_{k+1} \ldots x_m) \cup C_{k'} \backslash (x_k x_{k+1} \ldots x_m) \subseteq P \end{aligned}$$

Therefore $P' \subseteq P$ and $P \rightarrow x$ not minimal in Σ_{cd}. This achieves the proof.

case 2 Let $P_1 \rightarrow y$ be the implication issued from Eq 13. Let $P_2 \rightarrow y$ be the implication obtained with $A \rightarrow c$ treated at first, i.e

$$P_2 \rightarrow x = (C_m \rightarrow x_m \otimes_{\mathbb{D}} (\ldots \otimes_{\mathbb{D}} (C_1 \rightarrow x_1 \otimes_{\mathbb{D}} (B \rightarrow x \otimes_{\mathbb{D}} A \rightarrow y))))$$

Using Definition 3 of the $\otimes_{\mathbb{D}}$-operator, one can verify that P_2 is included in P_1, thus $P_1 \rightarrow y$ not minimal in Σ_{cd} as stated.
Indeed, application of the last $\otimes_{\mathbb{D}}$-operator to $A \rightarrow y$ gives raise to the addition of $A \backslash x$ to P_1, whereas $A \backslash (x x_1 \ldots x_m)$ will finally be added to P_2 when $A \rightarrow y$ is treated at first. Therefore $A \backslash (x x_1 \ldots x_m) \subseteq A \backslash x$ and $P_2 \subseteq P_1$, thus P_1 not minimal.

case 3 Let $P \rightarrow y_m$ be the implication issued from Eq 14. Since $A_i \rightarrow y_i$ are Σ'_{cd} implications, they can equivalently be replaced, using Proposition 5, by the implication $A \rightarrow y_m$ with $A \subseteq A_1 \cup A_2 \backslash y_1 \cup \ldots A_m \backslash (y_{n-1})$. Using Definition 3 of the $\otimes_{\mathbb{D}}$-operator, one can verify that A is then included in P, thus $P \rightarrow y_m$ not minimal in Σ_{cd} as stated.

Algorithm 2 is a direct implementation of Theorem 4. It first computes the two sets $Left$ and $Right$ of Σ'_{cd}-implications for which the make-direct treatment has to be considered: the implications with the conclusion included in B are in Left, and the implications containing x in their premisse are in Right. It then manages the set $LeftSet$ of implications that contains, at each iteration k, a description of every set of k implications $\{C_1 \rightarrow x_1, \ldots, C_k \rightarrow x_k\} \subseteq Left$ verified Eq 9 for wich the make-direct treatment has to be applied. Each set of k implications is given by a pair (P, S) such that $P = C_1 + \ldots + C_k$ and $S = x_1 \ldots x_k$. The make-direct treatment (the $\otimes_{\mathbb{D}}$-operator) has then to be applied to $LeftSet \times \{B \rightarrow x\}$ (case 1 of Theorem 4) and to $LeftSet \times \{B \rightarrow x\} \times Right$ (case 2 of Theorem 4).

4.3 Comparison

Complexity of Algorithm 1 stays exponential in the worst case, as in Bertet and Nebut's algorithm and Wild's algorithm. However, it is important to notice that the reduction treatment is performed together with the make-direct treatment (the $\otimes_{\mathbb{D}}$-operator) by Algorithm 2 each time a new implication is added, thus a better worst case complexity as stated by Proposition 7.

Proposition 7. *1. Algorithm 2 computes the canonical direct basis issued from the addition of a new implication $B \rightarrow x$ in a canonical direct basis Σ'_{cd} in $O(|S||\Sigma'_{cd}|^{|B|+1})$.*

Name: `AddDirect`
Input: A canonical direct basis Σ'_{cd}
A proper and Σ'_{cd}-minimal implication $B \to x$
Output: The canonical direct basis $(\Sigma' \cup \{B \to x\})_{cd}$
begin
$\backslash **$ `Initialisations` $** \backslash$
Left= $\{C \to e \in \Sigma'_{cd} \,:\, e \in B$ and $x \notin C\}$
Right= $\{A \to d \in \Sigma'_{cd} \,:\, x \in A$ and $d \notin B\}$
newImpl= $\emptyset$; LeftSet= $\{(\emptyset, \emptyset)\}$
repeat
foreach $(P, S) \in$ *LeftSet* **do**
$\backslash **$ `Make-direct treatement T4 according to LeftSet` $** \backslash$
add $P \cup B \backslash S \to x$ to *newImpl*
foreach $A \to d \in$ *Right* **do** add $P \cup B \backslash S \cup A \backslash (S + x) \to d$ to *newImpl*
$\backslash **$ `Updating of LeftSet for the next iteration` $** \backslash$
delete (P, S) from LeftSet
foreach $C \to e \in$ *Left* **do**
if $C \not\subset P$ *and* $e \notin S$ *and* $e \notin P$ **then** add $(P \cup C, S + e)$ in LeftSet
until *LeftSet*$\neq \emptyset$;
$\backslash **$ `Reduction treatment T5 according to newImpl` $** \backslash$
foreach $A \to b \in$ *newImpl* **do**
if *there exists* $C \to b \in \Sigma'_{cd}$ *such that* $A \subset C$ **then** delete $C \to b$ from Σ'_{cd}
if *there exists* $C \to b \in \Sigma'_{cd}$ *such that* $C \subset A$ **then** delete $A \to b$ from newImpl
return $\Sigma'_{cd} \cup$ *newImpl*
end

Algorithm 2: Addition of an implication $B \to x$ in a canonical direct basis Σ'_{cd}

2. *Algorithm 1 incrementally computes the canonical direct basis of an UIS* $\Sigma = \{B_i \to x_i \,:\, i \in [1, n]\}$ *in* $O(|S|\, 2^{|B_1| * \ldots * |B_n|})$.

Proof. 1. Initialisations are clearly in $O(|\Sigma'_{cd}||S|)$. The reduction treatment is in $O(|\Sigma'_{cd}||S||newImpl|)$, and thus depends on $newImpl$ that is generated by the `repeat` loop. Each iteration k of this loop increases $newImpl$ with at most $|Right| * |LeftSet| \leq |\Sigma'_{cd}||LeftSet|$ implications. So, to estimate $newImpl$, let us provide a majoration of $LeftSet$.
At each iteration k, $LeftSet$ contains a description of every set of k implications $\{C_1 \to x_1, \ldots, C_k \to x_k\} \subseteq Left$ for wich the make-direct treatment has to be applied. Thus, at each iteration k, $LeftSet$ described at most $|Left|^k \leq |\Sigma'_{cd}|^k$ implications, and $newImpl$ is increased with at most $|\Sigma'_{cd}|^{k+1}$ implications. Moreover, one can also deduce that complexity of one iteration of the `repeat` loop is in $O(|S||\Sigma'_{cd}|^k)$.
The number of iterations can be majored by $|B|$ since, as precised in Theorem 4, conclusions of the implications described in $LeftSet$ have to be different, and included in B. Therefore a final complexity in

$$O(|S|(|\Sigma'_{cd}| + |\Sigma'_{cd}|^2 + \ldots + |\Sigma'_{cd}|^{|B|})) \leq O(|S||\Sigma'_{cd}|^{|B|+1}) \tag{17}$$

2. Complexity of Algorithm 1 is a direct consequence of complexity of Algorithm 2.

Therefore, Algorithm 1 would gives better results in practice than Bertet and Nebut's algorithm and Wild's algorithm. Notice that better results would also be obtained with an addition of the implications according to a decreased order on the size of their premisse. A simple experimentation to compare the incremental Algorithm 1 with the non-incremental algorithm described by Proposition 1 has been realized, where UIS are randomly generated for $|S| = 20$ and 10 implicationnal rules of premisse majorated by $\frac{|S|}{3}$. Table 1 gives the number of rules that have been added by the make-direct treatment before to be deleted by the reduction treatment by each of the two algorithms. This number of rules corresponds to the intermediary rules that are generated, thus a comparison between these algorithms. It clearly appears that Algorithm 1 generates less intermediary rules.

Canonical direct Basis	51	52	52	68	31	30	32
Incremental (Algorithm 1)	61	60	0	28	2	12	0
Non-incremental (Proposition 1)	961	447	709	2038	425	211	266

Table 1. number of rules of the canonical direct basis and number of rules intermediary generated by each algorithm

5 Conclusion

This paper focuses on some algorithmical aspects of the canonical direct basis, and proposes a new incremental generation algorithm of the canonical direct basis from an UIS $\Sigma = \{B_i \to x_i : i \in [1,n]\}$ in $O(|S|\, 2^{|B_1|*\ldots*|B_n|})$. This algorithm incrementally adds a new implication $B \to x$ in a canonical direct basis Σ'_{cd} and then computes $(\Sigma'_{cd} \cup B \to x)_{cd}$ in $O(|S||\Sigma|^{|B|+1})$. This new algorithm has better worst case complexity than existing algorithms. It has been implemented in a Java class, and a first experimentation also give good results.

References

1. K. Bertet and B. Monjardet. The multiple facets of the canonical implicationnal basis. In *Les cahiers du CAMS*. 2005.
2. K. Bertet and M. Nebut. Efficient algorithms on the Moore family associated to an implicational system. *DMTCS*, 6(2), 2004.
3. G. Birkhoff and O. Frink. Representations of lattices by sets. *Transactions of the American Mathematical Society*, 64:299–316, 1948.
4. N. Caspard and B. Monjardet. The lattice of closure systems, closure operators and implicational systems on a finite set: a survey. *Discrete Applied Mathematics*, 127(2):241–269, 2003.

5. M.L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
6. B. Ganter. Two basic algorithms in concept lattices. Technical report, Technische Hochschule Darmstadt, 1984.
7. B. Ganter and R. Wille. *Formal concept analysis, Mathematical foundations.* Springer Verlag, Berlin, 1999.
8. J.L. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathematiques & Sciences Humaines*, 95:5–18, 1986.
9. D. Maier. *The Theory of Relational Databases.* Computer Sciences Press, 1983.
10. H. Mannila and K.J. Räihä. *The design of relational databases.* Addison-Wesley, 1992.
11. L. Nourine. *Une structuration algorithmique de la théorie des treillis.* PhD thesis, Université of Montpellier I, July 2000.
12. L. Nourine and O. Raynaud. A fast algorithm for building lattices. *Information Processing Letters*, 71:199–204, 1999.
13. S. Obiedkov and V. Duquenne. Incremental construction of the canonical implication basis. In *Fourth International Conference Journées de l'Informatique Messine*, pages 15–23, 2003. submitted to Discrete Applied Mathematics.
14. A. Rush and R. Wille. Knowledge spaces and formal concept analysis. In H.H. Bock and W. Polasek, editors, *Data Analysis and Information Systems*, pages 427–436, Berlin, 1995. Springer Verlag.
15. R. Taouil and Y. Bastide. Computing proper implications. In 9^{th} *International Conference on Conceptual Structures*, Stanford, USA, 2002.
16. M. Wild. A theory of finite closure spaces based on implications. *Advances in Mathematics*, 108(1):118–139, 1994.
17. M. Wild. Computations with finite closure systems and implications. In *Proceedings of the 1st Annual International Conference on Computing and Combinatorics*, volume 959 of *LNCS*, pages 111–120. Springer, 1995.