



**HAL**  
open science

# The multiple facets of the canonical direct unit implicational basis

Karell Bertet, Bernard Monjardet

► **To cite this version:**

Karell Bertet, Bernard Monjardet. The multiple facets of the canonical direct unit implicational basis. 2008. halshs-00308798v1

**HAL Id: halshs-00308798**

**<https://shs.hal.science/halshs-00308798v1>**

Preprint submitted on 1 Aug 2008 (v1), last revised 26 Jan 2010 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The multiple facets of the canonical direct unit implicational basis

K. Bertet and B. Monjardet

*L3I - Université de La Rochelle - av Michel Crépeau - 17042 La Rochelle*

`karell.bertet@univ-lr.fr`

*CERMSEM - Maison des Sciences Economiques - Université Paris 1 - 106-112 bd  
de l'Hopital - 75013 Paris*

`bernard.monjardet@univ-paris1.fr`

---

## Abstract

The notion of *dependencies* between "attributes" arises in many areas such as relational databases, data analysis, data-mining, formal concept analysis, knowledge structures . . . . Formalization of dependencies leads to the notion of so-called *full implicational systems* (or full family of functional dependencies) which is in one-to-one correspondence with the other significant notions of *closure operator* and of *closure system*. An efficient generation of a full implicational system (or a closure system) can be performed from equivalent implicational systems and in particular from bases for such systems, for example, the so-called *canonical basis*. This paper shows the equality between five other bases originating from different works and satisfying various properties (in particular they are unit implicational systems). The three main properties of this unique basis are the directness, canonical and minimal properties, whence the name *canonical direct unit implicational basis* given to this unit implicational system. The paper also gives a nice characterization of this canonical basis and it makes precise its link with the prime implicants of the Horn function associated to a closure operator. It concludes that it is necessary to compare more closely related works made independently, and with different terminology in order to take advantage of the really new results in these works.

*Key words:* implicational system, closure operator, closure system, canonical direct basis, lattice, Horn function.

---

## 1 Introduction

In this paper, we deal with "implications", and more detailed explanations are first required for our use of this term. Consider data organized as a set  $\Omega$  of

”objects” (also denoted prototypes, observations, ...) together with a set  $S$  of ”attributes” (also denoted characteristics, descriptors, fields, ...), and where each object is related to a subset of attributes by a binary relation between the objects and the attributes. Such a *data set* appears in several domains, for instance in Data Analysis ([19]), in Data Mining ([31]), in Knowledge Spaces ([18]), in Formal Concept Analysis (FCA, [26]). For example, objects are patients, consumers, students or planets ; attributes are symptoms, products, problems, characteristic. Each patient is described by the list of the symptoms he manifests ; each consumer is described by the list of products he buys ; each student is described by the list of problems he solves ; each planet by the list of the characteristics that it possesses. It is convenient to adopt here FCA’s terminology and to call a *context* the triple composed of the set  $\Omega$  of objects, the set  $S$  of attributes and the binary relation  $R$  between  $\Omega$  and  $S$ .

When all the consumers buying the two products  $x$  and  $y$  also buy the product  $z$ , or, when all the students solving the two problems  $x$  and  $y$  also solve the problems  $z$ , there is a dependence between  $x$  and  $y$  on one hand, and  $z$  on the other hand. In the general case, there is a dependence between two subsets  $X$  and  $Y$  of attributes when all objects related to the attributes of  $X$  also are related to the attributes of  $Y$ . Such a dependence is called a *valid association rule* in Data Mining, i.e. an association rule where the proportion of objects related to  $X$  and  $Y$  among the objects related to  $X$  (also called the *confidence*) is equal to 100%. In Formal Concept Analysis, one says that  $X$  *implies*  $Y$ . It is in this sense that the term *implication* is used in this paper, and an implication between  $X$  and  $Y$  will be denoted  $X \rightarrow Y$ . It is clear that these implications between attributes are ”contextual” since they depend on the given context.

The theory of relational databases induces the same notion of implication between attributes. Data is organized as *tables* (or *relations* according Codd’s 1970 terminology [13]) that corresponds to a relation between a list of ”records” and a set of  $t$  multi-valued attributes. A record is then a tuple of values, one for the domain of values of each attribute. Consider the case where all the records related to the same values on a set  $X$  of attributes are also related to the same values on another set  $Y$  of attributes. Then in the theory of relational databases one says that  $Y$  *functionally depends* on  $X$  or that  $X$  *determines*  $Y$  or that there is a *functional dependency* (FD) between  $X$  and  $Y$ . It is easy to define a binary relation between the set of all pairs of records and the set of attributes so that  $Y$  functionally depends on  $X$  if and only if  $X$  implies  $Y$  with respect to this context (see [26]).

Consider a context and the set of all associated implications between subsets of the set  $S$  of attributes. Formally, the implication  $X \rightarrow Y$  is an ordered pair  $(X, Y)$  of subsets of  $S$ . So, the set of all implications between attributes is a binary relation on the power set  $\mathcal{P}(S)$  of the attributes. It is useful to consider any binary relation on  $\mathcal{P}(S)$  (it will be clear why below). Such an (arbitrary)

binary relation on  $\mathcal{P}(S)$  is called here an *implicational system* (it is called a *closed set of implications* in FCA and a *set of functional dependencies* in the relational data model).

It is also useful to consider a *unit implicational system* defined as a binary relation between  $\mathcal{P}(S)$  and  $S$ . It is clear that one can associate a unit implicational system to an implicational system : any implication  $X \rightarrow Y$  can be replaced by the set of unit implications  $\{X \rightarrow y, y \in Y\}$ . Conversely, one can associate an implicational system to a unit implicational system: for instance, the set of implications  $X \rightarrow Y = \{y \in S : X \rightarrow y\}$ . Observe that this correspondence is not "one-to-one" (see [26]).

Let us now return to the implicational system associated to a context  $(\Omega, S, R)$ . It is not an arbitrary relation on  $\mathcal{P}(S)$ . For instance if  $X \rightarrow Y$  and  $Y \rightarrow Z$  one also has  $X \rightarrow Z$  (check what it means in the context associated to a data set context as well as in the context associated to databases). Such an implicational system is called here a *full implicational system*; in the theory of knowledge structures it is called an *entail relation*, in FCA a *closed set of implications* and in the theory of the relational databases a *full family of functional dependencies* or a *relational databases scheme* or even a *relation scheme* (at least by some authors since the terminology of databases is far to be unified). A fundamental fact first observed by Armstrong in ([4]) in the theory of relational databases is the following: "there is a one to one correspondence between the set of all the full implicational systems defined on a set  $S$  and the set of all closure operators defined on  $S$ ". These sets are also in a one to one correspondence with many other sets (see [12]) and in particular with the set of all *full unit implicational systems* (called *entailments* in the theory of knowledge structures), the set of all *closure systems* and the set of all *pure Horn (Boolean) functions* (precise definitions and references are given in Sections 2, 5 and 6).

Now the same problem has been encountered in all the aforementioned domains. Take, for instance, the full family of functional dependencies associated to a table in a relational database. It contains many dependencies but some of them are trivial (for instance,  $X \rightarrow Y$  if  $Y \subseteq X$ ) and some can be deduced from others (for instance, if  $X \rightarrow y$  and  $y \rightarrow z$  one has also  $X \rightarrow z$ ). So one searches for "small" generating implicational systems allowing us to recover a given full implicational system (the definition of a generating system is given in section 2.3). Observe that thanks to the correspondence between full implicational systems and closure operators, a generating system allows us just as well to recover a closure operator. In this paper we will rather consider that one wants to efficiently recover a closure operator (which can be the closure operator corresponding to a full implicational system).

There exists a significant result on the minimal generation of a closure operator

(or of a full implicational system) by an implicational system. It has been obtained independently (and with different formulations) by Maier ([39]) and Guigues and Duquenne ([28]). The generating implicational system obtained is often called the Duquenne-Guigues *canonical basis*. Here we will not be concerned with this basis since our results bear on the generation of a closure operator by a unit implicational system. We will show that five generating unit implicational systems obtained by different authors in different fields and with different formalisms are in fact identical. This unique generating system has properties that justify calling it the *canonical direct unit implicational basis* (but it is not the unit implicational system associated with the Duquenne-Guigues canonical basis). Moreover, finding it is the same as finding the set of the *prime implicants* of a Boolean function.

We end this introduction by presenting the contents of the different sections of the paper. The second section recalls the notions about posets, lattices, closure operators or closure systems, and (unit) implicational systems we will use. In the third section we describe the five unit implicational systems proposed by different authors in order to efficiently generate a closure operator (for reasons explained later they are called "bases" of the closure operator). The fourth section contains our main results. We prove that these five bases are the same and thus they define an unique basis which can be called the canonical direct unit implicational basis. Whereas some of these equalities are easy to obtain, others are deduced from a non obvious characterization of a *direct basis*.

One of the corollaries of these results shows that the *necessary sets for  $x$*  (defined in the context of relational databases) can be identified with the  *$x$ -dominating sets* (defined in the context of choice functions in microeconomics). It is (more or less) well known that closure systems on a set  $S$  are in a one-to-one correspondence with the so-called *pure Horn Boolean functions* defined on  $\mathcal{P}(S)$ . In the fifth section we show that finding the canonical direct unit implicational basis is the same as finding the prime implicants (or the prime implicates) of a (pure) Horn Boolean function. The first part of section 6 is a historical note on the appearance of the notions considered in this paper. We also mention there the works that seek to relate these notions to traditional notions in logic ([20,24,26,25]). The second part of section 6 is an overview of the algorithmic results related to the notions and constructions considered in this paper. The conclusion mentions some possible further research, and, in particular, the need to compare more closely related works made independently in various domains in order to take advantage of the results really new.

## 2 Recalls and Definitions

### 2.1 Posets and Lattices

A *partially ordered set*  $P = (S, \leq)$ , also called a *poset*, is a set  $S$ <sup>1</sup> 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  $x \leq z$ ). We denote by  $<$  the irreflexive relation associated to  $\leq$ , and by  $\prec$  the *cover relation* defined by  $x \prec y$  if  $x < y$  and if there exists no  $z \in S$  with  $x < z < y$ . We then say that  $x$  is *covered* by  $y$  (or  $y$  *covers*  $x$ ). A poset  $P$  can also be given by its cover relation  $\prec$  ( $P = (S, \prec)$ ). The induced graphical representation is called the (*Hasse*) *diagram* of  $P$ . In what follows, we will write indifferently  $x \in S$  or  $x \in P$ . 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 element (according to the relation  $\leq$ ) called the *bottom* of the lattice, and denoted  $\perp_L$  (or simply  $\perp$ ). Respectively, a lattice contains a maximum element called the *top* of the lattice, and denoted  $\top_L$  (or simply  $\top$ ).

An element  $j$  (respectively,  $m$ ) of a lattice  $L$  is a *join-irreducible* (respectively, *meet-irreducible*) of  $L$  if it cannot be obtained as the join (respectively, meet) of elements of  $L$  all distinct from  $j$  (respectively, from  $m$ ). Equivalently, an element  $j$  (respectively,  $m$ ) of  $L$  is a join- (respectively, meet-) irreducible if it covers (respectively, is covered by) a unique element in  $L$ , which is then denoted by  $j^-$  (respectively,  $m^+$ ) and called the *lower cover* of  $j$  (respectively, *upper cover* of  $m$ ). The sets of join-irreducibles and of meet-irreducibles of a lattice  $(L, \leq)$  are respectively denoted by  $J_L$  and  $M_L$ . For an element  $x$  in  $L$ , we denote by  $J_x$  (respectively,  $M_x$ ) the set of all join-irreducibles  $j$  (respectively, meet-irreducibles  $m$ ) such that  $j \leq x$  (respectively,  $x \leq m$ ).

### 2.2 Set systems and Lattices

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  $\mathbb{F}$ . 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  $F_1 \vee F_2 = \bigcap\{F \in \mathbb{F} \mid F_1 \cup F_2 \subseteq F\}$ . Moreover, any lattice  $L$  is isomorphic

<sup>1</sup> In this paper, all the sets will be finite

to the lattice of closed sets of a closure system ([10]). The simplest closure system representing  $L$  is defined on  $J_L$ : it is the set system  $\{J_x \mid x \in L\}$ .

**Example 1** Consider the closure system<sup>2</sup> on the set  $S = \{1, 2, 3, 4, 5\}$ :

$$\mathbb{F} = \{\emptyset, 1, 2, 3, 4, 12, 13, 45, 234, S\}$$

One can verify that it is stable by intersection. The lattice  $(\mathbb{F}, \subseteq)$  is represented by its Hasse diagram in Figure 1. We will use this example to illustrate several notions in this paper.

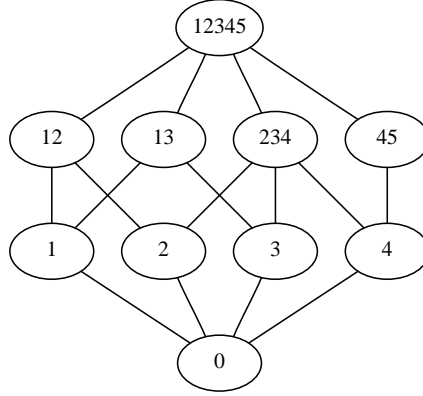


Fig. 1. The lattice  $(\mathbb{F}, \subseteq)$  represented by its Hasse diagram, where  $\mathbb{F}$  is the closure system of our example.

A *closure operator* on a set  $S$  is a map  $\varphi$  on  $\mathcal{P}(S)$  satisfying,  $\forall X, Y \subseteq S$ :

$$X \subseteq \varphi(Y) \Leftrightarrow \varphi(X) \subseteq \varphi(Y) \quad (1)$$

Equivalently, and more commonly, a closure operator is defined as a map  $\varphi$  satisfying the three following properties:  $\varphi$  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.  $\forall X \subseteq S, \varphi^2(X) = \varphi(X)$ ). Still equivalently, a closure operator is an extensive map satisfying the *path-independence* property (i.e.  $\forall X, Y \subseteq S, \varphi(X \cup Y) = \varphi(\varphi(X) \cup Y)$ ). The set  $\varphi(X)$  is called the *closure* of  $X$  by  $\varphi$ . The set  $X$  is said to be *closed* by  $\varphi$  whenever it is a fixed point of  $\varphi$ , i.e. when  $\varphi(X) = X$ .

Closure operators are in one-to-one correspondence with closure systems. On the first hand, the set of all closed elements of  $\varphi$  forms a closure system  $\mathbb{F}_\varphi$ :

$$\mathbb{F}_\varphi = \{F \subseteq S \mid F = \varphi(F)\} \quad (2)$$

<sup>2</sup> In this example as in the following, a subset  $X = \{x_1, x_2, \dots, x_n\}$  is written as the word  $x_1x_2 \dots x_n$ . Moreover, we abuse notation in the following and use  $X + x$  (respectively,  $X \setminus x$ ) for  $X \cup \{x\}$  (respectively,  $X \setminus \{x\}$ ), with  $X \subseteq S$  and  $x \in S$ .

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  $X$ :

$$\varphi_{\mathbb{F}}(X) = \bigcap \{F \in \mathbb{F} \mid X \subseteq F\} \quad (3)$$

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

A subset  $B$  of  $S$  is a *basis* of  $F$ , with  $F$  closed set for  $\varphi$ , if  $\varphi(B) = F$  and  $\varphi(A) \subset \varphi(B)$  for every  $A \subset B$  (in other words,  $B$  is a *minimal generating set* of  $F$ ). A subset  $B$  of  $S$  is *free* if for every  $x \in B$   $x \notin \varphi(B \setminus x)$ . Or, equivalently,  $B$  is free if and only if  $\varphi(A) \subset \varphi(B)$  for every  $A \subset B$ , or if and only if  $B$  is a basis of  $\varphi(B)$ . An element  $x$  of a subset  $X$  of  $S$  is an *extreme point* of  $X$  if  $x \notin \varphi(X \setminus x)$ . One denotes by  $\mathbb{E}x_{\varphi}(X)$  or simply  $\mathbb{E}x(X)$  the set of extreme points of  $X$ . Observe that  $X$  is free if and only if  $\mathbb{E}x(X) = X$ . A subset  $C$  of  $S$  is a *copoint* of  $x \in S$  if  $C$  is a maximal subset of  $S$  such that  $x \notin \varphi(C)$ . It is well known that in the lattice  $\mathbb{F}_{\varphi}$ , the copoints of  $x$  are meet-irreducible closed sets.

### 2.3 Unit Implicational System

A *Unit Implicational System* (UIS for short)  $\Sigma$  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  $\Sigma$ -*implication* whose *premise* is  $A$  and *conclusion* is  $b$ . It is written  $A \rightarrow_{\Sigma} b$  or  $A \rightarrow b$  (meaning “ $A$  implies  $b$ ”). A subset  $X \subseteq S$  *respects* a  $\Sigma$ -implication  $A \rightarrow b$  when  $A \subseteq X$  implies  $b \in X$  (i.e. “if  $X$  contains  $A$  then  $X$  contains  $b$ ”).

$X \subseteq S$  is  $\Sigma$ -*closed* when  $X$  respects all  $\Sigma$ -implications, i.e  $A \subseteq X$  implies  $b \in X$  for every  $\Sigma$ -implication  $A \rightarrow b$ . The set of all  $\Sigma$ -closed sets forms a closure system  $\mathbb{F}_{\Sigma}$  on  $S$ :

$$\mathbb{F}_{\Sigma} = \{X \subseteq S \mid X \text{ is } \Sigma\text{-closed}\} \quad (4)$$

Then, we can associate to  $\Sigma$  a closure operator  $\varphi_{\Sigma} = \varphi_{\mathbb{F}_{\Sigma}}$ . One can state ([59,60]) that  $\varphi_{\Sigma}$  is the closure operator obtained by the iteration of the following isotone and extensive map, with  $X \subseteq S$ :

$$\varphi_{\Sigma}(X) = \pi_{\Sigma}(X) \cup \pi_{\Sigma}^2(X) \cup \pi_{\Sigma}^3(X) \cup \dots \quad (5)$$

where

$$\pi_{\Sigma}(X) = X \cup \bigcup \{b \mid A \subseteq X \text{ and } A \rightarrow_{\Sigma} b\} \quad (6)$$

and

$$\pi_{\Sigma}^2(X) = \pi_{\Sigma}(X) \cup \bigcup \{b \mid A \subseteq \pi_{\Sigma}(X) \text{ and } A \rightarrow_{\Sigma} b\} \quad (7)$$



Observe that the procedure in (5) terminates since  $S$  is finite. Moreover,  $\varphi_\Sigma(X) = \pi_\Sigma^n(X)$  with  $n \leq |S|$  being the first integer such that  $\pi_\Sigma^n(X) = \pi_\Sigma^{n+1}(X)$ , and it is well known that iteration of an isotone and extensive map defined on a finite set leads to an idempotent map, i.e. a closure operator.

Now, consider a closure operator  $\varphi$  on  $S$ . Then the closed sets of  $\varphi$  coincide with the  $\Sigma$ -closed sets of the following UIS:

$$\Sigma_\varphi = \{X \rightarrow y \mid y \in \varphi(X) \text{ and } X \subseteq S\} \quad (8)$$

It is easy to see that  $\Sigma_\varphi$  satisfies the two following properties:

**F1**  $x \in X \subseteq S$  imply  $X \rightarrow_{\Sigma_\varphi} x$ .

**F2** for every  $y \in S$  and all  $X, Y \subseteq S$ , [ $X \rightarrow_{\Sigma_\varphi} y$  and  $\forall x \in X, Y \rightarrow_{\Sigma_\varphi} x$ ] imply  $Y \rightarrow_{\Sigma_\varphi} y$ .

Unit IS satisfying properties F1 and F2 are called full UIS and are in one-to-one correspondence with closure operators, and thus with closure systems and lattices (via the set representation of lattices by the closure system  $\{J_x \mid x \in L\}$ ).

The set of all full UIS is itself a closure system defined on the set of UIS. So, when a UIS  $\Sigma$  is not full, there exists a least full UIS containing it. This full UIS is nothing else than  $\Sigma_\varphi$  where  $\varphi = \varphi_\Sigma$  is the closure operator associated with  $\Sigma$  (see Equation 5). This full UIS  $\Sigma_\varphi$  can be obtained by applying recursively rules F1 and F2 to  $\Sigma$ . The UIS  $\Sigma$  is then called a *generating system* (or *cover* in relational data bases) for the full UIS  $\Sigma_\varphi$ , and thus for the induced closure operator  $\varphi$ , the closure system  $\mathbb{F}_\Sigma$ , and the induced lattice  $(\mathbb{F}_\Sigma, \subseteq)$ . When some UISs  $\Sigma$  and  $\Sigma'$  on  $S$  are generating systems for the same closure system, they are called *equivalent* (i.e.  $\mathbb{F}_\Sigma = \mathbb{F}_{\Sigma'}$ ).

An illustration of a generating system of a full UIS  $\Sigma_\varphi$ , is given by the UIS  $\Sigma_{free}$  composed of the subsets of  $S$  that also are free subsets:

$$\Sigma_{free} = \{X \rightarrow y : y \in \varphi(X) \setminus X \text{ and } X \text{ free subset of } S\} \quad (9)$$

An UIS  $\Sigma$  is called *direct* or *iteration-free* if for every  $X \subseteq S$ ,  $\varphi_\Sigma(X) = \pi_\Sigma(X)$  (see Equation (6)). An UIS  $\Sigma$  is *minimal* or *non-redundant* if  $\Sigma \setminus \{X \rightarrow y\}$  is not equivalent to  $\Sigma$ , for all  $X \rightarrow y$  in  $\Sigma$ . It is *minimum* if it is of least cardinality, i.e. if  $|\Sigma| \leq |\Sigma'|$  for all UIS  $\Sigma'$  equivalent to  $\Sigma$ . A minimum UIS is trivially non-redundant, but the converse is false.  $\Sigma$  is *optimal* if  $s(\Sigma) \leq s(\Sigma')$  for all UIS  $\Sigma'$  equivalent to  $\Sigma$ , where the *size*  $s(\Sigma)$  of  $\Sigma$  is defined by:

$$s(\Sigma) = \sum_{A \rightarrow b \in \Sigma} (|A| + 1) \quad (10)$$

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  $X \rightarrow_{\Sigma} x$  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 rule:

**F3** delete  $A \rightarrow_{\Sigma} b$  from  $\Sigma$  when  $b \in A$ .

In this paper, all UISs will be considered to be proper UISs. Then, for instance, the term full IS will means "the proper full IS deduced from the full IS by applying F3". Other definitions and bibliographical remarks can be found in the survey of Caspard and Monjardet in [12].

**Example 2** Consider the closure system of our example given by the lattice  $(\mathbb{F}, \subseteq)$  in Figure 1 and the generating system  $\Sigma_{free}$ :

$$\Sigma_{free} = \left\{ \begin{array}{l} (1) 5 \rightarrow 4 \quad (2) 23 \rightarrow 4 \quad (3) 24 \rightarrow 3 \quad (4) 34 \rightarrow 2 \\ (5) 14 \rightarrow 2 \quad (6) 14 \rightarrow 3 \quad (7) 14 \rightarrow 5 \quad (8) 25 \rightarrow 1 \\ (9) 35 \rightarrow 1 \quad (10) 15 \rightarrow 2 \quad (11) 35 \rightarrow 2 \quad (12) 15 \rightarrow 3 \\ (13) 25 \rightarrow 3 \quad (14) 123 \rightarrow 5 \quad (15) 15 \rightarrow 4 \quad (16) 25 \rightarrow 4 \\ (17) 35 \rightarrow 4 \quad (18) 123 \rightarrow 4 \end{array} \right.$$

Notice that  $\Sigma_{free}$  is a proper UIS since for every implication, the conclusion is not included in the premise. Concerning the direct property, it is clear that  $\Sigma_{free}$  is a direct UIS.

### 3 Some interesting bases

In this section we are going to define several proper UIS which are generating systems for a given closure operator  $\varphi$  (equivalently for a given closure system  $\mathbb{F}$ ) which can be the closure operator associated to a given UIS  $\Sigma$ . In the literature on IS, the term basis is often used not only for minimal IS but also for IS satisfying various minimality criteria. We will do the same by defining five such bases.

### 3.1 The direct-optimal basis $\Sigma_{do}$

A number of problems related to closure systems, (thus closure operators, lattices or implicational systems) can be answered by computing closures of the type  $\varphi_{\Sigma}(X)$ , for some  $X \subseteq S$ . According to the definition (see Eq.(5))  $\varphi(X)$  can be obtained given an UIS  $\Sigma$  by iteratively scanning  $\Sigma$ -implications:  $\varphi(X)$  is initialized with  $X$  then increased with  $b$  for each implication  $A \rightarrow_{\Sigma} b$  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  $\varphi_{\Sigma}(X) = \pi_{\Sigma}(X)$ . The *direct-optimal* property combines the directness and optimality properties:

**Definition 3** *A UIS  $\Sigma$  is direct-optimal if it is direct, and if  $s(\Sigma) \leq s(\Sigma')$  for any direct UIS  $\Sigma'$  equivalent to  $\Sigma$ .*

In [8], Bertet and Nebut show that a direct-optimal UIS is unique and can be obtained from any equivalent and proper UIS:

**Proposition 4** [8] *The direct-optimal basis  $\Sigma_{do}$  is obtained from any equivalent and proper UIS  $\Sigma$  as follows:*

- (1) *first apply recursively the following rule<sup>3</sup> to obtain a direct equivalent UIS:*  
**F7** *for all  $A \rightarrow_{\Sigma} b$  and  $C + b \rightarrow_{\Sigma} d$  with  $d \neq b$ , add  $A \cup C \rightarrow d$  to  $\Sigma$*
- (2) *then apply the **F3** rule to obtain a proper UIS, and the following rule to minimize premisses of the  $\Sigma$ -implications:*  
**F8** *for all  $A \rightarrow_{\Sigma} b$  and  $C \rightarrow_{\Sigma} b$ , if  $C \subset A$  then delete  $A \rightarrow_{\Sigma} b$  from  $\Sigma$ .*

**Example 5** *Consider our example given by  $(\mathbb{F}, \subseteq)$  in Figure 1. The basis  $\Sigma_{do}$  is:*

$$\Sigma_{do} = \begin{cases} (1) 5 \rightarrow 4 & (2) 23 \rightarrow 4 & (3) 24 \rightarrow 3 & (4) 34 \rightarrow 2 \\ (5) 14 \rightarrow 2 & (6) 14 \rightarrow 3 & (7) 14 \rightarrow 5 & (8) 25 \rightarrow 1 \\ (9) 35 \rightarrow 1 & (10) 15 \rightarrow 2 & (11) 35 \rightarrow 2 & (12) 15 \rightarrow 3 \\ (13) 25 \rightarrow 3 & (14) 123 \rightarrow 5 \end{cases}$$

*One can verify that  $\Sigma_{do}$  is direct like  $\Sigma_{free}$ . Moreover,  $s(\Sigma_{do}) < s(\Sigma_{free})$  and  $\Sigma_{do} \subset \Sigma_{free}$ .*

<sup>3</sup> when  $\Sigma$  is not proper, this rule has to be applied only when  $b \notin A$  and  $d \notin A \cup C$

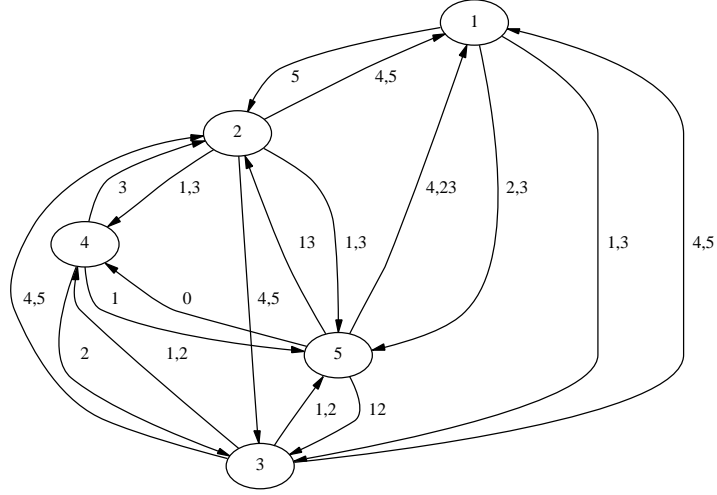


Fig. 2. Relation  $\delta_X$  for  $\mathbb{F}$  of our example represented by a directed graph where each relation  $a\delta_X b$  is represented by an arc and labeled by  $X$  ( $\emptyset$  is denoted by 0)

### 3.2 The dependence relation's basis $\Sigma_{dep}$

The dependence relation's basis  $\Sigma_\delta$  on  $S$  comes from the *dependence relation*  $\delta$  defined for a lattice, and introduced in [44] (see also [46]):

**Definition 6** The dependence relation's basis  $\Sigma_\delta$  is:

$$\Sigma_\delta = \{X + y \rightarrow x : x\delta_X y \text{ and } X \text{ is minimal for this property}\} \quad (11)$$

where the dependence relation  $\delta_X$  is defined on  $S$ , with  $x, y \in S$  and  $X \subset S$ , by:

$$x\delta_X y \text{ if and only if } x \notin \varphi(X), y \notin \varphi(X) \text{ and } x \in \varphi(X + y) \quad (12)$$

The dual relation of the relation  $\delta_X$  has been considered in [5] where it is called *domination*. One can observe that the *dependence relation*  $\delta$  on the lattice  $(\mathbb{F}, \subseteq)$  is then given by  $x\delta y$  if there exists  $X \subseteq S \setminus \{x, y\}$  such that  $x\delta_X y$  (so  $\delta = \cup\{\delta_X, X \subset S\}$ ).

**Example 7** Figure 2 gives the dependence relations  $\delta$  and  $\delta_X$  of our example, where two vertices  $x$  and  $y$  are linked by an arc if  $x\delta y$ . This arc is valued by the subsets  $X$  such that  $x\delta_X y$ . For instance,  $5\delta_4 1$ , and  $5\delta_{23} 1$ .

### 3.3 The canonical iteration-free basis $\Sigma_{cif}$

The canonical iteration-free basis on  $S$  is an implicational system introduced by Wild in [60]. As mentioned in the introduction, this implicational system can be transformed into a unit implicational system denoted  $\Sigma_{cif}$ :

**Definition 8** The unit basis  $\Sigma_{cif}$  deduced from the canonical iteration-free basis is:

$$\Sigma_{cif} = \{B \rightarrow x : x \in \varphi(B) \setminus \pi_\varphi(B) \text{ and } B \text{ is a free subset}\} \quad (13)$$

where  $\pi_\varphi$  is derived from  $\varphi$  as follows:

$$\pi_\varphi(B) = B \cup \{x \in S : \text{there exists } A \subset B \text{ with } x \in \varphi(A)\}^4$$

### 3.4 The left-minimal basis $\Sigma_{lm}$

The *left-minimal basis*  $\Sigma_{lm}$  is the restriction of the (proper) full UIS  $\Sigma_\varphi$  to implications where the premise is of minimal cardinality. Using the definition of  $\Sigma_\varphi$  (see 8),  $\Sigma_{lm}$  can be expressed directly from  $\varphi$ :

**Definition 9** The left-minimal basis  $\Sigma_{lm}$  is:

$$\Sigma_{lm} = \{X \rightarrow y : y \in \varphi(X) \setminus X \text{ and for every } X' \subset X, y \notin \varphi(X')\} \quad (14)$$

An implication  $X \rightarrow y$  is called *left-minimal* when it is a  $\Sigma_{lm}$ -implication. It is also called *proper implication* in [56] where implications are used in the data-mining area research, and *minimal functional dependency* in the domains of relational databases and Horn theories ([39,36]).

**Example 10** For our example,  $\Sigma_{lm}$  is the same as  $\Sigma_{do}$ . Remark that  $\Sigma_{lm}$  of our example has 14 implications, and not 15 as incorrectly written in [12] about the same example (p.37).

### 3.5 The weak-implication basis $\Sigma_{weak}$

The *weak-implication basis* has been introduced by Rush and Wille in [53] to show a connection between the theory of knowledge spaces ([18]) and formal concept analysis ([26]). It is based on the definition of a copoint (recall that a subset  $C$  of  $S$  is a copoint of  $x \in S$  if  $C$  is a maximal subset of  $S$  such that  $x \notin \varphi(C)$ ), and on the following classical notion of transversal set.

A subset  $B$  of a set  $S$  is a *transversal* of a family  $\mathcal{F}$  of subsets of  $S$  if  $B \cap F \neq \emptyset$  for every  $F \in \mathcal{F}$ . A transversal  $B$  is a *minimal transversal* of  $\mathcal{F}$  if for every  $A \subset B$ ,  $A$  is not a transversal of  $\mathcal{F}$  (i.e. there exists  $F \in \mathcal{F}$  with  $A \cap F = \emptyset$ ).

<sup>4</sup> When  $B$  is not a free subset, the condition  $\varphi(A) \subset \varphi(B)$  has to be added.

**Definition 11** [53] *The weak-implication basis  $\Sigma_{weak}$  is:*

$$\Sigma_{weak} = \{B \rightarrow x : B \subseteq S \text{ and } B \text{ is a blockade for } x\} \quad (15)$$

where a blockade for  $x \in S$  (also called  $x$ -block) is a minimal transversal of  $\mathcal{D}_x$ , the following family of subsets of  $S$ :

$$\mathcal{D}_x = \{S \setminus (C + x), C \text{ is a copoint of } x\} \quad (16)$$

**Lemma 12** *Let  $x \in S$  and  $B \subseteq S$ . Then  $B$   $x$ -block implies  $x \notin B$  and  $x \in \varphi(B)$  (i.e.  $B \rightarrow x$ )*

**Proof** Consider an  $x$ -block  $B \subseteq S$ . The first point is immediate: by definition of a blockade for  $x$ , we have  $x \notin B$ . For the second point, suppose  $x \notin \varphi(B)$ . Let  $F \subseteq S$  be a maximal closed set of  $\varphi$  such that  $x \notin F$  and  $\varphi(B) \subseteq F$ . Then  $F$  is a co-point of  $x$ . But  $B \subseteq F$  implies  $B \cap (S \setminus (F + x)) = \emptyset$ , a contradiction with  $B$  an  $x$ -block.  $\square$

## 4 The main results

The main result (Theorem 15) of this paper is to state the equality between the five bases defined in the previous section all of which are thus direct bases. The second main result (Theorem 14) is to give an interesting characterization of the direct property based on an *exchange property*.

This exchange property has been independently introduced in [17] and in a stronger form in [8]. In [17], Demetrovics and Nam Son use it to define the notion of Sperner village and to show its equivalence with the notion of closure operator. In [8], Bertet and Nebut use it in the generation of the direct-optimal basis  $\Sigma_{do}$  where rule F7 results directly from this exchange property.

The characterization of Theorem 14 uses another formulation of the direct property issued from the definition (i.e. for every  $X \subseteq S$ ,  $\varphi(X) = \pi_\Sigma(X)$ ).

**Lemma 13** *An UIS  $\Sigma$  is direct if and only if for every  $X \subseteq S$ ,  $\pi_\Sigma(X) = \pi_\Sigma^2(X)$ .*

**Theorem 14** *A proper UIS  $\Sigma$  is direct if and only if it satisfies the following exchange condition:*

$$\forall A, C \subseteq S, \forall b \in S \setminus A, \forall d \in S \setminus (A \cup C) \text{ and different from } b : \quad (17)$$

$A \rightarrow_\Sigma b$  and  $C + b \rightarrow_\Sigma d$  implies there exists  $G \subseteq A \cup C$  such that  $G \rightarrow_\Sigma d$

**Proof**  $\Rightarrow$ : Let  $\Sigma$  be a direct UIS. Assume that for  $b \in S \setminus A$ ,  $d \in S \setminus (A \cup C)$  and different from  $b$ , we have  $A \rightarrow_{\Sigma} b$  and  $C + b \rightarrow_{\Sigma} d$ , which means  $b \in \varphi_{\Sigma}(A)$  and  $d \in \varphi_{\Sigma}(C + b)$ . Then, using the path-independence property of a closure operator, we get

$$d \in \varphi_{\Sigma}(A \cup (C + b)) = \varphi_{\Sigma}(\varphi_{\Sigma}(A + b) \cup C) = \varphi_{\Sigma}(\varphi_{\Sigma}(A) \cup C) = \varphi_{\Sigma}(A \cup C)$$

Now,  $\Sigma$  being direct, there exists  $G \subseteq A \cup C$  such that  $G \rightarrow_{\Sigma} d$ .

$\Leftarrow$ : Let  $\Sigma$  be a UIS satisfying condition (17). One must show that  $\varphi_{\Sigma}(X) = \pi_{\Sigma}(X)$ , or equivalently by Lemma 13 that  $\pi_{\Sigma}(X) = \pi_{\Sigma}^2(X)$ , or still equivalently (since  $\pi_{\Sigma}$  is extensive) that  $\pi_{\Sigma}^2(X) \subseteq \pi_{\Sigma}(X)$ .

Assume that there exists  $X$  with  $\pi_{\Sigma}(X) \subseteq \pi_{\Sigma}^2(X)$ , i.e. that there exists  $z \in \pi_{\Sigma}^2(X) \setminus \pi_{\Sigma}(X)$ . Then there exists  $Z \subseteq \pi_{\Sigma}(X)$  with  $Z \rightarrow_{\Sigma} z$  and  $z \notin \pi_{\Sigma}(X)$ . We set  $p(Z) = |Z \cap (\pi_{\Sigma}(X) \setminus X)|$ . The proof of  $\varphi_{\Sigma}(X) = \pi_{\Sigma}(X)$  will follow immediately from the proof of the following result:

*if  $p(Z) = p$  then there exists  $Z' \subseteq S$  with  $Z' \rightarrow_{\Sigma} z$  and  $p(Z') < p(Z)$ .*

Indeed, by iteration of this result we would get some  $Z^{(k)}$  with  $Z^{(k)} \rightarrow_{\Sigma} z$  and  $p(Z^{(k)}) = 0$ , which means  $Z^{(k)} \subseteq X$  and  $z \in \pi_{\Sigma}(X)$ , a contradiction with our hypothesis.

First, observe that  $p(Z) > 0$ : if not,  $Z \in X$  and  $z \in \pi_{\Sigma}(X)$ , a contradiction.  $p(Z) > 0$  means that there exists  $y \in Z$  with  $y \in \pi_{\Sigma}(X) \setminus X$ . Thus there exists  $Y \subseteq X$  with  $Y \rightarrow_{\Sigma} y$ . Now writing  $Z = U + y$ , we have  $Y \rightarrow_{\Sigma} y$ ,  $U + y \rightarrow_{\Sigma} z$  with  $y \notin Y$  and (since  $z \notin \pi_{\Sigma}(X)$ )  $z \notin Y \cup U$  and  $z$  different from  $y$ . So, by applying the exchange condition, we get that there exists  $Z' \subseteq Y \cup U$  with  $Z' \rightarrow_{\Sigma} z$ . Moreover, since  $p(Y \cup U) = p(Z) - 1$ , we have  $p(Z') < p(Z)$  like wanted.  $\square$

Now, let us give our other main result.

**Theorem 15** *Let  $\varphi$  be a closure operator defined on a set  $S$ , and the five associated UISs above defined. Then*

$$\Sigma_{do} = \Sigma_{cif} = \Sigma_{dep} = \Sigma_{lm} = \Sigma_{weak}$$

**Proof** We prove first  $\Sigma_{cif} = \Sigma_{dep} = \Sigma_{lm} = \Sigma_{weak}$  by proving  $\Sigma_{cif} \subseteq \Sigma_{dep} \subseteq \Sigma_{lm} \subseteq \Sigma_{weak} \subseteq \Sigma_{cif}$ . Then we prove  $\Sigma_{do} = \Sigma_{lm}$

$\Sigma_{cif} \subseteq \Sigma_{dep}$ : Let  $B \rightarrow x$  be a  $\Sigma_{cif}$ -implication. This means that  $x \in \varphi(B) \setminus \pi_{\varphi}(B)$  where  $B$  is free, i.e.  $x \in \varphi(B)$  and  $x \notin \varphi(A)$  for every  $A \subset B$ . Take any  $y$  in  $B$ . Since  $B \setminus y \subset B$  and  $B$  is free, one has  $x \notin \varphi(B \setminus y)$ ,  $y \notin \varphi(B \setminus y)$  and (obviously)  $x \in \varphi((B \setminus y) + y)$ . If  $X \subset B \setminus y$ ,  $X + y \subset B$ , and so  $x \notin \varphi(X + y)$ .

Then  $B \setminus y$  is minimal such that  $x, y \notin \varphi(X)$  and  $x \in \varphi(X + y)$ , i.e.  $B \rightarrow x$  is a  $\Sigma_{dep}$ -implication.

$\Sigma_{dep} \subseteq \Sigma_{lm}$ : Let  $B = X + y \rightarrow x$  be a  $\Sigma_{dep}$ -implication. Then  $x \notin \varphi(X)$  and for every  $Y \subset X$ ,  $x \notin \varphi(Y + y)$ . So  $B \rightarrow x$  is a  $\Sigma_{lm}$ -implication.

$\Sigma_{lm} \subseteq \Sigma_{weak}$ : Let  $B \rightarrow x$  be a  $\Sigma_{lm}$ -implication. Let us first prove that  $B$  is a transversal of  $\mathcal{D}_x = \{S \setminus (C + x), C \text{ co-point of } x\}$  before to prove that it is a minimal transversal. Since  $x \notin B$ ,  $B$  is a transversal of  $\mathcal{D}_x$  if and only if  $B$  is a transversal of  $\mathcal{D}'_x = \{S \setminus C, C \text{ co-point of } x\}$ . Suppose there exists  $C$  co-point of  $x$  such that  $B \cap (S \setminus C) = \emptyset$  and so  $B \subseteq C$ . Then  $\varphi(B) \subseteq C$  which implies  $x \in C$ , a contradiction with  $C$  co-point of  $x$ .

Suppose now that  $B$  is not a minimal transversal of  $\mathcal{D}_x$ , i.e. that there exists  $Y \subset B$  with  $Y$  transversal of  $\mathcal{D}_x$ . Since  $B$  is left-minimal for the implication  $B \rightarrow x$ , we have  $x \notin \varphi(Y)$ . Then there exists a co-point  $C$  of  $x$  such that  $Y \subseteq \varphi(Y) \subseteq C$ . Therefore  $Y \cap (S \setminus C) = \emptyset$ , a contradiction with  $Y$  transversal of  $\mathcal{D}_x$ .

$\Sigma_{weak} \subseteq \Sigma_{cif}$ : Let  $B \rightarrow x$  be a  $\Sigma_{weak}$ -implication. This means that  $x \in \varphi(B) \setminus B$  and  $B$  is minimal transversal of  $\mathcal{D}_x = \{S \setminus (C + x), C \text{ co-point of } x\}$ . We prove first that  $B$  is free by showing that for any  $A \subset B$  one has  $\varphi(A) \subset \varphi(B)$ . Indeed, when  $A \subset B$ ,  $A$  is not a transversal of  $\mathcal{D}_x$  and there exists a copoint  $C$  of  $x$  such that  $A \cap (S \setminus C + x) = \emptyset$ . So  $A \subseteq C$  (since  $x \notin A$ ) and  $\varphi(A) \subseteq C$ . But  $x \notin C$  implies  $x \notin \varphi(A)$  and so  $\varphi(A) \subset \varphi(B)$ . Moreover, we have just proved that  $x \notin \varphi(A)$  for every  $A \subset B$ , i.e. that  $x \notin \pi_\varphi(B)$ . Finally,  $B \rightarrow x$  is a  $\Sigma_{cif}$ -implication.

$\Sigma_{lm} = \Sigma_{do}$ : To prove the equality  $\Sigma_{lm} = \Sigma_{do}$ , let us prove that  $\Sigma_{lm}$  is direct-optimal (since there is a unique direct-optimal basis). First we prove that  $\Sigma_{lm}$  is direct, i.e. that for every  $A \subseteq S$ ,  $\varphi(A) = A \cup \{x \in S : \text{there exists } B \subseteq A \text{ with } B \rightarrow_{\Sigma_{lm}} x\}$ . This is obvious since one can take for  $B$  a basis of  $\varphi(A)$  such that  $B \subseteq A$ .

Now, let us prove that  $\Sigma_{lm}$  is direct-optimal. Consider a direct and equivalent UIS  $\Sigma$ . It is sufficient to prove that, when  $B \rightarrow x$  is a  $\Sigma_{lm}$ -implication, it is also a  $\Sigma$ -implication. Assume that it is not the case. Since  $B \rightarrow x$  is left-minimal,  $A \rightarrow x \notin \Sigma$  for every  $A \subset B$ . Therefore,  $x \notin \varphi(B) = B \cup \{x \in S : \text{there exists } A \subseteq B \text{ with } A \rightarrow_\Sigma x\}$ , a contradiction with  $\Sigma$  direct.

□

The above result justifies the following definition:

**Definition 16** *The unique basis obtained in Theorem 15 is called the canonical direct unit basis, and is denoted  $\Sigma_{cd}$ .*

Theorem 14 induces others nice characterizations of the canonical direct unit basis:



**Corollary 17** *Let  $\varphi$  be a closure operator. The canonical direct unit basis  $\Sigma_{cd}$  is the smallest basis of the set of all direct unit bases ordered by inclusion.*

**Corollary 18** *An UIS  $\Sigma$  is the canonical direct basis if and only if it satisfies the two following properties:*

- (1) *for every  $x \in S$ ,  $B \rightarrow_{\Sigma} x$  and  $B' \rightarrow_{\Sigma} x$ ,  $B$  and  $B'$  are incomparable.*
- (2) *the exchange condition (17).*

One can also observe that Corollary 17 is equivalent to the property of  $\Sigma_{cif}$  being iteration-free in a canonical way, introduced in [60].

One can observe that the first property in Corollary 18 can equivalently be reformulated using the terminology of a Sperner family like in [17]: for every  $x \in S$ , the set  $\mathcal{B}_x$  of all premisses of the  $\Sigma$ -implications  $B \rightarrow_{\Sigma} x$  forms a Sperner family.

The fact that  $\Sigma_{lm} = \Sigma_{weak}$  shows that the Sperner family  $\mathcal{B}_x$  is the family of blockades of  $x$ , i.e. the family of minimal transversals of the family  $\mathcal{D}_x = \{S \setminus (C + x) : C \text{ co-point of } x\}$ . We show now that the *necessary sets for  $x$* , and the  *$x$ -dominating sets* introduced in the literature are the same that the sets  $S \setminus (C + x)$ . Mannila and Raiha ([40,41]) define a *necessary set for  $x$*  as a minimal transversal of  $\mathcal{B}_x$ . On the other hand, one finds in Aizerman and Aleskerov's book on choice functions ([1]) the definition of an  *$x$ -dominating set* as a subset  $T$  of  $S$  such that  $x \in \mathbb{E}x_{\varphi}(S \setminus T)$  and  $x \notin \mathbb{E}x_{\varphi}(U)$  for every  $U$  satisfying  $S \setminus T \subset U$ .

**Corollary 19** *Let  $\varphi$  be a closure operator on  $S$ ,  $T \subseteq S$  and  $x \in S \setminus T$ . The three following conditions are equivalent:*

- (1)  *$T$  is a necessary set for  $x$ ,*
- (2) *there exists a co-point  $C$  of  $x$  such that  $T = S \setminus (C + x)$ ,*
- (3)  *$T$  is an  $x$ -dominating set.*

### Proof

1  $\Leftrightarrow$  2 Let us denote by  $\mathcal{M}_x$  the family of necessary sets for  $x$ . By definition,  $\mathcal{M}_x = Tr(\mathcal{B}_x)$ , the family of minimal transversals of  $\mathcal{B}_x$ . And, as said above,  $\mathcal{B}_x = Tr(\mathcal{D}_x)$  the family of minimal transversals of  $\mathcal{D}_x = \{S \setminus (C + x) : C \text{ co-point of } x\}$ . But, it is well known that, when  $\mathcal{F}$  is a Sperner family,  $Tr(Tr(\mathcal{F})) = \mathcal{F}$ . Therefore  $\mathcal{M}_x = Tr(\mathcal{B}_x) = Tr(Tr(\mathcal{D}_x)) = \mathcal{D}_x$ .

2  $\Rightarrow$  3 If  $T = S \setminus (C + x)$ , one has  $S \setminus T = C + x$ . Since  $C$  is a maximal set such that  $x \notin C$ ,  $x \in \mathbb{E}x(S \setminus T)$ , whereas if  $U \supset S \setminus T = C + x$ , then  $U \setminus x \supset C$  and  $x \notin \mathbb{E}x(U)$ .

3  $\Rightarrow$  2 Let  $T$  be an  $x$ -dominating set. So,  $x \in \mathbb{E}x_{\varphi}(S \setminus T)$ , i.e.  $\{x \in \varphi((S \setminus T) \setminus x)\}$ . Now, if  $U \in S \setminus T$ ,  $U \setminus x \in (S \setminus T) \setminus x$  and  $x \in \mathbb{E}x_{\varphi}(U)$  means that

$x \in \varphi(U \setminus x)$ . Thus  $(S \setminus T) \setminus x = (S \setminus T + x)$  is a maximal set such that  $x \in \varphi(S \setminus T + x)$ , i.e. a co-point  $C$  of  $x$ . Then  $T = S \setminus (C + x)$ , with  $C$  co-point of  $x$ .

□

One can notice that the equivalence between 2 and 3 was proved in [47] but only for closure operators satisfying the anti-exchange property.

## 5 The canonical direct unit basis and the Horn functions

It is well known that families of subsets of a set  $S$  are in a one-to-one correspondence with the Boolean functions defined on the Boolean algebra  $\mathcal{P}(S)$ . Indeed, one can associate to a family  $\mathcal{F}$  of subsets of  $S$  its characteristic function  $f_{\mathcal{F}}$  :

$$f_{\mathcal{F}}(M) = \begin{cases} 1 & \text{if } M \in \mathcal{F} \text{ with } M \subseteq S \\ 0 & \text{if not} \end{cases} \quad (18)$$

And conversely, one can associate to a Boolean function  $f$  from  $\mathcal{P}(S)$  to  $\{0, 1\}$  the following family of subsets of  $S$  called the *models* or the *true points* of  $f$ :

$$\mathcal{F}_f = \{M \subseteq S : f(M) = 1\} \quad (19)$$

Observe that the set of all Boolean functions ordered by  $f \leq g$  if  $\mathcal{F}_f \subseteq \mathcal{F}_g$  is itself a Boolean algebra.

By considering dually the *false points*, one can provide another one-to-one correspondence between families on  $S$  and Boolean functions on  $\mathcal{P}(S)$ . In the following, we will prefer this second correspondence that associates to a Boolean function  $h$  the family  $\mathcal{F}_h$  of its *false points* or its *counter-models*:

$$\mathcal{F}_h = \{M \subseteq S : h(M) = 0\} \quad (20)$$

Conversely, one can associate to a family  $\mathcal{F}$  on  $S$  the Boolean function  $h_{\mathcal{F}}$ :

$$h_{\mathcal{F}}(M) = \begin{cases} 0 & \text{if } M \in \mathcal{F} \text{ with } M \subseteq S \\ 1 & \text{if not} \end{cases} \quad (21)$$

A less known and still less used fact is that the closure systems on  $S$  are in a one-to-one correspondence with the Boolean functions called *pure (or definite)*

*Horn functions* (see historical notes for references). Then, any result on closure systems (or closure operators or implicational systems) can be translated into results about Horn functions, and conversely. In this section we are going to do this translation for the canonical direct basis.

In order to define Horn functions we will recall some classic definitions and facts. We denote by  $\mathcal{Q} = (0, 1, \vee, \wedge, ')$  the Boolean algebra on two elements 0 and 1, with the two Boolean operations  $\vee$  (called *disjunction or sum*) and  $\wedge$  (called *conjunction or product*), and the unary operation  $'$  (called *complementation*). A *Boolean function* of  $n$  (Boolean) variables is then a function on  $\mathcal{P}(S)$  to  $\mathcal{Q}$ , where  $S = \{x_1, x_2, \dots, x_n\}$  is the set of  $n$  Boolean variables. We denote by  $(x_1, x_2, \dots, x_n)$  a  $n$ -tuple of values 0 or 1 taken by these variables (then, note that the same symbol  $x_i$  can represent the variable  $x_i$  or the value 0 or 1 taken by this variable according as it belongs either to a set or to a  $n$ -uple). The set of these  $n$ -uples is in one-to-one correspondence with the set  $\mathcal{P}(S)$  of subsets of  $S$  (by the map  $(x_1, x_2, \dots, x_n) \rightarrow \{x_i \in S : x_i = 1\}$ ) where such a subset will also be called a *point*.

A variable  $x$  is called a *literal* whereas the complemented variable  $x'$  is called a *complemented literal*. A *conjunction* (respectively, a *disjunction*) of literals and complemented literals, where each variable, complemented or not, appears at most once is called a *term* (respectively, a *clause*). A conjunction like  $x \wedge y' \wedge z$  will be generally written more simply  $xy'z$ .

Let  $f$  be a Boolean function on  $2^n$  and  $(x_1, x_2, \dots, x_n)$  a *true  $n$ -tuple* of  $f$  (respectively, a *false  $n$ -tuple* of  $f$ ), i.e. a  $n$ -tuple such that  $f(x_1, x_2, \dots, x_n) = 1$  (respectively,  $f(x_1, x_2, \dots, x_n) = 0$ ). A *true point* (respectively *false point*) of  $f$  is a subset of  $S$  corresponding to a true  $n$ -tuple (respectively false  $n$ -tuple) of  $f$ . To a true (respectively false)  $n$ -tuple (or point) of  $f$  one associates the term  $\wedge \{x_i : x_i = 1\} \wedge \{x'_i : x_i = 0\}$  (respectively, the clause  $\vee \{x_i : x_i = 0\} \vee \{x'_i : x_i = 1\}$ ).

The sum (respectively, the product) of all these terms (respectively, clauses) constitutes the *canonical disjunctive normal form* (respectively, the *canonical conjunctive normal form*) denoted as the canonical DNF (respectively as the canonical CNF) of  $f$ . But, using the well known properties of a Boolean algebra (such that  $x = x \vee x = xx = x \vee (xy) = x(y \vee x)$ ,  $x \vee x' = 1$ ,  $xx' = 0$ ), it is possible to get many other disjunctive or conjunctive normal forms representing  $f$ . Any two such normal forms representing the same Boolean function are called *equivalent*.

**Example 20** For instance, let us denote more simply the set of  $n$  Boolean variables by  $\{1, 2, \dots, n\}$  and consider the Boolean function defined on  $\{1, 2, 3, 4, 5\}$  by its canonical DNF:

$$h = 1'2'3'4'5' \vee 12'3'4'5' \vee 1'23'4'5' \vee 1'2'34'5' \vee 1'2'3'45' \vee 123'4'5' \vee 12'34'5' \vee 1'2'3'45 \vee 1'2345' \vee 12345$$

Then, since:

$$1'2'3'4'5' \vee 12'3'4'5' \vee 1'23'4'5' \vee 123'4'5' = 3'4'5'$$

$$\text{and } 1'2'34'5' \vee 12'34'5' = 2'34'5'$$

an equivalent DNF for  $h$  is:

$$h = 3'4'5' \vee 2'34'5' \vee 1'2'3'45 \vee 1'2345' \vee 12345.$$

A classic problem (called the *Boolean function minimization problem*) is to find minimum DNF (or CNF) of a Boolean function, i.e. DNF (or CNF) using a minimum number of literals (other minimization problems using other criteria can be also considered). The first step for this research is to find the so-called prime implicants (respectively, prime implicates) of the Boolean function  $f$ . A *prime implicant* (respectively, a *prime implicate*) of  $f$  is a term  $t$  (respectively, a clause  $c$ ) such that (in the order between Boolean functions),  $t \leq f$  (respectively,  $f \leq c$ ) and is maximal (respectively, minimal) with this property. Indeed, a Boolean function  $f$  is always the sum of its prime implicants and the product of its prime implicates. But, it is generally possible to delete some implicants (respectively, implicates) in these expressions to get an equivalent more economical DNF or CNF. Then the second step consists in searching for the expressions that use the least number of implicants (respectively, implicates). For an arbitrary Boolean function, the search of all its prime implicants (respectively, implicates) is a NP-complete problem.

We now define the so-called *pure (or definite) Horn functions*. Since we will consider only such Boolean functions, we will henceforth omit the word pure. A term is called *Horn* if it contains exactly one complemented literal. For instance,  $34'5$  is a *Horn term*. A DNF is called Horn if all its terms are Horn. A Boolean function is called a *Horn function* if it can be represented by a Horn DNF. Now we have the following well known result (see Section 6.1):

**Theorem 21** *A Boolean function  $h$  of  $n$  variables  $x_1, x_2, \dots, x_n$  is a Horn function if and only if the set of its false points is a closure system on  $S = \{x_1, x_2, \dots, x_n\}$ .*

**Remark.** In the literature one also finds another definition of a Horn function. A clause is called Horn if it contains exactly one literal. For instance,  $1 \vee 2' \vee 4' \vee 5'$  is a *Horn clause*. A CNF is called Horn if all its clauses are Horn. A Boolean function is called a *Horn function* if it can be represented by a

Horn CNF. This definition is not equivalent to the previous one. In fact, a Boolean function  $f$  is a Horn function in this second sense if and only if the complementary function  $f'$  (in the Boolean algebra of all Boolean functions) is Horn in the first sense. With this second definition, one has: “a Boolean function is a Horn function if and only if the set of its true points is a closure system”.

Now we can state the relationship between the prime implicants of a Horn function  $h$  and the canonical direct implicational basis  $\Sigma_{cd}$  of its associated closure operator. It is known that the prime implicants of a Horn function are Horn terms, and so we can write  $Bx'$  such a prime implicant, where  $B$  is the subset of  $S$  corresponding to the literals of this prime implicant. For completeness we give the proof of the following known result (see, for instance, Theorem 4.1 in [36] where the result is proved with the  $\Sigma_{lm}$  version of the canonical unit direct basis).

**Proposition 22** *Let  $S = \{x_1, x_2, \dots, x_n\}$  be a set of elements, and:*

- $h$  be a Horn function of  $n$  variables on  $\mathcal{P}(S)$ ;
- $\mathcal{F}_h$  the closure system defined on  $S$  by the false points of  $h$ ;
- $\varphi_h$  the associated closure operator on  $S$ ;
- $\Sigma_{cd}$  the corresponding canonical direct implicational basis.

*Then  $Bx'$  is a prime implicant of  $h$  if and only if  $B \rightarrow x \in \Sigma_{cd}$ .*

**Proof** Let  $Bx'$  be a prime implicant of  $h$  and consider the implication  $B \rightarrow x$ . It belongs to  $\Sigma_\varphi$  since  $h(\varphi_h(B)) = 0$  implies  $Bx'(\varphi_h(B)) = 0$  and so  $x \in \varphi_h(B)$ . Let  $A \subset B$ . Since  $Ax'$  is not an implicant of  $h$ , there exists  $X \subseteq S$  such that  $Ax'(X) = 1$  and  $h(X) = 0$ . Then,  $A \subseteq X \subseteq S \setminus x$  and  $X \in \mathcal{F}_h$  which means that  $x \notin \varphi_h(A)$ . So,  $A \rightarrow x \notin \Sigma_\varphi$  and  $B \rightarrow x \in \Sigma_{cd}$ .

Conversely, let  $B \rightarrow x \in \Sigma_{cd}$  and consider the Boolean term  $Bx'$ . For  $X \subseteq S$ , we have  $Bx'(X) = 1$  if and only if  $B \subseteq X$  and  $x \notin X$ . Then  $X \in \mathcal{F}_h$  and  $h(X) = 1$ , what shows  $Bx' \leq h$ . Moreover,  $B \leq h$  since  $B(\varphi_h(B)) = 1$  and  $h(\varphi_h(B)) = 0$ . Similarly, if  $A \subset B$ ,  $Ax' \not\leq h$ , since  $Ax'(\varphi_h(A)) = 1$  and  $h(\varphi_h(A)) = 0$ . Then  $Bx'$  is a prime implicant of  $h$ .  $\square$

**Corollary 23** *There is a one-to-one map between the set of prime implicants of a Horn function and the set of implications in the canonical direct unit basis of the closure operator corresponding to the Horn function.*

**Remark.** When one considers the definition of a Horn function mentioned in the remark following Theorem 21, one gets a one-to-one map between the set of prime implicants of the Horn function and the set of implications in the

canonical direct unit basis of the corresponding closure operator.

**Example 24** *In our example (Example 1), consider its closure system  $\mathbb{F}$  defined on  $S = \{1, 2, 3, 4, 5\}$  and its canonical direct UIS  $\Sigma_{cd}$  (equal to  $\Sigma_{do}$  given in Example 5). By Proposition 22,  $\mathbb{F}$  is the closure system given by the false points of the following Horn function whose prime implicants are deduced from  $\Sigma_{cd}$ :*

$$h = 54' \vee 234' \vee 243' \vee 342' \vee 142' \vee 143' \vee 145' \vee \\ 251' \vee 351' \vee 152' \vee 352' \vee 153' \vee 253' \vee 1235'$$

*For instance, one can verify that  $12 \in \mathbb{F}$  is equivalent to  $h(12) = 0$ ; and  $14 \notin \mathbb{F}$  is equivalent to  $h(14) = 1$ .*

## 6 Notes

### 6.1 Historical note

We try to give the origins of some notions and results used in this paper. It is well known that the notion of a binary relation on a set arose from works of De Morgan and Peirce in the second half of the 19th century. But it seems to be more difficult to know who introduced for the first time the notion of binary relation between subsets and elements of a set or used for the first time the notion of a binary relation on the power set of a set. It is clear that such relations can be used in many different contexts. For instance, a binary relation between subsets and elements of a set appears in Hertz's 1927 paper ([33]) where it formalizes a *consequence* relation, and a relation between elements and subsets of a set appears in Appert's paper ([2]) where a "contiguity" relation allowing to define a topological space is formalized.

Birkhoff ([9]) dates back the origin of the notions of closure systems and closure operators to Moore's 1909 and 1910 papers ([48,49]). Indeed, in his 1909 paper Moore, speaking in terms of a property of a class of functions, writes: "let a property satisfied by the class (of all functions) and by the greatest common subclass of subclasses satisfying it. Then this property is extensionally attainable in the sense that for every subclass  $S$  there exists a least extensive class containing  $S$ , given by the intersection of all subclasses containing  $S$ ". But it is probable that Moore's observations about the equivalence of these two notions would have been forgotten if these two concepts, under various names and in a more or less general way, had not played a significant role in the birth of the general topology as an axiomatic theory, in the beginning of the

last century. Many mathematicians (Alexander, Alexandroff, Frechet, Hausdorff, Kuratowski, Riesz, Sierpinski, Siskorski, Monteiro, Ribero, Appert, etc.) contributed to this creation, using systems of axioms based on several different primitive notions such as derivation, neighborhood, surrounding, closed or open sets, closure or interior operators. The notion of closure operator was also used in logics as early as in Tarski's 1929 and 1930's papers ([57,58]) where he defines the consequence relation of a logical deductive system as a closure operator on an infinite set  $S$  satisfying a *finitary* axiom (see Martin and Pollard's 1996 book ([42]) for the use of closure operators in logics). Also observe that there are many notions equivalent to the notion of closure operator (see [45]) and in particular that the theory of closure systems is closely related to lattice theory since every (finite) lattice can be represented by a closure system. One can date back the notion of Boolean (or truth) function to Boole (in his theory of *elective* functions). The definition of a Horn Boolean function as a Boolean function having a Horn (disjunctive normal) form appears for the first time (according to the authors) in Hammer and Kogan's 1992 paper ([30]). But the notion and name of *Horn clause* come from the logician Alfred Horn who first pointed out the significance of such clauses in his 1951 paper "On sentences which are true of direct unions of algebras" ([35]). This attribution is sometimes contested. For instance Hodges ([34]) writes: "Horn clause logic is a part of first-order logic. It was first isolated by J.C.C. McKinsey ([43]). The name 'Horn' is a historical accident. After McKinsey's paper in 1943, Alfred Tarski suggested investigating a more general class of sentences that are like Horn clauses except that they have arbitrarily many existential and universal quantifiers at the beginning. The sentences that Tarski described are now known as Horn sentences, because Tarski's colleague Alfred Horn ([35]) responded to Tarski's suggestion by showing that one of McKinsey's theorems is true for them too. This work of Horn is important in its own right, but it is not directly relevant to Horn clauses. (Henschen ([32]) p. 820 explains the name 'Horn clause' by a result of Horn ([35]) on Horn clauses; but the result is false, and it is not in [35])." On the other hand, Dechter and Pearl ([16]) write that the equivalence between Horn functions and families of subsets closed by intersection *appears to be a general folklore among many researchers, although we could not trace its precise origin.* But, in fact, Horn's 1951 paper ([35]) deals with Horn terms (i.e. propositional terms containing at most one complemented literal) and its Lemma 7 amounts exactly saying that a Boolean function  $h$  is Horn (in the sense that it admits a Horn DNF) if and only if the family of its false points is closed by intersection. Let us also observe that McKinsey ([43]) uses *sentences* like  $\bigwedge_{i \in I} \varepsilon_i \rightarrow \varepsilon$ , but where the  $\varepsilon$ 's are much more general terms than simple propositional variables. So, finally, the names Horn term and Horn (Boolean) function seem quite justified. It is interesting to mention that the equivalence between the Horn Boolean functions and the families of sets closed by intersection was also shown in an *applied* context. In his *Analyse booléenne des questionnaires* ([21]) Flament seeks to find systems of *implications* between dichotomous questions explaining the answers

of subjects to queries (a well-known example of such a system is the so-called Guttman's scale). Then, he associates with these answers a family of sets (the set of questions receiving a *positive* answer) and the corresponding Boolean function. And he writes (page 198): "le protocole est fermé pour l'intersection si et seulement si aucune des PCU ne comporte plus d'une réponse négative" (i.e., the family of sets associated to the positive answers of the subjects is closed for intersection if and only if the prime implicants of the corresponding Boolean function contain at most one negative answer).

It is apparently Armstrong ([4]) who in the context of relational data bases has shown for the first time the one-to-one correspondence between the full family of functional dependencies (called here full implicational systems) and closure systems (Armstrong called the closed sets saturated sets). But one already finds a one-to-one correspondence between the so-called "transitive topologies" and the closure operator in Appert's paper quoted above ([2], see also [3]). And the transitive topologies are nothing else than the binary relations between elements and subsets of a set which are the dual of the full unit implicational systems. These same correspondences have been rediscovered and/or generalized many times under various formulations. For instance, they appear in Buchi's book ([11]) where this author uses *dependence relations*, and in Doignon and Falmagne's book ([18]) between what they call *entailment relations* and the families of sets closed by unions (see also below).

One can ask what the link is between our implicational systems and logical systems ? First one can present the notions and results about implicational systems in the framework of propositional logic ([23]). More deeply, Fagin displays an equivalence between the functional dependencies of relational databases (our implications) and the *implicational statements* of propositional logic ([20]). An implicational statement of propositional logic is a conjunction of propositional (Boolean) variables implying a conjunction of propositional variables. Then Fagin proves that a functional dependency is a consequence of a set of functional statements if and only if the corresponding implicational statement is a consequence of the corresponding set of implicational statements. On the other hand there are formal links between implicational systems and the ways to formalize the notion of logical consequence (see Scott 1974 [54] for an overview). As already mentioned, Hertz ([33]) (respectively, Tarki) used a binary relation between subsets and elements of a set of sentences (respectively, a closure operator) to formalize a notion of consequence. The connection between the two presentations is the same as the one used in this paper between an implicational system and a closure operator:  $X \rightarrow y$  iff  $y \in \varphi(X)$ . Later, Gentzen in [27] introduced a relation where the right-hand side of the relation is a disjunction of sentences. Then in 1982 ([55]) Scott introduced the notion of *information systems* where there is an *entailment relation* between *consistent* subsets and elements of a set. And, later, a one-to-one correspondence between Scott's information systems and *algebraic*



$\cap$ -structures has been displayed (see [15]). In the finite case, this correspondence is exactly the correspondence between the full implicational systems and the closure systems.

Finally we point out the attempts to extend the *contextual attribute logic* of the Formal Concept Analysis to a *contextual Boolean judgment logic* by introducing formal negations and oppositions ([26,14]).

## 6.2 Algorithmical note

Closure systems appear in many areas where we need efficient algorithms to handle them. So, in these areas, a number of (generally independent) works have been made to address the many algorithmical problem raised. In particular, since these systems may have several “representations”, a general problem is to provide algorithms to go from a representation taken as input to another taken as output. We will return later to the notion of representation, but we consider first the two following important transformations problems:

- **Generation of the canonical direct unit basis**  $\Sigma_{cd}$  with an equivalent UIS  $\Sigma$  as input.
- **Generation of the closure operator  $\varphi$  or/and of the family  $\mathbb{F}$  of closed sets** with either an UIS  $\Sigma$  or the basis  $\Sigma_{cd}$  as input.

Since  $\mathbb{F}$  and  $\Sigma_{cd}$  are bounded by  $2^{|S|}$  in the worst case, and by 1 in the best case, with a reasonable size in practice, these problems belong to the more general class of problems having an input of size  $n$ , and an output of size  $N$  bounded by  $2^n$ . For this class of problems, a classical worst-case analysis makes them exponential, thus NP-complete in time and in space. However, a more precise information can be obtained by output-sensitive analysis techniques (see a survey in [50]). These analyses are relevant since the recent improvements storage and treatment capacity increasingly often allow us to handle some exponential data, what was not possible even some time ago.

Concerning the time-analysis, the idea is to consider the time complexity needed to generate only one element of the output (i.e. one implication or one closed set in our case). Time-complexity per element of the output can be computed using two main analysis techniques. As a first technique, the *amortized complexity* computes the “average cost” per element. It consists in extracting the *amortized cost*  $c$  per element from the general classical time complexity  $O(cN)$  where  $N$  is the size of the output. When  $c$  can be bounded by a polynomial, we speak of a *polynomial amortized time algorithm*. The second technique is the *delay complexity*: it considers the output as a sequence of generated elements. It then consists in more accurately computing the time between the generation of two consecutive elements called the *delay cost*. A

*polynomial delay algorithm* is then an algorithm with a delay cost bounded by a polynomial.

Concerning the space-analysis, one can introduce the *storage requirement* which is satisfied when the output has to be kept in the memory. Hence, one can distinguish between *counting algorithms* (output is only counted, and not stored), *generation algorithms* (output is generated, sometimes it has to be stored) and *construction algorithms* (output is generated and stored).

Consider as an example the special case where  $\mathbb{F}$  has an exponential size ( $2^{|S|}$ ) when it contains all the subsets of  $S$ , thus  $(\mathbb{F}, \subseteq)$  is a boolean lattice. The generation of  $\mathbb{F}$  then consists in generating all subsets of  $S$ , that can be performed in  $O(1)$  per element using algorithms of constant amortized time or of constant delay. Moreover, since storage is not required by these algorithms, they have a non exponential space complexity.

**6.2.0.1 Generation of  $\Sigma_{cd}$ .** Since  $\Sigma_{cd}$  has an exponential size in the worst-case, any generation algorithm has to be analyzed by considering the time-complexity per implication. Currently, there exists no algorithm with a polynomial generation per implication, using the storage requirement. Moreover, the existence of such a polynomial algorithm is still an open problem. Wild in [60] provides an algorithm with an IS  $\Sigma$  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. Let us also mention in the area of data-mining the algorithm of Taouil and Bastide in [56] where the left-minimal implications are called *proper implications*. It has the same exponential time and space complexity per implication. Bertet and Nebut's algorithm in [8], described by Definition 4, also generates an intermediate and exponential but direct UIS  $\Sigma_d$  (rule F7) before minimizing it (rule F3), and thus computes  $\Sigma_{cd}$  in  $O(|S||\Sigma_d|^2)$ . An incremental generation algorithm has been proposed in [6] where each implication  $B_i \rightarrow x_i$  is incrementally added in the canonical direct basis issued from  $\{B_j \rightarrow x_j; j < i\}$  in order to limit the number of intermediate implications that have to be generated to obtain the direct property, before the minimalisation treatment. This new algorithm keeps an exponential worst case complexity, but an experimental study indicates very significant improvements compared to existing algorithms.

**6.2.0.2 Generation of  $\varphi$  and  $\mathbb{F}$ .** In [40], Mannila and Rähkä propose the generation of a closure  $\varphi(X)$  (algorithm *Linclosure*) in  $O(|S|^2|\Sigma|)$ , with a given  $\Sigma$  as input. This algorithm iteratively scans implications of an UIS  $\Sigma$ . The computation cost depends on the minimal number of implications when using the Duquenne-Guigues canonical basis [28]  $\Sigma_{can}$  of where unit implications with the same premise are merged by  $X \rightarrow Y = \{y \in S : X \rightarrow$

$y\}$ . The computation cost also depends on the number of iterations, in any case bounded by  $|S|$ . In order to practically limit this number while keeping the same complexity, Wild in [60] modifies this algorithm using additional and sophisticated data structures. It is worth noticing that for direct UIS, and thus with  $\Sigma_{cd}$ , the computation of  $\varphi(X)$  requires only one iteration. Therefore, using  $\Sigma_{cd}$ , a closure  $\varphi(X)$  is obtained by Bertet and Nebut in [8] in  $O(|X||\Sigma_{cd}|)$  (when expressed with respect to  $X$ ) or in  $O(s(\Sigma_{cd}))$  (when expressed with respect to  $\Sigma_{cd}$ ). They also propose in [8] an algorithm to generate the family  $\mathbb{F}$  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  $c_\varphi$  is the cost of generating one closure  $\varphi(X)$ . Therefore, their algorithm is in  $O(|S|^3)$  using a given  $\Sigma$  as input, and an improvement in  $O(|S|^2 + |S|s(\Sigma_{cd}))$  is obtained using the canonical direct basis  $\Sigma_{cd}$  as input. This improvement is due to the direct property of  $\Sigma_{cd}$  and can be applied to every algorithm that uses the computation of a closure  $\varphi(X)$  as a basic step.

**6.2.0.3 Other representations of  $\mathbb{F}$ .** To efficiently handle a closure system  $\mathbb{F}$  (or its associated closure operator  $\varphi$ ), one can consider any implicational system  $\Sigma$  generating it as a representation of  $\mathbb{F}$ . However, the full unit implicational system  $\Sigma_\varphi$  (given by Formula (8)) would not be an efficient representation of  $\mathbb{F}$  for an algorithmical use. The efficiency of a representation can be described by some simple properties: a representation of a closure system  $\mathbb{F}$  is efficient when it is small, readily identifiable, and when it uniquely determines  $\mathbb{F}$  via simple and efficient generation algorithm. This paper focuses on the representation of  $\mathbb{F}$  by the canonical direct unit basis. However, one can find many other (but not always efficient) representations of a closure system  $\mathbb{F}$ : representation by the Duquenne-Guigues canonical basis in data analysis ([28]) ; representation by Horn functions in logical programming ([22]) ; representation by a poset of irreducibles in lattice theory ([50]) ; representation by a table called a *context* in formal concept analysis ([26]) and data-mining.

Therefore, the generation of  $\mathbb{F}$  can also be considered with various representations as input. The well-known algorithm generating  $\mathbb{F}$  is the *Next-closure* algorithm due to Ganter ([23]) in the context of formal concept analysis ([26]). It accepts a table, and more generally a closure operator as input, and has a polynomial space-complexity (since the closed sets do not have to be stored) and a time complexity in  $O(|S|^3)$  per element. It generates closed sets according to a total order on all the closed sets (extending the inclusion order) called the *lectic order*. One can find various algorithms generating  $\mathbb{F}$  using different representations, 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 uses a poset or a table as representation and is due to Nourine and Raynaud in [51]. 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 (so it satisfies the storage requirement). Let us also mention an attribute-incremental algorithm generating the Duquenne-Guigues canonical basis from a context ([52]).

Another important point in this paper concerns the links between different representations of  $\mathbb{F}$ . For instance, in [36], one finds the correspondence between left minimal implications (called there the minimal functional dependencies) and prime implicates (Corollary 23). From an algorithmical point of view, let us mention the algorithm of Mannila in [40], where  $\Sigma_{cd}$  is generated with the irreducibles elements of  $\mathbb{F}$  as input. It has an exponential time per implication in the worst case, and is based on the generation of all minimal transversals, a problem known to be an open problem (i.e. there actually only exists an exponential algorithm to solve it). Stemming from the links between UIS and Horn functions, let us also mention the algorithm with the best known complexity that is due to Fredman and Khachiyan ([22]) with a DNF as input. It generates one implication in  $O(|S|^{\log |S|})$ , i.e a quasi-polynomial time and has recently been modified in [38] to solve this problem with a first step in deterministic polynomial time, following by  $O(\log^2 |S|)$  non deterministic steps. Although it is computationally difficult to compute  $\Sigma_{cd}$  with a general DNF as input, one can compute a  $\Sigma_{cd}$ -implication in polynomial time with a Horn DNF or the family  $\mathbb{F}$  as input, using an algorithm due to Ibaraki et al. ([36]). Bertet et al. in [7] propose a joint use of the two bases that are the Duquenne-Guigues canonical basis  $\Sigma_{can}$  ([28]) and the direct canonical basis for classification of images of symbols [29] in data analysis (the Duquenne-Guigues canonical basis offers the advantage of a minimal and condensed representation of data ; the canonical direct basis allows efficient algorithm treatments). However, we will conclude this algorithmical note by observing there is a need for more general studies on the links between these two canonical basis, the direct one and the non-direct one.

## 7 Conclusion

Since equivalent notions such as closure systems (or systems of sets closed by union), closure operators (or dual closure operators), full systems of implications (or of dependencies), (pure) Horn functions have been studied by different authors in different domains (topology, lattice theory, hypergraph theory, choice functions, relational data bases, data mining and concept analysis, artificial intelligence and expert systems, knowledge spaces, logic and logic programming, theorem proving...), it is not surprising that one finds the same notions, results or algorithms under various names. For instance, in AI the meet-irreducible elements of a lattice of closed sets are called its characteristic sets, the associated closure operator is called the forward chaining

procedure. In the context of Horn functions a directed graph introduced as earliest in 1987 by different authors on the set of the Boolean variables plays an important role. It can be shown that in the case of pure Horn function, the relation defined by this graph is the inverse of the dependency relation defined in section 3.2 (it is also the domination relation defined in [5]). On the other side, one can also find many original results or algorithms but which are generally known only in a specific domain. It would be very profitable to increase (or create) the communications between the various domains that use the same (or equivalent) notions and tools. Our paper is a first step in this direction and we intend to take further steps.

We also intend to work on the relationship between the canonical direct unit implicational basis, and the Duquenne-Guigues canonical basis mentioned in the introduction. Recall that this basis is an implicational system (IS for short) i.e. a binary relation on  $\mathcal{P}(S)$  and that one can associate to it (as to any IS) an equivalent UIS by replacing each implication  $A \rightarrow B$  by the set of implications  $\{A \rightarrow b : b \in B\}$ . We denote  $\Sigma_{can}$  the UIS deduced of the Duquenne-Guigues basis by applying this rule. Consider in our example the two bases  $\Sigma_{can}$  (the UIS deduced from the canonical basis) and  $\Sigma_{cd}$  (the canonical direct unit basis):

$$\Sigma_{can} = \left\{ \begin{array}{l} (1) 5 \rightarrow 4 \quad (2) 23 \rightarrow 4 \quad (3) 24 \rightarrow 3 \quad (4) 34 \rightarrow 2 \\ (5) 14 \rightarrow 2 \quad (6) 14 \rightarrow 3 \quad (7) 14 \rightarrow 5 \quad (8) 2345 \rightarrow 1 \end{array} \right.$$

$$\Sigma_{cd} = \left\{ \begin{array}{l} (1) 5 \rightarrow 4 \quad (2) 23 \rightarrow 4 \quad (3) 24 \rightarrow 3 \quad (4) 34 \rightarrow 2 \\ (5) 14 \rightarrow 2 \quad (6) 14 \rightarrow 3 \quad (7) 14 \rightarrow 5 \quad (8) 25 \rightarrow 1 \\ (9) 35 \rightarrow 1 \quad (10) 15 \rightarrow 2 \quad (11) 35 \rightarrow 2 \quad (12) 15 \rightarrow 3 \\ (13) 25 \rightarrow 3 \quad (14) 123 \rightarrow 5 \end{array} \right.$$

Remark that  $\Sigma_{can}$  is a proper UIS since for every implication the conclusion is not included in the premise. Remark also that  $\Sigma_{can} \not\subseteq \Sigma_{free}$  (Example 2) since the  $\Sigma_{can}$ -implication (8) does not belong to  $\Sigma_{free}$ . One can also verify that  $\Sigma_{can}$  is not direct, by considering the  $\varphi_{\Sigma}$ -closure of 15:  $\pi_{\Sigma}(15) = 15 + 4$  by applying  $\Sigma_{can}$ -implication (1) and  $\pi_{\Sigma}^2(15) = (15 + 4) + 2 + 3$  by applying  $\Sigma_{can}$ -implications (5) and (6). Therefore  $\varphi_{\Sigma}(15) \neq \pi_{\Sigma}(15)$ .

We conclude that this paper is contradicting a conjecture of the literature (in [37]). Indeed, one observes that the premise of implication (10) of  $\Sigma_{cd}$  is not contained in a premise of any implication of  $\Sigma_{can}$ .

**Acknowledgements.** We would like to thank the referees who reviewed our manuscript for their helpful comments. We also thank Emily Tanimura and

Claude Del Vigna for their help on form and substance.

## References

- [1] M.A. Aizerman and F.T. Aleskerov. *Theory of choice*. Elsevier Science B.V, North Holland, 1995.
- [2] A. Appert. Sur les topologies transitives. *Bulletin de l'académie royale de Belgique (Classe des Sciences)*, 23(2):135–142, 1937.
- [3] A. Appert and Ky-Fan. *Espaces topologiques intermédiaires. Problème de la distanciation*. M. Hermann, 1951.
- [4] W.W. Armstrong. Dependency structures of database relationships. *Information Processing*, 74:580–583, 1974.
- [5] A. Berry and E. San Juan. Generalized domination in closure systems. *Discrete Applied Mathematics*, 154(7):1064–1084, 2006.
- [6] K. Bertet. Some algorithmical aspects using the canonical direct implicational basis. In *Fourth International Conference on Concept Lattice and their Applications (CLA'06)*, pages 101–114, 2006.
- [7] K. Bertet and S. Guillas and J-M. Ogier. On the joint use of two implicational bases. In *Fifth International Conference on Formal Concept Analysis (ICFCA'07)*, pages 23–39, Clermont-Ferrand, France, February 12-16 2007.
- [8] K. Bertet and M. Nebut. Efficient algorithms on the Moore family associated to an implicational system. *DMTCS*, 6(2):315–338, 2004.
- [9] G. Birkhoff. *Lattice theory*. American Mathematical Society, 1st edition, 1940.
- [10] G. Birkhoff and O. Frink. Representations of lattices by sets. *Transactions of the American Mathematical Society*, 64:299–316, 1948.
- [11] J.R. Buchi. *Finite Automatas, their Algebras and Grammars. Toward a theory of formal expressions*. Springer Verlag, Berlin, 1989.
- [12] 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.
- [13] E.F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [14] F. Dau and J. Klinger. From formal concept analysis to contextual logic. In Berlin / Heidelberg Springer, editor, *Formal Concept Analysis, Foundations and Applications, Lecture Notes in Computer Science*, volume 3626, pages 81–199, 2005.

- [15] B.A. Davey and H.A. Priestley. *Introduction to lattices and orders*. Cambridge University Press, 2nd edition, 1991.
- [16] R. Dechter and J. Pearl. Structure identification in relational data. *Artificial Intelligence*, 58:237–270, 1992.
- [17] J. Demetrovics and H. Nam-Son. Databases, closure operations and Sperner families. In P. Frankl et al., editor, *Extremal Problems for Finite Sets*, pages 199–203, New Haven, 1991. János Bolyai Mathematical Society.
- [18] J.P. Doignon and J.C. Falmagne. *Knowledge spaces*. Springer Verlag, Berlin, 1999.
- [19] V. Duquenne. Latticial structures in data analysis. *Theoretical Computer Science*, 217(2):407–436, 1999.
- [20] R. Fagin. Functional dependencies in a relational database and propositional logic. *IBM J. of Research and Development*, 21(6):534–544, 1977.
- [21] C. Flament. *L'analyse booléenne des questionnaires*. Mouton, Paris, 1976.
- [22] M.L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
- [23] B. Ganter. Two basic algorithms in concept lattices. Technical report, Technische Hochschule Darmstadt, 1984.
- [24] B. Ganter. Attribute exploration with background knowledge. *Theoretical Computer Science*, 217:215–233, 1999.
- [25] B. Ganter and R. Wille. Contextual attribute logic. In W.M. Tepfenhart and W. Cyre, editors, *Proceedings of the 7th International Conference on Conceptual Structures: Standards and Practices, Lecture Notes In Artificial Intelligence*, volume 1640, 1999.
- [26] B. Ganter and R. Wille. *Formal concept analysis, Mathematical foundations*. Springer Verlag, Berlin, 1999.
- [27] G. Gentzen. Investigations into logical deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, North-Holland, 1969.
- [28] 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.
- [29] S. Guillas, K. Bertet, and J.M. Ogier. Graphic recognition : the concept lattice approach. *Lecture Notes in Computer Science, Graphic Recognition: Recent Advances and Perspectives*, 3088, 2003.
- [30] P.L. Hammer and A. Kogan. Horn's functions and their dnfs. *Information Processing Letters*, 44:23–29, 1992.
- [31] D. Hand, H. Mannila, and P. Smyth. *Principles of Data-Mining*. MIT Press, 2001.

- [32] L.J. Henschen. Semantic resolution for Horn sets. *IEEE Transactions on Computers*, 25:816–822, 1976.
- [33] P. Hertz. Über axiomensysteme für beliebige satzsysteme. *Mathematische Annalen*, 101:457–514, 1927.
- [34] W. Hodges. Logical features of Horn clauses. In M. Gabbay, C.J. Hogger, J.A. Robinson, and J. Siekmann, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Logical Foundations*, volume 1, pages 449–503, 1993.
- [35] A. Horn. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, 16:14–21, 1951.
- [36] T. Ibaraki, A. Kogan, and K. Makino. Functional dependencies in Horn theories. *Artificial Intelligence*, 108:1–30, 1999.
- [37] K. Ben-Khalifa and S. Montameny. Horn-representation of a concept lattice. In J. Diatta, P. Eklun, and M. Liquière, editors, *Fifth International Conference on Concept Lattices and their Applications (CLA'2007)*, pages 181–191, Montpellier, France, October 24-26 2007.
- [38] D.J. Kavvadias and E.C. Stravropoulos. Monotone boolean dualization is in  $\text{co-np}[\log^2 n]$ . *Information Processing Letter*, 85:1–6, 2003.
- [39] D. Maier. *The Theory of Relational Databases*. Computer Sciences Press, 1983.
- [40] H. Mannila and K.J. Rähä. *The design of relational databases*. Addison-Wesley, 1992.
- [41] H. Mannila and K.J. Rähä. Algorithms for inferring functionnal dependencies from relations. *Data & Knowledge Engineering*, 12(1):83–99, 1994.
- [42] N.M. Martin and S. Pollard. Closure spaces and logic. *Kluwer Academic Publisher, Dordrecht*, 1996.
- [43] J.C.C. McKinsey. The decision problem for some classes of sentences without quantifiers. *Journal of Symbolic Logic*, 8:61–76, 1943.
- [44] B. Monjardet. Arrowian characterizations of latticial federation consensus functions. *Mathematical Social Sciences*, 20(1):51–71, 1990.
- [45] B. Monjardet. The presence of lattice theory in discrete problems of mathematical social sciences. Why ? *Mathematical Social Sciences*, 46(2):103–144, 2003.
- [46] B. Monjardet and N. Caspard. On a dependance relation in finite lattices. *Discrete Mathematics*, 165:497–505, 1997.
- [47] B. Monjardet and V. Raderinirina. The duality between the anti-exchange closure operators and the path independent choice operators on a finite set. *Mathematical Social Sciences*, 41(2):131–150, 2001.
- [48] E.H. Moore. On a form of general analysis with applications to linear differential and integral equations. In *Atti del IV Congr. Int. dei Mat. II*, pages 98–114, Roma, 1909.



- [49] E.H. Moore. *Introduction to a form of general analysis*. Yale University Press, New Haven, 1910.
- [50] L. Nourine. *Une structuration algorithmique de la théorie des treillis*. PhD thesis, Université of Montpellier I, July 2000.
- [51] L. Nourine and O. Raynaud. A fast algorithm for building lattices. *Information Processing Letters*, 71:199–204, 1999.
- [52] S. Obiedkov and V. Duquenne. Attribute-incremental construction of the canonical implication basis. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):77–99, 2007.
- [53] 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.
- [54] D. Scott. Completeness and axiomatizability in many-valued logic. In *Proceedings of the Tarski Symposium, American Mathematical Society*, pages 411–435, 1974.
- [55] D.S. Scott. Domains for denotational semantics. In *Automata, Languages and Programming*, volume 140, pages 577–613, Berlin, 1982.
- [56] R. Taouil and Y. Bastide. Computing proper implications. In *9<sup>th</sup> International Conference on Conceptual Structures*, Stanford, USA, 2002.
- [57] A. Tarski. Remarques sur les notions fondamentales de la méthodologie des mathématiques. *Annales de la Société Polonaise de Mathématiques*, 7:270–272, 1929.
- [58] A. Tarski. Fundamentale begriffe der methodologie der deductiven wissenschaften. *Monatshelpte Mathematik Physik*, 37:361–404, 1930.
- [59] M. Wild. A theory of finite closure spaces based on implications. *Advances in Mathematics*, 108(1):118–139, 1994.
- [60] 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.