Formal Concept Analysis with Constraints by Closure Operators*

Radim Bělohlávek and Vilém Vychodil

Department of Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic
Phone: +420 585 634 700, Fax: +420 585 411 643
{radim.belohlavek, vilem.vychodil}@upol.cz

Abstract. The paper presents a general method of imposing constraints in formal concept analysis of tabular data describing objects and their attributes. The constraints represent a user-defined requirements which are supplied along with the input data table. The main effect is to filter-out outputs of the analysis (conceptual clusters and if-then rules) which are not compatible with the constraint, in a computationally efficient way (polynomial time delay algorithm without the need to compute all outputs). Our approach covers several examples studied before, e.g. extraction of closed frequent itemsets in generation of non-redundant association rules. We present motivations, foundations, and examples.

1 Introduction and Motivation

Formal concept analysis (FCA) is a method of data analysis and visualization which deals with input data in the form of a table describing objects (rows), their attributes (columns), and their relationship (table entries $\times$'s and blanks indicate whether or not object has attribute) [4, 6]. Basic outputs of FCA are the following: First, a collection of maximal rectangles of the table which are full of $\times$'s. These rectangles are interpreted as concept-clusters (so-called formal concepts), can be hierarchically ordered and form a so-called concept lattice. Second, a (non-redundant) set of if-then rules describing attribute dependencies (so-called attribute implications). FCA proved to be useful in several fields either as a direct method of data analysis, see e.g. [4], the references therein and also [5, 11], or as a preprocessing method, see e.g. [12]. In the basic setting, it is assumed that no further information is supplied at the input except for the data table. However, it is often the case that there is an additional information available in the form of a constraint (requirement) specified by a user. In such a case, one is not interested in all the outputs (maximal full rectangles or if-then rules) but only in those which satisfy the constraint. The other outputs may be left out as non-interesting. This way, the number of outputs is reduced by

* Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

H. Schärfe, P. Hitzler, and P. Øhrstrøm (Eds.): ICCS 2006, LNAI 4068, pp. 131–143, 2006.

focusing on the "interesting ones". Needless to say, the general idea of constraints is not new. A reader can find examples of using constraints in data mining in [3].

In this paper, we develop a method of constraints in FCA which are expressed by means of closure operators. The constraints can be used both for constraining maximal rectangles and if-then rules. Our approach is theoretically and computationally tractable and covers several interesting forms of constraints. For instance, one can set the closure operator in such a way that the maximal full rectangles satisfying the constraint correspond exactly to closed frequent itemsets [10], used e.g. in generating non-redundant association rules [12], see also Section 4. As another example, one can set the closure operator in such a way that at the output one gets exactly the formal concepts respecting a given hierarchy of attributes (a user tells some attributes are more important than others), see [1].

In Section 2, we present preliminaries from FCA. Section 3 presents our approach, theoretical foundations, and algorithms. In Section 4, we present several examples of constraints by closure operators and demonstrating examples. Section 5 is a summary and an outline of future research.

2 Preliminaries

In what follows, we summarize basic notions of FCA. An object-attribute data table describing which objects have which attributes can be identified with a triplet $\langle X, Y, I\rangle$ where X is a non-empty set (of objects), Y is a non-empty set (of attributes), and $I \subseteq X \times Y$ is an (object-attribute) relation. Objects and attributes correspond to table rows and columns, respectively, and $\langle x, y\rangle \in I$ indicates that object x has attribute y (table entry corresponding to row x and column y contains $\times$; if $\langle x, y\rangle \notin I$ the table entry contains blank symbol). In the terminology of FCA, a triplet $\langle X, Y, I\rangle$ is called a formal context. For each $A \subseteq X$ and $B \subseteq Y$ denote by $A^{\uparrow}$ a subset of Y and by $B^{\downarrow}$ a subset of X defined by

$$A^{\uparrow} = \{y \in Y \mid \text{for each } x \in A:\ \langle x, y\rangle \in I\},$$
$$B^{\downarrow} = \{x \in X \mid \text{for each } y \in B:\ \langle x, y\rangle \in I\}.$$

That is, $A^{\uparrow}$ is the set of all attributes from Y shared by all objects from A (and similarly for $B^{\downarrow}$). A formal concept in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ of $A \subseteq X$ and $B \subseteq Y$ satisfying $A^{\uparrow} = B$ and $B^{\downarrow} = A$. That is, a formal concept consists of a set A (so-called extent) of objects which fall under the concept and a set B (so-called intent) of attributes which fall under the concept such that A is the set of all objects sharing all attributes from B and, conversely, B is the collection of all attributes from Y shared by all objects from A. Alternatively, formal concepts can be defined as maximal rectangles of $\langle X, Y, I\rangle$ which are full of $\times$'s: For $A \subseteq X$ and $B \subseteq Y$, $\langle A, B\rangle$ is a formal concept in $\langle X, Y, I\rangle$ iff $A \times B \subseteq I$ and there is no $A' \supset A$ or $B' \supset B$ such that $A' \times B \subseteq I$ or $A \times B' \subseteq I$.

A set $\mathcal{B}(X, Y, I) = \{\langle A, B\rangle \mid A^{\uparrow} = B, B^{\downarrow} = A\}$ of all formal concepts in data $\langle X, Y, I\rangle$ can be equipped with a partial order $\leq$ (modeling the subconcept-superconcept hierarchy, e.g. $\mathsf{dog} \leq \mathsf{mammal}$) defined by

$$\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \text{ iff } A_1 \subseteq A_2 \text{ (iff } B_2 \subseteq B_1). \tag{1}$$

Note that $^{\uparrow}$ and $^{\downarrow}$ form a so-called Galois connection [6] and that $\mathcal{B}(X,Y,I)$ is in fact a set of all fixed points of $^{\uparrow}$ and $^{\downarrow}$. Under $\leq$, $\mathcal{B}(X,Y,I)$ happens to be a complete lattice, called a concept lattice of $\langle X,Y,I\rangle$, the basic structure of which is described by the so-called main theorem of concept lattices [6]:

Theorem 1. (1) *The set $\mathcal{B}(X,Y,I)$ is under $\leq$ a complete lattice where the infima and suprema are given by*

$$\textstyle\bigwedge_{j\in J}\langle A_j,B_j\rangle = \langle \bigcap_{j\in J} A_j, (\bigcup_{j\in J} B_j)^{\downarrow\uparrow}\rangle,$$
$$\textstyle\bigvee_{j\in J}\langle A_j,B_j\rangle = \langle (\bigcup_{j\in J} A_j)^{\uparrow\downarrow}, \bigcap_{j\in J} B_j\rangle.$$

(2) *Moreover, an arbitrary complete lattice $\mathbf{V} = \langle V,\leq\rangle$ is isomorphic to $\mathcal{B}(X,Y,I)$ iff there are mappings $\gamma : X \to V$, $\mu : Y \to V$ such that*

(i) *$\gamma(X)$ is $\bigvee$-dense in V, $\mu(Y)$ is $\bigwedge$-dense in V;*
(ii) *$\gamma(x) \leq \mu(y)$ iff $\langle x,y\rangle \in I$.*

For a detailed information on formal concept analysis we refer to [4, 6] where a reader can find theoretical foundations, methods and algorithms, and applications in various areas.

Recall that a closure operator in a set Y is a mapping $C : 2^Y \to 2^Y$ satisfying

$$B \subseteq C(B),$$
$$B_1 \subseteq B_2 \text{ implies } C(B_1) \subseteq C(B_2),$$
$$C(C(B)) = C(B)$$

for any $B, B_1, B_2 \in 2^Y$, see e.g. [6].

3 Constraints by Closure Operators

Selecting "interesting" formal concepts from $\mathcal{B}(X,Y,I)$ needs to be accompanied by a criterion of what is interesting. Such a criterion can be seen as a constraint and depends on particular data and application. Therefore, the constraint should be supplied by a user along with the input data $\langle X,Y,I\rangle$. One way to specify "interesting concepts" is to focus on concepts whose sets of attributes are "interesting". This seems to be natural because "interesting concepts" are determined by "interesting attributes/properties of objects". Thus, for a formal context $\langle X,Y,I\rangle$, the user may specify a subset $Y' \subseteq 2^Y$ such that $B \in Y'$ iff the user considers B to be an interesting set of attributes. A formal concept $\langle A,B\rangle \in \mathcal{B}(X,Y,I)$ can be then seen as "interesting" if $B \in Y'$. In this section we develop this idea provided that the selected sets of attributes which are taken as "interesting" form a closure system on Y.

3.1 Interesting Formal Concepts (Maximal Full Rectangles)

We start by formalizing interesting sets of attributes using closure operators.

Definition 1. *Let Y be a set of attributes, $C\colon 2^Y \to 2^Y$ be a closure operator on Y. A set $B \subseteq Y$ of attributes is called a C-interesting set of attributes (shortly, a set of C-attributes) if $B = C(B)$.*

Throughout the paper, Y denotes a set of attributes and $C\colon 2^Y \to 2^Y$ denotes a closure operator on Y. Described verbally, Definition 1 says that C-interesting sets of attributes are exactly the fixed points of the closure operator C. Thus, given any set $B \subseteq Y$ of attributes, $C(B)$ can be seen as *the least set of C-interesting attributes containing B.*

Remark 1. (1) Representing interesting sets of attributes by closure operators has technical as well as epistemic reasons. Specifying particular $C\colon 2^Y \to 2^Y$, we prescribe a particular meaning of "being interesting". Given a set $B \subseteq Y$ of attributes, either we have $B = C(B)$, i.e. B is C-interesting, or $B \subset C(B)$ which can be read: "B is not C-interesting, but additional attributes $C(B) - B$ would make B interesting". Thus, C can be seen as an operator describing which attributes must be added to a set of attributes to make it interesting.

(2) A definition of C depends on particular application. In our approach, we assume that C is any closure operator, covering thus all possible choices of C. On the other hand, in real applications, it is necessary to have a collection of easy-to-understand definitions of such closure operators. In Section 4 we give several examples to define C which are intuitively clear for an inexperienced user.

Definition 2. *Let $\langle X, Y, I\rangle$ be a formal context, $C : 2^Y \to 2^Y$ be a closure operator on Y. We put*

$$\mathcal{B}_C(X,Y,I) = \{\langle A,B\rangle \in 2^X \times 2^Y \mid A^{\uparrow} = B,\ B^{\downarrow} = A,\ B = C(B)\}, \tag{2}$$

$$\mathrm{Ext}_C(X,Y,I) = \{A \subseteq X \mid \textit{there is } B \subseteq Y \textit{ such that } \langle A,B\rangle \in \mathcal{B}_C(X,Y,I)\}, \tag{3}$$

$$\mathrm{Int}_C(X,Y,I) = \{B \subseteq Y \mid \textit{there is } A \subseteq X \textit{ such that } \langle A,B\rangle \in \mathcal{B}_C(X,Y,I)\}. \tag{4}$$

Each $\langle A, B\rangle \in \mathcal{B}_C(X, Y, I)$ is called a C-interesting concept (C-concept); $A \in \mathrm{Ext}_C(X, Y, I)$ is called a C-interesting extent (C-extent); $B \in \mathrm{Int}_C(X, Y, I)$ is called a C-interesting intent (C-intent).

Remark 2. (1) According to Definition 2, $\langle A, B\rangle$ is a C-concept iff $\langle A, B\rangle$ is a concept (in the ordinary sense) such that B is a set of C-attributes. Therefore, C-concepts $\langle A, B\rangle$ can be seen as maximal rectangles in the input data table which are full of $\times$'s, see Section 2, with B being closed under C. Notice that two boundary cases of closure operators on Y are (i) $C(B) = B$ ($B \in 2^Y$), (ii) $C(B) = Y$ ($B \in 2^Y$). For C defined by (i), the notion of a C-concept coincides with that of a concept. In this case, $\mathcal{B}_C(X, Y, I)$ equals $\mathcal{B}(X, Y, I)$. In case of (ii), $\mathcal{B}_C(X, Y, I)$ is a one-element set (not interesting).

(2) Observe that B is a C-intent iff $B = B^{\downarrow\uparrow} = C(B)$. Denoting the set of all fixed points of C by $\mathrm{fix}(C)$, we have $\mathrm{Int}_C(X, Y, I) = \mathrm{Int}(X, Y, I) \cap \mathrm{fix}(C)$.

The following assertion characterizes the structure of C-concepts:

Theorem 2. *Let $\langle X, Y, I\rangle$ be a formal context, $C\colon 2^Y \to 2^Y$ be a closure operator. Then $\mathcal{B}_C(X, Y, I)$ equipped with $\leq$ defined by* (1) *is a complete lattice which is a $\bigvee$-sublattice of $\mathcal{B}(X, Y, I)$.*

Proof. In order to show that $\mathcal{B}_C(X, Y, I)$ equipped with $\leq$ is a complete lattice, it suffices to check that Int_C is closed under arbitrary infima. Take an indexed system $\{B_i \in \mathrm{Int}_C(X, Y, I) \mid i \in I\}$ of C-intents. Since $B_i \in \mathrm{Int}(X, Y, I)$, Theorem 1 gives that $\bigcap_{i\in I} B_i \in \mathrm{Int}(X, Y, I)$. Now, it remains to prove that $B = \bigcap_{i\in I} B_i$ is a set of C-attributes. Since each B_i is a set of C-attributes and C is a closure operator, we get $B = \bigcap_{i\in I} B_i = \bigcap_{i\in I} C(B_i) = C\big(\bigcap_{i\in I} C(B_i)\big) = C\big(\bigcap_{i\in I} B_i\big) = C(B)$. Hence, $B = \bigcap_{i\in I} B_i$ is a set of C-attributes. Altogether, $B \in \mathrm{Int}_C(X, Y, I)$. To see that $\mathcal{B}_C(X, Y, I)$ is a $\bigvee$-sublattice of $\mathcal{B}(X, Y, I)$ observe that $\mathrm{Int}(X, Y, I)$ and $\mathrm{Int}_C(X, Y, I)$ agree on arbitrary intersections and then apply Theorem 1. □

Remark 3. For each context $\langle X, Y, I\rangle$, $Y \in \mathrm{Int}(X, Y, I)$ and $C(Y) = Y$ because C is extensive. Therefore, $Y \in \mathrm{Int}_C(X, Y, I)$, i.e. the set of all attributes determines the least C-concept of $\mathcal{B}_C(X, Y, I)$, see (1). This might seem strange at first sight because the least C-concept of $\mathcal{B}_C(X, Y, I)$ which is also the least concept of $\mathcal{B}(X, Y, I)$ is rather not interesting—it is basically a concept of objects having all attributes. It might be tempting to "remove this concept from $\mathcal{B}_C(X, Y, I)$", however, this would dissolve important structural properties of $\mathcal{B}_C(X, Y, I)$. For instance, after the removal, $\mathcal{B}_C(X, Y, I)$ would not be a lattice in general.

We now focus on the computational aspects of generating all C-concepts. The naive way to compute $\mathcal{B}_C(X, Y, I)$ is to find $\mathcal{B}(X, Y, I)$ first and then go through all of its concepts and filter out the C-concepts. This method is not efficient because in general, $\mathcal{B}_C(X, Y, I)$ can be considerably smaller than $\mathcal{B}(X, Y, I)$. In the sequel we show that $\mathcal{B}_C(X, Y, I)$ can be directly computed using Ganter's NextClosure [6] algorithm without the need to compute $\mathcal{B}(X, Y, I)$.

In order to use the NextClosure [6] algorithm, we need to combine together two closure operators: ${}^{\downarrow\uparrow}$ (operator induced by the Galois connection given by a formal context $\langle X, Y, I\rangle$) and C (operator specifying interesting sets of attributes). For any $B \subseteq Y$ define sets B_i ($i \in \mathbb{N}_0$) and $\mathcal{C}(B)$ of attributes as follows:

$$B_i = \begin{cases} B & \text{if } i = 0, \\ C({B_{i-1}}^{\downarrow\uparrow}) & \text{if } i \geq 1. \end{cases} \tag{5}$$

$$\mathcal{C}(B) = \textstyle\bigcup_{i=1}^{\infty} B_i. \tag{6}$$

Theorem 3. *Let Y be a finite set of attributes, $\langle X, Y, I\rangle$ be a formal context, $C\colon 2^Y \to 2^Y$ be a closure operator on Y, $\mathcal{C}$ be defined by* (6)*. Then $\mathcal{C}\colon 2^Y \to 2^Y$ is a closure operator such that $B = \mathcal{C}(B)$ iff $B \in \mathrm{Int}_C(X, Y, I)$.*

Proof. Since both ${}^{\downarrow\uparrow}$ and C are closure operators, $B_0 \subseteq B_1 \subseteq \cdots$, and $B_i \subseteq \mathcal{C}(B)$ for each $i \in \mathbb{N}_0$. Extensivity and monotony of ${}^{\downarrow\uparrow}$ and C yield extensivity and monotony of $\mathcal{C}$. To check idempotency of $\mathcal{C}$, we show $C((\mathcal{C}(B))^{\downarrow\uparrow}) \subseteq \mathcal{C}(B)$ for

Input: $\langle X, Y, I \rangle$
Output: $\mathrm{Int}_C(X, Y, I)$
$B := \emptyset$, $\mathrm{Int}_C(X, Y, I) := \emptyset$
while $B \neq Y$:
  $B := B^+$
  add B *to* $\mathrm{Int}_C(X, Y, I)$

Input: $\langle X, Y, I \rangle$
Output: $\mathcal{P}_C$ (C-pseudo-intents of $\langle X, Y, I \rangle$)
$B := \emptyset$, $\mathcal{P}_C := \emptyset$
if $B \neq \mathcal{C}(B)$: *add* B *to* $\mathcal{P}_C$
while $B \neq Y$:
  $T := \{P \Rightarrow \mathcal{C}(P) \,|\, P \in \mathcal{P}_C\}$
  $B := B^+_T$
  if $B \neq \mathcal{C}(B)$: *add* B *to* $\mathcal{P}_C$

Fig. 1. Algorithms for computing C-intents (left) and C-pseudo-intents (right); B^+ denotes the lectically smallest fixed point of $\mathcal{C}$ which is a successor of B; B^+_T denotes the lectically smallest fixed point of cl_T which is a successor of B

each $B \subseteq Y$. For each $y \in \mathcal{C}(B)$ denote by i_y an index $i_y \in \mathbb{N}$ such that $y \in B_{i_y}$, where B_{i_y} is defined by (5). We have $\mathcal{C}(B) = \bigcup_{y \in \mathcal{C}(B)} B_{i_y}$. Since Y is finite, $\mathcal{C}(B)$ is finite, i.e. for an index $i = \max\{i_y \,|\, y \in \mathcal{C}(B)\}$, we have $\mathcal{C}(B) = B_i$, where B_i is defined by (5). Therefore, $C((\mathcal{C}(B))^{\downarrow\uparrow}) = C({B_i}^{\downarrow\uparrow}) = B_{i+1} \subseteq \mathcal{C}(B)$, i.e. $\mathcal{C}$ is idempotent. Altogether, $\mathcal{C}$ is a closure operator. We now prove that $B = \mathcal{C}(B)$ iff $B \in \mathrm{Int}_C(X, Y, I)$.

"$\Rightarrow$": Let $B = \mathcal{C}(B)$. Using the above idea, $\mathcal{C}(B) = B_i$ for some index $i \in I$. Therefore, $B = B_i = C({B_{i-1}}^{\downarrow\uparrow})$ for some $i \in I$ which proves that B is a set of C-attributes. Moreover, $B^{\downarrow\uparrow} = {B_i}^{\downarrow\uparrow} \subseteq C({B_i}^{\downarrow\uparrow}) = B_{i+1} \subseteq \mathcal{C}(B) = B$, i.e. $B \in \mathrm{Int}(X, Y, I)$. Putting it together, $B \in \mathrm{Int}_C(X, Y, I)$.

"$\Leftarrow$": Let $B \in \mathrm{Int}_C(X, Y, I)$. By definition, $B = C(B)$ and $B = B^{\downarrow\uparrow}$. Thus, for each $i \in \mathbb{N}$, $B_i = B$, yielding $B = \mathcal{C}(B)$. □

Theorem 3 gives a way to compute C-interesting intents and thus the complete lattice of C-concepts in case of finite Y: we can use Ganter's NextClosure [6] algorithm for computing fixed points of closure operators because the C-interesting intents are exactly the fixed points of $\mathcal{C}$. The algorithm is depicted in Fig. 1 (left).

Remark 4. Notice that NextClosure, being used in Fig.1 to compute the fixed points of $\mathcal{C}$, works with polynomial time delay provided that $C(B)$ ($B \subseteq Y$) can be computed with a polynomial time complexity. Indeed, for each $B \subseteq Y$, $B^{\downarrow\uparrow}$ can be computed with a polynomial time delay (well-known fact). Since Y is finite, there is an index $i \leq |Y|$ such that $\mathcal{C}(B) = B_i$, where B_i is defined by (5). Thus, if $C(B)$ can be computed in a polynomial time, NextClosure can use $\mathcal{C}$ with a polynomial time delay (the same complexity as if NextClosure were using $^{\downarrow\uparrow}$). In practical applications, the computation of $\mathcal{C}(B)$ is usually more time consuming than the computation of $B^{\downarrow\uparrow}$. Still, the number of C-interesting concepts is usually much smaller than the number of all concepts, thus, NextClosure with $\mathcal{C}$ is in most situations considerably faster than NextClosure with $^{\downarrow\uparrow}$.

3.2 Bases of Interesting Attribute Implications

In this section we show that each lattice of C-concepts can be alternatively described by particular sets of "interesting" implications between attributes. We

present a way to compute minimal sets of such implications. We suppose that Y is finite. Recall basic notions of attribute implications and their validity [6, 7]: an *attribute implication* (*over attributes* Y) is an expression $A \Rightarrow B$, where $A, B \in 2^Y$ are sets of attributes. An attribute implication $A \Rightarrow B$ is true in $M \subseteq Y$, written $M \models A \Rightarrow B$, if $A \subseteq M$ implies $B \subseteq M$. Given a set T of attribute implications, $M \subseteq Y$ is called a *model* of T if, for each $A \Rightarrow B \in T$, $M \models A \Rightarrow B$. The system of all models of T is denoted by $\mathrm{Mod}(T)$.

If we focus only on "interesting models" of sets of attribute implications (or sets of "interesting attribute implications"), we naturally come to the following notions of a C-implication and a C-model:

Definition 3. *Let Y be a set of attributes, $C\colon 2^Y \to 2^Y$ be a closure operator, T be a set of attribute implications in Y. An attribute implication $A \Rightarrow B$ in Y is called a C-implication if A and B are sets of C-attributes. $M \subseteq Y$ is called a C-model of T if M is a set of C-attributes and $M \in \mathrm{Mod}(T)$. Denote by $\mathrm{Mod}_C(T)$ the system of all C-models of T.*

Using the notion of a C-model, we define sets of attribute implications which are C-complete in a given formal context:

Definition 4. *Let $\langle X, Y, I\rangle$ be a formal context, $C : 2^Y \to 2^Y$ be a closure operator. A set T of attribute implications is called C-complete in $\langle X, Y, I\rangle$ if*

$$\mathrm{Mod}_C(T) = \mathrm{Int}_C(X, Y, I). \tag{7}$$

A set T of C-implications is called a C-basis of $\langle X, Y, I\rangle$ if T is C-complete in $\langle X, Y, I\rangle$ and no proper subset of T is C-complete in $\langle X, Y, I\rangle$.

Remark 5. (1) Described verbally, a set T of attribute implications is C-complete if the C-models of T are exactly the C-interesting intents. From this point of view, a C-complete set of attribute implications fully describes the lattice of C-concepts using the notion of a C-model. A C-basis is a set of C-implications T (i.e., implications of the form "set of C-attributes A implies a set of C-attributes B") fully describing C-concepts so that one cannot remove any C-implication from T without losing C-completeness. Hence, C-bases are the least C-complete sets of C-implications.

(2) In general, a C-complete set T of attribute implications (C-implications) has models which are not C-models. Also note that if C is given by $C(B) = B$ ($B \in 2^Y$), then the notions of a C-model and a C-completeness coincide with that of a model and a completeness [6].

We now show a way to find particular C-bases. For that purpose, we introduce the following generalized notion of a pseudo-intent:

Definition 5. *Let $\langle X, Y, I\rangle$ be a formal context, $C : 2^Y \to 2^Y$ be a closure operator, $\mathcal{C}$ be defined by* (6)*. A set P of C-attributes is called a C-pseudo-intent of $\langle X, Y, I\rangle$ if $P \subset \mathcal{C}(P)$ and, for each C-pseudo-intent Q of $\langle X, Y, I\rangle$ such that $Q \subset P$, we have $\mathcal{C}(Q) \subseteq P$.*

If C is the identity mapping, the notion of a C-pseudo-intent coincides with the notion of a pseudo-intent, see [6, 7]. All C-pseudo-intents determine a C-basis of a given formal context:

Theorem 4. *Let $\langle X, Y, I\rangle$ be a formal context, $C\colon 2^Y \to 2^Y$ be a closure operator, $\mathcal{C}$ be defined by* (6)*. Then*

$$T = \{P \Rightarrow \mathcal{C}(P) \,|\, P \text{ is a } C\text{-pseudo-intent of } \langle X, Y, I\rangle\} \tag{8}$$

is a C-basis of $\langle X, Y, I\rangle$.

Proof. We first check that T given by (8) is C-complete, i.e. we check equality (7) by showing both inclusions.

"$\subseteq$": Let $M \in \mathrm{Mod}_C(T)$. Thus, M is a set of C-attributes. By contradiction, let $M \neq \mathcal{C}(M)$, i.e. $M \subset \mathcal{C}(M)$ because $\mathcal{C}$ is extensive. Now, for each C-pseudo-intent Q, we have $M \models Q \Rightarrow \mathcal{C}(Q)$ because M is a model of T. Therefore, for each C-pseudo-intent Q, if $Q \subset P$ then $\mathcal{C}(Q) \subseteq P$, i.e. M is a C-pseudo-indent by Definition 5. On the other hand, $M \not\models M \Rightarrow \mathcal{C}(M)$ because $\mathcal{C}(M) \not\subseteq M$, a contradiction to $M \in \mathrm{Mod}_C(T)$.

"$\supseteq$": Let $M \in \mathrm{Int}_C(X, Y, I)$. Then $M = \mathcal{C}(M)$. For each C-pseudo-intent P, if $P \subseteq M$ then $\mathcal{C}(P) \subseteq \mathcal{C}(M) = M$, i.e. $M \models P \Rightarrow \mathcal{C}(P)$.

T is a C-basis: T is obviously a set of C-implications; for each C-pseudo-intent P, $P \models Q \Rightarrow \mathcal{C}(Q)$ where $Q \neq P$ is any C-pseudo-intent. Thus, P is a C-model of $T_P = T - \{P \Rightarrow \mathcal{C}(P)\}$ which gives $\mathrm{Mod}_C(T_P) \supseteq \mathrm{Int}_C(X, Y, I)$, i.e. T_P is not C-complete. □

Due to Theorem 4, in order to get a C-basis of $\langle X, Y, I\rangle$, it suffices to compute all C-pseudo-intents. We now turn our attention to the computation of C-pseudo-intents. Given a set T of attribute implications define sets B^{T_i}, $cl_T(B)$ ($i \in \mathbb{N}_0$):

$$B^{T_i} = \begin{cases} B & \text{if } i = 0, \\ C\big(B^{T_{i-1}} \cup \bigcup\{D \,|\, A \Rightarrow D \in T \text{ and } A \subset B^{T_{i-1}}\}\big) & \text{if } i \geq 1, \end{cases} \tag{9}$$

$$cl_T(B) = \textstyle\bigcup_{i=0}^{\infty} B^{T_i}\,. \tag{10}$$

Operator $cl_T\colon 2^Y \to 2^Y$ has the following property:

Theorem 5. *Let $\langle X, Y, I\rangle$ be a formal context, T be defined by* (8)*, $\mathcal{P}_C$ be the system of all C-pseudo-intents of $\langle X, Y, I\rangle$. Then cl_T defined by* (10) *is a closure operator such that $\{cl_T(B) \,|\, B \subseteq Y\} = \mathcal{P}_C \cup \mathrm{Int}_C(X, Y, I)$.*

Proof. cl_T is a closure operator (apply arguments from the proof of Theorem 3). We check that $\{cl_T(B) \,|\, B \subseteq Y\} = \mathcal{P}_C \cup \mathrm{Int}_C(X, Y, I)$.

"$\subseteq$": Let $B = cl_T(B)$. If $B \notin \mathrm{Int}_C(X, Y, I)$, it suffices to check that B is a C-pseudo-intent. Since Y is finite, $B = cl_T(B) = B^{T_{i_0}}$ for some $i_0 \in \mathbb{N}$. That is, B is of the form $C(\cdots)$, yielding that B is a set of C-attributes. Moreover, for each C-pseudo-intent Q, if $Q \subset B$ then $\mathcal{C}(Q) \subseteq B$ because $B = cl_T(B) = B^{T_{i_0}} = B^{T_{i_0+1}}$. Therefore, B is a C-pseudo-intent.

"$\supseteq$": Clearly, for each C-intent B, $B^{T_i} = B$ ($i \in \mathbb{N}$), i.e. B is a fixed point of cl_T. The same is true if B is a C-pseudo-intent. □

		a	b	c	d	e	f	g	h	i
leech	1	×	×					×		
bream	2	×	×					×	×	
frog	3	×	×	×				×	×	
dog	4	×		×				×	×	×
spike-weed	5	×	×		×		×			
reed	6	×	×	×	×		×			
bean	7	×		×	×	×				
maize	8	×		×	×		×			

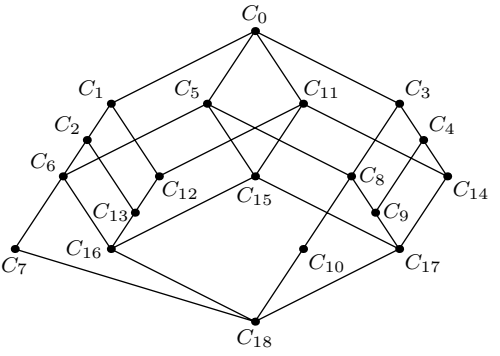

Fig. 2. Context (left) and concept lattice (right); the attributes are: a: needs water to live, b: lives in water, c: lives on land, d: needs chlorophyll to produce food, e: two seed leaves, f: one seed leaf, g: can move around, h: has limbs, i: suckles its offspring

Theorem 5 says that the set of all C-pseudo-intents and all C-intents is the set of all fixed points of cl_T. This provides us with a way to determine a C-basis: we can use the NextClosure [6] algorithm to compute all fixed points of cl_T and then $\{P \,|\, P = cl_T(P) \text{ and } P \neq \mathcal{C}(P)\}$ is the system of all C-pseudo-intents, i.e.

$$T = \{P \Rightarrow \mathcal{C}(P) \,|\, P = cl_T(P) \text{ and } P \neq \mathcal{C}(P)\}$$

is a C-basis due to Theorem 4. The algorithm is depicted in Fig. 1 (right).

4 Examples

Consider an illustrative formal context [6] $\langle X, Y, I\rangle$ given by Fig. 2 (left). The set X of objects contains objects $1, 2, \ldots$ denoting organisms "leech", "bream", $\ldots$; the set Y contains attributes $a, b, \ldots$ denoting certain properties of organisms, see the comment under Fig. 2. The concept lattice $\mathcal{B}(X, Y, I)$ corresponding with $\langle X, Y, I\rangle$ has 19 concepts, here denoted by $C_0, \ldots, C_{18}$:

$C_0 = \langle\{1,2,3,4,5,6,7,8\},\{a\}\rangle$, $C_1 = \langle\{1,2,3,4\},\{a,g\}\rangle$,
$C_2 = \langle\{2,3,4\},\{a,g,h\}\rangle$, $C_3 = \langle\{5,6,7,8\},\{a,d\}\rangle$,
$C_4 = \langle\{5,6,8\},\{a,d,f\}\rangle$, $C_5 = \langle\{3,4,6,7,8\},\{a,c\}\rangle$,
$C_6 = \langle\{3,4\},\{a,c,g,h\}\rangle$, $C_7 = \langle\{4\},\{a,c,g,h,i\}\rangle$,
$C_8 = \langle\{6,7,8\},\{a,c,d\}\rangle$, $C_9 = \langle\{6,8\},\{a,c,d,f\}\rangle$,
$C_{10} = \langle\{7\},\{a,c,d,e\}\rangle$, $C_{11} = \langle\{1,2,3,5,6\},\{a,b\}\rangle$,
$C_{12} = \langle\{1,2,3\},\{a,b,g\}\rangle$, $C_{13} = \langle\{2,3\},\{a,b,g,h\}\rangle$,
$C_{14} = \langle\{5,6\},\{a,b,d,f\}\rangle$, $C_{15} = \langle\{3,6\},\{a,b,c\}\rangle$,
$C_{16} = \langle\{3\},\{a,b,c,g,h\}\rangle$, $C_{17} = \langle\{6\},\{a,b,c,d,f\}\rangle$,
$C_{18} = \langle\{\},\{a,b,c,d,e,f,g,h,i\}\rangle$.

Fig. 2 (right) depicts the concept lattice $\mathcal{B}(X, Y, I)$ [6].

(a) Define C so that B is C-interesting iff $B = Y$ or $|B^{\downarrow}| \geq s$ where s is a non-negative integer. It is easy to see that C-interesting sets form a closure

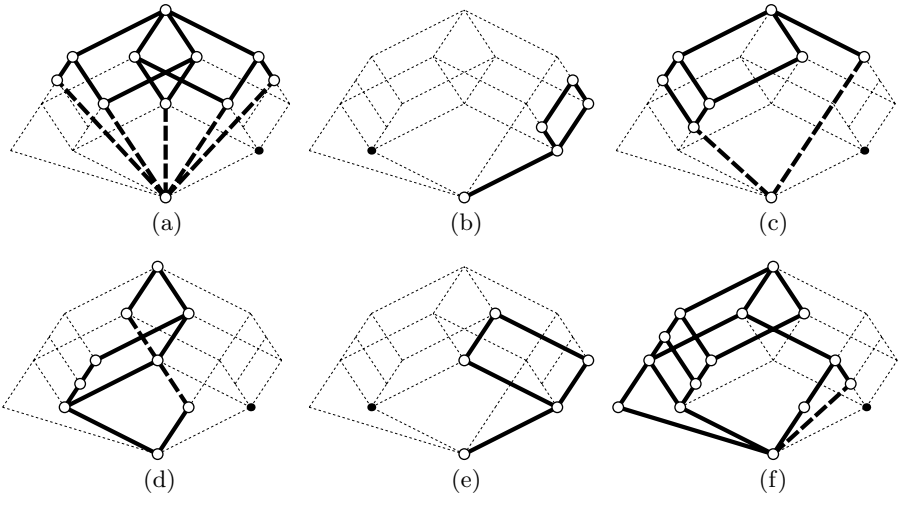

Fig. 3. Concept lattice constrained by various closure operators

system. $|B^{\downarrow}| \geq s$ means that the number of objects sharing all attributes from B exceeds a user-defined parameter s called *support* in association rules [13]. Condition $B = Y$ is a technical one to ensure that C-interesting sets form a closure system. The corresponding closure operator is defined by

$$C(B) = \begin{cases} B & \text{if } |B^{\downarrow}| \geq s, \\ Y & \text{otherwise.} \end{cases}$$

Then, the set $\mathrm{Int}_C(X, Y, I) - \{Y\}$ of all C-interesting intents without Y coincides with the set of closed frequent itemsets defined by Zaki [12] in order to get non-redundant association rules.

(b) Define C by

$$C(B) = \begin{cases} B & \text{if } |B| \leq n, \\ Y & \text{otherwise.} \end{cases}$$

That is, $B \subseteq Y$ is C-interesting iff B contains at most n attributes or $B = Y$. That is, C can be used to determine intents with at most n attributes. Fig. 3 (a) depicts the situation for $n = 3$: "$\bullet$" denote concepts of $\mathcal{B}(X, Y, I)$ which are not present in $\mathcal{B}_C(X, Y, I)$; "$\circ$" denote C-concepts; dotted lines denote edges of the original concept lattice which are not present in $\mathcal{B}_C(X, Y, I)$; bold solid lines denote edges which are presented in both $\mathcal{B}(X, Y, I)$ and $\mathcal{B}_C(X, Y, I)$; bold dashed lines denote new edges which are in $\mathcal{B}_C(X, Y, I)$ but are not in the original concept lattice.

(c) For any $Z \subseteq Y$, C defined by $C(B) = B \cup Z$ is a closure operator. This closure operator determines intents containing Z. Notice that the boundary cases mentioned in Remark 2 (1) are given by choices $Z = \emptyset$ and $Z = Y$, respectively. For instance, $Z = \{d, f\}$ determines a constraint on "organisms with one seed leaf that need chlorophyll to produce food", see Fig. 3 (b).

(d) For any $Z \subseteq Y$, we can define C so that B is C-interesting iff B does not contain any attribute from Z (or $B = Y$) by putting

$$C(B) = \begin{cases} B & \text{if } B \cap Z = \emptyset, \\ Y & \text{otherwise.} \end{cases}$$

Fig. 3 (c) contains a lattice for $Z = \{c, e, f\}$.

(e) A general method for defining C is the following. Consider a binary relation R on Y, $\langle y_1, y_2 \rangle \in R$ meaning that if y_1 is an attribute of a concept then y_2 should also be an attribute of that concept. Now, put

$$B_i = \begin{cases} B & \text{if } i = 0, \\ B_{i-1} \cup \{y' \mid \text{there is } y \in B_{i-1} \colon \langle y, y' \rangle \in R\} & \text{if } i \geq 1, \end{cases}$$
$$C(B) = \bigcup_{i=0}^{\infty} B_i.$$

Since Y is finite, we have $C(B) = B_{i_0}$ for some $i_0 \in \mathbb{N}$. B is C-interesting iff all dependencies given by R are satisfied. In more detail, B is C-interesting iff IF $\langle y_1, y_2 \rangle \in R$ and $y_1 \in B$ THEN $y_2 \in B$. Fig. 3 (d) depicts the resulting structure for $R = \{\langle g, b \rangle, \langle d, e \rangle\}$.

(f) A particular case of (e) is a constraint given by an equivalence relation (i.e., R is reflexive, symmetric, and transitive), see also [2]. In this case, $C(B) = \bigcup\{[y]_R \mid y \in B\}$, where $[y]_R$ denotes the class of R containing y. Fig. 3 (e) contains a structure determined by an equivalence R induced by a partition $\{\{a, b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g, h, i\}\}$.

(g) Let T be a set of attribute implications. The system of all models of T is a closure system [6], the corresponding closure operator C can be described as follows [6]:

$$B_i = \begin{cases} B & \text{if } i = 0, \\ B_{i-1} \cup \bigcup\{D \mid A \Rightarrow D \in T \text{ and } A \subseteq B_{i-1}\} & \text{if } i \geq 1, \end{cases}$$
$$C(B) = \bigcup_{i=0}^{\infty} B_i.$$

$C(B)$ is the least model of T containing B. Hence, B is C-interesting iff B is a model of attribute implications from T. For $T = \{\{b, c\} \Rightarrow \{h\}, \{d\} \Rightarrow \{c\}\}$, the resulting structure in depicted in Fig. 3 (f). Notice that this type of definition of a closure operator is, in fact, the most general one, because each closure operator on a finite set of attributes can be completely described by a set of attribute implications (i.e., the fixed points of C are exactly the models of some set of attribute implications).

Consider a closure operator C such that $C(B) = B$ ($B \in 2^Y$). Let T be the C-basis given by (8). Since C is an identical operator, T is a basis of the concept lattice $\mathcal{B}(X, Y, I)$. In this particular case, T is the following:

$$T = \{\{a, b, c, g, h, i\} \Rightarrow Y, \{a, b, d\} \Rightarrow \{a, b, d, f\}, \{a, c, d, e, f\} \Rightarrow Y,$$
$$\{a, c, g\} \Rightarrow \{a, c, g, h\}, \{a, d, g\} \Rightarrow Y, \{a, e\} \Rightarrow \{a, c, d, e\}, \{a, f\} \Rightarrow \{a, d, f\},$$
$$\{a, h\} \Rightarrow \{a, g, h\}, \{a, i\} \Rightarrow \{a, c, g, h, i\}, \{\} \Rightarrow \{a\}\}.$$

If we define C as in Example (b), T defined by (8) is a C-basis of the constrained lattice of C-concepts depicted in Fig. 3 (a):

$$T = \{\{a,b,d\} \Rightarrow Y, \{a,c,g\} \Rightarrow Y, \{a,d,g\} \Rightarrow Y, \{a,e\} \Rightarrow Y, \{a,f\} \Rightarrow \{a,d,f\},$$
$$\{a,h\} \Rightarrow \{a,g,h\}, \{a,i\} \Rightarrow Y, \{\} \Rightarrow \{a\}\}.$$

Observe that since C determines concepts with at most three attributes, each implication in the latter T has at most three attributes on both sides of "$\Rightarrow$" or the right-hand side of the implication consists of the whole set of attributes Y.

5 Further Issues

For limited scope, we did not present the following topics some of which will appear in a full paper or are subject of future research:

- Interactive specification of constraining closure operators. An expert might not be able to explicitly describe a constraining closure operator. However, he/she is usually able to tell which formal concepts from the whole $\mathcal{B}(X,Y,I)$ are interesting. If $\mathcal{I}$ is a subset of $\mathcal{B}(X,Y,I)$ identified as (examples of) interesting formal concepts, an important problem is to describe a possibly largest closure operator C such that each $\langle A,B\rangle \in \mathcal{I}$ is C-interesting. Namely, putting $C_1 \leq C_2$ iff for each $B \in 2^Y$ we have $C_1(B) \subseteq C_2(B)$ for closure operators C_1 and C_2, we have $C_1 \leq C_2$ iff $\mathrm{fix}(C_2) \subseteq \mathrm{fix}(C_1)$ where $\mathrm{fix}(C_i)$ is a set of all fixed points of C_i. Therefore, since we require $B \in \mathrm{fix}(C)$ for each $\langle A,B\rangle \in \mathcal{I}$, larger C means a better approximation of $\mathcal{I}$ by $\mathcal{B}_C(X,Y,I)$. The problem is to find a tractable description of C. For instance, if C is supposed to be given by an equivalence relation R, see Section 4 (e), then given $\mathcal{I}$, the largest closure operator C we look for is the one induced by a relation $R = R_{\mathcal{I}}$ where

$$\langle y_1, y_2\rangle \in R_{\mathcal{I}} \quad \text{iff} \quad \text{for each } \langle A,B\rangle \in \mathcal{I} : y_1 \in B \text{ iff } y_2 \in B.$$

  Then, one can present $\mathcal{B}_C(X,Y,I)$ to the expert who might then revise the selection of $\mathcal{I}$, etc., to finally arrive at a satisfactory closure operator C.
- Entailment of constraints. Intuitively, a constraint $\mathcal{C}_1$ (semantically) entails a constraint $\mathcal{C}_2$ iff each $B \subseteq Y$ satisfying $\mathcal{C}_1$ satisfies $\mathcal{C}_2$ as well. A study of entailment is important for obtaining small descriptions of constraining closure operators.
- More detailed results and more efficient algorithms for particular closure operators can be obtained (we omit details).

References

1. Bělohlávek R., Sklenář V., Zacpal J.: Formal concept analysis with hierarchically ordered attributes. *Int. J. General Systems* **33**(4)(2004), 283–294.
2. Bělohlávek R., Sklenář V.: Formal concept analysis constrained by attribute-dependency formulas. In: B. Ganter and R. Godin (Eds.): ICFCA 2005, *Lect. Notes Comp. Sci.* **3403**, pp. 176–191, Springer-Verlag, Berlin/Heidelberg, 2005.

3. Boulicaut J.-F., Jeudy B.: Constraint-based data mining. In: Maimon O., Rokach L. (Eds.): *The Data Mining and Knowledge Discovery Handbook*, Springer, 2005. pp. 399–416.
4. Carpineto C., Romano G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.
5. Dekel U., Gill Y.: Visualizing class interfaces with formal concept analysis. In *OOPSLA'03*, pages 288–289, Anaheim, CA, October 2003.
6. Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
7. Guigues J.-L., Duquenne V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. *Math. Sci. Humaines* **95**(1986), 5–18.
8. Maier D.: *The Theory of Relational Databases.* Computer Science Press, Rockville, 1983.
9. Norris E. M.: An algorithm for computing the maximal rectangles of a binary relation. *Journal of ACM* 21:356–266, 1974.
10. Pasquier N., Bastide Y., Taouil R., Lakhal L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. *Information Systems* **24**(1)(1999), 25–46.
11. Snelting G., Tip F.: Understanding class hierarchies using concept analysis. *ACM Trans. Program. Lang. Syst.* 22(3):540–582, May 2000.
12. Zaki M. J.: Mining non-redundant association rules. *Data Mining and Knowledge Discovery* **9**(2004), 223–248.
13. Zhang C., Zhang S.: *Association Rule Mining. Models and Algorithms.* Springer, Berlin, 2002.