Concept Analysis on structured, multi-valued and incomplete data

David Grosser, Henri Ralambondrainy

Laboratoire IREMIA, Université de la Réunion,
15 avenue René Cassin, BP 7151,
97715 Saint-Denis Msg. Cedex 9,
E-mail: {grosser,ralambondrainy}@univ-reunion.fr

Abstract. This paper presents an approach to Concept Analysis of structured, multivalued and incomplete data currently present in life science knowledge bases. We are concerned with tree structured objects, whose size may be variable. We focus on the composition relations between attributes in the learning process. The interest of the method is the ability to take into account both structural and value parts of the objects. An application on a coral knowledge base illustrates the advantages of the method.

1 Introduction

One of the essential issues in classification science concerns biological specimens and taxa representations and analysis [9]. This is particularly the case in marine environment for groups like corals, hydroids or sponges for which descriptions of specimens and taxonomy are particularly complex. Descriptions are often multi-valued due to variability inside of same species, structured to take into account characters dependencies and noisy or incomplete [4]. In the context of the "Knowledge Base on corals project"[5], we have developed a specific knowledge representation and analysis system: IKBS (*Iterative Knowledge based System*), to achieve identification, classification and conceptual analysis from systematic morphological descriptions.

To deal with such descriptions, we present a method for Concepts Analysis from structured, multivalued and incomplete objects. Formal Concept Analysis (FCA) [7] has been successfully applied to a range of knowledge engineering problems [14]. Traditional FCA methods and tools are usually concerned with objects described by binary contexts. Extracting concepts from more complex contexts is a recent and challenging trend of research on FCA [13]. Indeed, real-world data are often complex and difficult to be transformed in a binary format without loss of information. One key difficulty lies in the presence and management of relational attributes such as references or part-of relations between objects. For example, in [12] methods are proposed to find relational concepts in structured datasets in which individuals are described both by their own features and by their relations to other. Such data are currently found in relational or object oriented databases, or software models such as UML. In a similar research

trend, [3] shows how FCA can be used to support Ontology Engineering and how ontologies can be exploited in FCA applications as background knowledge to assure consistency and scalability of the results [3].

In our approach, we are concerned with tree structured objects corresponding to specimen descriptions. The object's structure is defined in a model that represents all characteristics (attributes, relations and values) and background knowledge of a particular concept, corresponding to a taxa (family, genus, species). However, the size of each object may be different from others (different special schemas called skeletons) because of inapplicable attributes and dependencies between them. In this paper, we focus on using some background knowledge and particularly the composition relations between attributes in the Concept Analysis process. The interest of the method is the ability to take into account both structural and value part of the objects.

The paper is organized as follows: Section 2 recalls results on Galois connection on semilattices. Section 3 gives the knowledge representation model used to describe objects. Section 4 presents a way to make Concept Analysis on structured and multivalued objects. The approach is illustrated by an application example in Section 5.

2 Preliminaries

In this section, we recall some results on Galois Connection (GC) between semilattices that will be used in further sections.

Let P and Q be ordered sets. We recall that a pair $GC = (f, g)$ of maps $f : P \to Q$ and $g : Q \to P$ is a Galois Connection (GC) between P and Q if, for all $p \in P$ and $q \in Q : f(p) \leq q \iff p \leq g(q)$. The mapping $h = g \circ f$ and $k = f \circ g$ are closure operators in P and Q. Any pair (p, q) such that $(p = g(q), q = f(p))$ is called concept [7].

The definition of GC between lattices can be found in [1] and GC between semilattices has been studied by [8] [10] [6], it is useful because it gives a suitable framework for concepts analysis for data which are not binary.

We denote by O a set of objects, and Γ a meet semilattice. Let $\delta : O \to \Gamma$ be the mapping which associate every element $o \in O$ with its description $\delta(o) \in \Gamma$. The context $\mathbb{K} = (O, \Gamma, \delta)$ is called *pattern structure* in [8]. The descriptor δ induces a GC between $(\mathcal{P}(O); \subset, \cup)$ and $(\Gamma; <, \wedge)$ by means of the map, such that for $\gamma \in \Gamma$ $ext(\gamma) = \{o \in O | \gamma \leq \delta(o)\}$ and for $L \subset O$ $int(L) = \wedge_{l \in L} \delta(l)$. The GC is denoted by $GC = (ext, int)$. A concept or *pattern concept* is a pair $c = (L, \gamma)$ such that $\gamma = int(L)$ and $L = ext(\gamma)$. The subset L is called the *extension* of the concept c and γ its *intension*.

3 Knowledge representation model

The knowledge representation model is made of the descriptive model and its instances, the structured objects.

3.1 Attributes

The descriptive model represents all the observable characteristics (objects, attributes and values) pertaining to individuals belonging to a particular taxa. It is organized in a structured schema forming a tree. Each node of the tree is a component of the description defined by a list of attributes with their respective definition domain and a set of meta-data as rules, comments, hyperlinks and pictures. See Figure 1 for an overview of a descriptive model structure composed by two components, "identification" and "description", itself composed by "colony", "microstructure" and so on.

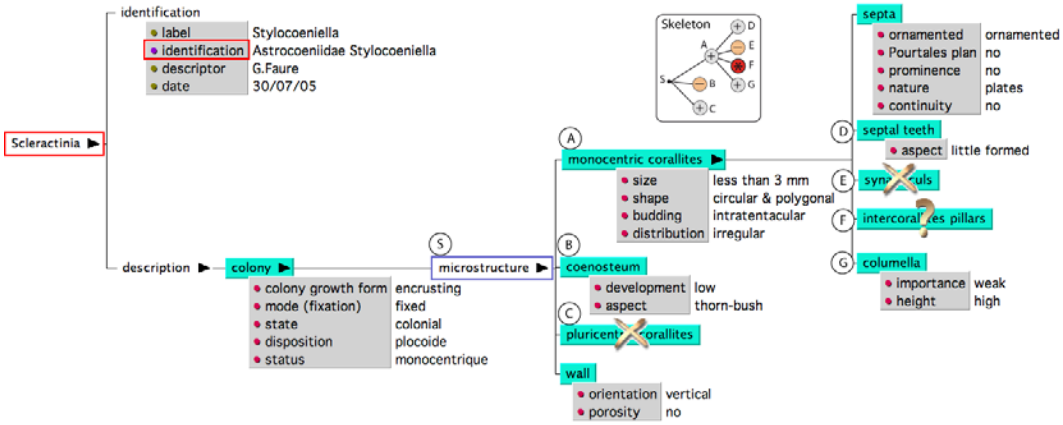

Fig. 1. Partial description of the genus *Astrocoeniidae Stylocoeniella*

Moreover, this component defines a particular boolean property, called "possible absence". It means that the component should necessarily be described or could be described in an object instance of the model. The attributes noted from A to G in the Figure 1 are "contingent", the other are "necessary".

Two types of attributes are considered. First, basic attributes that are usual attributes whose types are qualitative (ordinal, nominal, boolean) or quantitative (discrete or interval), and hierarchical attribute (also called taxonomic attribute) for which values are organized in a hierarchy. Second, structured attributes formed using any kind of distinct attributes.

3.2 Objects

In this section, we are concerned with the structures of objects and we shall define the meet semilattice of the structured description space of objects.

Notation We suppose given a set of basic attributes names, $A_Q = \{A_q\}_{q \in Q}$ and their corresponding domains $D_Q = \{D_q\}_{q \in Q}$. A structured attribute is recursively defined as a sequence $A :< A_1, \ldots, A_l, \ldots, A_p >$ of basic *or structured* attributes. We say that A_l is a component of A. A structured attribute is used to describe composite objects. We assume that a structured attribute A called a *schema* is given for describing a collection of objects. The set of attributes that composes A is denoted $\mathcal{A} = \{A_j\}_{j \in J}$. A structured attribute A is represented by a rooted tree $\mathcal{M} = (\mathcal{A}, \mathcal{U})$ where the set of nodes and edges are denoted by $\mathcal{A}$ and $\mathcal{U}$, respectively. The root of $\mathcal{M}$ is A, and the nodes are the attributes, basic attributes are the leaves. If $(B, B') \in \mathcal{U}$ is an edge, it means that B or B' is a component of the other.

Skeleton Let O be a set of objects described by a schema $\mathcal{M} = (\mathcal{A}, \mathcal{U})$. A *skeleton* represents the structure of an object. In the Figure 1, missing parts of the object are represented with a cross, and ? means that the component is undefined or unknown. In [11] to deal with unknown and missing values, an incomplete context is defined as $\mathbb{K}_i = (O, A, \{+, ?, -\}, J)$ with an extension of KLEENE-logic is proposed. In our approach, we use a semi-lattice to represent missing and unknown values. We give the formal definition of a skeleton:

$S = \{+ = \text{"existing, present"}, - = \text{"missing, absent"}, * = \text{"unknown,undefined"}\}$,

a skeleton is a labeled rooted tree $\mathcal{M}$ using the alphabet S i.e. each node $A_j \in \mathcal{A}$ of $\mathcal{M}$ is assigned a symbol from S. A map $\sigma : \mathcal{A} \to S$ defines a labeled rooted tree H_σ :

$$H_\sigma = (\mathcal{A}_\sigma, \mathcal{U}) \text{ with } \mathcal{A}_\sigma = \{(A_j, \sigma(A_j))_{j \in J}\}.$$

The skeleton nodes satisfy the following properties: the descendants of a missing (respectively unknown) node must be missing (respectively unknown). If a node is present, its children may be present, absent or unknown. Then, all the labeled rooted trees H_σ defined from a mapping $\sigma \in S^{\mathcal{A}}$ are not a valid representation of a skeleton object, it leads to:

Definition 1. *Let $B :< B_l >_{l \in L}$ be any structured attribute. The mapping $\sigma \in S^{\mathcal{A}}$ is said to be consistent, if it satisfies the following conditions:*
1) $\sigma(B) = - \Rightarrow \sigma(B_l) = -$ for $l \in L$, 2) $\sigma(B) = * \Rightarrow \sigma(B_l) = *$ for $l \in L$. The set of consistent maps is denoted $S_c^{\mathcal{A}}$.

We denote by $\mathcal{H}$ the set of skeletons related to consistent maps of $S_c^{\mathcal{A}}$.

Order Skeletons are defined from mapping $\sigma \in S_c^{\mathcal{A}}$. To order the skeleton space $\mathcal{H}$, it suffices to define an order on $S_c^{\mathcal{A}}$. The set $S = \{+, -, *\}$ is ordered as follows $* < +$ and $* < -$. In the context of information orderings, it means that $+$ and $-$ is more defined or precise than $*$. Let us notice that $+$ and $-$ are not comparable. Then $S^{\mathcal{A}}$ is pointwise ordered, for maps $s, s' \in S^{\mathcal{A}}$

$$s \leq s' \iff \forall A_j \in \mathcal{A}, s(A_j) \leq s(A_j)$$

The set $S^{\mathcal{A}}$ has **a minimum element** σ_* such as: $\forall A_j \in \mathcal{A}, s_*(A_j) = *$. The set $S_c^{\mathcal{A}} \subset S^{\mathcal{A}}$ inherits the pointwise order.

Semilattice We will define a semilattice structure on the skeleton set $\mathcal{H}$. The ordered set $(S = \{+, -, *\}, <)$ is a meet semilattice, because $* = + \wedge_S -$. It means that an undefined value is interpreted as a missing or existing node. Then $S^{\mathcal{A}}$ is also a meet semilattice, $\wedge_{S^{\mathcal{A}}}$ in $S^{\mathcal{A}}$ is defined from $\wedge_S$ as follows :

$$\forall A_j \in \mathcal{A}, s \wedge_{S^{\mathcal{A}}} s'(A_j) = s(A_j) \wedge_S s'(A_j).$$

Unfortunately, $S_c^{\mathcal{A}}$ is not a meet semilattice for $\wedge_{S^{\mathcal{A}}}$ because $S_c^{\mathcal{A}}$ is not stable under $\wedge_{S^{\mathcal{A}}}$. Consider $B =< B_1, B_2 >$ and $\sigma, \sigma' \in S_c^{\mathcal{A}}$ such as: $\sigma(B) = +, \sigma'(B) = -;\ \sigma(B_1) = +, \sigma'(B_1) = -;\ \sigma(B_2) = -, \sigma'(B_2) = -$. Then, we have: $\sigma(B) \wedge_{S^{\mathcal{A}}} \sigma'(B) = + \wedge_S - = *;\ \sigma(B_1) \wedge_{S^{\mathcal{A}}} \sigma'(B_1) = + \wedge_S - = *;\ \sigma(B_2) \wedge_{S^{\mathcal{A}}} \sigma'(B_2) = - \wedge_S - = -$

We see (Figure 2) that the value $-$ of the child B_2 of B is not equal to his father's value $*$, $\sigma \wedge_{S^{\mathcal{A}}} \sigma'$ is not consistent (we will say that the node B_2 is inconsistent for $\sigma \wedge_{S^{\mathcal{A}}} \sigma'$). Next proposition defines an operator $\wedge$ that associates

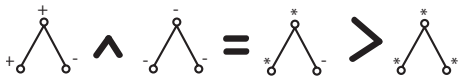

Fig. 2. Inf operator applied on simple skeletons.

a greatest lower bound in $S_c^{\mathcal{A}}$ to any $\sigma, \sigma' \in S_c^{\mathcal{A}}$.

Proposition 1. *Let $\sigma, \sigma' \in S_c^{\mathcal{A}}$. The set $S_c^{\mathcal{A}}$ is a meet semilattice such that:*

$$\sigma \wedge \sigma' = \bigvee([\sigma_*, \sigma \wedge_{S^{\mathcal{A}}} \sigma'] \cap S_c^{\mathcal{A}})$$

Proof. The set of all lower bounds of $\{\sigma, \sigma'\}$ is the interval $[\sigma_*, \sigma \wedge_{S^{\mathcal{A}}} \sigma']$ in $S^{\mathcal{A}}$. The set $[\sigma_*, \sigma \wedge_{S^{\mathcal{A}}} \sigma'] \cap S_c^{\mathcal{A}}$ is not empty because the minimum element $\sigma_* \in S_c^{\mathcal{A}}$. We are going to define the upper bound $\sigma \wedge \sigma'$ of $\{\sigma, \sigma'\}$ in $S_c^{\mathcal{A}}$. Let $B :< B_l >_{l \in L}$ be any structured attribute. Notice that if $\sigma \wedge_{S^{\mathcal{A}}} \sigma'(B) = -$, the node B does not lead to inconsistency. Hence, $\sigma \wedge_{S^{\mathcal{A}}} \sigma'(B) = - \Rightarrow \sigma(B) = \sigma'(B) = -$ as $\sigma, \sigma' \in S_c^{\mathcal{A}}$ for $l \in L, \sigma(B_l) = \sigma'(B_l) = -$ and $\sigma \wedge_{S^{\mathcal{A}}} \sigma'(B_l) = -$. Any inconsistency node B_l is such that $\sigma(B_l) \wedge_{S^{\mathcal{A}}} \sigma'(B_l) = -$ with $\sigma(B) \wedge_{S^{\mathcal{A}}} \sigma'(B) = *$. It means that the father B is present in one skeleton and missing in the other one and the child B_l is missing in the two skeletons (if B is indefinite in the two skeletons, all children will be indefinite because σ and σ' are consistent). In this case, we define $\sigma \wedge \sigma'(B_l) = *$. To sum up, we have $\sigma \wedge \sigma'(A_j) = \sigma \wedge_{S^{\mathcal{A}}} \sigma'(A_j)$ for all consistent nodes A_j, and $\sigma \wedge \sigma'(A_j) = *$ for all inconsistent nodes A_j. It is easy to see that $\sigma \wedge \sigma'$ is the greatest consistent lower bound of $\{\sigma, \sigma'\}$.

As $(S_c^{\mathcal{A}}; <, \wedge)$ is a meet semilattice, then the set of skeletons set $(\mathcal{H}; <, \wedge)$ is a meet semilattice, such that for $H_\sigma, H_{\sigma'} \in \mathcal{H}$:

$$H_\sigma \wedge H_{\sigma'} = H_{\sigma \wedge \sigma'}$$

The procedure that computes $\sigma \wedge \sigma'(A_j)$ is given below:

Procedure $\sigma \wedge \sigma'(A_j) = \bigwedge(\sigma(A_j), \sigma'(A_j))$

- If A_j is a basic attribute then *return* $\sigma(A_j) \wedge_S \sigma'(A_j)$;
- elseif $A_j :< A_l >_{l \in L}$ is a structured attribute,
  - If $\sigma(A_j) = \sigma'(A_j) = +$ then $\{$ $\sigma \wedge \sigma'(A_j) = +$; For $l \in L, \sigma \wedge \sigma'(A_l) = \bigwedge(\sigma(A_l), \sigma'(A_l))$; $\}$ elseif $\sigma(A_j) = \sigma'(A_j) = -$ then *return* $-$ else *return* $*$;

The skeleton $H_\sigma \wedge H_{\sigma'}$ is built from the root down by applying, in breadth-first way, the procedure $\bigwedge(\sigma(A), \sigma(A))$. It stops when all common present structured attributes have been processed. Then, descendant of missing nodes must be labeled with $-$ and descendant of unknown nodes with $*$. The procedure $\bigwedge$, only on common present nodes, computes the greatest lower bound recursively this leads us to the definition of the **skeleton level**. Let $l(A_j)$ be the level number of the node A_j i.e. the length of the unique simple path from the root to A_j.

Definition 2. *The level $\nu(H_\sigma)$ of the skeleton H_σ is the largest level number of present nodes in H_σ : $\nu(H_\sigma) = max\{l(A_j) | \sigma(A_j) = +, A_j \in \mathcal{A}\}$*

4 Concepts Analysis on structured and multivalued data

In the Section 2, GC on semilattices has been introduced, and in previous sections a semilattice structure has been built on the skeleton set. Here, we apply these results to concepts determination for structured data.

Let denote by H_{σ_o} the skeleton of the object o, where $\sigma_o : \mathcal{A} \to S$. The mapping $d : O \to \mathcal{H}$ associates every element $o \in O$ with its description $d(o) = H_{\sigma_o}$. Consider the semilattice skeleton $(\mathcal{H}; <, \wedge)$ and $(\mathcal{P}(O); \subset, \cup)$. The pair $GC = (int, ext)$ of maps $ext : \mathcal{H} \longrightarrow \mathcal{P}(O)$ and $int : \mathcal{P}(O) \longrightarrow \mathcal{H}$, is a GC such as, for any $H_\sigma \in \mathcal{H}$:

$$ext(H_\sigma) = \{o \in O | H_\sigma \leq H_{\sigma_o}\}$$

and for $L \subset O$

$$int(L) = \bigwedge_{l \in L} H_{\sigma_l} = H_{\wedge_{l \in L} \sigma_l}.$$

The structure context is $\mathbb{K}_s = (O, \mathcal{H}, d)$, and the set of concepts induced by GC will be denoted by $\mathcal{C}$.

Let r be the height of the rooted tree $\mathcal{M} = (\mathcal{A}, \mathcal{U})$ i.e. the largest level number of a node, and let k be an integer such that $1 \leq k \leq r$. and

- $\mathcal{A}_k = \{A_j \in \mathcal{A} | l(A_j \leq k\}$ the set of attributes with a level less or equal to k,
- $\mathcal{M}_k = (\mathcal{A}_k, \mathcal{U}_k)$ the rooted tree such that the eight is k,
- $S_{\mathcal{C}}^{\mathcal{A}_k}$ the set of consistent mappings $\sigma^k : \mathcal{A}_k \to S$,
- $\mathcal{H}_k = \{H_{\sigma^k}\}$ the semilattice skeleton defined by $\mathcal{M}_k$,

- d_k the mapping $d_k : O \rightarrow \mathcal{H}_k$ such that $d_k(o) = H_{\sigma_o^k}$, the subtree of H_{σ_o} limited to nodes whose levels are less or equal to k.
- $GC_k = (int_k, ext_k)$ the GC, related to the context $\mathbb{K}_k = (O, \mathcal{H}_k, d_k)$, such that $ext_k(H_{\sigma^k}) = \{o \in O | H_{\sigma^k} \leq H_{\sigma_o^k}\}$ and $int_k(L) = \bigwedge_{l \in L} H_{\sigma_l^k} = H_{\wedge_{l \in L} \sigma_l^k}$,
- $\mathcal{C}_k$ the set of concepts induced by GC_k

The relationship between the set of concept $\mathcal{C}_k$ and $\mathcal{C}$ is precised by the following proposition

Proposition 2. *Let k be an integer $1 \leq k \leq r$, and let $c_k \in \mathcal{C}_k$ be a concept induced by GC_k. If the level $\nu(int_k(c_k))$ of the skeleton $int_k(c_k)$ is strictely less than k then it exists one concept $c \in \mathcal{C}$ induced by GC, such that its intension $int(c)_k$, limited to nodes whose levels are less or equal to k, is $int_k(c_k)$ and $ext(c) = ext_k(c_k)$. Conversely, if $c \in \mathcal{C}$ is a concept such that its level $\nu(int(c)) < r$ then, for any integer k such that $\nu(int(c)) \leq k \leq r$, $c_k = (int_k(c), ext(c))$ is a concept of $\mathcal{C}_k$.*

Proof. Let us note that for any skeleton H_σ, the projection of H_σ on $\mathcal{A}_k$ is H_{σ^k}. Let $c_k \in \mathcal{C}_k$, and denoted by $L = ext(c_k)$ and $H_{\sigma^k} = int(c_k) = \bigwedge_{l \in L} H_{\sigma_l^k}$. Consider that the level $\nu(H_{\sigma^k})$ is strictely less than k then nodes $A_j \in H_{\sigma^k}$, such that $l(A_j) = k$, is missing or unknown. There is an unique consistent skeleton $H_\sigma \in \mathcal{H}$, such that its projection on $\mathcal{A}_k$ is H_{σ^k}. H_σ is obtained by labeling the descendants of missing nodes, whose level is greater or equal to k, by $-$ and the descendants of unknown nodes of H_{σ^k}, whose level is greater or equal to k., by $*$. Consider that level $\nu(H_{\sigma^k}) < k$, and $H_{\sigma^k} = \bigwedge_{l \in L} H_{\sigma_l^k}$, this means that the objects of the extension L of c_k have not present nodes in common such that the level is greater than k, then $H_\sigma = \bigwedge_{l \in L} H_{\sigma_o}$, is the intension of L and $c = (H_\sigma, L)$ is a concept of $\mathcal{C}$ with the same extension than c_k.

Let $c \in \mathcal{C}$ whose intension is $int(c) = H_\sigma = \bigwedge_{o \in\ ext(c)} H_{\sigma_o}$, whose level $\nu(int(c)$ is strictely less than r. Let denote by H_{σ^k} the projection of H_σ at the level k. The level of H_σ is strictely less than r, then, for k such that $\nu(int(c)) \leq k \leq r$, we can state that $H_{\sigma^k} = \bigwedge_{o \in\ ext(c)} H_{\sigma_o^k}$. And c_k is a concept of $\mathcal{C}_k$ such that its intension is H_{σ^k} and its extension $ext(c_k) = ext(c)$.

This proposition gives a top down algorithm for structure concepts search. If c_k is a concept of $\mathcal{C}_k$, we can derive concepts c of $\mathcal{C}$ from c_k as follows

Procedure $\{c\} = DeriveConcepts(H_{\sigma^k} = int(c_k), ext(c_k))$

- Compute $A_k^+ = \{A_j \in \mathcal{A}_k | \sigma^k(A_j) = +, l(A_j) = k\}$;
- If $A_k^+ = \emptyset$ then return $c = c_k$,
- elseif $\{c_{k+1}\} = ConceptAnalysis(\mathbb{K}_{k+1} = (ext(c_k), \mathcal{H}_{k+1}, d_{k+1}))$;
- For each c_{k+1} do $DeriveConcepts(int(c_{k+1}), ext(c_{k+1}))$;

The procedure $\{c_{k+1}\} = ConceptAnalysis(\mathbb{K}_{k+1} = (ext(c_k), \mathcal{M}_{k+1}, d_{k+1}))$ extracts concepts c_{k+1} from the extension of c_k. One can show that it may be implemented using a standard Formal Concept Analysis algorithm applied to the observations of c_k using only the attributes of A_k^+.

4.1 Semilattice on object values

In this section, we deal with the values of objects, we construct a meet semilattice structure on the values space of objects. Assume that is given a set of basic attributes names, $A_Q = \{(A_q\}_{q\in Q}$ and their corresponding domains $D_Q = \{D_q\}_{q\in Q}$. For any object o, a basic attribute is valued in D_q only if the attribute is present.

Denote by $\Gamma_q = D_q \cup \{\perp\} \cup \{*\}\}$ where $\perp$ is interpreted as "undefined" or "not applicable" values and will be used as the values for missing basic attributes, $*$ means that the value is unknown because the corresponding basic attribute is unknown. Denote by $\Gamma_Q = \sqcap_{q\in Q}\Gamma_q$. We assume that each set $D_q \in \mathcal{D}_Q$ is a meet semilattice according to the type of the basic attribute, it means that :

$$v_q, v'_q \in D_q \Rightarrow v_q \wedge v'_q \in D_q.$$

Consider $(\Gamma_q; <, \wedge, *)_{q\in Q}$ as the meet semilattice with $*$ as the minimum and the element $\perp$ is not comparable with $v_q \in D_q, v_q \wedge \perp = *$. Then $\Gamma_Q = \sqcap_{q\in Q}\Gamma_q$ is a meet semilattice as product of meet semilattice such as, for

$$v = (v_q)_{q\in Q}, v' = (v'_q)_{q\in Q} \in \Gamma_Q : v \wedge v' = (v_q \wedge v'_q)_{q\in Q} \in \Gamma_Q.$$

For example, for any categorical attribute (A_q, D_q), we will consider $D_q \cup \perp$ as an antichain, and the meet semilattice Γ_q has $*$ as minimum. If the type of a basic attribute is real interval, the domain is the set of values $u = [\underline{u}, \overline{u}]$ with $\underline{u}, \overline{u} \in \mathbb{R}$ such that $\underline{u} \leq \overline{u}$. The order relation chosen is the dual order of $\subset$, and the $\wedge$ operator is such that $u \wedge v = [\underline{u} \wedge \underline{v}, \overline{u} \vee \overline{v}]$.

The partial valuation function v_q related to the basic attribute A_q associates to each object o, a value $v_q \in \Gamma_q$ such as:

$$\sigma(A_q) = * \iff v_q = *;\ \sigma(A_q) = - \iff v_q = \perp;\ \sigma(A_q) = + \iff v_q \in D_q.$$

The valuation function $v : O \to \Gamma_Q$ is such as;

$$v(o) = (v_q(o))_{q\in Q} \text{ with } v_q(o) \in \Gamma_q$$

The value context is $\mathbb{K}_v = (O, \Gamma_Q, v)$.

Let $\delta : O \to \Gamma = \mathcal{H} \times \Gamma_Q$ be the mapping $d \times v$ which associates every object o with its skeleton $d(o) = H_{\sigma_o}$ and its values $v(o) = (v_q)_{q\in Q}$ taken on the basic attributes:

$$\delta(o) = d \times v(o) = (H_{\sigma_o}, v(o)) \in \Gamma = \mathcal{H} \times \Gamma_Q.$$

The conditions that the values must verify, lead us to

Definition 3. *Let $H_\sigma \in \mathcal{H}$ be a skeleton and let $v = (v_q)_{q\in Q} \in \Gamma_Q$, and let A_q be any basic attribute. (H_σ, v) is said to be consistent if σ and v satisfies the following conditions:*

$$\sigma(A_q) = * \iff v_q = *;\ \sigma(A_q) = - \iff v_q = \perp;\ \sigma(A_q) = + \iff v_q \in D_q.$$

In previous sections, we have shown how to provide the skeleton set $\mathcal{H}$ and Γ_Q with a meet semilattice structure. The description space $\Gamma = \mathcal{H} \times \Gamma_Q$ is a meet semilattice as a product of the meet semilattices $\mathcal{H}$ and Γ_Q. The greatest lower bound of the description of o and o' is written:

$$\delta(o) \wedge \delta(o') = (H_{\sigma_o \wedge \sigma_{o'}}, v(o) \wedge v(o'))$$

we shall ask the question : is this description consistent ? The next proposition shows that $\wedge$ preserves the consistency property

Proposition 3. *If (H_σ, v) and $(H_{\sigma'}, v')$ are consistent then $(H_{\sigma\wedge\sigma'}, v \wedge v')$ is consistent.*

Proof. Let $v = (v_q)_{q\in Q}$ and $v' = (v'_q)_{q\in Q}$ be values related to consistent descriptions (H_σ, v) and $(H_{\sigma'}, v')$. Assume A_q a basic attribute such that $\sigma\wedge\sigma'(A_q) = *$. Then, the first possibility is $\sigma(A_q) = \sigma'(A_q) = * \iff v_q = v'_q = *$, because the descriptions are consistent, then we have $v_q \wedge v'_q = *$. Or A_q is missing in one skeleton and present in the other one. Let us suppose that $\sigma(A_q) = + \iff v_q \in D_q$, and $\sigma'(A_q) = - \iff v'_q = \perp$. We always have $v_q \wedge v'_q = v_q \wedge \perp = *$, and conversely, if A_q is a basic attribute such that $\sigma \wedge \sigma'(A_q) = -$, then $\sigma(A_q) = \sigma'(A_q) = -$, and $v_q = v'_q = \perp$,then we have $v_q \wedge v'_q = \perp$. $(H_{\sigma\wedge\sigma'}, v \wedge v')$ is consistent.

4.2 Concepts

The goal of the previous sections has been to define a complex context $\mathbb{K} = (O, \Gamma, \delta)$ for structured, and multi-valued and incomplete data.

Let $\Gamma(O)$ be the meet semilattice generated by the descriptions of the objects $\Gamma(O) = \{\wedge_{l\in L}\delta(l)|L \subset O\} = \{\wedge_{l\in L}(H_{\sigma_l}, v(l))|L \subset O\}$. Consider the semilattice $(\Gamma(O), <, \wedge)$ and $(\mathcal{P}(O), \subset, \cup)$. The pair $GC = (int, ext)$ of maps
$ext : \Gamma(O) \longrightarrow \mathcal{P}(O)$ and $int : \mathcal{P}(O) \longrightarrow \Gamma(O)$, is a GC such as, for $L \in \mathcal{P}(O)$:

$$int(L) = \bigwedge_{l\in L}(H_{\sigma_l}, v(l)) = (H_{\wedge_{l\in L}\sigma_l}, \wedge_{l\in L} v(l))$$

which is a consistent description according to the previous proposition, and for any $(H_\sigma, \nu) \in \Gamma(O)$:

$$ext((H_\sigma, \nu)) = \{o \in O|(H_\sigma, \nu) \leq (H_{\sigma_o}, v(o))\} = \{o \in O|H_\sigma \leq H_{\sigma_o}, \nu \leq v(o)\}.$$

The set of concepts induced by GC is denoted by $\mathbb{C}$. The relationship between skeleton concepts of $\mathcal{C}$ and complex concepts of $\mathbb{C}$ is made precise below:

Proposition 4. *Let L be a set of objects, $\sigma_L = \wedge_{l\in L}\sigma_l$ and $v_L = \wedge_{l\in L} v(l)$. If $\gamma = (H_{\sigma_L}, L) \in \mathcal{C}$ is a skeleton concept, and if we have for any basic attribute $Aq, \sigma_L(A_q) \neq +$ then $\Upsilon = ((H_{\sigma_L}, v_L), L)$ is a complex concept of $\mathbb{C}$.*

Proof. The proof is easy as we can notice, that all basic values are missing or unknown if the corresponding basic attributes are missing or unknown.

5 Application: concepts extraction from coral base

A straight-forward application is conducted on coral description base. The concepts lattice is extracted from information about the structure of objects. One concept is exhibited with its resulting multi-values properties. We consider for this application a subset of 10 descriptions extracted from the coral genera base (16 families, 58 genera, 185 species). The whole knowledge base is actually made of 10 models corresponding to the 10 main coral families present in the south-west of the Indian ocean and about two thousands descriptions (see Figure 1). In order to use classical FCA methods, each structured attribute B is coded by two binaries attributes $B+$ and $B-$ to express the presence or absence of a component. For a given object o, an unknown state is represented by $B+(o) = B-(o) = 0$. Following table shows the resulted context:

			A	**B**	**C**	**D**	**E**	**F**	**G**
Object n°	**Family**	**Genus**	Monocentric corralites	Coenosteum	Pluricentric corralites	Septal teeth	Synapticuls	Intercorallites pillars	Columella
1	Astrocoeniidae	Stylocoeniella	+	+	-	+	-	+	+
2	Pocilloporidae	Pocillopora	+	+	-	+	-	-	+
3	Pocilloporidae	Stylophora	+	+	-	+	-	-	+
4	Pocilloporidae	Seriatopora	+	+	-	+	-	-	+
5	Pocilloporidae	Madracis	+	+	-	+	-	-	+
6	Siderastreidae	Psammocora	+	+	-	+	+	+	+
7	Siderastreidae	Siderastrea	+	-	-	+	+	-	+
8	Fungiidae	Fungia	+	-	-	+	+	-	+
9	Faviindae	Faviinea Leptoria	-	-	+	-	-	-	-
10	Acroporidae	Acropora	+	+	-	+	-	-	+

Fig. 3. An example of corals data set

We used the Galicia platform [12] and the Bordat algorithm [2] with the classical inf operator $\wedge_{S^{\mathcal{A}}}$ to build the concepts semilattice (Figure 4) on the previous context. Each concept is presented with its intension, extension and the associated skeleton. We verify that C2, C3, C8 and C9 are inconsistent concepts: for them, the structured attribute A is undefined whereas at least one of its subpart is defined. At this stage, a concept regroups objects having similar skeletons. The interest to use the consistent inf operator $\wedge$ (see proposition 1) is that inconsistent concepts are not computed. The concept C11 groups the different Pocilloporidae family's genera and the quite near genus Acropora of the Acroporidae family. From the expert's point of view, this analysis is meaningful to organize taxonomies, according to M. Pichon, an nternational coral expert. From the extension of skeleton concepts, further analysis such as Concept Analysis on multivalued contexts or clustering methods can be performed. For example, Figure 5 gives the intension of the concept C11 computed, using IKBS system, from the complete objects description of the extension of C11.

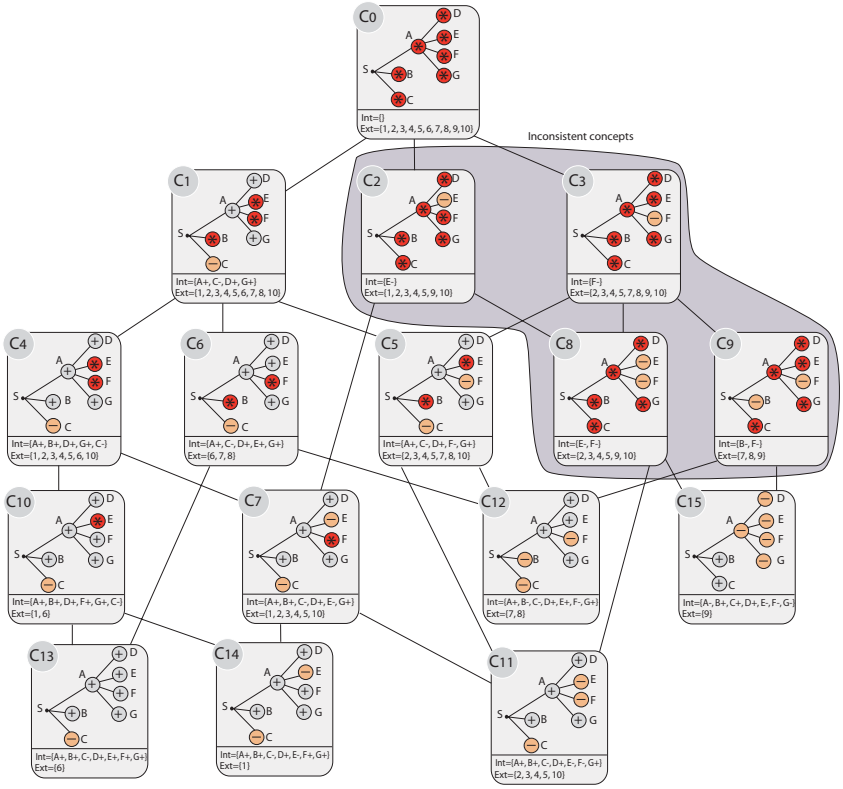

Fig. 4. Concepts semilattice build upon structured objects with $\wedge_{\mathcal{SA}}$.

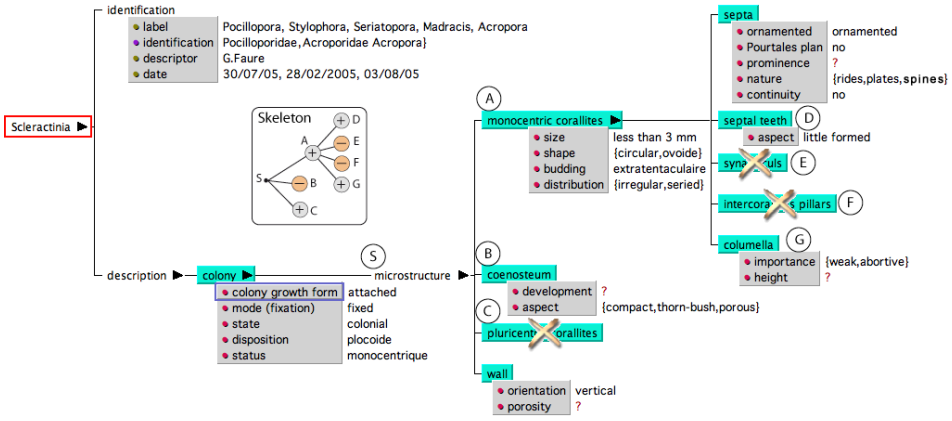

Fig. 5. Intension of the concept C11

6 Conclusion

In this paper, we presented an approach which allows Concept Analysis to deal with structured, multivalued and incomplete data. This kind of analysis is useful to extract knowledge from observations in Life Sciences and to help experts in the Knowledge Bases building process. However, the number of consistent concepts generated may be huge due to model's complexity. We are exploring strategies to reduce the concepts research space by using datamining methods such as clustering or suitable distances on structured and multi-valued objects.

References

[1] M. Barbut and B. Monjardet. *Ordre et classification.* Hachette, Paris, 1970.

[2] J.P. Bordat. Calcul pratique du treillis de galois d'une correspondance. In *Mathématiques et Sciences humaines, 96*, pages 31–47, 1986.

[3] P. Cimiano, A. Hotho, G. Stumme, and J. Tane. Conceptual knowledge processing wih formal concept analysis and ontologies. In P.W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis (ICFCA 2004), Sydney, Australia*, LNCS 2961, pages 189–207, 2004.

[4] N. Conruyt and D. Grosser. Knowledge engineering in environmental sciences with ikbs: Application to systematics of corals of the mascarene archipelago. *AI Communication*, 16(4):267–278, 2003.

[5] Grosser D. *Construction itérative de bases de connaissances descriptives et classificatoires avec la plate-forme à objets IKBS : application à la systématique des coraux des Mascareignes.* Thèse de doctorat, Université de la Réunion, 2002.

[6] R. Emilion E. Diday. Maximal and stochastic galois lattices. *Discrete Applied Mathematics*, 27(2):271–284, 2003.

[7] B. Ganter and R. Wille. *Formal concept analysis, Mathematical foundations.* Springer Verlag, Berlin, 1999.

[8] Bernhard Ganter and Sergei O. Kuznetsov. Pattern structures and their projections. *Lecture Notes in Computer Science*, 2120:129–142, 2001.

[9] Le Renard J. and Conruyt N. On the representation of observational data used for classification and identification of natural objects. *LNAI IFCS'93*, pages 308–315, 1994.

[10] H.Ralambondrainy J. Diatta. The conceptual weak hierarchy associated with a dissimilarity measure. *Mathematical Social Sciences*, pages 301–319, 2002.

[11] Burmeister P. and Holzer R. Treating incomplete knowledge in formal concept analysis. *LNAI 3626*, pages 114–126, 2005.

[12] C. Roume P. Valtchev, D. Grosser and M. Rouane Hacene. Galicia, an open platform for lattices. In *Proceedings of the 11th International Conference on Conceptual Structures (ICCS'03)*, pages 241–254. Shaker Verlag, 2003.

[13] P. Valtchev, R. Missaoui, and R. Godin. Formal concept analysis for knowledge discovery and data mining: The new challenges. In P.W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis (ICFCA 2004), Sydney, Australia*, LNCS 2961, pages 352–371. Springer, 2004.

[14] R. Wille. Methods of conceptual knowledge processing. In R. Missaoui and J. Schmid, editors, *International Conference on Formal Concept Analysis (ICFCA 2006), Dresden, Germany*, LNCS 3874, pages 1–29. Springer, 2006.