

# Using Concept Formal Analysis for Cooperative Information Retrieval

Ibtissem Nafkha<sup>1</sup>, Samir Elloumi<sup>1</sup>, and Ali Jaoua<sup>2</sup>

<sup>1</sup>University of Tunis, Department of Computer Science, Campus Universitaire,  
Le Belvédère, 1060, Tunis, Tunisia. Phone/Fax: +216 71 885190

<sup>2</sup>University of Qatar, Faculty of Sciences,

Department of Computer Science, Doha, Qatar.

<sup>1</sup>{ibtissem.nafkha,samir.elloumi}@fst.rnu.tn, <sup>2</sup>jaoua@qu.edu.qa

**Abstract.** The potentials of formal concept analysis (FCA) for information retrieval (IR) have been highlighted by a number of research studies since its inception. The growth of the web has favoured the emergence of new search applications. In this paper, we will focus on the unique features of FCA for searching in distributed information and for reducing the size of the set information. The development of a FCA-based applications for distributed information returns a major gain and the obtained results are promising. This study has several perspectives for real and fuzzy data.

## 1 Introduction

Along with the growth of the world wide web, information retrieval systems gain importance since they are often the only way to find the few documents actually relevant to a specific question in the vast quantities of text available. Moreover, with the advent of the Web along with the unprecedented amount of information available in electronic format and its distributed structure, Formal Concept Analysis (FCA) is more useful and practical than ever, because this technology addresses important limitations of the systems that currently support users in their quest for information. Over the last few years, the range of functionality has been expanded to include new tasks such as data reduction and collaborative (or cooperative) information retrieval. In fact, due to the huge quantity of available information and its distributed structure, it is necessary to abstract it and eliminate the redundancy data. In this context, a method for data reduction based on the formal concept analysis is proposed in [16,17]. At the same time, new IR domains have been investigated including different types of information (email messages, web documents,...). Thus, there is nowadays a much better awareness of the strengths and limitations of this technique for organising and searching distributed information. We are interested by searching in distributed information. So, this paper is organized as follows. In section 2, we introduce some basic definitions on formal analysis. Then in section 3, we present the data reduction. The section 4 is devoted to the presentation of several methods for searching in distributed information. In section 5, we present the complexity of those methods.

## 2 Mathematical Foundations

Among the mathematical theories found recently with important applications in computer science, lattice theory has a specific place for data organization, information

engineering, data mining and for reasoning. It may be considered as the mathematical tool that unifies data and knowledge or information retrieval [1,2,3,4,5,6,7,8,9,10,18,19,20]. In this section, we define formal context, formal concept, Galois connection and the lattice of concepts associated to the formal context.

**2.1 Formal Context**

*Definition 1:* A formal context is a triple  $k = \langle O, P, R \rangle$ , where  $O$  is a finite set of elements called objects,  $P$  a finite set of elements called properties and  $R$  is a binary relation defined between  $O$  and  $P$ . The notations  $(g, m) \in R$ , or  $R(g, m) = 1$ , mean that “formal object  $g$  verifies property  $m$  in relation  $R$ ” [12,21].

*Example 1:* Let  $O$  be a set of some animals and  $P$  be a set of the properties such as :  $a =$  needs water,  $b =$  lives in water,  $c =$  lives on land,  $d =$  can move around. Let  $O = \{Leech, Bream, Frog, Dog\}$  and  $P = \{a, b, c, d\}$ . The following context may be defined by the table 1.

		a	b	c	d
1	Leech	1	1	0	1
2	Bream	1	1	0	1
3	Frog	1	1	1	1
4	Dog	1	0	1	1

Table 1 : An example of a formal context.

**2.2 Galois Connection**

*Definition 2:* Let  $A \subseteq O$  and  $B \subseteq P$  two finite sets,  $R$  a relation on  $O \times P$ . For both sets  $A$  and  $B$ , operators  $f(A)$  and  $h(B)$  are defined as :

$$f(A) = \{m \mid \forall g, g \in A \rightarrow (g, m) \in R\}$$

$$h(B) = \{g \mid \forall m, m \in B \rightarrow (g, m) \in R\}$$

Operator  $f$  defines the properties shared by all elements of  $A$ . Operator  $h$  defines objects sharing the same properties included in set  $B$ . Operators  $f$  and  $h$  define a Galois Connection between sets  $O$  and  $P$  [12].

*Proposition 1:* Operators  $f$  and  $h$  define a Galois connection between  $O$  and  $P$ , such that if  $A_1, A_2$  are subsets of  $O$ , and  $B_1, B_2$  are two subsets of  $P$ , then  $f$  and  $h$  verify the following properties [12] :

- $A_1 \subseteq A_2 \Rightarrow f(A_1) \supseteq f(A_2)$  (1)
- $B_1 \subseteq B_2 \Rightarrow h(B_1) \supseteq h(B_2)$  (2)
- $A_1 \subseteq h \circ f(A_1)$  and  $B_1 \subseteq f \circ h(B_1)$  (3)
- $A \subseteq h(B) \Leftrightarrow B \subseteq f(A)$  (4)
- $f = f \circ h \circ f$  and  $h = h \circ f \circ h$  (5)

### 2.3 Formal Concept

*Definition 3:* A formal concept of the context  $\langle O,P,R \rangle$  is a pair  $(A,B)$ , where  $A \subseteq O$ ,  $B \subseteq P$ , such  $f(A) = B$  and  $h(B) = A$ . Sets  $A$  and  $B$  are called respectively the domain (extent) and range (intent) of the formal concept.

### 2.4 Concept Lattice

*Definition 4:* From a formal context  $\langle O,P,R \rangle$ , we can extract all possible concepts. In [12], we prove that the set of all concepts may be organized as a lattice, when we define the following partial order relation  $\ll$  between two concepts,  $(A_1,B_1) \ll (A_2,B_2) \Leftrightarrow (A_1 \subseteq A_2) \text{ et } (B_2 \subseteq B_1)$ . The concepts  $(A_1,B_1)$  and  $(A_2,B_2)$  are called nodes in the lattice.

### 2.5 Equivalence between an object and a subset of other objects

*Definition 5:* Let  $x \in O$  be an object,  $A \subseteq O$  be a finite set and  $R$  a relation on  $O \times P$ . We define  $\text{closure}(x) = g(f(x))$  and  $\text{closure}(A) = g(f(A))$  [16,17].

*Definition 6:* Let  $x \in O$  be an object,  $A \subseteq O$  be a finite set and  $R$  be a relation on  $O \times P$ . We say that an object  $x$  is equivalent to the objects  $A$ , relatively to a relation  $R$ , if and only if,  $\{x\} \cup A$  is a domain of a concept of  $R$ , and that the closure  $(x) = \text{closure}(A) = \{x\} \cup A$ , where  $x \notin A$ .

*Example 2:* Let  $R$  be the following relation, in the table, with 5 objects  $\{O1,O2,O3,O4,O5\}$  and three attributes  $\{A,B,C\}$ .

	A	B	C
O1	1	1	1
O2	1	0	1
O3	1	0	0
O4	0	1	1
O5	0	0	1

Table 2 : Example of a relation R

O5 is equivalent to  $\{O1,O2,O4\}$ , the reason is that the concept containing O5 is  $CP = \{O1,O2,O4,O5\} \times \{C\}$  ; and inversely the concept containing  $\{O1,O2,O4\}$  is also CP.

## 2 Current Search

Using FCA can complement the existing search systems to address some of their main limitations. Basically, FCA exploits the similarity between documents in order to offer an automatically support structure (i.e., the document lattice) in which to place the information retrieval process. The document lattice can be used to improve basic individual search strategies [1,2,4,13]. Moreover, query refinement is one of the

most natural applications of concept lattices. Its main objective is to recover from the null-output or the information overload problem. The concept lattice may be used to make a transformation between the representation of a query and the representation of each document [5,6,7,8,9]. The query is merged into the document lattice and each document is ranked according to the length of the shortest path linking the query to the document concept. In the other hand, on the set of terms describing the document, there exist hierarchies in the form of thesaurus [4,10,13,14]. The information searching using FCA takes as input a query that will be forwarded to a selected search engine [6,7,8]. The first pages retrieved by the search engine in response to the query are collected and parsed. At this point, a set of index units that describe each returned document is generated; such indices are next used to build the concept lattice corresponding to the retrieved results. The last step consists of showing the lattice to the user and managing the subsequent interaction between the user and the system. In spite of such limitations such as for larger information collection, generally we get a huge number of references, we are interested to build a FCA-based system for distributed information, which may affect both the efficiency and the effectiveness of the overall system. These limitations can be overcome by using data reduction that will be described in the next section.

### 3 Conceptual Reduction Method

The fundamental question for data reduction is the following : How can we reduce data without losing any knowledge ? The advantage of reduced data is that it can be used directly as a prototype for making decision, for supervised learning or reasoning [16,17]. The proposed algorithm is based on the elimination of any object that may be replaced by a subset of the equivalent objects [17]. Firstly, we use relational join operator in order to obtain a total relation. Secondly, we apply reduction algorithm on the final table to minimize the number of rows. Let  $R$  be the initial binary relation, representing the database, and  $RD$  be the expected output of the equivalent reduced concept. We describe in detail the different steps of the algorithm.

Algorithm reduction ( $RD, R$ )

- Initialize  $RD=R$
- For each object  $x$  in the domain of the remaining context  $RD$ , we do the following steps :
  - 1) we find the set of its properties  $P$  (i.e. subset of attributes satisfied by  $x$ ).
  - 2) we find the set of objects  $S$ , except  $x$ , sharing all the properties  $P$ .
  - 3) if  $S$  is not empty, we check if object  $x$  is included in the set of objects sharing the same properties as  $S$ . In the positive case, object  $x$  is removed from context  $RD$ .

End of the loop

End.

This method takes place on two steps. The first step consists in applying the relational join operator on the set of the tables. The second step is the execution of the conceptual reduction algorithm on the result table. The major problem is that the

treatment of an important size databases needs a considerable storage space and makes the reduction process sultry. Our idea is to propose an algorithm which makes simultaneously the join and the reduction of the tables set in only one reduced table. Then, we will present our system in the next section.

## 4 Search enhanced by FCA

In this section we will present some search that may be improved through a FCA for distributed information.

### 4.1 Cooperative conceptual Information Retrieval System

To improve upon regular Information Retrieval System (IRS), cooperative conceptual information retrieval system (C2IRS) was proposed [18]. As it is shown in figure 1, C2IRS is composed of two parts :

1. Part 1 is the cooperative information retrieval system that we present in the next subsection. In this phase, C2IRS uses a conceptual approach in the searching process from different databases. Each local result of each CIRS is a concept that we store in an *Answer Array*. We assume that we have N different formal contexts, so we can get a maximum of N concepts in the *Answer Array*.
2. Part 2 is the final Answer Formulation that we describe in subsection 4.1.2. First of all, we apply the *Merging1 Algorithm* on the *Answer Array*. Then, we apply the *Merging2 Algorithm* on the modified *Answer array* in order to obtain the final answer. We detail those algorithms with an illustrative example.

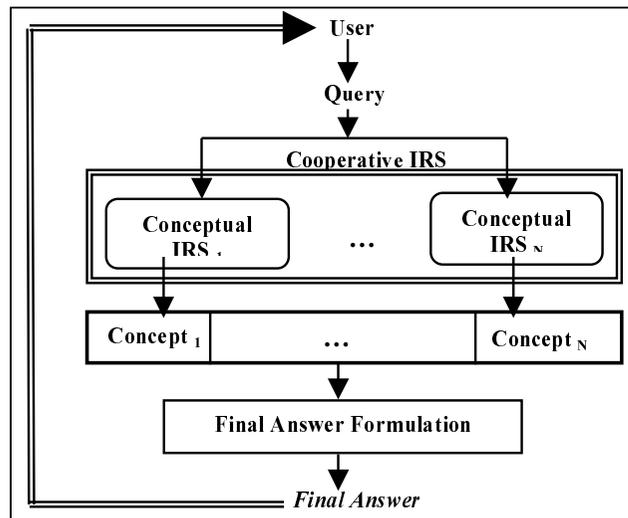


Figure 1: Architecture of C2IRS

**Cooperative Information Retrieval System.** Different information retrieval systems (IRS) cooperate to give a most complete answer to a query. Each IRS has a local database on which we apply the Galois connection (GC) to retrieve the documents

satisfying the query. In the query, we express that we want to find documents which are indexed by some indexing terms  $Qr$ . To resolve a query ( $Qr$ ), each CIRS executes the *Retrieve Algorithm* proposed in [18] on its local database ( $DBi$ ). We apply the Galois connection only on the query terms  $Tl$  existing in the local database. Thus, the result of each CIRS, stored in the *Answer array*, is a concept which domain is the local searching terms  $Tl$  and its codomain is the retrieved documents.

*Example 3* : Let the following illustrative example. Let the set of databases 1, 2 and 3 presented by formal contexts in figure 2. For the query of the form : "Which documents are indexed by the terms  $t_2, t_3$  and  $t_5$ ", the query is formed by three terms  $t_2, t_3$  and  $t_5$ .

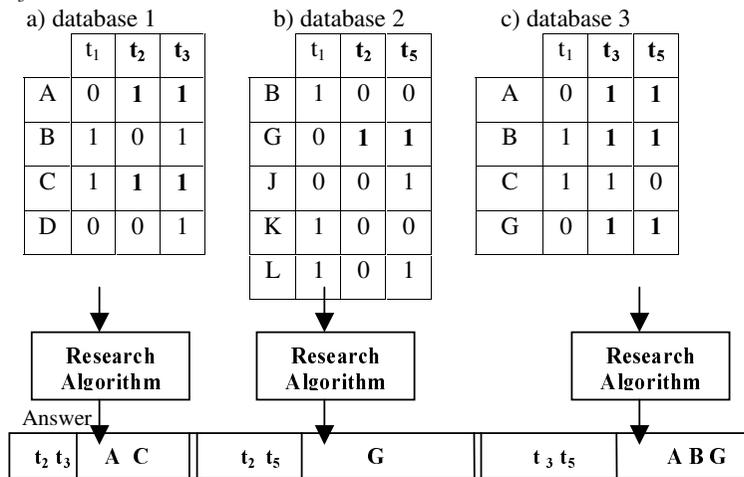


Figure 2 : Cooperative Information Retrieval System

We apply the *Retrieve Algorithm* already presented on the set of databases. The Galois connection applied on the set of query terms, existing in the first database,  $Tl=\{t_2,t_3\}$ , gives the set of documents  $\{A,C\}$ . Then, we put the first concept found  $(\{t_2,t_3\},\{ A, C\})$  in the first cell of the *Answer* array.

By the same way, we continue to find concept from each contexts . We obtain four concepts from the several databases that we place in the following *Answer* array.

	<b>1</b>	<b>2</b>	<b>3</b>
$t_2 t_3$	$t_2 t_5$	$t_3 t_5$	
A C	G	A B G	

Table 3 : THE *ANSWER* ARRAY.

Based on this array, we construct the final answer by the application of the *Final\_Answer\_Formulation* algorithm proposed in [18]. We detail the steps of this algorithm in the next subsection.

**Final Answer formulation.** In this section, we present the second part of our system which is the final answer formulation. The formulation of the final answer consists in the application of two proposed algorithms on the *Answer* array. This algorithm combines the concepts of the *Answer* array according to some conditions. The final answer is built somehow iteratively. Initially, the final answer is an the empty set. We read the set of concepts element by element. For each element, if the domain of a concept is equal or greater than the terms of the query, then we add its range (set of document references) to the final answer. If this condition is not satisfied, we build intersection between its range and ranges of other concepts in the answer array (i.e. a subset of documents), and we compute the union of the domains (to find a subset of terms). We continue to build these intersections until we find at least all the terms of the query.

*Example 4 :* For our example, we remind that our query is  $\{t_2, t_3, t_5\}$ . Initially, the final answer is an empty set. We read the first concept of *Answer* array. Their terms are different to the query. So, we merge their terms with terms of the second concept and we compute the intersection between their ranges. The union of terms is the set  $\{t_2, t_3, t_5\}$  which is equal to the query. But the result of the intersection between documents is an empty set. So, we continue with the next concept. We merge the terms of the first and the last concept. The result is the set  $\{t_2, t_3, t_5\}$  which is equal to the query. The intersection between their documents gives the set of documents  $\{A\}$ . We add this set to the final answer. By repeating this process with to the next concept, we find the union of terms  $\{t_2, t_5, t_3\}$  which is equal to the query. So, we add the result of the intersection of documents that is the set  $\{G\}$  to the final answer. The final answer is now the set  $\{A, G\}$ . The final answer obtained is the set of documents  $\{A, G\}$  that will be deliver to the user.

The retrieval process using this system is based on a formal concept analysis. For each database, we search the local concept. We use a cooperative approach to formulate the final answer basing on extracted concepts from several databases. The major problem is that when we search for pertinent information, we generally get a huge number of references. Moreover, the treatment of an important size databases needs a considerable storage space and makes the research process sultry. So, we need to minimize the size of each database. In the next section, we present the cooperative conceptual information retrieval system based on data reduction.

#### 4.2 Cooperative Conceptual Information Retrieval System based on Data Reduction

To improve upon cooperative conceptual information retrieval system, we propose a system for cooperative conceptual information retrieval basing on data reduction. As it is shown in figure 3, our proposed system is composed of two parts :

1. Part 1 is detailed in the next subsection. In this phase, our proposed system uses a conceptual approach, proposed in [17], in the reducing process of different databases. Each database is reduced in order to minimize its size and obtain a reduced database (RDB) and the equivalent objects set.
2. Part 2 is a search process which will be described in the next subsection. Basing on the set of reduced databases (RDB) and the equivalent objects set,

we cooperate to give a most complete answer to a query by using the cooperating process, proposed in [18], that will be detailed with an illustrative example.

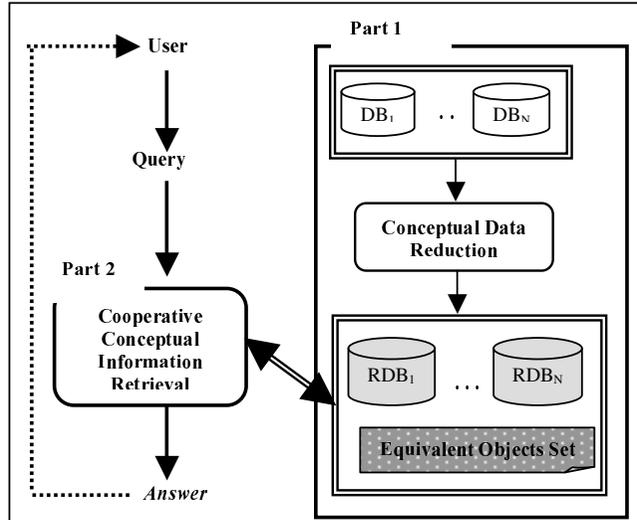


Figure 3 : Cooperative Conceptual Information retrieval based on data reduction

**Conceptual data reduction.** In [17], an accurate algorithm for data reduction is proposed. This algorithm assumes that we can minimize data without losing any knowledge. This algorithm is based on the elimination of any object that may be replaced by a subset of equivalent objects as defined in [17]. We describe in detail the different steps of the algorithm and we show its application on the example. Let  $R$  be the initial binary context, representing the database, and  $RD$  be the expected output of the equivalent reduced concept.

*Example 5 :* Let the following databases presented by the formal contexts in figure. For the database 1, the document  $D$  is equivalent to  $\{B,C,A\}$ , the reason is that the concept containing  $D$  is  $C1=\{A,B,C,D\} \times \{t_3\}$ ; and inversely the concept containing  $\{A,B,C\}$  is also  $C1$  that may be obtained by using the Galois connection. This means that document  $D$  can be removed without modifying the initial knowledge database. By the same way, we can remove  $B$  and  $J$  from the database 2. Moreover, we remove the document  $A$  from the database 3. The reduced databases are showed in the following table.

	$t_1$	$t_2$	$t_3$
A	0	1	1
B	1	0	1
C	1	1	1

	$t_1$	$t_2$	$t_5$
G	0	1	1
K	1	0	0
L	1	0	1

	$t_1$	$t_3$	$t_5$
B	1	1	1
C	1	1	0
G	0	1	1

Table 4 : The reduced databases.

Based on the reduced databases and the set of equivalent documents, the cooperative conceptual search process is executed to respond to the query. We detail the steps of this system in the next subsection.

**Cooperative Conceptual Information Retrieval.** In this section, we present the second part of our system which is a cooperative conceptual information retrieval which is proposed in [18]. Our system is formed by several conceptual information retrieval systems (CIRS) that cooperate to give a most complete answer to a query. Each one has a local reduced database on which we apply the Galois connection to retrieve the sites satisfying the query. In the query, we express that we want to find sites which are indexed by some indexing properties. We apply the Galois connection only on the query properties existing in the local reduced database. Thus, the result of each CIRS is a concept which domain is the local searching properties and its range is the retrieved sites. From this set of concepts, we formulate the final answer by applying a merging and a combining algorithms proposed in [18].

*Example 6 :* Let the reduced databases presented in table 4. For the query, treated in the last section, of the form : "Which documents are indexed by the properties  $t_2$ ,  $t_3$  and  $t_5$ ", the query is formed by three properties  $t_2$ ,  $t_3$  and  $t_5$ . We will find three concepts from reduced databases by applying Galois connection. The reduced database 1 contains only the properties  $t_2$  and  $t_3$ . So, the first concept is  $\{t_2, t_3\} \times \{A, C\}$ . The concepts founds from the second and the third reduced databases are simultaneous  $\{t_2, t_5\} \times \{G\}$  and  $\{t_3, t_5\} \times \{B, G\}$ . In order to formulate the final answer, we combine these concepts by applying merging and combining algorithms. The final answer is built somehow iteratively. Initially, the final answer is an the empty set. We treat the set of concepts element by element. We read the first concept, their properties are different to the query. So, we merge their properties with properties of the second concept and we compute the intersection between their ranges. The union of properties is the set  $\{t_2, t_3, t_5\}$  which is equal to the query. The intersection of documents is empty. At this stage, the final answer is an empty set. Then, we compute the union of properties and the intersection of documents between the first and the third concept. The result of this merge is the set  $\{t_2, t_3, t_5\}$  which is equal to the query. The result of the intersection between documents is an empty set. So, we continue with the next concept. We merge the properties of the second and the last concept. The result is the set  $\{t_2, t_3, t_5\}$  which is equal to the query. So, we add the result of the intersection of documents that is the set  $\{G\}$  to the final answer. The final answer is now the set  $\{G\}$ . The final answer obtained is the set of documents  $\{G\}$  that will be delivered to the user. The answer can be enriched by the additional information existing in the equivalent objects set. We have A is equivalent to the document G. The enriched final answer is so  $\{G, A\}$  that will be delivered to the user. We remark that we obtain the same answer as the cooperative conceptual information retrieval system presented in the last section that its databases contain in total 13 lines. Moreover, our proposed system is applied on the reduced databases which have only 9 lines that is more easy. Our system handles bases which have a small sizes resulting of the conceptual reduction of the initial bases having an important size. Thus, our

proposed system uses a storage space less important than that used by the cooperative conceptual information retrieval system and returns then the quicker research process. The major problem is that the treatment of databases which have an important size needs a considerable storage space and makes the reduction process sultry. Our idea is to propose an algorithm which makes simultaneously the join and the reduction of the tables set in only one reduced table. In the next section, we present our system.

#### 4.3 Conceptual information retrieval system based on cooperative conceptual data reduction

In this section, we present an approach for information retrieval [19] that is composed of two parts as it is shows in the following figure :

3. Part 1 : From a set of formal contexts representing the databases, we apply the proposed algorithm for *cooperative conceptual data reduction* in order to obtain the reduced table and the set of equivalent objects.
4. Part 2 : The research is based on this reduced table. To respond to a query that is a set of terms, we apply the Galois connection to retrieve the documents satisfying the query. The final answer can be enriched by the set of equivalent objects.

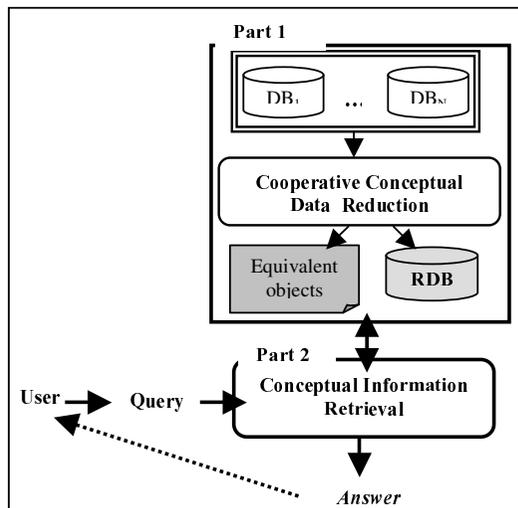


Figure 4 : Architecture of the conceptual information retrieval system based on cooperative conceptual data reduction

The advantage of this approach is that it is much faster than other methods. Indeed, the treatment of databases which have an important size needs a considerable storage space and makes the research process sultry. So, we need to minimize the size of the databases set. Based on the reduced database, the conceptual research process is executed. In the query, we express that we want to find documents which are indexed by some indexing terms. We apply the Galois connection only on the query terms

existing in the reduced documentary database to formulate the answer that will be delivered to the user.

**Cooperative conceptual data reduction.** In this section, we present the principle of our method for the cooperative conceptual data reduction and show how we apply in order to generate the reduced table and the set of equivalent objects. The algorithm proposed in [19] generates the reduced table from a set of tables without using the join operator. It is based on the use of a graph containing the concepts found in each new line addition. In another words, the construction of the concepts lattice is done in an incremental manner. This concepts lattice contains only the significant objects. The reduced table generation from the set of tables is done in an incremental manner. Our algorithm has as input the set of tables T and as output the graph G, the reduced table RT and the set of equivalent objects. It is based on the following proposition.

**Proposition.** Let  $C = (dom, range)$  be a concept and  $C_i = (dom_i, range_i)$ ,  $i = 1..N$ , be its children. Let  $E = dom \setminus (\cup dom_i)$ , where the operator “\” is the minus between two sets. If E is not empty and  $range = (\cap range_i)$ . Then E contains the objects to be eliminated.

We execute the principle of the cooperative conceptual data reduction on the same example to prove that it give the same result as the conceptual reduction data.

*Example 7:* Let the following databases presented in figure 2. We apply now our proposition. The table RT and the graph G are empty. For the first object  $\{A\} \times \{t_2, t_3, t_5\}$  generates a new node **1**, that has as domain  $\{A\}$  and as range is  $\{t_2, t_3, t_5\}$ , that is added to the  $\{$  **1**

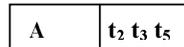


Figure 5 : The graph G.

The line is added to the table RT :

	t <sub>2</sub>	t <sub>3</sub>	t <sub>5</sub>
A	1	1	1

Table 5 : The table RT after adding the first object.

The treatment of the second document  $\{B\} \times \{t_1, t_3, t_5\}$  generates a new node **2**, that is  $(B, \{t_1, t_3, t_5\})$ , in the graph. The treatment with the node **1** gives the intersection  $\{t_3, t_5\}$ . We add a new node **3** that its domain is  $\{A, B\}$  and its range is the set  $\{t_3, t_5\}$ .

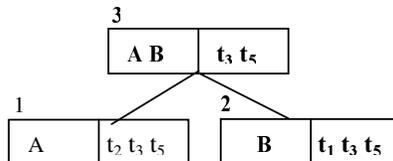


Figure 6 : The graph G.

The line is added to the table RT that becomes :

	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>5</sub>
A	0	1	1	1
B	1	0	1	1

Table 6 : The table RT after adding the second object.

For the object  $\{C\} \times \{t_1, t_2, t_3\}$ , a new node **4** is added to the graph. The treatment with the node **1** gives the intersection  $\{t_2, t_3\}$ . We add a new node **5** that its domain is  $\{A, C\}$  and its range is  $\{t_2, t_3\}$ . The treatment with the node **2** generate a new node **6** that is its domain is  $\{B, C\}$  and its range is the intersection found  $\{t_1, t_3\}$ .

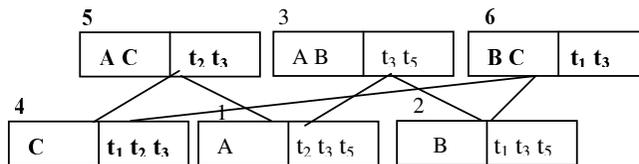


Figure 7 : The graph G.

The line is added to the table. The table is now :

	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>5</sub>
A	0	1	1	1
B	1	0	1	1
C	1	1	1	0

Table 7 : The table RT after adding the third object.

For the object  $\{D\} \times \{t_3\}$ , a new node **7** is added to the graph. The treatment with the node **3** and **5** gives an intersection  $\{t_3\}$  that is its range, it's a modified node. We add to domain the set  $\{A, B, C\}$ . We verify the propriety of the proposition presented. Indeed, we remark that the domain of the node **7** is different from the union of the domains of their children **3**, **5** and **6** ( $\{A, B, C, D\} \not\subset \{A, C\} \cup \{A, B\} \cup \{B, C\}$ ) and its range is equal to the intersection of the ranges of their children ( $\{t_3\} = \{t_2, t_3\} \cap \{t_3, t_5\} \cap \{t_1, t_3\}$ ). So, the object D ( $D = \{A, B, C, D\} \setminus \{A, C\} \cup \{A, B\} \cup \{B, C\}$ ) will be eliminate from the node **7** and from RT. The document D is equivalent to  $\{A, B, C\}$ .

The graph becomes :

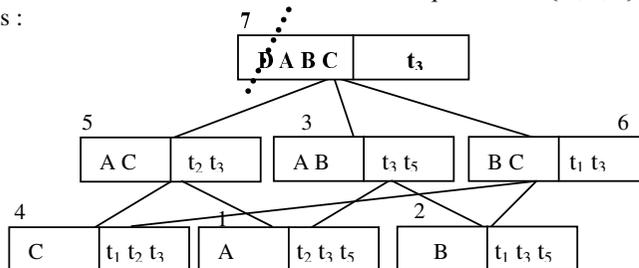


Figure 8 : The graph G.

The new row is added to the RT that becomes :

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>5</sub>
A	0	1	1	1
B	1	0	1	1
C	1	1	1	0

Table 8 : The table RT after adding the new object.

By the same way, we continue the treatment for the others objects. Finally, we obtain the following graph.

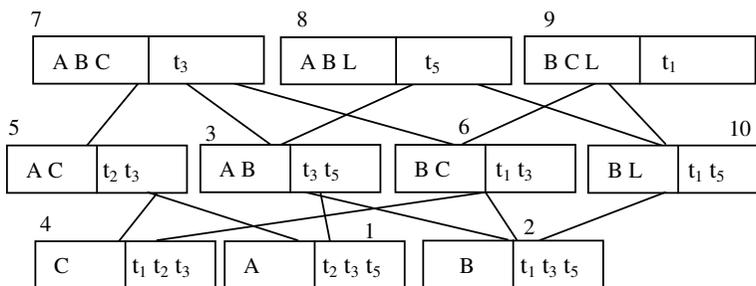


Figure 9 : The graph G.

Finally, we obtain the reduced database RT that is presented in the following table.

	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>5</sub>
B	1	0	1	1
C	1	1	1	0
G	0	1	1	1
L	1	0	0	1

Table 9 : The table RT.

The document D is equivalent to the set of documents {B,C,G}, the document J is equivalent to the set {B,G} and the document K is equivalent to the set {B,C,L}. Basing on the reduced database and the set of equivalent objects, we execute the conceptual research process that is detailed in the next section.

**Conceptual information retrieval.** The information retrieval system (IRS) is based on the reduced database to give a most complete answer to a query. We apply the Galois connection (GC) on the reduced database to retrieve the documents satisfying the query. In the query, we express that we want to find documents which are indexed by some indexing terms  $Q_r$ . To resolve a query ( $Q_r$ ), we apply the Galois connection on the query terms  $T_l$ . Thus, the result of IRS is a concept which range is the searching terms  $T_l$  and its domain is the retrieved documents.

*Example 8 :* Let us treat the databases presented in the table 9. For a query of the form: "Which documents are indexed by the terms  $t_2, t_3$  and  $t_5$ ", the query is formed by

three terms  $t_2$ ,  $t_3$  and  $t_5$ . We will find the concept from reduced database by applying Galois connection. The concept found is  $\{t_2, t_3, t_5\} \times \{G\}$ . The final answer obtained is the document  $G$  that will be delivered to the user. We have in the set of equivalent object the information that the document  $G$  is equivalent to the document  $A$ . So, the final answer will be  $\{A, G\}$ . We remark that we obtain the same result as the cooperative information retrieval based on the data reduction, presented in the last section, that treats three databases containing in total 9 rows. Moreover, our proposed system is applied on the four rows reduced database that is more easy. In the next section, we present the complexity of all presented systems for information retrieval.

## 5 Complexity Analysis

The complexity analysis of a treatment is ensured by calculating :

- i) the time complexity, which is the number of operations of the treatment, and
- ii) the space complexity, which is the number of memories cases used by the treatment.

We compute the complexity analysis of cooperative conceptual information retrieval system, conceptual information retrieval system based on data reduction and conceptual information retrieval system based on cooperative data reduction.

### 5.1 Cooperative Conceptual Information Retrieval System

We present the complexity of the cooperative conceptual information retrieval system (C2IRS) proposed in [11]. We suppose that the database has  $n$  sites and  $m$  properties. We remind first of all the steps of this system:

- Phase 1: the search of concepts from different databases.
- Phase 2: the formulation of the final answer from the found concepts.

**Time Complexity.** We suppose that each database has  $n$  sites and  $m$  properties. The time complexity of the system C2IRS is  $2(n*m) + n$ . So,  $C_T(n, m) = 2nm + n$ .

**Space complexity.** The system C2IRS uses  $k$  matrix with  $n$  lines and  $m$  columns. So, it uses  $(k*n*m)$  memory cells. So, the space complexity of this method is  $C_S = knm$ .

### 5.2 Cooperative conceptual information retrieval information based on Data reduction

We suppose that the database has  $n$  sites and  $m$  properties and the reduced database has  $n'$  sites and  $m'$  properties. We remind first all steps of this system:

- Phase 1: conceptual data reduction applied on several databases.
- Phase 2: cooperative conceptual information retrieval applied on the reduced databases.

**Time complexity.** It is easy to find the complexity  $C'_T(n, m)$  for this system which is equal in :  $C'_T(n, m) = C'_{TPhase 1}(n, m) + C'_{TPhase 2}(n', m')$ . The complexity of phase 1 is  $O(n*m)$ . The phase 2 realize  $(2(n'*m')+n')$  operations [17]. So, the system realizes  $n*m + [2(n'*m')+n']$  operations.

So,  $C'_T(n,m) = nm + 2n'm' + n'$ .

Since the size of the reduced database is lesser than that of the initial database ( $n' \ll n$ ). So, the complexity of our proposed system is lesser than that of the C2IRS .  
 $C'_T(n,m) = nm + 2n'm' + n' \ll C_T(n,m) = 2nm + n$ .

**Space Complexity.** Our retrieval system handles a reduced database. So, we can reserve only  $(k*n'*m')$  memory cells. So, the space complexity of our system is  $C'_S = kn'm'$ .

We note that our algorithm uses a less number of memory cells than the C2IRS :  $C'_S \ll C_S$ . This can be explained by the integration of the conceptual data reduction in our proposed system.

### 5.3 Conceptual information retrieval based on Cooperative data reduction

**Time complexity.** If we have  $K$  tables, each one has  $n$  lines and  $m$  properties, we realize  $N$  iterations where  $N = n^n$ . In each iteration, we cover the concepts graph  $G$  and we do then  $\|G\|$  operations where  $\|G\|$  represents the graph cardinality. So, we do  $\|G\|$  update operations on the graph (adding or modification operation). So ,  $C'_T \approx O(N * \|G\|^2)$ .

Then, our algorithm have a complexity less then that of the conceptual data reduction considering the cardinality of the graph is too small compared to the number of properties ( $\|G\| \ll M$ )

**Space Complexity.** Our algorithm treat in each iteration one line. So, we can reserve only  $M$  memory cases. For the treatment of each line, the algorithm uses a graph that is constructed in incremental manner. This graph necessities  $8*\|G\|$  memory cells since for each graph node, we reserve 8 memory cells. So, the space complexity of our algorithm is  $C'_S = M + (8*\|G\|)$ .

We note that our algorithm uses a less number of memory cells than the conceptual data reduction :  $C'_S \ll C_S$ . This can be to explain by the fact why the conceptual data reduction treats a matrix of  $N$  lines and  $M$  columns on the other hand our algorithm treats only one line.

## 6 Conclusion

In this paper, we have presented three conceptual information retrieval systems. The cooperative conceptual information retrieval system combines the results of several information retrieval systems. Each one has its local formal context representing a database on which we apply a conceptual search to obtain a first set of concepts. From this set of concepts, we formulate the final answer by applying a proposed algorithm merging and combining concepts with a suitable way. The volume of the available information is increasing exponentially. So, generally we get a huge number of reference. It is necessary to abstract it and eliminate the redundancy data. However, our main idea is to integrate the data reduction in the search process. The information retrieval is based on the data reduction. From a set of formal contexts presenting a databases set, we apply the relational join operator in order to fusion the

tables. Basing on the join resulting table, we apply the data reduction algorithm to obtain the reduced table on which the search process is executed. The major problem is that the treatment of an important size databases needs a considerable storage space and makes the reduction and the research process sultry. We are presented an information retrieval based on a cooperative conceptual data reduction. We have developed a fast algorithm, that combines the join and the reduction steps, for minimizing formal contexts and we have apply it in an information retrieval system. From a set of databases, we obtain a reduced database on which we apply a conceptual research for responding to a query. This application was experimented on many databases and the obtained results are promising. This study has several perspectives for real and fuzzy data.

## References

1. Aboud M., Chrisment C., Razouk R., Florence S., Soulé-Dupuy : Query a Hypertext Information Retrieval System by use of Classification. *Information Processing and Management*, 29(3), (1993) 387-396
2. Amati G., Carpineto C. , and Romano G. : FUB at TREC-10 Web Track: A Probabilistic Framework for Topic Relevance Term Weighting. In *Proceedings of the 10<sup>th</sup> Text REtrieval Conference (TREC-10)*, NIST Special Publication 500-250, Gaithersburg, MD, USA (2001) 182-191.
3. Bordat J.P.: Calcul pratique du treillis de Galois d'une correspondance. *Math. Sci. Hum.*, 96, (1986) 31-47.
4. Carpineto C. and Romano G. : Using Concept Lattices for Text Retrieval and Mining. In the first International Conference on Formal Concept Analysis, Darmstadt, Germany, (2003).
5. Carpineto C. and Romano G. : Information retrieval through hybrid navigation of lattice representations. *International Journal of Human-Computer Studies*, 45(5), (1996) 553-578.
6. Carpineto C. and Romano G : A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2), (1996) 1-28.
7. Carpineto C. and G. Romano. Effective reformulation of Boolean queries with concept lattices. In *Proceedings of the 3rd International Conference on Flexible Query-Answering Systems*, pages 83-94, Roskilde, Denmark, 1998.
8. Cole R. and Eklund P. : Browsing semi-structured web texts using formal concept analysis. In *Proceedings of the 9th International Conference on Conceptual Structures*, Stanford, CA, USA, (2001) 319-332.
9. Efthimiadis E. : Query expansion. In M. E. Williams, editor, *Annual Review of Information Systems and Technology*, v31, American Society for Information Science, Silver Spring, Maryland, USA, (1996) 121-187.
10. Ferrfie S. and Ridoux O. : A file system based on concept analysis. In *Proceedings of the 1st International Conference on Computational Logic*, London, UK, (2000) 1033-1047.
11. Gammoudi M. M., and Nafkha I.: A formal method for inheritance graph hierarchy construction. *Information Sciences* 140(3-4), (2002) 295-317.
12. Ganter B. and Wille R. : *Formal Concept Analysis - Mathematical Foundations*. Springer, 1999.
13. Godin R. and Mili. H. : Building and Maintaining Analysis Level Class Hierarchies Using Galois Lattices. In *Proceedings of the 8th Annual Conference on Object Oriented Programming Systems Languages and Applications*, Washington, D.C., USA, (1993) 394-410.
14. Godin R., Missaoui R., and April A. : Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38: (1993) 747-767.

15. Godin R. , Saunders E. , and Jecsei J. : Lattice model of browsable data spaces. *Journal of Information Sciences*, 40: (1986) 89-116.
16. Jaoua A., Bsaies Kh., and Consmtini W. : May reasoning be reduced to an Information Retrieval problem. *Relational Methods in Computer Science*, Quebec, Canada, (1999).
17. Jaoua A., Al-Rashdi A., AL-Muraikhi H., Al-Subaiey M., Al-Ghanim N., and Al-Misaifri S.: Conceptual Data Reduction, Application for Reasoning and Learning. The 4<sup>th</sup> Workshop on Information and Computer Science, KFUPM, Dhahran, Saudi Arabia, (2002).
18. Nafkha I, Elloumi S. and Jaoua A. : Conceptual Cooperative Information Retrieval System. In *International Arab Conference on Information Technology*, Doha December 16-19, Qatar, (2002) 534-539.
19. Nafkha I, Elloumi S. and Jaoua A. : Conceptual Information Retrieval System based on cooperative conceptual data reduction. 1<sup>st</sup> International Conference on Information & Communication Technologies : from Theory to Applications, Syria, (2004).
20. Salton G. : *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley, 1989.