Computing Intensions of Digital Library Collections

Carlo Meghini[1] and Nicolas Spyratos[2]

[1] Consiglio Nazionale delle Ricerche, Istituto della Scienza e delle Tecnologie della Informazione, Pisa, Italy meghini@isti.cnr.it
[2] Université Paris-Sud, Laboratoire de Recherche en Informatique, Orsay Cedex, France spyratos@lri.fr

Abstract. We model a Digital Library as a formal context in which objects are documents and attributes are terms describing documents contents. A formal concept is very close to the notion of a collection: the concept extent is the extension of the collection; the concept intent consists of a set of terms, the collection intension. The collection intension can be viewed as a simple conjunctive query which evaluates precisely to the extension. However, for certain collections no concept may exist, in which case the concept that best approximates the extension must be used. In so doing, we may end up with a too imprecise concept, in case too many documents denoted by the intension are outside the extension. We then look for a more precise intension by exploring 3 different query languages: conjunctive queries with negation; disjunctions of negation-free conjunctive queries; and disjunctions of conjunctive queries with negation. We show that a precise description can always be found in one of these languages for any set of documents. However, when disjunction is introduced, uniqueness of the solution is lost. In order to deal with this problem, we define a preferential criterion on queries, based on the conciseness of their expression. We then show that minimal queries are hard to find in the last 2 of the three languages above.

1 Introduction

In a Digital Library (DL for short), collections [14, 16, 1] are sets of documents defined to facilitate the tasks of various DL actors, ranging from content providers for whom physical collections are provided, to users, for whom logical collections are provided. The latter kind of collections typically helps the user in carrying out information access. For discovery, the user requires a "place" where to accumulate the discovered documents, similar to the shopping cart of an e-commerce Web site. This concept is commonly known as *static* collection [20, 2]. Static collections are also useful in other tasks, such as cooperative work, where they play the role of a shared information space within a community. A classical example of static collection is the *book-mark* (or *favorites*) of a Web browser. Users may also associate a description of their "view" of the DL to a collection, and access the collection whenever they need to explore this view. This concept is commonly captured by so-called *dynamic* collections [4, 5, 3]. Dynamic collections are not the only way users have in order to know at once the changes in the DL that may be of interest to them. Publish/subscribe (pub-sub for short) mechanisms are another way of achieving the same goal, but with a different modality: while in dynamic collection users are *active,* in the sense that they act by accessing collections, in pub-sub users are *passive,* in the sense that the system intercepts changes in the DL which may be of interest for users, and notifies them. This distinction is also known as *pull* vs. *push* access mode.

We argue that the notions of static and dynamic collections are two sides of the same coin, and propose a general notion of collection, which generalizes both. According to this notion, collections have an extension and an intension, very much like classes in object models or predicates in predicate logics. We then solve a basic problem, arising upon collection creation: the determination of the intension of a collection based on a given extension.

The paper is organized as follows: Sections 2 to 5 introduce our model of a DL, illustrating the most relevant concepts. Section 6 states in precise terms the problem we address. Sections 7 to 10 present different solutions to the problem, by examining different description languages for expressing collection intensions.

2 Terms

The basic ingredient of descriptions are *terms.* A term denotes a set of documents. As such, it may be a keyword describing the content of documents (such as *nuclear waste disposal* or *database*), or their type (*image*); or may be thought of as an attribute value (for instance, *creator= "CM"*). For generality, we do not impose any syntax on terms and treat them just as symbols making up a finite, non-empty set T, which is a proper subset of a countable domain $\mathcal{T}$, $\mathsf{T} \subset \mathcal{T}$, always containing the special term *true*, standing for truth.

Terms are arranged in a taxonomy, that is a binary relation $\leq_{\mathsf{T}}$ on T, reflexive and transitive, having *true* as the greatest element, that is

$$\forall \mathsf{t} \in \mathsf{T},\ \mathsf{t} \leq \mathit{true} \text{ and } \mathit{true} \leq \mathsf{t} \text{ implies } \mathsf{t} = \mathit{true}.$$

Based on $\leq_{\mathsf{T}}$, we define $\equiv_{\mathsf{T}}$ as follows: for any two terms $\mathsf{t}_1, \mathsf{t}_2 \in \mathsf{T}$,

$$\mathsf{t}_1 \equiv_{\mathsf{T}} \mathsf{t}_2 \text{ if and only if } \mathsf{t}_1 \leq \mathsf{t}_2 \text{ and } \mathsf{t}_2 \leq \mathsf{t}_1.$$

It is easy to see that $\equiv_{\mathsf{T}}$ is an equivalence relation. Let T_e be the set of equivalence classes induced by $\equiv_{\mathsf{T}}$, *i.e.*

$$\mathsf{T}_e = \{\ [\mathsf{t}] \mid \mathsf{t} \in \mathsf{T}\}.$$

Clearly, $[\mathit{true}] = \{\mathit{true}\}$. Furthermore, let us extend $\leq_{\mathsf{T}}$ to T_e as follows:

$$[\mathsf{t}_1] \leq_{\mathsf{T}} [\mathsf{t}_2] \text{ iff } \mathsf{t}_1 \leq_{\mathsf{T}} \mathsf{t}_2.$$

$(\mathsf{T}_e, \leq_{\mathsf{T}})$ is now a partial order, in which equivalent terms have been collapsed into the same equivalence class, having as greatest element $[\mathit{true}]$. To simplify notation, we will consider these equivalence classes as terms, therefore using the symbol T in place of T_e, and understand $\leq_\mathsf{T}$ as a partial order.

For any two terms $\mathsf{t}_1, \mathsf{t}_2 \in \mathsf{T}$, if $\mathsf{t}_1 \leq \mathsf{t}_2$ we say that t_1 *is a specialization (or sub-term) of* t_2, or that t_2 *is a generalization (or super-term) of* t_1.

3 The description directory

Description are used to annotate the documents of a digital library, which for the present purposes we just represent as a finite, non-empty subset D. The relation between documents and terms is stored in the *description directory,* which is a relation r from documents to terms, $\mathsf{r} \subseteq \mathsf{D} \times \mathsf{T}$, such that $(\mathsf{d}, \mathsf{t}) \in \mathsf{r}$ means that d is described (or indexed) by term t. We impose on r two requirements:

- it must be total, that is $dom(\mathsf{r}) = \mathsf{D}$. This is not a serious limitation for the users, because if no term qualifies as a satisfactory descriptor of a document, the term *true* can, and indeed should, be used;
- a document cannot be indexed by $\leq$-related terms:

$$\forall \mathsf{d} \in \mathsf{D},\ \mathsf{t}_1, \mathsf{t}_2 \in \mathsf{r}(\mathsf{d}) \text{ implies } \mathsf{t}_1 \parallel \mathsf{t}_2. \tag{1}$$

  This second constraint requires to select independent terms when indexing a document, which we think is not a serious limitation. The constraint also interacts with the previous one by imposing that if *true* is used for describing a document, then no other term can be used to describe that document, which is consistent with the usage of *true* postulated above.

From r we define two functions which will turn out very useful in the sequel:

- the *index,* a function $index : \mathsf{D} \to \mathcal{P}(\mathsf{T})$, giving the terms which a document is indexed by: $\forall \mathsf{d} \in \mathsf{D},\ index(\mathsf{d}) = \{\mathsf{t} \in \mathsf{T} \mid (\mathsf{d}, \mathsf{t}) \in \mathsf{r}\}$.
- the *extension,* a function $termext : \mathsf{T} \to \mathcal{P}(\mathsf{D})$, giving the documents which a term describes: $\forall \mathsf{t} \in \mathsf{T},\ termext(\mathsf{t}) = \{\mathsf{d} \in \mathsf{D} \mid (\mathsf{d}, \mathsf{t}) \in \mathsf{r}\}$

Constraint (1) just says that $index(\mathsf{d})$ consists of incomparable terms, for all documents $\mathsf{d} \in \mathsf{D}$.

4 General descriptions

In general, a description is a propositional formula over the alphabet T, built out of the connectives $\neg$ (negation), $\wedge$ (conjunction) and $\vee$ (disjunction). We will denote the set of such formulas as $\mathcal{L}_T$, or simply $\mathcal{L}$ when there is no danger of ambiguity.

Descriptions denote sets of documents. This is captured by the function *ans*, named after the fact that a typical usage of descriptions is for querying a DL.

ans is inductively defined as follows, where $\mathsf{t}, \mathsf{t}' \in \mathsf{T}$ and $\mathsf{q}, \mathsf{q}_1, \mathsf{q}_2 \in \mathcal{L}$:

$$\begin{aligned} ans(\mathsf{t}) &= \bigcup\{termext(\mathsf{t}') \mid \mathsf{t}' \leq \mathsf{t}\} \\ ans(true) &= \mathsf{D} \\ ans(\neg\mathsf{q}) &= \mathsf{D} \setminus ans(\mathsf{q}) \\ ans(\mathsf{q}_1 \wedge \mathsf{q}_2) &= ans(\mathsf{q}_1) \cap ans(\mathsf{q}_2) \\ ans(\mathsf{q}_1 \vee \mathsf{q}_2) &= ans(\mathsf{q}_1) \cup ans(\mathsf{q}_2) \end{aligned}$$

In the course of our study, we will need to consider several sub-languages of $\mathcal{L}$, corresponding to different types of descriptions. The simplest descriptions are conjunctions of incomparable terms. We will call these descriptions *simple queries,* and denote their set as $\mathcal{L}_S$. In fact, document descriptions can be regarded as simple queries given by the conjunction of the terms which describe the document. That is, assuming that the description of a document d is given by $index(\mathsf{d}) = \{\mathsf{t}_1, \mathsf{t}_2, \ldots, \mathsf{t}_n\}$ and recalling that (1) sanctions the incomparability of the terms $\mathsf{t}_1, \mathsf{t}_2, \ldots, \mathsf{t}_n$, we may, and in fact will assume that:

$$index(\mathsf{d}) = (\mathsf{t}_1 \wedge \mathsf{t}_2 \wedge \ldots \wedge \mathsf{t}_n) \in \mathcal{L}_S$$

Other important classes of descriptions will be introduced in due course.

5 Collections

A collection is a set of documents that make up a significant whole from an application point of view. We model collections as objects belonging to a finite, non-empty set C. The membership of documents into collections is stored in the *classification directory,* which is a relation e from documents to collections, $\mathsf{e} \subseteq \mathsf{D} \times \mathsf{C}$, such that $(\mathsf{d}, \mathsf{c}) \in \mathsf{e}$ means that d is a member of (or belongs to) collection c. In a DL it is usually required that every document belongs to at least one collection: $dom(\mathsf{e}) = \mathsf{D}$.

In order to best serve its purposes, a collection must have both an *extension* and an *intension,* very much like predicates in predicate logics. The extension of a collection is the set of objects that are members of the collection at a given point in time. It can then be defined as the total function $collext : \mathsf{C} \rightarrow \mathcal{P}(\mathsf{D})$ given by: $\forall \mathsf{c} \in \mathsf{C},\ collext(\mathsf{c}) = \{\mathsf{d} \in \mathsf{D} \mid (\mathsf{d}, \mathsf{c}) \in \mathsf{e}\}$. The intension of a collection is a description of the meaning of the collection, that is the peculiar property that the members of the collection collectively possess and that distinguishes the collection from other collections. This should not be confused with the so-called collection metadata (such as the owner or the creation date of the collection), which represent properties of collections required for administration purposes. Formally, the intension of a collection is a description, and is associated to its collection by the total function $collint : \mathsf{C} \rightarrow \mathcal{L}$. The question arises how these two notions should be related. An obvious requirement is that the set of documents belonging to the collection must *agree* with the collection intension. This can be expressed by requiring that the collection intension, when used as a query, should retrieve *at least* the documents in the collection extension. Formally:

$$\forall \mathsf{c} \in \mathsf{C},\ \ collext(\mathsf{c}) \subseteq ans(collint(\mathsf{c})). \tag{2}$$

As a consequence of this last requirement, we obtain two very useful properties of collections, namely: for any given query $\mathsf{q} \in \mathcal{L}$ and collection $\mathsf{c} \in \mathsf{C}$: (1) if $collint(\mathsf{c}) \wedge \mathsf{q}$ is unsatisfiable, then no document in (the extension of) c satisfies the query, that is: $ans(\mathsf{q}) \cap collext(\mathsf{c}) = \emptyset$. (2) if $collint(\mathsf{c})$ subsumes q, then all documents in (the extension of) c satisfy the query, that is: $collext(\mathsf{c}) \subseteq ans(\mathsf{q})$.

For a given collection $\mathsf{c} \in \mathsf{C}$, we define the *precision* of the collection intension, $prec(\mathsf{c})$, the set of documents denoted by $collint(\mathsf{c})$ which are not members of the collection:

$$prec(\mathsf{c}) = ans(collint(\mathsf{c})) \setminus collext(\mathsf{c})$$

If $prec(\mathsf{c}) = \emptyset$ we say that the collection is *precise,* and *imprecise* otherwise. Clearly, a collection is precise if and only if $collext(\mathsf{c}) = ans(collint(\mathsf{c}))$. More generally, we say that a description α is precise with respect to a set of documents X just in case $X = ans(\alpha)$.

6 The problem

The problem we want to address in this study is the following: given a DL and a subset X of the documents in it, to find a description $\alpha \in \mathcal{L}$ such that $X \subseteq ans(\alpha)$. This problem typically arises when a user has a set of documents and wants to create a collection having those documents as extension. The documents in question may have been gathered by the user through one or more discoveries, or may have been brought to the user attention by an expert, or may have been notified to him by the system as the result of the user registration to a publish-subscribe mechanism. These are just a few scenarios, in all of which the user likes the documents he has and wants to persist their set in the DL by creating a collection which holds them. To this end, an intension must be defined which satisfies the constraint (2), whence the problem.

Let us define as *conjunctive queries* the descriptions of the form:

$$\bigwedge_{1 \leq j \leq n} l_j \quad (n \geq 1)$$

where each l_j is a *literal,* that is is either a term $t \in \mathsf{T}$, in which case it is called a *positive* literal, or its negation $\neg t$ (negative literal), such that:

- a term and its negation do not occur: for no different indexes $i, j \in [1, n]$, $l_i = \mathsf{t}$ and $l_j = \neg\mathsf{t}$, for some $\mathsf{t} \in \mathsf{T}$.
- literals are pairwise incomparable: two literals are incomparable if they are either both positive or both negative, and the terms occurring in them are incomparable. Let $\mathcal{L}_C$ be the set of conjunctive queries.

A typical conjunctive query is the *description* of a document $\mathsf{d} \in \mathsf{D}$, $\delta(\mathsf{d})$, given by the conjunction of the terms describing the document with the negation of the terms *not* describing the document:

$$\delta(\mathsf{d}) = \bigwedge\{\mathsf{t} \mid \mathsf{t} \in index(\mathsf{d})\} \wedge \bigwedge\{\neg\mathsf{t}' \mid \mathsf{t}' \not\in index(\mathsf{d})\}$$

It is easy to see that $\delta(\mathsf{d})$ is more specific (*i.e.*, it is subsumed by) the index of d, $index(\mathsf{d})$; moreover, $\{\mathsf{d}\} \subseteq ans(\delta(\mathsf{d}))$ and a document $\mathsf{d}' \in ans(\delta(\mathsf{d}))$ just in case d' has exactly the same index as d, that is $index(\mathsf{d}) = index(\mathsf{d}')$. We assume this is not the case, *i.e.* all document indexes are different. This is not a serious limitation, since documents with the same index can be treated as a class, of which only one representative is considered.

Now *DNFS queries* are descriptions of the form:

$$\bigvee_{1 \leq i \leq m} D_i \quad (m \geq 1)$$

where each D_i is called a *disjunct* and is a conjunctive query. DNFS queries make up the language $\mathcal{L}_D$.

Evidently, any set of documents X has a trivial description in $\mathcal{L}_D$, given by:

$$\bigvee\{\delta(\mathsf{d}) \mid \mathsf{d} \in X\}$$

which is as precise as a description of X can be in the DL. However, this description is not very interesting: apart from being as large as X itself, it just replicates the index of every document in X, offering no additional information. A more satisfactory formulation of our problem is therefore: given a set of documents X, can we find a description of X which is *better* than the trivial one?

7 An easy solution: formal concepts

A simple query would certainly be a better description for X than the trivial one. Simple queries have a minimal logical structure (no negation, no disjunction) and therefore convey their meaning in a simple and intuitive way. So we reduce our problem to the following ones:

1. does X have a description in $\mathcal{L}_S$?
2. how precise can it be?

An answer to both questions comes from Formal Concept Analysis (FCA) [11, 10, 12]. The *formal context* of a DL is the triple $\mathcal{K} = (\mathsf{D}, \mathsf{T}, \mathsf{x})$, where:

$$(\mathsf{d}, \mathsf{t}) \in \mathsf{x} \text{ iff } \exists \mathsf{t}' \leq \mathsf{t} : \ (\mathsf{d}, \mathsf{t}') \in \mathsf{r}$$

The relation x, called the *incidence* of the context, extends r by taking into account the term taxonomy according to its intuitive meaning: it assigns a term t to a document d just in case d is described by a term t' that is more specific

	A	B	C	D	E	F	$true$
1	×		×	×	×		×
2		×	×				×
3	×		×	×		×	×
4		×	×	×	×	×	×
5	×	×			×	×	×

Fig. 1. A Formal Context

than t. Since t is more specific than itself (*i.e.*, $\leq$ is reflexive) we have that $\mathsf{r} \subseteq \mathsf{x}$. In particular, $\mathsf{r} = \mathsf{x}$ if no term is a sub-term of another term.

As an example, let us consider the DL whose formal context is shown in Figure 1 left in tabular form. In this DL, term D is a sub-term of C and in fact any document described by D is also described by C, and all documents are described by $true$.

A *formal concept* in $\mathcal{K}$ is a pair (D, T), where: (1) D, the *extent* of the concept, is a set of documents: $D \subseteq \mathsf{D}$; (2) T, the *intent* of the concept, is a set of terms: $T \subseteq \mathsf{T}$; and (3) T are the terms describing all documents in D and, vice-versa, D are all the documents described by the terms in T. Formally, (D, T) is a concept if and only if $D = \psi(T)$ and $T = \varphi(D)$, where

$$\begin{aligned}
\psi(T) &= \bigcap\{\varepsilon(\mathsf{t}) \mid \mathsf{t} \in T\} \text{ for all } T \subseteq \mathsf{T} \\
\varphi(D) &= \bigcap\{\iota(\mathsf{d}) \mid \mathsf{d} \in D\} \text{ for all } D \subseteq \mathsf{D} \\
\varepsilon(\mathsf{t}) &= \{\mathsf{d} \in \mathsf{D} \mid (\mathsf{d}, \mathsf{t}) \in \mathsf{x}\} \text{ for all } \mathsf{t} \in \mathsf{T} \\
\iota(\mathsf{d}) &= \{\mathsf{t} \in \mathsf{T} \mid (\mathsf{d}, \mathsf{t}) \in \mathsf{x}\} \text{ for all } \mathsf{d} \in \mathsf{D}
\end{aligned}$$

In the formal context shown in Figure 1, $(\{1, 3, 4\}, \{C, D, true\})$ is a concept, while $(\{1, 3\}, \{A, D\})$ is not.

Lemma 1. *For all sets of terms $Y \subseteq \mathsf{T}$, $\psi(Y) = ans(\bigwedge Y)$.*

Since in a concept (D, T), we have that $D = \psi(T)$, the previous Lemma tells us that $D = ans(\bigwedge T)$, that is the extent of a concept is the answer to the intent of the concept, seen as a conjunction of terms.

7.1 Solving the problem for $\mathcal{L}_S$

It should be evident that a formal concept strongly is a precise collection: the concept extent mirrors the extension of the collection; the concept intent consists of a set of terms, which can be viewed as a simple query which evaluates precisely to the extent. However, for our purposes concept intents tend to be redundant. For instance, given the concept $(\{4, 5\}, \{B, E, F, true\})$, there are simpler queries than $(B \wedge E \wedge F \wedge true)$ which return $\{4, 5\}$, for instance $(B \wedge E)$, $(B \wedge F)$ and $(E \wedge F)$. Part of the problem is that $true$ is the most general term, thus decidedly useless in queries other than $true$ itself. However the problem is more general since none of B, E and F is $\leq$-comparable with the others, yet one of the 3 is clearly redundant.

A term $t \in T$ is *redundant* in a set of terms $T \subseteq \mathsf{T}$, iff for all documents outside $ans(T)$, $d \in \mathsf{D} \setminus ans(T)$, whenever t does not describe d, $(d,t) \notin \mathsf{x}$, then there exists another term t' in T, such that t' does not describe d, $(d,t') \notin \mathsf{x}$. Now it is very simple to check that t is redundant in T iff $ans(T) = ans(T \setminus \{t\})$. That is, a term is redundant in a set if it can be removed without altering the denotation of that set. Given a set of terms T, a simplification function σ is any function that iterates through the elements of T removing the ones that are found redundant. Notice that if $t \leq t'$ then t' is redundant whenever it co-occurs with t, so by eliminating redundant terms we implicitly eliminate comparable terms. The order in which non-comparable, redundant terms are considered is very important to determine the result. Indeed, the one of the terms B, E, F which is considered first is always redundant, while the remaining 2 are not. So, $\sigma(\{B, E, F, true\})$ may be anyone of $\{E, F\}$, $\{B, F\}$, or $\{B, E\}$. This is not relevant for our study, so we will leave it unspecified.

Proposition 1. *A set of documents $X \subseteq \mathsf{D}$ has a unique precise description in $\mathcal{L}_S$ if and only if X is the extent of a concept in $\mathcal{K}$.*
Proof: ($\leftarrow$) Suppose (X, Y) is a concept in $\mathcal{K}$. By definition, $X = \psi(Y)$ and by the previous Lemma $X = ans(\bigwedge Y)$. Now $(\bigwedge Y)$ may not be in $\mathcal{L}_S$ since some terms in Y may not be incomparable. Now $\sigma(Y)$ consists of incomparable terms and $X = ans(\bigwedge \sigma(Y))$ therefore $(\bigwedge \sigma(Y))$ is an $\mathcal{L}_S$ precise description for X. Since no two concepts can have the same extent, $(\bigwedge \sigma(Y))$ is also unique.
($\rightarrow$) We must prove that (X, Y) is a concept in $\mathcal{K}$ for some set of terms $Y \subseteq \mathsf{T}$. Since X has a precise description in $\mathcal{L}_S$, there exists a set of incomparable terms $T \subseteq \mathsf{T}$, such that $X = ans(\bigwedge T)$. Let $Y = \bigcap\{\iota(\mathsf{d}) \mid \mathsf{d} \in X\}$. By construction, $Y = \varphi(X)$ therefore it remains to be proved that $X = \psi(Y)$. By the previous Lemma, this is the same as proving that $X = ans(Y)$. We do this in 2 steps. (1) $ans(Y) \subseteq X$. By construction, $T \subseteq Y$, hence $\psi(Y) \subseteq \psi(T)$. By applying twice the Lemma, we have $\psi(Y) = ans(Y)$ and $\psi(T) = ans(T) = X$, and therefore we have $ans(Y) \subseteq X$. (2) $X \subseteq ans(Y)$. Now, by construction, for all $x \in X$ and $y \in Y$, $(x, y) \in \mathsf{x}$, hence $x \in \varepsilon(y)$ hence $x \in \bigcap\{\varepsilon(y) \mid y \in Y\}$ hence $x \in \psi(Y) = ans(Y)$. Then $X \subseteq ans(Y)$. □

Now, coupled with the well-known result of FCA that, for all set of documents $X \subseteq \mathsf{D}$, $(\psi(\varphi(X)), \varphi(X))$ is the concept with the smallest extent containing X, this Proposition allows us to answer the questions posed at the beginning of this Section, as follows: all sets X of documents have a description in $\mathcal{L}_S$, which we call the *simple description* of X and denote as $\delta_S(X)$, given by:

$$\delta_S(X) = \sigma(\varphi(X)).$$

The precision of $\delta_S(X)$ is given by:

$$\psi(\varphi(X)) \setminus X$$

The most precise $\mathcal{L}_S$ description for $\{1, 2\}$ is therefore $\sigma(\{C, true\}) = \{C\}$, whose precision is $\{3, 4\}$.

	A	B	C	D	E	F	$true$	$\neg A$	$\neg B$	$\neg C$	$\neg D$	$\neg E$	$\neg F$	$false$
1	x		x	x	x		x		x				x	
2		x	x				x	x			x	x	x	
3	x		x	x		x	x		x			x		
4		x	x	x	x	x	x	x						
5	x	x			x	x	x			x	x			

Fig. 2. The augmentation of a formal context

In some cases, the precision may be too large a set for the user, who might therefore be looking for a more precise description. To this end, one of two routes may be followed: the extension relaxation route, in which the user gives up some of the documents in X, or the intension relaxation route, in which the user accepts a more complex description than a simple query. We have investigated the former route in [17], so we now concentrate on the latter.

The description language can be made more expressive than $\mathcal{L}_S$ in two different ways: by adding negation of single terms, in which case we end in $\mathcal{L}_C$, or by adding disjunction, in which case we end into a subset of $\mathcal{L}_D$, consisting of disjunctions of simple queries. We will consider each of these two languages in the sequel.

8 Conjunctive queries

FCA can be very useful also if we admit negation in descriptions. In order to see how, we extend the notion of context to include negated terms. These have been already informally introduced in Section 6. We now give them a more precise mathematical status.

Let $\neg$ be a bijection from T to $\mathsf{T}_\neg$, a subset of $\mathcal{T}$ disjoint from T. For simplicity, we will write $\neg\mathsf{t}$ in place of $\neg(\mathsf{t})$ to indicate the negation of any attribute $\mathsf{t} \in \mathsf{T}$. For clarity, we will denote as $false$ the term $\neg true$. If $T \subseteq \mathsf{T}$ is a set of terms, $\neg(T)$ is the set of the negation of each term in T, *i.e.* $\neg(T) = \{\neg\mathsf{t} \mid \mathsf{t} \in T\}$.

The *augmented* formal context of a DL is the triple $\mathcal{K}_\neg = (\mathsf{D}, \mathsf{T} \cup \mathsf{T}_\neg, \mathsf{x}_\neg)$, where:

$$\mathsf{x}_\neg = \mathsf{x} \cup \{(\mathsf{d}, \neg\mathsf{t}) \mid (\mathsf{d}, \mathsf{t}) \notin \mathsf{x}\}.$$

In practice, the augmentation of a formal context introduces negated terms, whose extensions are the complement of the extensions of the corresponding non-negated terms. We will use $\neg$ as a subscript to indicate that we refer to the augmented context, *e.g.*$\varphi_\neg$ is the correspondent of φ in the augmented context.

The augmentation of the formal context shown in Figure 1 is given in Figure 2. It can be easily seen that augmentation induces a total, one-to-one homomorphism from the concepts of a context to those of the augmentation. In general this is not an isomorphism, as the augmentation may have more concepts. In addition, concept intents may be larger in the augmented context, as they may include negated terms. So in moving from a context to its augmentation we are able to describe more sets of documents. Now, by equating intents of augmented concepts with conjunctive queries, we can state the following Proposition.

Proposition 2. *A set of documents $X \subseteq \mathsf{D}$ has a precise description in $\mathcal{L}_C$ if and only if X is the extent of a concept in $\mathcal{K}_\neg$.*

The proof of this proposition is identical to that of Proposition 1. The most precise $\mathcal{L}_C$ description for a given set of documents X, $\delta_C(X)$ is therefore

$$\delta_C(X) = \sigma(\varphi_\neg(X))$$

and its precision is given by:

$$\psi_\neg(\varphi_\neg(X)) \setminus X. \tag{3}$$

Let us find the most precise $\mathcal{L}_C$ description for the set $\{1,2\}$ in our running example. We recall that the most precise $\mathcal{L}_S$ description for $\{1,2\}$ is $\{C\}$, whose precision is $\{3,4\}$. Now it turns out that this set has a precise $\mathcal{L}_C$ description, since $\psi_\neg(\varphi_\neg(\{1,2\})) = \psi_\neg(\{C, true, \neg F\}) = \{1,2\}$. The sought description is given by $\sigma(\varphi_\neg(\{1,2\})) = \sigma(\{C, true, \neg F\}) = \{\neg F\}$.

We conclude by observing that the set (3) can be computed without computing the augmented context, of course. In fact, it can be verified that, for all sets of terms T and sets of documents D :

$$\psi(T) = \{d \in \mathsf{D} \mid (d,t) \in \mathsf{x} \text{ for all } t \in T \text{ and } (d,t) \notin \mathsf{x} \text{ for all } \neg t \in T\}$$
$$\varphi(D) = \{t \in \mathsf{T} \mid (d,t) \in \mathsf{x} \text{ for all } d \in D\} \cup \{\neg t \mid (d,t) \notin \mathsf{x} \text{ for all } d \in D\}$$

9 Introducing disjunction

Let $\mathcal{L}_U$ be the sub-language of $\mathcal{L}$ consisting of disjunctions of simple queries, which we call *disjunctive queries* for brevity.

Disjunctive queries can describe many more sets of documents, since disjunction allows to "accumulate" simple queries at will. So, the first question that naturally arises is whether all sets of documents have a precise description in $\mathcal{L}_U$. The answer, perhaps surprisingly, is "no," as the following Proposition shows. Let C^e and C^i denote the extent and the intent of concept C, respectively.

Proposition 3. *A set of documents $X \subseteq \mathsf{D}$ has a precise description in $\mathcal{L}_U$ if and only if $\gamma(\mathsf{d})^e \subseteq X$ for all $\mathsf{d} \in X$.*
Proof: ($\rightarrow$) Let β be the query

$$\beta = \bigvee\{\bigwedge \sigma(\gamma(\mathsf{d})^i) \mid \mathsf{d} \in X\}$$

By definition of ans, we have that:

$$ans(\beta) = \bigcup\{ans(\bigwedge \sigma(\gamma(\mathsf{d})^i)) \mid \mathsf{d} \in X\}.$$

By definition of σ, $ans(\bigwedge Y) = ans(\bigwedge \sigma(Y))$, for all sets of documents Y, therefore:

$$ans(\beta) = \bigcup\{ans(\bigwedge \gamma(\mathsf{d})^i) \mid \mathsf{d} \in X\}.$$

From Lemma 1 we have that $\gamma(\mathsf{d})^e = ans(\bigwedge \gamma(\mathsf{d})^i)$, therefore

$$ans(\beta) = \bigcup\{\gamma(\mathsf{d})^e \mid \mathsf{d} \in X\}$$

By the hypothesis it follows that $ans(\beta) \subseteq X$. By construction, $\mathsf{d} \in \gamma(\mathsf{d})^e$, hence $X \subseteq ans(\beta)$. Therefore $ans(\beta) = X$, and β is a precise description for X.
($\leftarrow$) We prove that if for some document $\mathsf{d} \in \mathsf{D}$ $\gamma(\mathsf{d})^e \not\subseteq X$, then X has no precise $\mathcal{L}_U$ description. Let $\mathsf{d}' \notin X$ and $\mathsf{d}' \in \gamma(\mathsf{d})^e$. Then, $\mathsf{d}' \in C^e$ for each super-concept of $\gamma(\mathsf{d})$. But there is no concept extent containing d other than those of the super-concepts of $\gamma(\mathsf{d})$. It follows that any description containing d also contain d', thus X has no precise $\mathcal{L}_U$ description. □

In order to exemplify this last proposition, let us consider again the formal context shown in Figure 1. In this context, $\gamma(2)^e = \{2, 4\}$. This is a consequence of the fact that $\varepsilon(2) \subseteq \varepsilon(4)$ and implies that all concepts having 2 in their extents ($i.e.\gamma(2)$ and its super-concepts) also have 4 in their extents, therefore any set of documents containing 2 but not 4 does not have a precise $\mathcal{L}_U$ description. However, the power of disjunction is not to be underestimated, because while $\mathcal{L}_S$ and $\mathcal{L}_C$ precise descriptions are unique, a set of documents X may have more than one precise $\mathcal{L}_U$ description. This is due to the fact that X may be covered by concept extents in more than one way. Let us consider for instance the set $\{2, 3, 4, 5\}$ in our running example. This set has a precise $\mathcal{L}_U$ description, since it satisfies the condition established by the last Proposition, namely $\gamma(2)^e \subseteq \{2, 3, 4, 5\}$ and the same holds for $\gamma(3)^e$, $\gamma(4)^e$ and $\gamma(5)^e$. According to the proof of the last Proposition,

$$\beta = (B \wedge C) \vee (A \wedge D \wedge F) \vee (D \wedge E \wedge F) \vee (A \wedge E \wedge F)$$

is a precise description of $\{2, 3, 4, 5\}$. However, since $\mu(B) = (\{2, 4, 5\}, \{B\})$ and $\gamma(3) = (\{3\}, \{A, C, D, F\})$, also $B \vee (A \wedge D \wedge F)$ is a precise description of $\{2, 3, 4, 5\}$. This latter description is intuitively preferable over the former, since it denotes the same set but it is much shorter. Indeed, every disjunct of the latter description is a subset of a disjunct of the former description; this means that the former description may have more as well as larger disjuncts (set-theoretically speaking), however both of these can be pruned to obtain an equivalent but shorter description.

In order to capture formally this preference criterion, we define a relation between disjunctive queries. To this end and for the sake of simplicity, we will regard simple queries as sets of terms. Given two disjunctive queries $\alpha = D_1 \vee \ldots \vee D_m$ and $\beta = E_1 \vee \ldots \vee E_n$, α *is preferred over* β, $\alpha \sqsubseteq \beta$, if and only if $ans(\alpha) = ans(\beta)$ and for every disjunct D_i in α there exists a disjunct E_j in β such that $D_i \subseteq E_j$. $\sqsubseteq$ is reflexive and transitive, thus $(\mathcal{L}_U, \sqsubseteq)$ is a pre-order. A description is said to be *minimal* if it is a minimal element of $(\mathcal{L}_U, \sqsubseteq)$, that is no description is preferred over it. We then set out to find minimal descriptions. FCA proves very helpful to this end. In order to show how, we must first introduce the notions of candidate concept and minimum set cover.

– Given a set of documents $X \subseteq \mathsf{D}$, a *candidate concept* for X is a concept C such that $C^e \subset X$ and no super-concept D of C exists such that $D^e \subset X$.
– Given a collection $\mathcal{C}$ of subsets of a finite set S, a *set cover* for S is a subset $\mathcal{C}' \subseteq \mathcal{C}$ such that every element in S belongs to at least one member of $\mathcal{C}'$. A set cover is *minimum* if no set cover exists with a smaller cardinality.

As it can be proved: For all sets of documents $X \subseteq \mathsf{D}$,

1. if $X = \psi(\varphi(X))$, then $\sigma(\varphi(X))$ is the only minimal $\mathcal{L}_U$ description of X;
2. if $X \subset \psi(\varphi(X))$, then a $\mathcal{L}_U$ description $D_1 \vee \ldots \vee D_n$ of X is a precise minimal $\mathcal{L}_U$ description for X iff, for all $1 \leq j \leq n$, $D_j = \sigma(C_j^i)$ where C_j is a candidate concept and $C_1^e, \ldots, C_n^e$ is a minimum set cover for X amongst the extents of all candidate concepts for X.

From a computational point of view, the above characterization of minimal precise descriptions does not look particularly good, since these are equated to minimum set covers, whose computation is strongly suspected to be intractable [13]. The question arises whether there exists an equivalent characterization that is more amenable to computation. Unfortunately, the answer is negative. The next Proposition shows that MINIMUM SET COVER can be reduced to the computation of a minimal description, thus giving a lower bound for the latter problem.

Proposition 4. *Computing a minimal $\mathcal{L}_U$ description is NP-hard.*
Proof: We reduce MINIMUM SET COVER to our problem. Given an instance of MINIMUM SET COVER, that is a collection $\mathcal{C}$ of subsets of a set S, we define the formal context (D, T, i) as follows:

– *$D = S \cup \{o\}$ where o is any object not in S;*
– *T has one term t_i for each element C_i of $\mathcal{C}$, plus an extra term u which is any object not in S.*
– *i is defined as follows:*
  - *for all $s \in S$, if $s \in C_i$ then $(s, t_i) \in i$;*
  - *$(o, u) \in i$;*
  - *nothing else is in i.*

It can be proved that each minimum set cover corresponds to a precise, minimal $\mathcal{L}_U$ description for S and vice-versa. □

Candidate concepts play a key role in computing minimal, precise $\mathcal{L}_U$ descriptions, since each of such descriptions is obtained by combining the extents of those concepts so as to form a minimum set cover for X. An efficient way to compute candidate concepts is therefore fundamental. Iterating through all concept extents and retaining the maximal subsets of X is certainly a way of doing it, but not necessarily an efficient one, since a context may have an exponential number of concepts (in the size of the context). Fortunately, there is a more efficient method. It can be easily checked that, for all sets of documents X, the extents of the candidate concepts for X are given by:

$$\max_{t \in \mathsf{T}} \{Y = (\varepsilon(t) \cap X) \mid Y = \psi(\varphi(Y))\} \tag{4}$$

```
procedure $c3$ ($X$ : set of document id)
begin
$\mathcal{A}_X \leftarrow \emptyset$
for each term $t$ in $\mathsf{T}$ do
    begin
    $Y \leftarrow \varepsilon(t) \cap X$
    if $\nexists\, Z \in \mathcal{A}_X$ such that $Y \subseteq Z^e$ and $Y = \psi(\varphi(Y))$ then
        begin
        for each concept $V \in \mathcal{A}_X$ such that $V^e \subset Y$ do $\mathcal{A}_X \leftarrow \mathcal{A}_X \setminus V$
        $\mathcal{A}_X \leftarrow \mathcal{A}_X \cup (Y, \varphi(Y))$
        end
    end
return $\mathcal{A}_X$
end
```

Fig. 3. The $c3$ procedure

where maximality is with respect to set-containment. Clearly, every member of this set is the extent of a concept, a subset of X, and a maximal one. Notice that if $X = \psi(\varphi(X))$, X is the only member of this set.

It follows that the set of candidate concepts of X, $\mathcal{A}_X$, can be computed efficiently by iterating through the terms, as the procedure $c3$ (Figure 3) does. For each term, $c3$ computes in Y the overlapping between the extension of the term and X. If there already is in $\mathcal{A}_X$ a concept with an equal or larger extent that Y, then Y needs no longer to be considered because, even though it turns out to be a concept extent, it will not be maximal. Otherwise, if Y is the extent of a concept, that is $Y = \psi(\varphi(Y))$, then it may be the extent of a candidate concept, so it is added to $\mathcal{A}_X$ after removing from it the concepts with a smaller extent. Thus, when all terms have been examined, $\mathcal{A}_X$ contains the concepts whose extents are all the members of the set (4).

Let us consider again the set $\{2, 3, 4, 5\}$ for which we wish to find a minimal, precise $\mathcal{L}_U$ description in our running example. By running $c3$ on the context, we have the results shown in Table 1. For each term, the Table shows the overlap of the term extension with X, if this is a concept extent, the intent is shown next, and in the last column whether or not the concept is candidate. There turns out to be only 2 candidate concepts, so there is only one minimum set cover for X that can be constructed with the extents of these 2 concepts, therefore the only minimal, precise $\mathcal{L}_U$ for X is:

$$(\bigwedge \sigma(\{B, true\})) \vee (\bigwedge \sigma(\{F, true\})) = B \vee F$$

In this example, the minimum set cover problem has no impact, due to the toy size of the example. In real cases, however, candidate concepts can be as many as the terms, and an approximation technique may have to be used in order to avoid long computations. In alternative, an incomplete method may be chosen, returning a non-minimal description.

t	$\varepsilon(t) \cap X$	*intent*	*candidate*
A	$\{3,5\}$	$\{A, F, true\}$	no
B	$\{2,4,5\}$	$\{B, true\}$	yes
C	$\{2,3,4\}$	no	
D	$\{3,4\}$	$\{C, D, F, true\}$	no
E	$\{4,5\}$	$\{E, F, true\}$	no
F	$\{3,4,5\}$	$\{F, true\}$	yes

Table 1. Run of $c3$ with $X = \{2,3,4,5\}$

9.1 Imprecise $\mathcal{L}_U$ descriptions

An imprecise $\mathcal{L}_U$ description might be desirable in case a precise one does not exist or is not satisfactory, for instance because too long. Here the problem is: to find the minimal description amongst the descriptions having minimal imprecision. This problem has a unique solution which we have already seen: $\sigma(\varphi(X))$. This is due to the fact that $(\psi(\varphi(X)), \varphi(X))$ is the smallest concept whose extent includes X. Thus, $(\psi(\varphi(X))$ is the only concept extent with minimal imprecision. In our example, if we do not like the description $(B \vee F)$, our best alternative in $\mathcal{L}_U$ is $\sigma(\varphi(X)) = true$.

10 DNFS descriptions

We conclude this study by considering DNFS descriptions, that is formulas in $\mathcal{L}_D$. As we have already observed in Section 6, a set of documents X has always a precise DNFS description, but from the results of the last Section, we know that there may be more such descriptions. However, since the definition of minimality devised for $\mathcal{L}_U$ descriptions carries over $\mathcal{L}_D$ descriptions, the same technique can be applied. In order to illustrate, let us consider the document set $\{1,2,3\}$. Table 2 shows the results of running $c3$ on this set, similarly to Table 1. The extents of the 3 candidate concepts identified by $c3$ allow us to construct two minimal, precise $\mathcal{L}_D$ descriptions for the given set of documents, namely:

$$(\bigwedge \sigma(\{A, \neg B, C, D, true\})) \vee (\bigwedge(\sigma(\{C, \neg E, true\})) = \neg B \vee \neg E$$

$$(\bigwedge \sigma(\{A, \neg B, C, D, true\})) \vee (\bigwedge(\sigma(\{C, \neg F, true\})) = \neg B \vee \neg F$$

11 Related work

The use of FCA in information system is not new (for a survey, see *e.g.* [19]). The structuring of information that FCA supports has inspired work on browsing [15, 6], clustering [7], and ranking [9, 18]. A basic drawback of these approaches is that they require the computation of the whole concept lattice, whose size may be exponential in that of the context, as it is well-known. An integrated approach to browsing and querying that uses only part of the lattice, and thus can be

t	$\varepsilon(t) \cap X$	*intent*	*candidate*
A	$\{1,3\}$	$\{A, \neg B, C, D, true\}$	yes
B	$\{2\}$	$\{\neg A, B, C, \neg D, \neg E, \neg F, true\}$	no
C	$\{1,2,3\}$	no	
D	$\{1,3\}$	already considered	no
E	$\{1\}$	non-maximal	no
F	$\{3\}$	non-maximal	no
$\neg A$	$\{2\}$	already considered	no
$\neg B$	$\{1,3\}$	already considered	no
$\neg C$	$\{\}$	non-maximal	no
$\neg D$	$\{2\}$	already considered	no
$\neg E$	$\{2,3\}$	$\{C, \neg E, true\}$	yes
$\neg F$	$\{1,2\}$	$\{C, \neg F, true\}$	yes

Table 2. Run of $c3$ on the augmented context with $X = \{1, 2, 3\}$

computed efficiently, is presented in [8], and extended to include user preferences in [17]. The usage of FCA for computing predicates describing sets of objects is novel, and complements the results of above mentioned approaches on the relationship between queries and concepts.

12 Conclusions

Thanks to the elementary notions of FCA, we have been able to solve a basic problem arising in DL collection management: the determination of a description for a given set of documents. We plan to expand the results obtained in this paper in 2 directions:

- by considering collection updates, in terms of insertion and removal of single documents from a collection extension; and
- by considering extensive usage of collection intensions for query processing, alluded to in Section 5. In fact, by introducing collection intensions we can reduce query processing in a DL to answering queries based on views, a problem that has been intensely studied in the database area in the last decade.

We also plan to set up experiments which would validate from a practical point of view the results obtained in this paper.

References

1. Donna Bergmark. Collection Synthesis. In *Proceeding of the second ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 253–262. ACM Press, 2002.
2. David C. Blair. The challenge of commercial document retrieval, Part II: a strategy for document searching based on identifiable document partitions. *Information Processing and Management*, 38:293–304, 2002.
3. Leonardo Candela. *Virtual Digital Libraries.* PhD thesis, Information Engineering Department, University of Pisa, 2006.

4. Leonardo Candela, Donatella Castelli, and Pasquale Pagano. A Service for Supporting Virtual Views of Large Heterogeneous Digital Libraries. In Traugott Koch and Ingeborg Sølvberg, editors, *7th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2003*, LNCS vol. 2769, pages 362–373, Trondheim, Norway, August 2003.
5. Leonardo Candela and Umberto Straccia. The Personalized, Collaborative Digital Library Environment CYCLADES and its Collections Management. In Jamie Callan, Fabio Crestani, and Mark Sanderson, editors, *Multimedia Distributed Information Retrieval*, LNCS vol. 2924, pages 156–172, 2004.
6. C. Carpineto and G. Romano. Information retrieval through hybrid navigation of lattice representations. *International Journal of Human-Computer Studies*, 45(5):553–578, 1996.
7. C. Carpineto and G. Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2):95–122, 1996.
8. C. Carpineto and G. Romano. Effective reformulation of boolean queries with concept lattices. In *Proceedings of International Conference on Flexible Query Answering Systems*, LNAI vol. 1495, pages 83–94, Roskilde, Denmark, May 1998.
9. C. Carpineto and G. Romano. Order-theoretical ranking. *Journal of American Society for Information Science*, 51(7):587–601, 2000.
10. B.A. Davey and H.A. Priestley. *Introduction to lattices and order*, chapter 3. Cambridge, second edition, 2002.
11. B. Ganter and R. Wille. Applied lattice theory: Formal concept analysis. `http://www.math.tu.dresden.de/∼ganter/psfiles/concept.ps`.
12. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer Verlag, 1st edition, 1999.
13. Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.
14. Gary Geisler, Sarah Giersch, David McArthur, and Marty McClelland. Creating Virtual Collections in Digital Libraries: Benefits and Implementation Issues. In *Proceedings of the second ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 210–218. ACM Press, 2002.
15. R. Godin, J. Gecsei, and C. Pichet. Design of a browsing interface for information retrieval. In *Proceedings of SIGIR89, the Twelfth Annual International ACM Conference on Research and Development in Information Retrieval*, pages 32–39, Cambridge, MA, 1989.
16. Carl Lagoze and David Fielding. Defining Collections in Distributed Digital Libraries. *D-Lib Magazine*, November 1998.
17. Carlo Meghini and Nicolas Spyratos. Preference-based query tuning through refinement/enlargement in a formal context. In J. Dix and S. Hegner, editors, *Proceedings of FoIKS 2006, the fourth Int. Symp. on Foundations of Information and Knowledge Systems*, LNCS vol. 3861, pages 278–293, Budapest, February 2006.
18. Uta Priss. Lattice-based information retrieval. *Knowledge Organization*, 27(3):132–142, 2000.
19. Uta Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40:521–543, 2006.
20. Ian H. Witten, David Bainbridge, and Stefan J. Boddie. Power to the People: End-user Building of Digital Library Collections. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries*, pages 94–103. ACM Press, 2001.