# A general framework for covering concepts using terminologies

B. Benatallah[1], M-S. Hacid[2], A. Léger[3], C. Rey[4], and F. Toumani[4]

[1] SCSE, University of New South Wales, Sydney, boualem@cse.unsw.edu.au
[2] LIRIS, Universit Lyon I, France, mshacid@bat710.univ-lyon1.fr
[3] France Telecom R&D, Rennes, France, alain.leger@francetelecom.com
[4] LIMOS, CNRS, Université Blaise Pascal, {rey,ftoumani}@isima.fr

**Abstract.** We describe a general framework for covering concepts using terminologies and briefly present the already investigated instances of this framework. Then, we formalize the best covering problem in the context of the $\mathcal{ALN}$ language, in which the difference operation is not semantically unique, and sketches the technique to solve the underlying computation problems.

## 1 Introduction

In [2] a general framework for *rewriting using terminologies* is defined as follows:

- given a terminology $\mathcal{T}$ expressed using a language $\mathcal{L}_t$,
- given a (query) concept description $Q$, expressed using a language $\mathcal{L}_s$ and that does not contain concept names defined in $\mathcal{T}$,
- given a binary relation $\rho : \mathcal{L}_s \times \mathcal{L}_t$, between $\mathcal{L}_s$ and $\mathcal{L}_t$ descriptions.

Can $Q$ be rewritten into a description $E$, expressed using a language $\mathcal{L}_d$ and built using (some) of the names defined in $\mathcal{T}$, such that $Q \rho E$ ? Additionally, some optimality criterion may be used in order to select the relevant rewritings.

Existing instances of this general framework can be distinguished with respect to the nature of the relation $\rho$, the optimality criterion as well as the languages $\mathcal{L}_t$, $\mathcal{L}_s$ and $\mathcal{L}_d$, respectively used to describe the terminology $\mathcal{T}$ the (query) concept $Q$ and the rewriting $E$. Examples of such instances are *(i)* the minimal rewriting problem [2], where $\rho$ is instantiated by equivalence modulo $\mathcal{T}$ while the size of the rewriting is used as the optimality criterion and *(ii)* rewriting queries using views [5], while $\rho$ is instantiated by subsumption and the optimality criterion is the inverse subsumption [2].

Subsumption relation plays a central role in the existing rewriting approaches. Indeed, all the mentioned instances of the general framework aim at reformulating a given query $Q$ into a description which is equivalent or subsumed by $Q$. The intuition here is that a given rewriting must capture all the *'information'* conveyed by a query $Q$. However, in many application contexts (e.g., see [3, 4]) it is not realistic to assume that such a rewriting always exists and it may be

interesting to look for a rewriting that *approximates* a given query. This observation motivates our work on a new instance of the general framework for rewriting, namely *covering concepts using terminologies* [3]. The salient feature of our approach is to use a measure of a *semantic distance* between concepts, instead of subsumption, to define rewritings, thereby enabling a more *flexible* rewriting process. More precisely, our aim is to reformulate a query $Q$ into a description that contain as much as possible of *common information* with $Q$. We call such a reformulation a *cover* of $Q$.

A key step toward handling such a problem is a precise definition of the measure used to compute the semantic proximity between concepts. We rely on a non standard operation in description logics, namely the (semantic) *difference or subtraction* operation, in order to define such a measure. The difference of two descriptions is defined in [6] as being a description containing all information which is a part of one argument but not a part of the other one. An interesting feature of the difference operation comes from its ability to produce a (set of) concept description(s) as output. In the sequel, we refer to descriptions obtained by the difference operation as *difference descriptions*. Note that, if the difference is not semantically unique, a difference description is in fact a set of descriptions. Difference descriptions characterize the notion of *'extra information'*, i.e., the information contained in one description and not contained in the other, thereby providing a means to measure a semantic distance between concepts.

With the difference operation at hand, and given an appropriate ordering $\prec_d$ on difference descriptions, we are then interested by the problem of rewriting a query $Q$ into a description $E$ such that the difference between $Q$ and $E$ is minimal w.r.t. $\prec_d$. In our previous work [3, 4], we investigated this problem in a restricted framework of languages in which the difference is semantically unique. In such a framework it turns out to be sufficient to specify $\prec_d$ as an ordering on the size of descriptions to capture the intuition behind the notion of best covers (i.e., covers whose descriptions contain as much as possible of *common information* with the original query). However, in the cases where the difference produces a set of descriptions, for example in the $\mathcal{ALN}$ language, a more subtle ordering is required. This paper extends our previous work in the following directions: *(i)* we define a general framework for covering concepts using terminologies and point out how the previous investigated instances can be formalized in this context, and *(ii)* we formalize the best covering problem in the context of the $\mathcal{ALN}$ language, in which the difference operation is not semantically unique, and sketches the technique to solve the underlying computation problems. Technical details regarding this new result are available in [1].

## 2 The general best covering problem

This section introduces some basic definitions to formally define the *general best covering problem*.

**Definition 1.** ***(Cover)*** *Let $\mathcal{L}$ be a DL in which the difference operation is computable, $\mathcal{T}$ (respectively, $Q$) be an $\mathcal{L}$-terminology (respectively, an $\mathcal{L}$-concept) and*

*let $E$ be a conjunction of some defined concepts from $\mathcal{T}$. $E$ is a* ***cover*** *of $Q$ using $\mathcal{T}$ iff: (i)$E$ is consistent with $Q$, i.e., $Q \sqcap E \not\equiv \bot$, and (ii) $E$ shares some information with $Q$, i.e., $Q \not\in_{\equiv} Q - lcs_{\mathcal{T}}(Q, E)$, where $\in_{\equiv}$ stands for set membership modulo equivalence.*

Hence, a cover of a concept $Q$ using $\mathcal{T}$ is defined as being a conjunction of defined concepts occurring in $\mathcal{T}$ which is consistent with $Q$ and that share some information with $Q$. We use the expression $rest_E(Q) = Q - lcs_{\mathcal{T}}(Q, E)$, called the *rest* of a cover, to denote the part of a query $Q$ that is not captured by the cover $E$. In practical situations, however, we are not interested in all kinds of covers. Therefore, we define additional criteria to characterize the notion of *relevant covers*. For example, it is clearly not interesting to consider those covers that do not minimize the rest. Then, given an appropriate ordering $\prec_d$ on cover rests, the notion of closest covers if defined below.

**Definition 2.** ***(Closest cover w.r.t.*** *$\prec_d$**)**. Let $\mathcal{L}$ be a DL in which the difference operation is computable, $\mathcal{T}$ (respectively, $Q$) be an $\mathcal{L}$-terminology (respectively, an $\mathcal{L}$-concept) and let $E$ be a conjunction of some defined concepts from $\mathcal{T}$. $E$ is a* ***closest cover*** *of $Q$ using $\mathcal{T}$ w.r.t. $\prec_d$ (or simply,* ***closest cover*** *of $Q$ using $\mathcal{T}$) iff: (i) $E$ is a* ***cover*** *of $Q$ using $\mathcal{T}$ and (ii) it does not exist a cover $E'$ of $Q$ using $\mathcal{T}$ such that $rest_{E'}(Q) \prec_d rest_E(Q)$.*

Closest covers correspond to those covers that minimize part of a query $Q$ not captured in a cover. Hence, they are clearly relevant rewritings in practical situations. For example, when contained or equivalent rewritings of $Q$ using $\mathcal{T}$ exist, they constitute the closest covers of $Q$.

However, usually it may not be interesting or efficient to compute all the possible closest covers. For example, in existing rewriting approaches, it does not make a lot of sense to compute all the rewritings contained in a given query. Usually, one is interested by either maximally-contained or equivalent rewritings. Similarly to the general framework introduced above, an additional optimality criterion can be used to select among the closest covers of a query $Q$, the most relevant ones. To abstract from particular optimality criterion, assume that we are provided with an ordering, noted $\prec_c$, on concept covers such that $E' \prec_c E$ means that the cover $E$ is better (or of higher quality) than the cover $E'$. As will be seen later, when defined appropriately the ordering $\prec_c$ can be used for example to capture the semantics of maximally-contained rewritings or, more interestingly, to maximize the user satisfaction with respect to a given set of Quality of Service (QoS) criteria.

Definition 3 given below characterizes the notion of *best covers* w.r.t. $\prec_d$, i.e., a closest cover that is considered as *optimal* according to the ordering $\prec_c$.

**Definition 3.** ***(Best cover w.r.t.*** *$(\prec_d, \prec_c)$**)**. Let $\mathcal{L}$ be a DL in which the difference operation is computable, $\mathcal{T}$ (respectively, $Q$) be an $\mathcal{L}$-terminology (respectively, an $\mathcal{L}$-concept) and let $E$ be a conjunction of some defined concepts from $\mathcal{T}$. Given two orderings $\prec_d$ and $\prec_c$, $E$ is a* ***best cover*** *of $Q$ using $\mathcal{T}$ w.r.t. $(\prec_d, \prec_c)$ iff: (i) $E$ is a* ***closest cover*** *of $Q$ using $\mathcal{T}$ w.r.t. $\prec_d$, and (ii) it does not exist a closest cover $E'$ of $Q$ using $\mathcal{T}$ w.r.t. $\prec_d$ such that $E' \prec_c E$.*

Finally, we are now able to provide a precise definition for the general *best covering problem*.

*Problem 1.* **($GBCP(\mathcal{T}, Q)$).** Let $\mathcal{L}$ be a DL in which the difference operation is computable, $\mathcal{T}$ (respectively, $Q$) be an $\mathcal{L}$-terminology (respectively, an $\mathcal{L}$-concept) and let $\prec_d$ be an ordering on difference descriptions and $\prec_c$ be an ordering on concept covers. The general best covering problem, denoted $GBCP(\mathcal{T}, Q)$, is the problem of computing all the best covers of $Q$ using $\mathcal{T}$ w.r.t. $(\prec_d, \prec_c)$.

Note that, problem 1 provides a general framework for covering concepts using terminologies. This framework can have different instantiations depending on the precise language $\mathcal{L}$ used to describe $\mathcal{T}$ and $Q$ as well as the precise definition of the orderings $\prec_d$ and $\prec_c$.

## 3 Investigated instances

Motivated by different application contexts, we have investigated three instances of the general best covering problem. A first line of demarcation between the studied instances comes from the properties of the difference operation used in each setting. We considered two cases in our work:

- **Languages in which the difference is semantically unique**. As described in [3, 4], in this case it is enough to consider $\prec_d$ as an ordering on the size of descriptions. In the sequel, we use the notation $\prec_d^{||}$ to denote that the ordering $\prec_d$ is defined on the size of descriptions.
- **The $\mathcal{ALN}$ language**. In the setting of $\mathcal{ALN}$ the difference operation is not semantically unique and produces a set of descriptions. In this case a more subtle definition of the ordering $\prec_d$ is required. In Section 4, we provide a definition for such an ordering based on an extension of the subsumption relation to sets of descriptions. To differentiate with the first case, we note such an ordering $\prec_d^S$ where the superscript $S$ is used to recall that the ordering $\prec_d$ is defined on sets of descriptions.

Table 1 presents the three instances of the best covering problem we have investigated in our work. The first two instances, respectively called $\mathcal{BCOV}(\mathcal{T}, Q)$ and *QoS*-$\mathcal{BCOV}(\mathcal{T}, Q)$, consider the family of languages equipped with structural subsumption[5]. Such a property ensures that the difference operation is semantically unique and can be determined using the *structural difference operation* [6]. Hence, both $\mathcal{BCOV}(\mathcal{T}, Q)$ and *QoS*-$\mathcal{BCOV}(\mathcal{T}, Q)$ use the ordering $\prec_d^{||}$ to characterize closest covers (i.e., they define $\prec_d = \prec_d^{||}$). However, these two instances differ in the specification of the ordering $\prec_c$ which was, in each case, motivated by the application context. In $\mathcal{BCOV}(\mathcal{T}, Q)$, the purpose was to select only the closest covers that contain as less as possible of *extra information* with

[5] Note that we use here the definition of structural subsumption in the sense of [6] which is different from the one usually used in the literature.

| **Instance Name** | $\mathcal{L}$ | **Ordering** $\prec_d$ | **Ordering** $\prec_c$ | **Applications** | **Refs** |
|---|---|---|---|---|---|
| $\mathcal{BCOV}(\mathcal{T}, Q)$ | Structural subsumption | $\prec_d^{\mid\mid}$ an ordering on size of the rest | $\prec_c^m$ an ordering on size of the missing information | Service discovery | [3] |
| *QoS*-$\mathcal{BCOV}(\mathcal{T}, Q)$ | Structural subsumption | $\prec_d^{\mid\mid}$ an ordering on size of the rest | $\prec_c^q$ an ordering w.r.t. a QoS function | Querying e-catalogs | [4] |
| $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ | $\mathcal{ALN}$ | $\prec_d^S$ an ordering on size of the rest | $\prec_c^S$ an ordering on size of the missing information | Service discovery | [1] |

**Table 1.** *Investigated instances of the general best covering problem.*

respect to a query $Q$. We call the part of a cover $E$ that is not contained in the description of the query $Q$ the *missing information*. Hence, the ordering $\prec_c^m$, used to instantiate $\prec_c$ in the context of $\mathcal{BCOV}(\mathcal{T}, Q)$, is defined as an ordering on the size of the missing information. In *QoS*-$\mathcal{BCOV}(\mathcal{T}, Q)$, however, the purpose was to select only the closest covers that maximize the user satisfaction with respect to a given set of Quality of Service (QoS) criteria (e.g., price, execution time, reliability, etc). Hence, in the context of *QoS*-$\mathcal{BCOV}(\mathcal{T}, Q)$, $\prec_c$ is instantiated as an ordering, noted $\prec_c^q$, that sorts the covers of a query $Q$ w.r.t. their *quality scores*.

Finally, the third instance, called $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$, is an extension of $\mathcal{BCOV}(\mathcal{T}, Q)$ to the language $\mathcal{ALN}$ in which the difference operation is not semantically unique (i.e., it produces sets of descriptions). Hence, $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ uses the ordering $\prec_d^S$ to define the closest covers (i.e., $\prec_d$=$\prec_d^S$) and the ordering $\prec_c^S$ to define the best covers (i.e., $\prec_c$=$\prec_c^S$). The ordering $\prec_c^S$ sorts covers of a query $Q$ by order of their missing information in the case where missing information are expressed as sets of descriptions.

## 4 The problem $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$

We consider the problem $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$, an extension of $\mathcal{BCOV}(\mathcal{T}, Q)$ to the language $\mathcal{ALN}$. A main feature that sets the $\mathcal{ALN}$ setting apart from the two previous ones lies in the possibility of nontrivial decomposition of the inconsistent concept $\bot$. As an example, consider the following two decompositions of $\bot$: $\bot \equiv (\leq 2R) \sqcap (\geq 4R) \equiv P \sqcap \neg P$, where $P$ denotes an atomic concept. Consequently, as highlighted below, new difficulties arise when dealing with the best covering problem in this context: *(i)* the difference operation is not semantically unique and produces sets of descriptions as a result. Therefore, to handle the best covering problem in this context there is a need for:

- an effective procedure to compute difference descriptions in $\mathcal{ALN}$, and
- a new formalization of the best covering problem. This is because the *rest* of a cover as well as the missing information are now expressed as sets of

descriptions and hence the orderings $\prec_d^{||}$ and $\prec_c^m$, respectively used in the case of $\mathcal{BCOV}(\mathcal{T}, Q)$ to instantiate $\prec_d$ and $\prec_c$, are no longer valid in this context.

*(ii)* The possibility to obtain inconsistent conjunctions of consistent concepts. Therefore, when computing best covers as conjunction of (consistent) defined concepts we have to ensure that only *'consistent'* covers are generated.

Altogether these points make the best covering problem much more complex to solve in the context of $\mathcal{ALN}$. We describe below a formalization of the best covering problem in this setting, then we sketch an approach to solve it.

### 4.1 Problem statement

Analogous to $\mathcal{BCOV}(\mathcal{T}, Q)$, in $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ we are interested by the computation of covers that contain as much as possible of *common information* with $Q$ and as less as possible of *extra information* with respect to $Q$. The main difficulty encountered when formalizing $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ lies in the specification of the orderings $\prec_d$ and $\prec_c$ which, in this context, will be respectively used for minimizing the rest and the missing information. To this end, we introduce below a slight extension of the subsumption relation to sets of descriptions and then we show how it can be used for specifying the orderings $\prec_d$ and $\prec_c$ in the context of $\mathcal{ALN}$.

**Definition 4.** ***(Subsumption between sets of descriptions )*** *Let $\mathcal{C} = \{c_1, \ldots, c_n\}$ and $\mathcal{D} = \{d_1, \ldots, d_m\}$ be two sets of $\mathcal{ALN}$-descriptions. The set $C$ is subsumed by $D$, noted $C \sqsubseteq_S D$ iff $\forall c_i \in C$, $\exists d_j \in D \,|\, c_i \sqsubseteq d_j$*

The orderings $\prec_d$ and $\prec_c$ of the general best covering problem $GBCP(\mathcal{T}, Q)$ are respectively instantiated in the setting of $\mathcal{ALN}$ using the orderings $\prec_d^S$ and $\prec_c^S$ defined below.

**Definition 5.** ***(The orderings $\prec_d^S$ and $\prec_c^S$)*** *Let $Q$ be an $\mathcal{L}$-concept description and $E$ and $E'$ two covers of $Q$ using $\mathcal{T}$.*

- *The ordering $\prec_d^S$ is defined as follows*
  *$E \prec_d^S E'$ iff $rest_{E'}(Q) \sqsubseteq_S rest_E(Q)$*
- *The ordering $\prec_c^S$ is defined as follows*
  *$E \prec_c^S E'$ iff $Miss_{E'}(Q) \sqsubseteq_S Miss_E(Q)$*

$\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ is then obtained from the general covering problem, by instantiating the language $\mathcal{L} = \mathcal{ALN}$ and the orderings $\prec_d = \prec_d^S$ and $\prec_c = \prec_c^S$. As in the case of $\mathcal{BCOV}(\mathcal{T}, Q)$, in $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ we are interested by the computation of the non redundant best covers.

*Problem 2.* ($\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$).

Let $\mathcal{T}$ (respectively, $Q$) be an $\mathcal{ALN}$-terminology (respectively, an $\mathcal{ALN}$-concept). $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ is the problem of computing all the non redundant best covers of $Q$ using $\mathcal{T}$ w.r.t. $(\prec_d^S, \prec_c^S)$.

The problem $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ is NP-hard. This complexity result is easily derived from the complexity of the $\mathcal{BCOV}(\mathcal{T}, Q)$ problem [3].

### 4.2 Dealing with $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$

In this section, we turn our attention to the computational problem underlying $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$. The aforementioned features of $\mathcal{ALN}$ make this problem much more complex to solve than the previous instances. A complete description of the solution to $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ is lengthly and technical. We sketch below the main steps of our approach for handling $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$. Technical details are given in [1] where an algorithm, called *compute*$\mathcal{ALN}$*BCov*, to solve $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ is proposed.

The first step in dealing with $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ consists in the design of an algorithm that implements the difference operation between $\mathcal{ALN}$ descriptions [1]. Then, with such an algorithm at hand, we investigated the computational problem underlying $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$. Let $\mathcal{C}_\mathcal{T}$ be the set of defined concept names that appear in $\mathcal{T}$. The search space of $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ is the power set of $\mathcal{C}_\mathcal{T}$. Unfortunately, a similar approach to the one used for $\mathcal{BCOV}(\mathcal{T}, Q)$ cannot be exploited here to avoid an exhaustive exploration of this search space. Indeed, in the case of $\mathcal{BCOV}(\mathcal{T}, Q)$ [3], a full characterization of best covers in terms of hypergraph transversals yields to a reduction of $\mathcal{BCOV}(\mathcal{T}, Q)$ to a computation of the minimal transversals with minimal cost of an associated hypergraph. The characterization of best covers in $\mathcal{ALN}$ context is more complex.

The approach we have developed to cope with $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ embody a *divide-and-conquer* strategy that enables to progressively reduce the search space. We decompose the $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ problem into a set of smaller tasks by considering separately each criterion that must be satisfied by best covers in this setting. Then, for each criterion we provide a characterization that enables to confine the search space.

Observe that a solution $E$ of an $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ problem must satisfy the following conditions: (Cond-1:) $E \sqcap Q \not\equiv \bot$, and (Cond-2:) $E$ is a cover of $Q$ (i.e., $Q \not\in_{\equiv} Q - lcs(Q, E)$), and (Cond-3:) $E$ is a closest cover of $Q$ (i.e., $Rest_E(Q)$ is minimal w.r.t. $\prec_d^S$).

For each of these conditions we provide a characterization that yields to an upper/lower bound in the search space. More precisely, given an $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ problem and let $\mathcal{C}_\mathcal{T}$ be the set of defined concept names appearing in $\mathcal{T}$, we provide full characterizations for the following borders:

- $S_{cons}$: the greatest subsets of $\mathcal{C}_\mathcal{T}$ that satisfy Cond-1.
- $S_{couv}$: the greatest subsets of $\mathcal{C}_\mathcal{T}$ that satisfy Cond-1 and Cond-2.
- $I_{couv}$: the smallest subsets of $\mathcal{C}_\mathcal{T}$ that satisfy Cond-1 and Cond-2.
- $S_{rest}$: the greatest subsets of $\mathcal{C}_\mathcal{T}$ that satisfy Cond-1, Cond-2 and Cond-3.
- $I_{rest}$: the smallest subsets of $\mathcal{C}_\mathcal{T}$ that satisfy Cond-1, Cond-2 and Cond-3.

Based on the characterizations of the aforementioned borders, the proposed algorithm *compute*$\mathcal{ALN}$*BCov* breaks $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$ into the following subproblems: *(1)* Computation of $S_{cons}$, *(2)* Computation of $S_{couv}$ and $I_{couv}$, *(3)* Computation of $S_{rest}$ and $I_{rest}$, and *(4)* Computation of the best covers.

Briefly stated, the algorithm *compute*$\mathcal{ALN}$*BCov* proceeds in four steps each of which consisting in the resolution of one subproblem. The first three steps

exploit the provided characterizations of the aforementioned borders to progressively confine the search space of $\mathcal{ALN}$-$\mathcal{BCOV}(\mathcal{T}, Q)$. Unfortunately, due to the non-monotonic nature of the ordering $\prec_c^S$, we were not able to provide a full characterization for the last step. Consequently, to handle step 4, the algorithm *compute*$\mathcal{ALN}$*BCov* enumerates all the covers confined between $S_{rest}$ and $I_{rest}$ and test for minimality w.r.t. $\prec_c^S$. Due to previous reductions of the search space, we expect step 4 to be still handled efficiently in practical cases. The implementation of *compute*$\mathcal{ALN}$*BCov* is an ongoing work. In our future work, we plan to conduct experiments on real cases as well as synthetic ontologies in order to evaluate the performance of this algorithm.

## 5 Conclusion

This paper focuses on the problem of covering concepts using terminologies. This work has relation with exiting works on concept/query rewriting. The salient feature of our approach is to use a measure of *semantic distance* between concepts, instead of subsumption relation, to define rewritings, thereby enabling a more *flexible* rewriting process. We provided a formal definition of a general framework for covering concepts using terminologies and described some already investigated instances of this problem. We then studied a new instance of the covering problem in the context of the $\mathcal{ALN}$ language, in which the difference operation is not semantically unique.

## References


1. C. Rey B. Benatallah, A. Leger and F. Toumani. Covering concepts using terminologies in the ALN language. Technical report, LIMOS, Clemont-Ferrand, France, http://www.isima.fr/rey/alnBcov.pdf, 2007.
2. F. Baader, R. Küsters, and R. Molitor. Rewriting concepts using terminologies – revisited. Report 00-04, LTCS, RWTH Aachen, Germany, 2000.
3. B. Benatallah, M-S. Hacid, A. Leger, C. Rey, and F. Toumani. On automating Web services discovery. *VLDB J.*, 14(1):84–96, 2005.
4. B. Benatallah, M-S. Hacid, H-Y. Paik, C. Rey, and F. Toumani. Towards semantic-driven, flexible and scalable framework for peering and querying e-catalog communities. *Information Systems*, 31(4-5), 2006.
5. Alon Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.
6. G. Teege. Making the difference: A subtraction operation for description logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *KR'94*, San Francisco, CA, 1994. Morgan Kaufmann.