# Complete and Incomplete Knowledge in Logical Information Systems

Sébastien Ferré**

IRISA, Campus Universitaire de Beaulieu, 35042 RENNES cedex,
ferre@irisa.fr

**Abstract** We present a generalization of Levesque's logic All I Know by presenting it as an extension of usual modal logics. We study how logic All I Know can represent complete and incomplete knowledge in Logical Information Systems. In these information systems, a knowledge base is a collection of objects (e.g., files, bibliographical references) described in a logical way, furthermore in the same logic used for expressing queries. We show that usual All I Know (transitive and euclidean accessibility relation) is convenient for representing complete knowledge, but not for incomplete knowledge. For this, we use *serial* All I Know (serial accessibility relation).

## 1 Introduction

Most common paradigms of information systems are *hierarchical systems* (e.g., File Systems), *relational databases*, and *deductive databases*. While the first paradigm is based on *navigation* in a hierarchy built by hand, the other ones are based on a *querying* language (e.g., SQL, first-order logic). However, it appears in practice that both navigation and querying are needed in information retrieval [GMA93]. Moreover, the large diversity of applications brings the need for an information system that is generic in its description and querying languages. For instance, considering a program library, one would like to search for functions by their type, which means that a dedicated logic is necessary for reasoning on types (e.g. [Di 95]). None of the three above paradigms satisfies any of these two features. There exists some work about the combination of navigation and querying [GJSO91,GMA93], but they use only a simple logic based on sets of attributes. There also exists some information systems based on various logics (modal logic [Rei92], description logic [DJ94]), but they are not generic in the logic used, and they offer no integrated means for navigation.

We designed a *Logical Information System* (LIS) [FR00a] in which descriptions of objects and queries are expressed in the same arbitrary logic. LIS is generic in the sense that a logic just has to be plugged in to get a dedicated information system. LIS combines navigation and querying by unifying queries and answers through Logical Concept Analysis [FR00b], which is a logical generalization of Formal Concept Analysis [Wil82]. We realized a prototype and made

** This author is supported by a scholarship from CNRS and Région Bretagne

some experiments. They rapidly showed that deduction capabilities in usual logic (e.g., propositions on valued attributes for a bibliography) were not fully satisfying. For instance, in a bibliographical application, the query $\neg$ *author* : "*Jones*" releases no answer because bibliographical references are described by the authors they have, and not by the authors they have not: i.e., negative facts are not explicitly represented. Thus, following Lesvesque and Reiter [Rei92], we argue that the logic must be equipped with epistemic features to enhance the expressiveness of queries. This would allow to describe what is known about the external world, and to query what is known by the information system.

This paper aims at showing how complete and incomplete knowledge can be represented in a Logical Information System (LIS). Section 2 gives some details on LIS and explicits the expressiveness problem. Section 3 shows that Levesque's logic All I Know (AIK) is well suited for representing complete knowledge. Section 4 explains why All I Know has to be somewhat generalized for representing incomplete knowledge. Both these sections also present some idiomatics that make the use of AIK easier and more natural for our purpose. Finally, Section 5 conclude the paper and draw some perspectives.

## 2 Logical Information Systems and Non-Monotonic Reasoning

Before focusing on the main issue of this paper, i.e., representing and reasoning on complete and incomplete knowledge, we first need to give more details on what we mean by a Logical Information System. As for deductive databases, both the knowledge base and queries are expressed in a logical way. A difference is that this knowledge base is composed of objects (e.g., files, bibliographical references, web pages) rather than of relations, and is called a *logical context* by reference to Logical Concept Analysis (LCA) [FR00b] that serves as a framework for LIS. Such a context is a triple $(\mathcal{O}, \langle\mathcal{L}; \models\rangle, d)$ where $\mathcal{O}$ is a set of objects, $\mathcal{L}$ is an almost arbitrary logical language equipped with a deduction relation $\models$, and $d$ is the mapping that describes objects with logical formulas. The only restriction on the deduction relation is that it forms a lattice (at least a partial order). This is important for the choice of AIK in Sections 3 and 4.

In this section, we consider as a running example a context where objects are bibliographical references (like BibTeX) and the logic is the propositional one whose atoms have been replaced by valued attributes, where values are of type string or integer. The following table shows both the content and the description of an object $o$:

```
c(o) = @InProceedings{FerRid2000b,
  author = {Sébastien Ferré and Olivier Ridoux},
  title = {A Logical Generalization of Formal Concept Analysis},
  year = {2000},
  keywords = {concept analysis, logic, context, information system} }
```

```
d(o) =
  type: is "InProceedings" ∧
  author: is "Sébastien Ferré and Olivier Ridoux" ∧
  title: is "A Logical Generalization of Formal Concept Analysis" ∧
  year: 2000 ∧
  keywords: is "concept analysis, logic, context, information system"
```

Now, as queries are expressed in the same logic as object descriptions, we test that an object $o$ is an answer to a query $q$ simply by evaluating $d(o) \models q$. For instance, the above object is an answer of the following queries: `author: contains "Ridoux"`, `year: 1990..2000`, `keywords: contains "logic"` $\vee$ `keywords: contains "language"`.

This is the querying part, but LIS also offers some navigation in a context. In our prototype, this navigation mimicks usual UNIX file system commands `cd` and `ls`, where paths are replaced by formulas. Command `cd` $q$ restricts the working place to answers of query $q$. Command `ls` lists some relevant links to sub-places, and objects that are in the working place while being in none of the sub-places. The important point is that these links (1) are formulas of the logic $\mathcal{L}$ (and so can be passed as argument to command `cd`), (2) are not defined a priori, but depends of the working place and the context through LCA, and (3) guarantees to avoid empty places, and the ability to reach every one object through navigation. So, links can be seen as relevant suggestions of LIS for subsequent selections by command `cd` (see [FR00a] for more details). This leads to a dialogue between LIS and the user, where both can answer to questions by assertions or questions.

A complete example of such a dialogue is given in Table 1. The left part of this table shows what is really displayed by our prototype, and the right part is an English translation of the dialogue. On the 2nd query, the question of the user is so open, that LIS only answers by questions. On the 3rd query, the user replies to one of these questions (`title: *`) by an assertion; but on the 4th query, he sends back to LIS another of these questions (`author: *`) to get some relevant suggestions. On the 5th query, he just selects a suggested author, `"Wille"`, and then gets his co-authors about Concept Analysis with the 6th query. On the 7th query, he selects a co-author and finally finds an object at the 8th query.

This example also shows that navigation may start with no knowledge of the contents of a LIS, neither the objects and descriptions, nor the logic in use. It is the answers to command `ls` that informs gradually the user on both the logic and the objects.

Up to now, no negative query has been considered. This is because they have empty answers since there is no reason to assert some negative facts about objects in the bibliographical application. For instance, we do not want to mention the fact that an object satisfies $\neg$ `author: is "Wille"`, while we want that this fact be deducible from its description because we have a complete knowledge on this object. This is obviously a formulation of the well-know *Closed World Assumption* (CWA), which led to many formalisms for non-monotonic reasoning (Minimal Belief and Negation as Failure [Lif91], Default Logic [Rei80], Auto-Epistemic Logic [Moo85], Circumscription [McC86], All I Know [Lev90]). However, logics used in LIS need a monotonic deduction relation $\models$ because of the framework on which it is based, i.e., Logical Concept Analysis. This framework requires that the logic has a deduction relation that forms a lattice. In other words, we need to apply the CWA hypothesis locally on formulas (especially on

| | |
|---|---|
| (1) pwd | (1) What is currently selected? |
| / | All objects. |
| (2) ls | (2) What do you have? |
| 209 type: * | What kind of type do you want? |
| 209 author: * | What kind of author do you want? |
| 209 year: .. | What kind of year do you want? |
| 209 title: * | What kind of title do you want? |
| 209 object(s) | 209 objects are currently selected. |
| (3) cd title: contains "Concept Analysis" | (3) I want the title contains "Concept Analysis"! |
| (4) ls author: * | (4) What kind of author do you have (for this)? |
| 1 author: contains "Mineau" | I have 1 object with author "Mineau"! |
| 1 author: contains "Lehmann" | I have 1 object with author "Lehmann"! |
| 1 author: contains "Stumme" | I have 1 object with author "Stumme"! |
| 1 author: contains "Prediger" | I have 1 object with author "Prediger"! |
| 3 author: contains "Wille" | I have 3 objects with author "Wille"! |
| 4 object(s) | 4 objects are currently selected. |
| (5) cd author: contains "Wille" | (5) I want the author contains "Wille"! |
| (6) ls | (6) What kind of author do you have (yet)? |
| 1 author: contains "Mineau" | I have 1 object with author "Mineau"! |
| 1 author: contains "Lehmann" | I have 1 object with author "Lehmann"! |
| 1 author: contains "Stumme" | I have 1 object with author "Stumme"! |
| 3 author: contains "Wille" | What kind of author "Wille" do you want? |
| 3 object(s) | 3 objects are currently selected. |
| (7) cd author: contains "Mineau" | (7) I want the author contains "Mineau"! |
| (8) ls | (8) What do you have? |
| 200 Guy W. Mineau and Gerd Stumme and Rudolf Wille. "Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis". INPROCEEDINGS, 1999. | |
| 1 object(s) | 1 object is currently selected. |
| (9) pwd | (9) What is currently selected? |
| /author: contains "Wille" | Objects with authors "Wille" and "Mineau", |
| /author: contains "Mineau" | and whose title contains "Concept Analysis". |
| /title: contains "Concept Analysis" | |

**Table1.** Example of User/LIS Dialogue in the bibliographical context.

object descriptions) rather than globally on the deduction relation. Levesque's logic All I Know (also noted $\mathcal{ONL}$) precisely defines such an operation. Moreover, $\mathcal{ONL}$ logic is proved to encompass all the non-monotonic formalisms cited above [Che94], and there exists a proof method for it [Ros00]. Next section presents the $\mathcal{ONL}$ logic and how it can be used in a LIS to represent complete knowledge.

## 3 Expressing Complete Knowledge

### 3.1 Logic $\mathcal{ONL}$

In this section, we recall the formalization of logic $\mathcal{ONL}$ [Ros00]. The logical language $\mathcal{ONL}$ is defined as a propositional language with connectives $\wedge$, $\neg$ ($\vee$ and $\Rightarrow$ are defined as abbrevations), whose atomic propositions belong to an infinite set $\mathcal{A}$, and that is extended with modal operators $K$, $N$, $O$. Logic $\mathcal{ONL}$ can be given a Kripke semantics: the *worlds* are valuations of $\mathcal{A}$ in $\{TRUE, FALSE\}$ extended in the usual way to propositional connectives, and the *accessibility relation* is defined as a relation between these worlds.

**Definition 1** *Let $w$ be a world , and let $R$ be a transitive and euclidean accessibility relation. We say that a structure $(w, R)$ is a* model *of a*

*formula* $\phi \in \mathcal{ONL}$*, and we note* $(w, R) \models \phi$*, iff the following conditions hold (*$R(w)$ *denotes the set of successor worlds of* $w$ *through* $R$*):*
*1. if* $\phi \in \mathcal{A}$*, then* $(w, R) \models \phi$ *iff* $w(\phi) = TRUE$*;*
*2. if* $\phi = \neg\phi_1$*, then* $(w, R) \models \phi$ *iff* $(w, R) \not\models \phi_1$*;*
*3. if* $\phi = \phi_1 \wedge \phi_2$*, then* $(w, R) \models \phi$ *iff* $(w, R) \models \phi_1$ *and* $(w, R) \models \phi_2$*;*
*4. if* $\phi = K\phi_1$*, then* $(w, R) \models \phi$ *iff for every* $w' \in R(w)$*,* $(w', R) \models \phi_1$*;*
*5. if* $\phi = N\phi_1$*, then* $(w, R) \models \phi$ *iff for every* $w' \notin R(w)$*,* $(w', R) \models \phi_1$*;*
*6. if* $\phi = O\phi_1$*, then* $(w, R) \models \phi$ *iff for every* $w'$*,* $w' \in R(w)$ *iff* $(w', R) \models \phi_1$*.*

Logic $\mathcal{ONL}$ can be equipped with a monotonic deduction relation $\models_{\mathcal{ONL}}$, that enables to compare object descriptions with queries, and also queries themselves.

**Definition 2** *A formula* $\phi \in \mathcal{ONL}$ *entails a formula* $\psi \in \mathcal{ONL}$ *(denoted* $\phi \models_{\mathcal{ONL}} \psi$*) iff* $\phi \Rightarrow \psi$ *is* $\mathcal{ONL}$*-valid, i.e., for every Kripke structure* $(w, R)$ *where* $R$ *is transitive and euclidean,* $(w, R) \models \phi \Rightarrow \psi$*.*

In order to better understand modal operators, we prove the following lemma.

**Lemma 1** *If* $\phi$ *is a* $\mathcal{ONL}$*-formula and* $W_R(\phi) = \{w | (w, R) \models \phi\}$ *is the set of worlds where* $\phi$ *is true, then for every structure* $(w, R)$
*1.* $(w, R) \models K\phi$ *iff* $R(w) \subseteq W_R(\phi)$*;*
*2.* $(w, R) \models N\neg\phi$ *iff* $R(w) \supseteq W_R(\phi)$*;*
*3.* $(w, R) \models O\phi$ *iff* $R(w) = W_R(\phi)$*.*

**Proof:** Proofs for each item is directly obtained from Definition 1, and are similar. So, we detail the proof only for modality $K$.
(1) $(w, R) \models K\phi \iff \forall w' \in R(w) : (w', R) \models \phi$
$\iff \forall w' : w' \in R(w) \Rightarrow w' \in W_R(\phi) \iff R(w) \subseteq W_R(\phi)$. ■

This lemma shows that in a model $(w, R)$ of a modal formula, what is important is not the initial world $w$, neither the accessibility relation itself, but the set of successor worlds $R(w)$. Therefore, these modal formulas $M\phi$ ($M \in \{K, N\neg, O\}$) describe sets of models of $\phi$, rather than individual models of $\phi$. For instance, modal formula $K\phi$ describes some subsets of $W_R(\phi)$, in which $\phi$ is always true but not only $\phi$. So, $K\phi$ can be read as "at least $\phi$". Dualy, modal formula $N\neg\phi$ can be read as "at most $\phi$", and modal formula $O\phi$, which is semantically equivalent to $K\phi \wedge N\neg\phi$ according to Definition 1, can be read as "exactly $\phi$" or "all I know is $\phi$" (hence, the original name of logic $\mathcal{ONL}$ [Lev90]).

### 3.2 Completing Knowledge with Logic $\mathcal{ONL}$

We now consider the representation of complete knowledge with logic $\mathcal{ONL}$. For instance, in the context of a bibliography, we want to represent the set of authors of a document $o$. If $A$ and $B$ are authors, we can logically describe this document with the propositional formula $d(o) = A \wedge B$. Then, $o$ is an answer of the query $q = A$ (because $d(o)$ entails $q$), but is not an answer of the query $q' = \neg C$ (because $d(o)$ does not entail $q'$). But, if the knowledge expressed in $d(o)$ is complete, we want to deduce from it that $C$ is not an author of the considered document.

For this, we propose to complete object decriptions by embedding them in modal operator $O$ (this idea has already been proposed, e.g., in the conclusion of [Rei92]). In our example, we establish the following entailments (by the mean of a tableau calculus [Ros00])

$$O(A \wedge B) \models_{\mathcal{ONL}} K(A), \quad O(A \wedge B) \models_{\mathcal{ONL}} \neg K(C),$$

which can be translated in English as "if $A$ and $B$ are authors and the only ones, then $A$ is an author and $C$ is not".

### 3.3 Idiomatics for Complete Knowledge

Following Reiter [Rei92], we think that modal operators are not convenient for naive users, and we introduce some idiomatics for expressing descriptions and queries in a more natural way:

**description:** $= d \equiv Od$, where $d$ is a non-modal formula describing an object: ex., $= A \wedge \neg B$;

**query:** proposition whose atoms are either $+q_1 \equiv Kq_1$ or $-q_1 \equiv \neg Kq_1$, $q_1$ being a non-modal formula: ex., $(+(A \wedge B) \vee +\neg B) \wedge -C$.

The precise meaning of these idiomatics is captured by the following lemma.

**Lemma 2** *Let $\models_{\mathcal{L}}$ be a deduction relation on non-modal formulas (propositions). For every non-modal formulas $d, q, q'$*
*1. $= d \models_{\mathcal{ONL}} +q$ iff $d \models_{\mathcal{L}} q$;*
*2. $= d \models_{\mathcal{ONL}} -q$ iff $d \not\models_{\mathcal{L}} q$.*

**Proof:**
(1) $= d \models_{\mathcal{ONL}} +q \iff (Od \Rightarrow Kq)$ is valid (Definition 2)
$\iff \forall (w,R) : (w,R) \models Od) \Rightarrow (w,R) \models Kq)$ (Definition 1)
$\iff \forall (w,R) : R(w) = W_R(d) \Rightarrow R(w) \subseteq W_R(q)$ (Lemma 1)
$\iff \forall R : W_R(d) \subseteq W_R(q) \iff d \models_{\mathcal{L}} q$ ($d, q$ are non-modal formulas).
(2) similar to proof of (1). ■

If a non-modal formula $d$ represents the knowledge we have about an object, then $= d \models_{\mathcal{ONL}} +q$ means "I know $q$" because this is equivalent to $d \models_{\mathcal{L}} q$ (Lemma 2). Conversaly, $= d \models_{\mathcal{ONL}} -q$ means "I do not know $q$". Now, if we use $= d$ to represent a complete knowledge, that is everything unknown is considered as false, we must read $+q$ as "$q$ is *true*", and $-q$ as "$q$ is *false*". Here, truth and falsity are expressed from a *knowledge point of view*, whereas from a *real world point of view*, they would be expressed by $q$ and $\neg q$.

Lemma 2 shows how a subset of logic $\mathcal{ONL}$ can be used to represent complete knowledge. While this subset is simple, it enables some fine distinctions. First, $+q_1 \vee +q_2 \models_{\mathcal{ONL}} +(q_1 \vee q_2)$ while the converse is false ($= (q_1 \vee q_2)$ is a counter-example): $+q_1 \vee +q_2$ represents alternative (either $q_1$ or $q_2$ is known as true), whereas $+(q_1 \vee q_2)$ represents some indetermination in knowledge ($q_1 \vee q_2$ is known as true, but which one is true can be unknown). Second, $+\neg q \models_{\mathcal{ONL}} -q$ while the converse is false ($= d$ is a counter-example if $d \not\models_{\mathcal{L}} q$): $+\neg q$ represents explicit falsity ($q$ is known as false), whereas $-q$ represents absence of truth ($q$ is not known as true, but it is not necessary known as false either). In the following section, we show that even more fine knowledge distinctions can be made by means of logic $\mathcal{ONL}$, e.g., taking into account incomplete knowledge.

# 4 Expressing Incomplete Knowledge

From Lemma 2, it follows that for every non-modal description $d$, and every non-modal query $q$, $= d$ always entails either $+q$ or $-q$, i.e., we have a complete knowledge about descriptions embedded by modal operator $O$. We recall that for every formula $\phi \in \mathcal{ONL}$, $O\phi$ can be defined as $K\phi \wedge N\neg\phi$, which can be read as "at least $\phi$ and at most $\phi$". Each part of this definition expresses an incomplete knowledge, and this is the conjunction of the application of both parts to a *same* formula that forms a complete knowledge. The issue of this section is to find how expressing incomplete knowledge by using modalities $K$ and $N$ in a less tight combination than in the definition of modality $O$.

## 4.1 Examples and Problems

As in Section 3, we consider as examples the representation of authors in a bibliographic application. Authors are simply represented by atoms ($A, B, C, D, \ldots$). We present in the following some modal formulas with the expected meaning:

1. $d(o_1) = K(A \wedge B)$: "A and B are certainly authors, but there are possibly other ones";
2. $d(o_2) = N\neg(A \wedge B \wedge C)$: "A, B, and C are the only possible authors, but we do not know exactly which ones are effectively";
3. $d(o_3) = K(A \wedge B) \wedge N\neg(A \wedge B \wedge C)$: "A and B are certainly authors, C is possibly also, but there are no other one".

In order to know if these formulas meets their expected meaning, we look at what can be deduced from them in $\mathcal{ONL}$. The following table summaries some entailments for each example. For instance, it shows that $d(o_3)$ entails $+A$, $-D$, and $?C$ means that neither $+C$, nor $-C$ is deducible.

| $d$ | $A$ | $\neg A$ | $C$ | $D$ |
|---|---|---|---|---|
| $o_1$ | + | ? | ? | ? |
| $o_2$ | ? | − | ? | − |
| $o_3$ | + | − | ? | − |

Two problems are revealed by these examples. The first one is that half of the above table is filled with ?, which represents an extra-logical property ($d \models_{\mathcal{ONL}} ?q$ iff $d \not\models_{\mathcal{ONL}} +q$ and $d \not\models_{\mathcal{ONL}} -q$). This means that we can not ask for documents where author $C$ is ***possible***, that is where $C$ is neither true, nor false. This notion of possibility is necessary with an incomplete knowledge, which we try to represent, but we want to represent it in the logic itself. For this, we need to complete not the incomplete knowledge itself, but rather the knowledge incompleteness. In other words, we want to represent knowledge about knowledge.

The second problem is that $\neg A$ is possible in $d(o_1)$, whereas $A$ is true. This means that a model $(w, R)$ where $R(w) = \emptyset$ is considered, i.e., object $o_1$ is considered as an impossible object. As objects do exist in the real world, we want to exclude this possibility, and to deduce that $\neg A$ is false like in $d(o_2)$. These two problems are addressed in the next section.

### 4.2 Completing Knowledge Incompleteness by Generalizing Logic $\mathcal{ONL}$

The first problem revealed in previous section is about expressing a *possible* fact $q$, that is a fact which is neither true $(+q)$, nor false $(-q)$. A usual representation of possibility in modal logics is $-q \wedge -\neg q$, which is already expressible in $\mathcal{ONL}$, but it expresses possibility from a real world point of view. A naive expression of possibility from a knowledge point of view could be $\neg + q \wedge \neg - q$. But this expression is equivalent to $-q \wedge +q$, which is easily shown contradictory (see Section 3.3). This problem is in fact similar to the one of Section 3.2 about complete knowledge, and it is tempting to adopt a similar solution, i.e., to embed object description of Section 4.1 by modal operator $O$:

1. $d(o_1)' = OK(A \wedge B)$;
2. $d(o_2)' = ON\neg(A \wedge B \wedge C)$;
3. $d(o_3)' = O(K(A \wedge B) \wedge N\neg(A \wedge B \wedge C))$.

Unfortunately, these formulas are not $\mathcal{ONL}$-satisfiable (see Example 2 of Section 2 in [Ros00]). For explaining this, we first need to notice that for transitive and euclidean accessibility relations, world set $R(w)$ does not depend on world $w$. In this case, a structure $(w, R)$ can be replaced by a structure $(w, W)$ where $W$ is the constant world set $R(w)$. Now, let $(w, W)$ be a model of $OK(A \wedge B)$. If $W \subseteq W(A \wedge B)$, then for every world $w'$, $(w', W)$ is a model of $K(A \wedge B)$ as $W = R(w')$ (Definition 1). Then, from semantics of modality $O$, every world $w'$ belongs to $W$ since $R(w) = W$. This is contradictory because $A \wedge B$ is not a tautology. However, $W \not\subseteq W(A \wedge B)$ contradicts that $(w, W)$ is a model of $OK(A \wedge B)$.

In the same way as formula $O(A \wedge B)$ enables us to reason on all models of $A \wedge B$, we would like that formula $OK(A \wedge B)$ enables us to reason on all models of $K(A \wedge B)$, i.e., on all structure $(w, R)$ such that $R(w) \subseteq W_R(A \wedge B)$. For this, it is necessary that $R(w)$ does depend on world $w$, in order to keep the meaning of an incomplete knowledge. This is why we propose to generalize logic $\mathcal{ONL}$ by removing transitive and euclidean conditions on the accessibility relation from Definition 1. Therefore, we can see logic $\mathcal{ONL}$ as a usual modal logic where $K$ is the main modal operator defined on accessible worlds $R(w)$, whereas $N$ is a dual modal operator defined on unaccessible worlds $\overline{R(w)}$, and $O$ is simply defined as a combination of $K$ and $N$ $(O\phi \equiv K\phi \wedge N\neg\phi)$. Then, a whole family of $\mathcal{ONL}$-logic can be derived by applying various conditions on the accessibility relation, as it is done for usual modal logics [Bow79]. For instance, usual logic $\mathcal{ONL}$ has a transitive and euclidean accessibility relation and can so be renamed as K45-$\mathcal{ONL}$, whereas our generalization leads to an arbitrary accessibility relation and can be named as K-$\mathcal{ONL}$.

**Definition 3** *Semantics and entailment of logic K-$\mathcal{ONL}$ are defined as in Definitions 1 and 2, except there are no condition on the accessibility relation.*

With logic K-$\mathcal{ONL}$, the knowledge is stratified because the accessibility relation is not transitive. Object description $d(o_1)' = OK(A \wedge B)$ is now satisfiable and can be read as three levels of knowledge: a model of

1. $A \wedge B$ is a world $w$ satisfying both $A$ and $B$;
2. $K(A \wedge B)$ is a world $w$ whose $R(w)$ is a set of models of $A \wedge B$;
3. $OK(A \wedge B)$ is a world $w$ whose $R(w)$ collects all models of $K(A \wedge B)$.

This description can be seen as a complete knowledge about an incomplete knowledge about the object $o_1$: "All I know about object $o_1$ is that it has at least authors $A$ and $B$". It allows the following entailment:

$$OK(A \wedge B) \models_{K-\mathcal{ONL}} KK(A) \wedge (\neg KK(\neg A) \wedge \neg K \neg K(\neg A)) \wedge (\neg KK(C) \wedge \neg K \neg K(C)).$$

This means that $d(o_1)'$ entails that one knows that $A$ is true (i.e., is an author), but that one do not know about $\neg A$ and $C$. So, the fact that $C$ is a possible author is correctly expressed by $(\neg KK(C) \wedge \neg K \neg K(C))$, which solves the first problem presented in Section 4.1. On the contrary, the second problem is not solved because $\neg A$ is proved possible rather than false as expected. The reason is that a Kripke structure $(w, R)$ where $R(w) = \emptyset$ is considered as a model of $K(A \wedge B)$, which means that an impossible object is considered. This is not convenient in our Logical Information System where objects do exist in the real world. To exclude these empty models, we just add a condition of *seriality* on the accessibility relation [Bow79], which forces any world to have at least one successor: we obtain logic KD-$\mathcal{ONL}$.

**Definition 4** *Semantics and entailment of logic KD-$\mathcal{ONL}$ are defined as in Definitions 1 and 2, except the accessibility relation is serial.*

This time, we get the expected entailment

$$OK(A \wedge B) \models_{KD-\mathcal{ONL}} KK(A) \wedge K \neg K(\neg A) \wedge (\neg KK(C) \wedge \neg K \neg K(C)),$$

which means that one knows that $\neg A$ is false. The next section introduces some idiomatics and generalizes examples presented in this section.

### 4.3 Idiomatics for Incomplete Knowledge

As for complete knowledge, we introduce some idiomatics:

**description:** $[d, d'] \equiv O(Kd \wedge N \neg(d \wedge d'))$, where $d$ and $d'$ are non-modal formulas, represents a kind of knowledge interval where $d$ represents what is known as true, and $d'$ represents what is known as possible, all the rest being considered as implicitly false. The three object descriptions given in Section 4.2 can be represented in this way:

1. $d(o_1)' = [A \wedge B, \bot]$ (because $N \neg \bot \equiv \top$);

2. $d(o_2)' = [\top, A \wedge B \wedge C]$ (because $K\top \equiv \top$);
3. $d(o_3)' = [A \wedge B, C]$.

A complete knowledge $d$ can also be represented as $[d, \top] \equiv OOd$.

**query:** a proposition whose atoms are either $+q_1 \equiv KKq_1$, or $-q_1 \equiv K\neg Kq_1$, or $?q_1 \equiv (\neg KKq_1 \wedge \neg K\neg Kq_1)$, $q_1$ being a non-modal formula: ex., $(+(A \wedge \neg B) \vee ?C) \wedge -D$.

It must be noticed that these idioms of descriptions and queries are not exhaustive. But they fit well enough usual needs in information systems. Anyway, these idioms are not limitative and it is always possible to come back to full logic $\mathcal{ONL}$. The following lemma captures the meaning of these idiomatics by relating them to the non-modal propositional logic.

**Lemma 3** *Let $\models_{\mathcal{L}}$ be a deduction relation on non-modal formulas (propositions). For every non-modal formulas $d, d', q$*
*1. $[d, d'] \models_{KD-\mathcal{ONL}} +q$ iff $d \models_{\mathcal{L}} q$;*
*2. $[d, d'] \models_{KD-\mathcal{ONL}} -q$ iff $d \wedge q \models_{\mathcal{L}} \bot$ or $d \wedge d' \not\models q$;*
*3. $[d, d'] \models_{KD-\mathcal{ONL}} ?q$ otherwise.*

**Proof:**
(1) $[d, d'] \models_{KD-\mathcal{ONL}} +q \iff O(Kd \wedge N\neg(d \wedge d')) \models_{KD-\mathcal{ONL}} KKq$
$\iff \forall(w, R) : (w, R) \models O(Kd \wedge N\neg(d \wedge d')) \Rightarrow (w, R) \models KKq$ (Definition 2)
$\iff \forall(w', R) : (w', R) \models Kd$ and $(w', R) \models N\neg(d \wedge d') \Rightarrow (w', R) \models Kq$ (semantics of $O$, $K$, and $\wedge$)
$\iff \forall(w', R) : R(w') \subseteq W_R(d)$ and $R(w') \supseteq W_R(d \wedge d') \Rightarrow R(w') \subseteq W_R(q)$ (Lemma 1)
$\iff \forall R : W_R(d) \subseteq W_R(q)$ (take $R(w') = W_R(d)$)
$\iff d \models_{\mathcal{L}} q$ ($d, q$ are non-modal formulas).
(2) similar to proof of (1).
(3) $[d, d'] \models_{KD-\mathcal{ONL}} ?q \iff O(Kd \wedge N\neg(d \wedge d')) \models_{KD-\mathcal{ONL}} \neg KKq \wedge \neg K\neg Kq$
$\iff \forall(w, R) : (w, R) \models O(Kd \wedge N\neg(d \wedge d')) \Rightarrow (w, R) \not\models KKq$ and $(w, R) \not\models K\neg Kq$ (Definition 2 and 1)
$\iff \forall(w, R) : (w, R) \models O(Kd \wedge N\neg(d \wedge d')) \Rightarrow \exists w'_1 \in R(w) : (w'_1, R) \not\models Kq$ and $\exists w'_2 \in R(w) : (w'_2, R) \models K$ (semantics of $K$ and $\neg$)
$\iff \forall w : \exists(w'_1, R_1) : (w'_1, R_1) \models Kd, (w'_1, R_1) \models N\neg(d \wedge d'), (w'_1, R_1) \not\models Kq$ and $\exists(w'_2, R_2) : (w'_2, R_2) \models Kd, (w'_2, R_2) \models N\neg(d \wedge d'), (w'_2, R_2) \models Kq$ (take $R_i(w) = \{w' | (w', R_i) \models Kd \wedge N\neg(d \wedge d')\}$)
$\iff \exists(w'_1, R_1) : R_1(w'_1) \subseteq W_{R_1}(d), R_1(w'_1) \supseteq W_{R_1}(d \wedge d'), R_1(w'_1) \not\subseteq W_{R_1}(q)$ and $\exists(w'_2, R_2) : R_2(w'_2) \subseteq W_{R_2}(d), R_2(w'_2) \supseteq W_{R_2}(d \wedge d'), R_2(w'_2) \subseteq W_{R_2}(q)$ (Lemma 1)
$\iff \forall R : W_R(d) \not\subseteq W_R(q)$ and $W_R(d) \cap W_R(q) \not\subseteq \emptyset$ and $W_R(d \wedge d') \subseteq W_R(q)$ (($\Rightarrow$) use seriality for $W_R(d) \cap W_R(q) \not\subseteq \emptyset$ and non-modality of $d, d', q$; ($\Leftarrow$) take $R_1(w'_1) = W_R(d)$ and $R_2(w'_2) = W_R(d) \cap W_R(q)$ which are non empty because of seriality)
$\iff d \not\models_{\mathcal{L}} q$ and $d \wedge q \not\models_{\mathcal{L}} \bot$ and $d \wedge d' \models_{\mathcal{L}} q$
$\iff [d, d'] \not\models_{D_{\mathcal{ONL}}} +q$ and $[d, d'] \not\models_{D_{\mathcal{ONL}}} -q$. ■

The exhaustiveness of Lemma 3 proves the completeness of descriptions in the form $[d, d']$ because a (non-modal) query is always either true, false, or possible. We see also that ? is disjoint from $+$ and $-$, but a query can be both true and false in the special case where $d \models_{\mathcal{L}} \bot$, i.e., the description is contradictory. In fact, in information systems, object descriptions must be checked against such contradiction. Finally, idiomatics presented in this section offers a simple way to reason in logic KD-$\mathcal{ONL}$ by relying only on a non-modal propositional prover.

## 5 Conclusion and Future Work

Section 4.2 presents a generalized form of logic $\mathcal{ONL}$ that is parallel to standard modal logics: a logic $\mathcal{ONL}$ is a modal logic extended with a new modal operator $N$ that enables to reason on unaccessible worlds, whereas the usual modal operator $K$ enables to reason on accessible worlds. Thus, as there are a whole family of modal logics depending on various conditions on the accessibility relation (AR), we get a whole family of $\mathcal{ONL}$-logics. Even if it is already known that logic $\mathcal{ONL}$ can be defined like a modal logic [Ros00], to our knowledge only K45-$\mathcal{ONL}$ (transitive and euclidean AR) has been studied. In this paper, we have studied K-$\mathcal{ONL}$ (any AR), then KD-$\mathcal{ONL}$ (serial AR), and we have showed they are more convenient for representing incomplete knowledge by enabling several levels of knowledge. Now, future works is to explore more deeply logic $\mathcal{ONL}$ both in its general and specific forms.

Recently, a tableau calculus has been proposed for logic K45-$\mathcal{ONL}$ [Ros00]. We think it would not be too difficult to extend it to any logic $\mathcal{ONL}$ by taking inspiration of what is done for modal logics with tableaux [Mas94].

In Section 1, we present the arbitrarity of the logic used as an important feature of our Logical Information Systems. A problem is that logic $\mathcal{ONL}$ sets the logic as soon as we want to represent knowledge. To combine $\mathcal{ONL}$ features with genericity, our idea is to build an abstraction of logic $\mathcal{ONL}$ by making the logic $\langle \mathcal{L}; \models_{\mathcal{L}} \rangle$ appearing in Lemmas 2 and 3 a logical parameter. We call such an abstraction a *logic functor*, and we have already done this work for several logics such as the propositional logic that we have abstracted over atoms. An $\mathcal{ONL}$ logic functor would allow to represent complete and incomplete knowledge as presented in this paper, but with non-modal part of descriptions and queries expressed in a dedicated logic (e.g., sets of valued attributes for the bibliographical application).

Finally, we intend to study how expressing integrity constraints in logic $\mathcal{ONL}$ itself [Rei92], and to design declarative revisions and updates in order to integrate in the description of an object new knowledge facts about it, while preserving its consistency, and without having to edit it by hand.

## Acknowledgements

A warm thank goes to Olivier Ridoux for insightful discussions and advices. His expertise in logic has been helpful to the highest point. I am also thankful to

Philippe Besnard who make me know about logic All I Know and has supported my work.

## References

[Bow79] K. A. Bowen. *Model Theory for Modal Logic.* D. Reidel, London, 1979.

[Che94] Chen. The logic of only knowing as a unified framework for non-monotonic reasoning. *FUNDINF: Fundamenta Informatica*, 21, 1994.

[Di 95] R. Di Cosmo. *Isomorphisms of Types: from λ-calculus to information retrieval and language design.* Progress in theoretical computer science. Birkhäuser, 1995.

[DJ94] P. T. Devanbu and M. A. Jones. The use of description logics in KBSE systems. In Bruno Fadini, editor, *Proceedings of the 16th International Conference on Software Engineering*, pages 23–38, Sorrento, Italy, May 1994. IEEE Computer Society Press.

[FR00a] Sébastien Ferré and Olivier Ridoux. A file system based on concept analysis. In Yehoshua Sagiv, editor, *International Conference on Rules and Objects in Databases*, LNCS 1861, pages 1033–1047. Springer, 2000.

[FR00b] Sébastien Ferré and Olivier Ridoux. A logical generalization of formal concept analysis. In Guy Mineau and Bernhard Ganter, editors, *International Conference on Conceptual Structures*, LNCS 1867, pages 371–384. Springer, 2000.

[GJSO91] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. Jr O'Toole. Semantic file systems. In *Proceedings of 13th ACM Symposium on Operating Systems Principles*, pages 16–25. ACM SIGOPS, October 1991.

[GMA93] R. Godin, R. Missaoui, and A. April. Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38(5):747–767, 1993.

[Lev90] Hector Levesque. All I know: a study in autoepistemic logic. *AI, -3, ; ACM CR 9102-0987*, 42(2), March 1990.

[Lif91] V. Lifshitz. Nonmonotonic databases and epistemic queries. In *12th International Joint Conference on Artificial Intelligence*, pages 381–386, 1991.

[Mas94] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In Alan Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, LNAI 814, pages 723–737, Berlin, 1994. Springer.

[McC86] John McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28(1), 1986.

[Moo85] Robert C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[Rei80] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.

[Rei92] Raymond Reiter. What should a database know? *Journal of Logic Programming*, 14(1-2):127–153, October 1992.

[Ros00] Riccardo Rosati. Tableau calculus for only knowing and knowing at most. In Roy Dickhoff, editor, *TABLEAUX*, LNCS 1847, pages 383–397. Springer, 2000.

[Wil82] Rudolf Wille. *Ordered Sets*, chapter Restructuring lattice theory: an approach based on hierarchies of concepts, pages 445–470. Reidel, Dordrecht Boston, 1982.