# Extending a Logic-based Model with Algebraic Knowledge

Jean-Pierre Chevallet, Yves Chiaramella

# Extending a Logic-Based Retrieval Model With Algebraic Knowledge

Jean-Pierre Chevallet
Institut d'Informatique et de Mathematique Appliquee de Grenoble (IMAG)
France
Jean-Pierre.Chevallet@imag.fr

Yves Chiaramella
Institut d'Informatique et de Mathematique Appliquee de Grenoble (IMAG)
France
Yves.Chiaramella@imag.fr

**Abstract**

A major need of precision-oriented retrieval systems is the availability of high-level indexing languages which can allow the representation and the manipulation of elaborated concepts for both indexing and querying processes. Such elaborated knowledge representations, as the one developed in the RIME project, need in turn to be fully integrated within the underlying retrieval model to ensure proper and optimal use of this knowledge in the retrieval process. In this paper we present further developments of the fuzzy modal logic retrieval model which was first developed and experienced in the RIME project. These developments are precisely aimed towards a better integration of this model and knowledge representation. We show that Conceptual Graphs of Sowa have formal properties which allow a good formal control of this integration, and are inherently well adapted to IR requirements. This study has triggered extensions of the initial logic model and adaptations of the conceptual graphs' formalism which are also presented here. Finally the paper presents the way every notion of the theoretical retrieval model is expressed in terms of a derived operational model which has been implemented.

## 1 Introduction

A major need of precision-oriented retrieval systems is the availability of high-level indexing languages which allow the representation and the manipulation of elaborated concepts for both indexing languages and querying processes. Here the term "precision-oriented systems" has a more specific meaning than currently understood in IR: it refers to the classical meaning of high precision performances and to the ability to deal with "precise" - i.e. elaborated - concepts. These two properties are mandatory for numerous application fields of IR where users are skilled specialists working in specific areas. Good examples of such applications fields are experienced within the RIME [2] and ELEN [6] projects developed in the CLIPS-IMAG laboratory: namely Software Reuse for ELEN and Medicine for RIME. In such environments, specialists are either engineers or physicians, corpuses are software component databases or medical reports and images. What characterizes these users is their information needs and their requirements about information retrieval: being domain specialists they use high level, precise, intricate notions in their current activities, and they need to communicate with a retrieval system using this level of domain concepts. A consequence is of course that they require from the retrieval system to process their queries at this level of conceptual expression: they often cannot support for a long time poor precision performances due to inadequate indexing languages.

The RIME and ELEN projects constitute the framework we have designed for investigating precision-oriented information retrieval systems: RIME is more specifically dedicated to multimedia information retrieval in medical information. The project deals with corpuses combining textual information from medical reports and related X-ray pictures. ELEN is a more recent project dedicated to complex object information retrieval whose application domain is Software Reuse. Experimentations of this prototype are on their way using corpuses of UNIX and LOOPS software libraries.

These two projects are based on the same paradigm about the underlying models: the *fuzzy modal logic model*. This model was designed and implemented for the first time in the RIME project [19]. In the next section we discuss

some limitations we found while implementing the first version of the model and which are mainly related to what we call a "loose integration" of the knowledge representation in the retrieval model. This means that the knowledge representation suffered a lack of explicit formal properties allowing to prove that the matching process of queries and documents implemented in this earlier version was completely defined. As we shall see it in the next section, this matching process between complex (tree-like) semantic structures was based on a set of preselected cases instead of being based on a more general (i.e. formal) definition of this match. A result of that situation is that we could not be sure that all relevant matching cases between document semantic structures and query semantic structures were implemented, and thus that the matching process was possibly missing relevant cases.

This consideration triggered us to find a more suitable knowledge representation model. In section 3 we explain why Sowa's Conceptual Graphs fit perfectly with these requirements and, more generally, why they are inherently more suitable to any logic-based retrieval model. As we shall see, this is mainly due to their formal properties (a correspondence with first order predicates), and to the existence of primitive operators which help in the control of their construction and their use in the matching process. In this section we also present some adaptations we have introduced to improve the adaptation of Conceptual Graphs to IR requirements.

In section 4 we show how the original fuzzy modal logic model was then extended to integrate these formal properties of Conceptual Graphs. In section 5 we present the main retrieval operators which have been derived from this improved theoretical model. All these aspects have been implemented and experimented within the ELEN project. Finally, in section 6 we present some guidelines for the valuation of the partial implication. We have chosen to emphasize here the modeling aspects of the project (both theoretical and operational); though already advanced implementation aspects will be detailed in further publications.

It should be clear to the reader that both RIME and ELEN research projects are dedicated to explore logic based approach to multimedia IR. Our main concern at this step is to identify and experiment useful features [1] of such models, the design of a complete formal logic being a next step that can be worthy achieved only when these preliminary steps are completed. The reader more familiar with mathematical logic, should not then be surprised to find below models that are still incomplete, considering habitual standards in formal logics.

# 2 The RIME experience

The underlying model of RIME is based on the paradigm introduced in [21] by Van Rijsbergen [25]. According to this model, the certainty for a document $D$ to be relevant to a query $Q$ is related to the certainty of the logical implication $D \rightarrow Q$. This implication is evaluated according to a given logic which of course encompasses the notions of documents' and queries' semantic content. This general definition is expressed by $P(D \rightarrow Q)$, where $P$ [2] is a function which estimates the certainty of the implication. This approach fits our needs because it makes explicit, coherent use of knowledge as a basis of the matching process between queries and documents.

In [25] and [19], we can find demonstrations that all previous retrieval model can be expressed in terms of the logical model. We think that aside from this property of generality, the main interest of the logic-based approach is that it is the only one which can provide a formal framework for designing "intelligent" matching processes involving knowledge and deduction mechanisms. All models used in IR are based on the representation of a set of "concepts" representing the semantic content of a document. Only few of them take into account the relations existing between these concepts. Explicit use of these relations is an efficient way of reducing the ambiguity of indexes and query concepts, improves the expression of "precise" concepts, and thus enables the system to fulfill the precision-oriented retrieval requirements introduced before.

The first formal instantiation of this logical model has been proposed in the RIME project. A more general discussion about logic and retrieval model can be found in [27, 17]. In this project, we designed a theoretical model based on fuzzy modal logic from which we derived an operational model which has been implemented. This operational model was entirely inspired from the Artificial Intelligence area (Prolog and Conceptual Dependency).

## 2.1 Principles of the Fuzzy Modal Logic Retrieval Model

The general principle proposed by VanRijsbergen was extended by Nie to the consideration of both the direct ($D \rightarrow Q$) and reverse implications ($Q \rightarrow D$) between documents and queries. Then the relevance measure $R_K$ between

[1] properties needed that can help a better understanding and lead to effective implementation of multimedia IR.

[2] In Rijsbergen paper [25] $P$ was a probability, but the proposed model does not use any property of such a function: for us it's only an uncertainty value.

document $D$ and query $Q$ was extended to the following definition:

$$R_K(D,Q) = F[P_K(D \rightarrow Q), P_K(Q \rightarrow D)]$$

where $P_K$ expresses that the evaluation of $P$ is related to the use of a knowledge base $K$. Function $P$ provides a continuous evaluation of the implications' certainty, while function $F$ combines these two certainties in a final evaluation of $R_K(D,Q)$ that expresses system relevance.

We must note that the symbol $\rightarrow$ cannot be understood directly in a formal logical sense but rather in an more intuitive way. In [20, pp 478] Nie says that a document is considered to be a set of sentences that are interpreted into a predefined semantic. He adds that a query is usually a single sentence and for a document $D$ to be a relevant answer for a query $Q$, it must "imply" the query. We think that this intuitive notion of implication is a good way to analyze the relationship between users' needs and documents, though it covers only a part of the complex problem of IR. So we agree with Nie when he says in [20, pp 478] that in information retrieval, this implication is always "plausible" rather than "strict". A measure of implication strength (or certainty measure) has then to be associated with it, and the key point of this proposal is now to find a formal approximation of this symbol using a formalized logic.

From the theoretical point of view, there are three ways for evaluating $P_K(X \rightarrow Y)$, called evaluation principles (see [20, pp486]). These principles are based on the certainties related to the modification of either the knowledge base $K$ used for proving the implication, the premise $X$, or the conclusion $Y$ of the implication.

The second principle of evaluation proposes a convenient way for computing this measure in practice. According to it, the measure of $P_K(X \rightarrow Y)$ is related to the certainty of the semantic transformations applied to $X$, needed to prove $X \rightarrow Y$. The theoretical model proposed by RIME bases this demonstration on modal logic where worlds (see below) represent the various possible (and possibly uncertain) deduction steps of this proof.

We propose the following meaning for the information retrieval implication $\rightarrow$. We propose to declare $X \rightarrow Y$ certain [3] iff $X \models Y$. In logic the symbol $\models$ is used when one deals with the *semantic* of formula: logic languages are formally described by a syntax and by a semantic that usually consist in translations of formulas into an other *mathematical world* where the notion of truth exists either under the form of values (finite or not) or under the form of sets. For example, in classical propositional logic, the semantic of a formula (also called *interpretation* or *model* in for example [8, 9]) are expressed using a function from the variable set of the language to a binary set $\{V, F\}$ (see [8, pp 29]). The semantic of a formula is then an element of $\{V, F\}$ computed via combination functions associated to logical operators. The assertion $M \models A$, often read as "the interpretation of $M$ validates the formula $A$", means in classical logic that $M$ is an interpretation [4] that makes $A$ true. In classical logic, one can also use this symbol $\models$ between a set of formulas $\Gamma$ and a formula $A$. The meaning of $\Gamma \models A$, read as "$\Gamma$ validate $A$" or "$A$ is a semantic consequence of $\Gamma$", is that any interpretation validating formulas of $\Gamma$, validates the formula $A$. So, when we write $A \models B$ for two formulas $A$ and $B$, this also means $\{A\} \models B$ (see [8, pp 31]). Consequently, without any formal mistake, in $A \models B$, we can either view $A$ as a formula or as an interpretation that models the formula $A$.

In modal logic, formulas are evaluated in relation to a set of interpretations called *worlds*. Therefore, $X \rightarrow Y$ is certain iff $\models^S_X Y$, which means that $Y$ is true in the world $X$ for a Kripke structure $S$. This structure $S$ includes the set of all possible worlds and the way they are linked: from a world one may go to several other possible worlds (but usually not all). Transitions between possible worlds are structured in a graph, according to a relation $\delta$ which expresses truth values for allowed transitions. When applying the second evaluation principle to IR, and if $X$ stands for a document $D$ and $Y$ stands for a query $Q$, worlds correspond to semantic interpretations of the document considered as a logical proposition, $Q$ is a formula whose truth has to be proved in a given world (an interpretation of the document's content).

A so-called modal operator is introduced to express the fact that a formula may be true depending on the existence of a possible world where $Y$ is true. This property is expressed by the modal operator $\diamond$ called *possibility*. Conversely, when a formula is true in all possible world, the formula is said to be certain which is expressed by the modal operator $\Box$ called *necessity*.

According to these definitions, the fact that $P_K(X \rightarrow Y)$ is uncertain (i.e. $P_K(X \rightarrow Y) \in ]0;1[$ ) is equivalent to $\models^S_X \diamond Y$ and not $\models^S_X Y$. Finally, the fact that $P_K(X \rightarrow Y)$ is false, or $P_K(X \rightarrow Y) = 0$, is equivalent [5] to $\models^S_X \Box\neg Y$. System knowledge $K$ are represented by the way the set of world are structured. So the parameter $K$ appears implicitly in the Kripke structure $S$ and the result is a model based on *truth values* rather than on *validity*.

[3] The term *certain* is rarely used in logic but as the symbol $\rightarrow$ does not yet belong to a formal language of a formalized logic, we cannot explain yet what we mean using this symbol in terms of formal properties such as logical validity, satisfaisability, etc.

[4] We prefer to use the word "model" for interpretations that make true a given formula. Then we say that this interpretation "models" this formula, or that it is a model for this formula.

[5] Because operator $\Box$ is equal to $\neg \diamond \neg$.

This application of modal logic to IR models was then extended to fuzzy modal logic by Nie mainly to obtain continuous evaluations of the certainty function $P$, instead of the binary truth values of standard modal logic. Details may be found about these aspects in [19].

## 2.2 Knowledge representation in RIME

Medical knowledge extracted from medical reports was first represented using a formalism derived from the notion of Conceptual Dependency [23]. This knowledge representation is mainly based on tree structures whose terminal nodes are atomic domain concepts, while non terminal nodes are semantic operators. Domain concepts represent the lowest level of semantic representation for documents and queries, and semantic operators allow the expression of more elaborated, structured concepts by combining either domain or already existing structured concepts.

This principle is applied recursively to define complex semantic structures which express accurately the semantic content of full sentences of medical reports or natural language queries. The indexing language based on this principle is completely defined by a grammar which is used to control the construction of allowed (i.e. meaningful, relevant) concepts. Each non-terminal of this grammar corresponds to a semantic class of the corresponding concept. The grammar thus fixes both the language structure and its semantics. For more details about this model and the corresponding indexing process see [2]. As an example the following concept is derived from the natural language sentence: "opacity of the lung'':

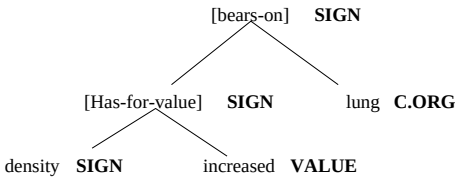


Figure 1: "opacity of the lung"

Domain atomic concepts here are "density", "increased", "lung", while symbols noted between square brackets are semantic operators. Capital bold symbols refer to semantic classes of the attached concept (a tree). The operator [has-for-value] denotes a valuation relationship between "density" and its observed value "increased" (i.e. abnormal), and the corresponding subtree is a concept of the semantic class SIGN. As an example of detailed expression of concepts (explicitly required by the radiologists) mentioned before, one may note that the original notion of "opacity" (of an area of the X-ray picture) in the sentence is expressed by the non-atomic concept of "increased density" (of the grays in an area of the picture).

## 2.3 The Matching Process in RIME

As a direct application of the second evaluation principle presented above, the matching process implemented in the operational model of RIME is based on a set of rules which are used to prove that a semantic expression $D$ representing the semantic content of a document, implies another semantic expression $Q$ representing the semantic content of the query. These semantic expressions could not be understood as formulas of a logic because only the expression language and some derivation rules were described in RIME: there is no deduction nor truth values. Moreover, this model was a first practical attempt to use a complex index language and a rule-based retrieval. This first pragmatic approach moreover has led us forward to the use of formal logics.

Very briefly these rules are expressed as follows, where $A$ and $B$ are semantic [6] expressions belonging to the index language, $a$ and $b$ are domain concepts, $[rel]$ is a semantic operator, and $\rightarrow_c$ expresses the certainty $c \in [0,1]$ of an implication:

- if $a \rightarrow_c b \in K$ and $a = A$ and $b = B$, then $A \rightarrow_c B$;
- if $A \subseteq B$, then $A \rightarrow_1 B$, where $\subseteq$ represents the inclusion of trees;
- if $A \rightarrow_c A'$, $A$ and $A'$ belong to the same semantic class then: $[rel](A, B) \rightarrow_c [rel](A', B)$;

[6] See [2] for more about the syntax of these semantic expressions. There is no formal semantic associated because only common sense and medical expert had been used to built this language.

- If $C = [rel_1](A, B)$ and $A = [rel_2](E, F)$ and $(A, C, E)$ belong to the same semantic class, then $[rel_1](A, B) \rightarrow_c [rel_1](E, B)$.

All these rules are correct in the sense that they do correspond to real cases where the implication may formally be demonstrated, with a certainty $0 < c \leq 1$. The problem here is that though being correct, this rule set is defined in extension. Then how to be sure that this extension is complete according to the general problem of proving that $A \rightarrow_c B$ for any configuration of $A$ and $B$ ?

This constitutes a considerable limitation of the operational model: given a query $Q$ we cannot then be certain that every document which logically implies $Q$ will be retrieved. Only those satisfying the above set of rules will be retrieved. The problem here is clearly a lack of formalization : the derivation scheme of RIME does not refer to any logical nor algebraic formal model. Said in other words, this means that what we need as part of the retrieval model is a knowledge model whose formal properties ensure a definition in intention of all these cases. If this can be done, the matching process between $D$ and $Q$ will make full use of these properties and we shall be certain that no relevant case will be omitted when designing the matching function within the operational model. Demonstrating that Conceptual Graphs fulfill these requirements (and of course the one related to adequate expressive power of complex concepts) is the main goal of the next section.

# 3 Guidelines of a general graph model of index

Logic modeling and semantic graphs are well known tools used for knowledge representation and we shall investigate later the relationship between these two points of view. Let's focus for now on graph-oriented knowledge representations we use to express concepts, and the kind of knowledge that is necessary to build them. The requirements we decided to consider about knowledge representation are the following:

- description of complex interrelated concepts organized in a taxonomy.
- control of the consistency of what can be expressed.
- consideration of IR specific requirements (uncertain knowledge).
- coupling with a theoretical model to ensure controlled properties of derived operational models (see the discussion in the previous section).
- efficiency consideration concerning derived matching and searching processes.

In this section we discuss why in our opinion the Conceptual Graphs fulfill some of these requirements, and we propose adaptations needed to satisfy those which are not fulfilled by this formalism in its original definition.

## 3.1 Index and knowledge model

The point we are starting from is a knowledge representation named by Sowa [24] *Conceptual Graphs*. We have chosen this representation for the following reasons:

- the knowledge structure representation is a graph of concepts and semantic relations. Taxonomies for concept and relation types are available.
- the use of algebraic construction operators enable some control on generated conceptual structures. These operators, in conjunction with a small set of graphs (called canonical basis), allow the control of all that can be expressed to just what makes sense [7].
- the existence of a formal relationship with logic (see below the $\Phi$ function) eases the coupling with the logical IR model.
- the partial order on conceptual graphs leads to a projection operator. The use of this operator ensures search effectiveness.

[7] In [24, pp90-91] Sowa writes "A conceptual graph is a combination of concept nodes and relation nodes [...]. But not all such combination make sense [...]. To distinguish the meaningful graphs that represent real or possible situations in the external world, certain graphs are declared to be *canonical*".

We shall develop all these points later, after a presentation of the basic definitions and properties of Conceptual Graphs, as they are given by Sowa [24]. Then we shall focus on some useful features we have added to make this formalism more suitable to the previous requirements.

## 3.2 Building semantic structures with algebraic operators: the Conceptual Graphs

### *3.2.1 Definitions*

This model for knowledge representation is based on two general principles: the knowledge base includes the notion of concept taxonomy and may be derived from a set of canonical graphs and algebraic operators. Formally, a conceptual graph is a bipartite graph of *concepts* and *conceptual relations*. A concept has a type (which corresponds to a semantic class) and a referent (which corresponds to an instantiation of the class). A conceptual relation has only a type.

Both relation types and concept types are organized in a taxonomy (a lattice). Referents are items which have to conform to a related concept type. This conformity relation is transitive among the taxonomy lattice: for example, $[HUMAN : *]$ stands for the concept of all possible human beings. This concept is called a *generic* concept also noted $[HUMAN]$. $[HUMAN : \#]$ stands for a given human being, and $[HUMAN : \#John]$ stands obviously for the concept of a human named John. These referents, different from $*$, are called *individual markers*.

**Definition 1 (Type denotation)** *A denotation of a type is the set of all possible individual markers that conform to a concept type - i.e., that can be the referent of a concept of that type. One note by* :: *the conformity relation between a type and an individual marker.*

The syntax of conceptual graphs used in this paper is defined as follow using a BNF grammar. It is a small subset of the one defined in [10]. For simplification we shall restrain ourself to a grammar describing the most basic features of CG. Note that the bold brackets **[ ]** denote options in the following rules:

| | | |
|---|---|---|
| cgraph | ::= | c-node \| relation arc c-node. |
| c-node | ::= | concept **[**rlink**]**. |
| rlink | ::= | arc r-node \| arc r-node "," rlink. |
| r-node | ::= | relation **[**arc c-node**]**. |
| concept | ::= | "[" name **[**reffield**]** "]". |
| relation | ::= | "(" name ")". |
| reffield | ::= | "#" **[**name**]** \| "*" **[**name**]**. |
| arc | ::= | "←" \| "→". |

The non terminal "name" is any sequence of digits or characters.

**Definition 2 (Referent)** *Referents using"*" are called generic and those using "#" are called specific or individual markers. Referents with a name (both generic and specific) are called named referents and referents without name (both generic "*" and specific "#") are called anonymous referents.*

All concepts and relations are *different* from each other except for concepts that have the same type and the same named referent. For example, the graph $[HUMAN : \#] \leftarrow (r) \leftarrow [HUMAN : \#]$ contains one relation and two concepts, but the graph $[HUMAN : \#john] \leftarrow (r) \leftarrow [HUMAN : \#john]$ contains one concept and a reflexive relation. Named referents like $\#name$ or $*name$ are cross references used to describe a graph in a linear way. The concept $[BOY : \#John]$ is *well formed* (see below and also 5.1) if the concept $BOY$ is a sub-type of the concept $HUMAN$ and if $John$ is a possible referent for $BOY$. The notion of well formed concept can be defined as:

**Definition 3 (Well formed concept)** *A concept is well formed if the referent is generic (noted *), specific (noted #) or if the named individual marker satisfies the conformity relation with the concept type. We say then that this referent belongs to the denotation of the concept type.*

The taxonomy of conceptual types can also be defined in terms of type denotations.

**Definition 4 (Type taxonomy)** *The taxonomy of conceptual types is built from the set inclusion of type denotations.*

To make clear the linear syntax used to describe graphs in this paper, we precise the notion of concept equality:

**Definition 5 (Concept equality)** *Two concepts are equal (are the same) iff they have the same type and the same named referent.*

With this definition, every generic concepts without a named referent are different, and graphs like:
$[A] \leftarrow (r) \leftarrow [A]$
$[A : \#] \leftarrow (r) \leftarrow [A : \#]$
$[A : *] \leftarrow (r) \leftarrow [A : \#]$
$[A : *x] \leftarrow (r) \leftarrow [A : \#x]$
count two concepts and one conceptual relation, but graphs like
$[A : *x] \leftarrow (r) \leftarrow [A : *x]$
$[A : \#x] \leftarrow (r) \leftarrow [A : \#x]$
are reflexive graphs made from only one concept and one conceptual relation. There are some cases left to be discussed concerning named referents and types. Let's consider the following graphs:
$[A : *x] \leftarrow (r) \leftarrow [B : *x]$
$[A : \#x] \leftarrow (r) \leftarrow [B : \#x]$
$[A] \leftarrow (r) \rightarrow [B]$

The informal meaning for the first graph is "there is at least one $A$ connected to at least one $B$ which are the same", and for the second, we would like it to mean that "$x$ is an $A$ connected to the same $x$ that is a $B$". By "$x$ is an $A$" we mean that $x$ belongs to the denotation of $A$. Consequently the denotations of $A$ and $B$ are two sets whose intersection must not be empty to ensure that we can associate some meaning to these graphs. An informal meaning of the last graph can be that "there is a relation out going to $A$ and $B$". Although this graph can be obtained using the grammar, we are not convince these graphs could have an interressing use in IR model: in our experiments off manual indexing, only graphs with conceptual relation in the same direction seems useful. Moreover, the semantic of this third graph does not seem obvious. We then propose the following definition of "well formed graph", which will be used throughout this paper as the formal definition of Conceptual Graphs.

**Definition 6 (Well formed conceptual graph)** *A conceptual graph is well formed iff all concepts are well formed and if two concepts of the graph with the same named referent (generic or specific) have the same type and if every conceptual relation has exactly one incoming link and one outgoing link.*

Using this restriction, the tree previous graphs are not well formed. From now, we will only consider with well formed graphs.

Four elementary operators are used to build graphs from existing ones, a given taxonomy of concepts and relations. For the definition of the join operator presented after, we must first precise the notion of common concepts in one graph or between two graphs.

**Definition 7 (Common concept)** *Two concepts of one or two graphs are common if they share the same type and referent (both generic or both specific).*

Obviously, two equal concepts are common concepts but the reverse is false. The four operators apply only on well formed graphs and produce only well formed graphs:

- **Copy**: if $w$ is a conceptual graph then the copy $u$ of $w$ is a conceptual graph with is a duplicate of $w$. We must notice that without a definition of equality on graphs, we cannot decide if the copy operator produces a new graph equal to the original one. We shall then propose later a definition of equality on graphs to solve this problem.
- **Restriction**: A graph is restricted when a concept type or a relation type is replaced by a subtype, or when the referent of a generic concept is replaced by an individual marker.
- **Simplification**: when two concepts are linked by two identical relations, then one may be deleted. The simplification of $[A : *x] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [A : *x]$ is $[A : *x] \rightarrow (r) \rightarrow [B]$.
- **Join**: two graphs having one common concept can be joined to form one graph by sharing this common concept. The graph $[A] \rightarrow (r) \rightarrow [B]$ can be joined to the graph $[C] \rightarrow (r') \rightarrow [B]$ on their common concept $[B]$. The result is the graph $[A] \rightarrow (r) \rightarrow [B] \leftarrow (r') \leftarrow [C]$ . The join operator is not an operator in the strict mathematical sense since there can exist different ways of joining two graphs. For example, the graph $[A] \rightarrow (r) \rightarrow [A]$ can be joined to the graph $[A] \rightarrow (r') \rightarrow [B]$ in two manners and produces either the graph : $[A] \rightarrow (r) \rightarrow [A] \rightarrow (r') \rightarrow [B]$ or the graph : $[B] \leftarrow (r') \leftarrow [A] \rightarrow (r) \rightarrow [A]$.

These operators can be used to define a construction relation between graphs:

**Definition 8 (Graph order)** *Two graphs $G$ and $G'$ are said to be in relation order if $G$ is derived from $G'$ by using at least one of the four algebraic operators* [8]*. Under certain conditions, this relation is a partial order which is noted $G \leq G'$.*

Given a set of concept type and relation type, we can build an infinite set of well formed graph. Sowa propose to define a reduced set of graph using the definition of *canonical conceptual graphs*. In [24, pp 91] we can read *"To distinguish the meaningful graphs that represent real or possible situations in the external world, certain graphs are declared to be canonical."*

**Definition 9 (Canonical basis and canonical graphs)** *A canonical basis is a finite sub set of well formed graphs. A canonical graph is either member of the canonical basis or obtained from canonical graphs using the four operators.*

### *3.2.2 Informal semantic of operators*

The initial CG formalism was presented in [24] and as Wermelinger says in the abstract of [26], *"Conceptual Structures (CS) Theory is a logic-based knowledge representation formalism. To show that conceptual graphs have the power of first-order logic, it is necessary to have a mapping between both formalisms. A proof system, i.e. axioms and inference rules, for conceptual graphs is also useful. It must be sound (no false statement is derived from a true one) and complete (all possible tautologies can be derived from the axioms). [..] Sowa's original definition of the mapping is incomplete, incorrect, inconsistent and unintuitive, and the proof system is incomplete too"*. Unfortunately, there exists by now no complete formal semantics for this formalism. Our attempt here is to propose an informal and intuitive meaning for the CG formalism used here after.

**Axiom 1 (Informal meaning of operators)** *The four graph operators (copy, restriction, simplification and join) reduce the "possible meaning" of an obtained graph because they are specialization operators.*

If we accept this axiom, one can understand axiom 2 since the relation $\leq$ on graphs is defined using the four operators.

**Axiom 2 (Informal meaning of graph order)** *The graph relation $\leq$ expresses a specialization relation.*

The relationship $\leq$ is of extreme importance for the IR area: $G \leq G'$ expresses the fact that one graph $G$ is derived from another graph $G'$. This means in turn that $G$ includes the *meaning* contained in $G'$ because $G'$ is more "general" than $G$. We can also understand the previous axiom noticing that:

- if G has been derived from G' by the join operator then G' is clearly included in G.
- if G has been derived from G' using the restriction operator then G contains at least one more specific concept than its corresponding in G'.
- the two other operators correspond to trivial cases considering this inclusion.

We show in the next section that one can precise the meaning of this order when using the correspondence between graphs and first order logic. This correspondence is partial and possible only for the restricted formalism we use in this paper.

### *3.2.3 Partial order and join operator*

The join operator can be defined in several different ways: using these different join operators, we can obtain different conceptual graphs with particular properties. Before that, we must define an equivalence relation on graphs stating be able under what conditions two graphs are equal. We first define the notion of sub-graph.

**Definition 10 (Sub-graph)** *A graph $G_1$ is a sub graph of $G_2$ if there exists a mapping $\Pi$ from the concepts and conceptual relations of $G_1$ into the concepts and conceptual relations of $G_2$ such that : for each concept $c$ of $G_1$, $\Pi(c)$ is a concept of $G_2$ with the same type and the same referent, and for each conceptual relation $r$ of $G_1$, $\Pi(r)$ is a conceptual relation of $G_2$ with the same type. Moreover, if $c$ and $r$ are linked in $G_1$, then $\Pi(c)$ and $\Pi(r)$ must be linked in $G_2$ in the same way (from $c$ to $r$, from $r$ to $c$ or both links on $c$ and $r$ in case or reflexive relation).*

[8] The join operator is a two place combination: G is derived from G' by joining G' to an other graph G".

**Definition 11 (Graph Equality)** *Two graphs $G_1$ and $G_2$ are equal if $G_1$ is a sub-graph of $G_2$, and $G_2$ is a subgraph of $G_1$.*

This equality on graphs is a topological (or syntaxical) one in the sense that it only deals with nodes and the way nodes (i.e. concepts and conceptual relations) are linked together to form the graph. For example, suppose we have the two graphs: $[A] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [C]$ and $[C] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [A]$. These two graphs are equal. With the join operator (defined previously) on the common concept $[A]$ we can obtain the graph:

$$[C] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [A] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [C]$$

which means that we consider the two concepts $[A]$ as the same but we don't know about the identity of concepts $[B]$.

We may also obtain the graph: $[B] \leftarrow (r) \leftarrow [C]$,
$\leftarrow (r) \leftarrow [A]$,
$\leftarrow (r) \leftarrow [C]$,
$\leftarrow (r) \leftarrow [A]$.

or the graph: $[A] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [C] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [A]$

but not the graph : $[A] \rightarrow (r) \rightarrow [B] \leftarrow (r) \leftarrow [C]$,
$\leftarrow (r) \leftarrow [C]$.

whenever both $[A]$ and $[B]$ are considered as the same on both graphs. If we want to build this last graph, one have to consider a join on two common concepts in the same graph that we call *internal join*. The initially defined join on two different graphs is now called *external join*. It may be shown that using only the external join and the other operators ensures that the corresponding construction relation is a partial order. The main drawbacks of using only external is expression limitation: as the external join can be done only on on two different graphs, one cannot reduce two concept that turn to be the same. For example consider the following graph having one relation and two concepts:

$$[A] \leftarrow (r) \leftarrow [A]$$

Using a restriction on one concept we obtain:

$$[A : \#x] \leftarrow (r) \leftarrow [A]$$

When we restrict the other concept with the same referent we obtain a reflexive graph with only one concept and one relation:

$$[A : \#x] \leftarrow (r) \leftarrow [A : \#x]$$

We can obtain the same result considering first an internal join on the two generic concepts $[A]$ and second a restriction of the concept. This example shows that the restriction operator can reduce the number of concepts in a graph. Thus this can be avoided by explicitly performing an internal join. This internal join enables the production of new smaller graphs. As a consequence, two graphs that are different at the topological point of view defined previously, will be considered as identical for the partial order. This results has also being obtained by Chein and Mugnier in [3, p 383].

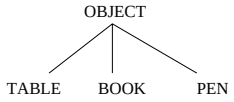


Figure 2: "A simple taxonomy"

For example, let us consider the concept taxonomy of figure 2 and the graph:

$$(1)[OBJECT] \rightarrow (LiesOn) \rightarrow [TABLE]$$

Using the restriction on $OBJECT$, one produce:

$(2)[BOOK] \rightarrow (LiesOn) \rightarrow [TABLE]$

Now an external join on graphs (1) and (2) produces:

$(3)[OBJECT] \rightarrow (LiesOn) \rightarrow [TABLE] \leftarrow (LiesOn) \leftarrow [BOOK]$

One can restrict again the type $OBJECT$ and obtain the graph:

$(4)[BOOK] \rightarrow (LiesOn) \rightarrow [TABLE] \leftarrow (LiesOn) \leftarrow [BOOK]$

Now using the internal join on concepts BOOK and the simplification operator, we obtain the graph:

$(5)[BOOK] \rightarrow (LiesOn) \rightarrow [TABLE]$

The graphs (2) and (5) are equal. If we keep to the original definition of the partial order, then we must consider the paradox $1 \geq 2 \geq 3 \geq 4 \geq 2$. Thus, graphs (2), (3) and (4) must be considered as equal from the point of view of the partial order ! Going back to more practical issues of IR, this means that, only one of these graphs may be used in an index. In fact, one can feel that these graphs are different from an *intentional* point of view.

### 3.3 Association of a first order formula to a conceptual graph

A most important statement of Sowa about conceptual graphs is that they can be mapped to first order logical formulas through a function noted $\Phi$. This mapping is not complete and an other mapping is proposed by Wermelinger in [26]. As we use a restricted formalization of the original model, this mapping is easier to define: we propose a mapping only for well formed graphs.

1. Each concept is associated to a predicate of arity one whose name is equal to the concept type. The predicate variable is either a constant value equal to the concept referent, or a quantified independent variable. For example, the concept $[T : \#r]$ is associated to the predicate $T(r)$ while the generic concept $[T : *]$ is associated to the logic formula $\exists x T(x)$.
2. Each relation is associated to a two place predicate whose name is equal to the relation type name. Predicate's parameters are those used in the predicates which correspond to the linked concepts.
3. For a conceptual graph $u$, $\Phi(u)$ is the first order logical formula obtained from the conjunction of predicates of types 1) and 2) associated to all components of $u$.

More formally we propose the following transformation for a subset of graphs that have only binary conceptual relations:

- $\Phi([T]) = \Phi([T : *]) = \exists x T(x)$ where x is a new variable.
- $\Phi([T : *x]) = \exists x T(x)$ where x is a variable.
- $\Phi([T : \#]) = T(a)$ where a is a new logical constant.
- $\Phi([T : \#idf]) = T(idf)$ where idf is a logical constant.
- $\Phi(C_1 \leftarrow (r) \leftarrow C_2) = \Phi(C_1) \wedge r(x_1, x_2) \wedge \Phi(C_2)$ where $x_1$ is a variable or a constant of the $\Phi$ transformation of concept $C_1$ (idem for $x_2$ and $C_2$).
- $\Phi(C_1 \rightarrow (r) \rightarrow C_2) = \Phi(C_1) \wedge r(x_2, x_1) \wedge \Phi(C_2)$ where $x_1$ is a variable or a constant of the $\Phi$ transformation of concept $C_1$ (idem for $x_2$ and $C_2$).

The transformation $\Phi$ for a graph is a formula obtained with a conjunction of the previous transformations. We can transform the obtained formula by moving quantifiers at the beginning because all variables are bound with $\exists$ without any ambiguity, and because we only use conjunctions: we do not use negations nor implications (see for example [1, p134]). For example, the graph

$[PROCEDURE : \#Open] \rightarrow (ActsOn) \rightarrow [WINDOW]$

may be associated to the formula:

$\exists x PROCEDURE(Open) \wedge ActsOn(Open, x) \wedge WINDOW(x)$

In the next part, we go further and propose some logical behavior of graph operators.

## 3.4 Logical behavior of algebraic graphs operators

Based on this definition of function $\Phi$, we propose here some theorems relative to the semantic of the four algebraic operators. Instead of giving the proof of these theorems, we prefer to discuss on possible links between an algebraic system (Conceptual Graphs) and a logic system (First order logic), and advantages that may be found for Information Retrieval. However, for each theorem, we will introduce the elements that could be used to build a complete and formal proof. We first propose the basic notion of logical equivalence between graphs.

**Definition 12 (Logical equivalence)** *Two graphs $G$ and $G'$ are logically equivalent iff* $\models \Phi(G) \Leftrightarrow \Phi(G')$

### *3.4.1 Simplification*

When two relations of the same type exist between two concepts, then one can remove a redundant relation. The obtained graph is simpler than the original one, but is also logically equivalent.

**Theorem 1 (Simplify)** *If $G_2$ is a simplification of $G_1$ then* $\Phi(G_1) \Leftrightarrow \Phi(G_2)$.

This theorem may be proved directly from the definition of the $\Phi$ function: when two relations $R$ of the same type exists between the same two concepts and in the same direction, then the associated formula given by $\Phi$ contains two instances of the predicate $r(x, y)$ linked by a conjunction. However, the reverse of this theorem is not true: though the following graphs containing only one concept $[A]$ and $[A : *x]$ are logically equivalent but there is no conceptual relations simplification [9] that leads from one to the other graph. Note that we loose an amount of "meaning" when using the $\Phi$ transformation. There is no total correspondence between graph operators and these logic definitions.

### *3.4.2 Join*

If we note by $G_1 +_{C_1,C_2} G_2$ the join between two graphs $G_1$ and $G_2$ on a common generic concepts $C_1$ of $G_1$ and $C_2$ of $G_2$, we can propose:

**Theorem 2 (External join)** $\Phi(G_1 +_{C_1,C_2} G_2) \Leftrightarrow \Phi(G_1) \wedge \Phi(G_2) \wedge (x_1 = x_2)$ *where $x_1$ and $x_2$ are variables of concepts $\Phi(C_1)$ and $\Phi(C_2)$.*

If we note by $+_{C,C'}G$ the unary operator expressing the internal join between two common concepts $C$ and $C'$ of a graph $G$ with anonymous referents (generic or specific).

**Theorem 3 (Internal join)** $\Phi(+_{C,C'}G) \Leftrightarrow \Phi(G) \wedge (x = x')$ *where $x$ and $x'$ are variables associated to the two predicates $\Phi(C)$ and $\Phi(C')$.*

### *3.4.3 Restriction*

We note the restriction $\mathcal{R}(G, C, C')$, where $C'$ is a restricted concept of $C$. In the case of a referent restriction $x$ to $r$, we propose the logical formula of the restricted graph as:

**Theorem 4 (Referent restriction)** $\Phi(\mathcal{R}(G, C(x), C(r))) \Leftrightarrow \Phi(G) \wedge (x = r)$ *where $x$ is a quantified variable and $r$ is a constant (i.e. a referent).*

When the restriction operator transforms a concept type $C$ to $C'$, we have:

**Theorem 5 (Type restriction)** $\Phi(\mathcal{R}(G, C(x), C'(x))) \Leftrightarrow \Phi(G) \wedge C'(x)$

[9] However, one could introduce new operators that simplify a graph not only by removing redundant relations.

*3.4.4 Properties*

If we note by $+$ all possible join operations, from these theorems we can easily deduce the following properties:

- $\Phi(G1 + G2) \supset \Phi(G1)$ (external join)
- $\Phi(G1 + G2) \supset \Phi(G2)$ (external join)
- $\Phi(+G) \supset \Phi(G)$ (internal join)
- $\Phi(\mathcal{R}(G, C(x), C(r))) \supset \Phi(G)$ (referent restriction)
- $\Phi(\mathcal{R}(G, C(x), C'(x))) \supset \Phi(G)$ (type restriction)

And finally we may deduce the following important property:

**Theorem 6 (Logical properties of the graph order)** *For two graphs $G$ and $G'$ , $G \leq G' \Rightarrow \models (\Phi(G) \supset \Phi(G'))$*

An equivalent result can be found in [4] and also in [3, p 392]. Going back to IR considerations, one can easily see that this property is of prime importance when considering logic-based retrieval models. It establishes a link between the partial order of conceptual graphs and the implication of the first order predicates. This gives a very important clue about the way one can derive an operational model from the theoretical model: considering again the second evaluation principle, it becomes obvious that computing $G_D \leq G_Q$, $G_D$ and $G_Q$ being respectively the conceptual graphs representing the semantic content of $D$ and $Q$, demonstrates the logical implication from $D$ to $Q$.

This means in turn that the basic matching mechanism to be implemented at the operational level may be a demonstrator based on the four algebraic graph operators described before. We shall develop more precisely this important point in section 5 when introducing the projection operator. As we have now first order predicates as a basis for knowledge representation for documents and queries, we have to adapt the theoretical model to this kind of data (remembering that it was designed for logical propositions and formulae).

# 4 The extended theoretical model

The modal logic is based on formulae valuation in relation to a set of interpretations called a world [11, 5, 22]. According to propositional logic (zero order), an interpretation is a function that associates logical values to each proposition of the language (see for example [12, 8, 9]). When we switch from one world to an other world, we may change the logical value of some variables and so the formulae may change their logical value. For example the formula $tree \supset pine$ is not true for all interpretations of the logic variables $tree$ and $pine$, but one can imagine a world where this formula is true: it is enough to interpret $pine$ as true.

In this part, we will extend the theoretical model proposed in [20, 21, 19] from propositional modal logic to first order modal logic. First of all, we recall the syntax of a first order language $\mathcal{F}$:

$LetF \in \mathcal{F}$ :

$F ::= P(t_1, ..., t_n)|F \wedge F|\neg F|\exists x_i F$

$withP \in \mathcal{P}, t_j \in \mathcal{V} \cup \mathcal{N}, x_i \in \mathcal{V}$

where $\mathcal{F}$ is the set of all formulae, $\mathcal{P}$ the set of the predicates, $\mathcal{V}$ the set of variables and $\mathcal{N}$ the set of name symbols. The $t_j$ stand for constants or variables and $x_i$ stand for variables. At first, a formula valuation $\models_W^{\mathcal{S}} F$ is no longer two-valued but becomes continuous and is noted now $V_W^{\mathcal{S}}(F)$. This valuation is computed from a given valuation of the predicates in a given world and using a combination of: $C_w(P)$. The interpretation system $\mathcal{S}$ is the set $(\mathcal{W}, \delta, \Delta, C, V, D, I)$ of all the items of the model:

- $\mathcal{W}$ is the set of all possible worlds;
- $\delta$ is a function from $\mathcal{W} \times \mathcal{W}$ to $[0; 1]$; This measures the truth value associated to the transition between two worlds.

- $\Delta$ is a function from $[0;1] \times [0;1]$ to $[0;1]$; It is used to combine the truth value of the world transition and the truth value of a formula in a world.
- C is a function from $\mathcal{P} \times \mathcal{W}$ to $[0;1]$; It gives initial values to all predicates in all worlds.
- V is a function from $\mathcal{F} \times \mathcal{W}$ to $[0;1]$; This function computes the continuous truth value of a formula.
- D is a domain: in first order logic, a new set D is introduced because of the use of predicates.
- I is an interpretation function which associates each predicate variable to an element of D using a specific assignment $s$ (see for example [12, 159],each constant and each predicate having $n$ parameters to a function of $D^n$ into $\mathcal{B}$, the set of Boolean values. As a formal simplification, we shall consider only one domain (the union of all possible domains) denoted D.

A modal interpretation is a set of worlds, where each world is a non modal interpretation of predicates and predicate variables. In the context of zero order modal logic that deals with propositions (and not with predicates), we don't have to consider the problem of what items are changing their interpretation when changes to other worlds occur, because we manipulate only one kind of entity (propositions, also called Boolean variables). When we want to shift to first order modal logic [7, 13, 16], we have to decide between predicates and predicate variables, which change their interpretation.

The idea behind the computation of the truth value is a computation that searches among the possible world for the value of a formula in a given world. Therefore, in this formalization, we take the maximum between the truth value known in the starting world and the truth value of the evaluated formula in the possible worlds. So, for all predicates $P \in \mathcal{P}$ we have:

$$V_W^{\mathcal{S}}(P) = MAX[C_w(P), V_W^{\mathcal{S}}(\diamond P)]$$

For every well formulated formulae $f, f_1, f_2$ of $\mathcal{F}$:

- the valuation of a conjunction is defined as: $V_W^{\mathcal{S}}(f_1 \wedge f_2) = MIN(V_W^{\mathcal{S}}(f_1), V_W^{\mathcal{S}}(f_2))$
- the valuation of the negation is: $V_W^{\mathcal{S}}(\neg f) = 1 - V_W^{\mathcal{S}}(f)$

The possibilility operator expresses that there exists a world where the formula is true. When associating a certainty value to it, we propose to take the maximum of a function $\Delta$ combining the valuation of world transition $\delta$ and the truth value in any ending world where $f$ is true:

$$V_W^{\mathcal{S}}(\diamond f) = MAX_{w' \in W}[\Delta(\delta(w, w'), V_{w'}^{\mathcal{S}}(f))]$$

The main difference here from Nie's formalization [21] is the use of predicates: in binary logic, the existential quantifier in $\exists x f$ means that there exists an interpretation of $x$ in $D$ that makes true the formula $f$. In continuous logic, the logic value of $\exists x f$ is the maximum value of $f$ when $x$ is interpreted in the domain.

$$V_W^{\mathcal{S}}(\exists x f) = MAX_{(I(x) \in D(w))}[V_W^{\mathcal{S}'}(f)]$$

where $x$ is a free variable of $f$, $\mathcal{S}'$ is an interpretation structure identical to $\mathcal{S}$ except for the interpretation of $x$ which is constant and belongs to the domain of world $w$, and $D(w)$ is a subset of the domain $D$: from lots of possible definitions of the existential quantifier interpretation, we choose the one based on domain inclusion. All first order formulae are expressed using a unique domain set $D$, and each world $w$ is associated to a subset of the domain noted $D(w) \subset D$.

We can now construct worlds in which a formula $f$ is true : we only have to reduce the domain $D(w)$ to the elements which correspond to a true value of $f$. As the logic valuation must be continuous, we define the value of a quantified formula as the maximum value obtained among all possible interpretation of $x$ in the domain $D(w)$.

# 5 A logical model for conceptual graph based retrieval

As explained before, we use the extended theoretical model as a framework for the design of a further operational model based on Conceptual Graphs. At first, we show in this section how every notion of the theoretical model is expressed in terms of the operational model (or graphs). Also important in our point of view, we show how this theoretical model leads to the definition of an algebraic operator for the evaluation of matching between graphs. To do so, we have to choose a meaning for the domain set $D$ introduced above.

## 5.1 Domain definition

As presented before, the function $\Phi$ gives the expression of any conceptual graph in terms of first order formulae, the domain set $D$ gives the interpretation of all predicate constants or variables. In the context of the Conceptual Graph formalism there exists the notion of percept. A percept belongs to the real world and consequently each referent has to be associated with a unique percept. Moreover, a concept is *well formed* (see [24, p87]) if the percept associated to the referent conforms to the percept associated to the conceptual type. As an example, the concept $[HUMAN : \#John]$ is well formed if we interpret the referent John as the name of a human being, and the concept type $HUMAN$ as the set of all human beings. We can write $I(HUMAN) :: I(John)$, where :: is the conformity relation of conceptual graphs. From the logical point of view, the definition of $D$ is then obvious: it is the set of all possible percept and then:

- The interpretation function $I$, applied to predicate constants, associates every constant to a percept in $D$.
- The interpretation function applied to a n-adic predicate is, by definition, a function of $D^n$ into $\mathcal{B}$. We have shown that concepts are associated to monadic predicates by the function $\Phi$. We define the interpretation $I(C)$ of this monadic predicate $C$ in this way: if $x$ is a referent and $C$ a concept, and if $x$ conforms to $C$, then $I(C)[I(x)] = 1$ which means that the interpretation function of $C$ applied to the interpretation of $x$ returns the true value.

We can now describe the other logical items of the model.

## 5.2 The set of worlds

Given the properties obtained with the logical expression of the algebraic operators, we have pointed out formally a relation between the graph partial order and the implication of the associated formula. This partial order is based on construction operators, and one can notice that any algebraic operator used to build the graphs can only add information to a given graph (or at least lets it intact). An interpretation of the above relation may then be stated as follows: a graph $G$ implies a graph $G'$, if every information contained in $G'$ can be found in $G$ (as the same notion or as a restricted notion). So we apply this property to the following definition of worlds.

The logical model defined as $P_K(G \rightarrow G')$ corresponds to a fuzzy valuation of $\models^{\mathcal{S}}_{\Phi(G)} \Phi(G')$. Therefore we may define a world $w$ as the set of all models of the logical formula $\Phi(G)$. As a consequence, the set $\mathcal{W}$ of all possible worlds is isomorphic to the set of all canonical graphs, which are correct conceptual graphs. The function $C_w$ gives a fuzzy valuation to a single predicate in a world $w$. A single predicate is associated either to a concept or to a conceptual relation. In the operational model we decided to restrain the valuation of $C_w(P)$ to $\{0, 1\}$. Value 1 is assigned when the predicate $P$ is true in the world $w$. This means that $C_w(P) = 1$ if and only if $\Phi(G) \supset P$, $C_w(P) = 0$ otherwise. The way the evaluation function $V^{\mathcal{S}}_W$ (see §4) and the function $\Phi$ are defined preserves the truth of the equivalence between partial order on graphs and the logical implication on associated logical formulae. The difference is that now expression (1) is asserted in a fuzzy modal logic instead of being related to a standard Boolean logic.

## 5.3 The knowledge certainty

In the Conceptual Graph representation, knowledge is stored in a *canonical base*, which is a set of primitive canonical graphs used to derive any possible canonical graphs. Now we show how the use of knowledge to build graphs, (i.e., to change to another world) can be associated to a certainty value. The way $C_w$ and $V_w$ are defined implies that given a world $w$ associated to the first order formula $\Phi(G)$, $V_w(\Phi(G)) = 1$ means that all individual predicates of $\Phi(G)$ are true. This is due to the definition of the function $\Phi$ in combination with the definition of algebraic operators. So, having $V_{\Phi(G)}(\Phi(G')) = 1$ means that graphs $G$ and $G'$ are ordered: $G \leq G'$ :

**Definition 13** *For two graphs $G$ and $G'$, $G \leq G' \Leftrightarrow V_{\Phi(G)}(\Phi(G')) = 1$*

From the IR point of view, and if we keep to the second evaluation principle, any document will correspond to a world, and the query will correspond to a formula. Retrieving the documents that satisfy the query will then correspond to find if there is a world (an interpretation of the document) where a non-null valuation $V_{\Phi(D)}(\Phi(Q))$ of the query graph $Q$ may be computed. Having $V_{\Phi(D)}(\Phi(Q)) = 1$ means that the document associated to the word implies the query with a total certainty. This situation is encountered whenever there is a document $D$ that corresponds to an exact

specialization of $Q$ (i.e. in this interpretation, every concept and relation in $Q$ is found in $D$, either as a specialized expression or as its original expression).

In a more general case, we have to deal with documents that are not exact specializations of a given query. In that case, we use the possibility operator in the valuation of $V_W^{\mathcal{S}}(\diamond P)$ which is defined as $MAX_{w' \in W}[\Delta(\delta(w, w'), V_{w'}^S(P)]$. This valuation expresses that there exist other possible worlds, and so other possible correct conceptual graphs, where $\Phi(Q)$ can be evaluated to a non-null value. The resulting value is a combination $\Delta$ of this valuation and of the certainty $\delta(w, w')$ related to this transformation. The choice made for $\Delta$ is the same made by Nie : the multiplication.

The interpretation given here to the transition from a world to another possible world is the following: we have chosen (see ) to build the set of possible worlds in conformity to the set of all possible correct Conceptual Graphs (also called canonical graphs [24]). To remain consistent with the previous definition, we base the relation $\delta$ on the partial order on graphs. Therefore, a change from a world $w$ to another world $w$' is possible if there exists a way of building the graph $\mathcal{G}(w')$ from $\mathcal{G}(w)$ using one of the algebraic operators. The possibility to change from a world $w$ to $w'$ is expressed by $\delta(w, w') \geq 0$. The remaining problem is the assignation of to certainty $\delta(w, w')$ associated to the transition between two worlds. Only heuristic considerations can lead the determination of such a valuation. We think that it might be related to a probabilistic modeling about the use of concepts in a given corpus or domain.

## 5.4 The Projection operator

The operational model leads us to the algebraic *projection operator* defined is [24]: If $G'$ is a specialization of a graph $G$, there must exist a mapping $\pi$ from $G$ to $G'$ called a projection:

- For each concept $c$ in $G$, $\pi(c)$ is a restriction of $c$ or is equal to $c$.
- For each conceptual relation $r$ in $G$, $\pi(r)$ is lower or equal to $r$.
- If concepts $c1$ and $c2$ are linked by relation $r$, then $\pi(c1)$ and $\pi(c2)$ are linked by $\pi(r)$.

The subgraph of $G$ identified by this mapping $\pi$, is called the projection of $G$ on $G'$. According to this definition , if we note $\pi(G)$ the projection of $G$ in $G'$, then it is clear that we have $G' \leq G$ and $\pi(G) \leq G$. Hence from property (1) we have also $\Phi(G) \supset \Phi(G')$.

This shows that the projection operator may be viewed as the basic retrieval operator: retrieving documents that imply query $Q$ is equivalent to retrieving documents that contain a projection of $Q$.

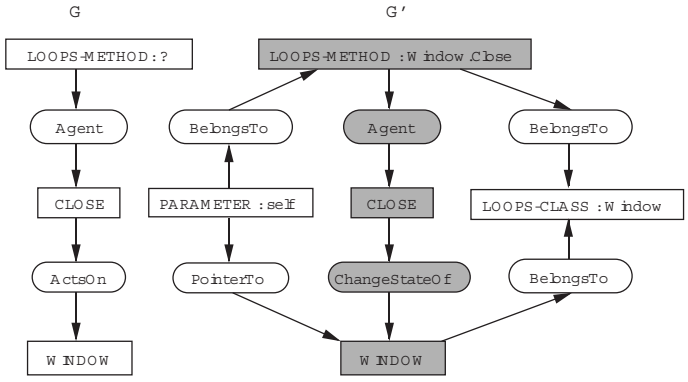


Figure 3: "An example of projection"

For example in figure 3, graph $G$ can be viewed as a query, and $G'$ as a document index ( in that case the index of a LOOPS language method called "close" from the "window" class). The subgraph of $G'$ containing darkened nodes corresponds to the projection of $G$. Note that in the projection, relation "Change state of" is a restriction of the corresponding relation "Acts on" of $G$. This is the same between concepts "Loops Method" of $G$ and "Loops Method: window.close" of $G'$.

This subgraph defines the matching of $G$ (the query) and $G'$ (the document) in a retrieval process.

# 6 Guidelines for weighting

This theoretical model proposes an approach of what could be a matching between graphs, but it doesn't say much about the valuation in use. It only describes the way values expressing uncertainties about facts are used to compute an implication strength. Putting some restriction to this implication, we can define some rules to guide the weighting.

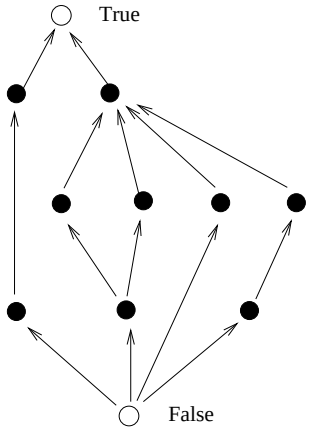


Figure 4: "Implication graph"

## 6.1 Basic rules

Let's consider the set of all possible queries and documents. The implication graph is the one obtained by linking queries and documents that imply them self with a certainty equal to 1.

**Definition 14 (Total implication)** *When the certainty of an implication is equal to 1, we call it a total implication :* $P(Y \rightarrow X) = 1$.

In the following, we emphasize about the total implication and we propose to examine possible properties of this implication graph. At first, let us recall the notion of equivalence:

**Rule 1 (Equivalence)** $P(X \rightarrow Y) = 1$ *and* $P(Y \rightarrow X) = 1$ *iff* $X$ *is logically equivalent to* $Y$.

In the context of IR, logical equivalence expresses a perfect matching between a document and a query. The total implication between a document and a query, means that when we consider as true all the knowledge within the document, then the query is true. When this query may be true without taking into account the document content, this means in fact that the query expresses some more general assertion than the document. Thus we propose the obvious consistency rule as follows:

**Rule 2 (Consistency)** *if* $P(X \rightarrow Y) = 1$ *then* $Y$ *is more "general" than* $X$*, thus* $X$ *is more "specific".*

The notion of "general" an "specific" introduced here is still informal. To be consistent with the meaning of the implication, and its use in the IR context, we must also notice that the implication is transitive:

**Rule 3 (Transitivity)** *the total implication viewed as a relation is transitive.*

As an extension to this rule, we must now consider the partial implication. When the certainty of an implication is not zero this expresses that there exists some way of transforming the premise to the conclusion of this implication. As an extension of the transitivity of the total implication, we introduce the fuzzy transitivity rule:

**Rule 4 (Fuzzy transitivity)** *if* $P(X \rightarrow Y) \neq 0$ *and* $P(Y \rightarrow Z) \neq 0$ *then* $P(X \rightarrow Z) \neq 0$.

Given these four basic rules, we can now analyze the possible cases of implication arising in the implication graph associated to a given query and a set of documents.

## 6.2 Basic cases of implication

Let us consider the set of all possible logical propositions, this set is organized in a graph where each vertices expresses a total implication. Each logical proposition can be used as an index for document or an expression for a query. Of course, in this graph, only few logical expressions are associated to actual documents.

Figure 4 represents a possible set of formulae with their implication relation. For clarity, we hide the transitive relations. There always exists a logic formula that is implied by every formula, the so called $True$ formula. On the other hand, the $False$ formula implies every other formula. $True$ is a formula which has the truth value $True$ in every interpretation, whereas $False$ is a formula which has the truth value $False$ in every interpretation. In the following, we will use $A \vee \neg A$ as a definition of $True$, and $A \wedge \neg A$ as a definition for false, where $A$ is a propositional predicate. These two values are beyond the scope of our interest for IR: they can be used neither as index document nor as a query.

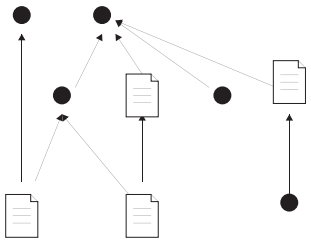

Figure 5: "Implication of documents"

When using this graph for indexing, some of these formulae will be used as actual index documents while every remaining formula is a possible query (see figure 5). We can now examine all possible situations for a given query:

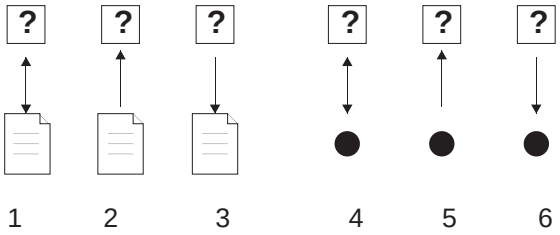


Figure 6: "All cases of implication"

The first case of Figure 6 corresponds to the rare case of the equivalence rule. The second possible situation is more frequent: here the query is an exact generalization of the document which is in then a possible good answer. In the third case of figure 6, the document is an exact generalization of the query. Though possible, this situation is less usual than the former because documents generally contain more information than queries.

The three remaining cases (4,5 and 6) arise when no document can be found as an exact specialization of the query: we only have possible total implications with formulae that does not correspond to actual documents. Case 4 expresses there exist a virtual document that could be logically equivalent to the query. Case 5 expresses there exists such a virtual specialized document. It is more pertinent to consider the 6th case as a possible generalization of the query rather than a possible virtual document. For all these three cases, we must detail the notion of closed document to the query.

## 6.3 Fuzzy cases of implication

The three last implication cases (4,5 and 6) of Figure 6 are the most common ones in IR situations: when the user doesn't have any knowledge about the documents in the corpus, he often builds queries that partially implies some documents and rarely queries that totally imply a document. If there is no document implying or being implied by an intermediate formula, the value of the implication is equal to zero. Due to the transitive rule, the only two cases remain in the situation (see Figure 7).

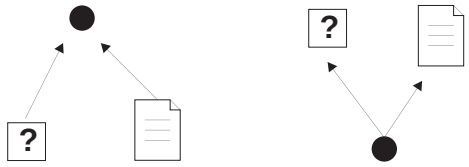


Figure 7: "Fuzzy case of implication"

In these two cases , there is no document implying or implied by the query, but there exists a formula implying or implied by both the document and the query. Thus, we consider that there exists a possible relation between them, because, there exist an uncertainty due to the query expression and due to the indexing. It is express in two fuzzy implication rules:

**Rule 5 (Fuzzy conclusion)** *If* $P(D \rightarrow Q) \neq 1$ *and* $P(Q \rightarrow D) \neq 1$, *and if there exists* $X$ *so that* $P(D \rightarrow X) = 1$ *and* $P(Q \rightarrow X) = 1$ *then* $P(D \rightarrow Q) \neq 0$ *and* $P(Q \rightarrow D) \neq 0$.

**Rule 6 (Fuzzy premise)** *If* $P(D \rightarrow Q) \neq 1$ *and* $P(Q \rightarrow D) \neq 1$, *and if there exists* $X$ *so that* $P(X \rightarrow Q) = 1$ *and* $P(X \rightarrow D) = 1$ *then* $P(D \rightarrow Q) \neq 0$ *and* $P(Q \rightarrow D) \neq 0$.

These rules express the fact that if there exists a specific or generic formula for a document and a query not in total relation, then there exists a non null uncertain implication between them. This formula $X$ can be viewed as a query or document modification (See Figure 8). For example, in the first case, we could consider the query $Q$ as being too specialized for this corpus, while in the second case, we would cope with uncertainty of document index.

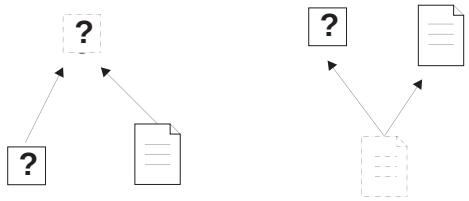


Figure 8: "Document or query modification"

In fact, these rules are introduced as general guidelines for what could be a weaker valuation for the uncertain implication. In the following we go further and we consider situations involving two documents.

## 6.4 Precision rule

As we consider precision-oriented systems, we should favor the most specialized documents as responses. According to the consistency rule, the specialization is based on the reverse total implication. In the situation where one document $D_1$ implies the query $Q$ and is implied by an other document $D_2$, a precision oriented system would propose $D_1$ as a better response than $D_2$ (see Figure 9).

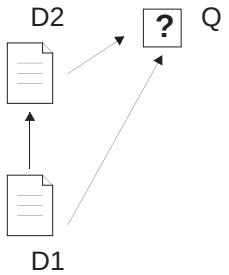


Figure 9: "Precision-oriented situation"

In this situation, the value of the reverse implication makes the difference to have:

$$F[P(D_2 \rightarrow Q), P(Q \rightarrow D_2)] < F[P(D_1 \rightarrow Q), P(Q \rightarrow D_1)]$$

As we have in this situation $P(D_2 \rightarrow Q) = 1, P(D_1 \rightarrow Q) = 1$, and if we define the combination function $F$ as an increasing, monotonous function when one of the two parameter are fixed [10], we obtain the following condition:

$$P(Q \rightarrow D_2) < P(Q \rightarrow D_1)$$

So we propose the following precision rule:

**Rule 7 (Precision oriented)** *Given $D_2$, $Q$ and $D_1$ so that $P(D_2 \rightarrow Q) = 1$, $P(D_1 \rightarrow Q) = 1$, then $P(Q \rightarrow D_2) < P(Q \rightarrow D_1)$.*

In this situation of precision oriented systems, this shows the importance of the value of the reverse implication for the computation of the final matching value. This preliminary idea has, in our opinion, to be further investigated, and could be related for example to the retrieval of structured documents: in the above examples, if $D_1$ were a section of document $D_2$, both satisfying query $Q$, then the most precise (i.e. focussed) response would be section $D_1$.

# 7 Conclusion

The first description of a formal retrieval model based on logic was proposed by Nie and this model used propositional modal logic (zero order). Experimentation within the RIME project has demonstrated the necessity of a better integration of knowledge representation in the retrieval model to guarantee full effectiveness of the retrieval process. This integration is possible when the chosen formalism for expressing and manipulating knowledge has clear and complete formal properties. We have proposed the use of Conceptual Graphs to fulfill these goals, and demonstrated how the basic properties and operators fit well with the problematic of logic-based information retrieval. The integration of this knowledge model in the theoretical model implied an extension of the latter to first order logic. This extension of the original model has also been detailed. Finally, we have shown how this extended theoretical model is used as a guideline for designing an operational model which is the basis of any implementation of the theoretical approach. In this framework, we had to define also a more precise semantics of Conceptual-Graph to investigate this formalism in our model. This led us to some classifications and simplification of Sowa's original model. One should also remind that some aspects of the above propositions still need further developments: our main goal here was to investigate an approach and see how it fits with IR requirements. The IR graph based approach has been studied in [14] to discover basic IR assumption expressed in derivation rules. This model is also use in the HYPERIME system [15], an hyper-media extension to RIME, and EMIR [18] an image retrieval system.

All these ideas have been developed and implemented within the ELEN project: implementation has been carried on using object-oriented languages (XEROX LOOPS language). The application field here is Software Reuse: an index language based on Conceptual Graphs has been developed for supporting the retrieval of software components within software libraries. The ELEN retrieval component, based on the projection operator presented before, has been tested on UNIX and LOOPS libraries. Operational optimizations of the prototype are on their way using C++ and O2 (an object oriented database system).

Going back to the previous statement of this paper saying that the RIME and ELEN projects are based on the same underlying retrieval model, theoretical and practical results of the ELEN project will allow us to improve and achieve the implementation of the RIME prototype in a fully operational environment based on object-oriented database (O2).

# References


[1] Jon Barewise and John Etchemendy. *The language of First-Order Logic*, volume 23 of *Lecture Notes*. Center for the Study of Language and Information, Stanford, Ventura Hall Stanford, CA 94305, 1990.

[2] Catherine Berrut. Indexing medical reports: The rime approach. *Information Processing and Management*, 26(3):93–109, 1990.


[10] i.e. for every $a \in [0,1], F(a, D_2) : [0,1] \rightarrow [0,1]$ is continuous and monotonous function, and for every $b \in [0,1], F(D_2, b) : [0,1] \rightarrow [0,1]$ is continuous and monotonous function.

[3] Michel Chein and Marie-Laure Mugnier. Conceptual graphs : Fundamental notions. *Revue d'intelligence artificielle*, 6(4):365–406, 1992. In english.

[4] Michel Chein and Marie-Laure Mugnier. Conceptual graphs are also graphs. Technical Report R.R. LIRMM 003-95, LIRMM (CNRS & Universite de Montpellier II), 161, rue Ada, 34392 Montpellier Cedex 5, France, January 1995.

[5] Brian Chellas. *Modal Logic, An Introduction*. Cambridge University Press, 1980.

[6] Jean-Pierre Chevallet. *Un Modèle Logique de Recherche d'Informations appliqué au formalisme des Graphes Conceptuels. Le prototype ELEN et son expérimentation sur un corpus de composants logiciels*. PhD thesis, Université Joseph Fourier, Grenoble, 1992.

[7] E. A. Emerson. *Temporal and Modal Logic*, volume B. MIT Press, 1990.

[8] Richard L. Epstein. *The Semantic Foundation of Logic volume 1: Propositional Logics*, volume 35 of *Nijhoff International Philosophy*. Kluwer academemic publishers, 1990.

[9] Richard L. Epstein. *The semantic Foundation of Logic : Predicate Logic*. Oxford University Press, 1994.

[10] John Esh, Maurice Pagnucco, Michel Wermelinger, and Heather Pfeiffer. Linear - linear notation interface. In Robert Levinson and Gerard Ellis, editors, *Fourth International Workshop on PEIRCE: A Conceptual Graph Workbench*, pages 46–54, University of Maryland, Maryland, USA, August 1994.

[11] D. Gabbay, C. J. Hogger, and J.A. Robinson, editors. *Handbook of logic in Artificial Intelligence and Logic Programming*, volume 1. Clarendon press, oxford, 1993.

[12] Jean H. Gallier. *Logic for Computer Science. Foundation of Automatic Theorem Proving*. Harper & Row, Publishers, New York, 1986.

[13] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic*. Methuen and Co, 1972.

[14] Theo Huibers, Iadh Ounis, and Jean-Pierre Chevallet. Axiomatization of a conceptual graph formalism for information retrieval in a situated framework. Technical Report RAP95-004, Groupe MRIM of the Laboratoire de Génie Informatique in Grenoble, France, July 1995.

[15] Ammar Kheirbek and Yves Chiaramella. Modeling hypermedia with conceptual graphes. In *WorkShop on Intelligent Hypertext, GaitherSburg, Maryland, USA, December*, 1994.

[16] Saul A. Kripke. Semantical analysis of modal logic in normal modal propositional calculi. *Zeitsxhr. f. math. Logik und Grundlagen d. math.*, 9:67–96, 1963.

[17] Mounia Lalmas. The use of logic in information retrieval modelling. Technical Report TR-1996-1, Department of Computing Science, University of Glasgow, Glasgow G128QQ, Scotland, January 1996.

[18] Mourad Mechkour, Catherine Berrut, and Yves Chiaramella. Using conceptual graph frame work for image retrieval. In *International conference on MultiMedia Modeling (MMM'95), Singapore*, pages 127–142, 14-17 November 1995.

[19] Janyun Nie and Yves Chiaramella. A retrieval model based on an extended modal logic and its application to the rime experimental approach. In *ACM SIGIR 90 Bruxelles*, pages 25–43, 1990.

[20] Jianyun Nie. An information retrieval model based on modal logic. *Information Processing & Management*, 25(5):477–491, 1989.

[21] Jianyun Nie. *Un modèle logique général pour les Systèmes de Recherche d'Informations. Application au prototype RIME*. PhD thesis, Université Joseph Fourier, 1990.

[22] S. Popkorn. *First steps in modal logic*. Cambridge univerity press, 1994.

[23] Schank. Conceptual dependency: a theory of natural language understanding. *Cognitive Psychology*, 3:552–631, 1972.

[24] John F. Sowa. *Conceptual Structures: Information Processing in Mind and machine*. Addison-Wesley, 1984.

[25] C. J. van Rijsbergen. A new theoretical framework for information retrieval. In *ACM Conference on Research and development in Information Retrieval, Pisa*, pages 194–200, 1986.

[26] Michel Wermelinger. Conceptual graphs and first-order logic. In Gerard Ellis, Robert Levinson, William Rich, and John f. Sowa, editors, *Third International Conference on Conceptual Structures, ICCS'95*, volume 954 of *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science*, pages 323–337, Santa Cruz CA, USA, August 1995. Springer.

[27] Jean Pierre Chevallet Yves Chiaramella. About retrieval model and logic. *The Computer Journal*, 35(3):233–242, 1992.