ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval
(Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

Document Retrieval by Relevance Terminological Logics

Umberto Straccia

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

Document Retrieval by Relevance Terminological Logics

Umberto Straccia *

Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche
Pisa, Italy

Abstract

Information Retrieval (IR) is presented as the task of retrieving the documents that are *relevant* to a given query. In the context of a Terminological Logic (TL) based approach to IR, this amounts to embodying a notion of relevance in the logical implication relation of the chosen TL. Among the many possible readings of the term “relevance”, the one captured by *relevance logic*, and in particular by first-order tautological entailment, can be viewed as a promising source of inspiration, to the end of incorporating a logic-based form of relevance in the inference mechanism of TLs.

The aim of this paper is to present a *Relevant Terminological Logic*, while maintaining the desired “relevance” flavour of relevance logics, which could be considered as a base towards a suitable document description logic for IR purposes.

1 Introduction

Terminological Logics (TLs, for short), also called Description Logics or Concept Logics, have widely gained popularity because of the vastity of their possible applications. These logics allow both the representation of concepts / terms / classes and their structuring according to a partial order, and to state that individuals are instances of concepts. Typically, concepts are build out of two types of symbols, *primitive concepts* and *primitive roles* (called attributes in databases). These primitive symbols can be combined, using appropriate operators, in order to obtain more complex concepts and roles, whose semantics is given in a Tarsky style: an interpretation maps concepts into subsets of the domain and roles into binary relations over the domain. For example, given the concepts `Document`, `InformationRetrieval`, `TerminologicalLogics` and `Italian`, the roles `dealswith` and `author`, the operators for conjunction of concepts ($\sqcap$), existential ($\exists$) and universal ($\forall$) quantification over roles, then the concept (or *document description*)

```
Document
  ⊓ (∃dealswith.InformationRetrieval)
  ⊓ (∃dealswith.TerminologicalLogics)
  ⊓ (∀author.Italian)
```

is a complex concept which denotes the set of documents dealing with Information Retrieval, with TLs, and whose authors are all Italian.

Assertions are build up using individuals, concepts and roles. For example, given the individuals `Umberto` and `doc211`, and the complex concept above, then the assertion

```
(Document
  ⊓ (∃dealswith.InformationRetrieval)
  ⊓ (∃dealswith.TerminologicalLogics)
  ⊓ (∀author.Italian))(doc211)                                     (1)
```

states that “`doc211` is a document dealing with Information Retrieval and TLs, and whose authors are all Italian”; whereas the assertion

```
author(doc211, Umberto)                                           (2)
```

*This work has been carried out in the context of the project FERMI 8134 - “Formalization and Experimentation in the Retrieval of Multimedia Information”, funded by the European Community under the ESPRIT Basic Research scheme.

states that “Umberto is an author of document doc211”.

It is easy to see, that structured documents, like HTML documents, etc., can be represented by means of TLs. For example, the HTML document `html2`

```
<HEAD>
  <TITLE>
    ``FERMI Anchor"
  </TITLE>
</HEAD>
<A HREF=``http://www.dcs.gla.ac.uk/fermi/"> FERMI </A>
```

could be described by means of the following assertion

```
(HTMLDocument
  □ (∃head.∃title.“FERMI Anchor”)
  □ (∃hrefanchor.
    (∃token.“FERMI”)□
    (∃ref.“http://www.dcs.gla.ac.uk/fermi/”))) (html2) (3)
```

Of course, SGML documents can be represented in a similar way, too. Finally, for example, the bibliography record of each document in the system DIENST, an architecture for distributed digital document libraries, which is an ARPA-sponsored Computer Sciences Technical Report project, can be easily represented by means of TLs. For example, the bibliography record of the document `tr9512`

```
-----
ID:: tr9512
TITLE:: Document retrieval by relevance TLs
AUTHOR:: Umberto Straccia
DATE:: September, 12th, 1995
ABSTRACT::
Information Retrieval (IR) is presented as the task ...
END::
-----
```

could be described by means of the following assertions

```
(TechReport
  □ (∃id.“tr9512”)
  □ (∃title.“Document retrieval by relevance TLs”)
  □ (∃author.Researcher)
  □ (∃date.“September, 12th, 1995”)
  □ (∃abstract.“Information Retrieval ...”)) (tr9512), (4)

  author(tr9512, Umberto),

  ((∃name.“Umberto”)□
  (∃surname.“Straccia”))(Umberto)
```

In recent years a substantial amount of research has been carried out in the context of TLs. This work ranges from the extension of TLs (from an expressive power and semantics point of view), as for example in [2, 6, 9, 11, 18, 20, 22] to the development of real systems based on TLs, as for example in [12, 13, 14] to the identification of new reasoning techniques and their computability and complexity analysis, as for example in [4, 5, 17, 19].

Recently, a TL, MIRTl, has been proposed as a model for a logic-based approach to Information Retrieval [9, 18] (IR, for short).

In a logic based approach to IR, an IR system can be described as in Figure 1 (note, that our aim is to describe only a possible base schema and not a detailed one).

There are mainly two important parts in such a system. The *Document Base* (DocB) and the *Document Knowledge Base* (DocKB).

Figure 1: A logic-based IR system

DocB : The DocB is the set of the multimedia documents handled by the system, persistently stored in some way. Each document is identified by an unique ID.

DocKB : The DocKB is a set of logical descriptions (formulae of the underlying logic) describing both the documents of the DocB and knowledge about the application domain. The DocKB is partitioned into two subsets: the *Document Description Base* (DocDB) and the *Domain Description Base* (DomDB).

DocDB : For a document of the DocB, by using indexing technics, pattern recognition technics, natural language processing, manual processing, etc., the system gets logical descriptions about the content and layout of this document. The set of all the descriptions of the documents in DocB forms the DocDB. Therefore, DocDB is the set of logical descriptions of the documents in the DocB.

DomDB : The DomB describes knowledge about the application domain. Typically, it contains three subsets: the *Domain Knowledge* (DomK), the *Dictionary Base* (DicB) and the *Synonymous Base* (SynB).

DomK : The DomK contains knowledge about a specific application domain of the system, *i.e.* documents are all about laws, or are invoices, or are about music styles, or are computer science technical reports, or are medical reports, etc. and, thus, includes specific information about document's content and layout.

DicB and SynB : The DicB and SynB contain term and synonymous definitions which are specific to the application domain, as general term definitions.

The other modules handles the interaction of an user to the system. In particular, the *User Query Module* (UQM) enables an user to formulate a query Q by means of a language, *e.g.* a formal query language as SQL, or natural language, or natural language with images and sound (a multimedia document), or whatever is needed. Note that a query could be also a (short) multimedia document. In fact in this case the goal of the system could be: retrieve those documents whose content is relevant to the content of the query document.

The query Q is successively "formalized" into a formula of the underlying logic. The *Logic-based Query Module* (LQM) is responsible for query solving, and thus queries the DocKB about those documents which are "relevant" to the given query, getting a set of unique document identifiers (the IDs of the relevant documents), called answer set. This

set is sent to the *Answer and User Relevance Feedback Module* (AURFM). This module interacts with the DocB and, by means of the answer set, returns in some form information about the set of documents identified by the answer set, e.g. the documents themselves, an explanation about the retrieval of each document (*i.e.* why the system has retrieved a document), etc..

Finally, user relevance feedback could be achieved in the following way:

1. the user selects those informations about the retrieved documents, *i.e.* by selecting a subset of documents, pieces of documents, explanations, etc., which he retains relevant to his purposes;
2. this set is then “formalized” into a formula of the underlying logic in a similar way as for the query formulated by the user, getting a relevance feedback formula, which is sent to the LQM. Now the LQM can combine the user query with the relevance feedback formula, yielding a new query which can be seen as a “more precise reformulation” of the original user query;
3. finally, a new answer set is retrieved.

Now, typically, IR is presented as the task of retrieving the documents that are *relevant* to a given information need (query). The task of IR can thus be described in logical terms as the extraction, from a DocKB, of those documents d that, given a query q make the formula $d \rightarrow q$ valid, where $\rightarrow$ is the logical implication relation of the adapted logic: for example, in TLs, the task of IR is formalized as: given a set of assertions Σ (the DocKB) and a query concept Q , what is the set of individuals (documents id) d such that Σ *logically implies* $Q(d)$ (in symbols, $\Sigma \models Q(d)$)?

In the context of TL-based approach to IR, this amounts to embodying a notion of relevance in the logical implication relation of the chosen TL. It is clear that the notion of logical implication alone is not adequate for IR purposes, as, for example, probabilities are missing. This means that TLs needs to be extended including other features. A good candidate seems to be a TL based logic which incorporates:

Probabilistic extension : Probabilistic versions of TLs [18, 22] could be investigated as a means of making explicit various sources of uncertainty, such as uncertainty related to domain knowledge and uncertainty related to automatic document representation, which is typical in IR;

Concrete domain : Incorporating kinds of data types of the underlying concrete domain [2] as “string”, “integer”, “link” (link to the position of a keyword in a document, link to another related document, etc.), etc.;

Rule language : Including rules for the DocKB module, as, for example, in [7];

Closed World Assumption, Closed Domain Reasoning, etc. Close world reasoning and closed domain reasoning seem to be suitable for IR purposes, as they are close to usual databases reasoning [6, 15, 16].

Therefore, in order to achieve a notion of relevance implication in such an extended TL, as first step, a notion of relevance must be enclosed into the inference engine of the “pure” TL (TL without any extensions).

Among the many possible readings of the term “relevance”, the one captured by *relevance logic* [1], and in particular by first-order tautological entailment, can be chosen as a promising source of inspiration to the end of incorporating a logic-based form of relevance in the inference mechanism of TLs. The underlying tenet of the criticism of relevance logicians to classical logic is that relevance of a premise to a conclusion is essential for asserting the implication between the premise and the conclusion. As a consequence of this criticism, the key concern underlying relevance logics is that of formalizing in a logic-based way a more suitable form of relevance than material implication.

In semantical terms, this amounts to adopting a four-valued semantics [3, 8] for TLs, thus obtaining *Relevant Terminological Logics*, where assertions can be not only *true* or *false*, but also neither true nor false (a state of affairs which is known as *unknown*), and also both true and false (a state of affairs which is known as *contradiction*).

Logics of this kind have already been used in Knowledge Representation and Reasoning in order to avoid the so-called paradoxes of logical implication, like $a \wedge \neg a \models b$ and $b \models a \vee \neg a$ for all b , when reasoning on concepts and individuals. These logics have also been proven to have a generally better computational behaviour than their two-valued analogues [8, 11].

Unfortunately, we have observed that the adoption of the by now classical four-valued semantics [11] results in a too drastic loss of inferential capabilities for IR.

The aim of our work is to present a less restrictive four-valued semantics for TLs, while maintaining the desired “relevance” flavour of relevance logics, which could be considered as a suitable core, towards a TL based extended logic for IR purposes.

In particular, we will present a four-valued variant of $\mathcal{ALC}$ (a powerful concept language including conjunction, disjunction, and negation of concepts, together with existential and universal quantification on roles), which we will call $(\mathcal{ALC})_4$, and which can be considered as a good representative for $\mathcal{AL}$ -languages.

We will also show that the defined entailment relation captures a close (structural) relationship between a DocKB and a query. Therefore, the defined entailment relation could arguably be a good theoretical and practical bases towards a logic-based approach to IR.

The rest of this paper is organized as follows. In the next section we will give the syntax and semantics of $(\mathcal{ALC})_4$. In Section 3 we will discuss our semantics and we will show, by means of examples, differences to standard two-valued semantics, differences to existing four-valued semantics and its suitability for IR, whereas Section 4 concludes.

2 The four-valued concept language $(\mathcal{ALC})_4$

In this section we present the syntax and semantics of $(\mathcal{ALC})_4$ ¹. For a more general presentation with respect to the operators allowed in TLs, see [10]. For a more extensive discussion on four-valued TLs, see [11].

2.1 Syntax

We assume two disjoint alphabets of symbols, called *primitive concepts* and *primitive roles*. The letter A will always denote a primitive concept and the letter R will denote a primitive role. The *concepts* (denoted below by C and D) and the *roles* (which in $(\mathcal{ALC})_4$ are always primitive) of the language $(\mathcal{ALC})_4$ are formed out of primitive concepts and roles according to the following syntax rule:

C, D	$\longrightarrow$	A		(primitive concept)
		$C \sqcap D$		(concept conjunction)
		$C \sqcup D$		(concept disjunction)
		$\neg C$		(concept negation)
		$\forall R.C$		(universal quantification)
		$\exists R.C$		(existential quantification)

Furthermore, we assume an alphabet of symbols called *individuals*, disjoint from the alphabets of primitive concepts and primitive roles. Individuals will be denoted below by a and b . An *assertion* is an expression of type $C(a)$ (meaning that a is an instance of C), where a is an individual and C is an $(\mathcal{ALC})_4$ concept, or an expression of type $R(a, b)$ (meaning that a is related to b by means of R), where a and b are individuals and R is an $(\mathcal{ALC})_4$ role. Finally, an $(\mathcal{ALC})_4$ *knowledge base* is a finite set Σ of assertions.

Note that in the following, we use parentheses whenever we need to disambiguate concept expressions. For example, we will write $(\forall R.C) \sqcap D$ to mean that the concept D is not in the scope of $\forall R$.

2.2 Semantics

The formal semantics of the logic $(\mathcal{ALC})_4$ is four-valued. The four truth values are the elements of $2^{\{t, f\}}$, the powerset of $\{t, f\}$, i.e. $\{t, f\}, \{\}, \{t\}$ and $\{f\}$. These values are best understood as epistemic states of a reasoning system about some proposition. Under this view, if the truth value of a proposition contains t , then the system has evidence to the effect – or beliefs – that the proposition is true. Similarly, if the truth value of a proposition contains f , then the system believes that the proposition is false. The truth value $\{\}$ corresponds to lack of knowledge, and the truth value $\{t, f\}$ corresponds to inconsistent knowledge.

In four-valued semantics it is possible to have inconsistent knowledge about some proposition without being totally inconsistent. This property, which is shared by other relevance logics, is touted as one of the advantages of relevance logics, especially when modelling states of knowledge.

Definition 2.1 An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non empty set $\Delta^{\mathcal{I}}$ (the domain of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the interpretation function of $\mathcal{I}$) such that

1. $\cdot^{\mathcal{I}}$ maps every concept into a function from $\Delta^{\mathcal{I}}$ to $2^{\{t, f\}}$;

¹ Although we restrict our attention to a four-valued variant of $\mathcal{ALC}$, our framework can be applied to other languages as well.

2. $\cdot^{\mathcal{I}}$ maps every role into a function from $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to $2^{\{t,f\}}$;
3. $\cdot^{\mathcal{I}}$ maps every individual into $\Delta^{\mathcal{I}}$;
4. $a^{\mathcal{I}} \neq b^{\mathcal{I}}$, if $a \neq b$.

The interpretation function can best be understood as an extension function of two separate two-valued extensions – the positive extension and the negative extension – defined next.

Definition 2.2 Let $\mathcal{I}$ be an interpretation. The positive extension of a concept C , written $C_+^{\mathcal{I}}$, is the set of domain elements that are known to belong to the concept, and is defined as $\{d \in \Delta^{\mathcal{I}} : t \in C^{\mathcal{I}}(d)\}$. The negative extension of a concept C , written $C_-^{\mathcal{I}}$, is the set of domain elements that are known not to belong to the concept, and is defined as $\{d \in \Delta^{\mathcal{I}} : f \in C^{\mathcal{I}}(d)\}$. The positive and negative extension of roles are defined similarly.

Note that, unlike standard semantics, these two sets need not to be complement of each other³. Domain elements that are members of neither set are not known to belong to the concept and are not known not to belong to the concept. This is a perfectly reasonable state for a system that is not a perfect reasoner or does not have complete information. Domain elements that are members of both sets can be thought of as inconsistent with respect to that concept in that there is evidence to indicate that they are in the extension of the concept and, at the same time, not in the extension of the concept. This is a slightly harder state to rationalize but can be considered a possibility in the light of inconsistent information.

The extensions of concepts and roles have to meet certain restrictions, designed so that the formal semantics respects the informal meaning of concepts and roles. For example, the positive extension of the concept $A \sqcap B$ must be the intersection of the positive extension of A and B and its negative extension must be the union of their negative extensions, thus formalizing the intuitive notion of conjunction in the context of the four-valued semantics.

Definition 2.3 Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation. The interpretation function $\cdot^{\mathcal{I}}$ has to meet the following equations for concepts: for each $d \in \Delta^{\mathcal{I}}$

$$\begin{aligned}
 t \in (C \sqcap D)^{\mathcal{I}}(d) & \text{ iff } t \in C^{\mathcal{I}}(d) \text{ and } t \in D^{\mathcal{I}}(d) \\
 f \in (C \sqcap D)^{\mathcal{I}}(d) & \text{ iff } f \in C^{\mathcal{I}}(d) \text{ or } f \in D^{\mathcal{I}}(d) \\
 t \in (C \sqcup D)^{\mathcal{I}}(d) & \text{ iff } t \in C^{\mathcal{I}}(d) \text{ or } t \in D^{\mathcal{I}}(d) \\
 f \in (C \sqcup D)^{\mathcal{I}}(d) & \text{ iff } f \in C^{\mathcal{I}}(d) \text{ and } f \in D^{\mathcal{I}}(d) \\
 t \in (\neg C)^{\mathcal{I}}(d) & \text{ iff } f \in C^{\mathcal{I}}(d) \\
 f \in (\neg C)^{\mathcal{I}}(d) & \text{ iff } t \in C^{\mathcal{I}}(d) \\
 t \in (\forall R.C)^{\mathcal{I}}(d) & \text{ iff } \forall e \in \Delta^{\mathcal{I}}, t \in R^{\mathcal{I}}(d, e) \text{ implies } t \in C^{\mathcal{I}}(e) \\
 f \in (\forall R.C)^{\mathcal{I}}(d) & \text{ iff } \exists e \in \Delta^{\mathcal{I}}, t \in R^{\mathcal{I}}(d, e) \text{ and } f \in C^{\mathcal{I}}(e) \\
 t \in (\exists R.C)^{\mathcal{I}}(d) & \text{ iff } \exists e \in \Delta^{\mathcal{I}}, t \in R^{\mathcal{I}}(d, e) \text{ and } t \in C^{\mathcal{I}}(e) \\
 f \in (\exists R.C)^{\mathcal{I}}(d) & \text{ iff } \forall e \in \Delta^{\mathcal{I}}, t \in R^{\mathcal{I}}(d, e) \text{ implies } f \in C^{\mathcal{I}}(e)
 \end{aligned}$$

Observe that, according to the intuitive meaning of the qualified existential operator $\exists$, $(\exists R.C)_+^{\mathcal{I}} = (\neg \forall R. \neg C)_+^{\mathcal{I}}$ and $(\exists R.C)_-^{\mathcal{I}} = (\neg \forall R. \neg C)_-^{\mathcal{I}}$.

The notion of subsumption between two concepts is defined in terms of their positive extensions as follows.

Definition 2.4 Given two $(\mathcal{ALC})_4$ -concepts C and D :

1. C subsumes D (written $D \sqsubseteq C$) iff $D_+^{\mathcal{I}} \subseteq C_+^{\mathcal{I}}$, for every interpretation $\mathcal{I}$;
2. C is equivalent to D , written $C \equiv D$, iff $C_+^{\mathcal{I}} = D_+^{\mathcal{I}}$, for every interpretation $\mathcal{I}$.

With respect to assertions, we have the following definitions.

²This restriction on individuals, called *unique name assumption*, ensures that different individuals denote different elements of the domain.

³In two-valued standard semantics, we have: $C_+^{\mathcal{I}} \cap C_-^{\mathcal{I}} = \emptyset$ and $C_+^{\mathcal{I}} \cup C_-^{\mathcal{I}} = \Delta^{\mathcal{I}}$, or equivalently $C_-^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C_+^{\mathcal{I}}$.

Definition 2.5 An interpretation $\mathcal{I}$ satisfies an assertion α iff

$$\begin{array}{ll} t \in C^{\mathcal{I}}(a^{\mathcal{I}}) & \text{when } \alpha = C(a) \\ t \in R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) & \text{when } \alpha = R(a, b) \end{array}$$

■

Definition 2.6 An interpretation $\mathcal{I}$ satisfies a knowledge base Σ iff $\mathcal{I}$ satisfies all assertions in Σ . A knowledge base Σ entails an assertion α , written $\Sigma \models \alpha$, iff for all interpretations $\mathcal{I}$, if $\mathcal{I}$ satisfies Σ then $\mathcal{I}$ satisfies α . ■

As last, note that we voluntarily left out the so called *terminological axioms* which mainly able the definition of new concepts in terms of previously defined concepts and, thus, can be used to encode, for example, thesaural knowledge, i.e. included into the DKB or DicB. An example, is the definition of $\mathcal{L}^{\text{TeX}}$ typesetted documents:

$$\text{LaTeXTypesettedDocument} = (\text{Document} \sqcap \exists \text{typesetwith.LaTeX})$$

The inclusion of terminological axioms can be done in a similar way as for standard two-valued TLs (see, for example, [10]).

3 Discussion of the semantics

In this section we will discuss our semantics and we will show, by means of examples, differences to standard two-valued semantics, differences to existing four-valued semantics and its suitability for IR purposes.

In order to compare our four-valued semantics with the two-valued one, we will use the notation " $\models_2$ " for the classical two-valued logical implication relation, and " $\sqsubseteq_2$ " for the two-valued subsumption relation.

3.1 Soundness of four-valued semantics with respect to classical semantics

In this section we will show that reasoning with respect to four-valued semantics is sound with respect to classical semantics. To this purpose, we will show that the set of two-valued interpretations is a (proper) subset of the set of four-valued interpretations.

Consider a four-valued interpretation $\mathcal{I}$ such that for every primitive concept A and primitive role P , the positive and negative extensions are together disjoint and exhaustive, i.e. $A_-^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A_+^{\mathcal{I}}$ and $P_-^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus P_+^{\mathcal{I}}$. By case analysis on the operators, $\sqcap, \sqcup, \neg, \forall$ and $\exists$, it can be shown that such an interpretation is in fact an usual two-valued interpretation for TLs. In fact, note that for such interpretations, given a concept C and a role R , the following hold:

$$t \in C^{\mathcal{I}}(d) \text{ iff } f \notin C^{\mathcal{I}}(d)$$

$$t \in R^{\mathcal{I}}(d, e) \text{ iff } f \notin R^{\mathcal{I}}(d, e)$$

Therefore, the set of two-valued interpretations is a (proper) subset of the set of four-valued interpretations. It then follows easily the soundness of reasoning in this logic with respect to standard two-valued semantics.

Proposition 3.1 Let Σ be a knowledge base, α an assertion and C, D two concepts. Then

1. if $C \sqsubseteq D$ then $C \sqsubseteq_2 D$;
2. if $\Sigma \models \alpha$ then $\Sigma \models_2 \alpha$.

■

Soundness of the subsumption relation and the entailment relation is an important requirement if the semantics is to capture some of the intuitive ideas underlying TLs.

3.2 Some subsumption relations

The subsumption and entailment relations supported by this logic are interesting subsets of the two-valued relations, suitable for IR, as we will see below.

In $(\mathcal{ALC})_4$, the concept

$\text{Document} \sqcap \exists \text{dealswith.Relevance}$

subsumes, as desired, the concept

Document
 $\sqcap (\exists \text{dealswith.}(\text{InformationRetrieval} \sqcap \text{Relevance}))$
 $\sqcap (\forall \text{author.Italian})$

i.e., a document dealing with IR and relevance, whose authors are all Italian is also a document dealing with relevance. A more complicated subsumption relationship is the one between concept C_1

Document
 $\sqcap (\exists \text{dealswith.InformationRetrieval})$
 $\sqcap (\exists \text{dealswith.TerminologicalLogics})$
 $\sqcap (\exists \text{dealswith.Relevance})$
 $\sqcap (\forall \text{dealswith.LogicBased})$
 $\sqcap (\exists \text{type.Article})$
 $\sqcap (\forall \text{author.Italian})$
 $\sqcap (\exists \text{author.Researcher})$
 $\sqcap (\exists \text{typesetwith.LaTeX})$ (5)

and concept C_2

Document
 $\sqcap (\exists \text{dealswith.}(\text{Relevance} \sqcap \text{LogicBased}))$
 $\sqcap (\exists \text{type.}(\text{Article} \sqcup \text{Book} \sqcup \text{TechnicalReport}))$ (6)

A document (*a*) of type article, (*b*) typeset in $\text{\LaTeX}$, (*c*) whose authors are Italian and at least one of them is a researcher, and (*d*) dealing with IR, TLs and relevance, all of these are logic-based, *is also* a document dealing with a logic-based approach to relevance, whose type is either article, or book or technical report. Therefore, if doc324 is described by means of concept C_1 , *i.e.* $C_1(\text{doc324})$ is in Σ , then doc324 would be retrieved by query C_2 . In fact, $\Sigma \models C_2(\text{doc324})$, which is a perfectly reasonable inference for IR purposes.

3.3 Some entailment relations

In the following we will restrict our attention only to the entailment relation, as subsumption can be defined in terms of entailment. In fact, note that all problems listed below, which are usually considered interesting with respect to TLs, are reducible to entailment.

Let Σ be a knowledge base.

Subsumption problem: Does a concept C subsume a concept D ?

Instance checking problem: Does Σ entail an assertion α ?

Retrieval problem: Let C be a concept. What is the set of all the individuals a such that $\Sigma \models C(a)$?

Therefore, the task of IR can be described in terms of the retrieval problem: each document has an unique identifier a which is an individual and DocKB is a set of assertions describing a set of documents. A query is a concept Q , *i.e.* a document description. The IR problem becomes: to retrieve all documents a which are instances of the query concept Q and, thus, satisfy the query concept Q .

Subsumption can be defined in terms of entailment as the following lemma shows.

Proposition 3.2 *Let C, D two concepts and a an individual. Then $C \sqsubseteq D$ if and only if $\{C(a)\} \models D(a)$.*

The realization problem can be reduced to the instance checking problem and the subsumption problem, while the retrieval problem can be reduced to the instance checking problem. It follows that we can concentrate our attention to the instance checking problem: *i.e.* the problem whether $\Sigma \approx \alpha$, without losing generality.

Consider the assertion

$$\begin{aligned}
 (\text{MultimediaDocument} \\
 & \sqcap (\forall \text{component}.\exists \text{dealswith.Pisa}) \\
 & \sqcap (\exists \text{component}.\left((\exists \text{doctype.Movie}) \sqcap \{\text{vt2}\}\right)) \\
 & \sqcap (\exists \text{component}.\left((\exists \text{doctype.Text}) \sqcap \{\text{t2}\}\right)) \\
 & \sqcap (\exists \text{component}.\{\text{ut2}\}) \\
 & \sqcap (\exists \text{component}.\{\text{ut3}\}))(\text{doc2})
 \end{aligned} \tag{7}$$

saying that `doc2` is a multimedia document with at least four components dealing with `Pisa`: a text `t2` a movie `vt2`, and `ut2` and `ut3` of undefined type. Note that the above assertion contains concepts of type $\{a\}$, usually called singleton concepts, not included in $(\mathcal{ALC})_4$, but present in MRTL. The meaning of $\{a\}$ is that of singleton concept: *i.e.* $t \in (\{a\})^I(d)$ iff $a^I = d$; $f \in (\{a\})^I(d)$ iff $a^I \neq d$.

Let us now consider the knowledge base Σ_1 defined as follows,

$$\begin{aligned}
 \Sigma_1 = \{ & (\text{Document} \\
 & \sqcap (\exists \text{dealswith.InformationRetrieval}) \\
 & \sqcap (\exists \text{dealswith.TerminologicalLogics}) \\
 & \sqcap (\exists \text{dealswith.Relevance}) \\
 & \sqcap (\forall \text{dealswith.LogicBased}) \\
 & \sqcap (\exists \text{type.Article}) \\
 & \sqcap (\forall \text{author.Italian}) \\
 & \sqcap (\exists \text{author.Researcher}) \\
 & \sqcap (\exists \text{typesetwith.LaTeX}))(\text{doc324}), \\
 & (\text{Document} \\
 & \sqcap (\exists \text{dealswith.InformationRetrieval}) \\
 & \sqcap (\exists \text{dealswith.TerminologicalLogics}) \\
 & \sqcap (\forall \text{author.Italian}))(\text{doc211}), \\
 & (\text{MultimediaDoc} \\
 & \sqcap (\forall \text{component}.\exists \text{dealswith.Pisa}) \\
 & \sqcap (\exists \text{component}.\left((\exists \text{doctype.Movie}) \sqcap \{\text{vt2}\}\right)) \\
 & \sqcap (\exists \text{component}.\left((\exists \text{doctype.Text}) \sqcap \{\text{t2}\}\right)) \\
 & \sqcap (\exists \text{component}.\{\text{ut2}\}) \\
 & \sqcap (\exists \text{component}.\{\text{ut3}\}))(\text{doc2}), \\
 & \text{author}(\text{doc211}, \text{Umberto}), \text{author}(\text{doc2}, \text{Umberto}), \\
 & \text{French}(\text{Umberto}) \\
 & \} \\
 & \cup \Sigma_2 \\
 & \cup \Sigma_3
 \end{aligned} \tag{8}$$

where

1. Σ_2 states that `French` and `Italian` are “disjoint” concepts⁴, *i.e.* for our purpose consider only

$$\Sigma_2 = \{\neg \text{Italian}(\text{Umberto})\} \tag{9}$$

2. Σ_3 states that multimedia document components are all of type movie or of type text, *i.e.* for our purpose consider only

$$\Sigma_3 = \{(\exists \text{doctype}.\left(\text{Movie} \sqcup \text{Text}\right))(\text{ut2})\} \tag{10}$$

Note that $\Sigma_2 \cup \Sigma_3$ could be part of the DKB.

⁴If someone is known to be French, then it is known not to be Italian.

3.3.1 About *modus ponens* on roles

The following entailment relationships are readily verified:

$$\Sigma_1 \approx (\text{Document} \sqcap (\exists \text{dealswith.}(\text{Relevance} \sqcap \text{LogicBased})) \sqcap (\exists \text{type.}(\text{Article} \sqcup \text{Book} \sqcup \text{TechnicalReport}))) (\text{doc324}) \quad (11)$$

as seen above, `doc324` is a document dealing with a logic-based approach to relevance, whose type is article, book or technical report;

$$\Sigma_1 \approx (\text{MultimediaDoc} \sqcap \exists \text{author.Italian}) (\text{doc2}) \quad (12)$$

i.e. from the fact that `Umberto` is the author of `doc211` and that all authors of `doc211` are Italian, it can be inferred that `Umberto` is Italian, hence that, `doc2` is a multimedia document with an Italian author;

$$\Sigma_1 \approx ((\exists \text{dealswith.Pisa}) \sqcap (\exists \text{doctype.Movie})) (\text{vt2}) \quad (13)$$

i.e. `vt2` is retrieved as a response to the query “retrieve all movies dealing with `Pisa`”. In fact, `vt2` is a component of `doc2` of type `movie`. All components of `doc2` are dealing with `Pisa`. Therefore, `vt2` is dealing with `Pisa`.

We claim that all three inferences are reasonable for IR purposes, in particular note that (12) and (13) are not trivial inferences. All three inferences are mainly based on the following key observation on our semantics, which in this case differs significantly from all other approaches. In fact, we allow *modus ponens on roles* (MPR, for short): *i.e.* for all concepts C and D , for any role R , and for all individuals a, b

$$\{(\forall R.C)(a), R(a, b)\} \approx C(b) \text{ and } \{(\forall R.C)(a), (\exists R.D)(a)\} \approx (\exists R.C \sqcap D)(a) \quad (14)$$

This kind of inference is not allowed by other four-valued TLs, as, for example, in [11]). The key difference lies into the semantics of the $\forall$ operator. Patel-Schneider’s condition in this case looks like

$$\begin{aligned} t \in (\forall R.C)^{\mathcal{I}}(d) & \quad \text{iff} \quad \forall e \in \Delta^{\mathcal{I}}, f \in R^{\mathcal{I}}(d, e) \text{ or } t \in C^{\mathcal{I}}(e) \\ f \in (\forall R.C)^{\mathcal{I}}(d) & \quad \text{iff} \quad \exists e \in \Delta^{\mathcal{I}}, t \in R^{\mathcal{I}}(d, e) \text{ and } f \in C^{\mathcal{I}}(e) \end{aligned}$$

It is easy to see that, according to this semantics, there exists an interpretation $\mathcal{I}$ which satisfies Σ_1 and such that both $t \in \text{author}^{\mathcal{I}}(\text{doc211}^{\mathcal{I}}, \text{Umberto}^{\mathcal{I}})$ and $f \in \text{author}^{\mathcal{I}}(\text{doc211}^{\mathcal{I}}, \text{Umberto}^{\mathcal{I}})$ without being $t \in \text{Italian}^{\mathcal{I}}(\text{Umberto}^{\mathcal{I}})$, and thus, Σ_1 does not entail $\text{Italian}(\text{Umberto})$ and thus, $\Sigma_1 \not\approx (\text{MultimediaDoc} \sqcap \exists \text{author.Italian})(\text{doc2})$.

We claim that MPR is very useful for IR and, in general, for real problems, and therefore we provided it in our framework.

3.3.2 About paradoxes of logical implication

In the following we will show what kind of inferences are not captured in our framework. The first two examples are about the so-called “paradoxes of logical implication” when reasoning on concepts and individuals.

First, note that the knowledge base Σ_1 has a “local inconsistency” in classical terms about `Umberto`’s nationality, without being totally inconsistent. In fact, we have

$$\Sigma_1 \approx (\text{Italian} \sqcap \neg \text{Italian})(\text{Umberto}) \quad (15)$$

and thus,

$$\Sigma_1 \approx (\text{MultimediaDoc} \sqcap \exists \text{author.Italian})(\text{doc2}) \quad (16)$$

and

$$\Sigma_1 \approx (\text{MultimediaDoc} \sqcap \exists \text{author.}\neg \text{Italian})(\text{doc2}) \quad (17)$$

This means that `doc2` is retrieved in both cases since in Σ_1 there is *evidence* to the fact that `doc2` is an instance off the queries $\text{MultimediaDoc} \sqcap \exists \text{author. Italian}$ and $\text{MultimediaDoc} \sqcap \exists \text{author. } \neg \text{Italian}$. On the other hand, in the document base Σ_1 there is nothing about `vt2`'s authors. Therefore, as we would expect,

$$\Sigma_1 \not\models ((\exists \text{doctype. Text}) \sqcap (\forall \text{author. \{Umberto\}}))(\text{vt2}) \quad (18)$$

In two-valued semantics, since Σ_1 is inconsistent

$$\Sigma_1 \models_2 ((\exists \text{doctype. Text}) \sqcap (\forall \text{author. \{Umberto\}}))(\text{vt2}) \quad (19)$$

Clearly, this last kind of inference is not acceptable in IR: there is nothing in Σ_1 about `vt2` which is relevant to the query $(\exists \text{doctype. Text}) \sqcap (\forall \text{author. \{Umberto\}})$.

This example shows, in a simple way, one of the advantages of a four-valued semantics: inconsistent knowledge bases (from a two-valued semantics point of view) do not entail everything.

Dually, concepts, whose extensions are always the entire domain of an interpretation, are not necessarily entailed by every knowledge base. In fact,

$$\Sigma_1 \approx (\exists \text{doctype. (Movie } \sqcup \text{ Text)})(\text{ut2}) \quad (20)$$

and

$$\Sigma_1 \not\models (\forall \text{doctype. (Movie } \sqcup \neg \text{Movie)})(\text{ut3}) \quad (21)$$

which we feel both correct. In fact, (20) follows directly from Σ_1 , whereas (21) holds since there is an interpretation $\mathcal{I}$, such that $e \in \Delta^{\mathcal{I}}$ and $t \in \text{doctype}^{\mathcal{I}}(\text{ut3}, e)$ and $\text{Movie}^{\mathcal{I}}(e) = \emptyset$. This is a state of affairs which models the fact that in Σ_1 there is no evidence about `ut3`'s document type, *whatever* it could be.

Whereas, with respect to two-valued semantics, we have

$$\Sigma_1 \models_2 (\forall \text{doctype. (Movie } \sqcup \neg \text{Movie)})(\text{ut3}) \quad (22)$$

To our opinion, missing this last kind of inference is important for IR purposes, since we want relevance of the premise to the conclusion.

3.3.3 About reasoning by cases

Finally, *case reasoning* does not work within our semantics. Consider the following knowledge base

$$\Sigma_4 = \{ \text{Document}(\text{doc74}), \text{component}(\text{doc74}, \text{t741}), \text{component}(\text{doc74}, \text{t742}), \\ \text{translation}(\text{t741}, \text{t742}), \text{translation}(\text{t742}, \text{t743}), \\ \text{ItalianText}(\text{t741}), \neg \text{ItalianText}(\text{t743}), \\ (\exists \text{dealswith. Galileo})(\text{doc74}) \\ \}$$

The meaning of Σ_4 is: `doc74` is a document, dealing with `Galileo`, with two components, `t741`, which is an Italian text, and `t742`. `t743` is not an Italian text. Moreover, `t742` is a translation of `t741` and `t743` is a translation of `t742`.

Consider now the assertion

$$\alpha = ((\exists \text{dealswith. Galileo}) \sqcap \\ (\exists \text{component. (ItalianText } \sqcap \exists \text{translation. } \neg \text{ItalianText)})))(\text{doc74})$$

Let $\mathcal{I}$ be an interpretation which satisfies Σ_4 such that $\text{ItalianText}^{\mathcal{I}}(\text{t742}^{\mathcal{I}}) = \emptyset$. It is easy to see that such an interpretation exists. Now, it follows that $\mathcal{I}$ does not satisfy α and, thus,

$$\Sigma_4 \not\models \alpha \quad (23)$$

whereas, perhaps surprisingly,

$$\Sigma_4 \models_2 \alpha$$

At first sight it would seem that $\Sigma_4 \not\models_2 \alpha$: since t742 and t741 are the only known components of doc74 , and $\Sigma_4 \not\models_2 (\text{ItalianText} \sqcap \exists \text{translation} \neg \text{ItalianText})(\text{t741})$ and $\Sigma_4 \not\models_2 (\text{ItalianText} \sqcap \exists \text{translation} \neg \text{ItalianText})(\text{t742})$. However, reasoning by case analysis, one realizes that $\Sigma_4 \models_2 \alpha$. Consider any two-valued interpretation $\mathcal{I}$ which satisfies Σ_4 . In this interpretation either $\text{ItalianText}(\text{t742})$ is true or $\text{ItalianText}(\text{t742})$ is false. In the former case doc74 has t742 as a component which is an Italian text and whose not Italian text translation is t743 . In the latter case doc74 has t741 as a component which is an Italian text and whose not Italian text translation is t742 . Therefore, in both cases α is true in $\mathcal{I}$ and, thus, $\Sigma_4 \models_2 \alpha$. Note that $\Sigma_4 \not\models \alpha$ since it could be the case $\text{ItalianText}^{\mathcal{I}}(\text{t742}^{\mathcal{I}}) = \emptyset$. That is, we are uncertain about t742 's text language. Therefore, it is easy to see that if we know something about t742 's text language then α holds: *i.e.*

$$\Sigma_4 \cup \{(\text{ItalianText} \sqcup \neg \text{ItalianText})(\text{t742})\} \models \alpha \quad (24)$$

We feel that (23) could be acceptable in the light of (24): *i.e.* since we have no relevant information about t742 's text language (in fact, we have no information about t742 's text language) case reasoning no longer holds.

To sum up, what kind of relevance relation is captured by $\approx$? Roughly speaking, a knowledge base Σ entails everything that is *explicitly known*, *i.e.* that is in the transitive closure of Σ by means of MPR and the operators $\sqcap, \sqcup, \neg, \exists$, as (11), (12), (13), (16) and (17) demonstrate. All other inferences are left out, as (18), (21) and (23) show. More precisely, in order $\Sigma \approx \alpha$ to hold, the structural components of α must have an analogue in Σ , modulo MPR.⁵ We can assert therefore, that this notion of entailment ($\approx$) seems to be arguably closer to the one of IR than the one of logical implication of classical logic, as, for example, $\models_2$ (see, for example, (20) versus (21)). Clearly, the framework needs to be extended, as described in Section 1, including, thus, probabilities, rules, etc.

4 Conclusions

Recently, TLs have been proposed as a model for Information Retrieval. In this context this amounts to embodying a notion of relevance in the logical implication relation of the chosen TL. Among the many possible readings of the term “relevance”, the one captured by relevance logic, can be chosen as a promising source of inspiration to the end of incorporating a logic-based form of relevance in the inference mechanism of TLs and, thus, this amounts to adopting a four-valued semantics for TLs.

In this paper we have presented a Relevant TL, $(\mathcal{ALC})_4$, such that its inference mechanism could be considered as suitable core logic for IR purposes. Moreover, we have shown that the defined entailment relation captures a close structural relationship between a knowledge base and a query and, thus, it could arguably be a good theoretical and practical bases for a logic-based approach to IR.

Of course, in order to perform automated reasoning in this logic, there is a need of an inference algorithm (which is described in [21]).

Of course, the entailment relation defined in this paper is far from being a satisfactory relation of “relevance” suitable for IR purposes. But, it can be seen as the core of a logic-based approach to IR, towards a satisfactory logic for IR. In order to obtain such a satisfactory logic, at least the following aspects (and not limited to) must be worked out and needs to be included:

1. *Complexity analysis* of our calculus. To this purpose, the literature has shown that four-valued logics have been proven to have a generally better computational behaviour than their two-valued analogues;

⁵In [21] there is a formal proof about this statement.

2. *Extension to constructs of the $\mathcal{AL}$ -family.* Actually we considered only the TL $(\mathcal{ALC})_4$. The aim is to extend our calculus to MIRTL, and in general, to other $\mathcal{AL}$ languages. The modularity of the calculus presented in [21] facilitates this task;
3. *Probabilistic extension.* Probabilistic versions of TLs[18, 22] need to be investigated as a means of making explicit various sources of uncertainty, such as uncertainty related to domain knowledge and uncertainty related to automatic document representation, which is typical in IR;
4. *Concrete domain.* Incorporating kinds of data types of the underlying concrete domain [2] as “string”, “integer”, “link”, etc.;
5. *Rule language.* Including rules for the DocKB module, as, for example, in [7];
6. Closed World Assumption, Closed Domain Reasoning, etc.. Close world reasoning and closed domain reasoning are certainly suitable for IR purposes, as they are close to usual databases reasoning [6, 15, 16].

References

- [1] Alan R. Anderson and Nuel D. Belnap. *Entailment - the logic of relevance and necessity*. Princeton University Press, Princeton, NJ, 1975.
- [2] Franz Baader and Philipp Hanschke. A schema for integrating concrete domains into concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 452–457, Sydney, 1991.
- [3] Nuel D. Belnap. A useful four-valued logic. In Gunnar Epstein and J. Michael Dunn, editors, *Modern uses of multiple-valued logic*, pages 8–37. Reidel, Dordrecht, NL, 1977.
- [4] Paolo Buongarzone, Carlo Meghini, Rossella Salis, Fabrizio Sebastiani, and Umberto Straccia. Logical and computational properties of the description logic MIRTL. In Borgida A., Lenzerini M., Nardi D., and Nebel B., editors, *International Workshop on Description Logics*, pages 80–84, Rome, Italy, 1995.
- [5] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. Tractable concept languages. In *Proceedings of IJCAI-91, 12th International Joint Conference on Artificial Intelligence*, pages 458–463, Sidney, Australia, 1991.
- [6] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, and Andrea Schaerf. Adding epistemic operators to concept languages. In *Proc. of the 3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-92)*, pages 342–353. Morgan Kaufmann, Los Altos, 1992.
- [7] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. A hybrid system integrating datalog and concept languages. In *Proc. of the 2nd Conf. of the Italian Association for Artificial Intelligence (AI*IA-91)*, number 549 in Lecture Notes In Artificial Intelligence. Springer-Verlag, 1991. An extended version appeared also in the Working Notes of the AAAI Fall Symposium “Principles of Hybrid Reasoning”.
- [8] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of AAAI-84, 4th Conference of the American Association for Artificial Intelligence*, pages 198–202, Austin, TX, 1984.
- [9] Carlo Meghini, Fabrizio Sebastiani, Umberto Straccia, and Costantino Thanos. A model of information retrieval based on a terminological logic. In *Proceedings of SIGIR-93, 16th International Conference on Research and Development in Information Retrieval*, pages 298–307, Pittsburgh, PA, 1993.
- [10] Bernhard Nebel. *Reasoning and revision in hybrid representation systems*. Springer, Heidelberg, FRG, 1990.
- [11] Peter F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38:319–351, 1989.
- [12] Peter F. Patel-Schneider, Ronald J. Brachman, and Hector J. Levesque. ARGON: Knowledge representation meets information retrieval. In *Proc. of the 1st Conf. on Artificial Intelligence Applications*, 1984. Also available as Fairchild Technical Report 654 and FLAIR Technical Report 29.

- [13] Peter F. Patel-Schneider, D. L. McGuinness, Ronald J. Brachman, Lori Alperin Resnick, and A. Borgida. The CLASSIC knowledge representatin system: Guiding principles and implementation rational. *SIGART Bulletin*, 2(3):108–113, 1991.
- [14] C. Peltason. The BACK system – an overview. *SIGART Bulletin*, 2(3):114–119, 1991.
- [15] R. Reiter. On asking what a database knows. In J. W. Lloyd, editor, *Symposium on Computational Logics*, pages 96–113. Springer-Verlag, ESPRIT Basic Research Action Series, 1990.
- [16] Raymond Reiter. On closed world data bases. In Hervé Gallaire and Jack Minker, editors, *Logic and data bases*, pages 55–76. Plenum Press, New York, NY, 1978.
- [17] Manfred Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In *Proceedings of KR-89, 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 421–431, Toronto, Ontario, 1989.
- [18] Fabrizio Sebastiani. A probabilistic terminological logic for modelling information retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 122–130, Dublin, IRL, 1994. Published by Springer Verlag, Heidelberg, FRG.
- [19] Fabrizio Sebastiani and Umberto Straccia. A computationally tractable terminological logic. In *Proceedings of SCAI-91, 3rd Scandinavian Conference on Artificial Intelligence*, pages 307–315, Roskilde, Denmark, 1991.
- [20] Umberto Straccia. Default inheritance reasoning in hybrid KL-ONE-style logics. In *Proceedings of IJCAI-93, 13th International Joint Conference on Artificial Intelligence*, pages 676–681, Chambéry, France, 1993.
- [21] Umberto Straccia. Four-valued terminological logics for information retrieval. Technical report, Istituto di Elaborazione dell’Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy, 1995. FERMI Report 7/95.
- [22] John Yen. Generalizing term subsumption languages to fuzzy logic. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 472–477, Sydney, Australia, 1991.