Terminological Knowledge Representation: A Proposal for a Terminological Logic

Franz Baader, Hans-Jürgen Bürckert, Jochen Heinsohn,
Bernhard Hollunder, Jürgen Müller, Bernhard Nebel,
Werner Nutt, Hans-Jürgen Profitlich
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Postfach 2080, W-6750 Kaiserslautern
Stuhlsatzenhausweg 3, W-6600 Saarbrücken 11
Germany

Abstract

This paper contains a proposal for a terminological logic. The formalisms for representing knowledge as well as the needed inferences are described.

1 Introduction

An important aspect of intelligence is the use of existing knowledge. In order to realize this in AI-Systems we need both adequate methods to represent knowledge and effective procedures to retrieve and reuse the needed knowledge. One of the basic mechanisms of human knowledge representation and processing is the division of the world into classes or concepts ("find the right pigeonhole") which usually are given with a hierarchical structure.

Let us consider some knowledge base about families and relationships. We have to deal with persons which are of sex male or female. We have parents, mothers, fathers etc. A verbal description of this knowledge might be as follows:

- *Persons* are of *sex Male* or *Female*.
- *Woman* is a *Person* with *sex Female*.
- *Man* is a *Person* with *sex Male*.
- *Parents* are defined as *Persons* which have some *child* (which is also a *Person*).
- *Mothers* are defined to be *Parents* with *sex Female*.
- *Fathers* are defined to be *Parents* with *sex Male*.
- *Mother_with_many_children* is defined as *Mother* with at least three *children*.

We also have individuals (or objects) which are instances of concepts. For example,

- *John* is a *Father*.

- *Tom* is a *child* of *John*.
- *Mary* is a *Woman*.

Now every knowledge representation system should offer a couple of services that allow to arrange, manage, modify or retrieve information of the above kind. It should be able to answer the following questions:

- Is an introduced concept defined in a meaningful way at all (or does it denote the empty concept in all worlds) ? (*satisfiability*)
- Is a concept more general than another one ? (*subsumption*)
- Where exactly is the concept situated in a concept hierarchy ? (*classification*)
- Is the represented knowledge consistent ? (*consistency*)
- What facts are deducible from the knowledge ? (*instantiation*)
- Which are the concepts an object is instance of ? (*realization*)
- Which are the instances of a given concept ? (*retrieval*)

Building such a system we are confronted with the following questions:

1. How can the above properties been found out at all ?

And then, if we know procedures that might do this:

2. How can we find out, whether the procedures really do what they should do ?
3. How efficient are these procedures ?
4. How efficient may an optimal procedure for the problem be ?

Terminological logics based on concept description languages like KL-ONE [BS85] are such formalisms that make classification, description of relations among the classes and especially their hierarchical structure possible. However, concept description languages are not only one among a lot of possibilities, but meanwhile they offer compared to other KR-formalisms some fundamental advantages:

- There is a well understood *declarative* semantics.
  This means that the meaning of the constructs is not given operationally, e.g. by the implementation ("John is a father", because my system answers to the question "What is John?" just "father"), but the meaning is given by its description and its models ("John is a father", because he is a father in all models—in all worlds—where the description suits to.)
- There is a characterization of the tasks of the KR-systems by the declarative semantics.
- There is a number of procedures and algorithms that realize these tasks, and their properties are well investigated now. Important properties are
  1. Correctness
     (If the system answers "John is a father", then John is a father within the meaning of the semantics—that is in all suitable worlds.)

2. Completeness
   (The system answers "John is a father", if John is a father within the meaning of the semantics.)
3. Decidability, Complexity
   (Are the services decidable at all, and how fast are they executable ?)

If we want to design a knowledge base, we first need a formal language that we can use. In the following we will present a proposal for a terminological language in both abstract form and machine readable form (LISP notation). As a kernel, our language contains all the constructs provided by $\mathcal{ALC}$ [SS88] and some additional operators which (sometimes?) can be translated into $\mathcal{ALCFNR}$ [HN90].

2 Symbols

The terminological language is based on the following primitives, the symbols of the alphabet:

- Concept names: CN
- Role names: RN
- Attribute names: AN
- Individual names: IN
- Object names: ON

Examples with respect to our introductory example are: `Person`, `Woman`, `Man`, `Parent` are concept names, `child` is a role name, `sex` is an attribute name, `Male` and `Female` are individual names, and `John` and `Mary` are objects names.

With this primitives we are allowed to form more complex expressions as specified in the next two sections:

- Concept expressions: C
- Role expressions: R
- Attribute expressions: A

The meaning of these is given by interpretations $\mathcal{I}$. They consist of a set $\Delta^\mathcal{I}$—the domain—and an interpretation function $\cdot^\mathcal{I}$, that assigns a set

$$CN^\mathcal{I} \subseteq \Delta^\mathcal{I}$$

to each concept name CN, a set-valued function (or equivalently a binary relation)

$$RN^\mathcal{I} : \Delta^\mathcal{I} \longrightarrow 2^{\Delta^\mathcal{I}} \qquad (RN^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I})$$

to each role name RN, a single-valued partial function

$$AN^\mathcal{I} : dom\, AN^\mathcal{I} \longrightarrow \Delta^\mathcal{I},$$

where $dom\, AN^\mathcal{I} \subseteq \Delta^\mathcal{I}$, to each attribute name AN, and an element

$$I^\mathcal{I} \in \Delta^\mathcal{I}$$

to each individual name IN and object name ON. We assume that different individuals and objects denote different elements in every interpretation. This property is called *unique name assumption* and is usually assumed in the database world.

3 Concept Forming Operators

Besides the concept, role, and attribute names our alphabet includes a number of operators that permit to compose more complex concepts, roles, and attributes. We allow for the following concept forming operators:

Concrete Form	**Abstract Form**	**Semantics**
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
(and $C_1 \ldots C_n$)	$C_1 \sqcap \ldots \sqcap C_n$	$C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
(or $C_1 \ldots C_n$)	$C_1 \sqcup \ldots \sqcup C_n$	$C_1^{\mathcal{I}} \cup \ldots \cup C_n^{\mathcal{I}}$
(not C)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
(all R C)	$\forall R : C$	$\{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \subseteq C^{\mathcal{I}}\}$
(some R)	$\exists R$	$\{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \neq \emptyset\}$
(some R C)	$\exists R : C$	$\{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \cap C^{\mathcal{I}} \neq \emptyset\}$
(atleast n R)	$\geq nR$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d)\rvert \geq n\}$
(atmost n R)	$\leq nR$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d)\rvert \leq n\}$
(exact n R)	nR	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d)\rvert = n\}$
(atleast n R C)	$\geq nR : C$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d) \cap C^{\mathcal{I}}\rvert \geq n\}$
(atmost n R C)	$\leq nR : C$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d) \cap C^{\mathcal{I}}\rvert \leq n\}$
(exact n R C)	$nR : C$	$\{d \in \Delta^{\mathcal{I}} \mid \lvert R^{\mathcal{I}}(d) \cap C^{\mathcal{I}}\rvert = n\}$
(eq R_1 R_2)	$R_1 = R_2$	$\{d \in \Delta^{\mathcal{I}} \mid R_1^{\mathcal{I}}(d) = R_2^{\mathcal{I}}(d)\}$
(neq R_1 R_2)	$R_1 \neq R_2$	$\{d \in \Delta^{\mathcal{I}} \mid R_1^{\mathcal{I}}(d) \neq R_2^{\mathcal{I}}(d)\}$
(subset R_1 R_2)	$R_1 \subseteq R_2$	$\{d \in \Delta^{\mathcal{I}} \mid R_1^{\mathcal{I}}(d) \subseteq R_2^{\mathcal{I}}(d)\}$
(in A C)	$A : C$	$\{d \in dom\, A^{\mathcal{I}} \mid A^{\mathcal{I}}(d) \in C^{\mathcal{I}}\}$
(is A IN)	$A : IN$	$\{d \in dom\, A^{\mathcal{I}} \mid A^{\mathcal{I}}(d) = IN^{\mathcal{I}}\}$
(eq A_1 A_2)	$A_1 = A_2$	$\{d \in \Delta^{\mathcal{I}} \mid A_1^{\mathcal{I}}(d) = A_2^{\mathcal{I}}(d)\}$
(neq A_1 A_2)	$A_1 \neq A_2$	$\{d \in \Delta^{\mathcal{I}} \mid A_1^{\mathcal{I}}(d) \neq A_2^{\mathcal{I}}(d)\}$
(subset A_1 A_2)	$A_1 \subseteq A_2$	$\{d \in \Delta^{\mathcal{I}} \mid d \in dom\, A_1^{\mathcal{I}} \Rightarrow d \in dom\, A_2^{\mathcal{I}} \wedge A_1^{\mathcal{I}}(d) = A_2^{\mathcal{I}}(d)\}$
(oneof $IN_1 \ldots IN_n$)	$\{IN_1, \ldots, IN_n\}$	$\{IN_1^{\mathcal{I}}, \ldots, IN_n^{\mathcal{I}}\}$

Examples: The concept *mother* can be described as

$$\texttt{Person} \sqcap (\texttt{sex} : \texttt{Female});$$

Mother_with_many_children can be described as

$$\texttt{Mother} \sqcap (\geq 3\texttt{child} : \texttt{Person});$$

Father_with_sons_only can be described as

$$\texttt{Parent} \sqcap (\texttt{sex} : \texttt{Male}) \sqcap (\texttt{child} = \texttt{son}).$$

Please note that the semantics of $A_1 = A_2$ and $A_1 \neq A_2$ for attributes is defined analogously to the semantics of $R_1 = R_2$ and $R_1 \neq R_2$ for roles. In particular, $A_1^{\mathcal{I}}(d) = A_2^{\mathcal{I}}(d)$ also covers the case where both values are undefined. This differs from the definitions used in [HN90] and computational linguistics in that we do not require that both attributes have to be defined on d. However, these definitions can be expressed using our constructs:

$$(A_1 = A_2) \sqcap (A_1 : \top) \sqcap (A_2 : \top)$$
$$(A_1 \neq A_2) \sqcap (A_1 : \top) \sqcap (A_2 : \top)$$

As abbreviations for these two expressions we propose $A_1 \overset{\downarrow}{=} A_2$ and $A_1 \overset{\downarrow}{\neq} A_2$, where the downarrow is meant to express the condition "is defined".

4 Role Forming and Attribute Forming Operators

Similar as for concepts our terminological logic provides a variety of role forming and attribute forming operators:

Concrete Form	**Abstract Form**	**Semantics**
(and $R_1 \ldots R_n$)	$R_1 \sqcap \ldots \sqcap R_n$	$R_1^{\mathcal{I}} \cap \ldots \cap R_n^{\mathcal{I}}$
(or $R_1 \ldots R_n$)	$R_1 \sqcup \ldots \sqcup R_n$	$R_1^{\mathcal{I}} \cup \ldots \cup R_n^{\mathcal{I}}$
(not R)	$\neg R$	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$
identity	id	$\{(d,d) \mid d \in \Delta^{\mathcal{I}}\}$
(inverse R)	R^{-1}	$\{(d,d') \mid (d',d) \in R^{\mathcal{I}}\}$
(restrict R C)	$R \mid C$	$\{(d,d') \in R^{\mathcal{I}} \mid d' \in C^{\mathcal{I}}\}$
(compose $R_1 \ldots R_n$)	$R_1 \circ \ldots \circ R_n$	$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}}$
(domrange C_1 C_2)	$C_1 \times C_2$	$C_1^{\mathcal{I}} \times C_2^{\mathcal{I}}$
(trans R)	R^+	$\bigcup_{n \geq 1} (R^{\mathcal{I}})^n$
(transref R)	R^*	$\bigcup_{n \geq 0} (R^{\mathcal{I}})^n$
(inverse A)	A^{-1}	$\{(A^{\mathcal{I}}(d), d) \mid d \in dom\, A^{\mathcal{I}}\}$
(restrict A C)	$A \mid C$	$A^{\mathcal{I}} \mid_{C^{\mathcal{I}}}$
(compose $A_1 \ldots A_n$)	$A_1 \circ \ldots \circ A_n$	$A_1^{\mathcal{I}} \circ \ldots \circ A_n^{\mathcal{I}}$

Notice that the inverse of an attribute is a role, but in general not an attribute. The range restriction $R \mid C$ can be seen as an abbreviation for $R \cap (\top \times C)$. Similarly, a domain restriction on the role R could be expressed as $R \cap (C \times \top)$.

Examples: The role *daughter* can be defined as

$$\texttt{female\_relative} \sqcap \texttt{child};$$

the role *successor* can be defined as

$$(\text{inverse } \texttt{predecessor}).$$

5 Terminological Axioms

The terminological axioms (definitions, specializations, and restrictions) are used to specify the knowledge about the world or a part of the world. A set of terminological axioms specifies a terminology $\mathcal{T}$. It selects from all possible interpretations of the language the models of $\mathcal{T}$, i.e., the interpretations satisfying the axioms of $\mathcal{T}$ as described below.

Concrete Form	**Abstract Form**	**Semantics**
(defconcept CN C)	$CN \doteq C$	$CN^{\mathcal{I}} = C^{\mathcal{I}}$
(defrole RN R)	$RN \doteq R$	$RN^{\mathcal{I}} = R^{\mathcal{I}}$
(defattribute AN A)	$AN \doteq A$	$AN^{\mathcal{I}} = A^{\mathcal{I}}$
(defprimconcept CN C)	$CN \sqsubseteq C$	$CN^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
(defprimrole RN R)	$RN \sqsubseteq R$	$RN^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
(defprimattribute AN R)	$AN \sqsubseteq R$	$AN^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
(defdisjoint $CN_1 \ldots CN_n$)	$CN_1 \parallel \ldots \parallel CN_n$	$CN_i^{\mathcal{I}} \cap CN_j^{\mathcal{I}} = \emptyset, i \neq j$

Usually the following restrictions are imposed on terminologies. Any name should appear only once as a left hand side of an axioms, and disjointness axioms should only contain names of primitive concepts.

An alternative way of expressing disjointness could be the use of disjointness groups in the definition of primitive concepts. In this case the introduction of primitive concepts would be of the form $CN \sqsubseteq C/g_1, \ldots, g_n$, where the g_i's are names of disjointness groups. Two different primitive concepts must have disjoint extensions if a disjointness group occurs in the definitions of both concepts.

In the abstract form there is no syntactic distinction between definitions of concepts, roles, and attributes. One possibility to distinguish between concepts, roles, and attributes could be to group the definitions, as done in the following example.

Example (our introductory example in formal notation):

Attributes:

$\texttt{sex} \sqsubseteq \top \times \top$

Roles:

$\texttt{child} \sqsubseteq \top \times \top$

Concepts:

$\texttt{Person} \sqsubseteq \texttt{sex} : \{\texttt{Male}, \texttt{Female}\}$

$\texttt{Woman} \sqsubseteq \texttt{Person} \sqcap \texttt{sex} : \texttt{Female}$

$\texttt{Man} \sqsubseteq \texttt{Person} \sqcap \texttt{sex} : \texttt{Male}$

$\texttt{Parent} \doteq \texttt{Person} \sqcap \exists\texttt{child} : \texttt{Person} \sqcap \forall\texttt{child} : \texttt{Person}$

$\texttt{Mother} \doteq \texttt{Parent} \sqcap \texttt{sex} : \texttt{Female}$

$\texttt{Father} \doteq \texttt{Parent} \sqcap \texttt{sex} : \texttt{Male}$

$\texttt{Mother\_with\_many\_children} \doteq \texttt{Mother} \sqcap \geq 3\texttt{child} : \texttt{Person}$

$\texttt{Father\_with\_sons\_only} \doteq \texttt{Father} \sqcap (\texttt{child} = \texttt{son})$.

Please note that the disjointness axiom $\texttt{Woman} \parallel \texttt{Man}$ would be redundant since disjointness of woman and man is a consequence of the fact that sex is an attribute and male and female are individuals which are interpreted with unique name assumption.

6 Assertional Axioms

In order to fill our world with objects we allow for assertional axioms which have the following forms.

Concrete Form	**Abstract Form**	**Semantics**
$(C\ ON)$	$ON \in C$	$ON^{\mathcal{I}} \in C^{\mathcal{I}}$
$(R\ ON\ ON')$	$\langle ON, ON' \rangle \in R$	$(ON^{\mathcal{I}}, ON'^{\mathcal{I}}) \in R^{\mathcal{I}}$
$(A\ ON\ ON')$	$\langle ON, ON' \rangle \in A$	$ON^{\mathcal{I}} \in dom\, A^{\mathcal{I}} \wedge A^{\mathcal{I}}(ON^{\mathcal{I}}) = ON'^{\mathcal{I}}$

Examples:

$\texttt{John} \in \texttt{Father}$

$\texttt{Mary} \in \texttt{Woman}$

$\langle \texttt{John}, \texttt{Tom} \rangle \in \texttt{child}$.

7 Services

Now we are able to give a formal specification of the services mentioned in the introduction.

1. Satisfiability of a concept C in a terminology $\mathcal{T}$:
   Does there exist a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$?
   ($\texttt{Man} \sqcap \texttt{Woman}$ is not satisfiable.)

2. Subsumption within a terminology $\mathcal{T}$:
   $C \sqsubseteq_{\mathcal{T}} D$ iff in all models $\mathcal{I}$ of $\mathcal{T}$: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
   (e.g. $\texttt{Mother} \sqsubseteq_{\mathcal{T}} \texttt{Woman}$).

3. Equivalence of concepts within a terminology $\mathcal{T}$:
   $C \approx_{\mathcal{T}} D$ iff in all models $\mathcal{I}$ of $\mathcal{T}$: $C^{\mathcal{I}} = D^{\mathcal{I}}$

4. Classification of C in $\mathcal{T}$:
   For a given concept C, find all minimal (w.r.t. the subsumption relation) concepts D in $\mathcal{T}$ such that $C \sqsubseteq_{\mathcal{T}} D$.

5. Find the smallest binary relation on the concepts in $\mathcal{T}$ such that its transitive closure is the subsumption relation (modulo $\approx_{\mathcal{T}}$).

6. Consistency of the represented knowledge.
   Does there exist a model $\mathcal{I}$ of the terminological and assertional axioms ?

7. What facts are deducible from the knowledge ?
   A fact α is deducible from the knowledge iff all models of the terminological and assertional axioms satisfy α. In particular, if α is of the form $ON \in C$, then we talk about instantiation.

8. Realization.
   Given an object *ON* occurring in an assertional axiom. Which are most specific concepts of $\mathcal{T}$ w.r.t. the subsumption relation *ON* is instance of?

9. Retrieval.
   Given an concept C. Which objects occurring in the assertional axioms are instances of C ?

With this formalization of our services we can develop procedures or algorithms for the services and prove their correctness, completeness, complexity, decidability; see for example [SS88, HN90, Hol90, Baa91, DLNN91a, DHLMNN91, HB91, BH91, DLNN91b].

References

[Baa91] F. Baader. *Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles.* DFKI Research Report RR-90-13, DFKI, Postfach 2080, W-6750 Kaiserslautern, Germany. To appear in *Proceedings of IJCAI '91.*

[BH91] F. Baader, P. Hanschke. *A Schema for Integrating Concrete Domains into Concept Languages.* DFKI Research Report RR-91-10, DFKI, Postfach 2080, W-6750 Kaiserslautern, Germany. To appear in *Proceedings of IJCAI '91.*

[BS85] R. J. Brachman, J. G. Schmolze. "An Overview of the KL-ONE knowledge representation system." *Cognitive Science*, 9(2):171-216, April 1985.

[DHLMNN91] F. Donini, B. Hollunder, M. Lenzerini, A. Marchetti Spaccamela, D. Nardi, W. Nutt. *The Complexity of Existential Quantification in Concept Languages.* DFKI Research Report RR-91-02, DFKI, Postfach 2080, W-6750 Kaiserslautern Germany.

[DLNN91a] F. Donini, M. Lenzerini, D. Nardi, W. Nutt. "The Complexity of Concept Languages." In J. A. Allan, R. Fikes, E. Sandewall (editors), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, Mas., 1991.

[DLNN91b] F. Donini, M. Lenzerini, D. Nardi, W. Nutt. "Tractable Concept Languages." To appear in *Proceedings of IJCAI '91.*

[Hol90] B. Hollunder. "Hybrid Inferences in KL-ONE-based Knowledge Representation Systems." In *Proceedings of the 14th German Workshop on Artificial Intelligence*, pp. 38–47, Eringerfeld, Germany, 1990.

[HB91] B. Hollunder, F. Baader. "Qualifying Number Restrictions in Concept Languages." In J. A. Allan, R. Fikes, E. Sandewall (editors), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, Mas., 1991.

[HN90] B. Hollunder, W. Nutt. *Subsumption Algorithms for Concept Languages.* DFKI Research Report RR-90-04, DFKI, Postfach 2080, W-6750 Kaiserslautern, Germany.

[SS88] M. Schmidt-Schauß, G. Smolka. "Attributive Concept Descriptions with Unions and Complements". In *Artificial Intelligence,* 47, 1991.