Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-90-06

Hybrid Inferences in KL-ONE-based Knowledge Representation Systems

Bernhard Hollunder

May 1990

Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Philips, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❑ Intelligent Engineering Systems
- ❑ Intelligent User Interfaces
- ❑ Intelligent Communication Networks
- ❑ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

Hybrid Inferences in KL-ONE-based Knowledge Representation Systems

Bernhard Hollunder

DFKI-RR-90-06

A short version of this paper will be published in the Proceedings of the 14th German Workshop on Artificial Intelligence, Springer-Verlag, 1990.

Hybrid Inferences in KL-ONE-based Knowledge Representation Systems

Bernhard Hollunder
German Research Center for Artificial Intelligence
Postfach 2080
D-6750 Kaiserslautern, West Germany
e-mail: hollunde@informatik.uni-kl.de

Abstract

We investigate algorithms for hybrid inferences in KL-ONE-based knowledge representation systems. Those systems employ two kinds of formalisms: the terminological and the assertional formalism. The terminological formalism consists of a concept description language to define concepts and relations between concepts for describing a terminology. On the other hand, the assertional formalism allows to introduce objects, which are instances of concepts and relations of a terminology. We present algorithms for hybrid inferences such as

- determining subsumption between concepts
- checking the consistency of such a knowledge base
- computing the most specialized concepts an object is instance of
- computing all objects that are instances of a certain concept.

Contents

1 Introduction

Knowledge representation systems of the KL-ONE family, for example [BS85, BPGL85, KBR86, MB87, NvL88, Pat84] employ two kinds of formalisms for the representation of knowledge. The terminological formalism consists of a concept description language for the definition of concepts and relations between concepts to describe a TBox (terminology). These concept descriptions are built out of two kinds of primitive symbols, concepts and roles. On the other hand, the assertional formalism allows to state that individuals are instances of concepts or roles for the description of an ABox. We give a unified declarative semantics in a Tarski style for the two formalisms that allows to conceive them as sublanguages of predicate logic [BL84]. An interpretation interprets (*i*) concepts as subsets of the domain, (*ii*) roles as binary relations over the domain, and (*iii*) individuals as elements of the domain.

To give an example, assume that Person and Female are concepts, Child is a role, and Mary and Tom are individuals. If connectives like concept conjunction and complement are present in the concept description language, then one can describe "persons that are not female" by $\mathsf{Person} \sqcap \neg\mathsf{Female}$. Since concepts are interpreted as sets, we interpret concept conjunction and complement as set intersection and complement. Almost all concept description languages provide restrictions on roles. *Value restrictions* can be used for instance to describe "all individuals for which all children are female" by the expression $\forall \mathsf{Child}.\mathsf{Female}$. *Number restrictions* allow for instance to describe "individuals having at least (most) two children" by the expressions $(\geq 2\ \mathsf{Child})$ and $(\leq 2\ \mathsf{Child})$, respectively. Furthermore, we can build more complex concept descriptions like $\mathsf{Person} \sqcap (\geq 3\ \mathsf{Child}) \sqcap \forall \mathsf{Child}.\mathsf{Female}$, which can be read as "all persons with at least three children and only female children".

The assertional formalism allows to state that individuals are instances of concepts or roles, for example, Mary is a Person and Tom is a child of Mary can be expressed by $\mathsf{Mary}:\mathsf{Person}$ and $(\mathsf{Mary}, \mathsf{Tom}):\mathsf{Child}$, respectively.

We assume that a KL-ONE system consisting of a TBox and an ABox presents sound and complete algorithms for determining if a concept description C is more general than a concept description D (in other words, whether C subsumes D), and for checking the consistency of the represented knowledge (knowledge base). Besides these decision problems, the system should offer algorithms for *realization* and *retrieval.* A realization algorithm computes for some individual occurring in an ABox the set of most specialized concept descriptions of a TBox the individual is instance of. A retrieval algorithm computes for a given concept description all individuals which are instances of it. We will see, that the different services are not independent from each other; in fact, an algorithm for deciding the consistency of a knowledge base can be used to solve the other problems.

One of the first systems employing several formalisms to represent different knowledge is the legendary KL-ONE [BS85]. More recently developed systems pursuing this philosophy—so-called hybrid systems—are for example KRYPTON [BPGL85], KANDOR [Pat84], KL-TWO [Vil85], BACK [NvL88, Neb89], LOOM [MB87]. As a drawback—to the best of my knowledge—all these systems use incomplete realization algorithms. Such an incomplete algorithm sometimes fails in recognizing that an individual is instance of some concept description. Realization can be seen as *abstraction*—generating a concept description (the most specific generalization) in terms of the TBox—followed by *classification*[1] of the description [NvL88, Vil85]. In this classical approach incompleteness arises for two reasons. First, it is not easy to com-

[1] Which means the computation of subsumption relations between the most specific generalization and other concepts in the TBox.

pute the most specific generalization, which is in some sense complete. Furthermore, complete classification needs complete subsumption algorithms, which were not known until 1988 (except for some rather trivial terminological languages). In this paper we describe a method, different from the abstraction/classification approach, to perform realization.

In [SS88, HN90, DHL*90] it is shown how to obtain sound and complete subsumption algorithms for a broad range of terminological languages. The technique underlying these algorithms is constraint propagation, which was firstly introduced by Schmidt-Schauß and Smolka to determine subsumption between concept descriptions.

In this paper we show that this constraint propagation approach can be generalized such that hybrid inferences like realization can be computed. We exemplify this technique by taking a terminological language, which contains conjunction, disjunction, and negation of concepts, as well as value restrictions and number restrictions [HN90]. Nevertheless, the advantage of this approach is, that changing the terminological language, i.e., allowing other concept- and role-forming constructs, causes only very small modifications in the presented algorithm[2]. Moreover, we obtain a uniform method to design algorithms for hybrid inferences. Since hybrid inferences are more general than subsumption, lower complexity bounds for determining subsumption are also lower complexity bounds for hybrid inferences.

This paper is organized as follows. In the next chapter we formally introduce the syntax and semantics of the terminological and assertional language. In chapter 3 we discuss the inferences we will draw. We show that hybrid inferences can be reduced to checking the consistency of knowledge bases. Finally in chapter 4 we design a consistency checking algorithm using a constraint propagation technique.

2 The Hybrid Formalism

In this chapter we formally introduce the hybrid formalism. In 2.1 we define the terminological formalism and TBoxes (terminologies). In 2.2 we single out special classes of concept descriptions and terminologies, respectively, as

[2] Generally, if the algorithm is applied to problems of a restricted language, then it behaves better.

normal forms. The assertional formalism is presented in 2.3.

2.1 The Terminological Formalism

We assume two disjoint alphabets of symbols, called *concepts* and *roles.* The special concept symbols $\top$ and $\bot$ are called *top symbol* and *bottom symbol.* The *concept descriptions* (denoted by C and D) are formed out of concepts (denoted by A and B) according to the abstract syntax rule

$$\begin{array}{rcll} C, D & \rightarrow & A \mid & \text{(atomic concept)} \\ & & \neg C \mid & \text{(complement)} \\ & & C \sqcap D \mid C \sqcup D \mid & \text{(conjunction, disjunction)} \\ & & \forall R.C \mid \exists R.C \mid & \text{(value restrictions)} \\ & & (\geq n\ R) \mid (\leq n\ R) & \text{(number restrictions),} \end{array}$$

where n is a nonnegative integer. The abstract syntax rule can be read as, e.g., if A is a concept, then A is a concept description, or if R is a role and C is a concept description, then $\forall R.C$ is a concept description.

An *interpretation* $\mathcal{I} = (D^{\mathcal{I}}, \mathcal{I}[\cdot])$ of a concept description consists of a set $D^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$) and a function $\mathcal{I}[\cdot]$ (the *interpretation function* of $\mathcal{I}$). This function

- maps every concept description to a subset of $D^{\mathcal{I}}$ and every role to a subset of $D^{\mathcal{I}} \times D^{\mathcal{I}}$
- interprets $\top$ as $D^{\mathcal{I}}$ and $\bot$ as the empty set,
- interprets $\sqcap$ as intersection, $\sqcup$ as union, and $\neg$ as complement of sets, and
- satisfies the following equations:

$$\begin{array}{rcl} \mathcal{I}[\forall R.C] & = & \{a \in D^{\mathcal{I}} \mid \forall (a,b) \in \mathcal{I}[R] : b \in \mathcal{I}[C]\} \\ \mathcal{I}[\exists R.C] & = & \{a \in D^{\mathcal{I}} \mid \exists (a,b) \in \mathcal{I}[R] : b \in \mathcal{I}[C]\} \\ \mathcal{I}[(\leq n\ R)] & = & \{a \in D^{\mathcal{I}} \mid |\{b \in D^{\mathcal{I}} \mid (a,b) \in \mathcal{I}[R]\}| \leq n\} \\ \mathcal{I}[(\geq n\ R)] & = & \{a \in D^{\mathcal{I}} \mid |\{b \in D^{\mathcal{I}} \mid (a,b) \in \mathcal{I}[R]\}| \geq n\}, \end{array}$$

where $|\cdot|$ denotes the cardinality of sets.

A concept description C is *consistent* if there exists an interpretation $\mathcal{I}$ such that $\mathcal{I}[C]$ is nonempty, and *inconsistent* otherwise. We say C *subsumes* D if $\mathcal{I}[C] \supseteq \mathcal{I}[D]$ for every interpretation $\mathcal{I}$, and C is *equivalent* to D if $\mathcal{I}[C] = \mathcal{I}[D]$ for every interpretation $\mathcal{I}$.

Example 2.1 Let Person and Female be concepts and Child be a role. Person with at least two children can be expressed as $\mathsf{Person} \sqcap (\geq 2\ \mathsf{Child})$; person with at least three children and only female children can be expressed as $\mathsf{Person} \sqcap (\geq 3\ \mathsf{Child}) \sqcap \forall \mathsf{Child}.\mathsf{Female}$. Obviously, the former concept description subsumes the latter. The concept description $\mathsf{Person} \sqcap (\geq 2\ \mathsf{Child})$ is equivalent to $\mathsf{Person} \sqcap \neg(\leq 1\ \mathsf{Child})$ and the concept description $\mathsf{Person} \sqcap (\geq 2\ \mathsf{Child}) \sqcap (\leq 1\ \mathsf{Child})$ is inconsistent.

Let A be a concept and C be a concept description. A *terminological axiom* has the form $A \sqsubseteq C$ (*concept specialization*) or $A \doteq C$ (*concept definition*). We say A is the *concept name* for C. A *terminology* (*TBox*) $\mathcal{T}$ is a finite set of terminological axioms with the additional restriction that every concept symbol may appear at most once as the left hand side of a terminological axiom in $\mathcal{T}$. The concept specialization $\sqsubseteq$ defines necessary conditions (partial definition) whereas the concept definition $\doteq$ defines necessary and sufficient conditions (complete definition).

A TBox $\mathcal{T}$ contains a *cycle* iff there exists a concept name A in $\mathcal{T}$ such that the concept symbol A occurs in the concept description, which is obtained from A's right hand side by iterated substitutions of some of the concept names by their right hand sides. In this paper we only consider terminologies without cycles. Almost all TBox formalisms don't allow the use of cycles. See [Baa90a, Neb89] for a discussion of terminological cycles.

An interpretation $\mathcal{I}$ *satisfies* a terminological axiom σ iff

$$\begin{array}{ll} \mathcal{I}[A] \subseteq \mathcal{I}[C] & \text{if} \quad \sigma = A \sqsubseteq C \\ \mathcal{I}[A] = \mathcal{I}[C] & \text{if} \quad \sigma = A \doteq C. \end{array}$$

An interpretation $\mathcal{I}$ is a *model* for a TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all terminological axioms in $\mathcal{T}$. A TBox $\mathcal{T}$ *implies* a terminological axiom σ, written $\mathcal{T} \models \sigma$, if σ is satisfied by all models of $\mathcal{T}$.

Example 2.2 Consider the TBox $\mathcal{T}$ defined by

$$\begin{aligned} \mathsf{Woman} &\sqsubseteq \mathsf{Human} \\ \mathsf{Mother\text{-}of\text{-}daughters} &\doteq \mathsf{Woman} \sqcap (\geq 1\ \mathsf{Child}) \sqcap \forall \mathsf{Child}.\mathsf{Woman}. \end{aligned}$$

That means a woman is a human (concept specialization) and a mother-of-daughters is a woman with at least one child and all her children are women (concept definition). Obviously, $\mathcal{T} \models \mathsf{Mother\text{-}of\text{-}daughters} \sqsubseteq \mathsf{Woman}$.

2.2 Normalization

It is useful to have a certain normal form for concept descriptions and terminologies. In the following we single out a special class of concept descriptions as normal forms and describe how to compute them.

A concept description is called *simple* if it contains only complements of the form $\neg A$, where A is a concept symbol.

Proposition 2.3 *For every concept description one can compute in linear time an equivalent simple concept description.*

Proof. We transform concept descriptions into simple concept descriptions by rewriting in top-down order with the following rules:

$$\begin{aligned} \neg\forall R.C &\to \exists R.\neg C \\ \neg\exists R.C &\to \forall R.\neg C \\ \neg(C \sqcap D) &\to \neg C \sqcup \neg D \\ \neg(C \sqcup D) &\to \neg C \sqcap \neg D \\ \neg\neg C &\to C \\ \neg(\geq n\ R) &\to \begin{cases} (\leq (n-1)\ R) & \text{if} \quad n > 0 \\ \bot & \text{if} \quad n = 0 \end{cases} \\ \neg(\leq n\ R) &\to (\geq (n+1)\ R). \end{aligned}$$

It is easy to see that applications of the rules preserve equivalence of concept descriptions. □

We now define a normal form for cycle free TBoxes. A TBox $\mathcal{T}$ is *expanded* iff

1. $\mathcal{T}$ contains only terminological axioms of the form $A \doteq C$

2. every concept description on a right hand side in $\mathcal{T}$ contains only concepts that do not occur as a left hand side in $\mathcal{T}$ (so-called *primitive* concepts)

3. every concept description in $\mathcal{T}$ is simple.

The following proposition shows that every cycle free TBox $\mathcal{T}$ can be transformed into an "equivalent" expanded TBox $\mathcal{T}'$; in other words, $\mathcal{T}$ can be expressed by $\mathcal{T}'$. For a discussion of expressiveness of knowledge representation languages see [Baa90b].

Proposition 2.4 *Every TBox $\mathcal{T}$ without cycles can be transformed into an expanded TBox $\mathcal{T}'$ such that for every model $\mathcal{M}$ for $\mathcal{T}$ there exists a model $\mathcal{M}'$ with the same domain for $\mathcal{T}'$ with $\mathcal{M}[A] = \mathcal{M}'[A]$ for every concept A in $\mathcal{T}$ and $\mathcal{M}[R] = \mathcal{M}'[R]$ for every role R in $\mathcal{T}$, and vice versa.*[3]

Proof. The proof is disposed in four parts. First, we eliminate the specialization operators. Then every concept description on the right hand side of a concept definition is expanded, that is, defined concepts are substituted by their definition until the concept description contains only primitive concepts. Third, each expanded concept description is transformed into a simple concept description. Hence we obtain an expanded TBox $\mathcal{T}'$ for a TBox $\mathcal{T}$. Finally, we show that for every model $\mathcal{M}$ for $\mathcal{T}$ there exists a model $\mathcal{M}'$ with the same domain for $\mathcal{T}'$ such that $\mathcal{M}$ and $\mathcal{M}'$ are identical on the concepts and roles occurring in $\mathcal{T}$, and vice versa.

Elimination of specializations. For every specialization $A \sqsubseteq C$ we introduce a new concept A^* and substitute $A \sqsubseteq C$ by the concept definition $A \doteq C \sqcap A^*$.

Expansion. For every concept description C on the right hand side of concept definitions we substitute each concept in C which is defined by another concept description by its definition, i.e., its right hand side. This process it iterated until there remain only primitive concepts on the right hand sides of the concept descriptions. This process stops if and only if $\mathcal{T}$ contains no cycle.

[3] See also [Neb89].

Simplification. Every concept description on the right hand side of concept definitions is transformed into an equivalent simple concept description. Thus we have obtained an expanded TBox $\mathcal{T}'$ for $\mathcal{T}$.

Model transformation. Let $\mathcal{M}$ be a model for $\mathcal{T}$. We define a model $\mathcal{M}'$ for $\mathcal{T}'$ by putting $\mathcal{D}^{\mathcal{M}'} = \mathcal{D}^{\mathcal{M}}$, $\mathcal{M}'[A] = \mathcal{M}[A]$ for every concept A in $\mathcal{T}$ and $\mathcal{M}'[R] = \mathcal{M}[R]$ for every role R in $\mathcal{T}$. Furthermore, for every new concept A^* in $\mathcal{T}'$ with $A \doteq C \sqcap A^*$ let $\mathcal{M}'[A^*] = \mathcal{M}[A]$. It is easy to see that $\mathcal{M}'$ is a model for $\mathcal{T}'$. Conversely, suppose $\mathcal{M}'$ is a model for $\mathcal{T}'$. We obtain a model $\mathcal{M}$ for $\mathcal{T}$ by putting $\mathcal{D}^{\mathcal{M}} = \mathcal{D}^{\mathcal{M}'}$, $\mathcal{M}[A] = \mathcal{M}'[A]$ for every concept A in $\mathcal{T}$ and $\mathcal{M}[R] = \mathcal{M}'[R]$ for every role R. Obviously, $\mathcal{M}$ is a model for $\mathcal{T}$. □

Note that an expanded TBox is in the worst case exponential in the size of the input TBox. The elimination of the specializations and the simplification of concept descriptions is harmless whereas the expansion may enlarge the TBox significantly. For instance[4], the expansion of the TBox $\mathcal{T}_n$

$$\begin{array}{rcl} C_0 & = & \forall R.A \sqcap \forall R'.A \\ C_1 & = & \forall R.C_0 \sqcap \forall R'.C_0 \\ & \vdots & \\ C_n & = & \forall R.C_{n-1} \sqcap \forall R'.C_{n-1} \end{array}$$

leads to a TBox, which is exponential in n and thus in the size of $\mathcal{T}_n$.

Corollary 2.5 *Let $\mathcal{T}$ be a TBox and $\mathcal{T}'$ the expanded TBox of Proposition 2.4. Then:*

$$\mathcal{T} \models C \sqsubseteq D \quad \textit{iff} \quad \mathcal{T}' \models C \sqsubseteq D.$$

Proof. Let $\mathcal{T}$ be a TBox and C, D concept descriptions in $\mathcal{T}$. Assume that $\mathcal{M}[C] \not\subseteq \mathcal{M}[D]$ for a model $\mathcal{M}$ for $\mathcal{T}$. There exists a model $\mathcal{M}'$ for $\mathcal{T}'$ such that $\mathcal{M}'[C] = \mathcal{M}[C]$ and $\mathcal{M}'[D] = \mathcal{M}[D]$. Hence $\mathcal{M}'[C] \not\subseteq \mathcal{M}'[D]$. The other direction can be shown in the same way. □

Example 2.6 For the TBox of Example 2.2 we obtain the following expanded TBox:

$$\mathsf{Woman} \doteq \mathsf{Human} \sqcap \mathsf{Woman}^*$$

[4] This example is taken from [NS89].

$$\begin{aligned}\mathsf{Mother\text{-}of\text{-}daughters} &\doteq \mathsf{Human} \sqcap \mathsf{Woman}^* \sqcap (\geq 1\ \mathsf{Child}) \sqcap \\ &\quad \forall \mathsf{Child}.(\mathsf{Human} \sqcap \mathsf{Woman}^*)\end{aligned}$$

2.3 The Assertional Formalism

The assertional formalism allows for the assertion of objects (individuals). We can describe a concrete world by stating that objects are instances of concepts and roles.

We assume a further alphabet of symbols, called *objects*, disjoint from concepts and roles. Objects are denoted by a and b. An *object description* has the form $a : A$ where a is an object and A a concept. A *relation description* has the form $(a, b) : R$ where a and b are objects and R is a role. A *description* is either an object description or a relation description. A *world description* (*ABox*) is a finite set descriptions.

We extend the interpretation function $\mathcal{I}[\cdot]$ of an interpretation $\mathcal{I}$ to objects by mapping them to elements of $D^{\mathcal{I}}$ such that $\mathcal{I}[a] \neq \mathcal{I}[b]$ if $a \neq b$. This restriction on the interpretation function ensures that different objects are assumed to denote different individuals in the world. This property is called *unique name assumption*, which is usually assumed in the database world.

An interpretation $\mathcal{I}$ *satisfies* a description α iff

$$\begin{aligned} \mathcal{I}[a] \in \mathcal{I}[A] &\quad \text{if} \quad \alpha = a : A \\ (\mathcal{I}[a], \mathcal{I}[b]) \in \mathcal{I}[R] &\quad \text{if} \quad \alpha = (a, b) : R. \end{aligned}$$

An interpretation $\mathcal{I}$ is a *model* for an ABox $\mathcal{A}$ if $\mathcal{I}$ satisfies all descriptions in $\mathcal{A}$.

Example 2.7 Consider the ABox that consists of the following descriptions:

$$\begin{gathered} \mathsf{Mary} : \mathsf{Woman} \\ (\mathsf{Mary}, \mathsf{Tom}) : \mathsf{Child} \end{gathered}$$

The ABox describes a world in which Mary is an instance of the concept Woman. Tom and Mary are related by the Child role, which means, that Tom is a Child of Mary.

3 Reasoning in the Hybrid Formalism

In this chapter we consider what kind of information, which is implicitly represented by a TBox and an ABox, can be made explicitly. One kind of reasoning is the computation of the subsumption relation between concepts. An algorithm doing this work is called *classifier.* We will see, that subsumption between concepts only depends on the TBox. As a link between the different formalisms, object and relation descriptions in an ABox may refer to concepts and roles, which are defined in the TBox. This involves another kind of reasoning. We need an algorithm for deciding whether an ABox $\mathcal{A}$ with respect to a TBox $\mathcal{T}$ is consistent, e.g. does there exist an interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a model for $\mathcal{A}$ and for $\mathcal{T}$. Besides the consistency test we need more constructive inferences. For instance, the computation of the most specialized concepts of a TBox an object is instance of (so-called *realization*), and the computation of the set of all objects of an ABox that are instances of a given concept description (so-called *retrieval*).

An interpretation $\mathcal{I}$ is a *model for an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$*, if $\mathcal{I}$ is a model for $\mathcal{A}$ and for $\mathcal{T}$. An ABox w.r.t. a TBox is *consistent* if it has a model. An ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ *imply* a description α if all models for $\mathcal{A}$ w.r.t. $\mathcal{T}$ satisfy α, written $\mathcal{A} \models_{\mathcal{T}} \alpha$.

Example 3.1 Consider the TBox $\mathcal{T}$, which contains the axioms

$$\begin{array}{rcl} \mathsf{Woman} & \sqsubseteq & \mathsf{Human} \\ \mathsf{Mother\text{-}of\text{-}daughters} & \doteq & \mathsf{Woman} \sqcap (\geq 1\ \mathsf{Child}) \sqcap \forall \mathsf{Child}.\mathsf{Woman}, \end{array}$$

and the ABox $\mathcal{A} = \{\mathsf{Mary} : \mathsf{Mother\text{-}of\text{-}daughters},\ (\mathsf{Mary}, \mathsf{Susi}) : \mathsf{Child}\}$.

The fact, that Susi is a Woman is not mentioned explicitly but is implied by $\mathcal{A}$ and $\mathcal{T}$. We assume, that in almost every case one can describe only a small part of the world. Thus, we have chosen an *open world semantics* as opposed to the *closed world assumption.* We cannot conlude, for instance, that Susi is the only child of Mary, since there may exist a world in which Mary has several children (which of course are women).

As mentioned above, subsumption between concepts depends only on the TBox. Since the ABox formalism is very restricted, especially it doesn't allow universally quantified assertions, no new subsumption relations hold between concepts[5]. In other words, the ABox is a *conservative extension* of a TBox.

We now formally describe what kind of inferences we will draw from the hybrid formalism. Therefore let $\mathcal{T}$ be a TBox and $\mathcal{A}$ be an ABox.

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

Consistency problem: Does there exist a model for $\mathcal{A}$ w.r.t. $\mathcal{T}$?

Instance problem: Do $\mathcal{A}$ and $\mathcal{T}$ imply an object description[6] $a : A$?

Realization problem: Let a be an object occurring in $\mathcal{A}$. The set of most specialized concepts of which a is an instance is defined as

$$\begin{aligned} MSC_{\mathcal{A},\mathcal{T}}(a) := \big\{ A \mid \mathcal{A} \models_{\mathcal{T}} a : A,\ \nexists B \text{ with } A \neq B, \\ \mathcal{A} \models_{\mathcal{T}} a : B \text{ and } \mathcal{T} \models B \sqsubseteq A \big\}, \end{aligned}$$

where A and B are concept names in $\mathcal{T}$. The computation of the MSC set for some object is called realization.

Retrieval problem: Let A be a concept name in $\mathcal{T}$. The retrieval problem is to compute the set of all objects a occurring in $\mathcal{A}$ such that $\mathcal{A} \models_{\mathcal{T}} a : A$.

In the following we will show that an algorithm for deciding the consistency problem can be used to solve the subsumption, instance, realization and retrieval problem. Proposition 3.2 shows a reduction from the subsumption problem to the instance problem, and Proposition 3.3 shows a reduction from the instance problem to the consistency problem. Finally we describe a method to obtain a realization and retrieval algorithm, respectively, using an algorithm that solves the instance problem.

[5] Actually, the claim holds if and only if the ABox w.r.t. the TBox is consistent.

[6] Note that $\mathcal{A} \models_{\mathcal{T}} (a, b) : R$ holds if and only if $(a, b) : R \in \mathcal{A}$, since every role in $\mathcal{T}$ is completely undefined and hence not related to any other role in $\mathcal{T}$. The situation changes in the presence of role-forming constructs such as intersection of roles or inverse roles.

Proposition 3.2 *The concept description C subsumes the concept description D iff*

$$\{a\!:\!B\} \models_{\mathcal{T}} a\!:\!A \quad \textit{where} \quad \mathcal{T} = \{A \doteq C,\ B \doteq D\}$$

where A, B new concepts.

Proposition 3.3 *Let $\mathcal{T}$ be a TBox, $\mathcal{A}$ be an ABox, $a\!:\!A$ an object description, and $\bar{A}$ a new concept. Then:*

$$\mathcal{A} \models_{\mathcal{T}} a\!:\!A \quad \textit{iff} \quad \mathcal{A} \cup \{a\!:\!\bar{A}\} \textit{ has no model w.r.t. } \mathcal{T} \cup \{\bar{A} \doteq \neg A\}.$$

An algorithm that computes for a given object a the most specialized concepts of which a is an instance can be obtained as follows. Let $\mathcal{T}$ be a TBox, $\mathcal{A}$ an ABox, and let a be an object occurring in $\mathcal{A}$. First, for every concept name A in $\mathcal{T}$ we decide whether $\mathcal{A} \models_{\mathcal{T}} a\!:\!A$ holds. Thus we know all concepts object a is instance of. In a second step, we eliminate all those concepts, which subsume other concepts, to obtain the most specialized concepts. Hence we have solved the realization problem.

Suppose we want to compute all instances of a concept description C. A very simple but not very smart algorithm[7], which solves the retrieval problem, proceeds as follows. First we add the new concept description $A \doteq C$ where A is a new concept name to $\mathcal{T}$. Then we test for every object in $\mathcal{A}$ whether $\mathcal{A} \models_{\mathcal{T}} a\!:\!A$ holds. Thus we get all objects occurring in $\mathcal{A}$ that are instances of C.

4 The Consistency Problem

We are going to devise a calculus for solving the consistency problem. The calculus will operate on constraints consisting of variables, concept descriptions and roles.

We assume that there exists an alphabet of variable symbols, which will be denoted by the letters x, y and z and which is a superset of the objects. A constraint is a syntactic object of one of the forms

$$x\!:\!C, \quad xRy,$$

[7] For implementations this algorithm might not be appropriate, but there are (worst case) examples for which the presented algorithm is in some sense optimal.

where C is a simple concept description and R is a role. Let $\mathcal{I}$ be an interpretation. An *$\mathcal{I}$-assignment* is a function α that maps every variable to an element of $D^{\mathcal{I}}$. We say that α *satisfies* $x\colon C$ if $\alpha(x) \in \mathcal{I}[C]$, and α *satisfies* xRy if $(\alpha(x), \alpha(y)) \in \mathcal{I}[R]$. A constraint s is *consistent* if there is an interpretation $\mathcal{I}$ and an $\mathcal{I}$-assignment α such that α satisfies s. A *constraint system* S is a finite, nonempty set of constraints. An $\mathcal{I}$-assignment α *satisfies* a constraint system S if α satisfies every constraint in S. A constraint system S is *consistent* if there is an interpretation $\mathcal{I}$ and an $\mathcal{I}$-assignment α such that α satisfies S, and *inconsistent* otherwise.

Let $\mathcal{T}$ be an expanded TBox and let $\mathcal{A}$ be an ABox such that every concept occurring in $\mathcal{A}$ is a concept name in $\mathcal{T}$.[8] The constraint system S is *induced by $\mathcal{T}$ and $\mathcal{A}$* iff

$$S = \{a\colon C \mid a\colon A \in \mathcal{A} \text{ and } A \doteq C \in \mathcal{T}\} \cup \{aRb \mid (a,b)\colon R \in \mathcal{A}\}.$$

The following proposition, which is easy to prove, shows the relationship between ABoxes/TBoxes and constraint systems.

Proposition 4.1 *Let S be a constraint system induced by $\mathcal{T}$ and $\mathcal{A}$. Then $\mathcal{A}$ w.r.t. $\mathcal{T}$ has no model if and only if S is inconsistent.*

We now present a method to decide the consistency of constraint systems, and hence the consistency of an ABox w.r.t. a TBox. Our calculus starts with a constraint system S and adds in successive propagation steps constraints to S. This constraint propagation process is iterated until either a contradiction occurs or an interpretation, which is a model for $\mathcal{A}$ w.r.t. $\mathcal{T}$, can be obtained from the resulting constraint system.

Before we formulate the rules we need some notation. Let S be a constraint system. For a variable x we count the number of variables y with xRy for some role R. We therefore define $n_{R,S}(x) := |\{y \mid xRy \in S\}|$. With $[y/z]S$ we denote the constraint system that is obtained from S by replacing each occurrence of y by z. The *propagation rules* are:

1. $S \rightarrow_{\sqcap} \{x\colon C_1,\ x\colon C_2\} \cup S$

   if $x\colon C_1 \sqcap C_2$ is in S, and $x\colon C_1$ and $x\colon C_2$ are not both in S

[8] This is not a restriction since, if $a : A \in \mathcal{A}$ and A is not a concept name in $\mathcal{T}$, then the concept definition $A \doteq A^*$ is inserted to $\mathcal{T}$ and each occurence of A in $\mathcal{T}$ is replaced by A^*.

2. $S \rightarrow_{\sqcup} \{x : D\} \cup S$

   if $x : C_1 \sqcup C_2$ is in S, neither $x : C_1$ nor $x : C_2$ is in S, and $D = C_1$ or $D = C_2$

3. $S \rightarrow_{\forall} \{y : C\} \cup S$

   if $x : \forall R.C$ and xRy are in S and $y : C$ is not in S

4. $S \rightarrow_{\exists} \{y : C,\ xRy\} \cup S$

   if $x : \exists R.C$ is in S, and there is no variable z such that xRz and $z : C$ is in S and y is a new variable

5. $S \rightarrow_{\geq} \{xRy\} \cup S$

   if $x : (\geq n\ R)$ is in S, $n_{R,S}(x) = 0$, and y is a new variable

6. $S \rightarrow_{\leq} [y/z]S$

   if $x : (\leq n\ R)$, xRy, xRz are in S, $n_{R,S}(x) > n$, and y is not an object

7. $S \rightarrow_{\perp} \{x : \perp\}$

   if $x : A$ and $x : \neg A$ are in S, or
   if $x : (\geq n\ R)$, $x : (\leq m\ R)$ are in S and $n > m$, or
   if $x : (\leq 0\ R)$, xRy are in S, or
   if $x : (\leq m\ R)$, $xRa_1, \ldots, xRa_{m+1}$ are in S,
   and $a_1, \ldots, a_{m+1}$ are different objects.

The $\rightarrow_{\sqcap}$-, $\rightarrow_{\sqcup}$- and $\rightarrow_{\forall}$-rule are obvious. An application of the $\rightarrow_{\exists}$-rule introduces two constraints to "satisfy" the constraint $x : \exists R.C$. Suppose an "at least"-restriction is imposed on some variable x for some role R in a constraint system S. If $n_{R,S}(x) = 0$, the $\rightarrow_{\geq}$-rule forces to introduce the constraint xRy. Otherwise, if $n_{R,S}(x) > 0$, the $\rightarrow_{\geq}$-rule doesn't apply. In Proposition 4.3 we will see that this restricted version of the general rule

- $S \rightarrow_{\geq^*} \{xRy\} \cup S$

  if $x : (\geq n\ R)$ is in S, $n_{R,S}(x) < n$, and y is a new variable

is sufficient to satisfy "at least"-restrictions. Note that applications of the $\rightarrow_{\geq^*}$-rule insert at least 100 constraints of the form xRy for a constraint

$x : (\geq 100\ R)$ whereas applications of the $\rightarrow_{\geq}$-rule add at most one constraint. Applications of the $\rightarrow_{\leq}$-rule guarantee that "at most"-restrictions are satisfied. We ensure that no object is substituted by a variable or another object to maintain the unique name assumption on the objects. Finally, the $\rightarrow_{\perp}$-rule detects inconsistencies in constraint systems.

Proposition 4.2 *Let S and S' be constraint systems. Then:*

1. *If S' is obtained from S by application of the (deterministic) $\rightarrow_{\sqcap}$-, $\rightarrow_{\forall}$-, $\rightarrow_{\exists}$-, $\rightarrow_{\geq}$- or $\rightarrow_{\perp}$-rule, then S is consistent if and only if S' is consistent.*

2. *If S' is obtained from S by application of the (nondeterministic) $\rightarrow_{\sqcup}$- or $\rightarrow_{\leq}$-rule, then S is consistent if S' is consistent. Furthermore, if the $\rightarrow_{\sqcup}$-rule or the $\rightarrow_{\leq}$-rule applies, then there is a choice for S' such that S' is consistent if and only if S is consistent.*

Proof. Easy. □

Let S be a constraint system. The *canonical interpretation* $\mathcal{I}_S$ of S is obtained by taking for $D^{\mathcal{I}_S}$ all variables occurring in S, for $\mathcal{I}_S[A]$ all x such that $x : A$ is in S, where A is a concept symbol different from $\top$ and $\perp$, for $\mathcal{I}_S[R]$ all pairs (x, y) such that xRy is in S, and by taking the set $\mathcal{I}_S[C]$ for complex concept descriptions as required by the definition of an interpretation. The *canonical assignment* α_S of S is obtained by mapping variables to themselves.

A constraint system is *complete* if no propagation rule applies to it. A *clash* is a constraint of the form $x : \perp$.

Proposition 4.3 *Every clash free complete constraint system is consistent.*

Proof. Let S be a clash free, complete constraint system. We extend S to a constraint system S' such that the following properties hold:

1. $S \subseteq S'$

2. S' is clash free and complete

3. $x : (\geq n\ R) \in S'$ implies $n_{R,S'}(x) \geq n$.

The third property ensures that the $\rightarrow_{\geq^*}$-rule is not applicable to S'. Such an extension S' can be obtained from S by constructing a sequence $S = S_1$, $S_2, \ldots, S_k = S'$ where S_j is transformed into S_{j+1} using the following steps: select a variable x with $\{x : (\geq n\ R), xRy\} \subseteq S_j$ and $n_{R,S_j}(x) = m < n$; let $y_1, \ldots, y_{n-m}$ be new variables; let S_{j+1} be obtained from S_j by adding $xRy_1, \ldots, xRy_{n-m}$, by adding $y_1 : C, \ldots, y_{n-m} : C$ for every constraint in S_j of the form $y : C$, and by adding $y_1Rz, \ldots, y_{n-m}Rz$ for every constraint in S_j of the form yRz. The process eventually halts when a constraint system with the required properties is reached. Since the newly introduced constraints are copies of constraints already occurring in S, there is no clash in S'. Furthermore, it is easy to prove that S' is a complete constraint system.

Now we will show that the constraint system S' is consistent. Let $\mathcal{I}_{S'}$ be the canonical interpretation of S' and $\alpha_{S'}$ the canonical assignment of S'. We prove that $\alpha_{S'}$ satisfies every $s \in S'$. If s has the form xRy, then $\alpha_{S'}$ satisfies xRy by definition of $\mathcal{I}_{S'}$ and $\alpha_{S'}$. Otherwise, if s has the form $x : C$, we show by induction on the structure of C that $\alpha_{S'}(x) \in \mathcal{I}_{S'}[C]$.

Base case: If C is a concept symbol different from $\top$ and $\bot$, then $\alpha_{S'}(x) \in \mathcal{I}_{S'}[C]$ by defintion of $\mathcal{I}_{S'}$ and $\alpha_{S'}$. If $C = \top$, then obviously $\alpha_{S'}(x) \in \mathcal{I}_{S'}[\top]$. Since S' is clash free we have $C \neq \bot$.

Induction step: If $C = \neg A$, the constraint $x : A$ is not in S' since S' is clash free and complete. Then $\alpha_{S'}(x) \notin \mathcal{I}_{S'}[A]$ and $\alpha_{S'}(x) \in D^{\mathcal{I}_{S'}} - \mathcal{I}_{S'}[A]$. Hence $\alpha_{S'}(x) \in \mathcal{I}_{S'}[\neg A]$. If $C = \exists R.D$, then xRy and $y : D$ are in S' since the $\rightarrow_{\exists}$-rule doesn't apply to S'. Then $\alpha_{S'}(y) \in \mathcal{I}_{S'}[D]$ holds by the induction hypothesis and hence $\alpha_{S'}(x) \in \mathcal{I}_{S'}[\exists R.D]$. Similarly, it can be shown that constraints of the form

$$x : C \sqcap D,\ x : C \sqcup D,\ x : \forall R.C$$

are satisfied by $\alpha_{S'}$ and $\mathcal{I}_{S'}$. If $C = (\geq n\ R)$, then $n_{R,S'}(x) \geq n$ since the $\rightarrow_{\geq^*}$-rule doesn't apply to S'. Hence $\alpha_{S'}(x) \in \mathcal{I}_{S'}[(\geq n\ R)]$. If $C = (\leq n\ R)$, then $n_{R,S'}(x) \leq n$ since the $\rightarrow_{\leq}$-rule doesn't apply to S'. Contradictory number restrictions constraints such as $x : (\geq n\ R)$, $x : (\leq m\ R)$, where $n > m$, lead to clashes. □

Since obviously a constraint system containing a clash is inconsistent, the preceding proposition implies the following result.

Theorem 4.4 *A complete constraint system is consistent if and only if it contains no clash.*

Now we will show that every constraint system, which is induced by a TBox and an ABox, can be extended to a complete constraint system.

Theorem 4.5 *If S is an induced constraint system, then in at most exponentially many propagation steps one can nondeterministically compute a complete constraint system S' for S such that S' is consistent if and only if S is consistent.*

Proof. An induced constraint system S can be extended to a complete constraint system preserving consistency and inconsistency using the propagation rules. Every constraint system obtained from S using the propagation rules is at most exponential in the size of S. Thus the $\rightarrow_{\leq}$-rule can be applied at most exponentially many times to a variable x. Since the other rules introduce new constraints, we conclude that it takes at most exponentially many steps to transform S into a complete constraint system.[9] □

Thus we have shown that an expanded TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ can be transformed into a constraint system S, such that $\mathcal{A}$ w.r.t. $\mathcal{T}$ has no model if and only if S is inconsistent (Proposition 4.1). Furthermore, S can be extended to a complete constraint system, which can be checked in polynomial time on consistency. Since it has been shown that the terminological language proposed in this paper has a PSPACE-complete subsumption problem [HN90], we conclude with following claim.

Corollary 4.6 *The consistency, instance, realization and retrieval problems are decidable, PSPACE-hard problems.*

5 Conclusion

In this paper we gave a uniform method for developing sound and complete algorithms for hybrid inferences in KL-ONE-based knowledge representation systems. We showed that the constraint propagation technique which was

[9] A complete estimation is given in [Hol89].

first introduced in [SS88] to determine subsumption between concepts can be generalized such that hybrid inferences can be computed. An advantage of this approach is, that changing the terminological language causes only slight modifications of the propagation rules in a straightforward way as shown in [SS88, HN90, DHL*90]. Lower complexity bounds for subsumption in terminological languages are also lower complexity bounds for hybrid inferences. On the other hand, although the consistency problem is more general than the subsumption problem, we have a strong feeling that for almost all terminological languages with respect to the proposed assertional language upper complexity bounds for subsumption are also upper complexity bounds for hybrid inferences.

Acknowledgement. I would like to thank my colleagues in the WINO project. Especially I am grateful to Franz Baader and Werner Nutt for our joint discussions about realization and for reading earlier drafts.

References

[Baa90a] F. Baader *Terminological Cycles in* KL-ONE-*based Knowledge Representation Languages.* DFKI Research Report RR-90-01, DFKI, Postfach 2080, D-6750 Kaiserslautern, West Germany.

[Baa90b] F. Baader *A Formal Definition for the Expressive Power of Knowledge Representation Languages.* DFKI Research Report RR-90-05, DFKI, Postfach 2080, D-6750 Kaiserslautern, West Germany.

[BL84] R. J. Brachmann, H. J. Levesque. "The tractability of subsumption in frame based description languages." In *Proceedings of the 4th National Conference of the AAAI,* pp. 34–37, Austin, Tex., 1984.

[BPGL85] R. J. Brachman, V. Pigman Gilbert, H. J. Levesque. "An essential hybrid reasoning system: knowledge and symbol level accounts in KRYPTON." In *Proceedings of the 9th IJCAI,* pp. 532–539, Los Angeles, Cal., 1985.

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

[DHL*90] F. Donini, B. Hollunder, M. Lenzerini, A. Marchetti Spaccamela, Daniele Nardi, W. Nutt. *A Source of Complexity in Terminological Reasoning.* DFKI Research Report, DFKI, Postfach 2080, D-6750 Kaiserslautern, West Germany. Forthcoming.

[Hol89] B. Hollunder. "Subsumption Algorithms for Some Attributive Concept Description Languages." SEKI Report SR-89-16, FB Informatik, Universität Kaiserslautern, D-6750, Kaiserslautern, West Germany, 1989.

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

[KBR86] T. S Kaczmarek, R. Bates, G. Robins. "Recent developments in NIKL." In *Proceedings of the 5th National Conference of the AAAI*, pp. 578–587, Philadelphia, Pa., 1986.

[LB87] H. J. Levesque, R. J. Brachman. "Expressiveness and tractability in knowledge representation and reasoning." *Computational Intelligence*, **3**:78–93, 1987.

[MB87] R. MacGregor, R. Bates. *The Loom Knowledge Representation Language.* Technical Report ISI/RS-87-188, University of Southern California, Information Science Institute, Marina del Rey, Cal., 1987.

[Neb89] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, PhD thesis, Universität des Saarlandes, Saarbrücken, West Germany, 1989. To appear in Lecture Notes in Artificial Intelligence, Springer Verlag.

[NvL88] B. Nebel, K. von Luck. "Hybrid reasoning in BACK." In Z. W. Ras, L. Saitta (editors), *Methodologies for Intelligent Systems*, pp. 260–269, North Holland, Amsterdam, Netherlands, 1988.

[NS89] B. Nebel, G. Smolka. *Representation and Reasoning with Attributive Descriptions.* IWBS Report81, August 1989, IWBS, IBM Deutschland, Stuttgart, W: Germany.

[Pat84] P. Patel-Schneider. "Small can be beautiful in knowledge representation." In *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, pp. 11–16, Denver, Colo., 1984.

[SS88] M. Schmidt-Schauß, G. Smolka. *Attributive Concept Descriptions with Unions and Complements.* SEKI Report SR-88-21, FB Informatik, Universität Kaiserslautern, D-6750, Kaiserslautern, West Germany, 1988. To appear in Artificial Intelligence.

[Vil85] M. B. Vilain. "The restricted language architecture of a hybrid representation system." In R. J. Bachmann, H. J. Levesque, R. Reiter (editors), *Proceedings of the 9th IJCAI*, pp. 547–551, Los Angeles, Cal., 1985.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
Stuhlsatzenhausweg 3
6600 Saarbrücken 11
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen oder die aktuelle Liste von erhältlichen Publikationen können bezogen werden von der oben angegebenen Adresse.

DFKI Publications

The following DFKI publications or the list of currently available publications can be ordered from the above address.

DFKI Research Reports

RR-90-01
Franz Baader
Terminological Cycles in KL-ONE-based Knowledge Representation Languages
33 pages

Abstract: Cyclic definitions are often prohibited in terminological knowledge representation languages, because, from a theoretical point of view, their semantics is not clear and, from a practical point of view, existing inference algorithms may go astray in the presence of cycles. In this paper we consider terminological cycles in a very small KL-ONE-based language. For this language, the effect of the three types of semantics introduced by Nebel (1987, 1989, 1989a) can be completely described with the help of finite automata. These descriptions provide a rather intuitive understanding of terminologies with cyclic definitions and give insight into the essential features of the respective semantics. In addition, one obtains algorithms and complexity results for subsumption determination. The results of this paper may help to decide what kind of semantics is most appropriate for cyclic definitions, not only for this small language, but also for extended languages. As it stands, the greatest fixed-point semantics comes off best. The characterization of this semantics is easy and has an obvious intuitive interpretation. Furthermore, important constructs – such as value-restriction with respect to the transitive or reflexive-transitive closure of a role – can easily be expressed.

RR-90-02
Hans-Jürgen Bürckert
A Resolution Principle for Clauses with Constraints
25 pages

Abstract: We introduce a general scheme for handling clauses whose variables are constrained by an underlying constraint theory. In general, constraints can be seen as quantifier restrictions as they filter out the values that can be assigned to the variables of a clause (or an arbitrary formulae with restricted universal or existential quantifier) in any of the models of the constraint theory. We present a resolution principle for clauses with constraints, where unification is replaced by testing constraints for satisfiability over the constraint theory. We show that this constrained resolution is sound and complete in that a set of clauses with constraints is unsatisfiable over the constraint theory iff we can deduce a constrained empty clause for each model of the constraint theory, such that the empty clauses constraint is satisfiable in that model. We show also that we cannot require a better result in general, but we discuss certain tractable cases, where we need at most finitely many such empty clauses or even better only one of them as it is known in classical resolution, sorted resolution or resolution with theory unification.

RR-90-03
Andreas Dengel & Nelson M. Mattos

Integration of Document Representation, Processing and Management

18 pages

Abstract: This paper describes a way for document representation and proposes an approach towards an integrated document processing and management system. The approach has the intention to capture essentially freely structured documents, like those typically used in the office domain. The document analysis system ANASTASIL is capable to reveal the structure of complex paper documents, as well as logical objects within it, like receiver, footnote, date. Moreover, it facilitates the handling of the containing information. Analyzed documents are stored by the management system KRISYS that is connected to several different subsequent services. The described integrated system can be considered as an ideal extension of the human clerk, making his tasks in information processing easier. The symbolic representation of the analysis results allow an easy transformation in a given international standard, e.g., ODA/ODIF or SGML, and to interchange it via global network.

RR-90-04
Bernhard Hollunder & Werner Nutt

Subsumption Algorithms for Concept Languages

34 pages

Abstract: We investigate the subsumption problem in logic-based knowledge representation languages of the KL-ONE family and give decision procedures. All our languages contain as a kernel the logical connectives conjunction, disjunction, and negation for concepts, as well as role quantification. The algorithms are rule-based and can be understood as variants of tableaux calculus with a special control strategy. In the first part of the paper, we add number restrictions and conjunction of roles to the kernel language. We show that subsumption in this language is decidable, and we investigate sublanguages for which the problem of deciding subsumption is PSPACE-complete. In the second part, we amalgamate the kernel language with feature descriptions as used in computational linguistics. We show that feature descriptions do not increase the complexity of the subsumption problem.

RR-90-05
Franz Baader

A Formal Definition for the Expressive Power of Knowledge Representation Languages

22 pages

Abstract: The notions "expressive power" or "expressiveness" of knowledge representation languages (KR-languages) can be found in most papers on knowledge representation; but these terms are usually just used in an intuitive sense. The papers contain only informal descriptions of what is meant by expressiveness. There are several reasons which speak in favour of a formal definition of expressiveness: For example, if we want to show that certain expressions in one language *cannot* be expressed in another language, we need a strict formalism which can be used in mathematical proofs. Though we shall only consider KL-ONE-based KR-language in our motivation and in the examples, the definition of expressive power which will be given in this paper can be used for all KR-languages with model-theoretic semantics. This definition will shed a new light on the tradeoff between expressiveness of a representation language and its computational tractability. There are KR-languages with identical expressive power, but different complexity results for reasoning. Sometimes, the tradeoff lies between convenience and computational tractability. The paper contains several examples which demonstrate how the definition of expressive power can be used in positive proofs – that is, proofs where it is shown that one language can be expressed by another language – as well as for negative proofs – which show that a given language cannot be expressed by the other language.

RR-90-06
Bernhard Hollunder
Hybrid Inferences in KL-ONE-based Knowledge Representation Systems
21 pages

Abstract: We investigate algorithms for hybrid inferences in KL-ONE-based knowledge representation systems. Those systems employ two kinds of formalisms: the terminological and the assertional formalism. The terminological formalism consists of a concept description language to define concepts and relations between concepts for describing a terminology. On the other hand, the assertional formalism allows to introduce objects, which are instances of concepts and relations of a terminology. We present algorithms for hybrid inferences such as

- determining subsumption between concepts
- checking the consistency of such a knowledge base
- computing the most specialized concepts an object is instance of
- computing all objects that are instances of a certain concept.

DFKI Technical Memos

TM-89-01
Susan Holbach-Weber
Connectionist Models and Figurative Speech
27 pages

Abstract: This paper contains an introduction to connectionist models. Then we focus on the question of how novel figurative usages of descriptive adjectives may be interpreted in a structured connectionist model of conceptual combination. The suggestion is that inferences drawn from an adjective's use in familiar contexts form the basis for all possible interpretations of the adjective in a novel context. The more plausible of the possibilities, it is speculated, are reinforced by some form of one-shot learning, rendering the interpretative process obsolete after only one (memorable) encounter with a novel figure of speech.

TM-90-01
Som Bandyopadhyay
Towards an Understanding of Coherence in Multimodal Discourse
18 pages

Abstract: An understanding of coherence is attempted in a multimodal framework where the presentation of information is composed of both text and picture segments (or, audio-visuals in general). Coherence is characterised at three levels: coherence at the syntactic level which concerns the linking mechanism of the adjacent discourse segments at the surface level in order to make the presentation valid; coherence at the semantic level which concerns the linking of discourse segments through some semantic ties in order to generate a wellformed thematic organisation; and, coherence at the pragmatic level which concerns effective presentation through the linking of the discourse with the addressees´ preexisting conceptual framework by making it compatible with the addressees´ interpretive ability, and linking the discourse with the purpose and situation by selecting a proper discourse typology. A set of generalised coherence relations are defined and explained in the context of picture-sequence and multimodal presentation of information.

DFKI Documents

D-90-01
Wissenschaftlich-Technischer Jahresbericht 1989
45 Seiten

Zusammenfassung: Dieses Dokument enthält den Wissenschaftlich-Technischen Jahresbericht 1989 des Deutschen Forschungszentrums für Künstliche Intelligenz.

Hybrid Inferences in KL-ONE-based
Knowledge Representation Systems

Bernhard Hollunder

RR-90-06
Research Report