Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Technical Memo
TM-90-03

KRIS :

*K*nowledge *R*epresentation and *I*nference *S*ystem

-System Description-

Franz Baader, Bernhard Hollunder

November 1990

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

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

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
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, Nixdorf, Philips and Siemens. 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

KRIS : *K*nowledge *R*epresentation and *I*nference *S*ystem
-System Description-

Franz Baader, Bernhard Hollunder

DFKI-TM-90-03

$\mathcal{KRIS}$: $\mathcal{K}$nowledge $\mathcal{R}$epresentation and $\mathcal{I}$nference $\mathcal{S}$ystem —System Description—

Franz Baader, Bernhard Hollunder
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Projektgruppe WINO
Postfach 2080, D-6750 Kaiserslautern, West Germany
E-mail: {baader, hollunde}@dfki.uni-kl.de

Abstract

The knowledge representation system KL-ONE first appeared in 1977. Until then many systems based on the idea of KL-ONE have been built. The formal model-theoretic semantics which has been introduced for KL-ONE languages [BL84] provides means for investigating soundness and completeness of inference algorithms. It turned out that almost all implemented KL-ONE systems such as BACK, KL-TWO, LOOM, NIKL, SB-ONE use sound but incomplete algorithms.

Until recently, sound *and* complete algorithms for the basic reasoning facilities in these systems such as consistency checking, subsumption checking (classification) and realization were only known for rather trivial languages. However, in the last two years concept languages (term subsumption languages) have been thoroughly investigated (see for example [SS88,Neb90,HNS90,DHL*90]). As a result of these investigations it is now possible to provide sound and complete algorithms for relatively large concept languages.

In this paper we describe $\mathcal{KRIS}$ which is an implemented prototype of a KL-ONE system where all reasoning facilities are realized by sound and complete algorithms. This system can be used to investigate the behaviour of sound and complete algorithms in practical applications. Hopefully, this may shed a new light on the usefulness of complete algorithms for practical applications, even if their worst case complexity is NP or worse.

$\mathcal{KRIS}$ provides a very expressive concept language, an assertional language, and sound and complete algorithms for reasoning. We have chosen the concept language such that it contains most of the constructs used in KL-ONE systems with the obvious restriction that the interesting inferences such as consistency checking, subsumption checking, and realization are decidable. The assertional language is similar to languages normally used in such systems. The reasoning component of $\mathcal{KRIS}$ depends on sound and complete algorithms for reasoning facilities such as consistency checking, subsumption checking, retrieval, and querying.

Contents

1 Introduction and Motivation

In the last decade many knowledge representation systems in the tradition of KL-ONE [BS85] have been built, for example BACK [NvL88, Neb90], CLASSIC [BBMR89], KANDOR [Pat84], KL-TWO [Vil85], KRYPTON [BPGL85], LOOM [MB87], NIKL [KBR86], SB-ONE [Kob89]. A common feature of these systems is the separation of the knowledge into a terminological part and an assertional part. Knowledge about classes of individuals and relationships between these classes is stored in the *TBox*, and knowledge concerning particular individuals can be described in the *ABox*.

The TBox formalism provides a *concept language* (or term subsumption language) for the definition of concepts and roles, where concepts are interpreted as sets of individuals and roles as binary relations between individuals. Starting with primitive concepts and roles the language formalism is used to build up more complex concepts and roles.

For example, assume that `person`, `female`, and `shy` are primitive concepts, and `child` and `female_relative` are primitive roles. Taking the connectives concept conjunction (and), disjunction (or), and negation (not) one can express "persons who are female or not shy" by

(and `person` (or `female` (not `shy`))).

Since concepts are interpreted as sets, concept conjunction can be interpreted as set intersection, concept disjunction as set union, and negation of concepts as set complement. In addition to these operations on sets one can also employ roles for the definition of new concepts. *Value restrictions* can be used for instance to describe "individuals for whom all children are female" by the expression (all `child` `female`). *Number restrictions* allow for instance to describe "individuals having at most three children" by the expression (atmost 3 `child`). Beside the above mentioned constructs there are other well-known concept-forming constructs which are available in $\mathcal{KRIS}$ (see Section 2). An example for a role-forming construct is the conjunction of roles. We can define the role (and `child` `female_relative`), which intuitively yields the role daughter. The concept language presented in the next section also provides functional roles, so-called *attributes*. These attributes are interpreted as partial functions and not as arbitrary binary relations. Natural examples for attributes may be `father` or `first_name`. An *agreement* between two attribute chains for example allows to describe "individuals whose father and grandfather have the same first name" by the expression

(equal (compose `father` `first_name`) (compose `father` `father` `first_name`)).

Interestingly, agreements between attribute chains do not make reasoning in the language undecidable[HN90], whereas agreements between arbitrary role chains cause undecidability [Sch89].

The basic reasoning facilities concerning the TBox are the determination whether a concept denotes nothing, i.e., whether a concept denotes the empty set in every interpretation, and the computation of the subsumption hierarchy. A concept C subsumes (is more general than) a concept D iff in every interpretation the set denoted by C is a superset of the set denoted by D.

The ABox formalism consists of an assertional language which allows the introduction of individuals to express facts about a concrete world. One can state that individuals are instances of concepts, and that pairs of individuals are instances of roles or attributes.

The reasoning facilities concerning both the TBox and the ABox are classified as follows. We need algorithms for inferences such as

- checking the consistency of the represented knowledge,
- given an individual of the ABox, compute the most specific concepts in the TBox this individual is instance of,
- computing all individuals of the ABox that are instances of a given concept.

The formal model-theoretic semantics which has been introduced for KL-ONE languages [BL84] provides means for investigating soundness and completeness of inference algorithms. It turned out that the above mentioned systems use sound but incomplete algorithms. If a sound but incomplete subsumption algorithm detects a subsumption relation, this relation really exists; but if it fails to recognize that a concept subsumes another one, then we do not know anything. A subsumption relation may or may not exist. Thus, the results of the algorithms only partially coincides with what the formal semantics expresses.[1]

Until recently, sound *and* complete algorithms for the above mentioned inferences and for the subsumption problem were only known for rather trivial languages which explains the use of incomplete algorithms in existing KL-ONE systems. Another argument in favour of incomplete algorithms was that for many languages the subsumption problem is at least NP-hard [LB87, Neb88]. Consequently, complete algorithms have to be intractable, whereas incomplete algorithms may still be polynomial. However, one should keep in mind that these complexity results are worst case results. It is not at all clear how complete algorithms may behave for typical knowledge bases.

In [SS88, HNS90, Hol90a] it is shown how to devise sound and complete algorithms for the above mentioned inferences in various concept languages. Thus it has become possible to implement a KL-ONE system ($\mathcal{KRIS}$) which provides

- a very expressive concept language,
- powerful reasoning facilities, and
- sound and complete algorithms for these facilities.

The purpose of this paper is as follows. Firstly, we will enumerate the language constructs which are available in $\mathcal{KRIS}$, and will give a formal semantics for their meaning. We have chosen the concept language such that it contains most of the constructs used in KL-ONE systems with the obvious restriction that the interesting inferences such as consistency checking, subsumption checking, and realization are decidable. Of course, taking such a large language means that the complexity of the inference algorithms is relatively high. But $\mathcal{KRIS}$ also provides faster algorithms for certain sublanguages.[2] Secondly, we will describe the inference mechanisms provided by $\mathcal{KRIS}$. Finally, we will give an overview of the implemented $\mathcal{KRIS}$ system.

---

[1] But see Patel-Schneider who uses a four-valued semantics to formally describe the behaviour of an algorithm which is incomplete w.r.t. two-valued semantics.

[2] That coincides with what Ramesh Patil proposed at the Workshop on Term Subsumption Languages in Knowledge Representation: "He therefore strongly opposed any attempt to further restrict the expressiveness of TSL (term subsumption language) systems. Instead, he proposed that such systems be configured on a "pay as you go" basis—if the application uses only a small portion of the expressive power of the TSL, then everything will be fast; if more expressive power is used, then the system may slow down, but still be able to represent and reason with the knowledge given to it." (see [PSO*90]).

2 Formalisms for Representing Knowledge

In this section we will introduce the formalisms for representing knowledge in $\mathcal{KRIS}$. In Subsection 2.1 the syntax and semantics of the concept language and terminological axioms is presented. In Subsection 2.2 the assertional language and its semantics is introduced.

2.1 The Concept Language Underlying $\mathcal{KRIS}$

Assume that we have three disjoint alphabets of symbols, called *concept names*, *role names*, and *attribute names*. The special concept name *top* is called *top concept.*

The sets of *concept terms*, *role terms*, and *attribute terms* are inductively defined as follows. Every concept name is a concept term, every role name is a role term, and every attribute name is an attribute term. Now let C, $C_1, \ldots, C_k$ be concept terms, R, $R_1, \ldots, R_l$ be role terms, f, g, $f_1, \ldots, f_m$ be attribute terms already defined, and let n be a nonnegative integer. Then

(and $C_1 \ldots C_k$),	(conjunction)
(or $C_1 \ldots C_k$),	(disjunction)
(not C),	(negation)
(all R C), (all f C),	(value restriction)
(some R C), (some f C),	(exists restriction)
(atleast n R) (atmost n R)	(number restrictions)
(equal f g),	(agreement)
(not-equal f g)	(disagreement)

are concept terms,

(and $R_1 \ldots R_l$),	(role conjunction)
(restr R C),	(restriction)

are role terms, and

(and $f_1 \ldots f_m$),	(attribute conjunction)
(compose $f_1 \ldots f_m$)	(composition)

are attribute terms.

So-called *terminological axioms* are used to introduce names for concept, role, and attribute terms. A finite set of such axioms satisfying certain restrictions is called a terminology (TBox). There are three different ways of introducing new concepts (respectively roles or attributes) into a terminology.

Let A (P, f) be a concept (role, attribute) name, and let C (R, g) be a concept (role, attribute) term. By the terminological axioms

(defprimconcept A), (defprimrole P), (defprimattribute f)

new concept, role, and attribute names are introduced without restricting their interpretation. The terminological axioms

(defprimconcept A C), (defprimrole P R), (defprimattribute f g)

impose necessary conditions on the interpretation of the introduced concept, role, and attribute names. Finally, one can impose necessary and sufficient conditions by the terminological axioms

$$(\textsf{defconcept}\ A\ C),\quad (\textsf{defrole}\ P\ R),\quad (\textsf{defattribute}\ f\ g).$$

A *terminology* (*TBox*) $\mathcal{T}$ is a finite set of terminological axioms with the additional restriction that (i) every concept, role, and attribute name may appear at most once as a first argument of a terminological axiom in $\mathcal{T}$ (unique definition), and (ii) $\mathcal{T}$ must not contain cyclic definitions (acyclicity). For a discussion of terminological cycles see [Neb88, Baa90a].

A terminology which describes knowledge about persons and relationships between persons is shown in Figure 1. At first, the attribute `sex` and the concept `male` is intro-

```
(defprimattribute sex)
(defprimconcept male)
(defprimconcept female (not male))
(defprimconcept person (some sex (or male female)))
(defprimrole child)
(defconcept parent (and person (some child person)))
(defconcept mother (and parent (some sex female)))
(defconcept father (and parent (not mother)))
(defconcept grandparent (and parent (some child parent)))
(defconcept parent_with_two_children (and parent (atleast 2 child)))
(defconcept parent_with_sons_only (and parent (all child (some sex male))))
```

Figure 1: A terminology (TBox).

duced. The axioms which define the concepts `female` and `person` can be read as follows: "no individual is both male and female"[3], and "a person has sex male or female." These axioms impose necessary conditions on the interpretation of the introduced concepts. The definition of the concept `parent` impose necessary and sufficient conditions: "an individual is a parent if and only if it is a person and has some child who is a person." The other concepts are defined such that they correspond to their intuitive meaning.

We will now give a formal model-theoretic semantics for the concept language and the terminological axioms. An *interpretation* $\mathcal{I}$ consists of a set $\Delta^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$). The interpretation function maps every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, every role name P to a subset $P^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every attribute name f to a partial function $f^{\mathcal{I}}$ from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}}$. With $dom\, f^{\mathcal{I}}$ we denote the domain of the partial function $f^{\mathcal{I}}$ (i.e., the set of elements of $\Delta^{\mathcal{I}}$ for which f is defined).

The interpretation function—which gives an interpretation for concept, role, and attribute names—can be extended to concept, role, and attribute terms as follows. Let C, $C_1, \ldots, C_k$ be concept terms, R, $R_1, \ldots, R_l$ role terms, f, g, $f_1, \ldots, f_m$ attribute terms, and let n be a nonnegative integer. Assume that $C^{\mathcal{I}}$, $C_1^{\mathcal{I}}, \ldots, C_k^{\mathcal{I}}$, $R^{\mathcal{I}}$, $R_1^{\mathcal{I}}, \ldots, R_l^{\mathcal{I}}$, $f^{\mathcal{I}}$, $g^{\mathcal{I}}$,

[3] A more intensional input language which, for instance, provides disjointness axioms, could of course be useful (cf. Summary and Outlook).

$f_1^{\mathcal{I}}, \ldots, f_m^{\mathcal{I}}$ are already defined. Then

$$
\begin{array}{rcl}
(\mathsf{*top*})^{\mathcal{I}} & := & \Delta^{\mathcal{I}} \\
(\mathsf{and}\ C_1 \ldots C_k)^{\mathcal{I}} & := & C_1^{\mathcal{I}} \cap \ldots \cap C_k^{\mathcal{I}} \\
(\mathsf{or}\ C_1 \ldots C_k)^{\mathcal{I}} & := & C_1^{\mathcal{I}} \cup \ldots \cup C_k^{\mathcal{I}} \\
(\mathsf{not}\ C)^{\mathcal{I}} & := & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\mathsf{all}\ R\ C)^{\mathcal{I}} & := & \{\, a \in \Delta^{\mathcal{I}} \mid \forall b : (a,b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}} \} \\
(\mathsf{all}\ f\ C)^{\mathcal{I}} & := & \{\, a \in \Delta^{\mathcal{I}} \mid a \in \mathit{dom}\, f^{\mathcal{I}} \Rightarrow f^{\mathcal{I}}(a) \in C^{\mathcal{I}} \} \\
(\mathsf{some}\ R\ C)^{\mathcal{I}} & := & \{\, a \in \Delta^{\mathcal{I}} \mid \exists b : (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \} \\
(\mathsf{some}\ f\ C)^{\mathcal{I}} & := & \{\, a \in \mathit{dom}\, f^{\mathcal{I}} \mid f^{\mathcal{I}}(a) \in C^{\mathcal{I}} \} \\
(\mathsf{atleast}\ n\ R)^{\mathcal{I}} & := & \{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}| \geq n\} \\
(\mathsf{atmost}\ n\ R)^{\mathcal{I}} & := & \{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}| \leq n\} \\
(\mathsf{equal}\ f\ g)^{\mathcal{I}} & := & \{\, a \in \mathit{dom}\, f^{\mathcal{I}} \cap \mathit{dom}\, g^{\mathcal{I}} \mid f^{\mathcal{I}}(a) = g^{\mathcal{I}}(a)\} \\
(\mathsf{not\text{-}equal}\ f\ g)^{\mathcal{I}} & := & \{\, a \in \mathit{dom}\, f^{\mathcal{I}} \cap \mathit{dom}\, g^{\mathcal{I}} \mid f^{\mathcal{I}}(a) \neq g^{\mathcal{I}}(a)\} \\
(\mathsf{and}\ R_1 \ldots R_l)^{\mathcal{I}} & := & R_1^{\mathcal{I}} \cap \ldots \cap R_l^{\mathcal{I}} \\
(\mathsf{restr}\ R\ C)^{\mathcal{I}} & := & \{\, (a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \} \\
(\mathsf{and}\ f_1 \ldots f_m)^{\mathcal{I}} & := & f_1^{\mathcal{I}} \cap \ldots \cap f_m^{\mathcal{I}} \\
(\mathsf{compose}\ f_1 \ldots f_m)^{\mathcal{I}} & := & f_1^{\mathcal{I}} \circ \ldots \circ f_m^{\mathcal{I}},
\end{array}
$$

where $|X|$ denotes the cardinality of the set X and $\circ$ denotes the composition of functions. Note, that if $f_1^{\mathcal{I}}, \ldots, f_m^{\mathcal{I}}$ are partial functions, then $f_1^{\mathcal{I}} \cap \ldots \cap f_m^{\mathcal{I}}$ and $f_1^{\mathcal{I}} \circ \ldots \circ f_m^{\mathcal{I}}$ are also partial functions.

The semantics of terminological axioms is now defined as follows. An interpretation $\mathcal{I}$ *satisfies* the terminological axiom

$$
\begin{array}{lll}
(\mathsf{defprimconcept}\ A\ C) & \text{iff} & A^{\mathcal{I}} \subseteq C^{\mathcal{I}}, \\
(\mathsf{defconcept}\ A\ C) & \text{iff} & A^{\mathcal{I}} = C^{\mathcal{I}}, \\
(\mathsf{defprimrole}\ P\ R) & \text{iff} & P^{\mathcal{I}} \subseteq R^{\mathcal{I}}, \\
(\mathsf{defrole}\ P\ R) & \text{iff} & P^{\mathcal{I}} = R^{\mathcal{I}}, \\
(\mathsf{defprimattribute}\ f\ g) & \text{iff} & f^{\mathcal{I}} \subseteq g^{\mathcal{I}}, \\
(\mathsf{defattribute}\ f\ g) & \text{iff} & f^{\mathcal{I}} = g^{\mathcal{I}},
\end{array}
$$

where A (P, f) is a concept (role, attribute) name, and C (R, g) is a concept (role, attribute) term. Note that the terminological axioms (**defprimconcept** A), (**defprimrole** P), and (**defprimattribute** f) are satisfied in every interpretation by the definition of interpretation. An interpretation $\mathcal{I}$ is a *model* for a TBox $\mathcal{T}$ iff $\mathcal{I}$ satisfies all terminological axioms in $\mathcal{T}$.

2.2 Assertions

The assertional formalism allows to introduce objects (individuals). We can describe a concrete world by stating that objects are instances of concepts, and that pairs of objects are instances of roles or attributes.

Assume that we have a further alphabet of symbols, called *object names.* Names for objects are introduced by *assertional axioms* which have the form

$$(\mathsf{assert\text{-}ind}\ a\ C), \quad (\mathsf{assert\text{-}ind}\ a\ b\ R), \quad (\mathsf{assert\text{-}ind}\ a\ b\ g),$$

where a, b are object names, and C (R, g) is a concept (role, attribute) term. A *world description* (*ABox*) is a finite set of assertional axioms.

Figure 2 shows an example of an ABox. This ABox describes a world in which `Tom` is

(assert-ind Tom father)	
(assert-ind Tom Peter child)	(assert-ind Tom Harry child)
(assert-ind Mary parent_with_sons_only)	
(assert-ind Mary Tom child)	(assert-ind Mary Chris child)

Figure 2: A world description (ABox).

father of Peter and Harry. Furthermore, Mary has only sons; two of them are Tom and Chris.

Note that an ABox can be considered as a relational database where the arity of each tuple is either one or two. However, in contrast to the closed world semantics which is usually employed in databases, we assume an *open world semantics*, since we want to allow for incomplete knowledge. Thus, we cannot conclude in the above example that Tom has exactly two children, since there may exist a world in which Tom has some additional children.

The semantics of object names and assertional axioms is defined as follows. The interpretation function $\cdot^{\mathcal{I}}$ of a TBox interpretation $\mathcal{I}$ can be extended to object names by mapping them to elements of the domain such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$. This restriction on the interpretation function ensures that objects with different names denote different individuals in the world. It is called *unique name assumption*, which is usually also assumed in the database world.

Let a, b be object names, C (R, g) be a concept (role, attribute) term. An interpretation $\mathcal{I}$ *satisfies* the assertional axiom

$$\begin{array}{lll} (\text{assert-ind } a\ C) & \text{iff} & a^{\mathcal{I}} \in C^{\mathcal{I}} \\ (\text{assert-ind } a\ b\ R) & \text{iff} & (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \\ (\text{assert-ind } a\ b\ f) & \text{iff} & f^{\mathcal{I}}(a^{\mathcal{I}}) = b^{\mathcal{I}}. \end{array}$$

The semantics of an ABox together with a TBox is defined as follows. We say that an interpretation $\mathcal{I}$ is a *model* for an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all assertional axioms in $\mathcal{A}$ and all terminological axioms in $\mathcal{T}$.

3 Reasoning

In this section we describe the inference mechanisms provided by $\mathcal{KRIS}$. The reasoning component of $\mathcal{KRIS}$ allows to make knowledge explicit which is only implicitly represented in an ABox and a TBox. For example, from the TBox and Abox given in the previous section one can conclude that Mary is a grandparent, though this knowledge is not explicitly stored in the ABox.

An obvious requirement on the represented knowledge is that it should be consistent since everything would be deducible from inconsistent knowledge (from a logical point of view). If, for example, an ABox contains the axioms (assert-ind Chris mother) and (assert-ind Chris father), then the system should detect this inconsistency.[4] The underlying model-theoretic semantics allows a clear and intuitive definition of consistency. We say that an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ is *consistent* if it has a model. Thus, we have the

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

[4]However, in general it is not always as easy as in this example to check whether the represented knowledge is consistent.

Beside an algorithm for checking consistency of an ABox w.r.t. a TBox $\mathcal{KRIS}$ provides algorithms for the basic reasoning facilities such as subsumption and instantiation.

We say that a concept term C *subsumes* a concept term D iff $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$ in every interpretation $\mathcal{I}$. Thus, an algorithm for checking subsumption takes concept terms C and D as arguments and has to solve the

Subsumption problem: Does C subsume D ?

The subsumption problem in concept languages has been thoroughly investigated in [SS88, HNS90, DHL*90]. In these papers subsumption algorithms for various concept languages and sublanguages are given and their computational complexity is discussed. In fact, the papers do not directly describe subsumption algorithms but algorithms for a closely related problem. These algorithms check whether a given concept term C is meaningful, i.e., whether there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$. Since C subsumes D if and only if $(\mathsf{and}\ D\ (\mathsf{not}\ C))^{\mathcal{I}} = \emptyset$ for every interpretation $\mathcal{I}$, these algorithms can also be used to decide subsumption.

An algorithm for instantiation decides whether an assertional axiom is deducible from the represented knowledge. More formally, let α be an assertional axiom. We say that an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ *implies* α iff all models for $\mathcal{A}$ w.r.t. $\mathcal{T}$ satisfy α, written $\mathcal{A}, \mathcal{T} \models \alpha$. Thus we define the

Instantiation problem: Is α implied by $\mathcal{A}$ and $\mathcal{T}$?

If α is of the form $(\mathsf{assert\text{-}ind}\ a\ b\ R)$ or $(\mathsf{assert\text{-}ind}\ a\ b\ f)$, then it is relatively easy to solve the instantiation problem since the concept language allows only few constructs to build complex role or attribute terms. If α is of the form $(\mathsf{assert\text{-}ind}\ a\ C)$, the instantiation problem can be reduced to the consistency problem using the well-known deduction theorem. In [Hol90a] a sound and complete algorithm for the consistency and instantiation problem is given.

$\mathcal{KRIS}$ also provides algorithms which find out certain relationships between the defined concepts, roles, attributes, and objects. These algorithms are based on the algorithms for subsumption and instantiation. Assume that $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.

The *subsumption hierarchy* is the preordering of the concept names in $\mathcal{T}$ w.r.t. the subsumption relation. The so-called *classifier* has to solve the

Classification problem: Compute the subsumption hierarchy.

Given an object in $\mathcal{A}$, one wants to know the set of concept names in $\mathcal{T}$ which describe it most accurately. To be more formal, let a be an object occurring in $\mathcal{A}$. The set of *most specialized concepts* of a is a set $\{A_1, \ldots, A_n\}$ of concept names occurring in $\mathcal{T}$ such that

1. $\mathcal{A}, \mathcal{T} \models (\mathsf{assert\text{-}ind}\ a\ A_i)$,
2. for every A_i there does not exist a concept name A in $\mathcal{T}$ such that $\mathcal{A}, \mathcal{T} \models (\mathsf{assert\text{-}ind}\ a\ A)$, A_i subsumes A, and A does not subsume A_i, and
3. for every concept name A in $\mathcal{T}$ such that $\mathcal{A}, \mathcal{T} \models (\mathsf{assert\text{-}ind}\ a\ A)$, there exists an A_i such that A subsumes A_i.

The first condition means that each A_i is in fact a description of a. The second condition guarantees that the set contains only the smallest description w.r.t. the subsumption relation, and the third condition means that we do not omit any nonredundant description. Thus, to describe an object most accurately we need an algorithm for the

Realization problem: Compute for an object in $\mathcal{A}$ the set of most specialized concepts in $\mathcal{T}$.

Conversely, we want to know the objects of $\mathcal{A}$ which are instances of a given concept term. Let C be a concept term. The set *INST*(C) contains all the objects $a_1, \ldots, a_n$ of $\mathcal{A}$ such that $\mathcal{A}, \mathcal{T} \models$ (assert-ind a_i C) holds. We have the

Retrieval problem: Compute for a given concept term the set *INST*(C).

4 $\mathcal{KRIS}$: the Overall Structure

In this section we give a short description of $\mathcal{KRIS}$. The representation component offers the formalisms presented in Section 2: a very expressive concept language and an assertional language which is similar to the languages used in most KL-ONE systems. The reasoning component of $\mathcal{KRIS}$ provides sound and complete algorithms which solve the problems mentioned in the previous section.

$\mathcal{KRIS}$ is implemented in Symbolics Common Lisp on a Symbolics Lisp machine. The main menu of $\mathcal{KRIS}$ is shown in Figure 3.

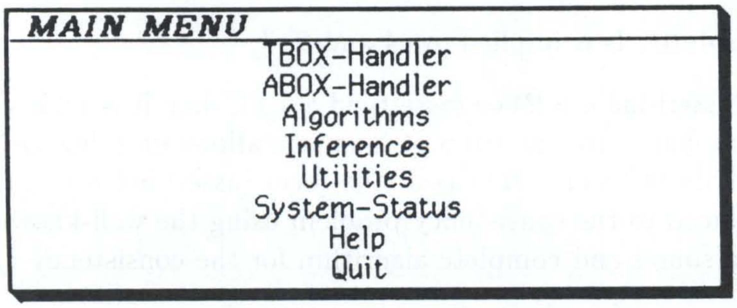

Figure 3: $\mathcal{KRIS}$ main menu.

Clicking one of the menu items causes $\mathcal{KRIS}$ to generate submenus. They allow the following operations.

- The *TBox-Handler* organizes the treatment of terminologies. That means, it can be used to create, load, edit, and delete TBoxes.
- Similarly, the *ABox-Handler* manages ABoxes.
- The item *Algorithms* allows to choose an appropriate algorithm. We have implemented several algorithms for the inferences which are based on different data-structures. Furthermore, for some sublanguages of the concept language presented in Section 2 we have implemented optimized algorithms.
- We can start a chosen algorithm using *Inferences*. $\mathcal{KRIS}$ provides algorithms which solve the consistency problem, the subsumption problem, the instantiation problem, the classification problem, the realization problem, and the retrieval problem.
- *Utilities* provides possibilities to measure the run-time of algorithms.
- *Help* and *System-Status* give more informations about the system.

$\mathcal{KRIS}$ can be used as follows. First of all, the user has to edit the terminological and assertional knowledge of the domain of interest using *TBox-Handler* and *ABox-Handler*. Assume that the TBox of Figure 1 and the ABox of Figure 2 have been edited, and hence are known to $\mathcal{KRIS}$. The consistency algorithm will find out that the represented knowledge is consistent. That means, there exists a model for the ABox w.r.t. the TBox. The classification algorithm computes the subsumption hierarchy as shown in Figure 4.

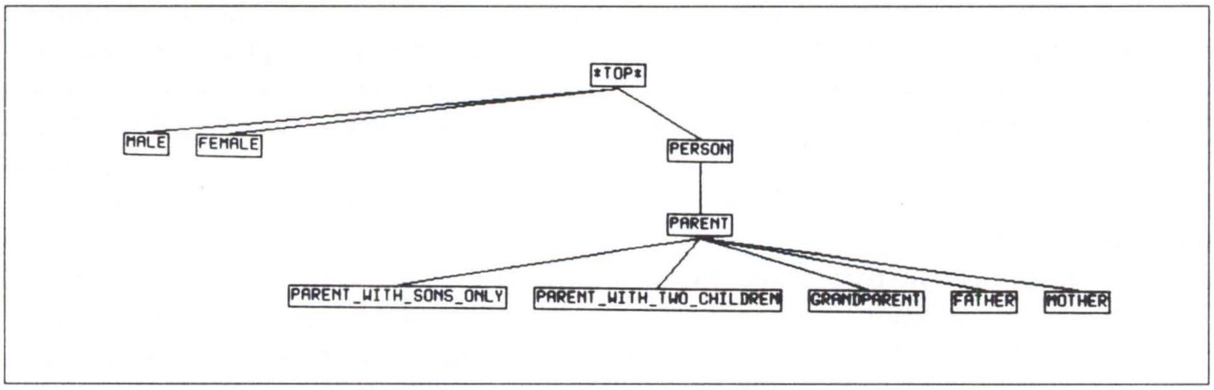

Figure 4: The subsumption hierarchy of the TBox given in Figure 2.

One can use the instantiation algorithm to get the most accurate information about an object. For example, the algorithm will detect the following relationships:

object	most specialized concepts
`Tom`	`father, parent_with_two_children`
`Mary`	`parent_with_two_children, grandparent, parent_with_sons_only`

The retrieval algorithm computes for a given concept term the objects of the ABox which are instances of it:

concept term	objects
`grandparent`	`Mary`
`parent_with_two_children`	`Mary, Tom`
`(some sex male)`	`Tom, Chris`

That means, for instance, (i) the fact that `Tom` and `Chris` have sex male is implied by the represented knowledge, and (ii) for the other objects in $\mathcal{A}$ this property cannot be concluded.

The user may cause $\mathcal{KRIS}$ to compute for a given TBox and ABox (i) the subsumption hierarchy, (ii) for every object in the ABox the most specialized concepts, and (iii) for every concept name in the TBox the objects which are instances of it.[5] After $\mathcal{KRIS}$ has once determined these structures, it is able to access this information efficiently. Note that only a small amount of memory is needed to store this information. Consequently, the subsumption problem and the retrieval problem for concepts defined in the TBox, and the instantiation problem can afterwards be solved very fast by looking into the precomputed structures.

At any time the user may add terminological and assertional axioms to an already existing TBox and ABox. Assume that $\mathcal{KRIS}$ has computed the structures mentioned

[5]The idea that some of the important inferences can be computed in advance was already used in the original KL-ONE system. Cf. [BS85] p. 178: "In KL-ONE the network (i.e. the subsumption hierarchy) is computed first from the forms of descriptions, and subsumption questions are always read off from the hierarchy."

before. In this case $\mathcal{KRIS}$ gives the user the possibility to update these structures. If a terminological axiom is added, then, for instance, the subsumption hierarchy is enlarged by the inserting concept name defined by the axiom at the appropriate place.

5 Summary and Outlook

The $\mathcal{KRIS}$ system which has been presented in this paper distinguishes itself from all the other implemented KL-ONE based systems in that it employs complete inference algorithms. Nevertheless its concept language is relatively large. Of course, the price one has to pay is that the worst case complexity of the algorithms is worse than NP. But it is not clear whether the behaviour for "typical" knowledge bases is also that bad. An important reason for implementing the $\mathcal{KRIS}$ system was that it could be used to investigate this question.

Thus an important part of our future work will be to test the system with typical applications. In addition, we intent to further extend the system. On the one hand, we want to integrate the possibility to refer to concrete domains (such as integers, real numbers, strings, etc.) in the definition of concepts [BH90]. On the other hand, we will allow further concept forming operators such as qualified number restrictions [Hol90b] and role forming operators such as transitive closure of roles [Baa90a] (at least for a sublanguage of the presented concept language); for additional constructs see [BBHH*90].

Another point is that until now the user has to specify, which algorithm should be used. In an improved $\mathcal{KRIS}$ version, this system will itself choose the optimal algorithm by inspecting what combination of language constructs are used.

The main objective of our research group WINO—as a part of the larger project AKA (Autonomous Cooperating Agents)—is the investigation of logical foundations of knowledge representation formalisms which can be used for applications in cooperating agent scenarios [BM90]. Thus our long term goals also comprise further extensions of $\mathcal{KRIS}$ such as

- a constrained-based approach for integrating full first order predicate logics with concept languages [BBHNS90, Bür90] which can be used to represent non-taxonomical knowledge,

- modal-logical approaches for the integration of knowledge concerning time and space.

Acknowledgements. We would like to thank Erich Achilles, Armin Laux, and Jörg Peter Mohren for their implementational work. This research was supported by the German Bundesministerium für Forschung und Technologie under grant ITW 8903 0.

References

[Baa90a] F. Baader. *Terminological Cycles in KL-ONE-based Knowledge Representation Languages.* In *Proceedings of the 8th National Conference of the AAAI*, pp. 621-626, Boston, Mas., 1990.

[Baa90b] F. Baader. *Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles.* DFKI Research Report, DFKI, Postfach 2080, D-6750 Kaiserslautern, West Germany. Forthcoming.

[BBHH*90] F. Baader, H.-J. Bürckert, J. Heinsohn, B. Hollunder, J. Müller, B. Nebel, W. Nutt, H.-J. Profitlich. *Terminological Knowledge Representation: A Proposal for a Terminological Logic.* To appear.

[BBHNS90] F. Baader, H.-J. Bürckert, B. Hollunder, W. Nutt, J. H. Siekmann. "Concept Logics" To appear in *Proceedings of the Symposium on Computational Logics*, Brüssel, November 1990.

[BH90] F. Baader, P. Hanschke. *A Schema for Integrating Concrete Domains into Concept Languages.* DFKI Research Report, DFKI, Postfach 2080, D-6750 Kaiserslautern, West Germany. Forthcoming.

[BBMR89] A. Borgida, R. J. Brachman, D. L. McGuinness, L. A. Resnick. "CLASSIC: A Structural Data Model for Objects." In *Proceedings of the International Conference on Management of Data*, Portland, Oregon, 1989.

[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.

[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.

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

[Bür90] H.-J. Bürckert. "A Resolution Principle for Clauses with Constraints" In *Proceedings of the 10th International Conference on Automated Deduction*, Lecture Notes in Artificial Intelligence, LNAI 449, Springer Verlag, pp. 178-192, 1990.

[BM90] H.-J. Bürckert, J. Müller. "RATMAN: A Rational Agent Testbed for Multi Agent Networks", To appear in *Proceedings of Modeling Autonomous Agents in Multi-Agent Worlds*, Elsevier Publishers, 1991.

[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.

[Hol90a] 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.

[Hol90b] B. Hollunder. *Qualified Number Restrictions in Concept Languages.* DFKI Research Report, DFKI, Postfach 2080, D-6750 Kaiserslautern, West Germany. Forthcoming.

[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.

[HNS90] B. Hollunder, W. Nutt, M. Schmidt-Schauß. "Subsumption Algorithms for Concept Description Languages." In *Proceedings of the 9th European Conference on Artificial Intelligence*, pp. 348–353, Stockholm, Sweden, 1990.

[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.

[Kob89] A. Kobsa. "The SB-ONE knowledge representation workbanch" In *Preprints of the Workshop on Formal Aspects of Semantic Networks*, Two Harbors, Cal., February 1989.

[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.

[Neb90] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, Lecture Notes in Artificial Intelligence, LNAI 422, Springer Verlag, 1990.

[Neb89] B. Nebel. "Terminological Cycles: Semantics and Computational Properties." In *Proceedings of the Workshop on Formal Aspects of Semantic Networks*, Two Harbors, Cal., February 1989.

[Neb88] B. Nebel. "Computational complexity of terminological reasoning in BACK." *Artificial Intelligence*, 34(3):371–383, 1988.

[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.

[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.

[PSO*90] P. Patel-Schneider, B. Owsnicki-Klewe, A. Kobsa, N. Guarino, R. MacGregor, W. S. Mark, D. L. McGuinness, B. Nebel, A. Schmiedel, J. Yen. "Term Subsumption in Knowledge Representation." In *AI Magazine*, 11(2):16-23, 1990. pp. 11–16, Denver, Colo., 1984.

[Sch89] M. Schmidt-Schauß. "Subsumption in KL-ONE is undecidable." In R. J. Bachmann, H. J. Levesque, R. Reiter (editors), *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pp. 421–431, Toronto, Ont., 1989.

[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, 47, 1991.

[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

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

RR-90-03
Andreas Dengel & Nelson M. Mattos
Integration of Document Representation, Processing and Management
18 pages

RR-90-04
Bernhard Hollunder & Werner Nutt
Subsumption Algorithms for Concept Languages
34 pages

RR-90-05
Franz Baader
A Formal Definition for the Expressive Power of Knowledge Representation Languages
22 pages

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

RR-90-07
Elisabeth André, Thomas Rist
Wissensbasierte Informationspräsentation: Zwei Beiträge zum Fachgespräch Graphik und KI:

1. Ein planbasierter Ansatz zur Synthese illustrierter Dokumente
2. Wissensbasierte Perspektivenwahl für die automatische Erzeugung von 3D-Objektdarstellungen

24 pages

RR-90-08
Andreas Dengel
A Step Towards Understanding Paper Documents
25 pages

RR-90-09
Susanne Biundo
Plan Generation Using a Method of Deductive Program Synthesis
17 pages

RR-90-10
Franz Baader, Hans-Jürgen Bürckert, Bernhard Hollunder, Werner Nutt, Jörg H. Siekmann
Concept Logics
26 pages

RR-90-11
Elisabeth André, Thomas Rist
Towards a Plan-Based Synthesis of Illustrated Documents
14 pages

RR-90-12
Harold Boley
Declarative Operations on Nets
43 pages

DFKI Technical Memos

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

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

TM-90-02
Jay C. Weber
The Myth of Domain-Independent Persistence
18 pages

TM-90-03
Franz Baader, Bernhard Hollunder
KRIS: Knowledge Representation and Inference System
-System Description-
15 pages

DFKI Documents

D-89-01
Michael H. Malburg & Rainer Bleisinger
HYPERBIS: ein betriebliches Hypermedia-Informationssystem
43 Seiten

D-90-01
DFKI Wissenschaftlich-Technischer Jahresbericht 1989
45 pages

D-90-02
Georg Seul
Logisches Programmieren mit Feature - Typen
107 Seiten

D-90-03
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
Abschlußbericht des Arbeitspaketes PROD
36 Seiten

D-90-04
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
STEP: Überblick über eine zukünftige Schnittstelle zum Produktdatenaustausch
69 Seiten

KRIS :*K*nowledge *R*epresentation and *Inf*erence *S*ystem
-System Description-

Franz Baader, Bernhard Hollunder

TM-90-03
Technical Memo