Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report

RR-93-40

Queries, Rules and Definitions as Epistemic Statements in Concept Languages

Francesco M. Donini, Maurizio Lenzerini,
Daniele Nardi, Werner Nutt, Adrea Schaerf

October 1993

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, 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 and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, Sema Group, 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
Computer Linguistics
Programming Systems
Deduction and Multiagent Systems
Document Analysis and Office Automation.

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.

Dr. Dr. D. Ruland
Director

Queries, Rules and Definitions as Epistemic Sentences in Concept Languages

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Adrea Schaerf

DFKI-RR-93-40

This paper will also be published in: G. Lakemeyer (editor), The Theoretical Foundations of Knowledge Representation and Reasoning, LNAI, Springer Verlag, 1993.

This work has been supported ba a grant from The Federal Ministry for Research and Technology (FKZ ITWM-9201).

Queries, Rules and Definitions as Epistemic Sentences in Concept Languages

Francesco M. Donini* Maurizio Lenzerini*
Daniele Nardi* Werner Nutt† Andrea Schaerf*

Abstract

Concept languages have been studied in order to give a formal account of the basic features of frame-based languages. The focus of research in concept languages was initially on the semantical reconstruction of frame-based systems and the computational complexity of reasoning. More recently, attention has been paid to the formalization of other aspects of frame-based languages, such as non-monotonic reasoning and procedural rules, which are necessary in order to bring concept languages closer to implemented systems. In this paper we discuss the above issues in the framework of concept languages enriched with an epistemic operator. In particular, we show that the epistemic operator both introduces novel features in the language, such as sophisticated query formulation and closed world reasoning, and makes it possible to provide a formal account for some aspects of the existing systems, such as rules and definitions, that cannot be characterized in a standard first-order framework.

*Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, I-00198 Roma, Italy (email: {donini, lenzerini, nardi, aschaerf}@assi.ing.uniroma1.it)

†German Research Center for Artificial Intelligence (DFKI), Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany (email: nutt@dfki.uni-sb.de)

Contents

1 Introduction

Structured or taxonomical representations of knowledge have been studied in Artificial Intelligence with the aim of providing for both a compact representation and efficient reasoning methods. Semantic networks and frames are well known examples of this kind of knowledge representation languages.

Concept languages (also called terminological languages or description logics) have been studied for several years in order to provide a formalization of structured knowledge representation languages and to analyze the computational properties of the associated reasoning tasks [3, 5, 6, 19, 28]. However, concept languages are given a set-theoretic first-order semantics and leave out several aspects of practical systems. Therefore, it seems now appropriate to enrich such languages both to explore novel language features and to account for some of those aspects that cannot be easily described in a standard first-order framework.

This need is discussed in the literature (see for example [11, 29]) and can be easily recognized by looking at recent knowledge representation systems based on concept languages such as [2, 30]. Work in this direction has already begun with proposals of extending concept languages to deal with different forms of non-monotonic reasoning (see for example [1, 23]).

We proposed in [7] to enrich concept languages with an epistemic operator defined in the style of [14, 15, 25]. While the main emphasis of that paper was to show that answering queries formulated in the epistemic concept languages can be done by extending the calculus for instance checking developed in [10], here we aim at discussing in more detail the advantages provided by such an extension both for enhancing the capabilities of concept languages, and for formalizing non-standard features of existing systems.

In particular, we focus our attention on the use of the epistemic operator in order (1) to define a more powerful query language; (2) to be able to formulate queries requiring some forms of closed world reasoning; (3) to formalize the notion of procedural rule; (4) to precisely characterize weak forms of concept definition. All these aspects show that the epistemic operator turns out to be flexible enough to account for several different notions in an elegant and uniform way.

With regard to Point (1), we provide several examples that show how the new query language allows one to address both aspects of the external world as represented in the knowledge base, and aspects of what the knowledge base knows about the external world. It is worth noting that one advantage of the extended query language is the formalization of integrity constraints, which are viewed as sentences referring to what the knowledge base knows about the world (see [7]). This aspect, however, is not further discussed in the present paper.

With regard to Point (2), we show that a careful usage of the epistemic operator allows one to express queries whose processing forces the system to assume complete knowledge about (part of) the knowledge base. Note that this approach is different from assigning a closed world semantics to the knwoledge base itself. In fact, the nonmonotonicity is not in the semantics of the knowledge base, but a form of nonmonotonic reasoning is achieved by the system when answering special kinds of queries.

Points (3) and (4) are concerned with the formalization of two important features of some existing systems. In particular, systems like [2, 12, 22, 30] include suitable structures for the representation of procedural rules, enabling both behavioral models of objects and expertise in an application domain to be expressed. We propose to express procedural rules as special epistemic sentences in the knowledge base. While procedural rules are usually defined informally in existing systems, we show that a nice formalization of these features can be achieved in our framework, thus clarifying both their semantics, and their interaction with the other parts of the knowledge base. Moreover, we show that epistemic sentences provide an account for weak forms of concept definitions similar to those found in LOOM [16] and other systems. This formalization makes it clear that weak definitions provide a form of incomplete reasoning that is both computationally advantageous, and semantically well founded.

The paper is organized as follows. Section 2 recalls the basic notions about the concept language $\mathcal{ALC}$, which is a powerful concept language (including concept conjunction, disjunction, negation, as well as existential and universal quantification of roles), together with its usage in the definition of knowledge bases. Section 3 presents the epistemic concept language $\mathcal{ALCK}$, obtained by adding an epistemic operator to $\mathcal{ALC}$. Section 4 elaborates on the features of $\mathcal{ALCK}$ when used as a query language over knowledge bases expressed in $\mathcal{ALC}$. Section 5 focuses on some forms of closed world reasoning that can be expressed with the epistemic operator. Section 6 proposes a formalization of procedural rules as special classes of epistemic sentences, while Section 7 discusses the use of epistemic sentences in expressing weak forms of concept inclusions and definitions. Finally, conclusions are drawn in Section 8.

2 Concept Knowledge Bases

We make use of the concept language $\mathcal{ALC}$ (see [5, 28]) to define a knowledge base.[1] Like any concept language, $\mathcal{ALC}$ allows one to express the knowledge about the classes of interest in a particular application through the notions of *concept* and *role*. Intuitively, concepts represent the classes of objects in the domain to be modeled, while roles represent relationships between objects. Starting with primitive concepts and roles, one can construct complex expressions by means of various concept forming operators.

The syntax and semantics of $\mathcal{ALC}$ are as follows. We assume that two alphabets of symbols, one for *primitive concepts*, and one for *primitive roles*, are given. The

[1] Although we restrict our attention to $\mathcal{ALC}$, our framework can be applied to other languages as well.

letter A will always denote a primitive concept, and the letter P will denote a role, which in $\mathcal{ALC}$ is always primitive. The *concepts* (denoted by the letters C and D) of the language $\mathcal{ALC}$ are built out of primitive concepts and primitive roles according to the syntax rule:

$$
\begin{array}{rcll}
C, D & \longrightarrow & A \mid & \text{(primitive concept)} \\
& & \top \mid & \text{(top)} \\
& & \bot \mid & \text{(bottom)} \\
& & C \sqcap D \mid & \text{(intersection)} \\
& & C \sqcup D \mid & \text{(union)} \\
& & \neg C \mid & \text{(complement)} \\
& & \forall P.C \mid & \text{(universal quantification)} \\
& & \exists P.C & \text{(existential quantification).}
\end{array}
$$

We use parentheses whenever we have to disambiguate concept expressions. For example, we write $(\exists P.D) \sqcap E$ to indicate that the concept E is not in the scope of $\exists P$.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$) that maps every concept to a subset of $\Delta^{\mathcal{I}}$ and every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that the following equations are satisfied:

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\forall P.C)^{\mathcal{I}} & = & \{d_1 \in \Delta^{\mathcal{I}} \mid \forall d_2 : (d_1, d_2) \in P^{\mathcal{I}} \rightarrow \ d_2 \in C^{\mathcal{I}}\} \\
(\exists P.C)^{\mathcal{I}} & = & \{d_1 \in \Delta^{\mathcal{I}} \mid \exists d_2 : (d_1, d_2) \in P^{\mathcal{I}} \ \wedge \ d_2 \in C^{\mathcal{I}}\}.
\end{array}
$$

An interpretation $\mathcal{I}$ is a *model* for a concept C if $C^{\mathcal{I}}$ is nonempty. A concept is *satisfiable* if it has a model and *unsatisfiable* otherwise. We say that C is *subsumed* by D if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every interpretation $\mathcal{I}$.

In terminological systems, the knowledge base includes both an intensional part, called terminology or simply *TBox*, and an extensional part, called assertional box or simply *ABox*. The TBox is constituted by a set of *inclusion statements* of the form

$$C \sqsubseteq D$$

where C, D are concepts. Inclusion statements are interpreted in terms of set inclusion: an interpretation $\mathcal{I}$ satisfies $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a *model* for a TBox if it satisfies all of its inclusions. As pointed out in [4], inclusions are more general than definitions, since definitions like $A \doteq C$ can be expressed as

$A \sqsubseteq C$ and $C \sqsubseteq A$. Moreover, cyclic definitions are admitted and interpreted by the descriptive semantics [19].

The ABox is constituted by a set of assertions that specify either that an individual is instance of a concept or that a pair of individuals is instance of a role. Let $\mathcal{O}$ be an alphabet of symbols, called *individuals*. Syntactically, assertions are expressed in terms of *membership statements*, of the form

$$C(a)$$
$$P(a,b)$$

where a and b are individuals, C is a concept, and P is a role. $C(a)$ means that a is an instance of C, while $P(a,b)$ means that a is related to b by means of P. In order to give a formal semantics to assertions,the interpretation must be enriched with an injective function from $\mathcal{O}$ to $\Delta^{\mathcal{I}}$, i.e. each individual is associated with a unique domain element (Unique Name Assumption). Therefore an interpretation is now a triple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \gamma^{\mathcal{I}})$, and an assertion $C(a)$ is satisfied by $\mathcal{I}$ if $\gamma^{\mathcal{I}}(a) \in C^{\mathcal{I}}$. Similarly, an assertion $P(a,b)$ is satisfied by $\mathcal{I}$ if $(\gamma^{\mathcal{I}}(a), \gamma^{\mathcal{I}}(b)) \in P^{\mathcal{I}}$.

To summarize we define an $\mathcal{ALC}$-knowledge base as follows:

Definition 1 *An $\mathcal{ALC}$-knowledge base is a pair $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a set of inclusion statements, and $\mathcal{A}$ is a set of membership assertions, whose concepts and roles belong to the language $\mathcal{ALC}$. An interpretation $\mathcal{I}$ is a* model *for $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ if it is a model for both $\mathcal{T}$ and $\mathcal{A}$.*

We say that Σ is *satisfiable* if it has a model. The set of models of Σ is denoted as $\mathcal{M}(\Sigma)$. Σ *logically implies* σ (written $\Sigma \models \sigma$), where σ is either an inclusion statement or a membership assertion, if every model in $\mathcal{M}(\Sigma)$ satisfies σ.

The most common kind of query to a knowledge base Σ is asking whether $C(a)$ (or $P(a,b)$) is logically implied by Σ. Notice that the semantics associated with concept languages is an open world semantics: the answer to a query Q will be `YES` if Q is true in every model for Σ, `NO` if Q is false in every model, and `UNKNOWN` otherwise.

It is well known (see for example [4]) that query answering in $\mathcal{ALC}$-knowledge bases is reducible to satisfiability. A calculus for knowledge base satisfiability in $\mathcal{ALC}$ is presented in [9] and shown to be complete and terminating.

3 An Epistemic Concept Language

In this section we present the epistemic concept language $\mathcal{ALCK}$, previously introduced in [7], which is an extension of $\mathcal{ALC}$ with an epistemic operator. Generally speaking, we follow [25], and use $\mathbf{K}C$ to denote the set of individuals *known* to be instances of the concept C in every model for the knowledge base. The syntax of

$\mathcal{ALCK}$ is the following (where C, D denote concepts, R denotes a role, A denotes a primitive concept and P a primitive role):

$$
\begin{array}{rcll}
C, D & \longrightarrow & A \mid & \text{(primitive concept)} \\
 & & \top \mid & \text{(top)} \\
 & & \bot \mid & \text{(bottom)} \\
 & & C \sqcap D \mid & \text{(intersection)} \\
 & & C \sqcup D \mid & \text{(union)} \\
 & & \neg C \mid & \text{(complement)} \\
 & & \forall R.C \mid & \text{(universal quantification)} \\
 & & \exists R.C \mid & \text{(existential quantification)} \\
 & & \mathbf{K}C & \text{(epistemic concept)} \\
R & \longrightarrow & P \mid & \text{(primitive role)} \\
 & & \mathbf{K}P & \text{(epistemic role).}
\end{array}
$$

The semantics of $\mathcal{ALCK}$ is an adaptation to the framework of concept languages of the one proposed in [14, 15, 25]. As in the cited papers, some issues typical of first-order modal systems arise. Such issues concern the interpretation structures and are dealt with by the following assumptions:

- every interpretation is defined over a fixed domain, called Δ (Common Domain Assumption);
- for every interpretation the mapping from the individuals into the domain elements, called γ, is fixed (Rigid Term Assumption).

An *epistemic interpretation* is a pair $(\mathcal{I}, \mathcal{W})$ where $\mathcal{I}$ is an interpretation and $\mathcal{W}$ is a set of interpretations such that the following equations are satisfied:

$$
\begin{array}{rcl}
\top^{\mathcal{I},\mathcal{W}} & = & \Delta \\
\bot^{\mathcal{I},\mathcal{W}} & = & \emptyset \\
A^{\mathcal{I},\mathcal{W}} & = & A^{\mathcal{I}} \\
P^{\mathcal{I},\mathcal{W}} & = & P^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I},\mathcal{W}} & = & C^{\mathcal{I},\mathcal{W}} \cap D^{\mathcal{I},\mathcal{W}} \\
(C \sqcup D)^{\mathcal{I},\mathcal{W}} & = & C^{\mathcal{I},\mathcal{W}} \cup D^{\mathcal{I},\mathcal{W}} \\
(\neg C)^{\mathcal{I},\mathcal{W}} & = & \Delta \setminus C^{\mathcal{I},\mathcal{W}} \\
(\forall R.C)^{\mathcal{I},\mathcal{W}} & = & \{d_1 \in \Delta \mid \forall d_2.\, (d_1, d_2) \in R^{\mathcal{I},\mathcal{W}} \rightarrow d_2 \in C^{\mathcal{I},\mathcal{W}}\} \\
(\exists R.C)^{\mathcal{I},\mathcal{W}} & = & \{d_1 \in \Delta \mid \exists d_2.\, (d_1, d_2) \in R^{\mathcal{I},\mathcal{W}} \wedge\ d_2 \in C^{\mathcal{I},\mathcal{W}}\} \\
(\mathbf{K}C)^{\mathcal{I},\mathcal{W}} & = & \bigcap_{\mathcal{J} \in \mathcal{W}} (C^{\mathcal{J},\mathcal{W}}) \\
(\mathbf{K}P)^{\mathcal{I},\mathcal{W}} & = & \bigcap_{\mathcal{J} \in \mathcal{W}} (P^{\mathcal{J},\mathcal{W}}).
\end{array}
$$

Notice that, since the domain is fixed independently of the interpretation, it is meaningful to refer to the intersection of the extensions of a concept in different interpretations. It follows that $\mathbf{K}C$ is interpreted in $\mathcal{W}$ as the set of objects that are instances of C in every interpretation belonging to $\mathcal{W}$. In this sense, $\mathbf{K}C$ represents those objects *known* to be instances of C in $\mathcal{W}$. Notice also that if one discards $\mathbf{K}$ and $\mathcal{W}$ in the equations, one obtains the standard semantics of $\mathcal{ALC}$.

An $\mathcal{ALCK}$-knowledge base Ψ is a pair $\Psi = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a set of inclusion statements and $\mathcal{A}$ is a set of membership assertions, whose concepts and roles belong to the language $\mathcal{ALCK}$. The truth of inclusion statements and membership assertions in an epistemic interpretation is defined in a straightforward way. An *epistemic model* for Ψ is a pair $(\mathcal{I}, \mathcal{W})$, where $\mathcal{I} \in \mathcal{W}$ and $\mathcal{W}$ is any maximal set of interpretations such that for each $\mathcal{J} \in \mathcal{W}$, every sentence (inclusion or membership assertion) of Ψ is true in $(\mathcal{J}, \mathcal{W})$.

Notice that the semantics of an $\mathcal{ALCK}$-knowledge base could be equivalently defined in terms of an accessibility relation on a set of possible worlds. More specifically, the constraints posed by the semantic equations on $\mathbf{K}C$ and $\mathbf{K}P$, correspond to a structure of possible worlds each one connected with all the others. Therefore, the accessibility relation would be an equivalence relation, as in the modal system S5. However, the epistemic models of a knowledge base correspond to S5 models with a maximal set of worlds. In particular, if Σ is an $\mathcal{ALC}$-knowledge base, i.e. it does not contain epistemic operators, then its epistemic models are all the pairs $(\mathcal{I}, \mathcal{M}(\Sigma))$ for every $\mathcal{I} \in \mathcal{M}(\Sigma)$.

An $\mathcal{ALCK}$-knowledge base Ψ is said to be *satisfiable* if there exists an epistemic model for Ψ, *unsatisfiable* otherwise. Ψ logically implies an assertion σ, written $\Psi \models C(a)$, if σ is true in every epistemic model for Ψ.

4 $\mathcal{ALCK}$ as a Query Language

In this section, we use $\mathcal{ALCK}$ as a query language to $\mathcal{ALC}$-knowledge bases. First of all we introduce the notion of epistemic query.

Definition 2 *Given an $\mathcal{ALC}$-knowledge base Σ, an $\mathcal{ALCK}$-concept C, and an individual a, the answer to the query $C(a)$ posed to Σ is* `YES` *if $\Sigma \models C(a)$,* `NO` *if $\Sigma \models \neg C(a)$, and* `UNKNOWN` *otherwise. Moreover, the* answer set *of C w.r.t. Σ is the set $\{a \in \mathcal{O}_\Sigma \mid \Sigma \models C(a)\}$, where $\mathcal{O}_\Sigma$ is the set of individuals appearing in Σ.*

To answer epistemic queries posed to an $\mathcal{ALC}$-knowledge base Σ one can check whether Σ plus the negation of the query is unsatisfiable. In [7], we defined a sound and complete calculus to answer epistemic queries to an $\mathcal{ALC}$-knowledge base consisting of an ABox only. Although such a calculus does not consider the TBox, it can be suitably extended in order to treat inclusion statements in the spirit of [4, 9]. We do not present the extended calculus in this paper. It is reported in [8], where the decidability of the problem of answering epistemic queries to an $\mathcal{ALC}$-knowledge base is proved and its computational properties are discussed.

Our goal here is to show that the use of epistemic operators in queries allows for a more sophisticated interaction with the a knowledge representation system. For this purpose we provide an example of an $\mathcal{ALC}$-knowledge base and discuss various kinds of queries that can be posed to it using the language $\mathcal{ALCK}$.

In Figure 1 we show an $\mathcal{ALC}$-knowledge base $\Sigma_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ describing information about a university. The TBox $\mathcal{T}_1$ contains information about the various classes of persons working in the university and the courses supplied by the university. We use $D \doteq C$ as a shorthand for $C \sqsubseteq D$ and $D \sqsubseteq C$. The ABox $\mathcal{A}_1$ keeps track of the actual persons and courses involved in the university, together with the relations between them. The ABox $\mathcal{A}_1$ is also shown in graph form in Figure 2.

It can be easily shown that Σ_1 is satisfiable and that it has several different models. In fact, it does not have complete knowledge about the represented world. For example, since EE282 is an intermediary course, Σ_1 knows that at least one graduate student is enrolled in EE282, but it doesn't know who she/he actually is. Similarly, Σ_1 knows that Susan is either a graduate or an undergraduate, without knowing which one.

Notice that the information in $\mathcal{T}_1$ plays a role in the deduction of properties of individuals in $\mathcal{A}_1$. For example, Σ_1 knows that Mary is a graduate student, because she has a bachelor's degree and thus, according to $\mathcal{T}_1$, she falls under the description of graduate student.

We consider now various $\mathcal{ALCK}$ queries directed to Σ_1. In particular, in order to understand the role of the epistemic operator $\mathbf{K}$, we consider both $\mathcal{ALC}$ queries and modified versions of them including $\mathbf{K}$. The comparison between their respective semantics highlights the role of $\mathbf{K}$ in the query language.

We start with a pair of queries involving one single existential quantifier:

- Query 1a: $\Sigma_1 \models \exists ENROLLED.Grad(ee282)$? Answer: `YES`.
- Query 1b: $\Sigma_1 \models \exists \mathbf{K} ENROLLED.\mathbf{K} Grad(ee282)$? Answer: `NO`.

Query 1a asks whether there is a graduate student enrolled in EE282. The answer is `YES` because EE282 is an intermediary course and therefore, according to $\mathcal{T}_1$, there is at least one graduate student enrolled in it. However, as we already mentioned, the name of the enrolled student is unknown. It might either be one of the individuals named in Σ_1 or a different one about whom no information is given. Moreover, it is not even ensured that it is the same one in all models.

On the other hand, Query 1b asks whether there exists an individual who is known both to be enrolled in EE282 and to be a graduate student. In other words, it asks for an individual, say `fred`, such that both the assertions $ENROLLED(ee282, fred)$ and $Grad(fred)$ hold in every model for Σ_1. Such an individual does not exist, thus the answer to the query is `NO`.

$$
\begin{aligned}
&AdvCourse \doteq Course \sqcap \forall ENROLLED.Grad,\\
&BasCourse \doteq Course \sqcap \forall ENROLLED.Undergrad,\\
&IntCourse \doteq Course \sqcap \exists ENROLLED.Grad \sqcap \exists ENROLLED.Undergrad,\\
&\exists TEACHES.Course \sqsubseteq Grad \sqcup Professor,\\
&Grad \doteq Student \sqcap \exists DEGREE.Bachelor,\\
&Undergrad \doteq Student \sqcap \neg Grad
\end{aligned}
$$

The TBox $\mathcal{T}_1$

$$
\begin{aligned}
&Professor(bob), TEACHES(bob, ee282), TEACHES(john, cs324),\\
&TEACHES(john, cs221), Course(cs221), Course(cs324),\\
&IntCourse(ee282), ENROLLED(ee282, peter), ENROLLED(cs221, mary),\\
&ENROLLED(cs221, susan), ENROLLED(cs324, susan),\\
&ENROLLED(cs324, peter), Undergrad(peter), Student(susan),\\
&Student(mary), DEGREE(mary, bs), Bachelor(bs)
\end{aligned}
$$

The ABox $\mathcal{A}_1$

Figure 1: The $\mathcal{ALC}$-knowledge base Σ_1

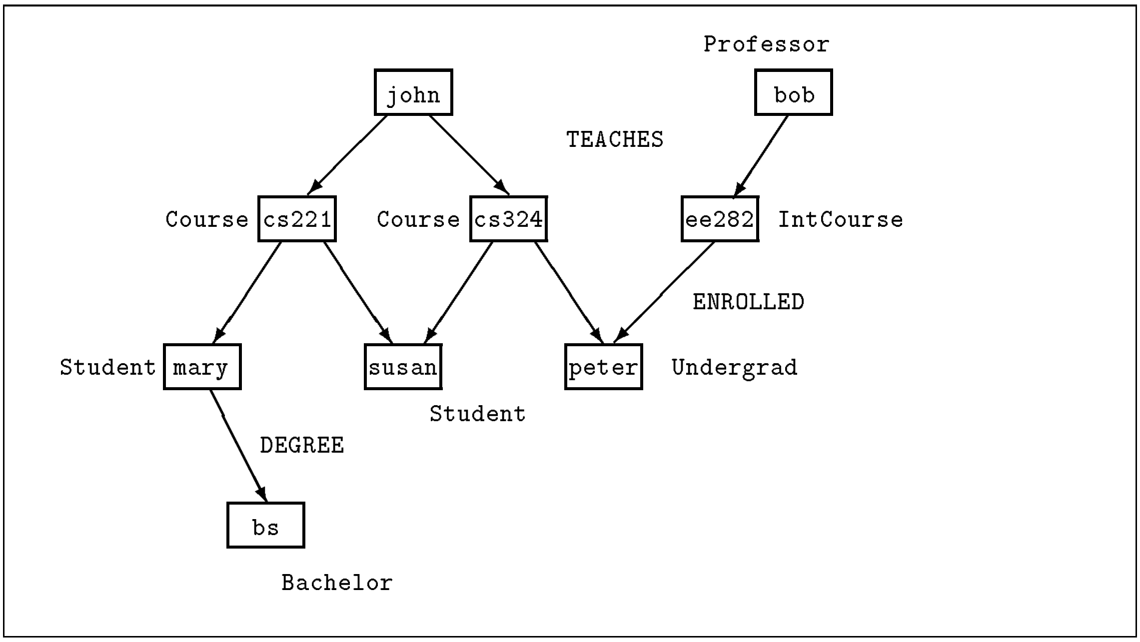

Figure 2: A pictorial representation of the ABox

The next pair of queries shows the interaction of the epistemic operator with the disjunction constructor:

- Query 2a: $\Sigma_1 \models Grad \sqcup Professor(john)$? Answer: `YES`.
- Query 2b: $\Sigma_1 \models \mathbf{K}Grad \sqcup \mathbf{K}Professor(john)$? Answer: `NO`.

Query 2a asks whether John is either a graduate student or a professor. The answer is `YES`, and it can be derived by the fact that it is stated in the ABox that he teaches two courses, and, according to the TBox, everybody who teaches at least one course is either a graduate student or a professor.

Query 2b, instead, asks whether he is either known to be a graduate student or known to be a professor. It is easy to verify that none of them is true and therefore the answer to this query is `NO`.

We consider now queries that involve also universal quantifiers:

- Query 3a: $\Sigma_1 \models \forall TEACHES.(IntCourse \sqcup \neg Course)?(bob)$ Answer: `UNKNOWN`.
- Query 3b $\Sigma_1 \models \forall \mathbf{K}TEACHES.?\mathbf{K}(IntCourse \sqcup \neg Course)(bob)$ Answer: `YES`.

Query 3a asks whether every course taught by Bob is an intermediary one. The answer is `UNKNOWN` because there are models for Σ_1 in which Bob teaches only intermediary courses as well as models in which he teaches also courses that are not intermediary.

Query 3b, instead, asks whether everything that is known to be taught by Bob is also known to be either an intermediary course or not to be a course. Since the only thing taught by Bob is EE282, and it is indeed an intermediary course, the answer to Query 3b is `YES`.

In the above example the addition of $\mathbf{K}$ has changed the answer from `UNKNOWN` to `YES`. Notice that it is also possible that Query 3a could be answered `NO` and Query 3b still be answered `YES`: Suppose that the assertion $\exists TEACHES.AdvCourse\ (bob)$ is added to Σ_1 and then the same queries are asked. Query 3a now gets the answer `NO`, because `AdvCourse` and `IntCourse` are disjoint concepts. However, the set of known courses taught by Bob is not changed, and therefore the answer to Query 3b is still `YES`.

We now consider some queries involving nested quantifiers: Queries 4a and 4b involve two levels of existential quantification. The innermost quantifier is carried by the concept $IntCourse$, which has existential quantifiers in its definition in $\mathcal{T}_1$.

- Query 4a: $\Sigma_1 \models \exists TEACHES.IntCourse(john)$? Answer: `YES`.
- Query 4b: $\Sigma_1 \models \exists \mathbf{K}TEACHES.\mathbf{K}IntCourse(john)$? Answer: `NO`.

Query 4a asks whether John teaches an intermediary course. At a superficial reading of the query, it might seem that the answer should be `NO`. The answer `NO` is supported by the fact that none of the courses taught by John is known to be an intermediary course, i.e. neither $IntCourse(cs221)$ nor $IntCourse(cs324)$ is a logical consequence of Σ_1. Nevertheless, the correct answer is `YES`, and in order to get it, one must reason by *case analysis*: As we have already remarked, the knowledge base does not provide the information as to whether Susan is a graduate or an undergraduate; however, since she is a student, according to $\mathcal{T}_1$, she must either be one or the other. This fact ensures that in every model for Σ_1 either $Grad(susan)$ or $Undergrad(susan)$ holds. Consider now the set of models for Σ_1 in which $Grad(susan)$ holds. In each of these models, the course CS324 is taken by both a graduate (Susan) and an undergraduate (Peter), thus it is an intermediary course. Similarly, consider the set of the remaining models for Σ_1, i.e. the ones in which $Undergrad(susan)$ holds. It is easy to see that in every model for this set the course CS221, this time, is taken by both a graduate (Mary) and an undergraduate (Susan), and therefore it is an intermediary course.

In conclusion, in every model for Σ_1 either CS324 or CS221 is an intermediary course. It follows that in every model for Σ_1 John teaches an intermediary course, proving that the correct answer to Query 4a is `YES`.

On the other hand, Query 4b asks whether John is known to teach a course that is known to be an intermediary course. The courses known to be taught by John are CS221 and CS324 and the only known intermediate course is EE282, therefore none of them is within the conditions required by the query.

Query 4a shows how, in some cases, the first order semantics might not agree with the intuitive reading of a query. In fact, most people tend to read Query 4a as requiring the reasoning pattern that is actually associated with the semantics of Query 4b. In other words, they tend to rule out the case analysis from the computation. One good reason to do so is that case analysis generally makes reasoning harder. In fact, as proved in [26], the problem of answering queries with existential quantification under the first order semantics, is in general coNP-hard. Whereas, as shown in [7], queries involving existential quantification only of the form $\exists \mathbf{K}P.\mathbf{K}C$ can be answered in polynomial time w.r.t. the size of the knowledge base. However, there are also cases in which the intuition agrees with the first order interpretation. For this reason, in our opinion, it is important to have the operator $\mathbf{K}$, which gives the possibility to choose between the two alternative readings of the query.

Regarding the interaction between the epistemic operator and the quantifiers, notice that we have always considered queries of the form $\exists \mathbf{K}P.\mathbf{K}C$ and $\forall \mathbf{K}P.\mathbf{K}C$, i.e. queries in which the $\mathbf{K}$ operator is placed in front of both the concept and the role. Such queries usually have an easy intuitive interpretation and therefore are the most interesting. Nevertheless, it might be worthwhile to consider even other possible variations of them, for example queries like $\exists \mathbf{K}P.C$ or $\forall P.\mathbf{K}C$. Such queries are perfectly legal in $\mathcal{ALC}K$, however, in some cases, they may lack an intuitive meaning. The reason is that they amalgamate $\mathcal{ALC}$-concepts with epistemic ones, resulting in something to which it is usually hard to give an intuitive meaning.

In other cases, though, they can play a useful role. As an example consider the following queries:

- Query 4c: $\Sigma_1 \models \exists \mathbf{K} TEACHES.IntCourse(john)$? Answer: YES.
- Query 4d: $\Sigma_1 \models \exists TEACHES.\mathbf{K} IntCourse(john)$? Answer: UNKNOWN.

Notice that Query 4c gets the same answer (YES) as Query 4a. In fact, since $TEACHES(john, cs221)$ and $TEACHES(john, cs324)$ are known, the addition of $\mathbf{K}$ in front of $TEACHES$ does not change the answer to the query. Query 4d, instead, is answered UNKNOWN because the only known intermediate course is EE282 and we can neither prove nor exclude that John teaches it.

The fact that Query 4c gets the answer YES and Query 4d the answer UNKNOWN may help us understand the answers to Query 4a and 4b. In particular, it clarifies which is the actual reason that makes Query 4a and 4b different: It tells us that the incompleteness of the knowledge base is related to the concept $IntCourse$ and not to the role $TEACHES$. In fact, $TEACHES(john, cs324)$ and $TEACHES(john, cs221)$ are both true in Σ_1, while $IntCourse(cs324)$ and $IntCourse(cs221)$ are not—only their disjunction is true.

5 Closed World Reasoning

The reason for the open world semantics of concept languages is that they are generally used in applications where one has to account for incomplete information. For example, even if all the known courses taught by Bob are intermediary, one does not want to conclude that all possible courses that Bob teaches are intermediary.

On the other hand, there are situations where it is natural to query a knowledge base under the Closed World Assumption. Referring to the knowledge base Σ_1 of Figure 1, consider the following examples:

- Query 5a: $\Sigma_1 \models \forall TEACHES.\exists ENROLLED.\top(john)$? Answer: UNKNOWN
- Query 5b: $\Sigma_1 \models \forall \mathbf{K} TEACHES.\exists ENROLLED.\top(john)$? Answer: YES

Query 5a gets the answer UNKNOWN because there is a model for Σ_1 where John teaches a course z, but there are no students enrolled in z, i.e. z is not an instance of the concept $\exists ENROLLED.\top$. On the other hand, the correct reading of Query 5b is as follows: Is it true that for every course z that John is known to teach, there is at least one student enrolled in z? It is easy to see the answer to the query is YES.

The above example shows that the use of $\mathbf{K}$ allows one to pose queries to a knowledge base Σ asking the system to assume complete knowledge on a certain individual a and a certain role P in Σ ($john$ and $TEACHES$ in the example). In particular, assuming complete knowledge on a and P here means assuming that for every pair (a,b) such that $\Sigma \not\models P(a,b)$, the assertion $P(a,b)$ is false in Σ. It is clear that this kind of reasoning is a form closed world reasoning.

We show here that under certain restrictions, our query language allows us to achieve at least the expressive power of the (naive) Closed World Assumption (CWA) (see [24]). The restrictions affect both the content and the language of the knowledge base. We say a knowledge base is *simple* if it does not contain inclusion statements. In the following we consider simple knowledge bases where the ABox is expressed in the language $\mathcal{AL}_0$, whose concepts are formed according to the rule:

$$C, D \quad \longrightarrow \quad A \mid \neg A \mid C \sqcap D \mid \forall R.C.$$

More complex languages and knowledge bases and more powerful forms of closed world reasoning (e.g. Generalized CWA [18]) require a more sophisticated treatment, which is outside the scope of this paper.

We briefly reformulate the CWA in the setting of a simple $\mathcal{AL}_0$ knowledge base Σ. Let Σ^{CWA} be the knowledge base obtained from Σ by adding $\neg A(a)$ or $\neg P(a,b)$, respectively, for every assertion $A(a)$ or $P(a,b)$ that is not entailed by Σ. Now, for any concept C the statement $C(a)$ follows from Σ under the CWA, written $\Sigma \models_{CWA} C(a)$, if $C(a)$ follows from Σ^{CWA}.

In the following, we assume that the $\mathcal{ALC}$-concepts used for querying a knowledge base are in *negation normal form*, i.e. negations signs are pushed down until they only occur in front of primitive concepts (see [28]). Given an $\mathcal{ALC}$-concept C in negation normal form, we define the $\mathcal{ALCK}$-concept $\overline{C}$ as follows:

$$\begin{aligned} \overline{A} &= \mathbf{K}A \\ \overline{\neg A} &= \neg\mathbf{K}A \\ \overline{C \sqcap D} &= \overline{C} \sqcap \overline{D} \\ \overline{C \sqcup D} &= \overline{C} \sqcup \overline{D} \\ \overline{\exists P.C} &= \exists \mathbf{K}P.\overline{C} \\ \overline{\forall P.C} &= \forall \mathbf{K}P.\overline{C}. \end{aligned}$$

The above transformation puts an epistemic operator in front of every primitive concept and primitive role. Now, it is possible to show that, if Σ is a simple $\mathcal{AL}_0$-knowledge base, C is an $\mathcal{ALC}$-concept, and a is an individual, then $\Sigma \models_{CWA} C(a)$ if and only if $\Sigma \models \overline{C}(a)$. Moreover, checking whether $\Sigma \models \overline{C}(a)$ can be done in time polynomial in the size of both the query and the knowledge base. This is in sharp contrast to answering queries that are formulated with arbitrary $\mathcal{ALC}$-concepts, which is a PSPACE-hard problem even for a fixed $\mathcal{AL}_0$-knowledge base.

Intuitively, the reason for the above result is that for an $\mathcal{AL}_0$-concept C the assertion $C(a)$ is logically equivalent to a finite set of Horn clauses and, therefore, simple $\mathcal{AL}_0$ knowledge bases are equivalent to sets of Horn clauses. As a consequence, if such a knowledge base is satisfiable, it always has one minimum model, say $\mathcal{I}_0$. Hence, evaluating a query under the CWA amounts to evaluating it in $\mathcal{I}_0$.

Now, putting a $\mathbf{K}$ in front of every primitive concept A and role P has the effect that A and P are taken as the intersection of their interpretations in all models of Σ, i.e., they are interpreted in $\mathcal{I}_0$. This explains why closed world reasoning can be enforced through the use of $\mathbf{K}$. That queries can be answered in polynomial time is due to the fact that on the one hand the Horn clauses corresponding to a simple $\mathcal{AL}_0$-knowledge base do not contain function symbols and on the other hand that concepts have a hierarchical structure that makes them suitable for efficient bottom up evaluation.

Notice that transforming a query C into $\overline{C}$ implies answering the query under the assumption that the knowledge about every role is complete, like for example in [19, p. 113]. On the other hand, as noted in [13], there are situations where we would like to apply the closed world assumption only to some of the concepts and the roles of the knowledge base.

We argue that the use of epistemic operators as described in the previous sections is a natural way to achieve such a flexible way of interacting with the knowledge base. Indeed, the careful introduction of the epistemic operator into the query induces the system to answer queries under the assumption that *part* of the knowledge base is complete, in contrast to assigning a closed world semantics to the knowledge base itself.

Consider the following query to the knowledge base Σ_1 given in Section 4:

- Query 4e: $\Sigma_1 \models \exists \mathbf{K} TEACHES.\mathbf{K}(Course \sqcap \exists ENROLLED.Grad \sqcap \exists ENROLLED.(Student \sqcap \neg \mathbf{K} Grad))(john)$? Answer: `YES`.

Notice that Query 4e is syntactically equal to Query 4b, except that the concept $IntCourse$ is replaced by the $\mathcal{ALCK}$-concept

$$Course \sqcap \exists ENROLLED.Grad \sqcap \exists ENROLLED.(Student \sqcap \neg \mathbf{K} Grad). \qquad (1)$$

Concept (1) differs from the definition of $IntCourse$ in the fact that $Undergrad$ is replaced by $(Student \sqcap \neg \mathbf{K} Grad)$. Concept (1) should be interpreted as the concept describing the courses that are intermediary under the assumption that every student is an undergraduate, unless the contrary is known. In fact, a course belongs to such a concept if both a graduate and a student not known to be a graduate are enrolled in it. It is easy to see that the course CS221 is an instance of Concept (1), and therefore the answer to Query 4e is `YES`.

Notice that asking queries like Query 4e is completely different from giving some kind of closed world semantics to the knowledge base. In fact, in our framework the knowledge base is perfectly monotonic, whereas using the epistemic operator the queries can be formulated in such a way that the reasoning which is required to compute the answers is nonmonotonic.

6 Rules as Epistemic Statements

In the previous sections we considered knowledge bases constituted by inclusions and membership assertions in $\mathcal{ALC}$. We now consider the case where epistemic sentences of a special kind are introduced into the knowledge base, and show that this extension formalizes the usage of procedural rules (or simply rules), as provided in many practical systems based on concept languages. In fact, systems such as CLASSIC [2] and LOOM [17], in addition to inclusions and membership assertions provide another mechanism for expressing knowledge, by means of so-called *rules*. Such rules are sentences the form

$$C \Rightarrow D$$

where C, D are concepts. The meaning of a rule is "if an individual is proved to be an instance of C, then derive that it is also an instance of D" (see [2]), and its behavior of rules is usually described in terms of a forward reasoning process that adds to the knowledge base the assertion $D(a)$ whenever $C(a)$ is proved to hold. We call *procedural extension* of a knowledge base Σ w.r.t. a set of rules the knowledge base resulting from such a forward reasoning process.

Rules in the context of frame-based systems are often defined informally. Attempts to precisely capture the meaning of such rules are based either on viewing them as knowledge base updates (see for example the *TELL* operation of [14]), or on ad hoc semantics (see [27]). Our aim in this section is to show that rules can be nicely formalized as particular epistemic sentences.

In the following we consider $\mathcal{ALCK}$-knowledge bases of the form $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} = \mathcal{T}' \cup \mathcal{R}$ with $\mathcal{T}'$ being a set of $\mathcal{ALC}$-inclusion statements, and $\mathcal{R}$ a set of epistemic sentences, each one of the form[2]

$$\mathbf{K}C \sqsubseteq D$$

where C and D are $\mathcal{ALC}$-concepts. We call these sentences *trigger rules*, since they are our formal counterpart of the rules $C \Rightarrow D$. We also call C the antecedent and D the consequent of the trigger rule. As a notational convenience we write the $\mathcal{ALCK}$-knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ as $\langle \Sigma, \mathcal{R} \rangle$, where $\Sigma = \langle \mathcal{T}', \mathcal{A} \rangle$.

From the definition of the semantics of $\mathcal{ALCK}$-knowledge bases it follows that an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ satisfies the trigger rule $\mathbf{K}C \sqsubseteq D$ if $(\mathbf{K}C)^{\mathcal{I},\mathcal{W}} \subseteq D^{\mathcal{I},\mathcal{W}}$. Intuitively, the set of epistemic sentences $\mathcal{R}$ restricts the set of models for Σ to the maximal subsets that satisfy every trigger rule in $\mathcal{R}$. More precisely, it can be shown that if $(\mathcal{I}, \mathcal{W})$ is an epistemic model for $\Phi = \langle \Sigma, \mathcal{R} \rangle$, then $\mathcal{W}$ is a maximal subset of $\mathcal{M}(\Sigma)$ such that for each $\mathcal{J} \in \mathcal{W}$, $(\mathcal{J}, \mathcal{W})$ satisfies every sentence in $\mathcal{R}$. Because of the form of such sentences, it can also be shown that there exists only

[2]In [7] we used the notation $\mathbf{K}C \Rightarrow \mathbf{K}D$. The two notations are equivalent in the semantics we give.

one maximal subset $\mathcal{W}$ of $\mathcal{M}(\Sigma)$ such that for all $\mathcal{J} \in \mathcal{W}$, $(\mathcal{J}, \mathcal{W})$ satisfies every sentence in Φ.

Observe that when a concept C is equivalent to $\top$, i.e. $C^{\mathcal{I}} = \Delta$ for every interpretation $\mathcal{I}$, a trigger rule $\mathbf{K}C \sqsubseteq D$ is equivalent to the inclusion $\top \sqsubseteq D$. Besides this case, however, trigger rules are not expressible by inclusions. Indeed, the main difference between rules and inclusions is that the formers are intended to provide a reasoning mechanism which applies them in one direction only, namely from the antecedent to the consequent. Our formalization of rules with the epistemic operator correctly captures this property, as shown in the following example.

Consider the knowledge base $\Phi = \langle\langle\emptyset, \{\neg B(a)\}\rangle, \{\mathbf{K}A \sqsubseteq B\}\rangle$, and observe that there exists an epistemic model $(\mathcal{I}, \mathcal{W})$ of Φ such that $\gamma(a) \notin \neg A^{\mathcal{I}}$. Therefore, $\neg A(a)$ is not a logical consequence of Φ.

In order to characterize the notion of procedural extension we now introduce the concept of *first-order extension* of an $\mathcal{ALCK}$-knowledge base $\langle\Sigma, \mathcal{R}\rangle$. The first-order extension of $\Phi = \langle\Sigma, \mathcal{R}\rangle$, where $\Sigma = \langle\mathcal{T}, \mathcal{A}\rangle$, is the $\mathcal{ALC}$-knowledge base $\Sigma_{\mathcal{R}}$, which is the least solution (w.r.t. to set inclusion) of the following equations:

$$X = \langle\mathcal{T}', \mathcal{A}'\rangle$$

where

$$\begin{aligned} \mathcal{T}' &= \mathcal{T} \cup \{\top \sqsubseteq D \mid \mathbf{K}C \sqsubseteq D \in \mathcal{R} \text{ and } X \models \top \sqsubseteq C\} \\ \mathcal{A}' &= \mathcal{A} \cup \{D(a) \mid \mathbf{K}C \sqsubseteq D \in \mathcal{R} \text{ and } X \models C(a)\}. \end{aligned}$$

We do not delve into the details of the computation of the first-order extension. We simply remark that the solution of the above equations is unique and can be incrementally constructed starting from Σ in a number of steps which is polynomial w.r.t. the size of Φ.

First-order extensions are linked to the semantics by the following property. Let $\Phi = \langle\Sigma, \mathcal{R}\rangle$ be an $\mathcal{ALCK}$-knowledge base, let $(\mathcal{I}, \mathcal{W})$ be an epistemic model for Φ, and let $\Sigma_{\mathcal{R}}$ be the first-order extension of Φ. Then $\mathcal{W}$ coincides with the set of models for the $\mathcal{ALC}$-knowledge base $\Sigma_{\mathcal{R}}$. In other words, the result of the forward reasoning process on a knowledge base and set of trigger rules, which is represented by the least solution of the above equations, is correctly captured by the semantics of the $\mathcal{ALCK}$-knowledge base Φ, where the trigger rules are expressed as epistemic sentences.

We now show an example of the usage of rules in our framework. Consider the $\mathcal{ALCK}$-knowledge base $\Phi = \langle\Sigma, \mathcal{R}\rangle$:

$$\begin{aligned} \Sigma &= \langle\emptyset, \{TEACHES(bill, cs248),\ Grad(bill)\}\rangle \\ \mathcal{R} &= \{\mathbf{K}Grad \sqsubseteq \forall TEACHES.BasCourse\}. \end{aligned}$$

The first-order extension of Φ is

$$\Sigma_{\mathcal{R}} = \langle\emptyset, \{TEACHES(bill, cs248),\ Grad(bill),\ \forall TEACHES.BasCourse(bill)\}\rangle.$$

Obviously, $\Sigma_{\mathcal{R}} \models BasCourse(cs248)$. From the semantics, one can verify that for every epistemic model $(\mathcal{I}, \mathcal{W})$ for Φ, we have $\gamma(bill) \in (\forall TEACHES.BasCourse)^{\mathcal{I},\mathcal{W}}$ and $\gamma(cs248) \in BasCourse^{\mathcal{I},\mathcal{W}}$, i.e., both the assertion $\forall TEACHES.BasCourse$ $(bill)$ and $BasCourse(cs248)$ are logical consequences of Φ, as one would expect.

It is worth noting that the calculus for answering epistemic queries, mentioned in Section 4, can be effectively used in the computation of the first-order extension of an $\mathcal{ALCK}$-knowledge base. In fact, the application of a trigger rule $\mathbf{K}C \sqsubseteq D$ requires to compute the answer set of the query $\mathbf{K}C$, which can be done by means of that calculus.

7 Weak Inclusions as Epistemic Statements

Recent studies on the formal properties of concept languages [4, 20, 21] show that one of the critical aspects of the implementation of knowledge representation systems based on concept languages is the treatment of inclusions. This problem is addressed for example in LOOM [16] by adopting a weak form of inclusion, which applies only to known individuals and disregards many inferences based on the use of contrapositives.

In this section we argue that the class of epistemic sentences used in the formalization of trigger rules can be regarded as a form of weak inclusion which may lead to significant computational advantages in comparison to inclusion statements as defined in Section 2.

To this purpose we introduce the notion of *weakening* of an $\mathcal{ALCK}$-knowledge base, which is the $\mathcal{ALCK}$-knowledge base obtained by replacing every inclusion statement $C \sqsubseteq D$ by the epistemic statement $\mathbf{K}C \sqsubseteq D$. More formally, let $\Phi = \langle\langle\mathcal{T}, \mathcal{A}\rangle, \mathcal{R}\rangle$ be an $\mathcal{ALCK}$-knowledge base as defined in the previous section. The weakening of Φ is the $\mathcal{ALCK}$-knowledge base

$$\Phi^- = \langle\Sigma', \mathcal{R}'\rangle$$

where

$$\Sigma' = \langle\emptyset, \mathcal{A}\rangle$$

and

$$\mathcal{R}' = \mathcal{R} \cup \{\mathbf{K}C \sqsubseteq D \mid (C \sqsubseteq D) \in \mathcal{T}\}.$$

Intuitively, every inference we can make in Φ^- can be done in Φ as well, while the converse of course is not true. Hence, Φ^- can be regarded as a sound approximation of Φ, where the lost inferences are traded with a gain in the efficiency of reasoning. Before addressing in more detail this computational aspect, we present an example of the weakening transformation.

Consider the knowledge base $\Phi_1 = \langle\Sigma_1, \emptyset\rangle$, where $\Sigma_1 = \langle\mathcal{T}_1, \mathcal{A}_1\rangle$ is the knowledge base used in Section 4. The weakening Φ_1^- will be $\langle\langle\emptyset, \mathcal{A}_1\rangle, \mathcal{R}_1\rangle$, where $\mathcal{R}_1$ is shown in Figure 3. Recall that all definitions of the form $C \doteq D$ are a shorthand for $C \sqsubseteq D$ and $D \sqsubseteq C$.

$\mathbf{K}AdvCourse \sqsubseteq (Course \sqcap \forall ENROLLED.Grad)$,
$\mathbf{K}(Course \sqcap \forall ENROLLED.Grad) \sqsubseteq AdvCourse$,
$\mathbf{K}BasCourse \sqsubseteq (Course \sqcap \forall ENROLLED.Undergrad)$,
$\mathbf{K}(Course \sqcap \forall ENROLLED.Undergrad) \sqsubseteq BasCourse$,
$\mathbf{K}IntCourse \sqsubseteq (Course \sqcap \exists ENROLLED.Grad \sqcap \exists ENROLLED.Undergrad)$,
$\mathbf{K}(Course \sqcap \exists ENROLLED.Grad \sqcap \exists ENROLLED.Undergrad) \sqsubseteq IntCourse$,
$\mathbf{K}\exists TEACHES.Course \sqsubseteq Grad \sqcup Professor$,
$\mathbf{K}Grad \sqsubseteq (Student \sqcap \exists DEGREE.Bachelor)$,
$\mathbf{K}(Student \sqcap \exists DEGREE.Bachelor) \sqsubseteq Grad$,
$\mathbf{K}Undergrad \sqsubseteq (Student \sqcap \neg Grad)$,
$\mathbf{K}(Student \sqcap \neg Grad) \sqsubseteq Undergrad$

Figure 3: The trigger rules of Φ_1^-, obtained by weakening the inclusions of $\mathcal{T}_1$

It can be verified that all queries asked to Σ_1 in Section 4 have the same answer in Φ_1^-, except for queries 4a and 4d, reported here for the sake of clarity.

- Query 4a: $\Sigma_1 \models \exists TEACHES.IntCourse(john)$? Answer: `YES`.
- Query 4d: $\Sigma_1 \models \exists \mathbf{K}TEACHES.IntCourse(john)$? Answer: `YES`.

These queries receive the answer `YES` in Σ_1 because of a case analysis on Susan. Recall that, according to $\mathcal{T}_1$, the TBox of Σ_1, the two concepts `Grad` and `Undergrad` partition the concept `Student`. Being a student, Susan can be either a graduate or an undergraduate. In the first case, the course CS221 is an inetrmediary course, while in the second case CS324 is an inetrmediary course. Hence, in both cases John teaches an inetrmediary course.

On the contrary, it is easy to see that this does not happen in Φ_1^-, as shown by the following queries.

- Query 4f: $\Phi_1^- \models \exists TEACHES.IntCourse(john)$? Answer: `UNKNOWN`.
- Query 4g: $\Phi_1^- \models \exists \mathbf{K}TEACHES.IntCourse(john)$? Answer: `UNKNOWN`.

This is because in Φ_1^- the two concepts `Grad` and `Undergrad` do not partition the concept `Student`. What we just know is that individuals known to be undergraduates are inferred to be students and nongraduates, and vice versa, that individuals known to be students and nongraduates are inferred to be undergraduates. Since Susan is in neither of the two conditions, we cannot infer anything about her. In fact, there are now epistemic models for Φ_1^- where Susan is neither a graduate nor an undergraduate. Therefore, the two queries 4f and 4g receive the answer UNKNOWN.

One can also verify that contrapositives are not applicable in Φ_1^-. Compare the answer to $\neg\exists DEGREE.Bachelor(peter)$ in the two knowledge bases:

- Query 5a: $\Sigma_1 \models \neg\exists DEGREE.Bachelor(peter)$? Answer: YES.
- Query 5b: $\Phi_1^- \models \neg\exists DEGREE.Bachelor(peter)$? Answer: UNKNOWN.

In fact, in Σ_1 Peter is known to be an undergraduate, hence a student who is a nongraduate. Since graduates are defined as students with a bachelor's degree, we can infer that Peter has none by using the contrapositive of the inclusion $(Student \sqcap \exists DEGREE.Bachelor) \sqsubseteq Grad$. Instead, in Φ_1^- we only can infer that Peter is a student and a nongraduate. This does not activate the contrapositive of the trigger rule $\mathbf{K}(Student \sqcap \exists DEGREE.Bachelor) \sqsubseteq Grad$.

Let us now go back to the computational advantages of weakening an $\mathcal{ALCK}$-knowledge base. In order to show such advantages, consider an $\mathcal{ALCK}$-knowledge base $\Phi = \langle \Sigma, \mathcal{R} \rangle$, where $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, and let $\Phi^- = \langle \Sigma', \mathcal{R}' \rangle$, where $\Sigma' = \langle \emptyset, \mathcal{A} \rangle$, be its weakening. Furthermore, assume that no rule in $\mathcal{R}'$ has an antecedent which is equivalent to $\top$.

Extending the results of the complexity analysis carried out in [4, 9], one can show that query answering in Φ can be solved in exponential space and double exponential time [8]. Since query answering in $\mathcal{ALC}$-knowledge bases with inclusions is known to be EXPTIME-hard [4], we do not expect to find any algorithm working in polynomial space, unless $EXPTIME = PSPACE$. On the other hand query answering in Φ^- amounts to solving the same problems in $\Sigma'_{\mathcal{R}'}$, which is the first-order extension of $\Sigma' = \langle \emptyset, \mathcal{A} \rangle$ w.r.t. $\mathcal{R}'$. Observing that $\Sigma'_{\mathcal{R}'}$ is a knowledge base constituted by an ABox only, we know from [7] that this problem can be solved in polynomial space. Since the size of $\Sigma'_{\mathcal{R}'}$ is polynomially related to the size of Φ^-, and therefore of Φ too, the above observation shows that weakening the inclusions of an $\mathcal{ALCK}$-knowledge base leads to an exponential decrease of the space required for query answering.

We can conclude that the notion of weakening proposed here provides a form of incomplete reasoning that is both computationally advantageous and semantically well-founded.

8 Conclusion

In this paper, we discussed the advantages of using an epistemic operator both for enhancing the capabilities of concept languages, and for formalizing non-standard features of existing knowledge representation systems based on concept languages. We have shown that the epistemic operator is flexible enough to account for several different notions in an elegant and uniform way, namely epistemic queries, closed world reasoning, procedural rules and weak forms of concept definition.

At the same time, we believe that our investigation on the epistemic operator raises a number of interesting issues related to the use of concept languages in practical systems, which we intend to address in future work. First of all, it is worth analyzing whether the class of epistemic sentences proposed for formalizing rules and definitions can be extended so as to capture more aspects, while retaining the nice computational properties. Moreover, it would be interesting to analyze whether epistemic sentences are powerful enough to express some form of default reasoning.

Acknowledgements

This work has been supported by the Esprit Basic Research Action 6810 (Compulog 2), by the German Ministry for Research and Technology (BMFT) under grant ITW 9201 as part of the TACOS project, and by the Italian National Research Council as part of the Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo, Sottoprogetto 7, LDR Ibridi.

References

[1] Baader, F. and Hollunder, B. Embedding defaults into terminological knowledge representation formalisms. In *Proc. of the 3nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-92*, pages 306–317. Morgan Kaufmann, 1992.

[2] Brachman, R. J., Borgida, A., McGuinness, D. L., and Alperin Resnick, L. The CLASSIC knowledge representation system, or, KL-ONE: the next generation. Preprints of the Workshop on Formal Aspects of Semantic Networks, Two Harbors, Cal., 1989.

[3] Brachman, R. J. and Levesque, H. J. The tractability of subsumption in frame-based description languages. In *Proc. of the 4th Nat. Conf. on Artificial Intelligence AAAI-84*, pages 34–37, 1984.

[4] Buchheit, M., Donini, F. M., and Schaerf, A. Decidable reasoning in terminological knowledge representation systems. In *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence IJCAI-93*, 1993. In press.

[5] Donini, F. M., Lenzerini, M., Nardi, D., and Nutt, W. The complexity of concept languages. In Allen, J., Fikes, R., and Sandewall, E., editors, *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-91*, pages 151–162. Morgan Kaufmann, 1991.

[6] Donini, F. M., Lenzerini, M., Nardi, D., and Nutt, W. Tractable concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*, pages 458–463, Sidney, 1991.

[7] Donini, F. M., Lenzerini, M., Nardi, D., Nutt, W., and Schaerf, A. Adding epistemic operators to concept languages. In *Proc. of the 3nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-92*, pages 342–353, 1992.

[8] Donini, F. M., Lenzerini, M., Nardi, D., Nutt, W., and Schaerf, A. Adding epistemic operators to concept languages. Technical report, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1993. Forthcoming.

[9] Donini, F. M., Lenzerini, M., Nardi, D., and Schaerf, A. A hybrid system integrating datalog and concept languages. In *Proc. of the 2nd Italian Conf. on Artificial Intelligence*, volume 549 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 1991. An extended version appeared also in the Working Notes of the AAAI Fall Symposium "Principles of Hybrid Reasoning", 1991.

[10] Donini, F. M., Lenzerini, M., Nardi, D., and Schaerf, A. From subsumption to instance checking. Technical Report 15.92, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1992.

[11] Doyle, J. and Patil, R. S. Two thesis of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, 48:261–297, 1991.

[12] Fikes, R. and Kehler, T. The role of frame-based representation in reasoning. *Communications of the ACM*, 28(9):904–920, 1985.

[13] Gelfond, M. and Przymusinska, H. Negation as failure: Careful closure procedure. *Artificial Intelligence*, 30:273–287, 1986.

[14] Levesque, H. J. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23:155–212, 1984.

[15] Lifschitz, V. Nonmonotonic databases and epistemic queries. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence IJCAI-91*, Sidney, 1991.

[16] MacGregor, R. A deductive pattern matcher. In *Proc. of the 6th Nat. Conf. on Artificial Intelligence AAAI-88*, pages 403–408, 1988.

[17] MacGregor, R. and Bates, R. The Loom knowledge representation language. Technical Report ISI/RS-87-188, University of Southern California, Information Science Institute, Marina del Rey, Cal., 1987.

[18] Minker, J. On indefinite data bases and the closed world assumption. In *Conf. on Automated Deduction, LNCS 138*, 1982.

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

[20] Nebel, B. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990.

[21] Nebel, B. Terminological cycles: Semantics and computational properties. In Sowa, J. F., editor, *Principles of Semantic Networks*, pages 331–361. Morgan Kaufmann, 1991.

[22] Quantz, J. and Kindermann, C. Implementation of the BACK system version 4. Technical Report KIT-Report 78, FB Informatik, Technische Universität Berlin, Berlin, Germany, 1990.

[23] Quantz, J. and Royer, V. A preference semantics for defaults in terminological logics. In *Proc. of the 3nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-92*, pages 294–305, 1992.

[24] Reiter, R. On closed world data bases. In Gallaire, H. and Minker, J., editors, *Logic and Databases*, pages 119–140. Plenum, 1978.

[25] Reiter, R. On asking what a database knows. In Lloyd, J. W., editor, *Symposium on computational logics*, pages 96–113. Springer-Verlag, ESPRIT Basic Research Action Series, 1990.

[26] Schaerf, A. On the complexity of the instance checking problem in concept languages with existential quantification. In *Proc. of the 8th Int. Symp. on Methodologies for Intelligent Systems ISMIS-93*, 1993. In press. Extended version to appear in *Journal of Intelligent Information Systems*.

[27] Schild, K. Towards a theory of frames and rules. Technical report, FB Informatik, Technische Universität Berlin, Berlin, Germany, 1989.

[28] Schmidt-Schauß, M. and Smolka, G. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[29] Woods, W. A. Understanding subsumption and taxomony: A framework for progress. In Sowa, J., editor, *Principles of Semantic Networks*, pages 45–94. Morgan Kaufmann, 1991.

[30] Yen, J., Neches, R., and MacGregor, R. CLASP: Integrating term subsumption sstems and production systems. *IEEE trans. on Knowledge and Data Engineering*, 3(1):25–31, 1991.