On Higher-Order Description Logics

Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati

Dipartimento di Informatica e Sistemistica
SAPIENZA Università di Roma
`www.dis.uniroma1.it/∼lastname`

Abstract. We investigate an extension of Description Logics with higher-order capabilities, based on Henkin-style semantics. Our study starts from the observation that the various possibilities of adding higher-order constructs to a DL form a spectrum of increasing expressive power, including domain metamodeling, i.e., using concepts and roles as predicate arguments, and full metamodeling, providing the ability of using the language constructors and operators as predicate arguments, in the style of RDF. We argue that higher-order features of the former type are sufficiently rich and powerful for the modeling requirements arising in many relevant situations, and therefore we carry out an investigation of the computational complexity of reasoning in DLs extended with such features. In particular, we show that adding domain metamodeling capabilities to expressive DLs has no impact on the complexity of the various reasoning tasks.

1 Introduction

Metamodeling allows one to treat concepts and properties as first-order citizens, and to see them as individuals whose properties can be asserted and reasoned upon. This feature is important in all applications where the need arises of modeling and reasoning about meta-concepts, i.e., concepts whose instances are themselves concepts, and meta-properties, i.e., relationships between meta-concepts.

It is well-known that in logic, and in Description Logics (DLs) in particular, higher-order constructs are needed for a correct representation of concepts and properties at the meta-level. However, the issue of devising suitable extensions to DLs for representing and reasoning about meta-level elements is largely unexplored. Recent research on this subject shows that there is a spectrum in the modeling capabilities of DLs. Four points in this spectrum represent specific notable cases, which we call domain modeling, metaquerying, domain metamodeling, and full metamodeling, respectively.

Domain modeling. In domain modeling, the language only focuses on the ability of specifying the domain of interest in terms of individuals, concepts and roles, and is therefore “first-order”. This is the simplest case of the spectrum, with no higher-order feature, and is actually the one addressed in most of the research on DLs.

Metaquerying. This is the case where the knowledge base does not contain any axiom regarding meta-concepts or meta-roles, but the query language allows for using meta-concepts, so that concepts and roles in the knowledge base can match the variables in the query, and may thus be returned as answers to the query [1]. Note that this mechanism allows to express queries that are beyond first-order logic. For instance, asking for the least common subsumers of two concepts, or for the most specific concept for an individual, can be done by means of suitable meta queries.

Domain metamodeling. This is the case where the language allows for using concepts and roles as predicate arguments, so that one can assert properties of concepts and roles, as if they were individuals. Note that domain metamodeling includes metaquerying as a special case. It is our opinion that higher-order features of this kind are sufficiently rich and powerful for the modeling requirements arising in many relevant situations. One of the most popular approaches to domain metamodeling, and one that is closely related to DLs, is HiLog [7]. HiLog is a logic with a higher-order syntax, thus allowing predicates to appear as arguments in atomic formulae, but with a Henkin-style semantics, which implies that the expressive power of the language actually remains within the first-order realm.

Full metamodeling. This is the most general case, where the modeling language allows not only for using concepts and roles as predicate arguments, but also for refining and extending the properties of the language operators, and to reason upon such properties. RDF and RDFS, which are again based on a Henkin semantics, are popular languages enabling this kind of metamodeling. Both languages allow for stating axioms not only on domain (meta-)elements, but also on the so-called built-in vocabulary (i.e., operators such as `rdf:type`) of the language.

In this paper[1], we investigate an extension of DLs with higher-order capabilities. We are especially interested in those features that allow us both to model and to query individuals, concepts, roles, meta-concepts and meta-roles with no limitations. Therefore, the extension that we study is geared towards domain metamodeling (thus including metaquerying), for which we provide the following specific contributions.

First, we present syntax and semantics of an extension of DLs with domain metamodeling features (see Section 2). In particular, we show how, starting from any DL $\mathcal{L}$, one can define its higher-order version, called $Hi(\mathcal{L})$. From the syntax point of view, our approach stems from two ideas. On one hand, every modeling element can be seen simultaneously as an individual, as a concept, and as a role. On the other hand, since concepts in DLs are denoted not only by names, but also by complex expressions, every complex expression is a modeling element in our language. From the semantic point of view, we adopt a Henkin semantics, as in HiLog and RDF(S).

Second, we carry out an investigation of the computational complexity of reasoning in DLs extended with higher-order features. By reasoning we mean not only logical implication, but also answering unions of conjunctive queries with metaquerying abilities. We show that adding domain metamodeling capabilities to expressive DLs, in particular to $\mathcal{SHIQ}$ [3], has no impact on the complexity of the various reasoning tasks, including conjunctive query answering (see Section 4).

The idea of representing concepts and properties at the meta-level is an old one in Knowledge Representation and Computer Science. Semantic networks and early Frame-based systems incorporated specific mechanisms for representing concepts whose instances are themselves concepts [10, 2]. Conceptual modeling languages proposed in the 70's, such as TAXIS [12], provided both the notion of meta-class, and suitable facilities for describing properties of operators on meta-classes. The notion of

[1] For the sake of brevity, proofs are omitted in this version.

meta-class is also present in virtually all object-oriented languages, including modern programming languages (see, e.g., the Java class `Class`).

As we said before, the issue of extending DLs with higher-order constructs has been addressed only by few research papers. In [4], probably the first paper on this subject, the notion of "reification of concepts" is proposed as a means to express meta-level classes, but the paper does not address neither the issue of meta-roles, nor the issue of query answering. A more recent paper is [8], where metamodeling capabilities are added to a DL of the *DL-Lite* family.

Our work has connections with recent investigations on full metamodeling, in particular on RDF, RDFS, and OWL Full. In [11], the author addresses the issue of decidability of reasoning on meta-properties in different fragments of OWL Full. It is shown that, although going from domain to full metamodeling easily leads to undecidability, reasoning in some fragments of OWL Full is decidable. Differently from the present paper, the focus of [11] is neither on the tractability frontier, nor on conjunctive query answering. Finally, reasoning (but not query answering) with metamodeling is also studied in [14], where the language OWL FA is proposed, which introduces a stratum number in class constructors and axioms to indicate the strata they belong to, and suitable contraints impose that TBox axioms are stated on classes of the same stratum, while ABox axioms can only involve elements of two consecutive strata.

The rest of the paper is organized as follows. In Section 2, we describe syntax and semantics of $Hi(\mathcal{L})$, by referring, in particular, to the higher-order DL $Hi(\mathcal{SHIQ})$. As we said above, $Hi(\mathcal{L})$ denotes the higher-order version of the Description Logic $\mathcal{L}$. In Section 3 we present our technique for satisfiability in $Hi(\mathcal{SHIQ})$, and in Section 4 we address query answering in the same DL. Both in Section 3 and in Section 4 we also characterize the computational complexity of the presented algorithms. We end the paper in Section 5, by pointing out future directions of our reseearch.

2 Higher-order Description Logics

In this section, we present our approach to higher-order DLs, by showing how, starting from a DL $\mathcal{L}$, one can define its higher-order version, called $Hi(\mathcal{L})$. During the presentation, we will also refer to a specific DL, namely $\mathcal{SHIQ}$. Therefore, we will describe in detail the higher-order DL $Hi(\mathcal{SHIQ})$.

Before delving into $Hi(\mathcal{L})$, we present some preliminary definitions. Every traditional DL $\mathcal{L}$ is characterized by a set $OP(\mathcal{L})$ of *operators*, used to form concept and role expressions, and a set of $MP(\mathcal{L})$ of *meta-predicates*, used to form assertions. Each operator and each meta-predicate have an associated arity. If symbol S has arity n, then we often write S/n to denote such symbol and its arity. For $\mathcal{SHIQ}$, we have

$$OP(\mathcal{SHIQ}) = \{Inv/1,\ And/2,\ Not/1\} \cup \{AtLeastQ_n/2 \mid n \in \mathbb{N}\}$$
$$MP(\mathcal{SHIQ}) = \{Inst_C/2, Inst_R/3, Isa_C/2, Isa_R/2, Tran/1\}.$$

We assume that the reader is familiar with $\mathcal{SHIQ}$. Therefore, the intuitive meaning of all the above symbols should be clear. The formal specification of their semantics will be given shortly.

Syntax. We assume the existence of two disjoint, countably infinite alphabets: $\mathcal{S}$, the set of *names*, and $\mathcal{V}$, the set of *variables*. The buidling blocks of a $Hi(\mathcal{L})$ knowledge base are assertions, which in turn are based on expressions. We define the set of *expressions*, denoted by $\mathcal{E}_{\mathcal{L}}(\mathcal{S})$, over the alphabet $\mathcal{S}$ for $Hi(\mathcal{L})$ inductively as follows:

- if $E \in \mathcal{S}$ then $E \in \mathcal{E}_{\mathcal{L}}(\mathcal{S})$;
- if $C/n \in OP(\mathcal{L})$ and $E_1, \ldots, E_n \in \mathcal{E}_{\mathcal{L}}(\mathcal{S})$ then $C(E_1, \ldots, E_n) \in \mathcal{E}_{\mathcal{L}}(\mathcal{S})$.

Example 1. If the names $\mathit{Course}, \mathit{Teaches}, \mathit{Full}$ belong to the alphabet $\mathcal{S}$, then the following is a $Hi(\mathcal{SHIQ})$ expression:

$$\mathit{And}(\mathit{Course}, \mathit{Not}(\mathit{AtLeastQ}_2(\mathit{Inv}(\mathit{Teaches})), \mathit{Full})))$$

which intuitively denotes the concept representing the set of courses that are taught by at most one full professor. □

A $Hi(\mathcal{L})$ *assertion* over $\mathcal{E}_{\mathcal{L}}(\mathcal{S})$ is a statement of the form $M(E_1, \ldots, E_n)$ where $M \in MP(\mathcal{L})$, $n \geq 0$ is the arity of M, and for every $1 \leq i \leq n$, $E_i \in \mathcal{E}_{\mathcal{L}}(\mathcal{S})$. A $Hi(\mathcal{L})$ *knowledge base (KB)* is a set of assertions over $\mathcal{E}_{\mathcal{L}}(\mathcal{S})$.

Thus, an assertion is simply an application of a meta-predicate to a set of expressions. Intuitively, an assertion is an axiom that predicate over a set of individuals, concepts or roles.

Example 2. Suppose that the alphabet $\mathcal{S}$ contains all names mentioned in Example 1, plus $\mathit{GradCourse}, \mathit{UniversityConcept}, \mathit{ObsoleteConcept}, \mathit{John}$, and $\mathit{DefinedBy}$. Then the following are $Hi(\mathcal{SHIQ})$ assertions:

$$\mathit{Isa}_C(\mathit{GradCourse}, \mathit{And}(\mathit{Course}, \mathit{Not}(\mathit{AtLeastQ}_2(\mathit{Inv}(\mathit{Teaches})), \mathit{Full}))))$$
$$\mathit{Inst}_C(\mathit{And}(\mathit{Course}, \mathit{Not}(\mathit{AtLeastQ}_2(\mathit{Inv}(\mathit{Teaches})), \mathit{Full}))), \mathit{UniversityConcept})$$
$$\mathit{Inst}_R(\mathit{UniversityConcept}, \mathit{John}, \mathit{DefinedBy})$$
$$\mathit{Inst}_R(\mathit{Not}(\mathit{ObsoleteConcept}), \mathit{John}, \mathit{DefinedBy})$$

The first assertion states that every graduate course is taught by at most one full professor. The intended meaning of the second assertion is that the concept $\mathit{And}(\mathit{Course}, \mathit{Not}(\mathit{AtLeastQ}_2(\mathit{Inv}(\mathit{Teaches})), \mathit{Full})))$ is an instance of the concept $\mathit{UniversityConcept}$ (which is therefore a meta-concept). Finally, the intended meaning of the third and the fourth assertions is that the concepts $\mathit{UniversityConcept}$ and $\mathit{Not}(\mathit{ObsoleteConcept})$ have been introduced in the knowledge base by John. □

Next, we introduce the notion of query, which in turn relies on the notion of "atom". Intuitively, an atom is constituted by a meta-predicate applied to a set of arguments, where each argument is either an expression or a variable. More formally, we define the set $\tau(\mathcal{S}, \mathcal{V})$ of *terms* over $\mathcal{S}$ and $\mathcal{V}$ to be $\mathcal{E}_{\mathcal{L}}(\mathcal{S}) \cup \mathcal{V}$. Terms of the form $\mathcal{E}_{\mathcal{L}}(\mathcal{S})$ are called ground. We define an *atom* to be constituted by the application of a meta-predicate in $MP(\mathcal{L})$ to a set of terms, and we call an atom *ground* if no variable occurs in it. Note that a ground atom has the same form of an assertion. An atom whose meta-predicate

1. for each $d_1 \in \Delta$, if $d = Inv^{\mathcal{I}_o}(d_1)$ then $d^{\mathcal{I}_r} = (d_1^{\mathcal{I}_r})^{-1}$;
2. for each $d_1, d_2 \in \Delta$, if $d = And^{\mathcal{I}_o}(d_1, d_2)$ then $d^{\mathcal{I}_c} = d_1^{\mathcal{I}_c} \cap d_2^{\mathcal{I}_c}$;
3. for each $d_1 \in \Delta$, if $d = Not^{\mathcal{I}_o}(d_1)$ then $d^{\mathcal{I}_c} = \Delta - d_1^{\mathcal{I}_c}$;
4. for each $d_1, d_2 \in \Delta$ and for each $n \in \mathbb{N}$, if $d = AtLeastQ_n(d_1, d_2)$ then $d^{\mathcal{I}_c} = \{e \mid \exists e_1, \ldots, e_n$ s.t. $e_i \neq e_j$ for $i \neq j$, and ($\forall i$ s.t. $1 \leq i \leq n, \langle e, e_i \rangle \in d_1^{\mathcal{I}_r}$ and $e_i \in d_2^{\mathcal{I}_c}$).

Fig. 1. Semantic conditions on interpretations for $\mathcal{SHIQ}$ predicate expressions.

is Isa_C or Isa_R is called an *ISA-atom*, while we call *instance-atom* an atom whose meta-predicate is $Inst_C$ or $Inst_R$.

A *higher-order conjunctive query (HCQ)* of arity n is an expression of the form

$$q(x_1, \ldots, x_n) \leftarrow a_1, \ldots, a_m$$

where q, called the query predicate, is a symbol that does not belong to $\mathcal{S} \cup \mathcal{V}$, every x_i belongs to $\mathcal{V}$, every a_i is a (possibly non-ground) atom, and all variables $x_1, \ldots, x_n$ occur in some a_j. The variables $x_1, \ldots, x_n$ are called the *free variables* (or distinguished variables) of the query, while the other variables occurring in $a_1, \ldots, a_m$ are called *existential variables*. A *higher-order union of conjunctive queries (HUCQ)* of arity n is a set of HCQs of arity n with the same query predicate. A HCQ/HUCQ is called *Boolean* if it has no free variable.

Example 3. Referring to the alphabet mentioned in Example 2, the following is a HCQ:

$$q(x) \leftarrow Inst_C(x, y), Inst_R(y, John, DefinedBy)$$

Intuitively, the query asks for the instances of all the concepts in the knowledge base defined by John. In our case, the answer will be simply $\{GradCourse, Not(AtLeastQ_2(Inv(Teaches)), Full)))\}$.

Example 4. Consider now the case where we want to ask for the instances of all the concepts y such that the expression $Not(y)$ is a concept in the knowledge base defined by John. The natural formulation of this query would be:

$$q(x) \leftarrow Inst_C(x, y), Inst_R(Not(y), John, DefinedBy)$$

However, according to our syntax for queries, variables cannot appear as arguments within terms, and therefore the above is *not* a query in $Hi(\mathcal{SHIQ})$. We discuss this kind of queries further in the conclusions. □

Semantics. The semantics of $Hi(\mathcal{L})$ is based on the notion of interpretation structure. An *interpretation structure* is a triple $\Sigma = \langle \Delta, \mathcal{I}_c, \mathcal{I}_r \rangle$ where: (*i*) Δ is a non-empty (possibly countably infinite) set; (*ii*) $\mathcal{I}_c$ is a function that maps each $d \in \Delta$ into a subset of Δ; and (*iii*) $\mathcal{I}_r$ is a function that maps each $d \in \Delta$ into a subset of $\Delta \times \Delta$. In other words, Σ treats every element of Δ simultaneously as: (*i*) an individual; (*ii*) a unary relation, i.e., a concept, through $\mathcal{I}_c$; and (*iii*) a binary relation, i.e., a role, through $\mathcal{I}_r$.

An *interpretation* over Σ is a pair $\mathcal{I} = \langle \Sigma, \mathcal{I}_o \rangle$, where $\Sigma = \langle \Delta, \mathcal{I}_c, \mathcal{I}_r \rangle$ is an interpretation structure, and $\mathcal{I}_o$ is a function that maps: (*i*) each element of $\mathcal{S}$ to a single domain object of Δ; and (*ii*) each element $C/n \in OP(\mathcal{L})$ to an n-ary function $C^{\mathcal{I}_o} : \Delta^n \to \Delta$ that satisfies the conditions characterizing the operator C/n. In particular, the conditions for the operators in $OP(\mathcal{SHIQ})$ are described in Figure 1. We extend $\mathcal{I}_o$ to expressions in $\mathcal{E}_{\mathcal{L}}(\mathcal{S})$ inductively as follows: if $C/n \in OP(\mathcal{L})$, then $(C(E_1, \ldots, E_n))^{\mathcal{I}_o} = C^{\mathcal{I}_o}(E_1^{\mathcal{I}_o}, \ldots, E_n^{\mathcal{I}_o})$.

To interpret non-ground terms, we need assignments over interpretations. An *assignment* μ over $\langle \Sigma, \mathcal{I}_o \rangle$ is a function $\mu : \mathcal{V} \to \Delta$.

We are now ready to describe how to interpret terms in $Hi(\mathcal{L})$. Given an interpretation $\mathcal{I} = \langle \Sigma, \mathcal{I}_o \rangle$ and an assignment μ over $\mathcal{I}$, we define the function $(\cdot)^{\mathcal{I}_o,\mu} : \tau(\mathcal{S}, \mathcal{V}) \to \Delta$ as follows:

- if $t \in \mathcal{S}$ then $t^{\mathcal{I}_o,\mu} = t^{\mathcal{I}_o}$;
- if $t \in \mathcal{V}$ then $t^{\mathcal{I}_o,\mu} = \mu(t)$;
- if t is of the form $C(t_1, \ldots, t_n)$, then $t^{\mathcal{I}_o,\mu} = C^{\mathcal{I}_o}(t_1^{\mathcal{I}_o,\mu}, \ldots, t_n^{\mathcal{I}_o,\mu})$.

Satisfaction of an assertion with respect to an interpretation $\mathcal{I}$ and an assignment μ over $\mathcal{I}$ is defined based on the semantics of the meta-predicates in $MP(\mathcal{L})$. For the meta-predicates used in $\mathcal{SHIQ}$, satisfaction in $\mathcal{I}, \mu$ is defined as follows:

- $\mathcal{I}, \mu \models Inst_C(E_1, E_2)$ if $E_1^{\mathcal{I}_o,\mu} \in (E_2^{\mathcal{I}_o,\mu})^{\mathcal{I}_c}$;
- $\mathcal{I}, \mu \models Inst_R(E_1, E_2, E_3)$ if $\langle E_1^{\mathcal{I}_o,\mu}, E_2^{\mathcal{I}_o,\mu} \rangle \in (E_3^{\mathcal{I}_o,\mu})^{\mathcal{I}_r}$;
- $\mathcal{I}, \mu \models Isa_C(E_1, E_2)$ if $(E_1^{\mathcal{I}_o,\mu})^{\mathcal{I}_c} \subseteq (E_2^{\mathcal{I}_o,\mu})^{\mathcal{I}_c}$;
- $\mathcal{I}, \mu \models Isa_R(E_1, E_2)$ if $(E_1^{\mathcal{I}_o,\mu})^{\mathcal{I}_r} \subseteq (E_2^{\mathcal{I}_o,\mu})^{\mathcal{I}_r}$;
- $\mathcal{I}, \mu \models Tran(E)$ if $(E^{\mathcal{I}_o,\mu})^{\mathcal{I}_r}$ is a transitive relation.

A $Hi(\mathcal{L})$ KB $\mathcal{H}$ is satisfied by $\mathcal{I}$ if all the assertions in $\mathcal{H}$ are satisfied by $\mathcal{I}$.[2] As usual, the interpretations $\mathcal{I}$ satisfying $\mathcal{H}$ are called the *models* of $\mathcal{H}$. A $Hi(\mathcal{L})$ KB $\mathcal{H}$ is *satisfiable* if it has at least one model.

Let $\mathcal{I}$ be an interpretation and μ an assignment over $\mathcal{I}$. A Boolean HCQ q of the form $q \leftarrow a_1, \ldots, a_n$ is *satisfied* in $\mathcal{I}, \mu$ if every assertion a_i is satisfied in $\mathcal{I}, \mu$. A Boolean HUCQ Q is *satisfied* in $\mathcal{I}, \mu$ if there exists a Boolean HCQ $q \in Q$ that is satisfied in $\mathcal{I}, \mu$. A Boolean HUCQ Q is satisfied in an interpretation $\mathcal{I}$, written $\mathcal{I} \models Q$, if there exists an assignment μ over $\mathcal{I}$ such that Q is satisifed in $\mathcal{I}, \mu$. Given a Boolean HUCQ Q and a $Hi(\mathcal{L})$ KB $\mathcal{H}$, we say that Q is *logically implied* by $\mathcal{H}$ (denoted by $\mathcal{H} \models Q$) if for each model $\mathcal{I}$ of $\mathcal{H}$ there exists an assignment μ such that Q is satisfied by $\mathcal{I}, \mu$.

Given a non-Boolean HUCQ q of the form $q(t_1, \ldots, t_n) \leftarrow a_1, \ldots, a_m$, a grounding substitution of q is a substitution θ such that $t_1\theta, \ldots, t_n\theta$ are ground terms. We call $t_1\theta, \ldots, t_n\theta$ a grounding tuple. The set of *certain answers* to q in $\mathcal{H}$ is the set of grounding tuples $t_1\theta, \ldots, t_n\theta$ that make the Boolean query $q_\theta \leftarrow a_1\theta, \ldots, a_n\theta$ logically implied by $\mathcal{H}$. Notice that, in general, the set of certain answers may be infinite even if the KB is finite. Therefore, it is of interest to define suitable notions of safeness,

[2] We do not need to mention assignments here, since all assertions in $\mathcal{H}$ are ground.

which guarantee that the set of answers is bounded. This issue, however, is beyond the scope of the present paper.

Indeed, in this paper, we focus on Boolean queries only, so as to address the computation of certain answers as a decision problem. Also, in our analysis, we measure the computational complexity in three different ways: with respect to the size of the whole KB (*KB complexity*), with respect to the size of the part of the KB formed by the assertions involving only the meta-predicates $Inst_C/2, Inst_R/3$ (*instance complexity*), and with respect to the size of the KB and the query together (*combined complexity*).

3 Satisfiability in $Hi(\mathcal{SHIQ})$

In this section we study the computational characterization of KB satisfiability in the higher-order DL $Hi(\mathcal{SHIQ})$. Query answering in the same DL is addressed in the next section.

We start by defining a translation Π from $Hi(\mathcal{SHIQ})$ to $\mathcal{SHIQ}$. First, we define three injective functions

$$\nu_O : \mathcal{E}_{\mathcal{SHIQ}}(\mathcal{S}) \to \mathcal{S}^o, \qquad \nu_C : \mathcal{E}_{\mathcal{SHIQ}}(\mathcal{S}) \to \mathcal{S}^c, \qquad \nu_R : \mathcal{E}_{\mathcal{SHIQ}}(\mathcal{S}) \to \mathcal{S}^r$$

where $\mathcal{S}^o$, $\mathcal{S}^c$ and $\mathcal{S}^r$ are three mutually disjoint alphabets of names, each one disjoint from $\mathcal{S}$. Then, we inductively define two functions τ_C and τ_R as follows:

- if $S \in \mathcal{S}$, then $\tau_C(S) = \nu_C(S)$ and $\tau_R(S) = \nu_R(S)$;
- $\tau_C(Not(E)) = Not(\tau_C(E))$;
- $\tau_C(And(E_1, E_2)) = And(\tau_C(E_1), \tau_C(E_2))$;
- $\tau_C(AtLeastQ_n(E_1, E_2)) = AtLeastQ_n(\tau_R(E_1), \tau_C(E_2))$;
- $\tau_R(Inv(E)) = Inv(\tau_R(E))$.

Now, let $Expr(\mathcal{H})$ denote the set of ground expressions occurring in $\mathcal{H}$ (notice that every subexpression of an expression occurring in $\mathcal{H}$ also belongs to $Expr(\mathcal{H})$). Then, given a $Hi(\mathcal{SHIQ})$ KB $\mathcal{H}$, we inductively define the $\mathcal{SHIQ}$ KB $\Pi(\mathcal{H})$ as follows:

1. if $Not(E) \in Expr(\mathcal{H})$, then $\nu_C(Not(E)) \equiv \tau_C(Not(E)) \in \Pi(\mathcal{H})$;
2. if $Inv(E) \in Expr(\mathcal{H})$, then $\nu_R(Inv(E)) \equiv \tau_R(Inv(E)) \in \Pi(\mathcal{H})$;
3. if $And(E_1, E_2) \in Expr(\mathcal{H})$, then $\nu_C(And(E_1, E_2)) \equiv \tau_C(And(E_1, E_2)) \in \Pi(\mathcal{H})$;
4. if $AtLeastQ_n(E_1, E_2) \in Expr(\mathcal{H})$, then $\nu_C(AtLeastQ_n(E_1, E_2)) \equiv \tau_C(AtLeastQ_n(E_1, E_2)) \in \Pi(\mathcal{H})$;
5. if $Inst_C(E_1, E_2) \in \mathcal{H}$, then $\nu_C(E_1)(\nu_O(E_2)) \in \Pi(\mathcal{H})$;
6. if $Inst_R(E_1, E_2, E_3) \in \mathcal{H}$, then $\nu_R(E_1)(\nu_O(E_2), \nu_O(E_3)) \in \Pi(\mathcal{H})$;
7. if $Isa_C(E_1, E_2) \in \mathcal{H}$, then $\nu_C(E_1) \sqsubseteq \nu_C(E_2) \in \Pi(\mathcal{H})$;
8. if $Isa_R(E_1, E_2) \in \mathcal{H}$, then $\nu_R(E_1) \sqsubseteq \nu_R(E_2) \in \Pi(\mathcal{H})$;
9. if $Tran(E) \in \mathcal{H}$, then $Tran(\nu_R(E)) \in \Pi(\mathcal{H})$.

Informally, the above translation, when applied to a $Hi(\mathcal{SHIQ})$ DL $\mathcal{H}$, provides a $\mathcal{SHIQ}$ KB $\Pi(\mathcal{H})$ in which for every ground term E occurring in $\mathcal{H}$ (notice that E may be a subterm of another term occurring in $\mathcal{H}$) there exists a concept name $\nu_C(E)$ (and a role name $\nu_R(E)$) that is defined, through the use of the function τ_C (respectively, τ_R) as equivalent to the term E seen as a concept (respectively, role) expression.

Based on the above translation, we get the first of our main results, namely a reduction of KB satisfiability in $Hi(\mathcal{SHIQ})$ to KB satisfiability in $\mathcal{SHIQ}$.

Theorem 1. *A $Hi(\mathcal{SHIQ})$ KB $\mathcal{H}$ is satisfiable iff the $\mathcal{SHIQ}$ KB $\Pi(\mathcal{H})$ is satisfiable.*

Proof (sketch). One direction of the proof is trivial: if there exists a model $\mathcal{I}$ for $\mathcal{H}$, then based on $\mathcal{I}$ it is immediate to define a model $\mathcal{I}'$ for $\Pi(\mathcal{H})$ which interprets objects according to $\mathcal{I}_o$, atomic concepts (i.e., concepts denoted by names) according to $\mathcal{I}_c$, and atomic roles according to $\mathcal{I}_r$. As for the other direction, given a model $\mathcal{I}$ for $\Pi(\mathcal{H})$ over a domain Δ, it is possible to define a model $\mathcal{I}'$ for $\mathcal{H}$ by considering the disjoint union of a countably infinite number of copies of $\mathcal{I}$ (over a countably infinite number of copies of Δ), and defining $\mathcal{I}'_o$ so that it coincides with $\mathcal{I}_o$ on the expressions occurring in $\mathcal{H}$, while every expression that does not occur in $\mathcal{H}$ is interpreted by $\mathcal{I}'_o$ to an element of the extra copies of Δ. Then, it is easy to define $\mathcal{I}'_c$ and $\mathcal{I}'_r$ in order to satisfy the semantic conditions of Figure 1. □

From the above theorem, and the computational characterization of KB satisfiability in $\mathcal{SHIQ}$ [3], we are able to provide the computational characterization of KB satisfiability in $Hi(\mathcal{SHIQ})$.

Theorem 2. *KB satisfiability in $Hi(\mathcal{SHIQ})$ is EXPTIME-complete w.r.t. KB complexity, and coNP-complete w.r.t. instance complexity.*

4 Query answering in $Hi(\mathcal{SHIQ})$

In this section we study query answering in $Hi(\mathcal{SHIQ})$. In particular, we restrict our attention to a specfic class of HUCQs, which we call *guarded*. For the definition of this class of queries, we need the notions of object position, concept position, and role position, whose goal is to characterize the various argument positions in both atoms and terms. If we use symbol $\mathbf{O}$ to mark object positions, symbol $\mathbf{C}$ to mark concept positions, and symobl $\mathbf{R}$ to mark role positions, then we have:

$$Inst_C(\mathbf{O},\mathbf{C})\ \ Inst_R(\mathbf{O},\mathbf{O},\mathbf{R})\ \ Isa_C(\mathbf{C},\mathbf{C})\ \ Isa_R(\mathbf{R},\mathbf{R})\ \ Tran(\mathbf{R})$$
$$Not(\mathbf{C})\ \ And(\mathbf{C},\mathbf{C})\ \ Inv(\mathbf{R})\ \ AtLeastQ_n(\mathbf{R},\mathbf{C})$$

Now, a HCQ q is called *guarded* if, for every variable x occurring in an ISA-atom of q, x also occurs in a concept or role position of an instance-atom of q. A HUCQ is called guarded is every HCQ q in Q is guarded.

We start our analysis of query answering by showing that answering guarded HUCQs is coNP-hard w.r.t. KB complexity (actually, w.r.t. instance complexity only) and Π_2^p-complete w.r.t combined complexity, as soon as the DL admits the $Inst_C$, $Inst_R$ and Isa_C meta-predicates (and even if the DL does not allow for any logical operator).

Theorem 3. *Let $\mathcal{L}$ be a DL such that $MP(\mathcal{L})$ contains the meta-predicates $Inst_C$, $Inst_R$ and Isa_C. Answering guarded HUCQs over $Hi(\mathcal{L})$ KBs is coNP-hard w.r.t. instance complexity, and Π_2^p-hard w.r.t. combined complexity, even if $OP(\mathcal{L}) = \emptyset$.*

We remark that the previous theorem implies that answering guarded HUCQs is intractable w.r.t. instance (and KB) complexity not only in $Hi(\mathcal{SHIQ})$, but in *all* the

DLs currently studied, since all DLs comprise the meta-predicates $Inst_C$, $Inst_R$ and Isa_C.

We now provide a technique for query answering over $Hi(\mathcal{SHIQ})$ KBs, which is based on the reduction to $\mathcal{SHIQ}$ provided by the function $\Pi()$ defined for KB satisfiability. For query answering, however, the function $\Pi()$ must be extended to account for expressions occurring in the query; moreover, we also need to define a translation π of HUCQs. Such functions are defined below.

Let Q be a HUCQ. We say that Q is a *metaground* HUCQ if it does not contain any variable in concept or role position. Moreover, we say that Q is an *instance* HUCQ if it only contains instance-atoms.

In the following, given a HUCQ Q, we denote by $Expr(Q)$ the set of ground expressions occurring in Q. Now let q be a HCQ and let $e_1, \dots, e_k$ be the ground expressions occurring as arguments of ISA-atoms in q. We define inductively the set of expressions $conj_{ISA}(q)$ as follows:

$$\begin{array}{l} \mathcal{C}_1 = \{e_1, \dots, e_k\} \\ \mathcal{C}_{i+1} = \{And(e, e_0) | e \in \mathcal{C}_i \text{ and } e_o \in \mathcal{C}_1\} \\ conj_{ISA}(q) = \mathcal{C}_k \end{array}$$

Informally, $conj_{ISA}(q)$ denotes the set of all the possible conjunctions of ground expressions occurring as arguments of ISA-atoms in q. We are now ready to define the set of ground expressions $Expr(\mathcal{H}, Q)$ as follows:

$$Expr(\mathcal{H}, Q) = Expr(\mathcal{H}) \cup Expr(Q) \cup \bigcup_{q \in Q} conj_{ISA}(q)$$

As we will show in the following, $Expr(\mathcal{H}, Q)$ constitutes the set of ground expressions that we use for grounding metavariables. Notice that $Expr(\mathcal{H}, Q)$ has size polynomial in the size of $\mathcal{H}$.

Let q be a HCQ. An *$\mathcal{H}$-metaground instantiation of q* is a HCQ obtained from q by replacing every variable occurring in at least one concept or role position with an expression of $Expr(\mathcal{H}, q)$. Given a HCQ q, we define $metaground(q, \mathcal{H})$ as the HUCQ corresponding to the union of all the $\mathcal{H}$-metaground instantiations of q. If there are no ground terms occurring in $\mathcal{H}$ (i.e., $\mathcal{H}$ is empty), we define $metaground(q, \mathcal{H})$ to be the HCQ obtained from q by replacing all variables occurring in concept and role positions with any name in $\mathcal{S}$. Given a HUCQ Q, we define $metaground(Q, \mathcal{H}) = \bigcup_{q \in Q} metaground(q, \mathcal{H})$.

Given a metaground HUCQ Q, we denote by $\pi(Q, \mathcal{H})$ the standard UCQ obtained from $metaground(Q, \mathcal{H})$ by: (i) replacing every ground term E occurring as an argument in object position of an atom in Q with $\nu_O(E)$; (ii) replacing every ground term E occurring as an argument in concept position of an atom in Q with $\nu_C(E)$; (iii) replacing every ground term E occurring as an argument in role position of an atom in Q with $\nu_R(E)$.

Finally, given a $Hi(\mathcal{SHIQ})$ KB $\mathcal{H}$ and a HUCQ Q, we denote by $\Pi(\mathcal{H}, Q)$ the $\mathcal{SHIQ}$ KB obtained starting from $\Pi(\mathcal{H})$ and adding, for every ground term E that occurs in $metaground(Q, \mathcal{H})$ and does not occur in $\mathcal{H}$, the inclusion assertions generated by the first 5 items in the definition of $\Pi(\mathcal{H})$ above.

Now we restrict our attention to both *metaground* and *instance* queries. For this class of HUCQs, the following property can be easily proved.

Theorem 4. *Let $\mathcal{H}$ be a $Hi(\mathcal{SHIQ})$ KB, and let Q be a metaground instance HUCQ. Then, $\mathcal{H} \models Q$ iff $\Pi(\mathcal{H}, Q) \models \pi(Q, \mathcal{H})$.*

Based on the known computational characterization of answering "standard" UCQs, i.e., both *metaground* and *instance* UCQs, in $\mathcal{SHIQ}$ [9, 6, 13], we immediately get the following result.

Theorem 5. *Answering metaground instance HUCQs over $Hi(\mathcal{SHIQ})$ KBs is coNP-complete w.r.t. instance complexity, EXPTIME-complete w.r.t. KB complexity, and 2-EXPTIME-complete w.r.t. combined complexity.*

We can actually extend the previous theorem to the whole class of instance HUCQs. First, we show the following crucial property, which holds for the whole class of guarded HUCQs.

Theorem 6. *Let $\mathcal{H}$ be a $Hi(\mathcal{SHIQ})$ KB, and let Q be a guarded HUCQ. $\mathcal{H} \models Q$ iff $\mathcal{H} \models metaground(Q, \mathcal{H})$.*

Proof (sketch). One direction (if $\mathcal{H} \models metaground(Q, \mathcal{H})$ then $\mathcal{H} \models Q$) is trivial. The proof of the other direction is quite involved. First, the following property (*) can be shown: if $\mathcal{H} \models Q$ then $\mathcal{H} \models metaground(Q)$, where $metaground(Q)$ is the query obtained from Q through the meta-grounding of the meta-variables over the set of *all* expressions of the language (not only those terms occurring in $Expr(\mathcal{H}, Q)$). Now suppose $\mathcal{H} \models Q$. If $\mathcal{H} \not\models metaground(Q, \mathcal{H})$, then there exists a model $\mathcal{I}$ for $\mathcal{H}$ such that $\mathcal{I} \models metaground(Q)$ and $\mathcal{I} \not\models metaground(Q, H)$. It is now possible to define a model $\mathcal{I}'$ for $\mathcal{H}$ which is essentially the disjoint union of a countably infinite number of copies of $\mathcal{I}$, in which the function $\mathcal{I}'_o$ is defined in such a way that $\mathcal{I}' \not\models metaground(Q)$, which contradicts the above property (*). Consequently, $\mathcal{H} \models metaground(Q, \mathcal{H})$. □

Theorem 6, Theorem 4, and Theorem 5 allow us to immediately derive the computational characterization of query answering in $Hi(\mathcal{SHIQ})$ for the whole class of instance HUCQs.

Theorem 7. *Answering instance HUCQs over $Hi(\mathcal{SHIQ})$ KBs is coNP-complete w.r.t. instance complexity, EXPTIME-complete w.r.t. KB complexity, and 2-EXPTIME-complete w.r.t. combined complexity.*

In order to go beyond instance HUCQs, and answer guarded HUCQs in $Hi(\mathcal{SHIQ})$, we now define a technique which reduces this problem to answering standard UCQs in $\mathcal{SHIQ}$.

In the following, we call *intensional (or, TBox) assertion* every assertion using one of the meta-predicates Isa_C, Isa_R, and $Tran$. Moreover, given a KB $\mathcal{H}$ and a HUCQ Q, we define $TA_{\mathcal{H},Q}$ to be the set of all the intensional assertions in $\mathcal{SHIQ}$ that can be obtained from the set of ground terms occurring in $Expr(\mathcal{H}, Q)$.

Let $\mathcal{T}'$ be a subset of $TA_{\mathcal{H},Q}$. We say that $\mathcal{T}'$ is *coherent with* $\mathcal{H}$ iff $\mathcal{T} \subseteq \mathcal{T}'$, where $\mathcal{T}$ is the set of TBox assertions occurring in $\mathcal{H}$, and $\mathcal{T}' \cup \mathcal{H} \not\models \phi$ for every

$\phi \in TA(Expr(\mathcal{H}, Q)) - \mathcal{T}'$. Then, we denote by $IntEval(Q, \mathcal{H}, \mathcal{T}')$ the metaground instance HUCQ Q' obtained starting from $Q' = metaground(Q, \mathcal{H})$ and then evaluating every intensional assertion over $\mathcal{T}'$ as follows:

- if ϕ is an intensional assertion occurring in a HCQ $q \in Q'$ and $\phi \in \mathcal{T}'$, then eliminate ϕ from q;
- if ϕ is an intensional assertion occurring in a HCQ $q \in Q'$ and $\phi \notin \mathcal{T}'$, then eliminate q from Q'.

Finally, we define $KB_{\mathcal{SHIQ}}(\mathcal{H}, \mathcal{T}', Q)$ as the $\mathcal{SHIQ}$ KB[3] obtained starting from $\mathcal{K}' = \Pi(\mathcal{H}', Q)$ (where $\mathcal{H}' = \mathcal{T}' \cup \mathcal{H}$) and then adding to $\mathcal{K}'$ the following assertions for every TBox assertion $\alpha \in TA(Expr(\mathcal{H}, Q)) - \mathcal{T}'$:

- if $\alpha = Isa_C(E_1, E_2)$ then add to $\mathcal{K}'$ the ABox assertions $\nu_C(E_1)(n)$ and $\nu_C(Not(E_2))(n)$, where n is a new individual name in $\mathcal{K}'$;
- if $\alpha = Isa_R(E_1, E_2)$ then add to $\mathcal{K}'$ the TBox assertion (role disjointness) $\nu_R(E_2) \sqsubseteq \neg Aux_i$, where i is such that Aux_i is a new role name in $\mathcal{K}'$, and the ABox assertions $\nu_R(E_1)(n_1, n_2)$ and $Aux_i(n_1, n_2)$, where n_1, n_2 are new individual names in $\mathcal{K}'$;
- if $\alpha = Tran(E)$ then add to $\mathcal{K}'$ the TBox assertion (role disjointness) $\nu_R(E) \sqsubseteq \neg Aux_i$, where i is such that Aux_i is a new role name in $\mathcal{K}'$, and the ABox assertions $\nu_R(E)(n_1, n_2)$, $\nu_R(E)(n_2, n_3)$ and $Aux_i(n_1, n_3)$, where n_1, n_2, n_3 are new individual names in $\mathcal{K}'$.

Intuitively, $KB_{\mathcal{SHIQ}}(\mathcal{H}, \mathcal{T}', Q)$ is such that, if $\alpha \in TA(Expr(\mathcal{H}, Q)) - \mathcal{T}'$, then α is forced to be false in every model of $KB_{\mathcal{SHIQ}}(\mathcal{H}, \mathcal{T}', Q)$.

The following theorem (whose proof relies on Theorem 6) reduces answering guarded HUCQs in $Hi(\mathcal{SHIQ})$ to answering standard UCQs in $\mathcal{SHIQ}$.

Theorem 8. *Let $\mathcal{H}$ be a $Hi(\mathcal{SHIQ})$ KB, and let Q be a guarded HUCQ. Then, $\mathcal{H} \not\models Q$ iff there exists a subset $\mathcal{T}'$ of $TA_{\mathcal{H},Q}$ such that $\mathcal{T}'$ is coherent with $\mathcal{H}$, and $KB_{\mathcal{SHIQ}}(\mathcal{H}, \mathcal{T}', Q) \not\models \pi(IntEval(Q, \mathcal{H}, \mathcal{T}'), \mathcal{H})$.*

Based on Theorem 8, Theorem 4 and Theorem 5, we get the computational characterization of answering guarded HUCQs in $Hi(\mathcal{SHIQ})$.

Theorem 9. *Answering guarded HUCQs over $Hi(\mathcal{SHIQ})$ KBs is coNP-complete w.r.t. instance complexity, EXPTIME-complete w.r.t. KB complexity, and 2-EXPTIME-complete w.r.t. combined complexity.*

5 Conclusions

In this paper we have presented a general mechanism for defining a family of Description Logics for domain metamodeling. We have shown how, starting from any DL $\mathcal{L}$,

[3] Actually, $KB_{\mathcal{SHIQ}}(\mathcal{H}, \mathcal{T}', Q)$ is a $\mathcal{SHIQ}$ KB with role disjointness assertions, however adding this kind of axioms to $\mathcal{SHIQ}$ does not change the complexity of query answering.

one can define a higher-order logic, called $Hi(\mathcal{L})$, that adds to $\mathcal{L}$ metamodeling features. Also, we have presented algorithms for both satisfiability and query answering in a specific expressive higher-order Description Logic, namely $Hi(\mathcal{SHIQ})$.

The present paper can be seen as an extension of both the approach and the results in [8]: in particular, on the one hand we have extended the results presented there to $\mathcal{SHIQ}$, and on the other hand we have characterized the computational complexity for a larger class of meta-queries.

The research presented here can be continued along different lines. First, while the query answering algorithm presented in this paper is suited for the class of guarded HUCQs, it is our goal to address query answering for the whole class of HUCQs. Second, it would be interesting to see whether more metamodeling features can be added to the query language. In particular, one might wonder whether the query answering method described in this paper can be extended to deal with the case where variables can appear freely within the terms in the query atoms (see Example 4 in Section 2). Unfortunately, our first investigation on this subject shows that allowing for a more flexible use of variables in the queries easily leads to undecidability of query answering. Another interesting direction is to add both domain and full metamodeling capabilities to tractable DLs, in particular, the DLs of the DL-Lite family [5], so as to check whether reasoning remains tractable in the resulting logics.

Finally, we remark that *punning*, i.e., using the same name for different elements of the ontology (for example, an individual and a concept), has been introduced in OWL 2 [4]. While punning can be treated trivially in classical reasoning tasks over the DL ontology, it poses interesting problems in the context of query processing. In particular, if variables are not typed a priori, punning introduces the kind of meta-querying discussed in this paper. Indeed, the DL query language presented in this paper is the first one (to our knowledge) that exploits punning in queries, since it allows for expressing joins involving variables which simultaneously denote both individuals and predicates.

Acknowledgments This research has been partially supported by MIUR FIRB project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT).

References

1. F. Angiulli, R. Ben-Eliyahu-Zohary, G. Ianni, and L. Palopoli. Computational properties of metaquerying problems. *ACM Trans. on Computational Logic*, 4(2):149–180, 2003.
2. G. Attardi and M. Simi. Consistency and completeness of OMEGA, a logic for knowledge representation. In *Proc. of the 7th Int. Joint Conf. on Artificial Intelligence (IJCAI'81)*, pages 504–510, 1981.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
4. L. Badea. Reifying concepts in description logics. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI'97)*, pages 142–147, 1997.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.

[4] http://www.w3.org/2007/OWL/wiki/OWL_Working_Group

6. D. Calvanese, G. De Giacomo, and M. Lenzerini. Conjunctive query containment and answering under description logics constraints. *ACM Trans. on Computational Logic*, 9(3):22.1–22.31, 2008.
7. W. Chen, M. Kifer, and D. S. Warren. HILOG: A foundation for higher-order logic programming. *J. of Logic Programming*, 15(3):187–230, 1993.
8. G. De Giacomo, M. Lenzerini, and R. Rosati. Towards higher-order *DL-Lite*. In *Proc. of the 2008 Description Logic Workshop (DL 2008)*, volume 353 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, 2008.
9. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 399–404, 2007.
10. F. Lehmann, editor. *Semantic Networks in Artificial Intelligence*. Pergamon Press, Oxford (United Kingdom), 1992.
11. B. Motik. On the properties of metamodeling in OWL. *J. of Logic and Computation*, 17(4):617–637, 2007.
12. J. Mylopoulos, P. A. Bernstein, and H. K. T. Wong. A language facility for designing database-intensive applications. *ACM Trans. on Database Systems*, 5(2):185–207, 1980.
13. M. Ortiz, D. Calvanese, and T. Eiter. Data complexity of query answering in expressive description logics via tableaux. *J. of Automated Reasoning*, 41(1):61–98, 2008.
14. J. Z. Pan and I. Horrocks. OWL FA: a metamodeling extension of OWL DL. In *Proc. of the 15th Int. World Wide Web Conf. (WWW 2006)*, pages 1065–1066, 2006.