From: AAAI Technical Report WS-96-05. Compilation copyright © 1996, AAAI (www.aaai.org). All rights reserved.

Datalog and Description Logics: Expressive Power

Preliminary report

Marco Cadoli
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
cadoli@dis.uniroma1.it

Luigi Palopoli
Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria
I-87036 Rende (CS), Italy
palopoli@unical.it

Maurizio Lenzerini
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
lenzerini@dis.uniroma1.it

Abstract

This paper investigates the possibility of exploiting formal analysis tools developed in the database field for the purpose of studying the expressive power of description logics augmented with rule-based query languages. We report a preliminary analysis on the expressive power of such hybrid languages. Two specific languages coupling a terminological component with Horn rules are considered. It is shown that: (1) the former language defines all database collections expressed by skolemized universally quantified second order formulae where quantified predicates are monadic, and (2) the latter language defines all database collections expressed by skolemized universally quantified second order formulae where quantified predicates are dyadic or monadic.

1 Introduction

Recently there was some attention on integration of description logics of the $\mathcal{AL}$-family with rule-based languages for querying relational data bases such as Datalog ([Donini *et al.*, 1991; Levy and Rousset, 1996b; 1996a]).

Computational analysis carried out in such papers is limited to *complexity*: i.e., how much time/space it is needed to answer to a specific query, the input being the relational data base, and/or the ABox, and/or the TBox, and/or the rules? The goal of this paper is to perform some considerations and give some preliminary results on the *expressiveness* of such hybrid languages. The importance of formal analysis of expressive power of query languages is acknowledged in the data base community [Kanellakis, 1990].

Intuitively, the expressive power of a query language tells us what "properties" it is possible to extract from a knowledge base. In the context of databases, expressive power of query languages has been measured in at least three different ways:

1. With respect to a specific property, such as transitive closure. For example, it is well-known that there is no fixed query in relational calculus that, for any graph G encoded as a relation $edge/2$ in the obvious way, determines whether the transitive closure of G contains a specific edge or not. Vice-versa, such a query does exist in Datalog.
2. With respect to a set of logical formulae, such as first- or second-order logic. As an example, relational calculus can express exactly the set of first-order properties over finite structures. "While queries" [Abiteboul and Vianu, 1992] can express exactly the set of second-order properties over finite structures.
3. With respect to a complexity class, such as P, NP, coNP, PSPACE, etc. As an example, Datalog with stable negation can express all NP properties of finite structures [Schlipf, 1990], e.g., whether a graph is 3-colorable or not.

One of the major results in this field, known as "Fagin's theorem" [Fagin, 1974], provides the basis for unifying the second and the third modalities. It says that the set of NP properties coincides with the set of properties expressed by existentially quantified second-order formulae. Such a result has been generalized to other complexity classes and sets of logical formulae.

Formal studies of expressive power of description logics have been recently pursued. In particular, [Borgida,

1996] shows some expressiveness results with respect to the second measure, by showing that description languages built using constructors usually considered in the description logic literature are characterized by subsets of first-order logic allowing only three variable symbols. [Baader, 1996] gives a methodological contribution, pointing out that expressiveness must be defined within a precise formal framework, and proposes the model-theoretic approach for the characterization of expressive power. Interestingly, he shows that the complexity of inference of two equally expressive languages may be different.

The major goals of our research are: 1) to investigate the possibility of exploiting the formal analysis tools developed in the database field within the context of description logics augmented with rules, and 2) to give some preliminary results about the expressive power of such hybrid systems.

The first difficulty that we encounter in such a project is the different notion of extensional knowledge that databases and description logics assume. In the first case, an extensional database denotes a single finite structure with a fixed domain of interpretation, whereas this is obviously not true for an ABox expressed in a description logic with existential and/or disjunctions. This aspect disallows direct exploitation of formal tools such as Fagin's theorem. Such a problem is also evident if we translate a relational database D into a first-order formula $\phi(D)$: in fact Reiter [Reiter, 1984] showed that, in order to preserve the intended meaning of D, $\phi(D)$ must be completed with sentences such as the domain-closure axiom, the unique-name axiom and the closed-world assumption, which constrain the set of allowed interpretations.

In the present work we assume that the ABox is empty. The intensional part is made out of a TBox and a set of Horn rules. As usual, we assume that predicates appearing either in the heads of rules or in the relational part (also called *ordinary* predicates) do not occur in the TBox. Therefore, in the following we shall refer to hybrid knowledge bases Δ consisting of three components: the TBox, denoted $\Delta_{\mathcal{T}}$, a finite set of Horn rules, denoted $\Delta_{\mathcal{R}}$ and, finally, a finite set of facts (i.e., a relational database), denoted $\Delta_{\mathcal{E}}$. $\Delta_{\mathcal{E}}$ is the extensional component of Δ, whereas $\Delta_{\mathcal{R}} \cup \Delta_{\mathcal{T}}$ forms the intensional component of Δ. The intensional component of a knowledge base defines a query that is evaluated over its extensional component.

It is easy to see that Datalog, when augmented with inclusion axioms typical of description logics, is able to capture some coNP-complete queries. As an example, to check 3-colorability of a graph $G = \langle V, A \rangle$ encoded as a set of facts $\Delta_{\mathcal{E}} = \{edge(a,b) | (a,b) \in A\}$, we can write a 2-components query:

TBox ($\Delta_{\mathcal{T}}$):

$$\begin{aligned} \top &\sqsubseteq red \sqcup green \sqcup blue \\ red &\sqsubseteq \neg green \\ green &\sqsubseteq \neg blue \\ blue &\sqsubseteq \neg red \end{aligned}$$

Datalog rules ($\Delta_{\mathcal{R}}$):

$$\begin{aligned} non\_3\_col &\leftarrow edge(X,Y), red(X), red(Y). \\ non\_3\_col &\leftarrow edge(X,Y), blue(X), blue(Y). \\ non\_3\_col &\leftarrow edge(X,Y), green(X), green(Y). \end{aligned}$$

The inclusion axioms in the TBox impose that the three colors actually partition the active domain, and indeed $\Delta_{\mathcal{T}} \cup \Delta_{\mathcal{R}} \cup \Delta_{\mathcal{E}} \models non\_3\_col$ iff G is not 3-colorable (where $\models$ denotes the usual logical consequence operator, i.e., validity in all models). In the terminology of [Levy and Rousset, 1996b], $\Delta_{\mathcal{T}}$ is "acyclic", and $\Delta_{\mathcal{R}}$ is "non-recursive". Moreover $\Delta_{\mathcal{T}}$ belongs to the class CARIN-MARC of "maximal (decidable) $\mathcal{ALCNR}$ Recursive CARIN", which includes the constructors $\sqcup, \sqcap, (\geq n\ R), \exists R.C$, and negation on primitive concepts. $\Delta_{\mathcal{R}}$ is "role-safe", i.e., each of its rules is such that for every atom of the form $R(x,y)$ in the antecedent, where R is a role, then either x or y appear in an ordinary atom of the antecedent. In fact, the TBox is a set of *inclusion axioms* [Buchheit *et al.*, 1993], and concept constructors used in the TBox are just boolean. Actually, this is an $\mathcal{AL}$-log program [Donini *et al.*, 1991].

The above example just proves that the *data complexity* (i.e., complexity considering the extensional component as the input and the intensional component (query) $\Delta_{\mathcal{T}} \cup \Delta_{\mathcal{R}}$ not part of the input) of $\mathcal{AL}$-log is coNP-hard, but it does not imply that either $\mathcal{AL}$-log or CARIN is able to express *all* queries in coNP. Such a distinction is important since the expressive power of a language is not necessarily the same as its complexity (it is always less than or equal to). Several languages with this property are known, cf. [Abiteboul and Vianu, 1992; Eiter *et al.*, 1994]. As an example, a language which does not capture NP –even if it has an underlying NP-complete problem– has been shown by Stewart in [Stewart, 1991].

We remark that imposing an empty ABox does not imply that there is a fixed domain of interpretation. Nevertheless, in the above example, this is harmless, since non-3-colorability is a property which is preserved for superstructures (i.e., if a graph G is not 3-colorable then any supergraph of G is not 3-colorable as well).

Very often, the TBoxes are assumed to contain predicates (concepts, roles) with fixed arity. In the present paper we restrict our attention to such languages. As a consequence, we use some results on the expressive power of fragments of universal second-order logic with fixed arity [Fagin, 1975]. This is in the spirit of the second modality to measure expressiveness, since those fragments do not naturally correspond to complexity classes, but form a hierarchy within coNP [Fagin, 1990], which is orthogonal to the complexity of definable collections: indeed even the smallest class in the hierarchy (coNP_1) contains coNP complete collections.

2 The expressive power of CARIN knowledge bases

In [Levy and Rousset, 1996b] it is proved that the data complexity (the input being the ABox and/or the relational database) of logical inference in both CARIN-MARC and ROLE-SAFE CARIN is coNP-complete, and we showed in Section 1 that coNP-complete problems are indeed expressed by very simple CARIN knowledge bases, where the Horn component is non-recursive. In this section we prove two results:

1. That ROLE-SAFE CARIN-MARC$^{=,\neq}$ with a non-recursive Horn component expresses all queries that are defined by formulae of the kind $\neg\exists\mathbf{S}\forall\mathbf{X}\exists\mathbf{Y}\phi(\mathbf{X},\mathbf{Y})$, where $\mathbf{S}$ is a list of monadic predicates, ϕ is a quantifier-free first-order formula, and $\mathbf{X}$, $\mathbf{Y}$ are lists of variables, provided that the input finite structure is given in suitable form, as specified below. The exponent $^{=,\neq}$ denotes availability of pre-interpreted symbols for equality and inequality. Formulae $\neg\exists\mathbf{S}\forall\mathbf{X}\exists\mathbf{Y}\phi(\mathbf{X},\mathbf{Y})$ define queries that form a subset of monadic coNP queries (hereafter, called coNP$_1$ cf. [Fagin, 1975; 1990; Cosmadakis, 1993]), which contains several coNP complete queries (e.g., the complement of 3-colorability of a graph). At the moment, we do not know whether this result can be generalized to the entire set of monadic coNP queries.
2. That CARIN-MARC$^{=}$ with a non-recursive Horn component enriched with boolean inclusion axioms on primitive roles expresses all queries that are defined by formulae of the kind $\neg\exists\mathbf{S}'\forall\mathbf{X}\exists\mathbf{Y}\phi'(\mathbf{X},\mathbf{Y})$, where $\mathbf{S}'$ is a list of predicates with arity at most 2 and ϕ' is a quantifier-free first order formula, provided that the input finite structure is given in suitable form, as specified below. Analogously to the previous case, such formulae define queries that form a subset of dyadic coNP (coNP$_2$), and we do not know whether this result can be generalized to the entire set of dyadic coNP queries.

The coNP-completeness of querying the CARIN-MARC and ROLE-SAFE CARIN knowledge bases established in [Levy and Rousset, 1996b] serves also as an upper bound to the expressiveness when the ABox is empty. In other words we know that no CARIN-MARC or ROLE-SAFE CARIN knowledge base can express queries which are not in coNP.

In the following, σ denotes a fixed set of relational symbols not including equality "=" and $\mathbf{S}$ denotes a list of variables ranging over monadic relational symbols distinct from those in σ. By Fagin's theorem [Fagin, 1974], any NP-recognizable collection $\mathbf{D}$ of finite structures over σ is defined by a second-order existentially quantified formula. In particular, NP-recognizable collections $\mathbf{D}$ of finite structures, defined by formulas where all existentially quantified relational symbols are 1-ary, form the set of NP$_1$ collections. As already noted, NP$_1$ indeed includes NP-complete collections (e.g., the collection of 3-colorable graphs). Nevertheless there are polynomial collections of databases (e.g., the collection of dyadic relations with even number of tuples) that are not in NP$_1$ [Fagin, 1990]. The class NP$_1$ has interesting properties: as an example, in [Cosmadakis, 1993] it is proven that monadic NP differs from monadic coNP, while this is a long-standing open question for unbounded NP and coNP.

In the following, we deal with skolemized second-order formulae of the following kind:

$$\phi = (\exists\mathbf{S})(\forall\mathbf{X})(\exists\mathbf{Y})(\theta_1(\mathbf{X},\mathbf{Y}) \vee \cdots \vee \theta_k(\mathbf{X},\mathbf{Y})), \quad (1)$$

where $\theta_1,\ldots,\theta_k$ are conjunctions of literals involving relational symbols in σ and $\mathbf{S}$, plus relational symbol "=", and all relational symbols in $\mathbf{S}$ are constrained to be monadic. Each conjunction θ_i contains occurrence of some variables among $\mathbf{X},\mathbf{Y}$. As usual, "=" is always interpreted as "equality". The set of uninterpreted relational symbols occurring in formula (1) –i.e., $\sigma \cup \mathbf{S}$– will be denoted either by $\mathcal{L}$ or by $\{a_1,\ldots,a_l\}$. In the following art(a) denotes the arity of a predicate a.

We illustrate a method that transforms a formula ψ of the kind (1) and a finite structure D into a ROLE-SAFE CARIN-MARC knowledge base $\Delta(\phi,D)$ and a query γ. Both $\Delta(\phi,D)$ and γ use an enlarged set of relational symbols $\mathcal{L}'$ which is built as follows: (1) each relational symbol $a \in \mathcal{L}$ is in $\mathcal{L}'$; (2) for each relational symbol $a \in \mathcal{L}$ there is one relational symbol $\overline{a}$ with the same arity as a in $\mathcal{L}'$; (3) there is a relational symbol t with the same arity as $\mathbf{X}$ in $\mathcal{L}'$; (4) there is a 0-ary relational symbol e in $\mathcal{L}'$. The ROLE-SAFE CARIN-MARC knowledge base $\Delta(\phi,D)$ is built as follows:

1. for each relational symbol $s \in \mathbf{S}$, the following axioms are in $\Delta_{\mathcal{T}}(\phi)$:
$$\top \sqsubseteq s \sqcup \overline{s}, \qquad s \sqsubseteq \neg\overline{s}$$
2. for each conjunct
$$\theta_i(\mathbf{X},\mathbf{Y}) = \neg w_1(\mathbf{X},\mathbf{Y}) \wedge \cdots \wedge \neg w_n(\mathbf{X},\mathbf{Y}) \wedge w_{n+1}(\mathbf{X},\mathbf{Y}) \wedge \cdots \wedge w_{n+m}(\mathbf{X},\mathbf{Y})$$
$(1 \le i \le k)$ in ψ, the rule
$$t(\mathbf{X}) \leftarrow \overline{v_1}(\mathbf{X},\mathbf{Y}),\ldots,\overline{v_n}(\mathbf{X},\mathbf{Y}), v_{n+1}(\mathbf{X},\mathbf{Y}),\ldots,v_{n+m}(\mathbf{X},\mathbf{Y})$$
is in $\Delta_{\mathcal{R}}(\phi)$, where:
   - $\overline{v_i}$ $(1 \le i \le n)$ is:
     - $\overline{eq}$, if w_i is $=$ (this is just used here to make the syntax used for equality uniform to that used for predicates in $\mathbf{S}$),
     - $\overline{w_i}$, otherwise;
   - v_{n+i} $(1 \le i \le m)$ is:
     - eq, if w_{n+i} is $=$,
     - w_{n+i}, otherwise.
3. for each relational symbol $a \in \sigma$, the following art(a) rules are in $\Delta_{\mathcal{R}}(\phi)$;
$$U(X) \leftarrow a(X,Y_1,\ldots,Y_{\text{art}(a)-1})$$
$$\ldots$$
$$U(X) \leftarrow a(Y_1,\ldots,Y_{\text{art}(a)-1},X)$$

4. $\Delta_{\mathcal{R}}(\phi)$ contains the two rules

$$t(X) \leftarrow U(X), \qquad e \leftarrow eq(X,Y), \overline{eq}(X,Y)$$

Furthermore, the query γ is simply equal to e.

We remark that $\Delta_{\mathcal{T}}(\phi) \cup \Delta_{\mathcal{R}}(\phi)$ is a ROLE-SAFE CARIN-MARC knowledge base.

Now, given a finite structure D, we define the complementary structure $\overline{D}$ as follows. For each relational symbol $r \in D$ there is a relational symbol $\overline{r}$ in $\overline{D}$ with the same arity as r. Then, for each relational symbol $\overline{r}$ in $\overline{D}$ and for each tuple $\mathbf{t} \in U^{\mathrm{art}(r)}$, $\overline{D} \models \overline{r}(\mathbf{t})$ iff $D \not\models r(\mathbf{t})$. Thus, finally, let $\Delta_{\mathcal{E}}(\phi, D) = D \cup \overline{D}$. We are now ready for the first main result about expressive power of ROLE-SAFE CARIN-MARC.

Theorem 2.1 *For any skolemized NP_1 collection* $\mathbf{D}$ *of finite structures over σ -characterized by a formula ψ of the kind (1)- the* ROLE-SAFE CARIN-MARC *knowledge base $\Delta(\phi, D)$ built according to the above rules is such that a structure D is in* $\mathbf{D}$ *if and only if $\Delta_{\mathcal{T}}(\phi) \cup \Delta_{\mathcal{R}}(\phi) \cup \Delta_{\mathcal{E}}(\phi, D) \not\models \gamma$.*

In other words, the theorem says that each collection of finite structures in skolemized coNP$_1$ is definable by a ROLE-SAFE CARIN-MARC knowledge base $\Delta(\phi, D)$ and query γ.

To allow role axioms to occur in $\Delta_{\mathcal{T}}(\phi)$ enhances the expressive power of the language. Indeed, consider formulae of the form:

$$\phi' = (\exists \mathbf{S}')(\forall \mathbf{X})(\exists \mathbf{Y})(\theta_1(\mathbf{X}, \mathbf{Y}) \vee \cdots \vee \theta_k(\mathbf{X}, \mathbf{Y})), \quad (2)$$

where $\theta_1, \ldots, \theta_k$ are conjunctions of literals involving relational symbols in σ and $\mathbf{S}'$, plus relational symbol "=", and all relational symbols in $\mathbf{S}'$ are constrained to be either monadic or dyadic. Such formulae define a subset of NP$_2$ that contains NP-complete problems. We remind that there are collections of polynomial-time recognizable databases (e.g., the collection of ternary relations with even number of tuples) that are not in NP$_2$ [Fagin, 1990], and that NP$_2$ strictly contains NP$_1$ (e.g., the collection of dyadic relations with even number of tuples is in NP$_2$).

To illustrate the transformation of a formula ψ of the kind (2) into a CARIN-MARC$^=$ knowledge base $\Delta'(\phi', D)$ with axioms on roles and a query γ', we modify the translation provided for the monadic case by adding to the TBox $\Delta'_{\mathcal{T}}$, for each dyadic relational symbol $s' \in \mathbf{S}'$, the following role axioms:

$$\top \sqsubseteq s' \sqcup \overline{s'} \qquad s' \sqsubseteq \neg \overline{s'} \qquad eq \sqsubseteq =$$
$$= \sqsubseteq eq \qquad \top \times \top \sqsubseteq eq \sqcup \overline{eq} \qquad eq \sqsubseteq \neg \overline{eq}$$

We are now ready for our second result about expressive power of CARIN languages.

Theorem 2.2 *For any skolemized NP_2 collection* $\mathbf{D}$ *of finite structures over σ -characterized by a formula ψ' of the kind (2)- the* CARIN-MARC$^=$ *knowledge base $\Delta'(\phi', D)$ built according to the above rules is such that a structure D is in* $\mathbf{D}$ *if and only if $\Delta'_{\mathcal{T}}(\phi') \cup \Delta'_{\mathcal{R}}(\phi') \cup \Delta'_{\mathcal{E}}(\phi', D) \not\models \gamma'$.*

3 Conclusions

In the present paper we showed the possibility of exploiting the formal analysis tools developed in the database field within the context of description logics augmented with rules. In particular we obtained lower bounds for the expressive power of two hybrid languages (Theorems 2.1 and 2.2). Upper bound for the expressiveness of the former language follows from the results of [Levy and Rousset, 1996b]. We still have to investigate the upper bound of expressiveness of the latter language.

In this work we assumed empty ABoxes. Nevertheless the results we presented are valid also if the ABox contains positive atomic assertions such as *red*(*node*1). Furthermore, it is important to stress that by allowing predicates with any arity to appear in the description logic component of knowledge bases, we obtain languages capturing all coNP properties.

Several questions are still open. Regarding the two languages we have analyzed in this paper, (1) determine whether there are queries in coNP$_2$ which cannot be expressed by former language, and (2) determine whether the languages express all (even non-skolemized) queries in coNP$_1$, resp. coNP$_2$. In general, how to define the expressive power over finite formulae possibly denoting infinitely many models?

References

[Abiteboul and Vianu, 1992] S. Abiteboul and V. Vianu. Expressive power of query languages. In J. D. Ullman, editor, *Theoretical Studies in Computer Science*. Academic Press, 1992.

[Baader, 1996] Franz Baader. A formal definition for the expressive power of terminological knowldege representation languages. *J. of Logic and Computation*, 6:33–54, 1996.

[Borgida, 1996] Alexander Borgida. On the relative expressiveness of description logics and predicate logics. *AIJ*, 82:353–367, 1996.

[Buchheit *et al.*, 1993] Martin Buchheit, Francesco M. Donini, and Andrea Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. of Artificial Intelligence Research*, 1:109–138, 1993.

[Cosmadakis, 1993] S. S. Cosmadakis. Logical reducibility and monadic NP. In *Proc. of FOCS-93*. 1993.

[Donini *et al.*, 1991] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. A hybrid system integrating datalog and concept languages. In *Proc. of AI*IA-91*, number 549 in LNAI. Springer-Verlag, 1991. An extended version appeared also in the Working Notes of the AAAI Fall Symposium "Principles of Hybrid Reasoning".

[Eiter *et al.*, 1994] T. Eiter, G. Gottlob, and H. Mannila. Adding Disjunction to Datalog. In *Proc. of PODS-94*, pages 267–278, 1994.

[Fagin, 1974] R. Fagin. Generalized First-Order Spectra and Polynomial-Time Recognizable Sets. In R. M.

Karp, editor, *Complexity of Computation*, pages 43–74. AMS, 1974.

[Fagin, 1975] Ronald Fagin. Monadic generalized spectra. *Zeitschr. f. mathem. Logik und Grundlagen d. Math.*, 21:89–96, 1975.

[Fagin, 1990] R. Fagin. Finite-model theory–a personal perspective. In *Proc. of ICDT-90*, volume 470 of *LNCS*, pages 3–23. Springer-Verlag, 1990.

[Kanellakis, 1990] P. Kanellakis. Elements of relational database theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 17. Elsevier, 1990.

[Levy and Rousset, 1996a] Alon Y. Levy and Marie-Christine Rousset. CARIN: A representation language combining Horn rules and description logics. In *Proc. of ECAI-96*, pages 323–327, 1996.

[Levy and Rousset, 1996b] Alon Y. Levy and Marie-Christine Rousset. The limits on combining recursive Horn rules with description logics. In *Proc. of AAAI-96*, pages 577–584, 1996.

[Reiter, 1984] Raymond Reiter. Towards a logical reconstruction of relational database theory. In M. L. Brodie, J. Mylopoulos, and J. W. Schmidt, editors, *On Conceptual Modelling*. Springer-Verlag, 1984.

[Schlipf, 1990] J. S. Schlipf. The expressive powers of the logic programming sematics. In *Proc. of PODS-90*, pages 196–204, 1990. Expanded version available as University of Cincinnati Computer Science Department Tech. Rep. CIS-TR-90-3.

[Stewart, 1991] I. Stewart. Comparing the Expressibility of Languages Formed Using NP-Complete Operators. *J. of Logic and Computation*, 1(3):305–330, 1991.