Reasoning on Anonymity in Datalog+/-*

Giovanni Amendola[1], Nicola Leone[2], Marco Manna[3], and Pierfrancesco Veltri[4]

1 **DEMACS, University of Calabria, Italy**
`amendola@mat.unical.it`
2 **DEMACS, University of Calabria, Italy**
`leone@mat.unical.it`
3 **DEMACS, University of Calabria, Italy**
`manna@mat.unical.it`
4 **DEMACS, University of Calabria, Italy**
`veltri@mat.unical.it`

Abstract

In this paper we empower the ontology-based query answering framework with the ability to reason on the properties of "known" (non-anonymous) and anonymous individuals. To this end, we extend Datalog+/- with epistemic variables that range over "known" individuals only. The resulting framework, called $\mathsf{datalog}^{\exists,\mathcal{K}}$, offers good and novel knowledge representation capabilities, allowing for reasoning even on the anonymity of individuals. To guarantee effective computability, we define shyK, a decidable subclass of $\mathsf{datalog}^{\exists,\mathcal{K}}$, that fully generalizes (plain) Datalog, enhancing its knowledge modeling features without any computational overhead: OBQA for shyK keeps exactly the same (data and combined) complexity as for Datalog. To measure the expressiveness of shyK, we borrow the notion of uniform equivalence from answer set programming, and show that shyK is strictly more expressive than the DL $\mathcal{ELH}$. Interestingly, shyK keeps a lower complexity, compared to other Datalog+/- languages that can express this DL.

1998 ACM Subject Classification D.1.6 Logic Programming

Keywords and phrases Datalog, query answering, Datalog+/-, ontologies, expressiveness

Digital Object Identifier 10.4230/OASIcs.ICLP.2017.3

1 Introduction

In ontology-based query answering (OBQA), a user query q is evaluated over a logical theory consisting of an extensional database D paired with an ontology Σ. This problem is attracting the increasing attention of scientists in various fields of Computer Science, ranging from *artificial intelligence* [3, 13, 17] to *database theory* [5, 18, 6] and *logic* [22, 4, 19]. In this context, Description Logics [2] and Datalog$^{\pm}$ [10] have been recognized as the two main families of formal knowledge representation languages to specify Σ, while conjunctive queries represent the most common and studied formalism to express q.

In this paper we concentrate on the Datalog$^{\pm}$ family whose intent is to collect all expressive extensions of Datalog which are based on existential quantification, equality-generating dependencies, negative constraints, negation, and disjunction. In particular, the

* The paper has been partially supported by the Italian Ministry for Economic Development (MISE) under project "PIUCultura – Paradigmi Innovativi per l'Utilizzo della Cultura" (n. F/020016/01-02/X27), and under project "Smarter Solutions in the Big Data World (S2BDW)" (n. F/050389/01-03/X32) funded within the call "HORIZON2020" PON I&C 2014-2020.

"plus" symbol refers to any possible combination of these extensions, while the "minus" one imposes at least decidability, as already the presence of existential quantification alone makes OBQA undecidable in the general case [23, 9], also because the ontology universe may be enlarged with infinitely many "anonymous" individuals to satisfy existential rules.

Originally, this family was introduced with the aim of "closing the gap between the Semantic Web and databases" [11] to provide the *Web of Data* with scalable formalisms that can benefit from existing database technologies. And in fact it generalizes well-known subfamilies of Description Logics —such as $\mathcal{EL}$ [8] and *DL-Lite* [1]— collecting the basic tractable languages for OBQA in the context of the Semantic Web and databases. Currently, Datalog$^{\pm}$ has evolved as a major paradigm and an active field of research. As a result, a number of syntactic properties that guarantee decidability by implicitly limiting the generation and the "interaction" among anonymous individuals have been single out: *weak-acyclicity* [16], *guardedness* [9], *linearity* [11], *stickiness* [12], and *shyness* [21].

2 Datalog$^{\pm}$ with epistemic variables

Following the Datalog$^{\pm}$ philosophy, on the one hand we extend the family with a novel knowledge representation feature that allows for consciously reasoning on the properties of "known" (non-anonymous) and anonymous individuals in different ways; on the other hand, we single out sufficient syntactic conditions to ensure decidability. More specifically, we start from a classical well-established setting introduced by [11], where an ontology Σ is a set of $\mathsf{datalog}^{\exists}$ (a.k.a. "existential") rules of the form $\forall\mathbf{X}\forall\mathbf{Y}\,(\phi(\mathbf{X},\mathbf{Y}) \rightarrow \exists\mathbf{Z}\; p(\mathbf{X},\mathbf{Z}))$, and a query $q(\mathbf{X})$ is an expression of the form $\exists\mathbf{Y}\,(\phi(\mathbf{X},\mathbf{Y}), \neg p_1(\mathbf{Z}_1), \ldots, \neg p_k(\mathbf{Z}_k))$, where symbol '$\neg$' stands for default negation (a.k.a. negation as failure). Both in rules and queries, $\phi(\mathbf{X},\mathbf{Y})$ is a conjunction of atoms; also, $p(\mathbf{X},\mathbf{Z})$ and each $p_i(\mathbf{Z}_i)$ are atoms (with $\mathbf{Z}_i \subseteq \mathbf{X}\cup\mathbf{Y}$ required as standard "safety" condition). Then, we enhance the framework with *epistemic variables* (denoted by $\widehat{X}$, $\widehat{Y}$, $\widehat{Z}$, ...) that complement standard variables (denoted by X, Y, Z, ...) adding some interesting modeling capabilities. We call $\mathsf{datalog}^{\exists,\mathcal{K}}$ the resulting language. Roughly, epistemic variables range over "known" (non-anonymous) individuals only; while standard variables range over all individuals.

Consider for example the database $D = \{person(john), person(tim), hasFather(john, tim)\}$, and the $\mathsf{datalog}^{\exists,\mathcal{K}}$ ontology Σ consisting of the following rules:

$$person(X) \rightarrow \exists Y\; hasFather(X,Y) \qquad (\rho_1)$$

$$hasFather(X,\widehat{Y}) \rightarrow hasKnownFather(X) \qquad (\rho_2)$$

The first rule states that every person has a father (note that the father is guaranteed to exists, even if he could be an unknown individual); while the second, using the epistemic variable $\widehat{Y}$, specifies the persons who have a known father. From ρ_1 we derive that also *tim* has a father, but his identity is not known. (Technically, this is represented by some fact $hasFather(tim, \eta)$ where η is a term not occurring in the ontology domain, namely an "anonymous" individual or a "null" in the database terminology.) From ρ_2, $hasKnownFather(john)$ is derived, while $hasKnownFather(tim)$ is not derived as $\widehat{Y}$ ranges over the ontology domain $\{john, tim\}$. Let us now consider the query:

$$q(X) = \exists Y\; hasFather(X,Y),\; \neg hasKnownFather(X),$$

which asks for those people whose father is not known. By evaluating q over $D\cup\Sigma$, we get the answer $X = tim$, as expected since the identity of tim's father is not known; while the father of john is known.

Roughly, epistemic variables behave as the operator K already in use in Description Logics [14]. In this context, expression $\mathsf{K}C$ is interpreted as the set of individuals on the ontology domain that are instances of the concept C in all models, or equivalently the "known" objects which are instances of C. For example, the inclusion axiom $\mathsf{K}C \sqsubseteq D$ of Description Logics can be expressed via the $\mathsf{datalog}^{\exists,\mathcal{K}}$ rule $C(\widehat{X}) \rightarrow D(\widehat{X})$.

3 A decidable and expressive language: shyK

Besides enhancing the KR features of the framework, we want to ensure decidable query answering. To this end, we single out a $\mathsf{datalog}^{\exists,\mathcal{K}}$ language called shyK. Intuitively, consider a database D, a shyK ontology Σ, and a chase step $\langle \rho, h \rangle(I) = I'$ employed in the construction of $chase(D, \Sigma)$ (for more details on the chase procedure, see [20]). The syntactic properties underlying shyK guarantee that: (1) if a standard variable X occurs in two different atoms of the body of ρ, then $h(X)$ is a constant; and (2) if two different standard variables X and Y occur both in the head of ρ and in two different atoms of the body of ρ, then $h(X) = h(Y)$ implies $h(X)$ is a constant.

We reduce the evaluation of conjuntive queries over shyK ontologies to the evaluation of conjunctive queries over shy ontologies.

▶ **Theorem 1.** QEVAL *for conjunctive queries over* shyK *ontologies is:* (*i*) ExpTime*-complete in combined complexity, and* (*ii*) PTime*-complete in data complexity.*

To measure the expressiveness of shyK we compare it with the DL $\mathcal{ELH}$. More precisely, consider an ontology Σ. We say that an ontology Σ' is *equivalent to* Σ if, for each database D over $\mathcal{R}(\Sigma)$, it holds that $chase(D, \Sigma')|_{\mathcal{R}(\Sigma)} = chase(D, \Sigma)$. Hence, a class $\mathcal{C}_1$ is *strictly more expressive* than $\mathcal{C}_2$ if (*i*) for each $\Sigma \in \mathcal{C}_2$ there is $\Sigma' \in \mathcal{C}_1$ being equivalent to Σ, and (*ii*) for some $\Sigma \in \mathcal{C}_1$ there is no $\Sigma' \in \mathcal{C}_2$ being equivalent to Σ.

We show that shyK is strictly more expressive than $\mathcal{ELH}$. In particular, we provide a polynomial-time transformation that maps each $\mathcal{ELH}$ ontology to an equivalent shyK one. Since the reduction is polynomial, this also shows that $\mathcal{ELH}$ is no more succinct than shyK.

▶ **Theorem 2.** shyK *is strictly more expressive than* $\mathcal{ELH}$.

4 Conclusion

In conclusion, we extend datalog rules to deal with both epistemic variables and existential quantification. The resulting framework offers good and novel knowledge representation capabilities, allowing for reasoning even on the anonymity of individuals. We define shyK, a $\mathsf{datalog}^{\exists,\mathcal{K}}$ language that supports epistemic variables, fully generalizes datalog, and that guarantees the decidability of OBQA for conjunctive queries with epistemic variables. Finally, to measure the expressive power of shyK, we borrow the notion of uniform equivalence from answer set programming [15]. Then, we compare shyK with the well-known Description Logic $\mathcal{ELH}$ [7, 8], showing that shyK is strictly more expressive than $\mathcal{ELH}$. Interestingly, shyK keeps a lower computational complexity, compared to other Datalog$^{\pm}$ languages that can express this Description Logic (namely guarded and its extensions).

References

1 Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The dl-lite family and relations. *J. Artif. Intell. Res.*, 36:1–69, 2009.

2 Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

3 Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.

4 Vince Bárány, Georg Gottlob, and Martin Otto. Querying the guarded fragment. *Logical Methods in Computer Science*, 10(2), 2014.

5 Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: A study through disjunctive datalog, csp, and MMSNP. *ACM Trans. Database Syst.*, 39(4):33:1–33:44, 2014.

6 Pierre Bourhis, Marco Manna, Michael Morak, and Andreas Pieris. Guarded-based disjunctive tuple-generating dependencies. *ACM TODS*, 41(4), November 2016.

7 Sebastian Brandt. On subsumption and instance problem in ELH w.r.t. general tboxes. In *Proceedings of DL 2004*, 2004.

8 Sebastian Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and - what else? In *Proceedings of ECAI 2004*, pages 298–302, 2004.

9 Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.

10 Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. Datalog$^{\pm}$: a unified approach to ontologies and integrity constraints. In *Proceedings of ICDT 2009*, pages 14–30, 2009.

11 Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.

12 Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.*, 193:87–128, 2012.

13 Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. *Artif. Intell.*, 195:335–360, 2013.

14 Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf, and Werner Nutt. Adding epistemic operators to concept languages. In *Proceedings of KR 1992*, pages 342–353, 1992.

15 Thomas Eiter and Michael Fink. Uniform equivalence of logic programs under the stable model semantics. In *Proceedings of ICLP 2003*, pages 224–238, 2003.

16 Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

17 Georg Gottlob, Stanislav Kikot, Roman Kontchakov, Vladimir Podolskii, Thomas Schwentick, and Michael Zakharyaschev. The price of query rewriting in ontology-based data access. *Artif. Intell.*, 213:42–59, 2014.

18 Georg Gottlob, Giorgio Orsi, and Andreas Pieris. Query rewriting and optimization for ontological databases. *ACM Trans. Database Syst.*, 39(3):25:1–25:46, 2014.

19 Georg Gottlob, Andreas Pieris, and Lidia Tendera. Querying the guarded fragment with transitivity. In *Proceedings of ICALP 2013*, pages 287–298, 2013.

20 David S. Johnson and Anthony C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. Comput. Syst. Sci.*, 28(1):167–189, 1984. URL: `https://doi.org/10.1016/0022-0000(84)90081-3`, `doi:10.1016/0022-0000(84)90081-3`.

21 Nicola Leone, Marco Manna, Giorgio Terracina, and Pierfrancesco Veltri. Efficiently computable datalog∃ programs. In *Proceedings of KR 2012*, 2012.

22 Héctor Pérez-Urbina, Boris Motik, and Ian Horrocks. Tractable query answering and rewriting under description logic constraints. *J. Applied Logic*, 8(2):186–209, 2010.

23 Riccardo Rosati. The limits of querying ontologies. In *Proceedings of ICDT 2007*, pages 164–178, 2007.