Informativeness of Query Answers for Knowledge Bases (Extended Abstract)

Luca Andolfi, Gianluca Cima, Marco Console and Maurizio Lenzerini

Department of Computer, Control and Management Engineering, 25 Via Ariosto, Rome, 00185

Abstract

This extended abstract summarises our recent work on the informativeness of query answers over Description Logics Knowledge Bases (KBs). We introduce a framework to characterise the information that query answers for KBs can represent and under its lens we study the informative power of current query answering definitions across the literature. Moreover, we also present novel notions of certain and possible answers, showing that they are able to represent a meaningful form of information. Lastly, we study the data complexity properties for relevant problems associated to the answers we introduced.

Keywords

Knowledge Bases, Query Answers, Knowledge Representation, Description Logics.

Query answering over a Knowledge Base (KB) [1, 2, 3, 4, 5, 6] is the problem of extracting the information specified by a query from the models of some KB and returning an object (*answer*) that represents it to the user. We are interested in characterising the properties that are needed to make this object meaningful: indeed, in order to be relevant, query answers should represent as much as possible the information required by the queries issued by users and, at the same time, they should be simple enough. The first aspect is related to their *informativeness*, whereas the second is crucial for their comprehension. We focus on answers expressed as sets of tuples, as this language is universally accepted and understood, and we characterise their informative content introducing novel formal tools tailored to this purpose.

So far, in order to construct answers for queries over KBs, most of the approaches in the literature chose either the *certain* or the *possible* semantics [7, 8] of the answer; according to the former, the answer is constituted by the *answer tuples* of constants satisfying the query in *every model* of the KB, while to the latter in *at least one* of them. Nevertheless, it is well known that in many practical scenarios [9, 10, 11, 12, 13, 14, 15, 16, 17, 18] both certain and possible answers are a lossy approximation of all the information entailed by a query and a KB. Indeed, as the following example shows, they suffer severe shortcomings w.r.t. their *informative content.*

Example 1. *Assuming the predicates* $\mathit{Employee}_{/1}(E)$ *and* $\mathit{hasSupervisor}_{/2}(\mathit{hS})$*, let* $\mathcal{T}$ *be the TBox* $\{E \sqsubseteq \exists hS, \exists hS^{-} \sqsubseteq E, hS \sqsubseteq \neg hS^{-}\}$ *(i.e. each employee has as a supervisor another employee and cannot supervise a supervisor) and the ABox* $\mathcal{D} = \{E(Ava), E(Bea), hS(Bea, Carl)\}$*. For* $q = \{(x, y) \mid \mathit{hS}(x, y)\}$ *over* $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$*, the certain answers are* $\{(Bea, Carl)\}$*. For every model* $\mathcal{I}$ *of* $\mathcal{K}$ *the information that there exist constants* a *and* b *s.t.* $(Ava^{\mathcal{I}}, a)$ *and* (a, b) *satisfy* q *in* $\mathcal{I}$ *is lost in the certain answers. Similar considerations hold for possible answers as well.*

This extended abstract summarises our recent results from [19]: we build a framework to formally determine the informativeness of query answers and we show how to use it to quantify objectively the informative content of different query answers definitions across the literature. This is a core contribution of our work, since there existed no rigorous approach conveying a similar measure of informativeness. Moreover, we address some of the limitations highlighted in Example 1 w.r.t. the

DL 2024: 37th International Workshop on Description Logics, June 18–21, 2024, Bergen, Norway

andolfi@diag.uniroma1.it (L. Andolfi); cima@diag.uniroma1.it (G. Cima); console@diag.uniroma1.it (M. Console); lenzerini@diag.uniroma1.it (M. Lenzerini)

https://www.diag.uniroma1.it/users/luca_andolfi (L. Andolfi)

0009-0004-5528-9224 (L. Andolfi); 0000-0003-1783-5605 (G. Cima); 0009-0004-5526-019X (M. Console); 0000-0003-2875-6187 (M. Lenzerini)

certain and possible answers by providing novel notions of *informative* query answers. Once again, we can achieve this in virtue of our framework, as it sheds light on properties of answer objects needed to increase their informative content. Indeed, equivalent results would be hard to obtain without a tool of this kind.

Specifically, we describe which is the logical behaviour of the information to be represented (*what answers need to represent*), and the core idea behind our framework is the one of characterising accordingly how effectively a given answer object *mimics* this behaviour (*how informative answers are*).

To formally model *what answers need to represent*, we leverage the following construction. First, we introduce a new predicate (not occurring in the KB alphabet) ans_k whose arity k matches the one of the given query q; when evaluating q over a first order interpretation $\mathcal{I}$, the answer to q over $\mathcal{I}$ are constituted by the assertion for ans_k over the constant answer tuples from $\mathcal{I}$ satisfying q (e.g., $\mathsf{ans}_2(Bea, Carl)$ is one of these assertions when evaluating the query in Example 1 over any model of the corresponding KB). Second, this approach is generalised to a KB: the set $q(\mathcal{K}) = \{q(\mathcal{I}) \mid \mathcal{I} \in \mathsf{Mod}(\mathcal{K})\}$ expresses all the information that can be derived from a query q and a KB $\mathcal{K}$, where $\mathsf{Mod}(\mathcal{K})$ are models of $\mathcal{K}$ and $q(\mathcal{I})$ is the answer to q over model $\mathcal{I}$. The construction just introduced allows us to define the behaviour of the set $q(\mathcal{K})$ (i.e., *what answers need to represent*) using properties definable with logical formulae over the predicate ans_k, as demonstrated in the example below.

Example 2. *The set $q(\mathcal{K})$ from Example 1 satisfies the formula $\varphi_1 = \exists x, y.\mathsf{ans}_2(Ava, x) \wedge \mathsf{ans}_2(x,y)$ for every $q(\mathcal{I}) \in q(\mathcal{K})$, because $\mathsf{ans}_2(Ava^{\mathcal{I}}, c_1) \in q(\mathcal{I})$ and $\mathsf{ans}_2(c_1, c_2) \in q(\mathcal{I})$ for each $\mathcal{I} \in \mathsf{Mod}(\mathcal{K})$ and for some constants c_1, c_2. Likewise, $q(\mathcal{K})$ does not satisfy $\varphi_2 = \exists x, y.\mathsf{ans}_2(x,y) \wedge \mathsf{ans}_2(y,x)$ for any $q(\mathcal{I}) \in q(\mathcal{K})$, because there is no model $\mathcal{I}$ of $\mathcal{K}$ and no constants (a, b) from $\mathcal{I}$ s.t. $\mathsf{ans}_2(a, b) \in q(\mathcal{I})$ and $\mathsf{ans}_2(b, a) \in q(\mathcal{I})$.*

Defining *how informative answers are* w.r.t. the set $q(\mathcal{K})$ introduced above constitutes the heart of our approach. The core idea is the one of associating to the objects returned as query answers a formal semantics and leverage it to determine the information (either present in every or at least one object) in $q(\mathcal{K})$ that the answers are able to convey. More precisely, in our framework every answer object that is returned in response to a query over a KB belongs to a family of objects (call it $\mathcal{A}$); each of these families is equipped with a relation specifying the (first order definable) properties their objects satisfy (call it $\models_A$). Below we show the case in which certain answers are used in the framework as definition of answers.

Example 3. *In relation to Example 1, assume the object returned as the answer to q over $\mathcal{K}$ is the set of the certain answers $\Theta = \{\mathsf{ans}_2(Bea, Carl)\}$, where Θ belongs to a family $\mathcal{A}$ of objects constituted by all the ground atoms over the alphabet $\{\mathsf{ans}\}$. In this case $\Theta \models_A \varphi$, iff φ is true over Θ seen as an interpretation. For instance, let $\varphi = \mathsf{ans}_2(Bea, Carl)$. Then it is readily seen that $\Theta \models_A \varphi$.*

Now, we can compare the set $q(\mathcal{K})$ with answer objects and define how similar they are according to the languages of properties (expressible as formulae over $\{\mathsf{ans}\}$) for which they satisfy the same formulae. In order to be as general as needed to accommodate for many query answers definitions, we use the notion of *Query Answering System* (QAS). A QAS is an abstraction of the query answering task that takes queries and KBs, from two languages of queries and KBs $\mathbb{Q}$ and $\mathbb{K}$ resp., and associates to them an answer from a family of objects $\mathcal{A}$ having semantics $\models_A$. This mapping happens in virtue of a function denoted as $eval : \mathbb{Q} \times \mathbb{K} \to \mathcal{A}$. For each query answer definition its informative power comes down to specify the (largest) language of formulae for which it exhibits an identical behaviour w.r.t. the set $q(\mathcal{K})$.

Definition 1. *Let $\mathcal{F}$ be a language of FO formulae over $\{\mathsf{ans}\}$ and $\mathbb{S} = \langle \mathbb{Q}, \mathbb{K}, \langle \mathcal{A}, \models_A \rangle, \mathsf{eval} \rangle$ be a QAS. We say that $\mathbb{S}$ preserves the certain knowledge of $\mathcal{F}$ if, for each $q \in \mathbb{Q}$, $\mathcal{K} \in \mathbb{K}$, and $\varphi \in \mathcal{F}$ it holds that $\mathsf{eval}(q, \mathcal{K}) \models_A \varphi$ iff $q(\mathcal{I}) \models \varphi$ for every $\mathcal{I} \in \mathsf{Mod}(\mathcal{K})$; likewise, $\mathbb{S}$ preserves the possible knowledge of $\mathcal{F}$ if, for each $q \in \mathbb{Q}$, $\mathcal{K} \in \mathbb{K}$ and $\varphi \in \mathcal{F}$, it holds that $\mathsf{eval}(q, \mathcal{K}) \models_A \varphi$ iff there exists $\mathcal{I} \in \mathsf{Mod}(\mathcal{K})$ so that $q(\mathcal{I}) \models \varphi$.*

QASs characterise the information content of query answers: the larger the language $\mathcal{F}$ that a QAS preserves, the more informative such a QAS is. We use the technical tool provided by Definition 1 to quantify and compare the informativeness of query answering definitions in the literature. In this study we consider queries in the language of conjunctive queries (CQ) and union thereof (UCQ), as well as KBs belonging to languages with widely accepted and well-known properties, in which many common description logic languages fall, such as the profiles of OWL 2 and tuple generating dependencies. For simplicity of exposition, we refer our results to the answer objects returned by the various query answering definitions, rather than their corresponding QAS; however, the findings we present are easily extended to QASs.

For certain answers, recall that the objects returned as answers have the shape of the one introduced in Example 3. Answer objects of this flavour preserve the certain knowledge of the language of existential positive formulae over $\{\mathsf{ans}\}$ with ground atoms only and they do not preserve the certain knowledge of those languages where existential variables occur.

Next, we consider an extended form of certain answers, called *minimal partial answers*, recently presented in [20, 21]. We prove that the answers constructed according to this definition preserve the certain knowledge expressible using the existential positive formulae where variables do not repeat across different atoms; moreover, they do not preserve those fragments of existential positive formulae in which these repetitions occur.

Lastly, we consider possible answers: in this case the answer object is constituted by all the assertions over $\{\mathsf{ans}\}$ for the constant answer tuples satisfying the query in at least one model of the KB. This definition preserves the possible knowledge of the language of ground atoms, but it does not preserve their conjunctions.

Finally, we focus on the definition of novel notions of (certain and possible) answers as sets of tuples, which are able to represent meaningful form of information. In particular, our idea consists in defining answers containing tuples sharing the same (unknown) existential individuals, represented through repeated variables. To this end, we say that a set of atoms A is Variable-Connected if for every atom $\alpha \in A$ there is one atom $\beta \in A$ s.t. α and β share at least one variable. Additionally, whenever A contains at most n distinct variables it is called *n-Connected Answer* (n-CA). Now, given a KB $\mathcal{K}$ and a query q, among all the n-CAs we are interested in those that are certain (resp. possible) for q over $\mathcal{K}$; we say that an n-CA A is certain (resp. possible) for q over $\mathcal{K}$ if for every (resp. some) $\mathcal{I} \in \mathsf{Mod}(\mathcal{K})$ there is a constant-preserving homomorphism from A to $q(\mathcal{I})$.

Example 4. *Consider again Example 1. For $n = 0$ the certain 0-CA is $\{\mathsf{ans}_2(\mathit{Bea,Carl})\}$; if $n = 1$, some certain 1-CAs are $\{\mathsf{ans}_2(\mathit{Carl}, x_1)\}$ and $\{\mathsf{ans}_2(\mathit{Ava}, y_1)\}$. For $n = 2$, some certain 2-CAs are $\{\mathsf{ans}_2(\mathit{Carl}, x_1), \mathsf{ans}_2(x_1, x_2)\}$ and $\{\mathsf{ans}_2(\mathit{Ava}, y_1), \mathsf{ans}_2(y_1, y_2)\}$.*

Our novel answer, called certain (resp. possible) n-answer, collects together all the n-CA that are certain (resp. possible and constant-free) for the query q over KB $\mathcal{K}$.

Definition 2. *Let $A = \bigcup_{i=1}^{m} A_i$, where $A_1, \ldots, A_m$ are n-CAs that share no variables (resp. share no variables and mention no constants). The set A is a certain (resp. possible) n-answer for q over $\mathcal{K}$ if: A_i is certain (resp. possible) for q over $\mathcal{K}$, for each $i \in \{1, \ldots, m\}$ and for every n-CA B that is certain (resp. possible) for q over $\mathcal{K}$ there is $j \in \{1, \ldots, m\}$ and a bijection h s.t. $h(B) = A_j$.*

Using the tools from our framework we prove that certain n-answers can preserve the certain knowledge that can be expressed using existential positive formulae in which at most n distinct variables occur; on the other hand, possible n-answers preserve the possible knowledge expressed by constant-free positive formulae with at most n different variables.

We then studied the data complexity versions [22, 23, 24] of relevant decision problems for n-CAs, as the ones of checking whether an n-CA is certain and possible for $q \in \mathsf{UCQ}$ and a KB $\mathcal{K}$. Notably, these problems can be reduced to Boolean UCQ entailment and consistency check over $\mathcal{K}$, resp. Our results imply that, given any UCQ q and KB $\mathcal{K}$ for which Boolean UCQ entailment (resp., consistency check) is decidable, we can construct a set of certain (resp. possible) n-answers to q over $\mathcal{K}$ by means of calls to an answer recognition oracle. We investigated the *non-redundancy* check as well. Consider the case of certain n-answers: intuitively, given variable x, constants a, b, c and the certain 1-answers $A = \{\langle a, x\rangle, \langle x, c\rangle\}$ and $B = \{\langle a, b\rangle, \langle b, c\rangle\}$, then A is redundant, as B conveys more information and $|B| \leq |A|$. Resorting to the query language $\mathsf{EQL\text{-}Lite}$(UCQ), the non-redundancy check has the same data complexity as EQL-entailment [25, 26]. Similar results are obtained for the non-redundancy check of possible n-answers.

Regarding the open problems for our work, a natural extension of the results we presented is the one of providing a definition of answers that is powerful enough to preserve the certain or possible knowledge of arbitrary existential positive formulae. This problem is far from trivial, because it forces us to rethink the language of the query answers: indeed, one notable negative result we showed in [19], regarding the expressive power of answers as sets of tuples, suggests that the solution requires a new representation of answers based on logical theories. Second, another research direction involves the design of efficient enumeration algorithms working with our novel notion of answers and their implementation in practice. In addition, the framework could also be used to improve the informativeness of answers in many different scenarios, such as those involving queries with aggregation or Consistent Query Answering.

Acknowledgments

This work has been supported by MUR under the PNRR project FAIR (PE0000013) and by the EU under the H2020-EU.2.1.1 project TAILOR (grant id. 952215).

References

[1] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family, Journal of Automated Reasoning 39 (2007) 385–429.

[2] M. Bienvenu, M. Ortiz, Ontology-mediated query answering with data-tractable description logics, in: Reasoning Web. Semantic Technologies for Intelligent Data Access – Eleventh International Summer School Tutorial Lectures (RW 2015), volume 9203 of *Lecture Notes in Computer Science*, 2015, pp. 218–307.

[3] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: Query answering under expressive relational constraints, Journal of Artificial Intelligence Research 48 (2013) 115–174.

[4] T. Catarci, M. Scannapieco, M. Console, C. Demetrescu, My (fair) big data, in: J. Nie, Z. Obradovic, T. Suzumura, R. Ghosh, R. Nambiar, C. Wang, H. Zang, R. Baeza-Yates, X. Hu, J. Kepner, A. Cuzzocrea, J. Tang, M. Toyoda (Eds.), 2017 IEEE International Conference on Big Data (IEEE BigData 2017), Boston, MA, USA, December 11-14, 2017, IEEE Computer Society, 2017, pp. 2974–2979. URL: https://doi.org/10.1109/BigData.2017.8258267. doi:10.1109/BIGDATA.2017.8258267.

[5] T. Lukasiewicz, E. Malizia, M. V. Martinez, C. Molinaro, A. Pieris, G. I. Simari, Inconsistency-tolerant query answering for existential rules, Artificial Intelligence 307 (2022) 103685.

[6] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.

[7] T. Imielinski, W. Lipski, Jr, Incomplete information in relational databases, Journal of the ACM 31 (1984) 761–791.

[8] W. Lipski, On semantic issues connected with incomplete information databases, ACM Trans. Database Syst. 4 (1979) 262–296. URL: https://doi.org/10.1145/320083.320088. doi:10.1145/320083.320088.

[9] M. Console, P. Guagliardo, L. Libkin, Approximations and refinements of certain answers via many-valued logics, in: Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR, 2016.

[10] M. Console, P. Guagliardo, L. Libkin, Propositional and predicate logics of incomplete information, in: Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR, 2018.

[11] M. Calautti, M. Console, A. Pieris, Benchmarking approximate consistent query answering, in: L. Libkin, R. Pichler, P. Guagliardo (Eds.), PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, 2021.

[12] M. Console, P. Guagliardo, L. Libkin, E. Toussaint, Coping with incomplete data: Recent advances, in: Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS, ACM, 2020, pp. 33–47.

[13] L. Libkin, Certain answers as objects and knowledge, Artificial Intelligence 232 (2016) 1–19.

[14] C. Civili, L. Libkin, Approximating certainty in querying data and metadata, in: Proceedings of the Sixteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2018), 2018, pp. 582–591.

[15] A. Borgida, D. Toman, G. E. Weddell, On referring expressions in query answering over first order knowledge bases, in: Proceedings of the Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2016), 2016, pp. 319–328.

[16] A. Borgida, D. Toman, G. E. Weddell, Concerning referring expressions in query answers, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017), 2017, pp. 4791–4795.

[17] M. Arenas, J. Pérez, J. L. Reutter, Data exchange beyond complete data, Journal of the ACM 60 (2013) 28:1–28:59. URL: https://doi.org/10.1145/2508028.2505985. doi:10.1145/2508028.2505985.

[18] G. Grahne, A. Onet, Representation systems for data exchange, in: A. Deutsch (Ed.), 15th International Conference on Database Theory, ICDT '12, Berlin, Germany, March 26-29, 2012, ACM, 2012, pp. 208–221. URL: https://doi.org/10.1145/2274576.2274599. doi:10.1145/2274576.2274599.

[19] Andolfi, Cima, Console, Lenzerini, What does a query answer tell you? informativeness of query answers for knowledge bases, Proceedings of the AAAI Conference on Artificial Intelligence (2024).

[20] C. Lutz, M. Przybylko, Efficiently enumerating answers to ontology-mediated queries, in: Proceedings of the Forty-First ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2022), 2022, pp. 277–289.

[21] C. Lutz, M. Przybylko, Efficient answer enumeration in description logics with functional roles, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023), 2023, pp. 6483–6490.

[22] M. Y. Vardi, The complexity of relational query languages (extended abstract), in: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC 1982), 1982, pp. 137–146.

[23] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Data complexity of query answering in description logics, Artificial Intelligence 195 (2013) 335–360. doi:10.1016/j.artint.2012.10.003.

[24] J.-F. Baget, M.-L. Mugnier, S. Rudolph, M. Thomazo, Walking the complexity lines for generalized guarded existential rules, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), 2011, pp. 712–717.

[25] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, EQL-Lite: Effective first-order query processing in description logics, in: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007), 2007, pp. 274–279.

[26] G. Cima, M. Console, M. Lenzerini, A. Poggi, Epistemic disjunctive datalog for querying knowledge bases, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23), 2023, pp. 6280–6288.