Fitting Description Logic Ontologies to ABox and Query Examples (Extended Abstract)

Maurice Funk[1,2], Marvin Grosser[1,2] and Carsten Lutz[1,2]

[1] *Leipzig University*
[2] *ScaDS.AI Center Dresden/Leipzig*

Keywords
Description Logic, Ontology-Mediated Querying, Ontology Fitting

1. Introduction

In many areas of computer science and AI, a fundamental problem is to fit a formal object to a given collection of examples. In inductive program synthesis, for instance, one wants to find a program that complies with a given collection of examples of input-output behavior [1]. In machine learning, fitting a model to a given set of examples is closely linked to PAC-style generalization guarantees [2]. And in database research, the query-by-example paradigm asks to find a query that fits a given set of data examples [3].

In this extended abstract, we study the problem of fitting an ontology formulated in a description logic (DL) to a given collection of positive and negative examples. Our concrete setting is motivated by the paradigm of ontology-mediated querying where data is enriched by an ontology that provides domain knowledge, aiming to return more complete answers and to bridge heterogeneous representations in the data [4, 5]. Guided by this application, we use labeled examples that take the form $(\mathcal{A}, q)$ where $\mathcal{A}$ is an ABox (in other words: a database) and q is a Boolean query. We then seek an ontology $\mathcal{O}$ that satisfies $\mathcal{A} \cup \mathcal{O} \models q$ for all positive examples $(\mathcal{A}, q)$ and $\mathcal{A} \cup \mathcal{O} \not\models q$ for all negative examples $(\mathcal{A}, q)$. The fact that q is required to be Boolean is not a restriction since our queries may contain individual constants from the ABox.

Example 1. *Consider the positively labeled examples*

$$(\{\mathsf{authorOf}(a, b), \mathsf{Publication}(b)\},\ \mathsf{Author}(a)),$$
$$(\{\mathsf{Reviewer}(a)\},\ \exists x\, \mathsf{reviews}(a, x) \wedge \mathsf{Publication}(x)),$$
$$\textit{and}\ (\{\mathsf{Publication}(a)\},\ \mathsf{Confpaper}(a) \vee \mathsf{Jarticle}(a)).$$

An $\mathcal{ALC}$-ontology that fits these examples (with no negative examples) is

$$\mathcal{O} = \{\ \exists \mathsf{authorOf}.\mathsf{Publication} \sqsubseteq \mathsf{Author}, \qquad \mathsf{Reviewer} \sqsubseteq \exists \mathsf{reviews}.\mathsf{Publication},$$

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland
maurice.funk@uni-leipzig.de (M. Funk); grosser@informatik.uni-leipzig.de (M. Grosser); clu@informatik.uni-leipzig.de (C. Lutz)
0000-0003-1823-9370 (M. Funk); 0009-0000-1969-205X (M. Grosser); 0000-0002-8791-6702 (C. Lutz)

$$\text{Publication} \sqsubseteq \text{Confpaper} \sqcup \text{Jarticle} \qquad \}.$$

There are, however, many other fitting $\mathcal{ALC}$-ontologies, including $\mathcal{O}_\bot = \{\top \sqsubseteq \bot\}$ and, say, $\mathcal{O}' = \mathcal{O} \cup \{\mathsf{Author} \sqsubseteq \exists \mathsf{authorOf}.\mathsf{Reviewer}\}$. We can make both of them non-fitting by adding the negative example $(\{\mathsf{Author}(a)\},\ \exists x\, \mathsf{authorOf}(a, x) \wedge \mathsf{Reviewer}(x))$.

A main application of fitting ontologies is to assist with ontology construction and engineering. This is in the spirit of several other proposals that have the same aim, such as ontology construction and completion using formal concept analysis [6, 7, 8] and Angluin's framework of exact learning [9], see also the survey of these and related approaches in [10]. We remark that there is a large literature on fitting DL concepts (rather than ontologies) to a collection of examples, sometimes referred to as concept learning. These have been investigated from a practical angle [11, 12, 13], and from a foundational perspective [14, 15, 16, 17]. Concepts can be viewed as the building blocks of an ontology and in fact concept fitting also has the support of ontology engineering as a main aim. The techniques needed for concept fitting and ontology fitting are, however, quite different, and to the best of our knowledge, fitting problems for ontologies have not yet been studied.

As ontology languages, we concentrate on the expressive yet fundamental DLs $\mathcal{ALC}$ and $\mathcal{ALCI}$, and as query languages for examples we consider atomic queries (AQs), conjunctive queries (CQs), full CQs (CQs without quantified variables), and unions of conjunctive queries (UCQs).

We formally define what we mean by ontology fitting. Let $\mathcal{Q}$ be a query language such as $\mathcal{Q} = \text{AQ}$ or $\mathcal{Q} = \text{CQ}$. An *ABox-$\mathcal{Q}$ example* is a pair $(\mathcal{A}, q)$ with $\mathcal{A}$ a ABox[1] and q a query from $\mathcal{Q}$ such that all individual names that appear in q are from $\mathsf{ind}(\mathcal{A})$, the individuals appearing in $\mathcal{A}$. By a *collection of labeled examples* we mean a pair $E = (E^+, E^-)$ of finite sets of examples. The examples in E^+ are the *positive examples* and the examples in E^- are the *negative examples*. We say that an $\mathcal{ALC}$ or $\mathcal{ALCI}$ ontology $\mathcal{O}$ *fits* E if $\mathcal{A} \cup \mathcal{O} \models q$ for all $(\mathcal{A}, q) \in E^+$ and $\mathcal{A} \cup \mathcal{O} \not\models q$ for all $(\mathcal{A}, q) \in E^-$.

Let $\mathcal{L}$ be an ontology language, such as $\mathcal{L} = \mathcal{ALCI}$, and $\mathcal{Q}$ a query language. Then *($\mathcal{L}$,$\mathcal{Q}$)-ontology fitting* is the problem to decide, given as input a collection of labeled ABox-$\mathcal{Q}$ examples E, whether E admits a fitting $\mathcal{L}$-ontology.

For all of the resulting combinations, we provide effective characterizations and determine the precise complexity of $(\mathcal{L}, \mathcal{Q})$-ontology fitting. The algorithms that we use to prove the upper bounds are able to produce concrete fitting ontologies.

2. Main Contributions

As a starting point, we study an ontology fitting problem in which the examples are only ABoxes and where we seek an ontology that is consistent with the positive examples and inconsistent with the negative ones. To characterize fitting existence for these consistency examples, we make use of the established connection between ontology-mediated querying and constraint satisfaction problems (CSPs) from [18], and obtain the following.

[1] We do not admit compound concepts in ABoxes.

Theorem 1. *Let $E = (E^+, E^-)$ be a collection of labeled ABox examples, $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$, and $\mathcal{A}^+ = \biguplus E^+$. Then the following are equivalent:*

1. *E admits a fitting $\mathcal{L}$-ontology;*
2. *$\mathcal{A} \not\to \mathcal{A}^+$ for all $\mathcal{A} \in E^-$.*

This characterization directly provides a coNP algorithm to decide fitting existence. Intuitively, a ontology that fits the examples can be derived from $\mathcal{A}^+$. We obtain a corresponding lower bound via reduction from the digraph homomorphism problem.

For ABox-AQ examples, the role of the positive and negative examples reverses, as now a fitting ontology $\mathcal{O}$ must be consistent with the ABox $\mathcal{A}$ in a negative example $(\mathcal{A}, Q(a))$, as otherwise $\mathcal{A} \cup \mathcal{O} \models Q(a)$. Additionally, the positive examples act as "rules", meaning that for some positive example $(\mathcal{A}, Q(a))$, whenever $\mathcal{A}$ can be homomorphically found in $\mathcal{A}^- := \biguplus_{(\mathcal{A},Q(a))\in E^-} \mathcal{A}$, any fitting ontology must derive Q at the image of a.[2] To account for this, we introduce the notion of *completions* which enrich $\mathcal{A}^-$ with additional concept assertions and enable us to precisely characterize fitting existence in the setting of AQs. Let $E = (E^+, E^-)$ be a collection of labeled ABox-AQ examples. A *completion* for E is an ABox $\mathcal{C}$ that extends $\mathcal{A}^-$ by assertions of the form $Q(b)$, with $b \in \mathsf{ind}(\mathcal{A}^-)$ and Q a concept name that occurs as an AQ in E^+.

Theorem 2. *Let $E = (E^+, E^-)$ be a collection of labeled ABox-AQ examples and let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then the following are equivalent:*

1. *E admits a fitting $\mathcal{L}$-ontology;*
2. *there is a completion $\mathcal{C}$ for E such that*
   a) *for all $(\mathcal{A}, Q(a)) \in E^+$: if h is a homomorphism from $\mathcal{A}$ to $\mathcal{C}$, then $Q(h(a)) \in \mathcal{C}$;*
   b) *for all $(\mathcal{A}, Q(a)) \in E^-$: $Q(a) \notin \mathcal{C}$.*

Note that an algorithm that directly follows this characterization yields a Σ_2^p upper bound. We obtain a coNP upper bound via a more careful algorithm that does not blindly guess a suitable completion, but constructs one step-by-step. For this the algorithm starts with $\mathcal{A}^-$, and then extends it by guessing, for some positive ABox-AQ example $(\mathcal{A}, Q(a))$, a homomorphisms h from $\mathcal{A}$ to $\mathcal{A}^-$, and then adding $Q(h(a))$. We show coNP-hardness using a similar reduction as in the consistency based setting.

Theorem 3. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then ($\mathcal{L}$, AQ)-ontology fitting is* coNP*-complete.*

Ontology fitting for ABox-FullCQ examples has similar properties as ABox-AQ case. One notable difference is that ABox-FullCQ examples can force fitting ontologies to be inconsistent with their ABoxes.

Example 2. *Consider a positive ABox-FullCQ example $(\mathcal{A}, r(a, a))$ with $r(a, a) \notin \mathcal{A}$. Every $\mathcal{ALCI}$-ontology $\mathcal{O}$ with $\mathcal{A} \cup \mathcal{O} \models r(a, a)$ must be inconsistent with $\mathcal{A}$.*

[2] The homomorphisms used here are not required to be the identity on ABox individuals (which would, in fact, trivialize them).

Thus, we arrive at a characterization that extends the ABox-AQ case with considerations for consistency. A modest modification of the AQ-algorithm then shows that ($\mathcal{L}$, FullCQ)-ontology fitting is coNP-complete. We remark that the obtained complexities for ontology fitting are lower than the complexities of the associated query entailment problems, which are ExpTime-complete for the cases discussed so far [19].

For ABox-CQ and ABox-UCQ examples, the intuition that positive examples behave like "rules" persists, but the presence of quantified variables results in higher expressive power. In fact, positive examples $(\mathcal{A}, q)$ now behave similarly to existential rules: if $\mathcal{A}$ is homomorphically found somewhere in the completion, then q must also be found there in a certain slightly unusual sense made precise in the paper that, notably, treats quantified variables in q in a similar way as existentially quantified variables in the head of an existential rule. It is thus easy to enforce that the completion contains, say, an infinite path. The completions that we construct in the CQ/UCQ case are thus ABoxes that extend $\mathcal{A}^-$ with potentially infinite tree-shaped components that are either rooted in an individual in $\mathcal{A}^-$ or disconnected. They thus take the same form as forest models which are well-known from algorithms for UCQ entailment.[3]

Example 3. *Consider the collection of labeled ABox-CQ examples $E = (E^+, E^-)$ where $E^+ = \{(\mathcal{A}, \exists x\, r(a,x) \wedge A(x))\}$, $E^- = \{(\mathcal{A}, \exists x \exists y\, r(a,x) \wedge r(x,y))\}$, and $\mathcal{A} = \{A(a)\}$. Any completion $\mathcal{C}$ of $\mathcal{A}$ contains $\mathcal{A}^- = \mathcal{A}$. Hence, a homomorphism of $\mathcal{A}$ into $\mathcal{C}$ is found, and to satisfy the positive example viewed as an existential rule $\mathcal{C}$ must contain an r-successor b of a with $A(b) \in \mathcal{C}$. There is thus another homomorphism from $\mathcal{A}$ to $\mathcal{C}$ that maps a to b and thus b must have an r-successor c. While in principle this continues indefinitely, already at this point we have satisfied the query from the negative example. By the characterization given in the full paper, this implies that E does not admit a fitting $\mathcal{ALC}$ or $\mathcal{ALCI}$ ontology.*

As a consequence of this effect, the computational complexity of fitting existence turns out to be much higher: 2ExpTime complete for both CQ and UCQ examples, no matter whether we want to fit an $\mathcal{ALC}$- or $\mathcal{ALCI}$-ontology. The upper bound is derived by a mosaic algorithm. The lower bound for $\mathcal{ALCI}$ is obtained via a reduction from query entailment and the lower bound for $\mathcal{ALC}$ is shown via a reduction from the word problem of exponentially space-bounded alternating Turing machines.

Theorem 4. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ and $\mathcal{Q} \in \{CQ, UCQ\}$. Then $(\mathcal{L}, \mathcal{Q})$-ontology fitting is 2ExpTime-complete.*

For $\mathcal{ALCI}$, the complexity thus coincides with that of query entailment, which is 2ExpTime-complete both for CQs and UCQs [20]. For $\mathcal{ALC}$, the complexity of the fitting problems is higher than that of the associated entailment problems, which are both ExpTime-complete [20]. The full paper [21] summarized in this extended abstract contains full proof details.

Acknowledgements. The third author was supported by DFG project LU 1417/4-1.

[3]In the full paper, we actually represent completions as forest models rather than as ABoxes.

References

[1] S. Jacindha, G. Abishek, P. Vasuki, Program synthesis—a survey, in: Computational Intelligence in Machine Learning, Springer Nature Singapore, Singapore, 2022, pp. 409–421.

[2] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning - From Theory to Algorithms, Cambridge University Press, 2014. URL: http://www.cambridge.org/de/academic/subjects/computer-science/pattern-recognition-and-machine-learning/understanding-machine-learning-theory-algorithms.

[3] H. Li, C.-Y. Chan, D. Maier, Query from examples: An iterative, data-driven approach to query construction, Proc. VLDB Endow. 8 (2015) 2158–2169. URL: https://doi.org/10.14778/2831360.2831369. doi:10.14778/2831360.2831369.

[4] M. Bienvenu, M. Ortiz, Ontology-mediated query answering with data-tractable description logics, in: Proc. of Reasoning Web, volume 9203 of *LNCS*, Springer, 2015, pp. 218–307. URL: https://doi.org/10.1007/978-3-319-21768-0_9. doi:10.1007/978-3-319-21768-0_9.

[5] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: A survey, in: Proc. of IJCAI, ijcai.org, 2018, pp. 5511–5519. URL: https://doi.org/10.24963/ijcai.2018/777. doi:10.24963/IJCAI.2018/777.

[6] F. Baader, B. Ganter, B. Sertkaya, U. Sattler, Completing description logic knowledge bases using formal concept analysis, in: M. M. Veloso (Ed.), Proc. of IJCAI 2007, 2007, pp. 230–235.

[7] F. Baader, F. Distel, Exploring finite models in the description logic, in: S. Ferré, S. Rudolph (Eds.), Proc. of ICFCA 2009, volume 5548 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 146–161.

[8] F. Kriegel, Efficient axiomatization of OWL 2 EL ontologies from data by means of formal concept analysis, in: M. J. Wooldridge, J. G. Dy, S. Natarajan (Eds.), Proc. of AAAI 2024, AAAI Press, 2024, pp. 10597–10606.

[9] B. Konev, C. Lutz, A. Ozaki, F. Wolter, Exact learning of lightweight description logic ontologies, J. Mach. Learn. Res. 18 (2017) 201:1–201:63.

[10] A. Ozaki, Learning description logic ontologies: Five approaches. where do they stand?, Künstliche Intell. 34 (2020) 317–327.

[11] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Mach. Learn. 78 (2010) 203–250.

[12] L. Bühmann, J. Lehmann, P. Westphal, S. Bin, DL-Learner structured machine learning on semantic web data, in: Proc. of WWW, ACM, 2018, pp. 467–471. URL: https://doi.org/10.1145/3184558.3186235. doi:10.1145/3184558.3186235.

[13] G. Rizzo, N. Fanizzi, C. d'Amato, Class expression induction as concept space exploration: From DL-Foil to DL-Focl, Future Gener. Comput. Syst. 108 (2020) 256–272. URL: https://doi.org/10.1016/j.future.2020.02.071. doi:10.1016/J.FUTURE.2020.02.071.

[14] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: Proc. of IJCAI 2019, ijcai.org, 2019, pp. 1682–1688. URL: https://doi.org/10.24963/ijcai.2019/233. doi:10.24963/IJCAI.2019/233.

[15] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Logical separability of incomplete data under

ontologies, in: Proc. of KR, 2020, pp. 517–528. URL: https://doi.org/10.24963/kr.2020/52. doi:10.24963/KR.2020/52.

[16] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Separating data examples by description logic concepts with restricted signatures, in: Proc. of KR, 2021, pp. 390–399. URL: https://doi.org/10.24963/kr.2021/37. doi:10.24963/KR.2021/37.

[17] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Logical separability of labeled data examples under ontologies, Artif. Intell. 313 (2022) 103785. URL: https://doi.org/10.1016/j.artint.2022.103785. doi:10.1016/J.ARTINT.2022.103785.

[18] M. Bienvenu, B. ten Cate, C. Lutz, F. Wolter, Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP, ACM Trans. Database Syst. 39 (2014) 33:1–33:44. URL: https://doi.org/10.1145/2661643. doi:10.1145/2661643.

[19] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[20] C. Lutz, Two upper bounds for conjunctive query answering in SHIQ, in: Proc. of DL 2008, volume 353 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2008. URL: https://ceur-ws.org/Vol-353/Lutz.pdf.

[21] M. Funk, M. Grosser, C. Lutz, Fitting description logic ontologies to ABox and query examples, in: Proc. of KR, 2025. To appear.